Aalborg Universitet

AALBORG UNIVERSITY

DENMARK

Embedded Controller Design for Pig Stable Ventilation Systems

Jessen, Jan Jakob

Publication date:
2007

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Jessen, J. J. (2007). Embedded Controller Design for Pig Stable Ventilation Systems. Aalborg: Department of Control Engineering, Aalborg University.

Embedded Controller Design for Pig Stable Ventilation Systems

Ph.D. Thesis

Jan Jakob Jessen

Draft version, submitted for Ph.D. defence

ii

Preface and Acknowledgements

The subject of the thesis is in the field of climate control in pig stables, focusing both on the development environment for such systems and control algorithms. The thesis is mainly aimed to investigate a new climate control concept based on the idea of *zones* and how to implement this concept in future climate control systems. The main focus of this thesis is application driven but theoretical contributions are given as well.

<div align="right">

Aalborg University, January 2007
Jan Jakob Jessen

</div>

Preface and Acknowledgements

Summary

This thesis focuses on zone based climate control in pig stables and how to implement climate controllers in a new range of products. The presented controllers are based on simple models of climate dynamics and simple models of actuators. The implementation uses graphical point and click features from the Mathworks' range of products and automatic code generation. It is furthermore shown how to build new climate control systems based on cheap and readily available hardware and software. An early result for performing system identification for zone based climate dynamics is also presented.

The thesis is a collection of eight papers. The first two papers deal with the implementation of controllers and how to integrate the development of controllers into a complete framework that can provide different services e.g. remote monitoring. The same framework is also capable of automatically generating source code for the actual target platform, on which the climate controller is expected to execute. The third paper also deals with the development cycle of controllers, showing how to build a graphical user interface for point and click modelling of zone based climate dynamics. The next two papers present an early result for performing system identification for zone based climate dynamics, based on an idea of guaranteed internal flow directions. Paper 6 presents a verified stable distributed temperature controller for pig stables divided into zones. Paper 7 is an expanded journal version of paper 6. Paper 8 presents a distributed temperature and humidity controller based on the ideas presented in papers 6 and 7, but where the controller synthesis is done automatically via a presented tool chain.

Keywords: Climate Modelling, Climate Control, Livestock Buildings, Embedded Software, Game Theory, Networked Control, Distributed Systems, Automatic Code Generation, Matlab, Simulink, Real-Time Workshop, Linux, COTS, System Identification, Parameter Estimation, Hybrid Systems

Summary

Sammenfatning

Titel: *Design af Indlejret Regulator til Ventilationssystemer til Grisestalde.*

Denne Ph.d. afhandling omhandler zonebaseret klimaregulering i grisestalde og hvorledes dette implementeres i nye produkter. De præsenterede regulatorer er baseret på simple modeller af henholdsvis klimadynamik og aktuatorer. Implementeringen er baseret på grafiske peg og klik værktøjer fra Mathworks og automatisk kodegenerering. Det er samtidig vist hvorledes nye klimareguleringssystemer kan baseres på billigt og lettilgængeligt software og hardware. Ydermere præsenteres et foreløbigt resultat til at foretage parameterestimering for zonebaseret klimadynamik.

Afhandlingen består af otte artikler. De første to artikler omhandler implementering af regulatorer og hvorledes udviklingen af disse kan integreres i et framework, der kan tilbyde forskellige services f.eks. fjernovervågning. Det samme framework er i stand til automatisk at generere kildekode til den platform, hvor klimaregulatoren skal eksekveres på. Den tredje artikel omhandler udviklingen af et udviklingsmiljø for grafisk peg og klik udvikling af zonebaserede klimaregulatorer. De følgende to artikler præsenterer et foreløbigt resultat for systemidentifikation af zonebaseret klimadynamik, baseret på en ide om at fastholde retningen af intern luftretning i stalden. Artikel seks omhandler en bevist stabil distribueret temperaturregulator for en grisestald inddelt i zoner. Artikel syv er en udvidet tidsskriftsudgave af artikel seks. Den ottende og sidste artikel omhandler en distribueret temperatur og luftfugtighedsregulator baseret på ideerne fra artikel seks og syv. Syntesen af regulatoren i artikel otte foretages automatisk via en præsenteret tool chain.

Contents

CONTENTS

Contents

Contents

CONTENTS

NOMENCLATURE

T^{i}	Indoor Temperature	[°C]
T^{amb}	Outdoor Temperature	[°C]
ρ	Density	[kg/m^3]
c	Specific heat capacity	[J/(kg°C)]
C	Heat capacity	[J/°C]
V	Volume	[m^3]
A	Area	[m^2]
Q	Ventilation rate	[m^3/s]
ϕ	Heat	[W]
U	U-value for building material	[W/(m^2°C)]
Φ	Solar radiation	[W/m^2]
H^{i}	Relative indoor humidity	[%]
H^{amb}	Relative outdoor humidity	[%]
h^{i}	Absolute indoor humidity	[kg$_{\mathrm{water}}$/kg$_{\mathrm{air}}$]
h^{amb}	Absolute outdoor humidity	[kg$_{\mathrm{water}}$/kg$_{\mathrm{air}}$]
P^{amb}	Ambient pressure	[Pa]
P^{i}	Indoor pressure	[Pa]
G^{i}	Indoor CO$_2$ concentration	[ppm]

Table 1: Notation

Nomenclature

Introduction

The work presented in this thesis addresses the field of climate control in pig stables, covering aspects from analysis of, and a development environment for controllers to actual implementation. The aim has been to develop new climate control algorithms taking into account the challenges for not only Danish pig production, but also the manufacturers of climate control systems.

1.1 Background and Motivation

Skov A/S is a manufacturer and international provider of climate control systems for pig and poultry stables. The product portfolio ranges from ventilation components, animal production monitoring, cleaning of livestock building exhaust air and climate control systems. The company was started in 1954 by the brothers Kristen and Kjeld Skov with a husbandry production of cattle and pigs. In 1966 the company started selling ventilation systems for pig and poultry stables, and through numerous stages of development cycles for both products and the company, Skov evolved into the company it is today.

In 2003 Skov made a strategic decision to boost their research and development activities, initiating a number of research projects in cooperation with different Danish research institutes including the newly started Center for Embedded Software Systems (CISS) at Aalborg University. The different projects are each part of a larger vision for future production of pigs, not only in Denmark but worldwide.

1.2 Vision for Future Pig Production

Denmark is reputed as being one of the most wage intensive countries in the world, and in the global market Danish industry in general focuses on knowledge based products

and a high level of production automation. The challenges posed by globalized competition has launched the idea of future animal husbandry characterized by automation and very little human interaction.

The vision for future pig production covers both climate and production control, with an overall ambition of increasing profit for the farmer while maintaining an acceptable *welfare level* for the animals (Hansen et al., 2005). The idea is to have a number of control loops with the innermost being the climate controller. This control loop takes care of maintaining the current climate within well defined boundaries on the basis of climate measurements e.g. temperature and humidity. The second loop is a production loop, which is responsible for generating profit for the farmer. The production loop should ensure that the pigs fulfill the requirements from the slaughter house with respect to weight and lipid level. The final loop is the welfare controller, which ensures the well being of the animals.

The presented vision corresponds to the daily life of a farmer today, except that *he* acts as both the production and welfare controller i.e. it is the farmer's responsibility to evaluate the production status and well being of the animals and act upon his observations. Figure 1.1 illustrates the current practice in today's animal production systems.
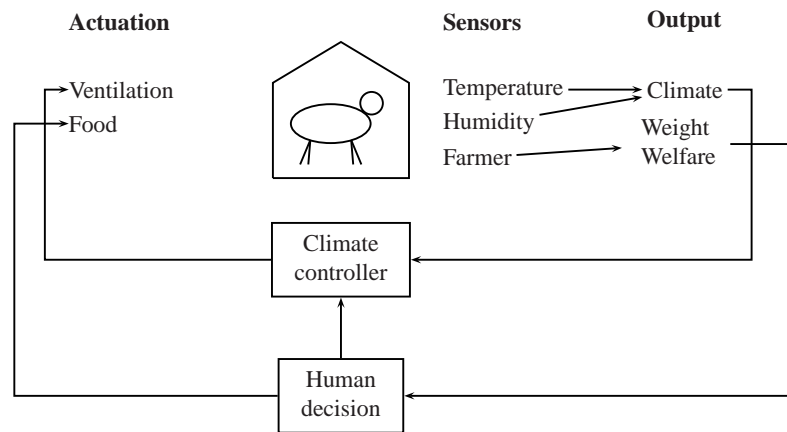
Figure 1.1: Schematic view of current practice in pig production.

Figure 1.1 illustrates a system where a climate control system is used to keep the climate within well defined boundaries. The climate is controlled on the basis of measurements of temperature and humidity and adjusts the ventilation rate accordingly. In this system the farmer acts as a sort of supervisor assessing the welfare of the animals. Possible actions from the farmer are adjustments of the climate controller and change in e.g. food type. If the farmer assesses that the animals are in a non healthy condition he can call a veterinarian or give medicine by himself.

The idea behind the vision for a future pig production is to automate the process illustrated in figure 1.1. The idea is illustrated in figure 1.2.
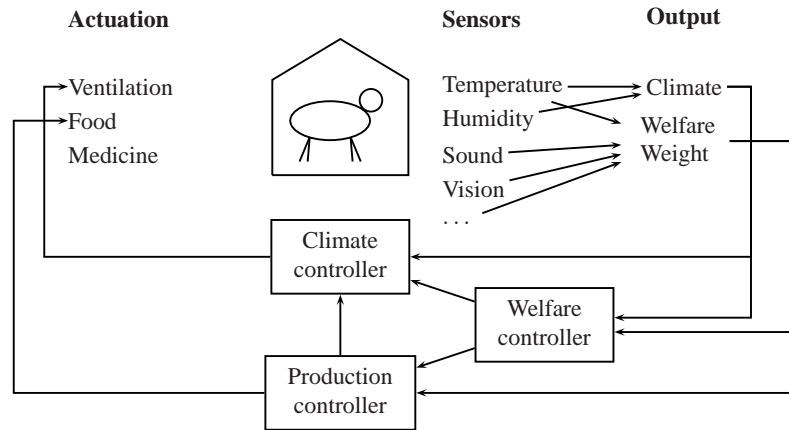
Figure 1.2: Schematic view of future pig production practice.

Figure 1.2 illustrates the vision for future production systems where the farmer is replaced by a production controller. This requires new sensors and sensor output e.g. weight or the more abstractly defined concept of welfare. In figure 1.2 it is illustrated how a welfare controller can influence both the climate and production controller. This thesis omits the discussion of animal welfare and welfare sensors, and points attention to the concurrent research projects that focuses particularly on animal welfare and welfare sensors. However, since this thesis considers a future implementation of climate controllers, the presented solutions are constructed in such a way that they easily can incorporate both production and welfare controllers.

1.3 Vision for Future Climate Controllers

The current practice for controlling climate in pig stables is based on measurements of mainly temperature and humidity. The reference point for the controller is set by the individual farmer, possibly with the aid of graphs showing the pigs' climate need as a function of their age and/or weight.

Climate control, without considering welfare, is thus a matter of keeping temperature, humidity and air quality sufficiently close to specified reference values under well known conditions and is as such an exercise in feedback control. However, the external environmental factors in pig production make it extremely difficult to set a correct reference point for a climate control system, since not only weight, but also breed, age,

daily feed intake and the surroundings e.g. time of day and floor type in the lying area affect the pigs' environmental requirements. Adding unknown disturbances and varying system dynamics as well as the possibility of component failure in sensors, actuators, communication systems and computers, the task is of far greater complexity than seen at a first glance.

1.3.1   Ventilation Systems

Many types of ventilation systems for animals houses exists, each possessing pros and cons depending on the requirements for the system. These requirements are based on the number and type of animals in the building, the physical dimension of the building and the climatic zone where the building is placed. Ventilation systems are placed in one of two categories: Natural or mechanical ventilation. In natural ventilation systems there are a number of openings letting air into and out from the stable. These openings are typically placed in the wall and the roof ridge (see figure 1.3(b)). It is the natural driving forces of air (pressure or temperature difference) that cause the ventilation. Natural ventilation is mostly found in livestock buildings for cattle because they in general have a better resistance to variation in climate than pigs. Mechanical ventilation uses at least one driving fan for air intake or exhaust, thus *forcing* air into/out of the stable.

(a) Diffuse   (b) Natural

(c) Wall   (d) Tunnel
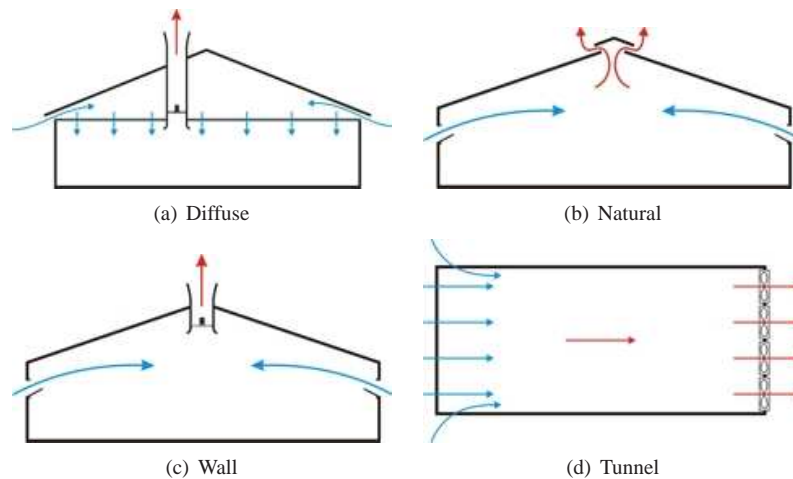
Figure 1.3: Illustration of various ventilation types.

Figure 1.3 illustrates four different ventilation systems: diffuse, natural, wall and tunnel. Diffuse and wall refer to the air intake method; in wall intake an exhaust fan exhales air from the stable, thus, causing a pressure difference between the internal and external environment while air is let into the building through wall inlets. The diffuse

type lets air into the stable through a diffuse ceiling material. Tunnel ventilation is in principal equal to wall and diffuse intake; the three methods use mechanical ventilation to exhale air from the stable. A key difference between tunnel and wall/diffuse intake exists, though, namely that in tunnel ventilation one of the gables consists more or less entirely of fans while the opposite gable is the air intake - see figure 1.3(d), thus, causing a draught to cool down the animals. Tunnel ventilation is primarily suitable for regions with hot climate where diffuse and wall intakes are inadequate for sufficient ventilation. In Danish pig production mechanical ventilation systems with wall inlets are the most common found type of ventilation systems, and the rest of this thesis, therefore, considers only this kind of ventilation system.

Control of Ventilation Components

The actuators found in the stable are wall inlets, ventilation fans (outlets) and possibly heating. An air inlet is an opening in the wall with a guiding plate to adjust the amount of incoming air and a grid to prevent leaves etc. to enter. A typical wall inlet (see figure 1.4(a)) has a protecting plate mounted on the external side to minimize the effect of direct wind gusts.

(a) Wall inlet   (b) Chimney
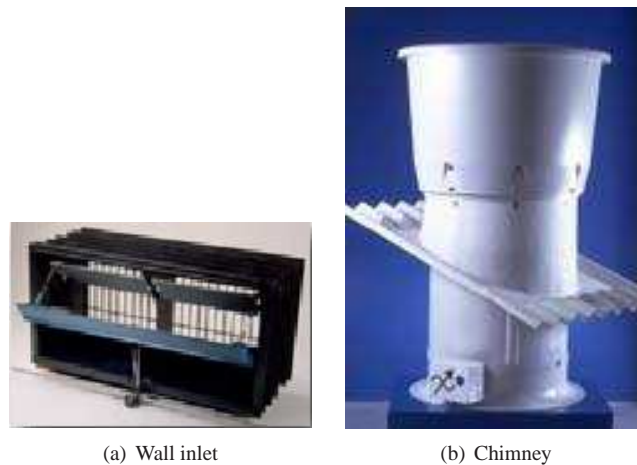
Figure 1.4: Image of wall type inlet and exhaust air chimney.

Ventilation fans are mounted in chimneys (see figure 1.4(b)) and consists of the ventilation fan itself and a damper plate giving an extra degree of freedom for control purposes. For simplicity this thesis does not deal with the use of damping plates, thus, the rest of this thesis assumes that when a fan is turned on, i.e. the applied voltage > 0, the damping plate is fully opened. When the fan is turned off the damping plate is closed.

5

In a one zone stable (see figure 1.5) all actuators are activated in a parallel like fashion. For inlets this is implemented with one motor that via a pulley system operates all inlets in both sides of the building simultaneously.
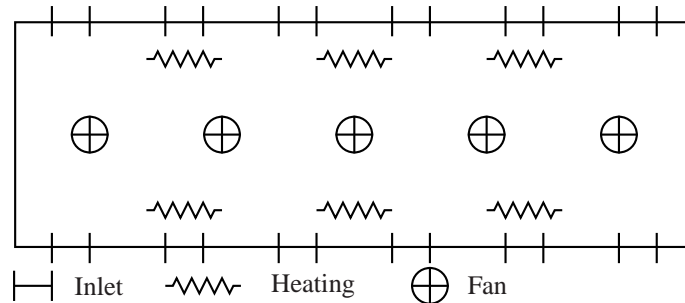
Figure 1.5: Illustration of typical climate components in pig stables.

If controlled heating is present the pipes are connected in series to a main boiler. Though figure 1.5 illustrates a number of radiators, controlled heating is actually a *one pipe system*.

1.3.2   Zone Based Climate Control

In the ventilation system depicted in figure 1.5 the climate control is based on measurements of temperature and possibly humidity from sensors located in the middle of the building approximately one meter above ground level. In these kinds of systems *one* temperature sensor is used to give feedback to the climate control, and a general assumption is, therefore, that no significant climate gradients within the livestock building exists. In the rest of this thesis this system concept will be referred to as *one zone* systems.

In a pig stable pigs can not move around freely in the entire stable, but are confined to pig pens with a density of approximately 0.75 [m^2/pig]. Pigs maintain a social hierarchy where lower ranking pigs often are bullied and/or physically molested by higher ranking pigs. The use of pig pens is, thus, not only of practical use for the farmer since he can overview the pigs more efficiently, but also serves as a control mechanism for limiting the hierarchy sizes. In case not all pig pens are filled Skov suggests to the farmers that pigs are kept near the center of the stable, i.e. near to the climatic sensors. In every day practice, though, pigs are placed in pig pens by the convenience of the pig farmer, and in general a uniform distribution of pigs around the building center can not be assumed.

As mentioned before climate control is indeed a complex task. Adding the possibility of nonuniform distribution of disturbances (pigs) and even the possibility of having leakages in the building envelope, it does not appear valid that the climatic parameters

are without significant gradients. This hypothesis was first stated theoretically in (Barber and Ogilvie, 1982) and later shown experimentally in (Barber and Ogilvie, 1984). For recent work in incomplete mixing of airspace in livestock buildings see (Moor and Berckmans, 1996; Janssens et al., 2004) and references therein. Based on the general acceptance that the airspace in livestock buildings is incompletely mixed, this thesis, therefore, proposes a multi zone based climate control system as illustrated in figure 1.6.
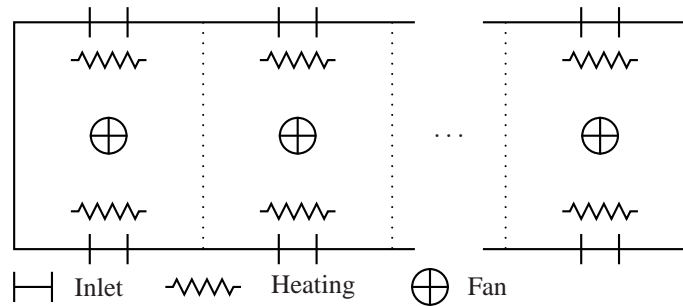
Figure 1.6: Illustration of climate components in zone partitioning.

In multi zone climate control systems the stable is partitioned into a number of separate zones, where each zone has individual actuation control as illustrated in figure 1.6. In the multi zone climate control concept each zone controller can have its own reference point.

Advantages

The advantages of upgraded climate control systems can be divided into two categories: Improved performance and extra functionality. Within these two categories a number of use cases to advocate the extra cost of this new system is presented.

Improved Performance

- Under normal production conditions pigs are placed in pig pens by the convenience of the pig farmer. This nonuniform distribution of the pigs leads to varying thermal disturbances in the stable, which can be compensated for by a zone based climate control system.

- Differences in a stable e.g. leakages in one end of the stable, unconventional buildings, broken windows or open doors can cause the zones to have different dynamics. A more accurate climate control can be expected in this case by using the zone based idea.

- The presented system is equipped with a surplus of sensors. This indicates the possibility of robustness in case of sensor failure.

- Since the stable is divided into a number of zones, the stable seen as *one* zone is over-actuated. In case of actuator failure in a zone, the controller can compensate by increasing ventilation in other zones.

- As an example, surplus heat generated by pigs in a relatively crowded zone may, by ventilation, be lead to zones of low population if allowed by air quality criteria.

Extra Functionality

- Different climatic needs dictated by age or race specific demands, can cause the zones to have different reference points. Introducing zones in the stable allows differentiated climate control.

- The stable as presented is in many situations over-actuated. The extra degrees of freedom introduced in this manner may be used to achieve energy optimal climate control. A search among a number of feasible equilibria may be performed and the optimal can be selected.

- The presented vision is also equipped with a surplus of sensors. Such a surplus facilitates error detection.

- Introducing zones allows for possible elimination of draught between zones.

1.4   Climatic Zones

The previous section presented the vision of a future climate control concept based on partitioning a stable into zones, and the available ventilation components. Due to climatic differences throughout the world, ventilation equipment from one part of the world can not be expected to deliver satisfactory performance in other parts of the world e.g. the wall inlet type of ventilation for a Danish pig stable is insufficient for a poultry stable in Saudi Arabia where tunnel ventilation is required.

Good indoor climate depends on many factors; temperature, draught and air quality e.g. level of CO_2 concentration, ammonia or NH_X gases. But in order to quantify quality, sensors are needed that in a practical setting would be too expensive for use in pig stables. The type of sensors available for climate control in pig stables are most often temperature and humidity. Throughout the rest of this thesis only these two measures are considered. Regarding the ambient temperature and humidity four scenarios exists as illustrated in table 1.1.

The geographical location typically confines a stable to one or two of the categories in table 1.1. As an example consider the following requirement for the stable climate: $T_{ref} = 20$ [˚C] and $H_{ref} = 70$ [%]. In the following the scenarios from table 1.1 are

Humid & Cold	Humid & Warm
Dry & Cold	Dry & Warm

Table 1.1: Partitioning of climatic zones with respect to average temperature and relative humidity.

described with respect to the required actuators. To simplify matters it is assumed that the heat produced by the pigs is always greater than the heat loss through the building. The following sections refer to the ambient environment, while their subsections refer to the indoor climate.

1.4.1 Cold & Humid

In general it is not a problem for pigs if the relative humidity is low since the presence of pigs automatically increases the moisture content of the air, but for the sake of completeness four cases with respect to the climate inside the stable are identified.

Too Hot and Too Humid

Since the ambient air is cold, ventilation will decrease the temperature. Because cold air with a high relative humidity contains less water than the warm indoor air, ventilation will decrease both temperature and humidity i.e. no extra actuators are required.

Too Hot and Too Dry

As stated, the dryness is not really a problem and ventilating will decrease the temperature.

Too Cold and Too Humid

This is a potential problem since stopping the ventilation will result in a temperature increase but also in a climate too humid for the pigs. In this case it could be necessary to have extra heating present in the stable, since ventilation will decrease both temperature and humidity. Alternatively, dehumidifiers could be installed.

Too Cold and Too Dry

In this case ventilation could be stopped, which will result in increased temperature and humidity level. At some point either the temperature or the humidity will be too high and a change in control action will take place.

1.4.2   Cold & Dry

This is basically the same case as in the previous section. Air at 20 °C and a relative humidity of 70 % contains approximately 0.01 [(kg H_2O)/(kg dry air)] (Schmidt, 1989), but air at 10 °C and a relative humidity of 100 % contains approximately 0.007 [(kg H_2O)/(kg dry air)]. So, according to the control objective ventilation will result in a decrease in both temperature and humidity.

1.4.3   Warm & Humid

In regions with hot and humid weather *standard* ventilation will never be able to maintain the required climate. Only when the climate inside the stable is too cold and too dry will ventilation be beneficial. In such regions alternatives, such as tunnel ventilation, are required. As explained previously, tunnel ventilation (cf. figure 1.3(d)) creates a huge draught in the stable, thus, using the wind chill factor to cool down the animals. If further cooling is required it is possible to let the air flowing into the stable pass through a water-drenched porous material to cool down the intake air before utilizing the wind chill factor.

1.4.4   Warm & Dry

In warm and dry ambient environment the same ventilation principles as for warm and humid ambient applies. Because warm air can contain more water than cold air, even warm and dry (relative humidity) air can actually increase the humidity level.

1.4.5   Climate Control in Denmark

Denmark is placed in a temperate climate with cool summers and mild winters. With an average yearly temperature around 8 °C it is, thus, fair to assume that ventilation will decrease the temperature. Assuming that $T^{\mathrm{amb}} < T^{\mathrm{ref}}$ just leaves the humidity to be investigated. As explained in the Cold & Dry and Cold & Humid section previously, cold air can contain less water than warm air, in fact in winter times in Denmark when the average temperature is 0.5 °C the air will always contain less water than required for the stable climate. Throughout the rest of this thesis it is, therefore, assumed that ventilation will decrease both temperature and humidity.

1.5   Climate Controller Development Environment

Embedded software is all around us from our mobile phone, TV and dish washer to more safety critical applications such as car electronics, aviation surveillance and nuclear power plants. In fact, most of the software (and electronics) around us that is not a normal personal computer may be considered as embedded software. Though climate

control in pig stables at first sight does not have the same public or scientific appeal as e.g. space exploration it is nevertheless important. Denmark is the world's largest exporter of pork, and with a gross domestic product at factor cost of 18131 million Danish kroner climate control in pig stables is important both in respect of the well being of the animals and revenue for the farmer. Thus, without actually quantifying it, software for climate control in pig stables has to be *safe*.

Software is often developed after a so-called V-model (Pressman, 1986; Biering-Sørensen et al., 1994) as illustrated in figure 1.7 (without the dotted lines). Following the V-model, a project starts at the highest abstraction level by making a system specification, and as time progresses (hopefully) the abstraction level decreases, ending up with the actual source code after which testing begins.
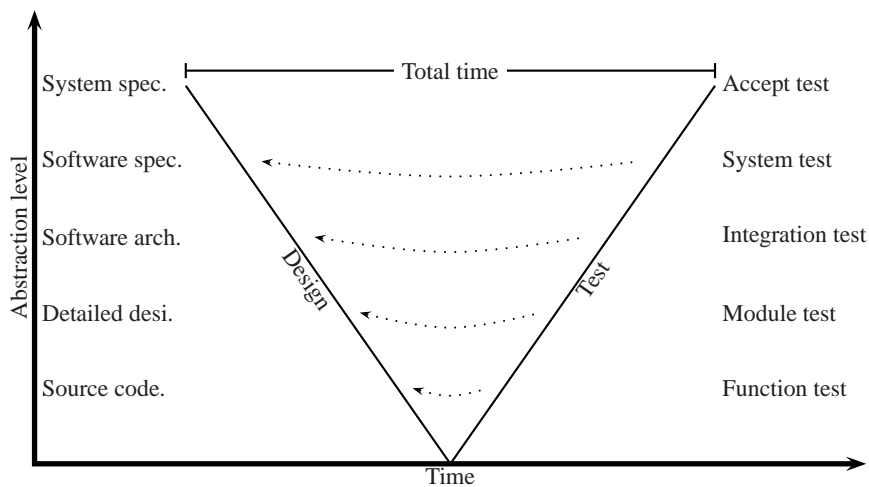
Figure 1.7: Abstract development model.

While figure 1.7 illustrates an ideal (the V-model) it also illustrates how things often work in practice. The dotted lines in figure 1.7 illustrate how development possibly consists of a number of iterations, since tests can reveal design flaws. At each iteration along the dotted line extra time is added to the total development time because development needs to go through the bottom of the V.

When the source code is ready it needs to be compiled in order to be executed on a target platform. Most people may not think deeply about it, but when source code is compiled into an executable, the programmer "trust" that the compiler works as expected i.e. the executable behaves according to the written code. If the programmer didn't trust the compiler, he would himself have to write the machine code for the target platform, which would be an immense task. The conclusion is that in order to develop software, a trusted tool chain is needed. This has fostered the Y-model illustrated in figure 1.8.

The Y-model uses automatic code generation in order to speed up the development
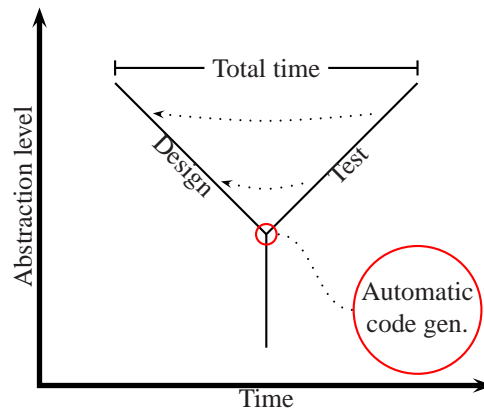
Figure 1.8: Abstract development model using automatic code generation.

cycle. Figure 1.8 illustrates how the design begins at some high abstraction level and as with the V-model the abstraction level decreases as development progresses. But contrary to the V-model the Y-model reaches a point where a certain tool automatically translates the design into source code, which is then compiled. Automatic code generation exists on many levels, in fact a C-compiler could be considered as performing automatic code generation translating C-code to machine code. But in order to really utilize the possible benefits of using automatic code generation it has to be performed at a "high" level in figure 1.8. A tool that allows automatic code generation from a high level is Real-Time Workshop (RTW) from Mathworks. With RTW it is possible to generate c-code from a Simulink diagram that can be executed on a target platform. Other tools exist but the Mathworks' tools have been chosen as a case study for this project.

1.6 Contributions

The following contributions are documented in this thesis:

1. Demonstrated how to integrate monitoring and remote control of livestock building climate with the development environment for such systems.

2. Demonstrated how to build remote monitoring systems for livestock buildings based on the integration of such tools in the development environment.

3. Demonstrated the applicability of using readily available hardware and software components (COTS - Commercial Off The Shelf) for climate control systems.

4. Showed how to perform parameter estimation for a dynamic climate model that is both hybrid and nonlinear.

5. Showed how to avoid algebraic loops in Simulink in order to build a graphical modular simulation system for zone based climate dynamics.

6. Showed how to design a verified stable climate controller for zone based climate dynamics using an already established method based on game theory, but with inclusion of coordinating logic variables.

7. Demonstrated a tool chain for automatic control synthesis for controllers modelled in UPPAAL TIGA, simulated in Simulink and with the possibility of generating code for real life testing.

1.7 Thesis Outline

The dissertation is organized as follows: Chapter 2 presents a survey of literature on climate modelling and control ending up with a presentation of the model used in this thesis. Chapter 3 through 10 are composed of the following papers[1]:

1. J. J. Jessen, H. Schiøler, J. F. D. Nielsen, M. R. Jensen, COTS Technologies for Integrating Development Environment, Remote Monitoring and Control of Livestock Stable Climate. IEEE International Conference on Systems, Man, and Cybernetics (SMC2006). Taipei, Taiwan. October 2006.

2. J. J. Jessen, J. F. D. Nielsen, H. Schiøler, M. R. Jensen, COTS Technologies for Internet Based Monitoring of Livestock Buildings. 10th IASTED International Conference on Internet and Multimedia Systems and Applications (IMSA2006). Honolulu, Hawaii. August 2006.

3. J. J. Jessen, H. Schiøler, ZoneLib: A Simulink Library for Modeling Zone Divided Climate Dynamics. ISCA 19th International Conference on Computer Applications in Industry and Engineering (CAINE2006). Las Vegas, Nevada, USA. November 2006.

4. J. J. Jessen, H. Schiøler, System Identification in a Multi Zone Livestock Building. International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA06). Sydney, Australia. December 2006. Accepted for publication. Withdrawn due to financial issues.

5. J. J. Jessen, H. Schiøler, Parameter Estimation for Zone Based Climate Dynamics. The Second IASTED International Conference on Computational Intelligence (CI2006). San Francisco, California, USA. November 2006.

---

[1]The papers have been reformatted from the original layout to comply with the layout in this thesis. The papers are reproduced under the conditions of the copyright agreement with the original publishers.

6. C. De Persis, J. J. Jessen, R. Izadi-Zamanabadi, H. Schiøler, Internal Flow Management in a Multi-Zone Climate Control Unit. Invited paper in the session *Networked Control Systems* 2006 CCA/CACSD/ISIC. Munich, Germany. October 2006.

7. C. De Persis, J. J. Jessen, R. Izadi-Zamanabadi, H. Schiøler, A Distributed Control Algorithm for Internal Flow Management in a Multi-Zone Climate Unit. Accepted for publication in International Journal of Control. To appear.

8. Jan J. Jessen, Jacob I. Rasmussen, Kim G. Larsen, Alexandre David, Optimal Controller Synthesis for Climate Controller Using UPPAAL TIGA. Submitted for review. 19th International Conference on Computer Aided Verification. 2007.

Experimental results are presented in chapter 11, while conclusions and recommendations for future work are given in chapter 12.

Climate Modelling and Control in Agro and Horticulture Buildings

This chapter presents a survey on models and suggestions for climate control in buildings for intensive animal production (typically pig or poultry). The modelling/control problem is similar to that of greenhouses, therefore the general modelling/control systems found in horticulture is also studied.

A vast amount of literature exists describing models and controllers for indoor climate. The cited papers in this chapter are chosen to represent the span that exists in perceptions and beliefs on how to model and control climate dynamics. The chapter is outlined as follows: Steady state models will be presented first, followed by dynamic models that only use *one* measurement point for climate e.g. temperature, humidity or CO_2 concentration. Next, models are presented that model more than one measurement point, and suggestions for controllers are presented. The chapter concludes with a section describing the basic model used in the rest of this thesis. Various authors have different opinions on notation. In this thesis the editorial freedom of presenting the models with a consistent notation has been taken.

2.1 Steady State Climate Models

Steady state climate models are based on the assumption that the evolution of the climatic states are constantly equal to zero e.g. for temperature $\frac{d}{dt}T^i = 0$. In (Schauberger et al., 2000) it is stated that steady state balance models can be used in a prognostic mode where they are used to help design ventilation systems and used in diagnostic mode where models are compared with measured values. The presentation given here will be limited to prognostic mode.

2.1.1 CIGR Based Steady State Models

The International Commission of Agricultural Engineering (CIGR, Commission Internationale du Génie Rural) technical section II: Farm Buildings, Equipment, Structures and Environment publishes a number of reports on animal heat, moisture, CO_2 and ammonia production, the latest being (Pedersen and Saalvik, 2002). In general the equations on the climate in animals houses found in (Pedersen and Saalvik, 2002) are used as basis for steady state models. The idea is to set up balance equations for e.g. heat as:

$$\phi_A + \phi_B + \phi_V = 0 \tag{2.1}$$

where ϕ_A is the sensible heat produced by the animals, ϕ_B is the heat loss through the building envelope and ϕ_V is the heat loss from ventilation.

The total heat production of the pigs is divided into sensible and latent heat dissipation. The former dissipates due to a temperature difference between deep body temperature and ambient, while the latter dissipates in form of moisture from respiration. The sensible heat production (temperature) of the pigs is set to:

$$\phi_A = \phi_{tot} f_s k_s \tag{2.2}$$

where ϕ_{tot} is the total heat production from the animals, f_s is the fraction of the total heat production that appears as sensible heat and k_s is a correction factor taking into account evaporation from wet surfaces. The sensible heat production from pigs constitutes approximately 80% of the total heat production (Pedersen and Saalvik, 2002).

The heat loss through the building envelope is calculated as;

$$\phi_B = U_B A (T^{amb} - T^i) \tag{2.3}$$

where U_B is the mean U-value for the building material and A is the corresponding area. The heat loss from ventilation is calculated as:

$$\phi_V = Q c_{air} \rho_{air} (T^{amb} - T^i) \tag{2.4}$$

where Q is the ventilation rate, c the specific heat capacity and ρ is the air density. Now, knowing the number of pigs, using the equations for heat production from pigs (Pedersen and Saalvik, 2002) and knowing the outside temperature, the required ventilation rate can be calculated. Here, the approach is presented using temperature. In (Schauberger et al., 2000; Pedersen et al., 2004, 1998) the same approach is used with humidity and CO_2 concentration. Of course such models are not applicable for implementing dynamics controllers, and though (Schauberger et al., 2000) designs a controller on basis of the balance equations in (Koerkamp et al., 1998) and (Seedorf et al., 1998) they are used as an estimate of emission of endotoxins, microorganisms and ammonia from livestock buildings.

2.1.2 Static Model

In (Berckmans and Goedseels, 1986) a steady state analysis is performed leading to the following model:

$$T^{\mathrm{i}} = T^{\mathrm{amb}} + \frac{\phi_{\mathrm{sens}} + \phi_{\mathrm{sup}}}{Q\rho_{\mathrm{air}}c_{\mathrm{air}} + \Sigma AU} \tag{2.5}$$

where T^{i} and T^{amb} are the indoor and outdoor temperature, ϕ_{sens} is the sensible heat production of the animals, ϕ_{sup} is the heat supply, Q is the ventilation rate, ρ_{air} is the air density, c_{air} is the specific heat of air and ΣAU is the sum of products of heat transfer coefficients between inside and outside and the corresponding area. Inspecting (2.1) through (2.4) it is clear that the model presented in (Pedersen and Saalvik, 2002) is identical to (2.5) except that (2.5) includes the possibility of controlled heating via the ϕ_{sup} term. A similar model to (2.5) is found in e.g. (Gutman et al., 1993; Linker et al., 2002).

2.2 Dynamic Climate Models

Dynamic climate models are characterized by modelling the evolution of some climatic parameter over time, typically temperature. Based on first engineering principles such models typically have the following appearance:

$$\frac{d}{dt}T^{\mathrm{i}}V = Q(T^{\mathrm{i}} - T^{\mathrm{amb}}) + \frac{\sum_j \phi^j}{\rho_{\mathrm{air}}c_{\mathrm{air}}} - \sum_k A_k U_k (T^{\mathrm{i}} - T^{\mathrm{amb}}) \tag{2.6}$$

where T^{i} is the temperature inside the building, V is the volume, Q is the ventilation rate, T^{amb} is the outside temperature, $\sum_j \phi^j$ is the sum of heat sources, A_k is the area of the different building materials and U_k is the corresponding heat transfer coefficients for the different building material types. By setting $\frac{d}{dt}T^{\mathrm{i}} = 0$ in (2.6) it is readily seen that the model in principle is similar to (2.5).

2.2.1 Stochastic Model of Heat Dynamics

In (Nielsen and Madsen, 1998) a linear lumped parameter model for temperature distribution in a greenhouse is introduced. The idea is illustrated in figure 2.1, where only the first node depends on the heating system, sun and outdoor temperature, while the rest of the nodes depend on each other in a cascade like fashion.

The model takes into account solar radiation Φ and the heating system ϕ. The first node is modelled as:

$$C_1 \frac{dT_1}{dt} = \frac{T_1 - T_2}{R_1} + \frac{T^{\mathrm{amb}} - T_1}{R_0} + \phi + A\Phi \tag{2.7}$$
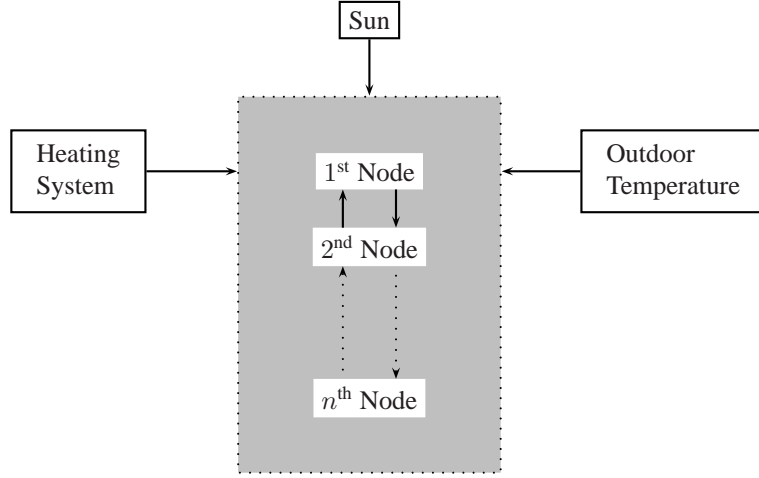
Figure 2.1: Illustration of lumped model for greenhouse temperature.

where C_1 is the heat capacity, T_1 is the temperature in the 1$^{\text{st}}$ node, T_2 the temperature in the 2$^{\text{nd}}$ node. R_1 is the resistance to heat transfer to the second node, T^{amb} is the ambient temperature and A_{s} is the area to the outside. R_0 is the resistance to heat transfer between inside and outside. The energy balance for the jth node becomes:

$$C_j \frac{dT_j}{dt} = \frac{T_{j-1} - T_j}{R_{j-1}} + \frac{T_{j+1} - T_j}{R_j} \tag{2.8}$$

where R_j is the resistance to heat transfer between node j and $j + 1$. An additive noise process is introduced and the entire system is modelled as:

$$dT = \mathbf{A}T dt + \mathbf{B}U dt + dw(t) \tag{2.9}$$

where $dw(t)$ is a wiener process. T is the temperature in the different nodes, $\mathbf{A}$ is a matrix with the different capacitances C_j and resistances R_j. $\mathbf{B}$ is a special matrix having only nonzero values in the first row taking into account that $U = [T^{\text{amb}} \ \Phi \ \phi]^{\text{T}}$ only applies to the first node. For $n = 3$, $\mathbf{A}$ and $\mathbf{B}$ are given as:

$$\mathbf{A} = \begin{bmatrix} -\frac{1}{C_1 R_0} - \frac{1}{C_1 R_1} & \frac{1}{C_1 R_1} & 0 \\ \frac{1}{C_2 R_1} & -\frac{1}{C_2 R_1} - \frac{1}{C_2 R_2} & \frac{1}{C_2 R_2} \\ 0 & \frac{1}{C_3 R_2} & -\frac{1}{C_3 R_2} - \frac{1}{C_3 R_3} \end{bmatrix} \tag{2.10}$$

$$\mathbf{B} = \begin{bmatrix} \frac{1}{C_1 R_0} & \frac{1}{C_1} & \frac{A}{C_1} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.11}$$

The idea of using a system description given as a number of coupled first-order differential equations is to perform experiments and later use a statistical measure of the model order that gives the best fit to the measured data.

2.2.2   Parameter Relation Model

The general model for temperature in (2.6) was based on physical modelling, and can be used as basis for parameter relation models. In (Linker et al., 1999) this approach is taken in order to model temperature and CO_2 concentration in a greenhouse. Identifying related parameters and actuator signals gives the following model:

$$
\begin{bmatrix} \dot{T^i} \\ \dot{G^i} \end{bmatrix} = \begin{bmatrix} \bullet & 0 \\ 0 & \bullet \end{bmatrix} \begin{bmatrix} T^i \\ G^i \end{bmatrix} + \begin{bmatrix} \bullet & 0 \\ \bullet & \bullet \end{bmatrix} \begin{bmatrix} Q \\ R \end{bmatrix} + \begin{bmatrix} \bullet & \bullet & \bullet \\ 0 & \bullet & 0 \end{bmatrix} \begin{bmatrix} T^{amb} \\ \Phi \\ h^i - h^{amb} \end{bmatrix} \tag{2.12}
$$

where $\bullet$ denotes a linear or nonlinear relation, Q is the ventilation rate, R is CO_2 actuation, Φ is solar radiation and h is absolute humidity.

A parameter relation model is also found in (Daskalov et al., 2004), where it is argued that the approach of fixed set points for environmental variables not always results in the best performance because of over simplifications of complex process variable interactions. It is furthermore stated that accurate temperature control is insufficient in order to maintain good indoor climate - humidity control is needed as well.

The following relations are presented for temperature T and humidity H:

$$
\frac{dT^i(t)}{dt} = f(T^i, T^{amb}, H^{amb}, v_{wind}, Q, \phi_{tot}, W_{tot}) \tag{2.13}
$$

$$
\frac{dH^i(t)}{dt} = f(H^i, T^{amb}, H^{amb}, v_{wind}, Q, \phi_{tot}, W_{tot}) \tag{2.14}
$$

where v_{wind} is the wind velocity, Q is the ventilation rate, ϕ_{tot} is the total heat production in the stable and W_{tot} is the total moisture production in the stable. The total heat production is given as:

$$
\phi_{tot} = \phi_{heat} + \phi_{an} - \phi_e = \phi_{heat} + \phi_{an} - \lambda W_e \tag{2.15}
$$

where ϕ_{heat} is the controlled heat, ϕ_{an} is the heat from animals and W_e is the moisture transfer from evaporation from wet surfaces. The total moisture production is given as:

$$
W_{tot} = W_{fog} + W_{an} + W_e \tag{2.16}
$$

It is stated that the terms in (2.13) and (2.14) are highly coupled and nonlinear, and that physical modelling can not explain the exact dependence of the rate of temperature and moisture content change on the other variables. Using a system identification technique the following model and corresponding constants are found:

$$
\frac{dT^i(t)}{dt} = \tau_T^{-1}\left(-T^i + a_{g1}T^{amb} + a_{g2}H^{amb} + a_{g3}v_{wind} + a_{g4}Q + a_{g5}\phi_{tot} + a_{g6}W_{tot}\right) \tag{2.17}
$$

19

$$\frac{dH^{\mathrm{i}}(t)}{dt} = \tau_H^{-1}(-H^{\mathrm{i}} + b_{g1}T^{\mathrm{amb}} + b_{g2}H^{\mathrm{amb}} + b_{g3}v_{\mathrm{wind}} + b_{g4}Q + b_{g5}\phi_{\mathrm{tot}} + b_{g6}W_{\mathrm{tot}}) \quad (2.18)$$

where a_{gi} and b_{gi}, $i = 1, \ldots, 6$, are model coefficients (a_{gi} and b_{gi} left out here) and $\tau_T = 34$ [min] and $\tau_H = 30$ [min].

In (Berckmans et al., 1993; Moor and Berckmans, 1996) it is argued that classical computational fluid dynamics often are developed for simulation in steady state and are therefore not applicable for control. Instead, a grey box model is suggested, defined as a model based on physical laws combined with a mathematical identification technique. For a control volume inside a test chamber, physical modelling results in the following relations for temperature and humidity:

$$\frac{dT^{\mathrm{i}}}{dt} = -\beta Q T^{\mathrm{i}} + \beta Q T^{\mathrm{amb}} + \delta\phi \quad (2.19)$$

$$\frac{dH^{\mathrm{i}}}{dt} = -\beta Q H^{\mathrm{i}} + \beta Q H^{\mathrm{amb}} + \delta h_{\mathrm{c}} \quad (2.20)$$

where Q is the air flow rate, ϕ is the total heat flow into the control volume consisting of animal heat production and e.g. electrical heating devices. h_{c} is the internal humidity production. The Greek letters are physical constants.

To model the temperature evolution in discrete time a zero order hold equation is used leading to:

$$T(k+1) = T(k)e^{A\Delta t} + \left(\frac{B + C/A}{A} + \frac{C}{A}k\Delta t\right)(e^{A\Delta t} - 1) - \frac{C}{A}k\Delta t \quad (2.21)$$

where A, B and C are matrices containing the parameters Q, ϕ and h_{c}.

2.3 Multi Zone Climate Models

In (Barber and Ogilvie, 1982) it is argued that incomplete mixing in ventilated air spaces should be considered when designing ventilation systems for e.g. pig stables. Three different scenarios are listed where the normal assumption of complete mixing is wrong: Ventilation short circuit from inlets to outlets, parallel coupling for zones where "direct ventilation" only affects the first zone which affects zone two and serial coupling between zones. The assumptions found in (Barber and Ogilvie, 1982) are documented and tested in (Barber and Ogilvie, 1984).

The fact that the airspace inside livestock buildings is incompletely mixed has fostered the idea of modelling climatic parameters more than one place. Such models can be deduced from (2.6) by extending it with an index i to denote the ith temperature, and how different T_i^{i} influence each other.

2.3.1 Non-Interacting Zone Model

In (Janssens et al., 2004) a model for 36 measuring points is presented. The scenario is a test chamber: length 4 m, width 2.5 m and height 3 m with a three dimensional grid consisting of 36 temperature and humidity sensors. Around the test chamber a buffer chamber is build. The following model is presented for temperature and moisture dynamics in the ith zone:

$$\frac{dT_i^{\text{i}}(t)}{dt} = \frac{Q}{V_1}T^{\text{amb}}(t-\tau) + \frac{UA}{V_1\rho_{\text{air}}c_{\text{air}}}T_{\text{buff}}(t) - \left(\frac{Q}{V_1} + \frac{UA}{V_1\rho c}\right)T^{\text{i}}(t) + \frac{\phi}{V_1\rho_{\text{air}}c_{\text{air}}} \quad (2.22)$$

$$\frac{dh_i^{\text{i}}(t)}{dt} = \frac{Q}{V_1}h^{\text{amb}}(t-\tau) - \frac{Q}{V_1}h_i^{\text{i}}(t) + \frac{M_{\text{c}}}{V_1\rho} \quad (2.23)$$

where V_1 is the well mixed zone volume, Q is the effective air flow rate entering the zone, T^{amb} is the temperature of the supplied air, U is the heat transfer coefficient between the well mixed zone and buffer zone, A is the surface area for which U applies, ρ_{air} is the air density, c_{air} is the specific heat capacity of air and ϕ is the fraction of internal heat production entering the zone, h denotes the absolute humidity and M_{c} is the fraction of the internal moisture production entering the zone, τ is the time delay for the supplied air before it enters the zone.

Denoting s as the derivative operator, and writing (2.22) and (2.23) in concise form yields the following:

$$\begin{bmatrix} T_1(t) \\ T_2(t) \\ \vdots \\ T_{36} \end{bmatrix} = \begin{bmatrix} \frac{\beta_1}{s+\alpha_1} & \frac{K_1}{s+\alpha_1} \\ \frac{\beta_2}{s+\alpha_2} & \frac{K_2}{s+\alpha_2} \\ \vdots & \vdots \\ \frac{\beta_{36}}{s+\alpha_{36}} & \frac{K_{36}}{s+\alpha_{36}} \end{bmatrix} \begin{bmatrix} T^{\text{amb}}(t-\tau) \\ T_{\text{buff}}(t) \end{bmatrix} \quad (2.24)$$

$$\begin{bmatrix} h_1(t) \\ h_2(t) \\ \vdots \\ h_{36}(t) \end{bmatrix} = \begin{bmatrix} \frac{\beta_1}{s+\beta_1} \\ \frac{\beta_2}{s+\beta_2} \\ \vdots \\ \frac{\beta_{36}}{s+\beta_{36}} \end{bmatrix} \begin{bmatrix} h^{\text{amb}}(t-\tau) \end{bmatrix} \quad (2.25)$$

where each subscript is a well mixed zone uniformly distributed in three dimensions, and

$$\beta = \frac{Q}{V_1} \ , \ K = \frac{UA}{V_1\rho_{\text{air}}c_{\text{air}}} \ , \ \alpha = \frac{Q}{V_1} + \frac{UA}{V_1\rho_{\text{air}}c_{\text{air}}}$$

In (Janssens et al., 2004) it is argued that the entire test chamber is imperfectly mixed, and it is assumed that the 36 zones are well mixed i.e. it basically sums to a compartment model. Zone interaction e.g. radiation or air flow is disregarded, since only air inlet temperature is used as input. Because all zones are affected by the inlet temperature there should have been an index i on each of the delays. In (Zerihun et al.,

2005) τ has been replaced with τ_i for a model similar to (2.22). In (Young et al., 2000; Brecht et al., 2005) a data-based mechanistic approach to modelling imperfectly mixed airspaces is presented. The model is basically identical to the one presented in (Janssens et al., 2004) and is left out here.

2.3.2   Interacting Zone Model

Contrary to (Janssens et al., 2004) a model with zone interaction is presented in (Caponetto et al., 2000). A multi layer model for a greenhouse is presented that divides the greenhouse in eight layers, whose thickness is proportional to the length from the southbound wall. The following model is presented for the heat balance in a layer:

$$E_{a1} = E_{a2} + E_{a3} + E_{a4} - E_{a5} + E_{a6} + E_{a7} + E_{a8} \tag{2.26}$$

where E_{a1} is the energy stored in the internal air volume as latency sensitive heat, E_{a2} is input from the heating system, E_{a3} is the convective heat exchange with the lateral wall, E_{a4} is the heat exchange with frontal walls (only layer 1 and 8), E_{a5} is the fraction of heat exchanged with the external air as latent and sensitive heat, E_{a6} is the conductive heat exchange with the adjacent layer, E_{a7} is the convective heat exchange with the soil and E_{a8} is the contribution from solar radiation.

The different terms in (2.26) are modelled as:

$$E_{a3} = U_{c,i} A_{c,i} (T_{ci} - T^{\mathrm{i}}) \tag{2.27}$$

where T_{ci} is the temperature of the wall, $U_{c,i}$ is the convective heat transfer coefficient and $A_{c,i}$ is the surface area of the wall.

$$E_{a4} = 2U A_{ci} (T^{\mathrm{amb}} - T^{\mathrm{i}}) \tag{2.28}$$

where K is the total heat transfer coefficient, A_{ci} is the surface area of the external wall and T^{amb} is the ambient temperature.

$$E_{a5} = \rho_a c_a Q_i (T^{\mathrm{i}} - T^{\mathrm{amb}}) + \rho_a \lambda \phi_i (U_i - U_e) \tag{2.29}$$

where ρ, c and λ is the air density, specific heat and latent heat energy respectively. U_i, U_e are the indoor and external relative humidity and Q is the air flow.

$$E_{a6} = 2 \frac{\lambda_a}{\Delta s_i - \Delta s_{i-1}} S_{c,J+1} (T_{i+1} - T_i) + 2 \frac{\lambda_a}{\Delta s_i - \Delta s_{i-1}} A_{c,J} (T_{i+1} - T_i) \tag{2.30}$$

where Δs is the thickness of the layer, λ_a is the thermal conductivity of the air and $A_{c,J}$ is the surface contact between two layers.

$$E_{a7} = U_{si} A_{si} (T_s - T^{\mathrm{i}}) \tag{2.31}$$

22

where U_{si} s the convective heat transfer between soil and air, A_{si} is the area and T_s is the soil temperature.

$$E_{a8} = \delta_v a_a (I_{NS} A_{c1} + I_{EW} A_{c6} + I_V A_b) \frac{V_i}{V} \qquad (2.32)$$

where δ_v is the trasmittivity coefficient, a_a is the area absorptivity coefficient, A_{c1} is the south layer, A_{c6} is the east layer, A_b is the base surface of the greenhouse, I_{NS} and I_{EW} is the fraction of the solar radiation in the north-south and east-west direction respectively and I_v is the radiation perpendicular to the soil.

2.4 Climate Controllers

This section gives a general overview of controllers for the agricultural industry including climate controllers and controllers for the ventilation rate.

2.4.1 P-Control of Steady State Model

In (Berckmans and Goedseels, 1986) a controller for (2.5) is suggested. The controller's principle of operation is illustrated in figure 2.2.
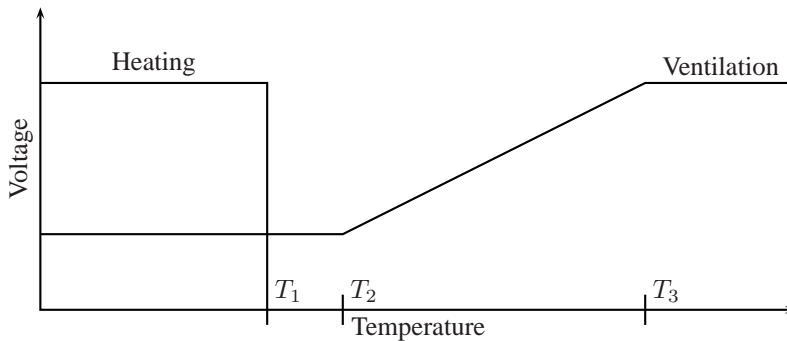
Figure 2.2: Illustration of proportional controller for steady state climate models.

When the indoor temperature T^i is below T_1, a predefined minimum ventilation is applied while the heaters are turned on. When $T_1 \le T^i \le T_2$ steady state exists, so the heating is cancelled. When $T_2 \le T^i \le T_3$ a proportional band to the voltage on the fan is defined, and when $T^i > T_3$ maximum ventilation is reached. A similar controller as described here is found in (Schauberger et al., 2000) except for the presence of controlled heating.

2.4.2   Soft Computing Climate Control

In (Caponetto et al., 2000) a fuzzy logic controller for greenhouse temperature is presented, which implements rules of the following Mamdani (Mamdami and Assilina, 1975) form:

$$\text{if } x_1 \text{ is } G_1^i \ \ldots \text{ and } x_m \text{ is } G_m^i \text{ then } y \text{ is } h^i \ , \ i = 1, 2...n \tag{2.33}$$

where x_k is the input variable, G^i is the input fuzzy set and h^i is the consequent singletons. A total of seven rules are defined with corresponding fuzzy sets, which are obtained by a trial and error strategy. Fine tuning of the membership functions shape and position is done with genetic algorithms.

2.4.3   Ventilation Rate Control

In (Taylor et al., 2004) a scheduled gain proportional-integral-plus (PIP) controller for a ventilation fan is presented. The controller is designed for a ventilation test chamber consisting of a control fan, ventilation rate sensor, throttling valve and disturbance fan.

PIP Control

For a description of PIP design see (Chotai et al., 1994) and references therein. Following (Taylor et al., 2004), the basic idea is to have a Non-Minimum State Space (NMSS) description of the system:

$$y(k) = \frac{b_1 z^{-1} + \ldots + b_m z^{-m}}{1 + a_1 z^{-1} + \ldots + a_n z^{-n}} u(k) \tag{2.34}$$

In the NMSS form the state vector consists of present and past sampled values of input and output variables defined as:

$$\begin{aligned} x(k) = [y(k) \ \ y(k-1) \ \ \ldots \ \ y(k-n+1) \ \ u(k-1) \\ u(k-2) \ \ \ldots \ \ u(k-m+1) \ \ z(k)]^{\mathrm{T}} \end{aligned} \tag{2.35}$$

where $z(k)$ is the integral of error $z(k) = z(k-1) + y_{\text{ref}}(k) - y(k)$. The NMSS equations are given as:

$$\begin{aligned} x(k) &= \boldsymbol{F} x(k-1) + \boldsymbol{g} u(k-1) + \boldsymbol{d} y_d(k) \\ y(k) &= \boldsymbol{h} x(k) \end{aligned} \tag{2.36}$$

In (Taylor et al., 2004) a number of experiments are performed in order to find the steady-state characteristics for the fan. An illustration of the measured characteristic is shown in figure 2.3.
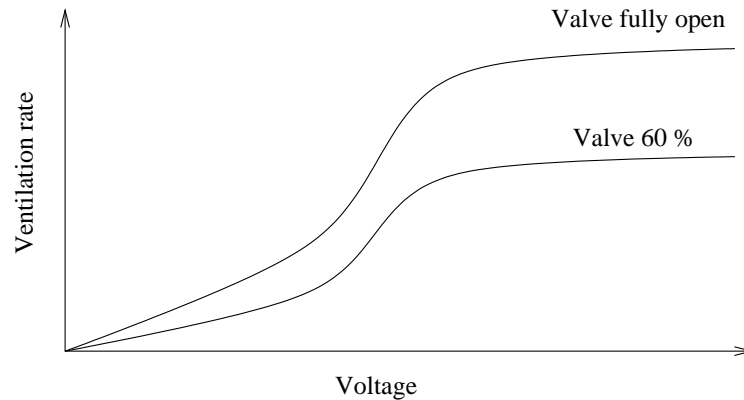
Figure 2.3: Illustration of steady-state fan characteristics.

By experiment, three models are presented for the ventilation rate as a function of the applied voltage for three operating conditions of the throttling valve. A PIP-LQ controller is designed for the three found models and the overall control system switches between the controllers based on the reference air flow rate. In (Linker et al., 1999) ventilation rate modelling/control is accomplished by training a neural network to output actuator signals based on a desired air flow rate. The found air flow rate in (Linker et al., 1999) is for both inlets and outlets, while (Taylor et al., 2004) focuses on the outlet only. In (Morcos et al., 1994) an electronic circuit for driving a ventilation fan is presented, but no model is presented.

2.4.4 Model Predictive Control

Model predictive control (MPC) is basically concerned with the computation of a finite sequence of control moves so that the predicted behavior of the system is close to a reference trajectory under some constraint, or generally optimizing some objective function. In (Brecht et al., 2005) a MPC is presented for controlling the temperature modelled as (2.22). Recalling section 2.3.1 the scenario is 36 temperature sensors uniformly distributed in a 3-dimensional space inside a test chamber. In (Brecht et al., 2005) it is stated that for the test chamber there are not enough independent control inputs to control the temperature in each of the 36 *well mixed zones*. Instead a MPC is evaluated of the average temperature of four temperature sensors. For MPC temperature control in a greenhouse see e.g. (Ghoumari et al., 2005). In (Piñón et al., 2005) MPC is mixed with a feedback linearization technique also for temperature control in a greenhouse. In (Coelho et al., 2005) the underlying programming problem of solving MPC for greenhouse temperature is done with the so-called particle swarm optimization algorithm.

2.5 Modelling Zone Climate by Flow Interaction

This section introduces the dynamic model for temperature that has been used in the enclosed articles constituting the rest of this thesis. The models are presented in (Jessen and Schiøler, 2006c,a,b). First, the models for ventilation components are introduced, which is followed by the model for temperature. The models presented here are not exact in the sense that they obey exact current physical knowledge, but focuses instead on simple relations suitable for control purposes. For details on modelling of ventilation components see (Brohus et al., 2002).

2.5.1 Air Inlets

As illustrated in figure 1.4(a) an air inlet is basically an opening in the wall with a guiding plate to adjust the direction and amount of incoming air. In the enclosed articles two different models are presented. In (Jessen and Schiøler, 2006c) the following model is presented:

$$Q_i^{\text{in}} = k_i \cdot \alpha_i \cdot (P^{\text{amb}} - P_i) \tag{2.37}$$

Where P_i is the pressure in zone i, P^{amb} is the ambient pressure, k_i is a constant and α_i is the control signal to the inlet determining the opening degree of the guiding plate. The direction of airflow through an inlet from the outside to the inside is defined to be positive. In (Jessen and Schiøler, 2006a,b) the following approximate affine model for airflow from the outside through the wall inlet Q^{in} in the ith zone is proposed:

$$Q_i^{\text{in}} = k_i \cdot (\alpha_i + l_i) \cdot (P^{\text{amb}} - P_i) \tag{2.38}$$

where l_i is a constant that can be interpreted as the possibility of modelling leakages in the building envelope.

2.5.2 Air Outlets

The following model for air outlets is proposed, which has a linear dominant term acting on the control signal (voltage applied to the fan) and a term subtracting flow due to pressure difference. The exhaust air flow Q^{fan} from the ith zone is thus found as:

$$Q_i^{\text{fan}} = u_i^f \, c_i - d_i \, (P^{\text{amb}} - P_i) \tag{2.39}$$

where u_i^f is the control signal for the fan, c_i and d_i are constants. The flow is defined positive from the stable to the outside.

2.5.3 Inter Zone Net Air Flow

The net air flow $Q_{i,i+1}$ between two zones (air flows are defined positive from a lower index to a higher index) is found by

$$Q_{i,i+1} = m_i \, (P_i - P_{i+1}) \tag{2.40}$$

where m_i is a constant.

2.5.4 Stationary Flows

A stationary flow balance for each zone i is found:

$$Q_{i-1,i} + Q_i^{\text{in}} = Q_{i,i+1} + Q_i^{\text{fan}} \tag{2.41}$$

where by definition $Q_{0,1} = Q_{N,N+1} = 0$.

2.5.5 Numerical Values for Inlet

Experiments conducted at Research Centre Bygholm (Bygholm, 2001) has led to the results for a single inlet (type DA 1200) shown in table 2.1. The values in table 2.1 are given as: Q^{in} [m^3/h], ΔP [Pa]. The opening degree α is unit less where 1 means fully open.

$\alpha \setminus \Delta P$	5	10	20	40
1	1000	1350	1940	2700
3/4	910	1300	1850	2580
1/2	690	980	1390	1970
1/4	360	430	700	980
1/8	170	240	330	460
1/16	90	120	170	260

Table 2.1: Flow capacity for inlet DA 1200 [m^3/h].

To find the constants in (2.38) and (2.37) first let $\boldsymbol{C}^{n \times m}$ denote the measured values of air flow shown in table 2.1. n is the number of values for opening degrees, m denotes the number of pressure differences. Vector y is defined as:

$$y = \begin{bmatrix} \boldsymbol{C}_{1,1} & \boldsymbol{C}_{2,1} & \dots & \boldsymbol{C}_{n,1} & \dots & \boldsymbol{C}_{1,m} & \dots & \boldsymbol{C}_{n,m} \end{bmatrix}^{\text{T}} \tag{2.42}$$

The values for inlet opening degree and pressure difference are put in the following vectors:

$$\alpha = \begin{bmatrix} 1 & 3/4 & 1/2 & 1/4 & 1/8 & 1/16 \end{bmatrix}^{\text{T}}$$

$$\Delta P = \mathcal{P} = \begin{bmatrix} 5 & 10 & 20 & 40 \end{bmatrix}^{\text{T}}$$

Constant for Non-Leakage Model

The model for inflow without leakages (2.37) only has one constant to be determined. Following (Montgomery, 2001) the signal vector is designed as:

$$x_{\mathrm{nl}} = \begin{bmatrix} \mathcal{P}(1) \cdot \alpha \\ \mathcal{P}(2) \cdot \alpha \\ \vdots \\ \mathcal{P}(m) \cdot \alpha \end{bmatrix} \tag{2.43}$$

The constant k is then found as:

$$k = x_{nl}^{+} \cdot y \tag{2.44}$$

where $^{+}$ denotes the pseudoinverse, yielding a least square solution to:

$$\min |k - x_{nl} \cdot y|^2 \tag{2.45}$$

For the data set in table 2.1 this yields $k = 90/3600$.

Constants for Leakage Model

The leakage model has two constants: k and l. The signal matrix then becomes:

$$x_{\mathrm{wl}} = \begin{bmatrix} \mathcal{P}(1) \cdot \alpha & \mathcal{P}(1) \cdot 1_{n \times 1} \\ \mathcal{P}(2) \cdot \alpha & \mathcal{P}(2) \cdot 1_{n \times 1} \\ \vdots & \\ \mathcal{P}(m) \cdot \alpha & \mathcal{P}(m) \cdot 1_{n \times 1} \end{bmatrix} \tag{2.46}$$

The least squares solution to $x_{wl}^{+} \cdot y$ consequently yields a 2×1 vector with:

$$\begin{bmatrix} k \\ k \cdot l \end{bmatrix} = x_{wl}^{+} \cdot y \tag{2.47}$$

For the data presented in table 2.1 this yields $k = 81/3600$ and $l = 0.08/3600$.

Quality of Fit

The data in table 2.1 has been used as basis for estimating the constants in (2.38) and (2.37). Table 2.2 presents the measured data, data for the model without and with leakage modeling in the following format: $y/\hat{Q}_{\mathrm{nl}}^{\mathrm{in}}/\hat{Q}_{\mathrm{wl}}^{\mathrm{in}}$.

28

$\alpha \setminus \Delta P$	5	10	20	40
1	1000/451/437	1350/901/873	1940/1803/1747	2700/3605/3493
3/4	910/338/336	1300/676/672	1850/1352/1343	2580/2704/2687
1/2	690/225/235	980/451/470	1390/901/940	1970/1803/1880
1/4	360/113/134	430/225/269	700/451/537	980/901/1074
1/8	170/56/84	240/113/168	330/225/336	460/451/671
1/16	90/28/59	120/56/117	170/113/235	260/225/470

Table 2.2: Measured and estimated values for flow capacity for inlet DA 1200 $[\mathrm{m}^3/\mathrm{h}]$.

To quantify the quality of fit the following norm is used:

$$|\cdot| = \frac{\Sigma(y - \hat{Q}^{\mathrm{in}})^2}{\Sigma(y - \bar{y})} \tag{2.48}$$

where $\bar{y}$ denotes the mean value. For the non-leakage model (2.48) yields 0.24 and 0.22 for the leakage model.

2.5.6 Numerical Values for Ventilation Fan

Experiments conducted at Skov A/S has led to the results for a ventilation fan presented in table 2.3.

$u \setminus \Delta P$	0	10	20	30	40
10	10196	9739	9231	8664	8026
9.4	9734	9281	8776	8178	7431
9	9143	8718	8116	7320	6185
8	7781	7198	6465	5327	4011
7	6468	5753	4643	3084	2021
6	5098	4122	2284	1084	0
5	3768	1990	517	0	0
4.5	2897	1009	0	0	0
4	2416	0	0	0	0
3.5	1697	0	0	0	0

Table 2.3: Flow capacity for ventilation fan $[\mathrm{m}^3/\mathrm{h}]$.

Using the same approach for parameter estimation as with the inlet models, the following constants are found: $c = 1021/3600$ and $d = 113/3600$. The difference between the measured data and the model is shown in figure 2.4. Using (2.48) yields 0.15.
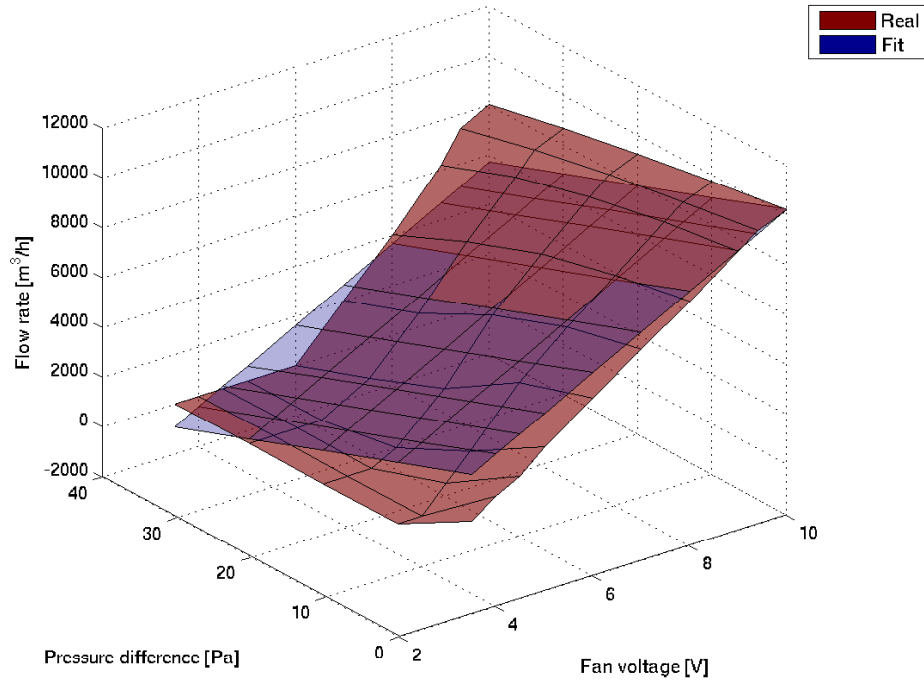
Figure 2.4: Measured data and model for flow rate through ventilation fan.

2.5.7 Modelling Climate Dynamics

A basic assumption in the enclosed articles is that interdependence of climatic parameters between zones is dominated by internal air flow between the zones. Non of the models presented, therefore, models radiation, convective heat transfer etc. By applying the flow balance (2.41), the following model for temperature in the ith zone is easily obtained:

$$
\begin{aligned}
\dot{T}_i V_i =& T^{\text{amb}} Q_i^{\text{in}} - T_i Q_i^{\text{fan}} + [Q_{i-1,i}]^+ T_{i-1} - [Q_{i-1,i}]^- T_i - \\
& [Q_{i,i+1}]^+ T_i + [Q_{i,i+1}]^- T_{i+1} + \frac{\Sigma \phi_i}{\rho_{\text{air}} c_{\text{air}}}
\end{aligned}
\tag{2.49}
$$

where V_i is the zone volume, T^{amb} is ambient temperature, $\Sigma \phi_i$ is the sum of heat sources (animals and/or controlled heating). It should be noted that modelling heat transfer through the building envelope has been deliberately omitted. As explained in (Jessen and Schiøler, 2006a) it is easy to include in parameter estimation if needed.

The use of square brackets in (2.49) is defined as:

$$[x]^+ \triangleq \max(0, x), \ [x]^- \triangleq \min(0, x) \tag{2.50}$$

The flows to/from a zone and the use of square brackets is illustrated in figure 2.5.
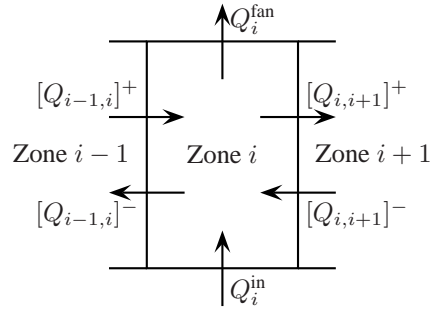
Figure 2.5: Illustration of flows for zone i.

It should be remarked specifically that the flows in (2.49) can not be set arbitrarily by the control signals $\{\alpha\}$ and $\{u\}$. The flows are given by simultaneous solving (2.38) through (2.41).

Paper 1: COTS Technologies for Integrating Development Environment, Remote Monitoring and Control of Livestock Stable Climate

J. J. Jessen* H. Schiøler* J. F. D. Nielsen,* M. R. Jensen[†]

Abstract

In this paper we present a flexible environment for development of climate control systems, especially aimed for livestock building facilities. Matlab/SimuLink as a de facto standard is used as development environment and is integrated with a remote monitoring and control system, allowing users to act as human in the loop controllers. For demonstration of technology maturity the entire system is developed using commercial off-the-shelf (COTS) technologies namely: PC architecture, Linux, Matlab/Simulink/RTW, MySql, Apache and ssh. The presented system is considered as prestudy/case for a future industrial solution.

3.1 Introduction

Danish pig and poultry production has on a national level for the last 30 to 40 years been characterized by fewer and fewer farmers with ever increasing holdings while the number of employees has decreased. The location of the individual livestock buildings is often physically located far from the farmers main building making it necessary to drive from one location to another in order to oversee the individual productions.

---

*Department of Control Engineering, Centre for Embedded Software Systems, Aalborg University, Fredrik Bajers Vej 7C, Aalborg Ø, Denmark

[†]Skov A/S, Glyngøre, 7870 Roslev Denmark

For intensive production systems e.g. pigs and poultry it is of crucial importance that an optimal climate is maintained at all times, since large climate deviations can be lethal for the animals and subsequently result in significant loss of profit. This necessitates the use of alarms in the computers controlling the climate inside the livestock buildings in case of climate component breakdown, or other unexpected behavior. A system allowing the farmer to automatically oversee climate conditions inside a number of livestock buildings and possibly act as a *human in the loop controller*, would not only increase the efficiency of the farmer since he can inspect multiple buildings simultaneously, but also decrease the time the animals would experience unwanted climatic conditions.

Aalborg University has in cooperation with Skov - a major Danish provider of climate control systems for livestock facilities - established a full scale laboratory for conducting research within climate control of livestock buildings. The laboratory provides not only a testbed for climate control algorithms but also a platform for testing remote monitoring of climatic conditions. The use case for the laboratory is that a user should be able to use a mobile device (e.g. PDA) to monitor the climatic conditions inside the stable and override the climate controller and act as a human in the loop controller without being physical present at the location.

3.1.1 Technologies

COTS technologies/products are mostly designed for mass markets not taking into account specific needs for individual applications. But because of budget and time to market demands, the use of COTS is gaining ground in applications ranging from engine simulation (Guerra and Marsch, 1997), data warehouse (Yale, 1997) and space applications (Ngo and Harris, 2001). Here we put forth a design philosophy in which the entire system is build from standard software components - commercial of-the-shelf (COTS) - and where the integration of COTS components relies only on well recognized open standards.

The article is to be seen as a prestudy for a possible industrial application, and as such there exists two entities of interest: Development environment and the embedded industrial solution. We present here the framework for integration of both entities; a climate control development environment supporting remote monitoring and control based on COTS technologies. The laboratory is located at a former poultry production facility located in the northern part of Denmark. The laboratory is equipped with climatic sensors, control computer and appropriate mechanisms for creating disturbances in form of heat production that would normally be experienced in a livestock building.

3.1.2 Previous Work

Adopting the COTS approach with respect to both hardware and software, has lead us to implement the control environment with a standard PC architecture using Linux as operating system and commercial IO-cards. The development environment is Matlab/SimuLink - a de facto standard in the control engineering and research community.

With the Real Time Workshop toolbox (RTW) it is possible to generate executable code from a SimuLink diagram. This idea is presented in (Quaranta and Mantegazza, 2001; Bucher and Balemi, 2006) where Matlab/SimuLink is used with RTW to generate a real time control application for an electro-mechanical application. Using Matlab/SimuLink to generate real time control applications is also presented in (cič, 1998) except that solution is targeted towards specific hardware.

The thermodynamic system presented in this article has many similarities with e.g. greenhouse climate control or in general just heating ventilation and air conditioning (HVAC) systems. Common for such systems is that dominating time constants τ typically are large ($\tau \gg 1$ s). A major advantage thus exists compared to the results of (Quaranta and Mantegazza, 2001; Bucher and Balemi, 2006; cič, 1998); because a standard Linux kernel easily fulfills the real time demands given. It is our experience that this makes both implementation and maintenance in real life products more flexible, following the KISS mantra (Keep It Simple Stupid).

The article is outlined as follows: First we shortly introduce some of the research being conducted in the laboratory, followed by a description of the laboratory setup. We then present the system concept followed by system component integration. Finally we present the overall proof of concept and conclude with guidelines for further research.

3.2   Conducted Research

It is widely known that the airspace inside livestock buildings is imperfectly mixed see e.g. (Barber and Ogilvie, 1982, 1984; Moor and Berckmans, 1996; Janssens et al., 2004). This has inspired the notion of zones, a concept by which a number of rectangular sub-areas of the stable are defined - see figure 3.1.
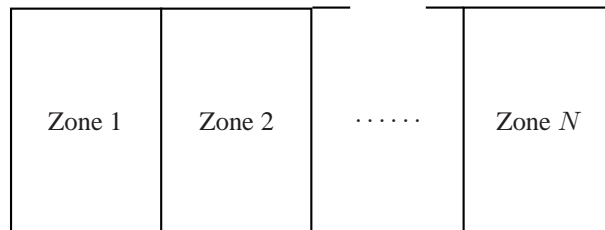
Figure 3.1: Series connection of N zones. Top-down view.

Each of the zones in figure 3.1 is equipped with ventilation equipment (air inlets, ventilation fans and heating) and is capable of exchanging air with neighboring zones, thus causing an internal airflow sideways in the building.

The introduction of zones and internal airflow introduces a number of challenges with respect to climate control algorithms. Without the zone idea two differential equations would typically be used to model the climate (temperature and humidity). With

the introduction of zones 2^N coupled differential equations are needed, and because of internal air flow theses equations include a sign function making them nonlinear.

Models and simulations of any plant behavior is inevitable in a controller design cycle, and indeed much of the research conducted with the laboratory focuses on modeling the climate inside livestock buildings. Within climate control of livestock buildings and greenhouses suggestions range from controllers based on statistical single zones (Berckmans and Goedseels, 1986), steady state analysis (Schauberger et al., 2000), stochastic models for greenhouse climate (Nielsen and Madsen, 1998), fuzzy logic controllers (Caponetto et al., 2000) to ventilation rate control (Taylor et al., 2004). Our research that is expected to use the stable as a testbed is currently focused on Model Predictive Control (MPC) and control synthesis based on game theory (Persis et al., 2006).

3.3 Climate Laboratory Overview

The climate laboratory is based on a former poultry stable (see figure 3.2), equipped with sensors, actuators, computer and a broadband Internet connection.

Figure 3.2: Decommissioned poultry stable acting as climate laboratory.

The stable is partitioned into three logically separated zones as illustrated in figure 3.3.

The climate control components consists of air inlets, air outlets and heating. For the three zones there are separate controls of inlets and heating in both sides of the zones as depicted in figure 3.3. Each zone has a ventilation fan associated to it, and the two zone boundaries have a ventilation fan.
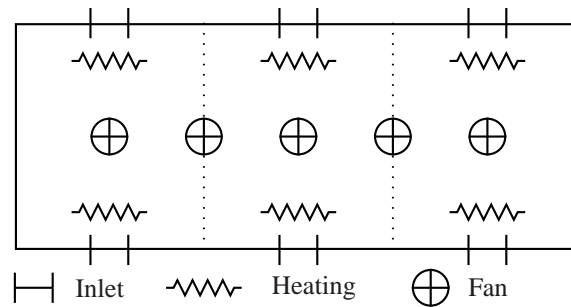
Figure 3.3: Illustration of climate components in zone partitioning.

Air Inlets

Fresh air is provided via air inlets build into the wall. An air inlet comprises a wall opening and a guide plate controlling air flow quantity and the direction of the supplied air. Outside the inlet is a fixed plate protecting from direct wind gusts. The guiding plate inside is controlled by a motor via a pulley system, and it has sensors mounted making the opening degree measurable.

Air Outlets

An air outlet comprises a chimney with an electrically controlled fan inside. In front of the fan is a rotating plate that can be set between 0 degrees (closed) and 90 degrees (open). On the shaft of the rotating plate is a mounted a sensor making the opening degree measurable.

Heating

The stable has a central boiler and a pipe system to each zone, where a number of radiators are connected. A valve system connected to the central boiler makes it possible to distribute the hot water to the individual radiators in the zones. These valves can be set in an arbitrary position between 0 degrees (closed) and 90 degrees (fully open).

All actuator components can be set in an arbitrary position between closed and fully open, or stopped and maximum speed.

3.4 System Concepts

In the following we shall emphasize on two major aspects of the remote laboratory facility; namely the embedded platform and the development environment.

3.4.1   Embedded Climate Control Platform

For any industrial application price, remote service and maintenance, robustness and integration with other software/hardware components are of major importance. While the system designer needs to have the Matlab environment installed in the presented system, we aim to show the feasibility of allowing the use of freely available COTS components on the embedded platform for production. The individual components are characterized by their wide and long term use. The track record of each component thus commends the use of these components.

The platform is build around a standard PC-architecture using Linux as operating system and with commercial IO-cards. This has the advantage that most of the software is open source, free and well tested. Three IO-cards from National Instruments are used for actuation and data acquisition.

Actuator Control

To get in contact with the IO-cards the Comedi library is used (Scleef, 2006). Comedi is a kernel module providing an API to user space programs as illustrated in figure 3.4. This has the advantage of making the application independent of the vendor specific IO-card. The three IO-cards each have a specific IO-driver associated, which the Comedi module gives access to.
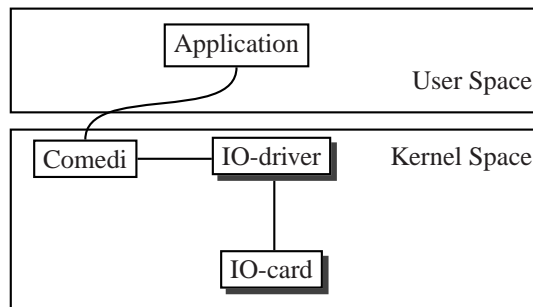
Figure 3.4: PC architecture with Comedi.

Remote Monitoring

The Linux PC is installed with Apache-2 as web server, php and MySQL database. Using only server side scripting with php and validated html code puts very few demands on the client platform (laptop, smart phone, pda etc.), since it only needs to support a standard web browser. While we postpone the discussion of security to future research we initially rely upon https as the communication protocol. To have a well defined interface between the web server and applications on the computer a MySQL database

is used. This database stores sensor values and retrieves user settings when controlled by human in the loop. For maintenance the PC is running a ssh server allowing remote administrator log-in from any computer with a ssh client installed.

3.4.2   Development Environment

Matlab/SimuLink is often the preferred tool for computer aided control system design (CACSD) for control engineers and research societies. With the Real Time Workshop toolbox (RTW) this development environment is not only a design and simulation environment, but also facilitates auto generation of production ready executable code from SimuLink diagrams.
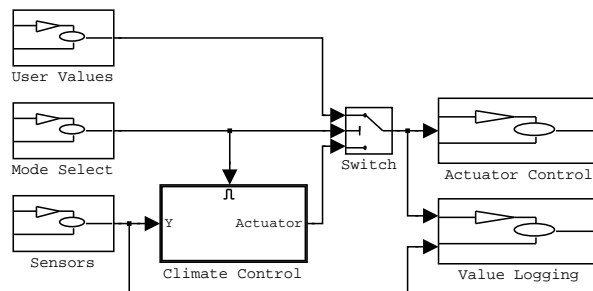
Figure 3.5: SimuLink diagram with user input and IO connection.

Figure 3.5 illustrates the concept for the development environment: At some point in any development cycle a controller has been designed, simulated and verified. A subsequent step is to generate a running application for the industrial platform. Figure 3.5 illustrates that the executable reads a mode select, which decides if user values are to be used for the actuators or some feedback algorithm should be employed. The climate control algorithm is put in an enabled subsystem, meaning it is only executed if the mode select is set properly. With this approach we thus have a simple mechanism for integrating user interaction into executable code generated with RTW. Both sensor values and actuator values are logged in the logging subsystem. With the construction illustrated in figure 3.5 integrator windup is avoided since execution of the control algorithm is suspended, when the mode select enforces *human in the loop* control.

As demonstrated in (Bucher and Balemi, 2006) it is possible to use SimuLink s-function wrapper functions for communicating with Comedi. A SimuLink s-function is a mechanism that allows to integrate custom code into SimuLink, and with this approach we have developed s-functions for communication with IO-cards through Comedi. The same approach is used for communicating to the MySQL database; mysql read and write functions are implemented with s-functions, thus allowing the climate control application to read user input and write sensor and log data to/from the database.

The developed s-functions for data acquisition, reading user values from a MySQL database and writing data to the database have been gathered in a SimuLink library as shown in figure 3.6.
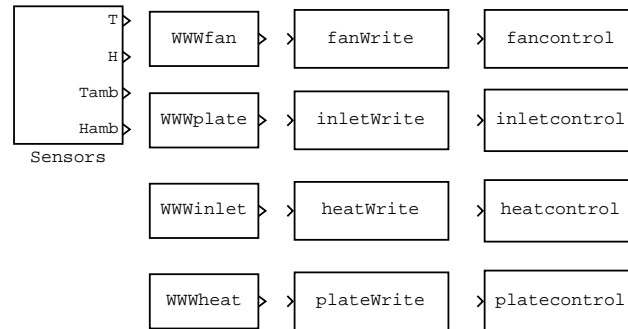
Figure 3.6: Simulink library for external communication.

The `Sensors` library shown in figure 3.6 reads the output voltage from the climatic sensors and outputs the corresponding temperature and relative humidity for the three zones and the ambient environment. `WWWx` blocks read the values set by the user in the MySQL database and corresponding `xwrite` blocks write to the database. The `xcontrol` blocks write data to the actuators.

3.5   System Component Integration

The remote controlled stable allows the user to set values directly to the actuators, overriding the climate controller. To avoid potential problems with integrator windup, climate control algorithms are only executed if explicitly stated via a manual/automatic flag in the database. This is done by the use of enabled subsystems in SimuLink, which means that the generated code for a particular subsystem only gets executed when then flag is set to 1 (see figure 3.5). This requires the control of the actuators to be decoupled from the climate controller. This is done with a classical cascade control loop as shown in figure 3.7.

The inlets and plates in front of the ventilation fans have position sensors mounted, making it possible to implement feedback controllers as shown in figure 3.7. The local controllers are implemented with a sampling frequency of 1 s. The ventilation fans and motor valves use feed forward control since no sensors are implemented. Cascading the climate controller from the actuator control thus implies that the actuator control block in figure 3.5 in fact is a actuator reference value block.

40
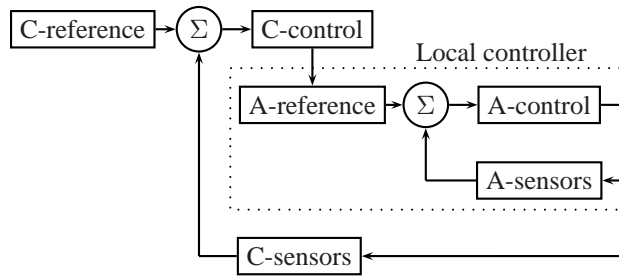
Figure 3.7: Overall control structure. C is climate and A is actuator.

3.5.1 MySQL as Component Integrator

In (Aaslyng et al., 2005) a custom application was developed for integrating a climate controller for a greenhouse with an environmental controller. The inter component connection in that approach was a database acting as buffer between the different computers and programs, and we adopt the same approach here. Via the developed s-function library shown in figure 3.6 SimuLink is capable of communicating with the MySQL database, and php has build in functions for communicating with MySQL databases. We thus have the basic components for integrating the web server with the climate controller. Figure 3.8 illustrates how the web interface, climate controller and actuator controller are connected.
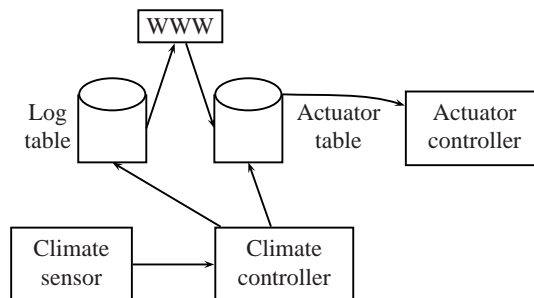
Figure 3.8: Connection of system components. Arrows illustrate information flow.

From the web interface measurement values are displayed to the user via a logging table in the database. The climate controller reads sensor values, calculates actuator reference values and writes both to the logging table. The climate controller also writes actuator reference values into the actuator table, which is read by the actuator controller. From the web interface the user can set a manual/automatic flag in the database, which decides if the climate controller or user should have write access to the reference values.

3.6 Proof of Concept

Using the library shown in figure 3.6 makes it possible to use SimuLink for generating code for the actuator controller and climate controller. The designer of the actuator controller then only has to drag and drop the blocks for reading database values, writing using Comedi and create a SimuLink diagram as shown in figure 3.9
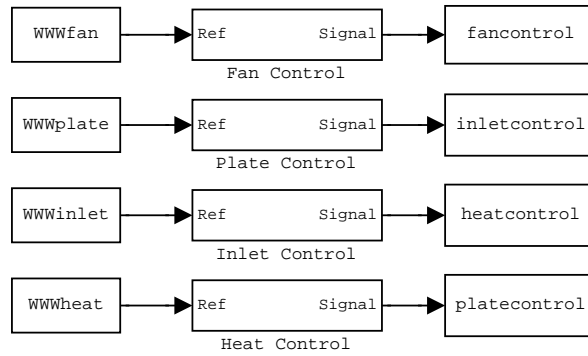
Figure 3.9: Actuator controller.

The control specific algorithms goes into the individual controller blocks in figure 3.9. The same approach applies for the climate controller as long as it is put in an enabled subsystem as shown in figure 3.5.

The web interface reads log data from the database and creates a graph with a 20 minutes window for both temperature and relative humidity. The graphs are created with PHPlot (Ottenheimer, 2006) - an open source project for creating graphs using php. A screen shot from the web interface is shown in figure 3.10

As shown in figure 3.10 the graph shows the ambient temperature and humidity and the temperature/humidity in the three zones. On the horizontal axis is a time stamp from the MySQL database. Using the button in the top of the web interface, makes it possible to set the manual/automatic flag in the database, and subsequently set values for the actuator for ventilation fans, ventilation plate, heating and inlets. The web interface is still in an early development version, but is has currently been tested to work using the following browsers: Firefox for Linux, Opera for Linux, Internet Explorer for Windows and Internet Explorer for Pocket PC.
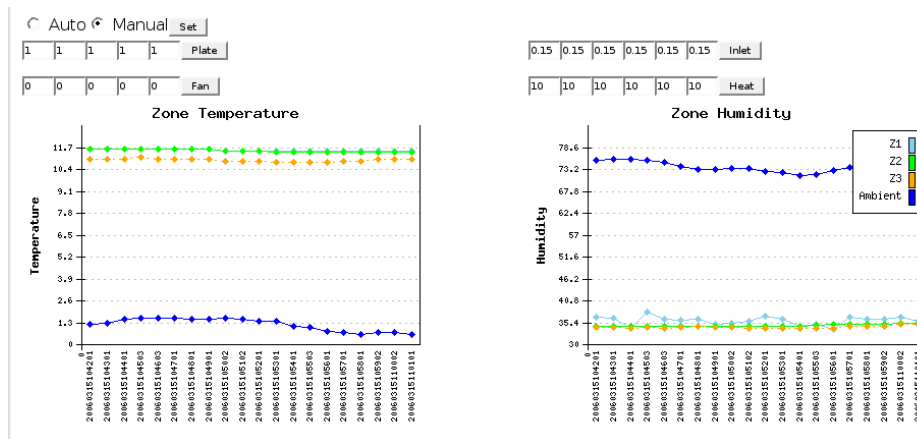
Figure 3.10: Screen shot showing zone temperature from web interface.

3.7 Conclusion

As a prestudy for a possible industrial application, we have demonstrated the feasibility of using COTS technologies to develop a uniform framework for developing climate control algorithms. Though the specific case is for livestock buildings we strongly believe the solution is suitable for other industrial applications too. The Matlab/SimuLink framework with RTW has been extended to include support for the Comedi library and we added mechanisms for reading and writing from a MySQL database. Through the use of an Apache web server with php server side scripting, we have demonstrated how a generated program from the Matlab environment can have a web interface with MySQL as component integrator.

3.7.1 Future Research

The presented system has proofed, by its very existence, that it is possible to develop an industrial application using COTS technologies - many of them being free and open source. Next we need to demonstrate the applicability of using COTS in a commercial product. We thus guide our future research towards security in industrial computers using Linux, focusing on the threats to embedded devices connected to the Internet, as well as focusing on predictability of the prioritized duties in the operating system. In case of component breakdown (hardware or software) the solution need to be robust, since the physical location of livestock buildings often are far from a technician. We therefore plan to investigate redundancy and graceful degradation with respect to component errors.

Paper 2: COTS Technologies for Internet Based Monitoring of Livestock Buildings

J. J. Jessen[*]   J. F. D. Nielsen[*]   H. Schiøler[*]   M. R. Jensen[†]

Abstract

In this paper we present an Internet based remote monitoring system for climate control in livestock buildings. The system is developed entirely using commercial off-the-shelf (COTS) technologies, allowing the farmer to monitor and set reference values for the climate controller using a mobile device connected to the Internet. The presented system is considered as pre-study/case for a possible industrial solution.

KEY WORDS
COTS, Linux, Climate Control, Remote Monitoring, Internet Architectures.

4.1   Introduction

As Danish pig and poultry production is characterized by fewer farmers with larger holdings, it is of crucial importance that the individual farmer can monitor the climate conditions in a number of stable buildings remotely. Danish regulation for animal production facilities requires that the farmer owns an amount of land per animal, which often means that livestock buildings are placed far from each other requiring the farmer to drive from location to location overseeing the individual productions.

---

[*]Department of Control Engineering, Centre for Embedded Software Systems, Aalborg University, Fredrik Bajers Vej 7C, Aalborg Ø, Denmark

[†]Skov A/S, Glyngøre, 7870 Roslev Denmark

A system where the farmer can monitor the climatic conditions inside the individual stables remotely, will allow for a more structured planning of the everyday duties. The farmer can then focus on planing with respect to feeding, coordination with veterinary officials etc. always knowing he has, for the animals, life critical information available.

4.1.1 Vision for Danish Livestock Production

Climate control, without considering welfare, is a matter of keeping temperature, humidity and air quality sufficiently close to specified reference values under well known conditions, and is as such an exercise in feedback control. However, the external environmental factors in livestock production make it extremely difficult to set a correct reference point for a climate control system, since not only weight, but also breed, age, daily feed intake and the surroundings e.g. time of day and building/floor type affect the animals environmental requirements. Adding unknown disturbances and varying system dynamics as well as the possibility of component failure in sensors, actuators, communication systems and computers, the task is of far greater complexity than seen at a first glance.

The difficulties with setting a correct reference point has led us to a vision for future climate control systems, in which the animals welfare is used to control the climate. In (Hansen, 2004) a general framework is presented allowing to use welfare based information in a feedback control loop, and we believe the presented approach is feasible for implementing welfare based climate control. Denmark has a reputation of being one of the worlds most wage intensive countries, so we are in general interested in a system where the entire production is characterized by a high level of automation. By incorporating sensors for the physical properties of the animals e.g. weight, a feedback loop controlling animal growth could be the foundation for a completely automated livestock production. For this purpose models of animal growth is needed (see e.g. (Bastianelli and Sauvant, 1997; Moughan et al., 1995) for models of pig growth).

With the presented vision in mind, Aalborg University has in cooperation with Skov - a major Danish provider of climate control systems for livestock facilities - established a full scale laboratory for conducting research within climate control of livestock buildings. The laboratory provides not only a testbed for climate control algorithms but also a platform for testing remote monitoring of climatic conditions and sensor fusion for welfare sensors.

4.1.2 Technologies

COTS technologies/products are mostly designed for mass markets not taking into account specific needs for individual applications. But because of budget and time to market demands the use of COTS is gaining ground in applications ranging from engine simulation (Guerra and Marsch, 1997), data warehouse (Yale, 1997) and space applications (Ngo and Harris, 2001).

Here we put forth a design philosophy in which the entire system is build from standard software components - commercial of-the-shelf (COTS) - and where the integration of COTS components relies only on well recognized open standards.

4.1.3 Previous Work

Adopting the COTS approach with respect to both hardware and software, has lead us to implement the control environment with a standard PC architecture using Linux as operating system and commercial IO-cards. The development environment is Matlab/SimuLink a de facto standard in the control engineering and research community. With the Real Time Workshop toolbox (RTW) it is possible to generate executable code from a SimuLink diagram. This idea is presented in (Quaranta and Mantegazza, 2001; Bucher and Balemi, 2006) where Matlab/SimuLink is used with RTW to generate a real time control application for an electro-mechanical application. Using Matlab/SimuLink to generate real time control applications is also presented in (cič, 1998) except that solution is targeted towards specific hardware.

The thermodynamic system presented in this article has many similarities with e.g. greenhouse climate control or in general just heating ventilation and air conditioning (HVAC) systems. Common for such systems is that dominating time constants τ typically are large ($\tau \gg 1$ s). A major advantage thus exists compared to the results of (Quaranta and Mantegazza, 2001; Bucher and Balemi, 2006; cič, 1998); because a standard Linux kernel easily fulfills the real time demands given. It is our experience that this makes both implementation and maintenance in real life products for industry more flexible, following the KISS mantra (Keep It Simple Stupid).

The article is outlined as follows: First we briefly present some of the research being conducted in the laboratory, followed by a description of the laboratory setup. We then present the systems concepts and the integration of system components. We conclude with a proof of concept and guidelines for our future work.

4.2 Conducted Research

It is widely known that the airspace inside livestock buildings is imperfectly mixed see e.g. (Barber and Ogilvie, 1982, 1984; Moor and Berckmans, 1996; Janssens et al., 2004), and real life livestock buildings are not always uniform (geometrically) and may even have leakages in the building shelf. This has inspired the notion of zones, a concept by which a number of rectangular subareas of the stable are defined - see figure 4.1.

Each of the zones in figure 4.1 is equipped with ventilation equipment (air inlets, ventilation fans and heating) and is capable of exchanging air with neighboring zone(s), thus causing an internal airflow sideways in the building.
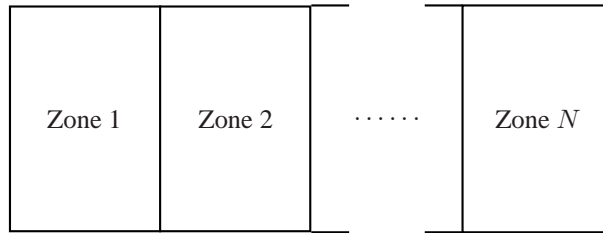
Figure 4.1: Series connection of N zones. Top-down view.

The introduction of zones and internal airflow introduces a number of challenges with respect to climate control algorithms. Without partitioning into zones, two differential equations would typically be used to model the climate (temperature and humidity). With the introduction of zones 2^N coupled differential equations are needed, and because of internal air flow theses equations include a sign function making them nonlinear.

Models and simulations of any plant behavior is inevitable in a controller design cycle, and indeed much of the research conducted with the laboratory focuses on modeling the climate inside livestock buildings. Within climate control of livestock buildings and greenhouses suggestions range from controllers based on statistical single zones (Berckmans and Goedseels, 1986), steady state analysis (Schauberger et al., 2000), stochastic models for greenhouse climate (Nielsen and Madsen, 1998), fuzzy logic controllers (Caponetto et al., 2000) to ventilation rate control (Taylor et al., 2004). Our research that is expected to use the stable as a testbed is currently focused on Model Predictive Control (MPC), controller synthesis based on game theory and sensor fusion based on Bayesian networks.

4.3 Climate Laboratory Overview

The climate laboratory is based on a former poultry stable (see figure 4.2), equipped with sensors, actuators, computer and a broadband Internet connection. The stable is partitioned into three logically separated zones as illustrated in figure 4.3.

The climate control components consists of air inlets, air outlets and heating. For the three zones there are separate controls of inlets and heating in both sides of the zones as depicted in figure 4.3. Each zone has a ventilation fan associated to it, and the two zone boundaries have a ventilation fan.

4.3.1 Air Inlets

Fresh air is provided via air inlets build into the wall. An air inlet comprises a wall opening and a guide plate controlling air flow quantity and the direction of the supplied

Figure 4.2: Decommissioned poultry stable acting as climate laboratory.
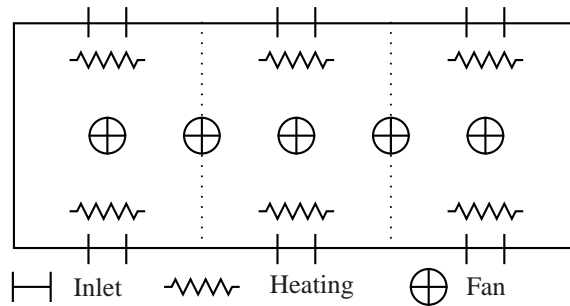
Figure 4.3: Illustration of climate components in zone partitioning.

air. Outside the inlet is a fixed plate protecting from direct wind gusts. The guiding plate inside is controlled by a motor via a pulley system, and has sensors mounted making the opening degree measurable.

4.3.2 Air Outlets

An air outlet comprises a chimney with an electrically controlled fan inside. In front of the fan is a rotating plate that can be set between 0 degrees (closed) and 90 degrees (fully open). On the shaft of the rotating plate is a mounted a sensor making the opening degree measurable.

4.3.3 Heating

The stable has a central boiler and a pipe system to each zone, where a number of radiators are connected. A valve system connected to the central boiler makes it possible to distribute the hot water to the individual radiators in the zones. These valves can be set in an arbitrary position between 0 degrees (closed) and 90 degrees (fully open). All actuator components can be set in an arbitrary position between closed and fully open, or stopped and maximum speed.

4.4 System Concepts

In the following we shall emphasize on a major aspect of the remote laboratory facility; namely the embedded platform. For any industrial application price, remote service and maintenance, robustness and integration with other software/hardware components are of major importance. While we previously stated that Matlab/SimuLink is used to develop the climate controller, we omit the discussion of the development environment itself, and aim to show the feasibility of allowing the use of freely available COTS components on the embedded platform for production. The individual components are characterized by their wide and long term use. The track record of each component thus commends the use of these components.

The platform is build around a standard PC-architecture using Linux as operating system and with commercial IO-cards. This has the advantage that most of the software is open source, free and well tested. Three IO-cards from National Instruments are used for actuation and data acquisition.

4.4.1 Actuator Control

For connectivity with the IO-cards the Comedi library is used (Scleef, 2006). Comedi is a kernel module providing an API to user space programs as illustrated in figure 4.4. By avoiding directly communication with vendor specific drivers, Comedi has the advantage of allowing the application to be independent of the specific IO-cards. The three IO-cards each have a specific IO-driver associated, which the Comedi module gives access to.

4.4.2 Remote Monitoring

The Linux PC is installed with Apache-2 as web server, php and MySQL database. Using only server side scripting with php and validated html code puts very few demands on the client platform (laptop, smart phone, pda etc.), since it only needs to support a standard web browser. While we postpone the discussion of security to future research we initially rely upon https as the communication protocol. To have a well defined interface between the web server and applications on the computer a MySQL database
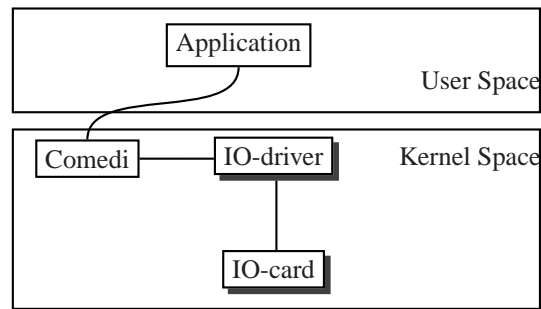
Figure 4.4: PC architecture.

is used. This database stores sensor values and retrieves user settings when controlled by human in the loop. For maintenance the PC is running a ssh server allowing remote administrator log-in from any computer with a ssh client installed.

As demonstrated in (Bucher and Balemi, 2006) it is possible to use SimuLink s-function wrapper function for communicating with Comedi. SimuLink s-functions is a mechanism that allows to integrate custom code into SimuLink, and with this approach we have developed s-functions for communication with IO-cards through Comedi. The same approach is used for communicating to the MySQL database; mysql read and write functions are implemented with s-functions, thus allowing the climate control application to read user input and write sensor and log data to/from the database.

4.5   COTS Integration and Control

The remote controlled stable allows the user to set values directly to the actuators, overriding the climate controller and reference points for the controller. To avoid potential problems with integrator windup, climate control algorithms are only executed if explicitly stated via a manual/automatic flag in the database there by allowing human in the loop control. This requires the control of the actuators to be decoupled from the climate controller, which is implemented with a classical cascade control loop.

As stated in section 4.1.2 we exploit the fact that the systems we are interested in are characterized by large time constants. This allows us to use COTS components in a way they were not originally implemented for e.g. the MySQL database.

Figure 4.5 illustrates the aforementioned decoupling of actuator and climate controllers. The actuator controllers are implemented with a sampling frequency of 1 s, while the climate controller sampling frequency is an order of magnitude slower. This opens the possibility of using COTS as shown in figure 4.5, where a standard Apache web server and MySQL database is used to give remote access to the farmer.
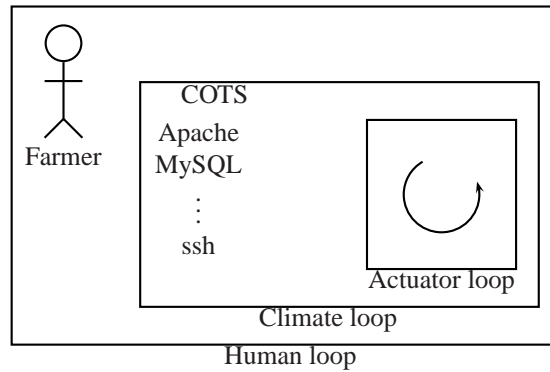
Figure 4.5: Illustration of control loop.

4.5.1 MySQL as Component Integrator

In (Aaslyng et al., 2005) a custom application was developed for integrating a climate controller for a greenhouse with an environmental controller. The inter component connection in that approach was a database acting as buffer between the different computers and programs, and we adopt the same approach here. Via a developed s-function library, SimuLink is capable of communicating with the MySQL database, and php has build in functions for communicating with MySQL databases. We thus have the basic components for integrating the web server with the climate controller. Figure 4.6 illustrates how the web interface, climate controller and actuator controller are connected.
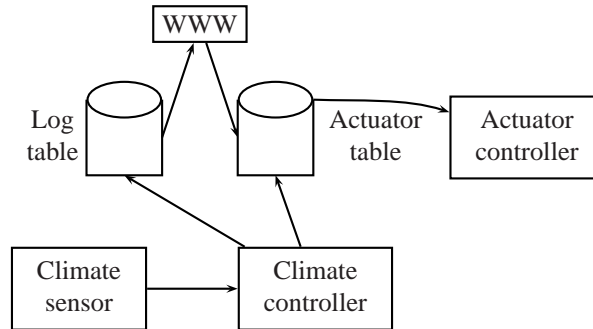
Figure 4.6: Connection of system components. Arrows illustrate information flow.

From the web interface measurement values are displayed to the user via a logging table in the database. The climate controller reads sensor values, calculates actuator reference values and writes both to the logging table. The climate controller also writes actuator reference values into the actuator table, which is read by the actuator controller.

From the web interface the user can set a manual/automatic flag in the database, which decides if the climate controller or user should have write access to the reference values.

4.6 Proof of Concept

With the presented approach, various COTS components have been integrated to develop a uniform framework for a remote controlled climate laboratory. In figure 4.6 it is illustrated how MySQL is used to interconnect system component. Figure 4.7 illustrates the overall system architecture with a farmer who uses a PDA to access the climate controller in the livestock building.
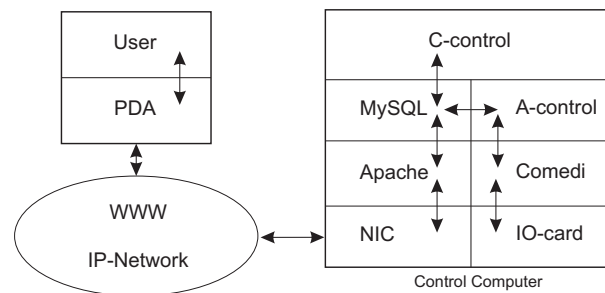
Figure 4.7: Component integration of overall system concept.

The control computer consists of network interface card (NIC), Apache web server, MySQL database, climate control program, actuator control, Comedi library and IO-cards. Figure 4.7 illustrates the applicability of using a web based solution to remote monitoring; the farmers choice of mobile device (or office computer) does not influence the architecture of the embedded platform, nor does the connectivity. The website on the climate computer uses server side scripting and validated html code only, which has the obvious advantages of not being limited to specific Internet browsers. Though the web interface still is in an early development version, is has currently been tested to work using the following browsers: Firefox for Linux, Opera for Linux, Internet Explorer for Windows and Internet Explorer for Pocket PC.

The developed web interface allows the user to choose between graphs for temperature and relative humidity, with a 20 minute interval. Figure 4.8 and figure 4.9 is a screen shot from the temperature and humidity graphs. The graphs in figure 4.8 and 4.9 are created with PHPlot - an open source project for creating graphs using php. Though the web interface has been tested with the aforementioned Internet browsers, it is currently being optimized for the screen size for an IPAQ 6340.

53
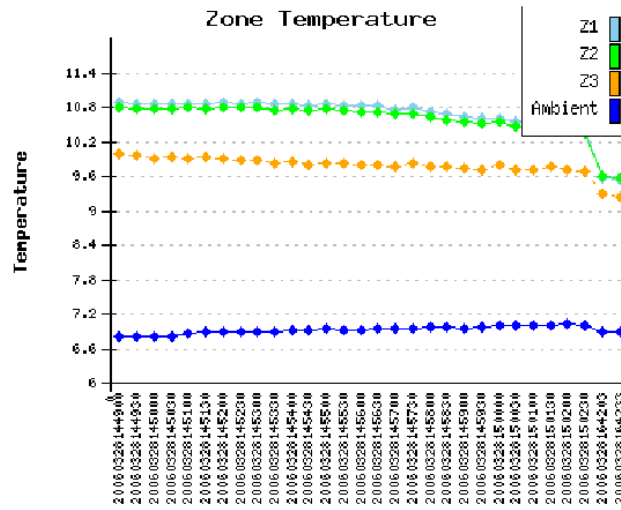
Figure 4.8: Screenshot from temperature graph.
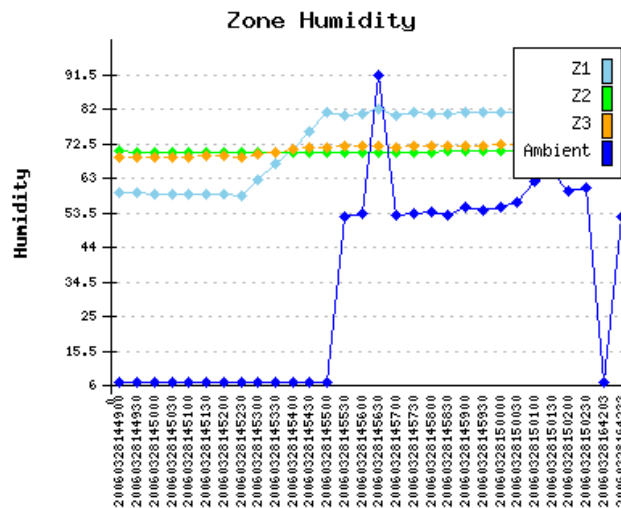
Figure 4.9: Screenshot from humidity graph.

4.7 Conclusion

We have demonstrated the feasibility of using COTS components/technologies for a possible industrial application. Though we have omitted details on the climate controller and its development environment we have showed how to use MySQL as an interconnection component between the sub parts of the entire system. The presented solution is targeted specifically for climate control in livestock buildings, but we strongly believe the solution is suitable for other industrial applications too.

4.7.1 Future Research

The presented system has proofed, by its very existence, that it is possible to develop an industrial application using COTS technologies - many of them being free and open source. Before we proceed with real life product implementation we guide our attention towards a general embedded platform design where we emphasize the following: Choice of Linux distribution, securing Linux for industrial applications, multipurpose IO, hardware/software redundancy and general system reliability and graceful degradation.

Paper 3: ZoneLib: A Simulink Library for Modeling Zone Divided Climate Dynamics

J. J. Jessen[*]   H. Schiøler[*]

Abstract

We present a dynamic model for climate in a livestock building divided into a number of zones, and a corresponding modular Simulink library (ZoneLib). While most literature in this area consider air flow as a control parameter we show how to model climate dynamics using actual control signals for the ventilation equipment. To overcome a shortcoming in Simulink to solve algebraic equations and matrix inversions, we have developed the library inspired by the so called dynamic node technique. We present simulation results using the presented library, and concludes with visions for further development of ZoneLib.

5.1   Introduction

Modern pig production is characterized by huge stock densities and large scale production facilities. For such intensive production systems it is extremely important that the climate control system is working properly, since a failure in a climate computer may result in the death of entire batches of livestock and additional loss of significant revenue.

---

[*]Department of Control Engineering, Centre for Embedded Software Systems, Aalborg University, Fredrik Bajers Vej 7C, Aalborg Ø, Denmark

Indispensable parts in the design process for any controller are models and simulations of the plant under investigation. For such modeling and simulation it is often desired to maintain a high abstraction level focusing on modeling aspects and not implementation details. Matlab/SimuLink from MathWorks (Inc., 2006) is a de-facto standard within the control engineering and research community, which offers the opportunity to use a graphical point and click interface to plant modeling, thus allowing the control engineer to focus on the modeling details.

It is widely known that the airspace inside livestock buildings is imperfectly mixed see e.g. (Barber and Ogilvie, 1982, 1984; Moor and Berckmans, 1996; Janssens et al., 2004). This has inspired the notion of zones, a concept by which a number of subareas of the stable are defined. Each of these subareas (zones) have individual ventilation components installed as illustrated in figure 5.1. We consequently assume that climatic parameters are perfectly mixed in each zone i.e. there are no significant gradients.
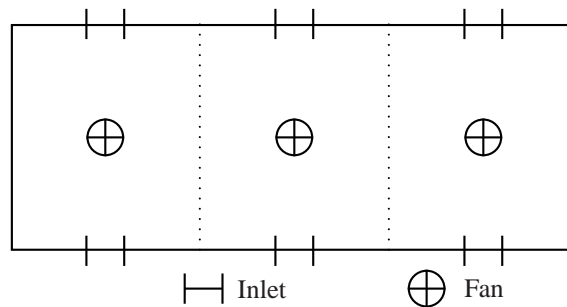
Figure 5.1: Series connection of zones with individual ventilation components.

In the present work we present a simulation model for a stable divided into a number of distinct climatic zones, suitable for later development of new climate control algorithms. The presented model requires simultaneous solution of a number of equations, which is implemented easily in Matlab using matrix inversion. This is however a shortcoming in Simulink; algebraic loops and matrix inversion. In order to overcome this we present ZoneLib; a Simulink library for modeling climate dynamics inspired by the dynamic node technique (Flinders and Oghanna, 1997; Flinders et al., 1993; Hansen et al., 1995).

5.1.1 Related Work

Similarities to the zone concept can be found in (Caponetto et al., 2000) where a so called multi layer model for a greenhouse is presented which divides the greenhouse in eight layers, whose thickness is proportional to the length from the southbound wall.

In (Berckmans and Goedseels, 1986) a steady state analysis is presented, and in (Moor and Berckmans, 1996) it is argued that classical computational fluid dynamics of-

ten are developed for simulation in steady state and therefore not applicable for control. Instead a grey box model is suggested, defined as a model based on physical laws combined with a parametric identification technique. In (Schauberger et al., 2000) a steady state model is used to asses the indoor climate and air quality of a pig stable, resembling the model for thermal energy balance for a greenhouse in (Gutman et al., 1993), except that (Gutman et al., 1993) includes the integral of the temperature to account for the slowly varying dynamics, and as such the accumulated impact of temperature on animal conditions.

Common for the above mentioned papers is that they all consider the ventilation rate as a control signal, not being concerned about *how* the ventilation rate is achieved. In the present work we have developed a simulation model for control purposes, including the actual control signals to the ventilation equipment. The task of modeling climate dynamics and beeing concerned about actual control signals have been covered in (Linker et al., 1999), where the ventilation rate is used as input to a neural network, which then outputs the actual control signals based on a desired air flow.

The paper is outlined as follows: First we present the modeling of ventilation components showing how to find flows. We then present a model for temperature in a zone based building ending with an algorithm for simulation using realistic control signals. We then present our developed Simulink library and a simulation using the presented library. We conclude with guidelines for further work.

5.2 Modeling Ventilation Components

The ventilation components in the stable are air inlets and air outlets. We assume pressure dynamics to be orders of magnitude faster than the mixing dynamics for e.g. temperature, humidity and CO2, so for a given control signal stationary values for pressures and air flows may be assumed. As an example we present the modeling of ventilation components in a three zone stable.

5.2.1 Air Inlets

An air inlet build into the wall is basically an opening in the wall with a guiding plate to adjust the direction and amount of incoming air. On the outside of the wall a protecting plate is mounted to minimize the effect of direct wind gusts on the inlet. We suggest the following approximate model for airflow Q^{in} [m^3/s] into the ith zone:

$$Q_i^{\text{in}} = k_i \cdot \alpha_i \cdot (P^{\text{amb}} - P_i) \tag{5.1}$$

Where P_i [Pa] is the pressure in zone i, P^{amb} is the ambient pressure, k_i is a constant and α_i is the control signal to the inlet determining the opening degree of the guiding plate.

5.2.2 Air Outlets

The air outlets are chimneys with an electrically controlled fan inside. We propose the following model which has a linear dominant term acting on the control signal (voltage applied to the fan) and a term subtracting flow due to pressure difference. The exhaust air flow Q^{fan} [m³/s] from the ith zone is thus found as:

$$Q_i^{\text{fan}} = u_i \, c_i - d_i \, (P^{\text{amb}} - P_i) \tag{5.2}$$

where u_i is the control signal for the fan, c_i and d_i are constants.

5.2.3 Inter Zone Net Air Flow

The net air flow $Q_{i,i+1}^{\text{st}}$ [m³/s] between two zones (air flows are defined positive from a lower index to a higher index) is found by

$$Q_{i,i+1}^{\text{st}} = e_i \, (P_i - P_{i+1}) \tag{5.3}$$

where e_i is a constant.

5.2.4 Stationary Flows

A stationary balance for each zone i is found:

$$Q_{i-1,i}^{\text{st}} + Q_i^{\text{in}} = Q_{i,i+1}^{\text{st}} + Q_i^{\text{fan}} \tag{5.4}$$

where by definition $Q_{0,1}^{\text{st}} = Q_{N,N+1}^{\text{st}} = 0$. Figure 5.2 illustrates the flow to/from a zone.
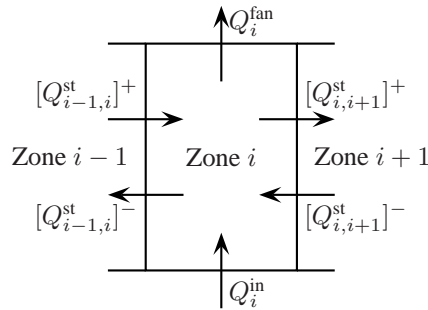
Figure 5.2: Illustration of flows for zone i.

We define the use of square brackets illustrated in figure 5.2 as:

$$[x]^+ \triangleq \max(0, x), \ [x]^- \triangleq \min(0, x) \tag{5.5}$$

Using the corresponding expressions for the flow terms in 5.4 we can set up the following expression:

$$A\,x = b \tag{5.6}$$

where vector x solves for the corresponding flows and pressures. With that construction we now have that entries in matrices A and b depend linearly on control variables $\{\alpha_i\}$ and $\{u_i\}$. We define the state vector as:

$$x = \begin{bmatrix} Q_1^{\text{in}} & Q_2^{\text{in}} & Q_3^{\text{in}} & Q_1^{\text{fan}} & Q_2^{\text{fan}} & Q_3^{\text{fan}} & Q_{1,2}^{\text{st}} & Q_{2,3}^{\text{st}} & P_1 & P_2 & P_3 \end{bmatrix}^{\text{T}} \tag{5.7}$$

and

$$A = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_1\alpha_1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_2\alpha_2 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_3\alpha_3 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -d_1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -d_2 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -d_3 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -e_1 & e_1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -e_2 & e_2 \\
1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & -1 & 0 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0
\end{bmatrix} \tag{5.8}$$

and

$$b = \begin{bmatrix} k_1\alpha_1 P^{\text{amb}} & k_2\alpha_2 P^{\text{amb}} & k_3\alpha_3 P^{\text{amb}} & u_1 c_1 - d_1 P^{\text{amb}} \\ u_2 c_2 - d_2 P^{\text{amb}} & u_3 c_3 - d_3 P^{\text{amb}} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{\text{T}}$$

5.3   Modeling Climate Dynamics

Though it would be relevant to model temperature, humidity, CO2 and ammonia we initially limit ourselves to modeling only temperature, in order to illustrate the zone concept. It would though be easy to include the disregarded climate parameters since the mixing dynamics roughly are identical.

Assumption 5.3.1 *Climatic interdependence between zones is assumed solely through internal air flow.*

With assumption 5.3.1 we thus neglect radiation and diffusion etc. between zones, claiming they are negligible compared to the effect from having air flow. Keeping in mind the flow balance (5.4) - c.f. figure 5.2 the following model for temperature T $[\text{˚C}]$

61

is easily obtained for the ith zone:

$$
\dot{T}_i V_i = T^{\text{amb}} Q_i^{\text{in}} - T_i Q_i^{\text{fan}} + [Q_{i-1,i}^{\text{st}}]^+ T_{i-1} - [Q_{i-1,i}^{\text{st}}]^- T_i -
$$

$$
[Q_{i,i+1}^{\text{st}}]^+ T_i + [Q_{i,i+1}^{\text{st}}]^- T_{i+1} + \frac{W_i^t}{\rho_{\text{air}} c_{\text{air}}} \tag{5.9}
$$

where V_i [m^3] is the zone volume, T^{amb} is ambient temperature, W^t [W] is the heat production from the pigs, c_{air} [J/kg/$^\circ$C] is the air heat capacity and ρ_{air} [kg/m^3] is the air density. $\dot{T}_i$ denotes the derivative $\frac{d}{dt} T_i$.

Rewriting 5.9 using vector notation we thus have for the three zone case:

$$
\mathbf{V}\dot{\mathbf{T}} = \mathbf{QT} + \mathbf{H} \tag{5.10}
$$

with

$$
\mathbf{V} = \begin{bmatrix} V_1 & 0 & 0 \\ 0 & V_2 & 0 \\ 0 & 0 & V_3 \end{bmatrix}
$$

$$
\mathbf{T} = \begin{bmatrix} T_1 & T_2 & T_3 \end{bmatrix}^{\text{T}}
$$

$$
\mathbf{Q} = \begin{bmatrix} -(Q_1^{\text{fan}} + [Q_{1,2}^{\text{st}}]^+) & [Q_{1,2}^{\text{st}}]^- \\ [Q_{1,2}^{\text{st}}]^+ & -(Q_2^{\text{fan}} + [Q_{1,2}^{\text{st}}]^- + [Q_{2,3}^{\text{st}}]^+) \\ 0 & [Q_{2,3}^{\text{st}}]^+ \end{bmatrix}
$$

$$
\begin{matrix} 0 \\ [Q_{2,3}^{\text{st}}]^- \\ -(Q_3^{\text{fan}} + [Q_{2,3}^{\text{st}}]^-) \end{matrix} \Bigg]
$$

$$
\mathbf{H} = \begin{bmatrix} T^{\text{amb}} Q_1^{\text{in}} + \frac{W_1^t}{\rho_{\text{air}} c_{\text{air}}} \\ T^{\text{amb}} Q_2^{\text{in}} + \frac{W_2^t}{\rho_{\text{air}} c_{\text{air}}} \\ T^{\text{amb}} Q_3^{\text{in}} + \frac{W_3^t}{\rho_{\text{air}} c_{\text{air}}} \end{bmatrix}
$$

In order to simulate the temperature for M time steps we define algorithm 1, while we assume a numeric finite difference method in order to solve (5.10).

Algorithm 1 Simulate temperature using actual control signals.

$\mathbf{T}(0) \leftarrow \mathbf{T}^{\text{init}}$
for $k = 0$ to $M - 1$ **do**
    $\mathbf{A}, b \leftarrow \{\alpha, u\}$
    $x = \mathbf{A}^{-1} b$
    $\mathbf{Q}, \mathbf{H} \leftarrow x$
    $\mathbf{T}(k+1) \leftarrow (\mathbf{V}^{-1}(\mathbf{QT}(k) + \mathbf{H}), \mathbf{T}(k))$
end for

With algorithm 1 it is possible to simulate the temperature in a zone divided stable using realistic control signals. For changing control signals e.g. by a controller, the α and u terms should just be vectors of length M.

5.4   Using ZoneLib for Climate Modeling

When the presented model is simulated in Matlab, we have been able to use the assumption that pressure dynamics are an order of magnitude faster than climate dynamics. In algorithm 1 we thus used stationary values for pressure at each simulation step. Simulink however has a shortcoming with algebraic loops and matrix inversion, so we add artificial pressure dynamics in order to overcome this.

5.4.1   Dynamic Node Technique

The idea of the dynamic node comes from electric circuit simulation. Consider the diagram illustrated in figure 5.3 consisting of two elements and the dynamic node (Flinders et al., 1993). The dynamic node is to be seen as a virtual node i.e. it is included to make simulation/analysis easier. The meaning of the dynamic node is to model parasitic capacitances in which, for a small transient period, the sum of currents into/out from the node does not equal zero. In steady state, however, the sum of inflow/outflow of current is in balance and the voltage is constant.
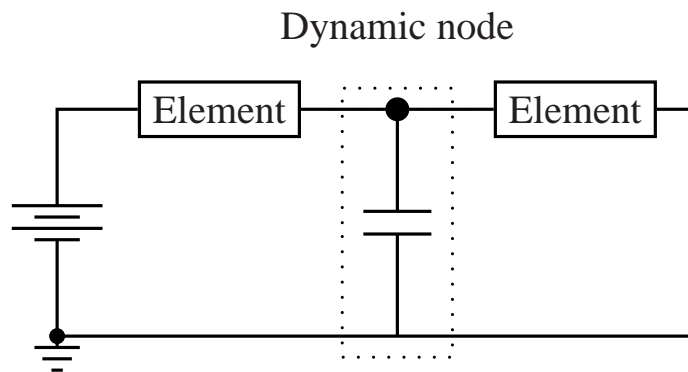
Figure 5.3: Dynamic node technique in electric circuit simulation.

The dynamic node technique can be used to solve algebraic equations. In figure 5.4 two situations are shown. First two blocks A and B are shown, where block A requires some external input and the output from block B. Block B also requires an external input and the input from block A. This is known as an algebraic loop which can be seen as a sort of deadlock since each block is waiting for the other block to calculate its output.
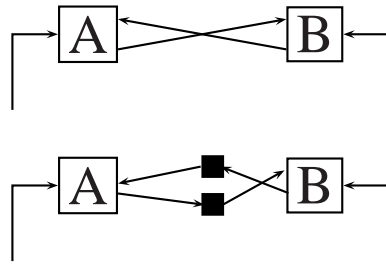
Figure 5.4: Dynamic node technique to solve algebraic loops.

In the bottom of figure 5.4 two black boxes are added on the line from each of the boxes to the opposite box. The black boxes denotes dynamics (e.g. first order) and are thus used to avoid the two algebraic loops.

5.4.2 Pressure Dynamics in Simulink

Using the idea of the dynamic node, we include the pressure dynamics in the Simulink library, as illustrated in figure 5.5.
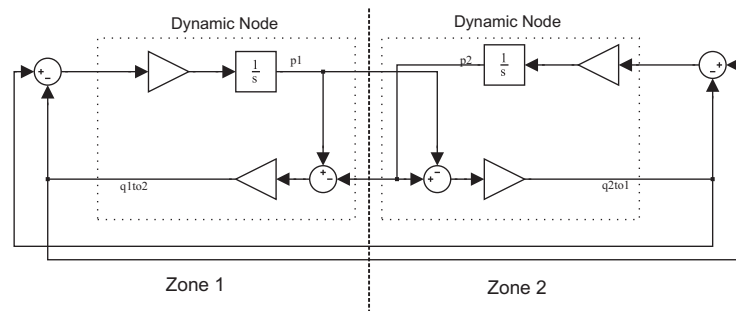
Figure 5.5: Modeling pressure dynamics in Simulink.

Figure 5.5 shows a Simulink model of two zones which models the pressure by integrating the net flow into a zone ($\frac{1}{s}$ is used as symbol for integration). We have thus avoided algebraic loops since the information path for pressure always encounters an integrator. The presented approach has been used to develop ZoneLib shown in figure 5.6.

The inlet and fan blocks in ZoneLib are Simulink implementations of (5.1) and (5.2) respectively. Three different zone blocks are implemented: Zone 1, Zone i and Zone N, which allows to model series connections of an arbitrary number of zones. The ith zone takes as its left input the right output from zone $i-1$, and its right input is the left output
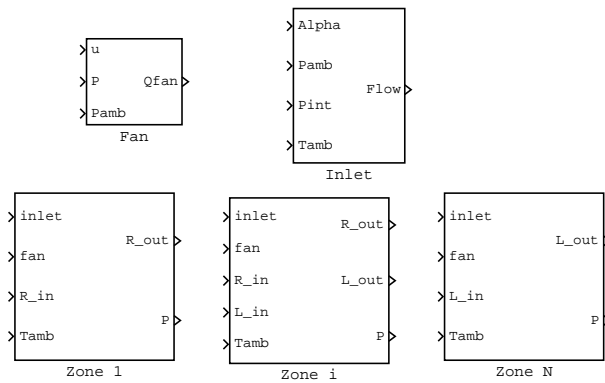
Figure 5.6: Available blocks in ZoneLib.

from zone $i + 1$. Due to space limitations we omit the presentation of the blocks: Zone 1 and Zone N, and present a generic Zone i in figure 5.7.
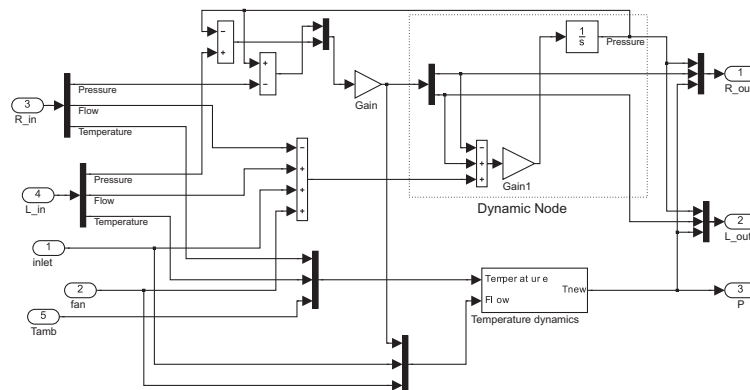
Figure 5.7: Generic ZoneLib block for Zone i.

The input to a zone block is ambient temperature and flow through inlets and fan. Input from/output to neighboring zones is a 3×1 vector consisting of pressure, flow and temperature. Each zone also have a separate output with pressure for easy connection with the inlet and fan blocks. The bottom right block in figure 5.7 implements the temperature dynamics in (5.9).

65

5.5 Simulation Results

The presented library is used to model a stable divided into a series connection of three zones. In order to illustrate the versatility of ZoneLib we configure the simulation scenario as follows: Zones 1 and 2 have no ventilation components, while zone 3 has an inlet and a fan with fixed control signals set to $\alpha_3 = 8$, $u_3 = 10$. The initial states are $T = [23\ 24\ 26]$ and $P = [101300\ 101100\ 101000]$. The simulink diagram using ZoneLib is shown in figure 5.8.
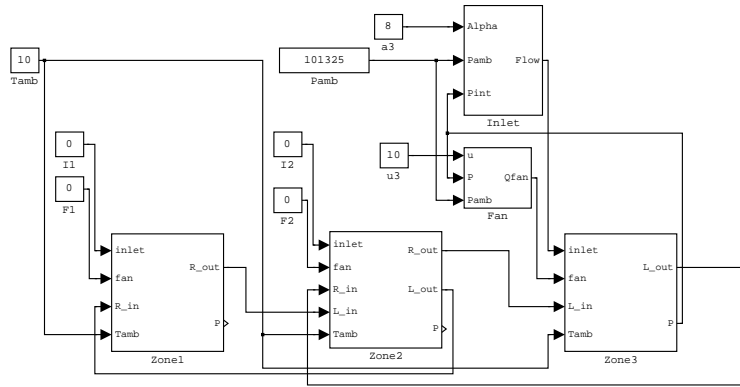
Figure 5.8: Simulink model of three zone stable using ZoneLib.

Each zone in figure 5.8 implements the temperature dynamics (5.9), but for the presented simulation we have omitted the presence of pigs in the stable i.e. $W_1^t = W_2^t = W_3^t = 0$, and set the zone volumes to $V_1 = V_2 = V_3 = 1000$. Figure 5.9 illustrates the evolution of temperature in the three zones, while figure 5.10 illustrates the evolution of pressure.

Figure 5.10 illustrates that the pressure in the three zones starts to converge approximately after 30 simulation steps towards the same value. This has the effect on the evolution of temperature that from 0 to simulation step 30 (approximately) air flows from zone 1 to zone 2 and from zone 2 to zone 3, resulting in a decreasing temperature in all the three zones. After the pressure in the three zones are equal it is only zone 3 that has a decrease in temperature towards the ambient temperature of 10 [˚C], since this is the only zone interacting with the ambient environment. This is a clear effect of assumption 5.3.1 that zones only interact by internal air flow. We intend to exploit this by actively causing internal air flow when we later develop controllers for multi zone livestock buildings.
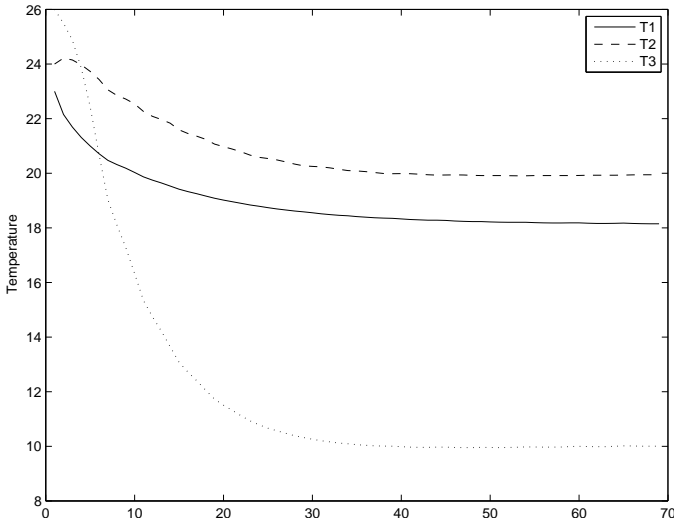
Figure 5.9: Simulation of temperature for three zone stable.
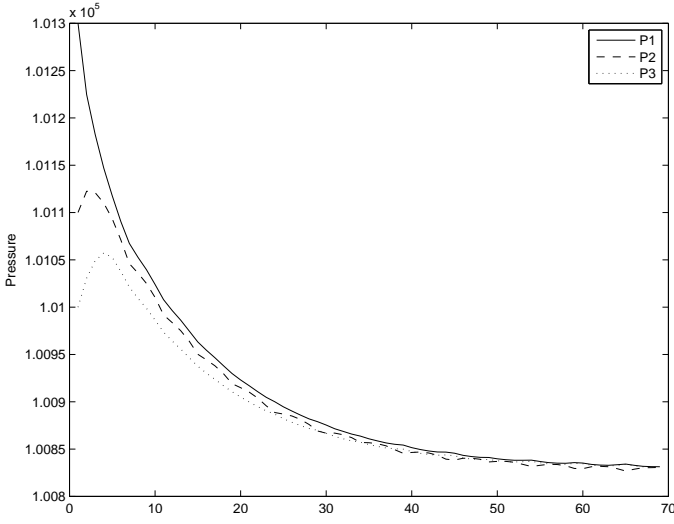
Figure 5.10: Simulation of pressure dynamic for three zone stable.

5.6 Conclusions and Further Work

We have presented a simulation algorithm for climate in a stable partitioned into a number of climatic zones, and shown how to model the temperature in Simulink using ZoneLib. The current available blocks in the library are: inlet, outlet, zone 1, zone i and zone N. We have presented a simulation illustrating the point and click feature of the library, in which an arbitrary number of zones can be connected and each zone can have its own set of ventilation components.

The presented library is still in an early development phase, and we focus our forthcoming activities on validating the model against a real data set. We are currently building a large scale test facility which is implementing separate actuation of ventilation components in three zones. We plan to use the test facility to, among other things, make a parameter estimation of the unknown constants in the presented models.

For the development of ZoneLib we plan to generalize it, so each zone can connect to other zones Manhattan wise i.e. North, South, East and West. Furthermore we plan to implement a more general fan block that can be connected to more than one zone. The zone blocks should also implement at least humidity dynamics and possible CO_2. Finally we plan to use the library to test new climate control algorithms that we are currently researching.

Paper 4: System Identification in a Multi Zone Livestock Building

J. J. Jessen*　　　　H. Schiøler*

Abstract

We present a technique for performing system identification in a closed environment partitioned into a number of distinct climatic zones. The technique is based on the idea of guaranteeing well defined directions of internal flow between zones. Each zone has a separate pair of inlets and outlets for ventilation purposes, and possible leakages in the building envelope. We present simulation results using the presented method and conclude with guidelines for further enhancements and outlines for a controller utilizing the proposed method.

6.1　Introduction

For any control system design cycle, modeling and simulation of plant and controller behavior is an inevitable part. Companies selling controllers in large quantities (e.g. cruise control systems, thermostats etc.) benefit from the fact that the uniformity of the products (due to large quantity) makes it possible to implement identical controllers in all the products. For companies selling controllers for larger plants e.g. power plants or livestock building climate control systems, it is often necessary to perform on site adjustments of the controllers, because the plants are different. This is especially true

*Department of Control Engineering, Centre for Embedded Software Systems, Aalborg University, Fredrik Bajers Vej 7C, Aalborg Ø, Denmark

for climate control systems for livestock buildings, which often differ in size and/or shape.

Companies developing livestock building climate control systems would benefit from having automatic tuning of controllers, since this would decrease the amount of time a technician would have to spend on fine tuning the final setup. The ultimate goal in this context would be a control system that only had to be installed, and then automatically performed a system discovery to find out about installed climate control components and where the manual entering of e.g. physical properties about the stable is minimal.

A practice often found in real life climate control systems for e.g. pig or poultry stables, is based on steady state analysis (Schauberger et al., 2000; Pedersen and Saalvik, 2002). This approach requires the farmer to enter a number of parameters about the animals e.g. weight/age and the number of animals. Based on this information a minimum ventilation can be calculated which ensures appropriate air quality. But even in more modern climate control approaches e.g. (Daskalov et al., 2004) or MPC information on the quantity of heat and moisture production from the animals is needed by the controller.

In this paper we present a method for performing system identification (SI) in a livestock building partitioned into a number of distinct climatic zones. The presented approach forms the basis for a future implementation of "intelligent" controllers utilizing SI and in which the information about animal age, weight and number is avoided.

That the airspace inside livestock buildings is imperfectly mixed is reported extensively in literature see e.g. (Barber and Ogilvie, 1982, 1984; Moor and Berckmans, 1996; Janssens et al., 2004). This has fostered the idea of multi zone climate control (see e.g. (Persis et al., 2006)), where a stable is divided into a number of zones and where each zone has its own ventilation equipment installed. The idea is illustrated in figure 6.1 where a stable is partitioned into 3 rectangular sub areas.
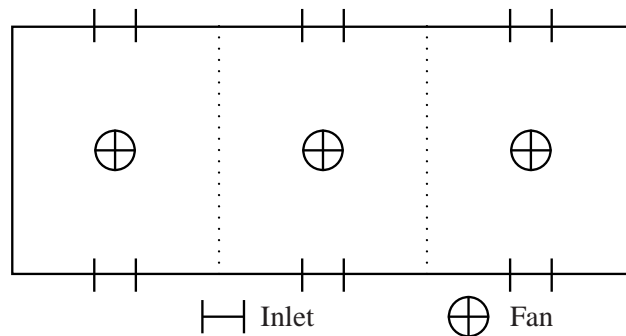
Figure 6.1: Series connection of zones with individual ventilation components.

The partitioning of the stable illustrated in figure 6.1 means that the zones interact with each other by internal air flow sideways in the stable. In the present work we present an early result from a proposed system identification technique. Though much work still need to be done, we believe the current status explains our idea satisfactory.

The paper is outlined as follows. First we introduce a model for temperature evolution in a zone divided stable, which is followed by the multi zone system identification technique. We then present a numerical study and conclude with guidelines for further work using the presented technique and further enhancements of the technique itself. Initially we limit our self to consider the temperature only.

6.2 System Description and Model

The ventilation actuator components in the stable are air inlets and air outlets. We assume pressure dynamics to be orders of magnitude faster than the mixing dynamics for e.g. temperature, humidity and CO2, so for a given control signal stationary values for pressures and air flows may be assumed.

6.2.1 Air Inlets

An air inlet is basically an opening in the wall with a guiding plate to adjust the direction and amount of incoming air. On the outside of the wall a protecting plate is mounted to minimize the effect of direct wind gusts on the inlet. We suggest the following approximate affine model for airflow from the outside through the wall inlet Q^{in} [m^3/s] in the ith zone:

$$Q_i^{\text{in}} = k_i \cdot (\alpha_i + l_i) \cdot (P^{\text{amb}} - P_i) \tag{6.1}$$

Where P_i [Pa] is the pressure in zone i, P^{amb} is the ambient pressure, k_i is a constant and α_i is the control signal to the inlet determining the opening degree of the guiding plate. We model possible leakages in the building envelope with l_i. We define the direction of airflow through an inlet from the outside to inside to be positive.

6.2.2 Air Outlets

The air outlets are chimneys with an electrically controlled fan inside. We propose the following model which has a linear dominant term acting on the control signal (voltage applied to the fan) and a term subtracting flow due to pressure difference. The exhaust air flow Q^{fan} [m^3/s] from the ith zone is thus found as:

$$Q_i^{\text{fan}} = u_i^f \, c_i - d_i \, (P^{\text{amb}} - P_i) \tag{6.2}$$

where u_i^f is the control signal for the fan, c_i and d_i are constants. The flow is defined positive from the stable to the outside.

71

6.2.3   Inter Zone Net Air Flow

The net air flow $Q_{i,i+1}$ [m^3/s] between two zones (air flows are defined positive from a lower index to a higher index) is found by

$$Q_{i,i+1} = m_i \left(P_i - P_{i+1} \right) \tag{6.3}$$

where m_i is a constant.

6.2.4   Stationary Flows

A stationary balance for each zone i is found:

$$Q_{i-1,i} + Q_i^{\text{in}} = Q_{i,i+1} + Q_i^{\text{fan}} \tag{6.4}$$

where by definition $Q_{0,1} = Q_{N,N+1} = 0$. Figure 6.2 illustrates the flow to/from a zone.
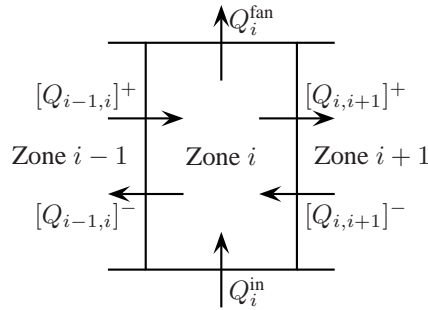
Figure 6.2: Illustration of flows for zone i.

We define the use of square brackets illustrated in figure 6.2 as:

$$[x]^+ \triangleq \max(0, x), \ [x]^- \triangleq \min(0, x) \tag{6.5}$$

6.2.5   Modeling Climate Dynamics

We assume that climatic interdependence between zones is dominated by internal air flow between the zones. We therefore omit to model radiation, convective heat transfer etc. and propose the following model for temperature in the ith zone:

$$
\begin{aligned}
\dot{T}_i V_i =& T^{\text{amb}} Q_i^{\text{in}} - T_i Q_i^{\text{fan}} + [Q_{i-1,i}]^+ T_{i-1} - [Q_{i-1,i}]^- T_i - \\
& [Q_{i,i+1}]^+ T_i + [Q_{i,i+1}]^- T_{i+1} + \frac{W_i^t}{\rho_{\text{air}} c_{\text{air}}}
\end{aligned} \tag{6.6}
$$

where V_i [m^3] is the zone volume, T^{amb} is ambient temperature, W^t [W] is the heat production from the animals, c_{air} [J/kg/$^\circ$C] is the air heat capacity and ρ_{air} [kg/m^3] is the air density.

6.2.6   Simulation Algorithm

Using the corresponding expressions (6.1), (6.2) and (6.3) for the flow terms in (6.4) we can set up the following expression:

$$A\,x = b \tag{6.7}$$

where vector x solves for the corresponding flows and pressures. For a three zone stable we define the state vector as:

$$x = \begin{bmatrix} Q_1^{\text{in}} & Q_2^{\text{in}} & Q_3^{\text{in}} & Q_1^{\text{fan}} & Q_2^{\text{fan}} & Q_3^{\text{fan}} & Q_{12} & Q_{23} & P_1 & P_2 & P_3 \end{bmatrix}^{\text{T}}$$

Due to space limitations we introduce an intermediate notation $o_i = \alpha_i + l_i$. The A and b matrices are then given as:

$$A = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_1 o_1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_2 o_2 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_3 o_3 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -d_1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -d_2 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -d_3 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -m_1 & m_1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -m_2 & m_2 \\
1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & -1 & 0 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}$$

and

$$b = \begin{bmatrix} k_1 o_1 P^{\text{amb}} & k_2 o_2 P^{\text{amb}} & k_3 o_3 P^{\text{amb}} & u_1 c_1 - d_1 P^{\text{amb}} \\ u_2 c_2 - d_2 P^{\text{amb}} & u_3 c_3 - d_3 P^{\text{amb}} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{\text{T}}$$

In order to simulate the temperature we define algorithm 2, while we assume a numeric finite difference method in order to solve (6.6). We use the notation $f(T^{\text{amb}}, Q)$ to denote the implementation of (6.6), where Q is the vector of flows.

With algorithm 2 we now have that the temperature in each zone can be simulated using actual control signals $\{\alpha\}, \{u\}$. This has the advantages over the models presented in e.g. (Berckmans and Goedseels, 1986; Moor and Berckmans, 1996; Schauberger et al., 2000; Gutman et al., 1993) since they consider the flow rate as a control signal. With algorithm 2 a model is created that is suitable for not only simulation but also control.

Algorithm 2 Simulate temperature using control signals.

$T_i^{\text{sim}}(1) \leftarrow T^{\text{init}}$
for $k = 1$ to $M - 1$ **do**
   $\mathbf{A}, b \leftarrow \alpha_i, u_i$
   $x = \mathbf{A}^{-1} b$
   $Q \leftarrow x$
   $T_i^{\text{sim}}(k + 1) \leftarrow (T_i^{\text{sim}}(k), f(T^{\text{amb}}, Q))$
end for

6.3   Multi Zone System Identification

The basic idea underlying our technique is to have a system description as:

$$\dot{x}_i = \theta_i \varphi_i \tag{6.8}$$

where $\dot{x}_i = \frac{d}{dt} T_i$, θ_i is the parameter vector and φ_i is the signal vector. If the bilinear parts from (6.6) is taken as input signals a standard linear regression model can be used to solve (6.8) (Montgomery, 2001). The model for temperature (6.6) makes it challenging to do system identification due to the $[\cdot]^+$ and $[\cdot]^-$ terms. We therefore propose to use an estimation signal which ensures well defined internal flow directions, meaning that the corresponding $[\cdot]^\pm$ terms can be omitted. As an example consider to open the inlets in zone 1 while closing the inlets in zone N and operating the ventilation fan in zone N at maximum speed. Then internal flow would only occur from left to right and only one model is needed to describe the climate dynamics.

6.3.1   Notation

Let $\alpha(k)$ denote the inlet signals in all zones at time k i.e.

$$\alpha(k) = [\alpha_1(k) \, \alpha_2(k) \, \ldots \, \alpha_N(k)]$$

and likewise with $u^f(k)$. We define a signal vector $U(k)$ as:

$$U(k) = \begin{bmatrix} \alpha(k) & u^f(k) \end{bmatrix}^{\text{T}} \tag{6.9}$$

and the control matrix $\boldsymbol{U}$ as:

$$\boldsymbol{U} = [U(1) \, U(2) \, \ldots U(M)] \tag{6.10}$$

We then assume the following

Assumption 6.3.1 *There exists a control matrix $\boldsymbol{U}$ s.t. $Q_{i,i+1} > 0$, $\forall i$*

In the present work we omit the discussion of the validity of assumption 6.3.1, and thus postpone the discussion of how to actually choose the control signals enforcing the direction of the internal flow. Since this work is concerned with simulations this does not pose a problem, but requires our outermost attention when we plan to do real life experiments.

6.3.2 Finding Dependent Control Signals

We wish to characterize the flow from the actuator signals $\{\alpha\}, \{u\}$ in order to construct the signal vector in (6.8). Using assumption 6.3.1 we rewrite (6.6) as:

$$\dot{T}_i V_i = T^{\text{amb}} Q_i^{\text{in}} - T_i Q_i^{\text{fan}} + Q_{i-1,i} T_{i-1} - Q_{i,i+1} T_i + \frac{W_i^t}{\rho_{\text{air}} c_{\text{air}}} \qquad (6.11)$$

To find the vector of external flows $Q^{\text{E}} = [Q_1^{\text{in}} \ Q_1^{\text{out}} \ \dots \ Q_i^{\text{in}} \ Q_i^{\text{out}} \ \dots \ Q_N^{\text{in}} \ Q_N^{\text{out}}]^{\text{T}}$ we set up the following:

$$Q^{\text{E}}(k) = H\alpha(k)^{\text{T}} + Ju^f(k)^{\text{T}} + R \qquad (6.12)$$

Where H and J are matrices. We include the R vector in order to express leakages in the building envelope as modeled in (6.1) with $l_i > 0$. Knowing the external flows gives the internal flow from the flow balance (6.4). We thus have that the vector of internal flows $Q^{\text{I}} = [Q_{1,2} \ Q_{i-1,i} \ Q_{i,i+1} \ Q_{N-1,N}]^{\text{T}}$ is given by:

$$Q^{\text{I}}(k) = \phi Q^{\text{E}}(k) \qquad (6.13)$$

where ϕ is a matrix. To find a specific internal flow we thus have:

$$\begin{aligned}
Q_l^{\text{I}}(k) &= \sum_j \phi_{lj} Q_j^{\text{E}}(k) \\
&= \sum_j \sum_i \phi_{lj} H_{ji} \alpha_i(k) + \phi_{lj} J_{ji} u_i(k) + \sum_j \phi_{lj} R_j
\end{aligned} \qquad (6.14)$$

Now from (6.14) we clearly have that internal and external flows are expressed as a linear combination of all the control signals.

6.3.3 Construction of Signal Vector

We define by $\mathbf{1}_{n \times m}$ a $n \times m$ matrix with 1 at all places, and let $\mathbf{T}_i$ denote the sequence of measured temperatures in the ith zone and likewise for $\mathbf{T}^{\text{amb}}$. We introduce an intermediate matrix $\mathcal{T}_i$ defined as:

$$\mathcal{T}_i = diag(\mathbf{T}_i) \qquad (6.15)$$

where $diag$ is the diagonal operator constructing a matrix with the sequence $\mathbf{T}_i$ on the diagonal. Knowing that (6.14) is needed to find (6.11) and using assumption 6.3.1 we

75

simply construct the signal matrix as:

$$\varphi_i = \begin{bmatrix} \boldsymbol{U}\mathcal{T}_{i-1} \\ \boldsymbol{U}(\mathcal{T}_i - \mathcal{T}^{\text{amb}}) \\ \mathbf{T}^{\text{amb}} \\ \mathbf{1}_{1 \times M} \end{bmatrix} \tag{6.16}$$

A remark on (6.16) is in order. The term $\boldsymbol{U}\mathcal{T}_{i-1}$ denotes the combination of all control signals and measured temperatures from the neighboring zone to the left (index $i-1$). This is clearly needed from (6.11). $\mathbf{T}^{\text{amb}}$ is included to account for leakages and is as such independent of the control signals. $\mathbf{1}_{1 \times M}$ is included to make a constant offset corresponding to the animals heat production.

With the signal matrix constructed in (6.16) we are left with the construction of the result vector. From a sequence of measurements we approximate $\dot{x}_i$ numerically as:

$$\dot{x}_i(k) = \frac{T_i(k+1) - T_i(k)}{\Delta_t} \ , k = 1, 2, \ldots, M - 1$$
$$\dot{x}_i(M) = \dot{x}_i(M-1) \tag{6.17}$$

where Δ_t is the sampling period. If $\boldsymbol{U}$ satisfies assumption 6.3.1, the parameter vector is given by:

$$\theta_i = \dot{x}_i \cdot \varphi_i^+ \tag{6.18}$$

where $^+$ denotes the pseudoinverse giving a least square solution to (6.8).

6.3.4 Simulating Temperature Using SI

Knowing the parameter vector θ_i for each zone, we are interested in simulating the temperature evolution T_i^{SI} in each zone. We therefore define algorithm 3 which uses θ_i from (6.18).

Algorithm 3 Simulate temperature using parameter vector.

$T_i^{\text{SI}}(1) \leftarrow T^{\text{init}}$
for $k = 1$ to $M - 1$ **do**
$\quad U(k) = [\alpha_1(k) \ \ldots \ \alpha_M(k) \ u_1(k) \ \ldots u_M(k)]^{\text{T}}$
$\quad \varphi_i(k) = [(T_i^{\text{SI}}(k) - T^{\text{amb}}(k))U(k)^{\text{T}}$
$\qquad\qquad (T_{i-1}^{\text{SI}}(k) - T^{\text{amb}}(k))U(k)^{\text{T}}$
$\qquad\qquad T^{\text{amb}}(k) \ \ 1]^{\text{T}}$
$\quad T_i^{\text{SI}}(k+1) = T_i^{\text{SI}}(k) + \theta_i \cdot \varphi_i(k)$
end for

6.4 Numerical Results

In this section we present the outcome of a simulation with the presented method. First we generate data with algorithm 2 in order to find the parameter vector. We then generate a new data set, and compare the result of simulating with the parameter vector with the new data set. We present the simulation for a stable partitioned into three zones.

We set the constants in (6.1), (6.2) and (6.3) to: $V_1 = V_2 = V_3 = 1500$, $k_1 = 0.7$, $k_2 = 0.5$, $k_3 = 0.6$, $l_1 = l_2 = l_3 = 0.1$, $c_1 = c_2 = c_3 = 0.3$, $d_1 = d_2 = d_3 = 0.03$, $m_1 = m_2 = 2$ and a constant ambient environment to: $T^{\text{amb}} = 13$ and $P^{\text{amb}} = 101325$.

We define a sequence of control signals with fixed values that have been found to fulfill assumption 6.3.1: $\alpha(k) = [8\ 5\ 3]$, $k = 1, 2, \ldots, M$ and $u^f(k) = [3\ 6\ 8]$, $k = 1, 2, \ldots, M$. Since a constant actuator signal vector is used and the ambient pressure is constant we get a constant solution to (6.7) resulting in $Q^{\text{E}} = [1.5\ 0.8\ 1.5\ 1.7\ 1.6\ 2.1]$ and $Q^{\text{I}} = [0.7\ 0.5]$. Setting the disturbance to $W_i^t/\rho_{\text{air}}/c_{\text{air}} = 2$, $i = 1, 2, 3$, the initial temperature to $T_1^{\text{init}} = 20$, $T_2^{\text{init}} = 45$, $T_3^{\text{init}} = 30$ using $M = 3000$ simulation steps result in the evolution of temperature in the three zones illustrated in figure 6.3.
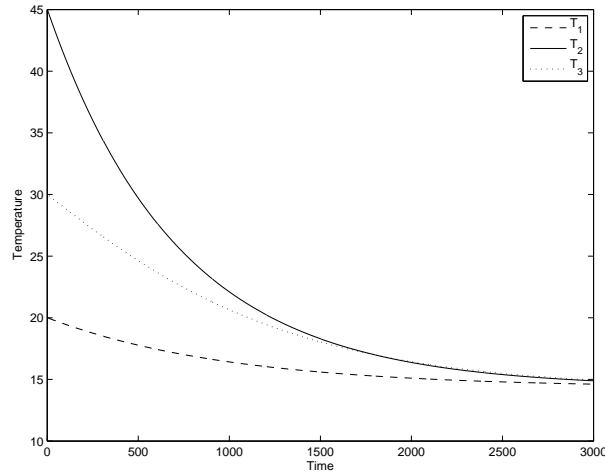
Figure 6.3: Simulation of temperature in a three zone stable.

Because a fixed control signal is used, the total signal matrix U becomes:

$$U = \begin{bmatrix} \begin{bmatrix} 8 & 5 & 3 \end{bmatrix}^{\text{T}} \cdot \mathbf{1}_{1 \times M} \\ \begin{bmatrix} 3 & 6 & 8 \end{bmatrix}^{\text{T}} \cdot \mathbf{1}_{1 \times M} \end{bmatrix} \tag{6.19}$$

77

For the first zone the signal matrix (6.16) reduces to:

$$\phi_1 = \begin{bmatrix} \boldsymbol{U}(\mathcal{T}_1 - \mathcal{T}^{\text{amb}}) \\ \mathbf{T}^{\text{amb}} \\ \mathbf{1}_{1 \times M} \end{bmatrix} \tag{6.20}$$

while ϕ_2 and ϕ_3 follows directly from (6.16). Solving (6.18) leads to the following parameter vectors for the three zones:

$$\theta_1 = 10^{-3} \cdot \begin{bmatrix} -0.04 & -0.02 & -0.01 & -0.01 \\ -0.03 & -0.04 & 0.10 & 0.01 \end{bmatrix}^{\text{T}} \tag{6.21}$$

$$\theta_2 = 10^{-3} \cdot \begin{bmatrix} 0.02 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.02 & -0.06 & -0.04 & -0.02 & -0.02 \\ -0.04 & -0.06 & -0.34 & -0.03 \end{bmatrix}^{\text{T}} \tag{6.22}$$

$$\theta_3 = 10^{-3} \cdot \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & -0.06 & -0.04 & -0.02 & -0.02 \\ -0.04 & -0.06 & -0.27 & -0.02 \end{bmatrix}^{\text{T}} \tag{6.23}$$

The parameter vectors θ_i (6.21), (6.22), (6.21) which were found using the data from the simulation illustrated in figure 6.3, are used to simulate the temperature in the three zone stable using new control signals that also satisfies assumption 6.3.1. Again we use fixed control signals setting the inlet signals to: $\alpha_1 = 6$, $\alpha_2 = 4$, $\alpha_3 = 2$ and the fan signals to $u_1 = 3$, $u_2 = 6$, $u_3 = 10$. The heat produces by the pigs are set to $W_i^t / \rho_{\text{air}} / c_{\text{air}} = 4$, $i = 1, 2, 3$. Figure 6.4 illustrates the outcome of the simulation.

Figure 6.4 shows that temperatures simulated with the parameter vectors are in good agreement with the temperatures simulated with the presented model (6.6). The dotted line in figure 6.4 illustrates the error between the two simulations with a constant offset. The largest error for the three simulated temperatures is found to: $e_1 = 0.64$, $e_2 = 0.76$, 0.85 which is satisfactory.

6.5   Conclusion

We have presented a simple method for performing system identification in a livestock building partitioned into a number of zones, under the assumption of having well defined internal flow directions. We have shown an algorithm for simulating temperature using realistic control signals, and shown how to find the parameter vector in a system identification model.
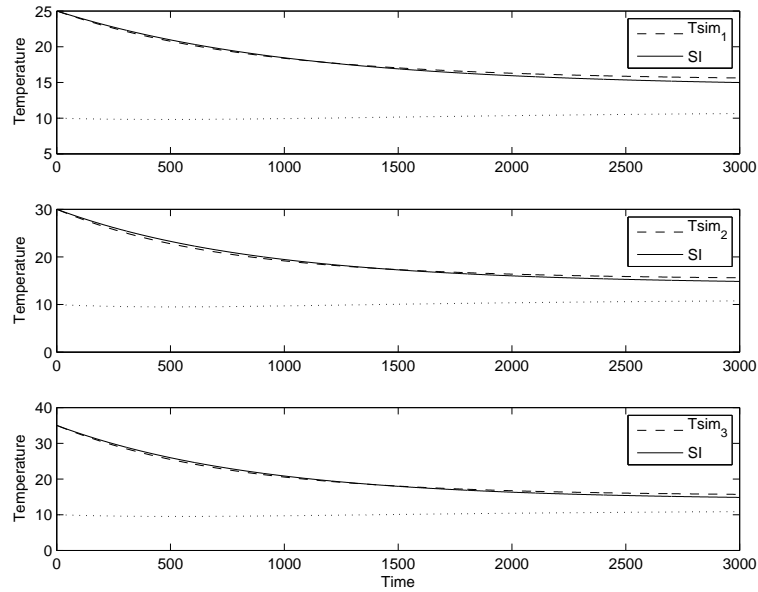
Figure 6.4: Simulation of temperature using θ_i. Dotted line illustrates the error plus a constant offset 10.

6.5.1 Further Work

Though we have shown that we can simulate the temperature using the parameter vector, we still need to quantify the quality of the result. We used (6.14) to construct the signal vector for the ith zone which included all control signals in the stable. We plan to investigate the model structure to find alternative to (6.16) and use Akaike's Final Prediction-error Criterion (Ljung, 1987) to compare different model structures.

With the presented method a model for temperature can be found, which should be expanded to include all the signals a controller would use. In livestock climate control it is common to control the temperature and humidity which easily can be included. For illustrative purposes we have decided to consider only temperature until the method has been tested.

With the assumption of having well defined flow directions, we have initially limited ourselves to having a future linear climate controller which only works with this specific flow direction. We therefore guide our future work to include all the different models based on one test signal. In general we have that for a N zone stable there are 3^{N-1} different flow combinations that need to be considered. In short our idea is to still use one estimation signal and then deduce the parameter vector for the total number of possible models.

Our idea of a future climate controller utilizing the proposed method is to automatically construct observers and controllers for each of the deduced models. Since we in a practical setting don't know the direction of the internal flow an observer supervisor should be constructed. This supervisor should then choose the model that currently has the best fit, and consequently choose the corresponding controller.

We have established a full scale laboratory for conducting research within climate control of livestock buildings, and are currently in the phase of adjusting minor details before the facility is fully operational. We plan to use this facility to generate real life data in order to validate our approach. We then hope to develop new climate controllers based on the presented work.

Paper 5: Parameter Estimation for Zone Based Climate Dynamics

J. J. Jessen* H. Schiøler*

Abstract

We present a technique for performing system identification in climatic model for a livestock building partitioned into zones. The dynamic model for livestock building climate takes into account possible leakages in the building envelope, and is both nonlinear and hybrid. The proposed technique is based on a novel idea of having well defined internal flow directions during data acquisition, which turns out to give a simple construction of the signal vector. Finally the obtained model parameters remain valid for arbitrary internal flow directions. We present the outcome of using the technique on a real life data set, and conclude with guidelines for further enhancements of the proposed method.

KEY WORDS
System Identification, Climate Control, Computer Modeling/Simulation, Embedded Systems.

7.1   Introduction

In order to design efficient climate controllers for livestock buildings models of sufficient precision are needed. Such models form the basis for simulation and controller

---

*Centre for Embedded Software Systems, Automation and Control, Aalborg University, Fredrik Bajers Vej 7C, Aalborg Ø, Denmark

synthesis, and are as such inevitable in the controller design process. Vendors offering controllers in large quantities (e.g. cruise control systems, thermostats etc.) can benefit from the fact that uniformity of the products (due to large quantity) makes it possible to implement identical controllers in all the specific products. For vendors offering controllers for larger plants e.g. livestock building climate control systems or power plants, it is often necessary to perform on site adjustments of the controllers, because the plants are different. This is especially true for climate control systems for livestock/agricultural buildings, which often differ in size and/or shape.

In this article we present an idea for performing automatic system identification in a zone based livestock building, which forms the basis for future climate controllers. Climate controllers based on the presented method would require less man power during setup. Though we omit the presentation of climate control in the present work, we consider the approach suitable for such future control systems.

That the airspace inside livestock buildings is imperfectly mixed is reported extensively in literature see e.g. (Barber and Ogilvie, 1982, 1984; Moor and Berckmans, 1996; Janssens et al., 2004). This has fostered the idea of multi zone climate control (see e.g. (Persis et al., 2006)), where a stable is divided into a number of zones and where each zone has its own ventilation equipment installed. The partitioning of a stable into zones, means that the zones can interact by having internal airflow sideways in the stable. As shown later this also means that the dynamic model for e.g. temperature consists of a number of coupled differential equations including sign functions depending on the direction of the internal air flow.

In principal there are two approaches for plant modeling: white box and black box. Where the former is based on physical knowledge the latter is based on input output analysis. The approach we take here is called grey box modeling (Ljung, 1987) where physical insight is used to choose the set of possible model structures from which the input output data is to be fitted. Where white and black box modeling spans a class of methods for performing system identification grey box modeling is somewhat in between these two methods. Furthermore one has to choose between performing system identification once and then develop the controller or design adaptive controllers for on site or real time system identification. For the former approach see (Moor and Berckmans, 1996), while (Cunha et al., 1997) discusses the latter approach for greenhouse climate control. For reasons that become clear later we initially rely on the first approach. This is due to the dynamic model being hybrid, so we are interested in choosing *one* of several possible models.

In the present work we limit ourselves to consider only the temperature. Initially we do this to keep things simple - expanding the technique to include e.g. humidity and CO_2 is straight forward but more involved. The paper is outlined as follows: In section 7.2 we introduce the dynamic model for temperature in a zone based livestock building and in section 7.3 we present the system identification technique. In section 7.4 we give an overview of the test facility and the results are presented in section 7.5. Conclusions and guidelines for further work are given in section 7.6.

82

7.2 System Description and Model

The ventilation components in the stable are air inlets and air outlets. We assume pressure dynamics to be orders of magnitude faster than the mixing dynamics for e.g. temperature, humidity and CO2, so for a given control signal stationary values for pressures and air flows may be assumed. We present simplified models for the ventilation equipment, focusing on parameter relations instead of exact physical models. We refer to (Brohus et al., 2002) for detailed models on e.g. air inlets and outlets.

7.2.1 Air Inlets

An air inlet is basically an opening in the wall with a guiding plate to adjust the direction and amount of incoming air. On the outside of the wall a protecting plate is mounted to minimize the effect of direct wind gusts on the inlet. We suggest the following approximate affine model for airflow from the outside through the wall inlet Q^{in} [m^3/s] in the ith zone:

$$Q_i^{\text{in}} = k_i \cdot (\alpha_i + l_i) \cdot (P^{\text{amb}} - P_i) \tag{7.1}$$

Where P_i [Pa] is the pressure in zone i, P^{amb} is the ambient pressure, k_i is a constant and α_i is the control signal to the inlet determining the opening degree of the guiding plate. We model possible leakages in the building envelope with l_i. We define the direction of airflow through an inlet from the outside to be positive.

7.2.2 Air Outlets

The air outlets are chimneys with an electrically controlled fan inside. We propose the following model which has a linear dominant term acting on the control signal (voltage applied to the fan) and a term subtracting flow due to pressure difference. The exhaust air flow Q^{fan} [m^3/s] from the ith zone is thus found as:

$$Q_i^{\text{fan}} = u_i^f \, c_i - d_i \, (P^{\text{amb}} - P_i) \tag{7.2}$$

where u_i^f is the control signal for the fan, c_i and d_i are constants. The flow is defined positive from the stable to the outside.

7.2.3 Inter Zone Net Air Flow

The net air flow $Q_{i,i+1}$ [m^3/s] between two zones (air flows are defined positive from a lower index to a higher index) is found by

$$Q_{i,i+1} = m_i \, (P_i - P_{i+1}) \tag{7.3}$$

where m_i is a constant.

7.2.4 Stationary Flows

A stationary flow balance for each zone i is found:

$$Q_{i-1,i} + Q_i^{\text{in}} = Q_{i,i+1} + Q_i^{\text{fan}} \tag{7.4}$$

where by definition $Q_{0,1} = Q_{N,N+1} = 0$.

7.2.5 Modeling Climate Dynamics

We assume that interdependence of climatic parameters between zones is dominated by internal air flow between the zones, and therefore omit to model radiation, convective heat transfer etc. By applying the flow balance the following model for temperature in the ith zone is easily obtained:

$$\dot{T}_i V_i = T^{\text{amb}} Q_i^{\text{in}} - T_i Q_i^{\text{fan}} + [Q_{i-1,i}]^+ T_{i-1} - [Q_{i-1,i}]^- T_i -$$
$$[Q_{i,i+1}]^+ T_i + [Q_{i,i+1}]^- T_{i+1} + \frac{W_i^t}{\rho_{\text{air}} c_{\text{air}}} \tag{7.5}$$

where V_i [m^3] is the zone volume, T^{amb} is ambient temperature, W^t [W] is the heat production from the animals, c_{air} [J/kg/$^\circ$C] is the air heat capacity and ρ_{air} [kg/m^3] is the air density. We remark that we deliberately omit to model heat transfer between the ambiance and the internal space of the building through the building structure. As we shall see later this is easily included in the parameter estimation if required. We define the use of square brackets in (7.5) as:

$$[x]^+ \triangleq \max(0, x), \ [x]^- \triangleq \min(0, x) \tag{7.6}$$

The flows to/from a zone and the use of square brackets is illustrated in figure 7.1.
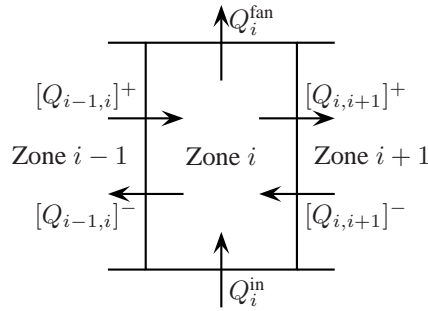
Figure 7.1: Illustration of flows for zone i.

We remark that the flows in (7.5) can not be set arbitrarily by the control signals $\{\alpha\}$ and $\{u\}$. The flows are given by simultaneous solving (7.1) through (7.4). We

84

furthermore note that the presented model may be considered both nonlinear (by the use of square brackets) and hybrid which arises naturally by the possibility of having different combinations of internal flow directions.

7.3 Multi Zone System Identification

For system identification purposes a mathematical model can be split in two: structure and parameter. Hence system identification can be seen as the process of choosing a model structure and consequently finding the parameters for the chosen structure. For the present work we have a model for temperature: namely coupled (by internal air flow) differential equations, hence the structure is given a priori and we are left with parameter estimation.

But, as previously noted the model for temperature (7.5) is both nonlinear and hybrid which makes the parameter estimation difficult. For an N zone livestock building there are $N-1$ zone boundaries and with three different zone boundary flow directions (none, left, right) there are a total of 3^{N-1} different models to choose between. In order to perform parameter estimation one has to know which model to apply the technique on. Now, for an arbitrary estimation signal the direction of the internal flow is unknown, hence it is not straight forward to choose the correct model. We therefore propose to use an estimation signal which ensures well defined internal flow directions, meaning that the corresponding $[\cdot]^+$ and $[\cdot]^-$ terms in (7.5) can be omitted. This means of course that the estimation signal should be designed carefully to ensure the desired flow directions. As explained in the introduction we initially rely on performing the system identification procedure once, due to the unknown flow directions for an arbitrary estimation signal.

7.3.1 Notation

Let $\alpha(k)$ denote the inlet signals in all zones at time k i.e. $\alpha(k) = [\alpha_1(k)\, \alpha_2(k)\, \ldots\, \alpha_N(k)]$ and likewise with $u^f(k)$. We define a signal vector $U(k)$ as:

$$U(k) = \begin{bmatrix} \alpha(k) & u^f(k) \end{bmatrix}^{\mathrm{T}} \tag{7.7}$$

and the signal matrix $\boldsymbol{U}$ as:

$$\boldsymbol{U} = [U(1)\, U(2)\, \ldots U(M)] \tag{7.8}$$

We then assume the following

Assumption 7.3.1 *There exists a signal matrix $\boldsymbol{U}$ s.t. $Q_{i,i+1} > 0$, $\forall i$*

7.3.2 Model Description

The temperature dynamics (7.5) uses air flow as a control signal which are found by simultaneously solving (7.1) through (7.4). We wish to have a model description in which the temperature dynamics is expressed directly given the actuator signals. Using assumption 7.3.1 each zone is only described by *one* model, and we therefore propose a system description for the ith zone as:

$$\dot{x}_i = \theta_i \varphi_i \tag{7.9}$$

where $\dot{x}_i = \frac{dT_i}{dt}$, θ_i is the parameter vector and φ_i is the signal vector. Now, if the bilinear parts from (7.5) are taken as input signals a standard linear regression model can be used to solve (7.9) (Montgomery, 2001).

7.3.3 Finding Dependent Control Signals

We wish to characterize the flow from the actuator signals $\{\alpha\}, \{u\}$ in order to construct the signal vector in (7.9). Using assumption 7.3.1 we rewrite (7.5) as:

$$\dot{T}_i V_i = T^{\text{amb}} Q_i^{\text{in}} - T_i Q_i^{\text{fan}} + Q_{i-1,i} T_{i-1} - Q_{i,i+1} T_i + \frac{W_i^t}{\rho_{\text{air}} c_{\text{air}}} \tag{7.10}$$

To find the vector of external flows $Q^{\text{E}} = [Q_1^{\text{in}} \ Q_1^{\text{fan}} \ \dots \ Q_i^{\text{in}} \ Q_i^{\text{fan}} \ \dots \ Q_N^{\text{in}} \ Q_N^{\text{fan}}]^{\text{T}}$ we set up the following:

$$Q^{\text{E}}(k) = H\alpha(k)^{\text{T}} + Ju^f(k)^{\text{T}} + R \tag{7.11}$$

Where H and J are matrices. We include the R vector in order to express leakages in the building envelope as modeled in (7.1) with $l_i > 0$. Knowing the external flows gives the internal flow from the flow balance (7.4). We thus have that the vector of internal flows $Q^{\text{I}} = [Q_{1,2} \ Q_{i-1,i} \ Q_{i,i+1} \ Q_{N-1,N}]^{\text{T}}$ is given by:

$$Q^{\text{I}}(k) = \phi Q^{\text{E}}(k) \tag{7.12}$$

where ϕ is a matrix. To find a specific internal flow we thus have:

$$
\begin{aligned}
Q_l^{\text{I}}(k) &= \sum_j \phi_{lj} Q_j^{\text{E}}(k) \\
&= \sum_j \sum_i \phi_{lj} H_{ji} \alpha_i(k) + \phi_{lj} J_{ji} u_i(k) + \sum_j \phi_{lj} R_j
\end{aligned}
\tag{7.13}
$$

Now from (7.13) we clearly have that internal and external flow depends on a linear combination of all the control signals. We use this in the forthcoming to construct the signal vector in (7.9).

7.3.4 Construction of Signal Vector

We define by $\mathbf{1}_{n\times m}$ a $n \times m$ matrix with 1 at all places, and let $\mathbf{T}_i$ denote the sequence of measured temperatures in the ith zone and likewise for $\mathbf{T}^{\text{amb}}$. We introduce an intermediate matrix $\mathcal{T}_i$ defined as:

$$\mathcal{T}_i = diag(\mathbf{T}_i) \tag{7.14}$$

where $diag$ is the diagonal operator constructing a matrix with the sequence $\mathbf{T}_i$ on the diagonal. Knowing that (7.13) is needed to find (7.10) and using assumption 7.3.1 we construct the signal matrix as:

$$\varphi_i = \begin{bmatrix} \boldsymbol{U}\mathcal{T}_{i-1} \\ \boldsymbol{U}(\mathcal{T}_i - \mathcal{T}^{\text{amb}}) \\ \mathbf{T}^{\text{amb}} \\ \mathbf{1}_{1\times M} \end{bmatrix} \tag{7.15}$$

A remark on (7.15) is in order. The term $\boldsymbol{U}\mathcal{T}_{i-1}^M$ denotes the combination of all control signals (length M) and measured temperatures from the neighboring zone to the left (index $i-1$). This is clearly needed from (7.10). $\mathbf{T}^{\text{amb}}$ is included to account for leakages and is as such independent of the control signals. $\mathbf{1}_{1\times M}$ is included to make a constant offset corresponding to the animals heat production. Though we have omitted the effect of heat transmission through the building envelope we see that, if needed, it could be included in the signal matrix by introducing a $\mathbf{1}_{1\times M}(\mathcal{T}_i - \mathcal{T}^{\text{amb}})$ term.

With the signal matrix constructed in (7.15) we are left with the construction of the result vector. From a sequence of measurements we approximate $\dot{x}_i$ numerically as:

$$\dot{x}_i(k) = \frac{T_i(k+1) - T_i(k)}{\Delta_t} \, , k = 1, 2, \ldots, M-1$$
$$\dot{x}_i(M) = \dot{x}_i(M-1) \tag{7.16}$$

where Δ_t is the sampling period. If $\boldsymbol{U}$ satisfies assumption 7.3.1, the parameter vector is given by:

$$\theta_i = \dot{x}_i \cdot \varphi_i^+ \tag{7.17}$$

where $^+$ denotes the pseudoinverse (Lay, 1997) giving a least square solution to (7.9).

7.3.5 Simulating Temperature Using SI

Knowing the parameter vector θ_i for each zone, we are interested in simulating the temperature evolution T_i^{SI} in each zone. We therefore define algorithm 4 which uses θ_i from (7.17), while we assume a numeric finite difference method in order to solve (7.9).

Algorithm 4 Simulate temperature using parameter vector.

$T_i^{\text{SI}}(1) \leftarrow T^{\text{init}}$
for $k = 1$ to $M - 1$ **do**
$\quad U(k) = [\alpha_1(k) \ \ldots \ \alpha_M(k) \ u_1(k) \ \ldots u_M(k)]^{\text{T}}$
$\quad \varphi_i(k) = [(T_i^{\text{SI}}(k) - T^{\text{amb}}(k))U(k)^{\text{T}}$
$\qquad\qquad (T_{i-1}^{\text{SI}}(k) - T^{\text{amb}}(k))U(k)^{\text{T}}$
$\qquad\qquad T^{\text{amb}}(k) \ \ 1]^{\text{T}}$
$\quad T_i^{\text{SI}}(k + 1) = T_i^{\text{SI}}(k) + \Delta_t \cdot \theta_i \cdot \varphi_i(k)$
end for

7.4   Test Setup

Aalborg University has in cooperation with Skov - a major Danish provider of climate control systems for livestock facilities - established a full scale laboratory for conducting research within climate control of livestock buildings. The laboratory provides not only a testbed for climate control algorithms but also a platform for testing remote monitoring of climatic conditions and sensor fusion for welfare sensors (Hansen, 2004). The climate laboratory is based on a decommissioned poultry stable (see figure 7.2), equipped with sensors, actuators, computer and a broadband Internet connection.

Figure 7.2: Decommissioned poultry stable acting as climate laboratory.

The test facility makes it possible to partition the stable into three separate zones. Besides air inlets and outlets a central heating system is installed, which is used to generate "disturbance" from the animals which would normally be experiences in a livestock building. The heating system consists of a central boiler and a pipe system to each zone, where a number of radiators are connected. A valve system connected to the central boiler makes it possible to distribute the hot water to the individual radiators in the zones. These valves can be set in an arbitrary position between 0 degrees (closed) and 90 degrees (fully open).

All actuator components (inlets, outlets, heating) can be set in an arbitrary position between closed and fully open, or stopped and maximum speed. The control of the components are implemented with IO cards taking as control signals values between 0 V (stopped/closed) and 10 V (maximum speed/fully open). Figure 7.3 illustrates the available components in the test facility. We remark that the test facility has five outlets (chimneys); one for each zone and one on each zone boundary. But initially we omit operating the two boundary outlets, using only the three outlets in the center of the zones.
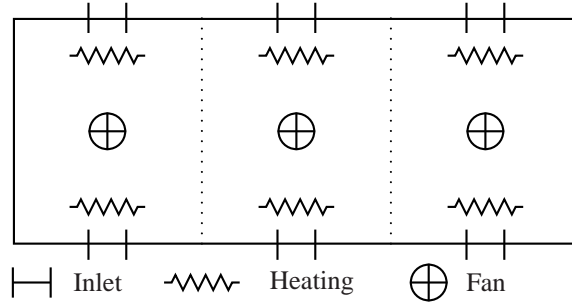
Figure 7.3: Illustration of climate components in zone partitioning.

The test facility is operated entirely over the Internet as described in (Jessen et al., 2006a). For synthesis of the executable performing the system identification procedure see (Jessen et al., 2006b).

7.5 Results

The test facility that we have at our disposal allows a three zone partitioning as described earlier. We have chosen an estimation signal that has been found to fulfill assumption 7.3.1. Figure 7.4 illustrates the signals for the three inlets and three fans.

Besides the actuation signals illustrated in figure 7.4, disturbances from animals is required - W^t in (7.5). We use the heating system to generate heat, with a fixed valve position of 8, 4 and 2 for zone 1,2 and 3 three respectively.

The experiment is conducted over a one hour period with $\Delta_t = 30$ s, giving $M = 120$ measurements. Initially we postpone the discussion of estimation signal excitation and rely on the signal illustrated in figure 7.4 for the illustrative purpose of our method.

For the first zone the signal matrix (7.15) reduces to:

$$\varphi_1 = \begin{bmatrix} \boldsymbol{U}(\mathcal{T}_1 - \mathcal{T}^{\mathrm{amb}}) \\ \mathbf{T}^{\mathrm{amb}} \\ \mathbf{1}_{1 \times M} \end{bmatrix} \tag{7.18}$$
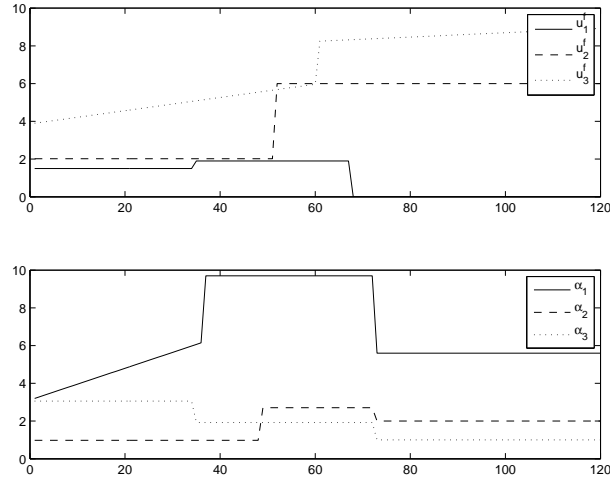
Figure 7.4: Actuator signals for estimation of parameter vector.

while φ_2 and φ_3 follows directly from (7.15)

The actuation signals illustrated in figure 7.4 are used as estimation signals, and the corresponding temperature in each zone is recorded. Using (7.16) the result vectors for the three zones are found, and applying (7.17) gives the following parameter vectors for the three zones:

$$\theta_1 = \begin{bmatrix} 0.002 & -0.002 & -0.004 & -0.002 \\ -0.007 & 0.005 & -0.177 & 3.132 \end{bmatrix}^T \tag{7.19}$$

$$\theta_2 = \begin{bmatrix} 0.003 & 0.013 & 0.021 & -0.017 & -0.008 \\ -0.001 & -0.018 & -0.071 & -0.109 & 0.099 \\ 0.043 & -0.001 & -0.243 & 4.430 \end{bmatrix}^T \tag{7.20}$$

$$\theta_3 = \begin{bmatrix} -0.004 & 0.152 & -0.004 & -0.012 & -0.059 \\ 0.008 & 0.020 & -1.064 & 0.027 & 0.098 \\ 0.407 & -0.066 & -0.327 & 6.182 \end{bmatrix}^T \tag{7.21}$$

The parameter vectors (7.19), (7.20) and (7.21) are used to simulate the temperature using the actuator signals illustrated in figure 7.4. Figure 7.5 illustrates the result of the simulation using algorithm 4 and the recorded temperatures during the experiment. The dotted lines shows the error between the recorded temperatures and the simulated temperatures.
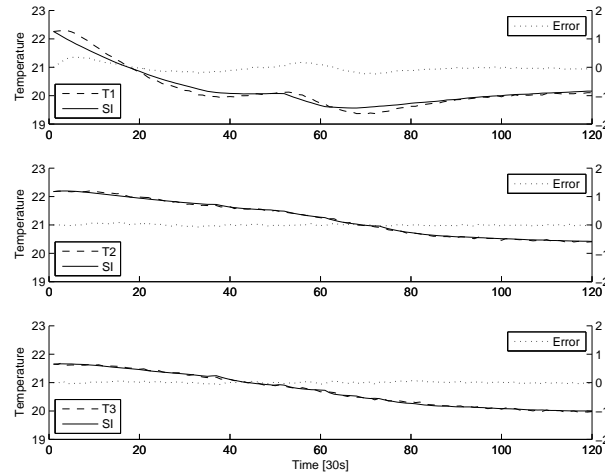
Figure 7.5: Measured temperature and simulated temperature using result from SI.

7.5.1 Quality of Fit

We postpone the discussion of quantifying the quality of our approach, and comment on the parameter fit instead. The found parameter vectors were simulated with the same actuator signal as were used to perform the initial experiment. It is therefore not surprising that the errors e_i illustrated in figure 7.5 are relatively small. The largest errors for the three simulated temperatures are found to $e_1 = 0.36$, $e_2 = 0.10$ and $e_3 = 0.09$. These results are not surprising since we have, in general, that $\lim_{M \to \infty} e_i = 0$.

Furthermore we postpone the discussion of persistent excitation of the estimation signal for correct identification of the parameters. We remark, that due to the nature of the system, we still have some practical problems to solve, in order to actually ensure the direction of the internal air flow during excitation.

7.6 Conclusion

By use of the assumption of having well defined internal flow directions we have shown a simple technique to perform system identification in a stable partitioned into multiple zones. Having well defined internal flow directions results in a simplified dynamic model for temperature and in turn a simple signal vector. We have demonstrated how to construct this signal vector and find the corresponding parameter vector. We have applied our approach to a real data set with satisfactory results leading us to believe our method is applicable for future implementation of climate controllers.

91

7.6.1 Future Research

In order to use the presented approach as a basis for a future climate controller design we need to guide our work in the following directions. First of all we plan to investigate the model structure to find alternative to (7.15) and use Akaike's Final Prediction-error Criterion (Ljung, 1987) to compare different model structures. This would also allow us to comment on the quality of our results and the idea as such.

In the presented approach we have chosen to perform the system identification once discarding the possibility of doing real time estimation. This is of course means that a climate controller need to ensure the internal flow directions as given by assumption 7.3.1. There are, however, reason for wanting real time or on line system identification - the obvious being the ambient environment constantly changing. The models used here are not exact e.g. the flow through an inlet depends on the square root of the pressure difference. The proposed system identification should therefore be executed periodically, in order to update the parameter vector.

Given the possible internal flow directions a different number of models to choose between exists. We plan to investigate the possibility of deducing the different model parameters from just one experiment fulfilling assumption 7.3.1. This should be possible since the found parameters remain valid even if the flow directions are interchanged. We assume to be able to report on this ongoing work soon.

Paper 6: Internal Flow Management in a Multi-Zone Climate Control Unit

C. De Persis*   J. J. Jessen†   R. Izadi-Zamanabadi †

H. Schiøler†

Abstract

In this contribution, we examine a dynamic model describing the evolution of internal climate conditions in a closed environment partitioned into zones for which different climate conditions must be guaranteed. The zones are not separated, large air masses are exchanged among them, and the behavior of each zone is strongly affected by those in the neighbor zones. We discuss a control strategy which, by acting on the heating and ventilation devices of the overall system, is able to achieve the control task while efficiently managing the internal flow. It is pointed out that the controller is hybrid and decentralized. An additional feature of the controller is that it takes on values in a finite set. The possible implementation in a networked environment is briefly discussed.

---

*Dipartimento di Informatica e Sistemistica "A. Ruberti", Università di Roma "La Sapienza", Via Eudossiana 18, 00184 Roma, Italy. E-mail: depersis@dis.uniroma1.it. Part of this project was developed while the author was visiting the Center for Embedded Software Systems, Aalborg University. The support of the Center is gratefully acknowledged.

†Department of Control Engineering, Center for Embedded Software Systems, Aalborg University, Fredrik Bajers Vej 7C, Aalborg Ø, Denmark

Nomenclature

T_i	Indoor air temperature of Zone i [°C]
T^{amb}	Temperature of the supplied air [°C]
x_i	Normalized indoor temperature of Zone i [°C]
$Q_{in,i}$	Airflow through inlet of zone i [m³/s]
$Q_{out,i}$	Airflow through outlet of zone i [m³/s]
$[Q_{ij}]^+$	Internal airflow from zone i to j [m³/s]
$[Q_{ij}]^-$	Internal airflow from zone j to i [m³/s]
u_T	Controlled heat production [J/s]
w_T	Indoor heat production (disturbance) [J/s]
V_i	Volume of zone i [m³]
c_{air}	Air Heat Capacity [J/kg/°C]
ρ_{air}	Air Density [kg/m³]

8.1   Introduction

We discuss the problem of guaranteeing prescribed indoor climate conditions in a building partitioned into communicating zones which exchange air flows. The prescribed climate conditions may differ very much from zone to zone. The ultimate goal is to act on the heating and ventilation devices in such a way that the climate requirement for each zone is reached even when large air masses are being exchanged and time-varying disturbances are present. These prescribed climate conditions typically mean that temperature and humidity should evolve within an interval of values (the "thermal region"). The focus of the paper will be on the temperature behavior only, but extensions to include the humidity dynamics are possible, although more involved. We refer the reader to (Taylor et al., 2004), (Pasgianos et al., 2003), (Arvanitis et al., 2006) and references therein for recent contributions on the problem of climate control, with special emphasis on agricultural and livestock buildings, which was the initial motivation for the present investigation.

Although we do not have space here to thoroughly compare our results with others, it is important to stress the main features of our contribution. We devise a controller which is suitable for implementation in a networked environment, in which sensors, controllers and actuators may be physically separated. Our controller is event-based, thus requiring sporadic measurements only. The actuators are required to provide control laws which take values in a finite and discrete set of values. Each controller governs the behavior of a single zone using information from neighbor zones, and cooperate with neighbor controllers to achieve different compatible control objectives and avoid conflicts. As a result of our approach, the overall controller turns out to be *hybrid* (van der Schaft and Schumacher, 2000), (Goebel et al., 2004) and *decentralized*.

We concentrate on actively cause such air masses exchange so as to make the heating and ventilation mechanism more efficient. This means that we aim to achieve an automatic mechanism to redirect warm air from hot zones (which need to be cooled down)

to cold zones (which need to be heated up), and to draw as much fresh air as possible to hot zones, relying on the ventilation capacity of neighboring "collaborative" zones. In order to achieve a controller capable of maintaining the climate conditions within the various thermal regions and at the same time capable of managing the internal flow, we introduce a set of coordinating logic variables (Heymann et al., 1998) which express the willingness of each zone to cooperate in the flow exchange, depending on the climate conditions of that zone and the neighboring ones. Then, the controller is designed to solve at each time a game theoretic problem (Heymann et al., 1998), (Lygeros et al., 1999) aimed to keep the state within the thermal zone despite the action of competitive players, namely thermal disturbances, given the constraints imposed by the neighboring zones and due to their willingness to cooperate or not in the air exchange. In additions, other constraints must be fulfilled at any time, namely flow balance for each zone.

In the next section, the dynamic model is introduced. The controller is chosen within the set of safety controllers, i.e. controllers which guarantee the temperature to lie within the thermal zone for all the times. The set of these controllers is described in Section 8.3, and then refined to characterize the safety controller with the additional capability to manage the internal flow. That the proposed controller guarantees the achievement of the control objective while fulfilling all the constraints (including the flow balance) is explicitly proven in Section 8.4. In Section 8.5, we illustrate the functioning of the controller for a three-zone climate control unit and conclusions are drawn in Section 8.6. Due to space limitations, most of the proofs are omitted.

8.2 System Description and Model

In this paper, we consider a cascade connection of N rectangular zones, as illustrated in Figure 8.1. This corresponds to the arrangement of zones in many real-life situations, such as livestock buildings.
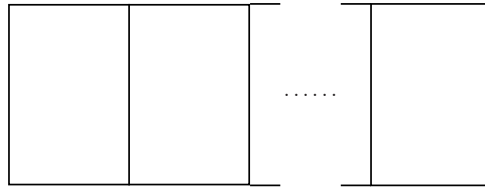
Figure 8.1: Cascade connection of N zones.

Each zone i, with $i \neq 1, N$, can exchange air with zones $i - 1$ and $i + 1$, while zone 1 and N can only exchange air with zone 2 and, respectively, $N - 1$. For each $i = 1, \ldots, N - 1$, we denote by $[Q_{i,i+1}]^{\pm}$ the amount of air flow exchanged between zone i and zone $i + 1$. More specifically, we have

$$[Q_{01}]^{\pm} = 0, \ [Q_{N,N+1}]^{\pm} = 0 \tag{8.1}$$

95

and, for each $i = 1, \ldots, N - 1$,

$$[Q_{i,i+1}]^+ = \begin{cases} |Q_{i,i+1}| & \text{if air flows from } i \text{ to } i+1 \\ 0 & \text{otherwise} \end{cases}$$

$$[Q_{i,i+1}]^- = \begin{cases} |Q_{i,i+1}| & \text{if air flows from } i+1 \text{ to } i \\ 0 & \text{otherwise} , \end{cases}$$

where the symbol $|Q_{i,i+1}|$ denotes a nonzero and positive constant value. We naturally assume that, it is not possible to have simultaneously air exchange from zone i to zone $i + 1$ and in the opposite direction. In other words, we assume that

$$[Q_{i,i+1}]^+ [Q_{i,i+1}]^- = 0 \tag{8.2}$$

for each $i = 1, \ldots, N$. Each zone is equipped with an inlet, an outlet, and a ventilation fan, which allow the zone to exchange air with the outside environment and with the neighboring zones. Indeed, by turning on the fan, air is forced out of each zone through the outlet. The amount of air outflow is denoted by the symbol $Q_{out,i}$. An amount $Q_{in,i}$ of inflow enters the zone through the inlet, and the following flow balance must hold: For each $i = 1, 2, \ldots, N$,

$$Q_{in,i} + [Q_{i-1,i}]^+ + [Q_{i,i+1}]^- = \\ Q_{out,i} + [Q_{i-1,i}]^- + [Q_{i,i+1}]^+ . \tag{8.3}$$

We explicitly remark that the amount of inflow depends on the outflow caused by the ventilation fan at the outlet. We now turn our attention to the equations describing the climate condition for each zone. Relevant quantities are the internal temperature $T_i \in \mathbb{R}$ and humidity $h_i \in \mathbb{R}_{\geq 0}$. For the sake of simplicity, in this paper we focus on temperature behavior only, which is therefore taken as state variable. In addition to the ventilation rates $Q_{out,i}$ provided by the fans, and the inflows $Q_{in,i}$ flowing through the inlets, another degree of control is given by the heating system, which provides a controlled amount u_i of heat. Moreover, we shall model the effect of internal disturbances which provide an additional amount of heat w_{Ti} power. Associated to the air masses which are flowing through the zones is an amount of power proportional to their temperature and the air heat capacity, which gives rise to changes in the temperature inside each zone. By balancing such power in each zone, the following equations are easily obtained (cf. e.g. (Janssens et al., 2004),(Arvanitis et al., 2006)) for $i = 1, 2, \ldots, N$:

$$\rho_{air} c_{air} V_i \tfrac{d}{dt} T_i = \rho_{air} c_{air} \left([Q_{i-1,i}]^+ T_{i-1} + Q_{in,i} \times \right. \\ \times T_{amb} - [Q_{i,i+1}]^+ T_i - Q_{out,i} T_i - [Q_{i-1,i}]^- T_i + \\ \left. [Q_{i,i+1}]^- T_{i+1} \right) + u_i + w_{Ti} . \tag{8.4}$$

Setting, by a slight abuse of notation,

$$u_i = u_i / (\rho_{air} c_{air}) , \quad w_{Ti} = w_{Ti} / (\rho_{air} c_{air}) ,$$

assuming that outside temperature T_{amb} is constant, and introducing the change of coordinates

$$x_i = T_i - T_{amb} , \quad i = 1, \ldots, N ,$$

we obtain, bearing in mind (8.3), and after easy calculations, the equations, for $i = 1, 2, \ldots, N$,

$$
\begin{aligned}
V_i \frac{d}{dt} x_i = & [Q_{i-1,i}]^+ x_{i-1} - [Q_{i,i+1}]^+ x_i - Q_{out,i} x_i - \\
& [Q_{i-1,i}]^- x_i + [Q_{i,i+1}]^- x_{i+1} + u_i + w_{Ti} .
\end{aligned}
\tag{8.5}
$$

In what follows, we shall refer to the x_i's simply as the temperature variables, although they differ from the actual temperature variables by a constant offset.

There are limitations on the control effort which can be delivered. In particular, the outflow $Q_{out,i}$ and the controlled heat must respectively fulfill

$$Q_{out,i} \in [0, Q_{out,i}^M] , \quad u_i \in [0, u_i^M] , \tag{8.6}$$

for some known constants $Q_{out,i}^M$ and u_i^M. The only way to regulate the amount of inflow is acting on the opening angle of a moving screen at the inlet, which can take only a finite number of positions. As a consequence, we assume that the inflow through the inlets can take only a finite number of values, i.e.

$$Q_{in,i} \in \Delta_i , \tag{8.7}$$

with Δ_i a finite set of nonnegative values which will become clear later (see (8.17) and the remark following it). We stack in the vector U all the control signals $[Q_{i-1,i}]^\pm$, $i = 2, \ldots, N$, $[Q_{i,i+1}]^\pm$, $i = 1, \ldots, N-1$, $Q_{in,i}$, $Q_{out,i}$, u_i, $i = 1, 2, \ldots, N$ and denote by $\mathcal{U}$ the set of admissible (piece-wise continuous) control signals which satisfy (8.1), (8.2), (8.3), (8.6), (8.7). Note that not all the components of the vector U are independent, as they are related through the constraints (8.2), (8.3). Additional constraints will be added by the introduction of the coordinating logic variables in the next subsection. Finally, we denote by $\mathbf{U}$ the set of values taken by the vector U, letting, for $i = 1, \ldots, N$, $Q_{out,i}$ and u_i range in the intervals given in (8.6), $Q_{in,i}$ take values in the set (8.7), and $[Q_{i-1,i}]^\pm$, $[Q_{i,i+1}]^\pm$ be such that (8.2), (8.3) are fulfilled.

The disturbance signals are not measured, but they are bounded $w_{Ti} \in [w_{Ti}^m, w_{Ti}^M]$, and the upper and lower bounds are assumed to be known. The vector $W = (w_{T1}, \ldots, w_{TN})^T$ of disturbance signals taking values on the above intervals is said to belong to the class $\mathcal{W}$ of admissible disturbances. The set $\mathbf{W} := [w_{T1}^m, w_{T1}^M] \times \ldots \times [w_{TN}^m, w_{TN}^M]$ denotes the range of values taken by the vector W.

8.2.1 Coordinating logic variables

To systematically resolve conflicts, which may arise when the control objectives of neighbor zones are contrasting, we introduce coordinating logic variables (Heymann

et al., 1998). Without loss of generality, we regard such variables as state variables which takes values in the binary set and whose derivatives are constantly equal to zero. Their values are reset from time to time by the hybrid controller to be specified below. For Zone 1, the logic variables are $[Q_{12}]_1^+$, $[Q_{12}]_1^-$, for Zone N, $[Q_{N-1,N}]_N^+$, $[Q_{N-1,N}]_N^-$, and for each zone $i \neq 1, N$, $[Q_{i-1,i}]_i^+$, $[Q_{i-1,i}]_i^-$, $[Q_{i,i+1}]_i^+$, $[Q_{i,i+1}]_i^-$. Each one of the logic variables takes values in the set $\{0, 1\}$. If Q is the vector in which all the logic variable are stacked, we have $Q \in \{0, 1\}^{2N}$. The logic variables $[Q_{i-1,i}]_i^\pm$ are set by zone i. Loosely speaking, if $[Q_{i-1,i}]_i^+ = 0$, this means that zone i does not want to accept air flow from zone i. On the contrary, if $[Q_{i-1,i}]_i^+ = 1$, the zone is willing to accept air flow from zone i. Note that $[Q_{i-1,i}]_i^+ = 1$ does not necessarily imply that flow will occur from zone $i - 1$ to i, i.e. $[Q_{i-1,i}]^+ \neq 0$, as this depends on whether or not zone $i - 1$ is willing to provide air to zone i. Similarly for the other logic variables. The rules followed to set the logic variables to a new value and when this should take place is discussed in the next section. Furthermore, for each zone, we introduce "cumulative" variables, which describe the amount of internal flow that the neighboring zones are willing to exchange in either one of the two directions. Such variables are recursively defined as follows:

$$
\begin{aligned}
Q_N^+ &= 0 \\
Q_i^+ &= (Q_{i+1}^+ + 1)[Q_{i,i+1}]_i^+ \cdot \\
&\quad [Q_{i,i+1}]_{i+1}^+ \; i = N - 1, \dots, 1 ,
\end{aligned}
\tag{8.8}
$$

and

$$
\begin{aligned}
Q_1^- &= 0 \\
Q_i^- &= (Q_{i-1}^- + 1)[Q_{i-1,i}]_{i-1}^- \cdot \\
&\quad [Q_{i-1,i}]_i^- \; i = 2, \dots, N .
\end{aligned}
\tag{8.9}
$$

8.3 Safety Controllers

In this section we characterize the set of all safety controllers, and the maximal controlled invariant set (Lygeros et al., 1999). Later, we shall single out in such a set the controller which additionally allows to manage the internal flows among the zones efficiently. By safety controller, we mean that controller which is able to maintain the state of the system within the so-called *thermal region*:

$$
\begin{aligned}
F &:= \{x_i : x_i \in [x_{im}, x_{iM}] , \quad i = 1, \dots, N\} \\
&= \Pi_{i=1}^N F_i := \Pi_{i=1}^N \{x_i : x_i \in [x_{im}, x_{iM}]\} ,
\end{aligned}
$$

where

$$
x_{im} = T_i^m - T^{amb} \geq 0 , \quad x_{iM} = T_i^M - T^{amb} > x_{im} ,
$$

for all the times, for any initial vector state, and under the action of any admissible disturbance $W \in \mathcal{W}$. The controller is designed following the indications of (Lygeros

et al., 1999). In the next subsection, we briefly recall the design procedure tackled there and refer the interested reader to the original source for more details. The procedure is applied to the design of the controller for each single zone. The interaction with the neighboring zones is tackled by a wise use of the coordination variables.

8.3.1 Design Procedure

The problem is that of designing a controller which guarantees the state x_i which describes the evolution of the temperature of zone i to belong to F_i, the projection on the x_i-axis of the thermal region F, for all the times. Following (Lygeros et al., 1999), the problem is addressed by formulating the two game problems:

$$
\begin{aligned}
J_i^{1*}(x,t) &= \max_{U(\cdot)\in\mathcal{U}} \min_{W(\cdot)\in\mathcal{W}} J_i^1(x, U(\cdot), W(\cdot), t) \ , \\
J_i^{2*}(x,t) &= \max_{U(\cdot)\in\mathcal{U}} \min_{W(\cdot)\in\mathcal{W}} J_i^2(x, U(\cdot), W(\cdot), t) \ ,
\end{aligned}
\tag{8.10}
$$

where the value functions

$$
\begin{aligned}
J_i^1(x, U(\cdot), W(\cdot), t) &= \ell_i^1(x(0)) := x_i(0) - x_{im} \ , \\
J_i^2(x, U(\cdot), W(\cdot), t) &= \ell_i^2(x(0)) := -x_i(0) + x_{iM} \ ,
\end{aligned}
$$

represent the cost of a trajectory $x(\cdot)$ which starts from x at time $t \leq 0$, evolves according to the equations (8.5) under the action of the control $U(\cdot)$ and the disturbance $W(\cdot)$. Clearly, $F_i = \{x \ : \ \ell_i^j(x) \geq 0 \text{ for } j = 1, 2\}$. In (Lygeros et al., 1999), the set of safe sets is defined as $\{x \ : \ J_i^{j*}(x) := \lim_{t\to-\infty} J_i^{j*}(x,t) \geq 0\}$, where the function $J_i^{j*}(x,t)$, $j = 1, 2$, is found by solving the Hamilton-Jacobi equation

$$
\begin{aligned}
-\frac{\partial J_i^{j*}(x,t)}{\partial t} &= \min\left\{0, H_i^{j*}\left(x, \frac{\partial J_i^{j*}(x,t)}{\partial x}\right)\right\} \\
J_i^{j*}(x,0) &= \ell_i^j(x) \ ,
\end{aligned}
\tag{8.11}
$$

$H_i^{j*}(x,p)$, the optimal Hamiltonian, is computed through the point-wise optimization problem

$$
H_i^{j*}(x,p) = \max_{U\in\mathbf{U}} \min_{W\in\mathbf{W}} H_i^{j*}(x,p,U,W) \ ,
\tag{8.12}
$$

and

$$
H_i^{j*}(x,p,U,W) = p^T f(x,U,W) \ .
$$

Notice that, by (8.11), at each time $J_i^{j*}(x,t)$ is non decreasing. Hence, if $J_i^{j*}(x) \geq 0$, then $J_i^{j*}(x,0) \geq 0$ as well, i.e. $\ell_i^j(x) \geq 0$. In other words, as expected, the set of safe states $\{x \ : \ J_i^{j*}(x) \geq 0\}$ is included in the set $\{x \ : \ \ell_i^j(x) \geq 0\}$.

8.3.2 Maximal Controlled Invariant Set

The maximal controlled invariant contained in F is the largest set of initial conditions for the state variables for which there exists a control action which maintains the state within F no matter what is the admissible disturbance acting on the system. In what follows, we show that in the present case, such a set coincides with F itself. This should not come as a surprise, the system being *bilinear* with the disturbance appearing linearly.

Lemma 8.3.1 *For any $i = 1, 2, \ldots, N$, if*

$$u_i^M \geq -w_{Ti}^m ,$$ (8.13)

we have

$$\{x \,:\, J_i^{1*}(x) \geq 0\} = \{x \,:\, \ell_i^1(x) \geq 0\} .$$ (8.14)

Lemma 8.3.2 *For any $i = 1, 2, \ldots, N$, if*

$$Q_{out,i}^M x_{iM} - w_{Ti}^M \geq 0 ,$$ (8.15)

then

$$\{x \,:\, J_i^{2*}(x) \geq 0\} = \{x \,:\, -x_i + x_{iM} \geq 0\} .$$ (8.16)

The two lemma above lead us to trivially conclude the following:

Proposition 8.3.1 *If for any $i = 1, 2, \ldots, N$ (8.13) and (8.15) hold, then the maximal controlled invariant set coincides with F.*

Proof: It is enough to notice that the entire set of safe sets, namely

$$\bigcap_{i=1}^{N} \bigcap_{j=1}^{2} \{x \,:\, J_i^{j*}(x) \geq 0\} ,$$

coincides with the set F, and hence any other controlled invariant set must be contained in the one given above. $\square$

Remark. Conditions (8.13) and (8.15) are very frequently encountered in practice and, loosely speaking, they are necessary for the control objective to be achieved. Notice that, as for condition (8.13), even for condition (8.15) a smaller safety set does not help to relax these requirements. Indeed, (8.13) is independent of the state, while (8.15) is such that if it holds for any x_i which is inside the thermal region, then it is *a fortiori* true for $x_i = x_{iM}$.

8.3.3   Safety Controller with Internal Flow Management

In this section we propose a safety controller which enjoys additional important features. First, for each zone, it takes into account the constraints imposed by the neighbor zones. In doing so, it is able to guarantee flow exchange among zones when all the zones are willing to carry out this action, while it avoids the raise of conflicts when the actions carried out by neighbor zones are not compatible with each other. We shall operate under the following:

Assumption 8.3.1 *At each time, each zone is either cooling down or heating up.*

Remark.   To have this fulfilled, it suffices to have conditions (8.13) and (8.15) fulfilled with strict inequalities.

Given that, for each $i = 1, \ldots, N$, the local controller has access at each time to the temperatures x_{i-1}, x_i, x_{i+1} (to x_i, x_{i+1} if $i = 1$, and to x_{i-1}, x_i if $i = N$), and to the coordinating variables, and that it also trivially knows whether the zone is in the "cooling down" or "heating up" mode, the values for the coordinating logic variables and controls are chosen so as to enforce the maximizing controller $U(\cdot)$ for the game $J_i^{1*}(x, t)$, if the zone is heating up, or for the game $J_i^{2*}(x, t)$, if the zone is cooling down, and taking into account the additional constraints imposed by the logic variables of the neighboring zones. Notice that we use the notation $Q_{in,i}^M$ to denote the value[1]

$$Q_{in,i}^M := Q_{out,i}^M + \sum_{j=1}^{Q_i^-} Q_{out,i-j}^M + \sum_{j=1}^{Q_i^+} Q_{out,j+i}^M \; . \tag{8.17}$$

Remark.   Depending on the values of Q_i^-, Q_i^+, which in turn depend on the values taken by the coordinating logic variables, the variable $Q_{in,i}^M$ can represent different values. All the possible values for $Q_{in,i}^M$ obtained from (8.17) define the set Δ_i introduced in Section 8.2.

We now introduce, for each Zone i, the controller which is able to handle the conflicting scenarios. To this purpose we need to explicitly take into account the conditions at the neighbor zones, namely temperatures and logic variables. As a result, for each Zone i, we precisely characterize the optimal controller which satisfies the game problems (8.10). Furthermore, by construction, whenever the neighbor zones agree on the actions to carry out (and this can be seen on the values taken by the coordinating logic variables), warm air is redirected from zones which are cooling down to zones which are heating up and are at lower temperatures. At the same time, the zones which are heating up collaborate with the neighbor zones which are cooling down to increase the amount of outflow. The controller is summarized in Table 8.1.

---

[1]In the sums below, if $Q_i^+ = 0$ ($Q_i^- = 0$), then $[Q_{i,i+1}]^+ = 0$ ($[Q_{i-1,i}]^- = 0$).

Mode		$Q_{out,i}$	$Q_{in,i}$	u_i	$[Q_{i-1,i}]^+_i$	$[Q_{i-1,i}]^-_i$	$[Q_{i,i+1}]^+_i$	$[Q_{i,i+1}]^-_i$
	Zone i Cooling Down							
1	Under any condition	$Q^M_{out,i}$	$Q^M_{in,i}$	0	0	1	1	0
	Zone i Heating Up							
2	$x_{i-1} \le x_i \wedge x_i \ge x_{i+1}$	0	0	u_{iM}	0	0	0	0
3	$x_{i-1} > x_i \wedge x_i \ge x_{i+1}$ $[Q_{i-1,i}]^+_{i-1} = 0$	0	0	u_{iM}	1	0	0	0
4	$x_{i-1} > x_i \wedge x_i \ge x_{i+1}$ $[Q_{i-1,i}]^+_{i-1} = 1$	$Q^M_{out,i}$	0	u_{iM}	1	0	1	0
5	$x_{i-1} \le x_i \wedge x_i < x_{i+1}$ $[Q_{i,i+1}]^-_{i+1} = 0$	0	0	u_{iM}	0	0	0	1
6	$x_{i-1} \le x_i \wedge x_i < x_{i+1}$ $[Q_{i,i+1}]^-_{i+1} = 1$	$Q^M_{out,i}$	0	u_{iM}	0	1	0	1
7	$x_{i-1} > x_i \wedge x_i < x_{i+1}$ $[Q_{i-1,i}]^+_{i-1} = 0 \wedge [Q_{i,i+1}]^-_{i+1} = 0$	0	0	u_{iM}	1	0	0	1
8	$x_{i-1} > x_i \wedge x_i < x_{i+1}$ $[Q_{i-1,i}]^+_{i-1} = 1 \wedge [Q_{i,i+1}]^-_{i+1} = 0$	$Q^M_{out,i}$	0	u_{iM}	1	0	1	0
9	$x_{i-1} > x_i \wedge x_i < x_{i+1}$ $[Q_{i-1,i}]^+_{i-1} = 0 \wedge [Q_{i,i+1}]^-_{i+1} = 1$	$Q^M_{out,i}$	0	u_{iM}	0	1	0	1
10	$x_{i-1} > x_i \wedge x_i < x_{i+1} \wedge x_{i-1} \ge x_{i+1}$ $[Q_{i-1,i}]^+_{i-1} = 1 \wedge [Q_{i,i+1}]^-_{i+1} = 1$	$Q^M_{out,i}$	0	u_{iM}	1	0	1	0
11	$x_{i-1} > x_i \wedge x_i < x_{i+1} \wedge x_{i-1} < x_{i+1}$ $[Q_{i-1,i}]^+_{i-1} = 1 \wedge [Q_{i,i+1}]^-_{i+1} = 1$	$Q^M_{out,i}$	0	u_{iM}	0	1	0	1

Table 8.1: Summary of the control law.

For the special cases $i = 1, N$, the controller simplifies, and it becomes easy to represent the behavior of the switched controller by a graph – see Figure 8.2 for $i = 1$.

Proposition 8.3.2 *Let Assumptions 8.3.1 and, for each $i = 1, 2, \ldots, N$, (8.13), (8.15) hold. Suppose additionally that, for each $i = 1, 2, \ldots, N - 1$ (respectively, $i = 2, \ldots, N - 2, N - 1$),*

$$[Q_{i,i+1}]^+ = \sum_{j=1}^{Q^+_i} Q^M_{out,j+i} \quad \left([Q_{i-1,i}]^- = \sum_{j=1}^{Q^-_i} Q^M_{out,i-j} \right) . \qquad (8.18)$$

Then, for each $i = 1, 2, \ldots, N$, the controller described in Table 8.1, renders F_i invariant and is the maximizing controller of the game problems (8.10).

Remark. As already mentioned, if both (8.13) and (8.15) hold with strict inequalities, then Assumption 8.3.1 can be removed from the statement. Moreover, we shall prove in the next section that the controller to be designed below guarantees (8.18) to be actually satisfied.

Before ending this section, we explicitly mention that the controller introduced above clearly renders F an invariant set. It is enough to verify that, on each edge at the boundary of F, the controller makes the velocity vector to point inwards F or to be
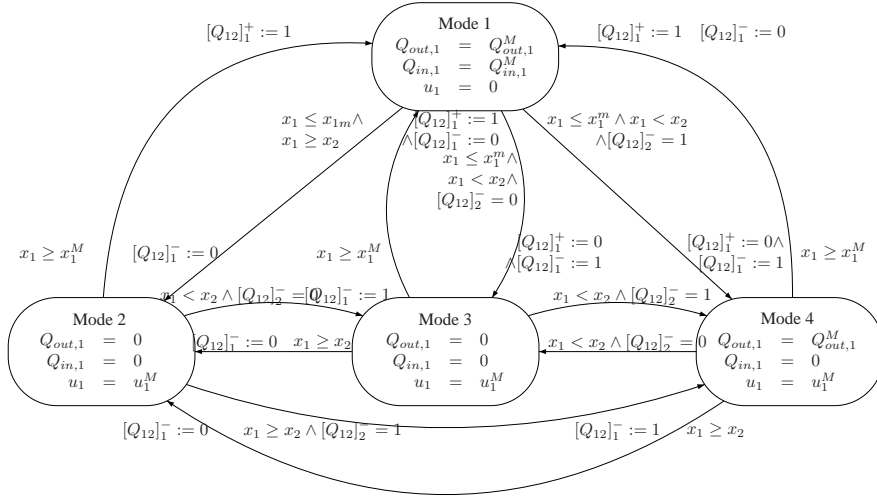
Figure 8.2: Hybrid controller for Zone 1. Following standard convention, when the edge has two labels the first one represents the guard, the second one (denoted by :=) the reset. When no reset is present, then all the variables remain unchanged upon the transition. All the guards at the edges linking the three states at the bottom should include the clause $\neg x_1 \geq x_{1M}$, which is omitted for the sake of simplicity. For each mode (discrete state) only three representative values are indicated. The remaining values can be derived from Table 8.1 specialized to the case $i = 1$.

tangent at the boundary of F. This is a straightforward exercise and is left as an exercise to the reader. In the next section, we show that the safety controller characterized here is actually a feasible controller, meaning that the flow balance (8.3) is fulfilled for each zone. As a consequence, it will be clear that (8.18) is actually guaranteed by our design of the controller.

8.4 Feasibility of the safety controller

The main obstacle to prove the feasibility of the safety controller investigated in the previous section comes from the fact that the dynamics of each zone are closely intertwined with those of the neighbor zones and that the number of zones are arbitrarily large. Nevertheless, we can exploit the topology of the system, namely the configuration according to which the zones are positioned, to approach the problem by an inductive argument.

In particular, we shall characterize conditions under which the flow balance is fulfilled for the first 2 zones. Then we shall proceed by showing the conditions under which, assuming that the flow balance is fulfilled up to Zone i, the flow balance is fulfilled even for Zone $i + 1$ and, finally, concluding the argument considering the zones $N - 1$ and N.

Lemma 8.4.1 *Consider the multi-zone climate control unit depicted in Fig. 8.1. The flow balance (8.3) is fulfilled for $i = 1, 2$, i.e. for Zone 1 and 2, provided that:*

- *If Zone 2 is in Mode 1, 4 or 8,*

$$[Q_{23}]^+ = \sum_{j=1}^{Q_2^+} Q_{out,\,j+2}^M \,, \tag{8.19}$$

  with $Q_2^+ = (Q_3^+ + 1)[Q_{23}]_3^+$, and $[Q_{23}]^- = 0$.

- *If Zone 2 is in Mode 2, 3, 5, 7 or 10,*

$$[Q_{23}]^+ = 0 \text{ and } [Q_{23}]^- = 0 \,. \tag{8.20}$$

- *If Zone 2 is in Mode 6, 9 or 11, $[Q_{23}]^+ = 0$ and*

$$[Q_{23}]^- = \sum_{j=1}^{Q_3^-} Q_{out,\,3-j}^M \,, \tag{8.21}$$

  with $Q_3^- = Q_2^- + 1$.

To make the statements below more concise we introduce the following definition:

Definition 8.4.1 *Let i be an integer such that $2 \leq i \leq N - 1$. Zones $1, 2, \ldots, i$ are said to* conditionally *satisfy the flow balance (8.3), if (8.3) is satisfied for $j = 1, 2, \ldots, i$ provided that (8.20), (8.19), (8.21) are satisfied with index 2 replaced by i and 3 by $i+1$.*

Then along the lines of the previous lemma, the following statement can be proven:

Lemma 8.4.2 *For some integer $2 \leq i \leq N - 2$, if Zones $1, 2, \ldots, i$ conditionally satisfy the flow balance (8.3), then also Zones $1, 2, \ldots, i, i + 1$ conditionally satisfy the flow balance (8.3).*

Zone N is different from the preceding zones in that air can be exchanged only through one side. Nevertheless, a similar statement holds:

Lemma 8.4.3 *Suppose Zones $1, 2, \ldots, N - 1$ conditionally satisfy the flow balance (8.3), then Zones $1, 2, \ldots, N - 1, N$ satisfy the flow balance (8.3).*

104

The previous statements allow us to conclude immediately with the following:

Proposition 8.4.1 *For each $i = 1, 2, \ldots, N$, let (8.13), (8.15) hold with strict inequalities. Then, the controller described in Table 8.1 renders F_i invariant and satisfies (8.18).*

8.5 Numerical results for a 3-zone climate control unit

In this section we present the outcome of a numerical simulation for the three zone case. In order to demonstrate the applicability of the presented controller we define the zones to be of different sizes, internal disturbances, ventilation rates, heating capacities and finally different thermal regions. The numerical values are set to resemble the values for a real livestock building.

We set the zone volumes to: $V_1 = 2000$, $V_2 = 1600$, $V_3 = 1800$ and the fan capacities: $Q_{out,1}^M = 1$, $Q_{out,2}^M = 1.5$, $Q_{out,3}^M = 0.8$ Keeping in mind the abuse of notation with u_i and $w_{T,i}$ the disturbances are piece wise constant in the set $\mathbf{W} := [2,3] \times [2,2] \times [5,7]$ resembling the heat production from pigs, in a stable corresponding to the given zone sizes. We refer to (Pedersen and Saalvik, 2002) for details on heat production from animals. We set the heating capacities to: $u_1 = 3$, $u_2 = 2$, $u_3 = 3$. The thermal zones are defined to: $x_{1m} = x_{3m} = 14$, $x_{1M} = x_{3M} = 16$, $x_{2m} = 12$, $x_{2M} = 14$.

The initial state is set in the thermal zone, with the following initial controller actions: The controller for zone 1 is heating up, while the controllers for zone 2 and zone 3 are cooling down. Figure 8.3 show the result of a simulation using the presented controllers pointing out that the controllers maintain the state within the thermal region. We omit graphs for the control signals, knowing that when a zone is heating up $u_i = u_i^M$ and when cooling down $Q_{i,out} = Q_{i,out}^M$.

A key feature of the presented controllers is the capability of using internal flow as a heating mechanism. Figure 8.4 shows the occurrence of internal flow between the zones. As we would expect, Figure 8.4 shows that internal flow only occurs from zone 1 and 3 to zone 2. The reason is that the thermal region for zone 2 is lower than the thermal region for zone 1 and zone 3: $x_{1m} \geq x_{2M} \leq x_{3m}$. This means that whenever Zone 2 is heating up and either one of Zones 1 or 3 is cooling down, internal flow takes place.
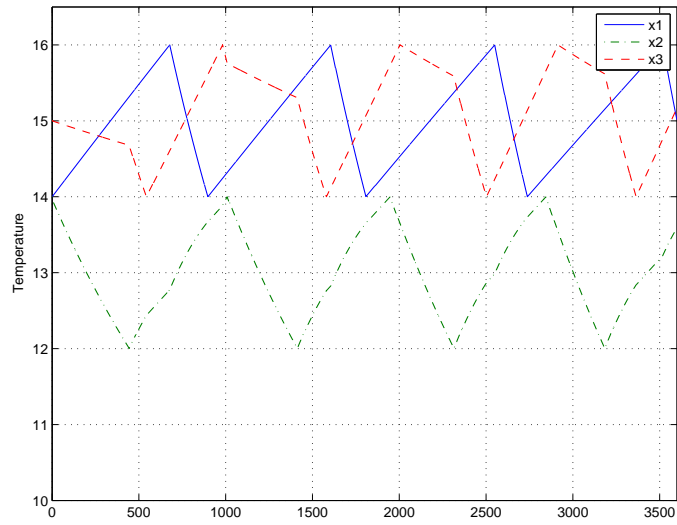
Figure 8.3: Trajectory of the system. Vertical axis represents absolute temperature.
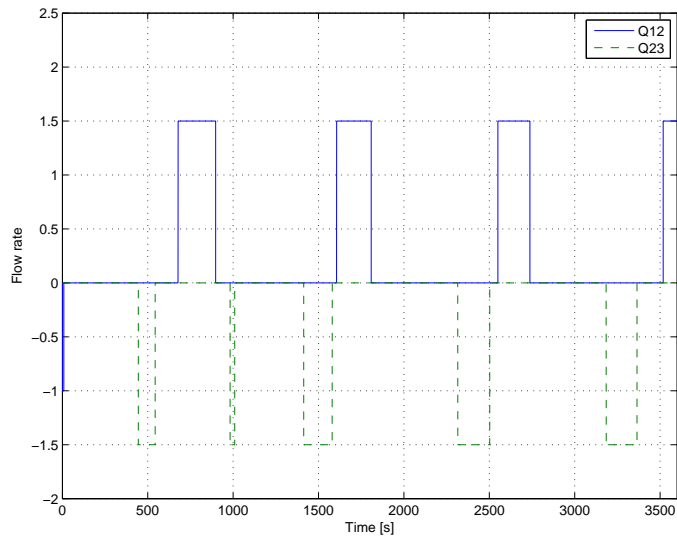
Figure 8.4: Time profile for the internal flows.

8.6   Conclusion

The paper has discussed a control strategy for a multi zone climate unit capable of maintaining the state within a safe set (*thermal region*), by management of internal flow between zones. The control law is inherently hybrid and decentralized in the sense that it only changes action when certain boundaries are met and/or when neighboring conditions change, and that the information requirements are limited to neighboring zones. Our motivation for considering the devised control strategy is the possible implementation in a resource constrained environment using wireless battery powered climatic sensors. Hence, we were after a solution to the problem which allowed to reach the control goal by transmitting feedback information only sporadically. We observe that the controller takes on values in a finite set, thus allowing for a potentially robust information transmission encoded using a finite number of bits. We have showed that the control law handles internal flow efficiently e.g. by using warm air from a neighbor zone to heat up, whenever certain conditions are met. An experimental facility to test our strategy has been constructed, and is currently being adjusted for minor details before we can move to real life experiments. We assume to be able to report on this ongoing work soon.

Paper 7: A Distributed Control Algorithm for Internal Flow Management in a Multi-Zone Climate Unit

C. De Persis[*]   J. J. Jessen[†]   R. Izadi-Zamanabadi [†]

H. Schiøler[†]

Abstract

We examine a distributed control problem for internal flow management in a multi-zone climate unit. The problem consists of guaranteeing prescribed indoor climate conditions in a cascade connection of an arbitrarily large number of communicating zones, in which air masses are exchanged to redirect warm air from hot zones (which need to be cooled down) to cold zones (which need to be heated up), and to draw as much fresh air as possible to hot zones, relying on the ventilation capacity of neighboring "collaborative" zones. The controller of each zone must be designed so as to achieve the prescribed climate condition, while fulfilling the constraints imposed by the neighboring zones - due to their willingness to cooperate or not in the air exchange - and the conservation of flow, and despite the action of unknown disturbances. We devise control laws which yield *hybrid* closed-loop systems, depend on local feedback information, take on values in a finite discrete set, and cooperate with neighbor controllers to achieve different compatible control objectives, while avoiding conflicts.

Keywords Climate control, distributed control, hybrid systems, nonlinear control, networked control systems.

Nomenclature

T_i	Indoor air temperature of Zone i [˚C]
T_{amb}	Temperature of the supplied air [˚C]
x_i	Normalized indoor air temperature of Zone i $T_i - T_{\text{amb}}$
$Q_{in,i}$	Airflow through inlet of zone i [m^3/s]
$Q_{out,i}$	Airflow through outlet of zone i [m^3/s]
Q_{ij}	Internal airflow from zone i to j [m^3/s]
u_T	Controlled heat production [J/s]
w_T	Indoor heat production (disturbance) [J/s]
V_i	Volume of zone i [m^3]
c_{air}	Air Heat Capacity [J/kg/˚C]
ρ_{air}	Air Density [kg/m^3]

9.1   Introduction

Distributed control systems have received considerable attention in the recent years due to progress in information and communication technology. Typically, distributed control systems comprises several subsystems for each one of which a local controller must guarantee the achievement of a control task in cooperation with neighboring subsystems. Sensors and actuators are usually not co-located with the system to control and the sensed or control information must be transmitted through a finite data-rate communication channel. The most common application is in the coordinated motion of mobile agents (see (Jadbabaie et al., 2003; Martinez et al., 2005; Gazi and Passino, 2005; Lin et al., 2005) just to cite a few), but other examples arise in power control problems in wireless communication (Wong and Sung, 1996), in automated highway systems (Swarop and Hedrick, 1996), etc.

In this paper, we discuss a distributed system which arises in a multi-zone climate control unit. The problem consists of guaranteeing prescribed indoor climate conditions in a building partitioned into communicating zones which exchange air flows. The prescribed climate conditions may differ very much from zone to zone. The ultimate goal is to act on the heating and ventilation devices in such a way that the climate requirement for each zone is reached even when large air masses are being exchanged and time-varying disturbances are present. We are interested in actively causing internal air masses exchange so as to make the heating and ventilation mechanism more efficient. Namely, we aim to achieve an automatic procedure to redirect warm air from hot zones (which need to be cooled down) to cold zones (which need to be heated up), and to draw as much fresh air as possible to hot zones, relying on the ventilation capacity of neighboring "collaborative" zones. See (Quijano et al., 2005) for other distributed problems in multi-zone temperature control. The prescribed climate conditions typically mean that temperature and humidity should evolve within an interval of values (the "thermal region"). The focus of the paper will be on the temperature behavior only, but extensions to include the humidity dynamics are possible, although more involved. We refer the

110

reader to (Taylor et al., 2004), (Pasgianos et al., 2003), (Arvanitis et al., 2006) and references therein for recent contributions on the problem of climate control, with special emphasis on agricultural and livestock buildings, which was the initial motivation for the present investigation. There is a large literature on climate control problems. The following additional references have some points of contact with the present contribution. In (Brecht et al., 2005) a second-order model is identified to describe the temperature of an imperfectly ventilated room, and then model predictive control is employed to achieve set-point regulation of the temperature. Temperature control for a multi-zone system is examined in (Zaheer-uddin et al., 1993), where a decentralized controller is proposed for a linearized model of the system. In (Caponetto et al., 2000), a fuzzy logic controller is proposed for a multi-layer model of greenhouse climate control unit. The fuzzy controller is tuned by genetic algorithms and its performance compared with bang-bang and PID controllers. The paper (Daskalov et al., 2004) develops a nonlinear adaptive controller for climate control in animal building to deal with nonlinear uncertainty. A hybrid control strategy for a heating/ventilation process modeled by a second-order linear system was employed in (Eker and Malmborg, 1999) to cope with different functioning regimes of the process. See (Balluchi et al., 1999) for another study on hybrid control synthesis for heating/ventilation systems. When compared with the papers above, our contribution appears to be the only one which proposes a model-based distributed event-driven control algorithm to achieve the prescribed climate requirements for a large-scale fully nonlinear model of the system.

We are interested in a solution which is suitable for implementation in a networked environment, in which sensors, controllers and actuators may be physically separated. As such, we devise control laws which are event-based, and may require sporadic measurements only. The actuators provide a finite and discrete set of values only. Each controller governs the behavior of a single zone using information from contiguous zones, and cooperate with neighbor controllers to achieve different compatible control objectives and avoid conflicts. As a result of our approach, the overall closed-loop system turns out to be *hybrid* (van der Schaft and Schumacher, 2000), (Goebel et al., 2004) and *distributed*. The advantages of having distributed controllers in our case lies in easy implementation, reduced computational burden and limited communication needs among the different components of the system. On the other hand, achievements of more global targets, such as fulfillment of an optimal criterion, appears harder and requires further investigation. In recent years, many contributors have focused on networked control systems (see (Hespanha et al., 2006) for a recent survey). Some of them have focused on the data rate constraint imposed by the finite bandwidth of the communication channel, for both linear (see e.g. (Nair and Evans, 2003; Tatikonda and Mitter, 2004; Ishii and Francis, 2002; Fagnani and Zampieri, 2003; Elia and Mitter, 2001; Brockett and Liberzon, 2000)) and nonlinear systems (Liberzon and Hespanha, 2005; Persis and Isidori, 2004; Nair et al., 2004a; Persis, 2005), others on actuators and sensors scheduling for time-based control of networked control systems (see e.g. (Nešić and Teel, 2004)), and a few have taken into account both data-rate constraints and decentralized nature of the
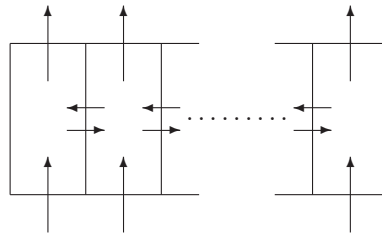
Figure 9.1: Cascade connection of N zones.

problem (see e.g. (Nair et al., 2004b), and (Johansson et al., 2005), for a multi-vehicle rendezvous control problem under the effect of quantization).

In order to achieve a controller capable of maintaining the climate conditions within the various thermal regions and at the same time capable of managing the internal flow, we introduce a set of coordinating logic variables (Heymann et al., 1998) which express the willingness of each zone to cooperate in the flow exchange, depending on the climate conditions of that zone and the neighboring ones. Then, the controller is designed to solve at each time a game theoretic problem (Heymann et al., 1998), (Lygeros et al., 1999) aimed to keep the state within the thermal zone despite the action of competitive players, namely thermal disturbances, given the constraints imposed by the neighboring zones, which are due to their willingness to cooperate or not in the air exchange. In addition, other constraints must be fulfilled at any time, namely conservation of flow for each zone. Besides proposing a solution to a novel distributed control problem, in which coordination is achieved while fulfilling algebraic constraints, in the paper the topology of the system is exploited to cope with the large dimension of the problem.

In the next section, the dynamic model is introduced. The design of the controllers is described in Section 9.3. It is explicitly proven in Section 9.4 that the proposed controllers guarantee the achievement of the control objective while fulfilling all the constraints (including the flow balance). In Section 9.5, we illustrate the functioning of the controller for a three-zone climate control unit and conclusions are drawn in Section 9.6.

9.2 System Description and Model

In this paper, we consider a cascade connection of N rectangular section zones, as illustrated in Figure 9.1. This corresponds to the arrangement of zones in many real-life situations, such as livestock buildings. However, it appears that the method will work with different arrangements (e.g. those founds in cars), provided that the direction and magnitude of the flows to be exchanged among the zones can be set by the actuators. Each zone i, with $i \neq 1, N$, can exchange air with zones $i-1$ and $i+1$, while zone 1 and N can only exchange air with zone 2 and, respectively, $N-1$. For each $i = 1, \ldots, N-1$, we denote by $Q_{i,i+1}$ the amount of air flow exchanged between zone i and zone $i + 1$.

More specifically, we have

$$Q_{01} = Q_{10} = 0 \, , \quad Q_{N,N+1} = Q_{N+1,N} = 0 \, , \tag{9.1}$$

and, for each $i = 1, \ldots, N - 1$,

$$Q_{i,i+1} \begin{cases} > 0 & \text{if air flows from } i \text{ to } i + 1 \\ = 0 & \text{otherwise} \end{cases}$$

$$Q_{i+1,i} \begin{cases} > 0 & \text{if air flows from } i + 1 \text{ to } i \\ = 0 & \text{otherwise .} \end{cases}$$

Implicitly, we are assuming that, it is not possible to have simultaneously air exchange from zone i to zone $i + 1$ and in the opposite direction. In other words, we assume that

$$Q_{i,i+1} \cdot Q_{i+1,i} = 0 \tag{9.2}$$

for each $i = 1, \ldots, N$. Each zone is equipped with an inlet, an outlet, and a ventilation fan, which allow the zone to exchange air with the outside environment and with the neighboring zones. Indeed, by turning on the fan, air is forced out of each zone through the outlet. The amount of air outflow is denoted by the symbol $Q_{out,i}$. An amount $Q_{in,i}$ of inflow enters the zone through the inlet, and the following flow balance must hold: For each $i = 1, 2, \ldots, N$,

$$Q_{in,i} + Q_{i-1,i} + Q_{i+1,i} = Q_{out,i} + Q_{i,i-1} + Q_{i,i+1} \, . \tag{9.3}$$

We explicitly remark that the amount of inflow depends on the outflow caused by the ventilation fan at the outlet. We now turn our attention to the equations describing the climate condition for each zone. Relevant quantities are the internal temperature $T_i \in \mathbb{R}$ and humidity $h_i \in \mathbb{R}_{\geq 0}$. For the sake of simplicity, in this paper we focus on temperature behavior only, which is therefore taken as state variable. In addition to the ventilation rates $Q_{out,i}$ provided by the fans, and the inflows $Q_{in,i}$ flowing through the inlets, another degree of control is given by the heating system, which provides a controlled amount u_i of heat. Moreover, we shall model the effect of internal disturbances which provide an additional amount of heat w_{Ti} power. Associated to the air masses which are flowing through the zones is an amount of power proportional to their temperature and the air heat capacity, which gives rise to changes in the temperature inside each zone. By balancing thermal power in each zone, the following equations are easily obtained (cf. e.g. (Janssens et al., 2004), (Arvanitis et al., 2006)) for $i = 1, 2, \ldots, N$:

$$\begin{aligned} \rho_{air} c_{air} V_i \frac{d}{dt} T_i &= \rho_{air} c_{air} \left(Q_{i-1,i} T_{i-1} + Q_{in,i} T_{amb} - Q_{i,i+1} T_i \right. \\ &\quad \left. - Q_{out,i} T_i - Q_{i,i-1} T_i + Q_{i+1,i} T_{i+1} \right) + u_i + w_{Ti} \, . \end{aligned} \tag{9.4}$$

Setting, by a slight abuse of notation,

$$u_i := u_i / (\rho_{air} c_{air}) \, , \quad w_{Ti} := w_{Ti} / (\rho_{air} c_{air}) \, ,$$

113

assuming that outside temperature T_{amb} is constant and $T_i > T_{amb}$[1], and introducing the change of coordinates

$$x_i = T_i - T_{amb} , \quad i = 1, \ldots, N ,$$

we obtain, bearing in mind (9.3), and after easy calculations, the equations, for $i = 1, 2, \ldots, N$,

$$V_i \frac{d}{dt} x_i = Q_{i-1,i} x_{i-1} - Q_{i,i+1} x_i - Q_{out,i} x_i - Q_{i,i-1} x_i + Q_{i+1,i} x_{i+1} + u_i + w_{Ti} . \quad (9.5)$$

In what follows, we shall refer to the x_i's simply as the temperature variables, although they differ from the actual temperature variables by a constant offset.

There are limitations on the control effort which can be delivered. In particular, the outflow $Q_{out,i}$ and the controlled heat must respectively fulfill

$$Q_{out,i} \in [0, Q_{out,i}^M] , \quad u_i \in [0, u_i^M] , \quad (9.6)$$

for some known constants $Q_{out,i}^M$ and u_i^M. The only way to regulate the amount of inflow is acting on the opening angle of a moving screen at the inlet, which can take only a finite number of positions. As a consequence, we assume that the inflow through the inlets can take only a finite number of values, i.e.

$$Q_{in,i} \in \Delta_i , \quad (9.7)$$

with Δ_i a finite set of nonnegative values which will become clear later (see (9.14) and the remark following it). We stack in a vector U all the control signals $Q_{i,i-1}$, $i = 2, \ldots, N$, $Q_{i,i+1}$, $i = 1, \ldots, N-1$, $Q_{in,i}$, $Q_{out,i}$, u_i, $i = 1, 2, \ldots, N$ and denote by $\mathcal{U}$ the set of admissible (piece-wise continuous) control signals which satisfy (9.1), (9.2), (9.3), (9.6), (9.7). Note that not all the components of the vector U are independent, as they are related through the constraints (9.2), (9.3). Additional constraints will be added by the introduction of the coordinating logic variables in the next subsection. Finally, we denote by $\mathbf{U}$ the set of values taken by the vector U, letting, for $i = 1, \ldots, N$, $Q_{out,i}$ and u_i range in the intervals given in (9.6), $Q_{in,i}$ take values in the set (9.7), and $Q_{i-1,i}, Q_{i,i-1}, Q_{i,i+1}, Q_{i+1,i}$ be such that (9.2), (9.3) are fulfilled.

The disturbance signals are not measured, but they are bounded

$$w_{Ti} \in [w_{Ti}^m, w_{Ti}^M] , \quad (9.8)$$

with $w_{Ti}^m, w_{Ti}^M \in \mathbb{R}$ assumed to be known. The vector

$$W = (w_{T1}, \ldots, w_{TN})^T$$

---

[1]Assuming T_{amb} constant results in no loss of generality, provided that T_i remains above T_{amb}. As a matter of fact, the effect of a time-varying ambient temperature can be easily incorporated in the disturbance signal w_{Ti}. Moreover, the case in which $T_i \leq T_{amb}$ can be easily tackled by the method proposed in this paper, provided that a cooling device – such as a sprinkling system in livestock buildings – is included in the model.

of disturbance signals taking values on the above intervals is said to belong to the class $\mathcal{W}$ of admissible disturbances. The set

$$\mathbf{W} := [w_{T1}^m, w_{T1}^M] \times \ldots \times [w_{TN}^m, w_{TN}^M]$$

denotes the range of values taken by the vector W.

9.2.1 Coordinating logic variables

Having in mind a cooperative behavior among zones, it is clear that decisions regarding each zone must take into account the behavior of neighboring zones. Furthermore, aiming at a decentralized controller, we would like to implement a controller for each single zone whose strategy is decided on the basis of local information concerning the zone itself and the neighboring zones only. This implies that, in some cases, the objectives for two or more zones can be contrasting and coordination is needed to achieve the overall control strategy. For illustrative purposes, one of these conflicting scenarios is reported below. *Example.* Consider a 4-zone system with the following scenario: Zone

1 and 3 are cooling down, Zone 2 and 4 are heating up, and the temperatures in the 4 zones satisfy $x_1 < x_2 < x_3 > x_4$. As Zone 1 is trying to cool down, it is interested in attracting fresh air from outside. The amount of inflow can be increased if, in addition to the outflow provided by the fan, internal flow from Zone 1 to Zone 2 takes place. Hence, the controller in Zone 1 would be motivated to increase the opening of the inlet 1 so as to let in an amount of air greater than $Q_{out,1}^M$. On the other hand, Zone 2 is warming up and the temperature of Zone 3 is higher than the one in Zone 2, whereas the temperature in Zone 1 is lower than in Zone 2. Moreover, Zone 3 is cooling down and therefore is interested to let out as much air as possible to the neighboring zones. This implies that Zone 2 will turn on its fan to attract air from Zone 3 (but not from Zone 1). Also notice that Zone 4 is heating up and interested in getting air from Zone 3, since $x_3 > x_4$. To avoid the fan in Zone 2 to attract air from Zone 1, the former must signal the latter that the inlet opening at Zone 1 should not allow for an inflow greater than $Q_{out,1}^M$, which is clearly in contrast with what Zone 1 is willing to do. At the same time, Zone 2 and 4 must signal Zone 3 they are interested in getting its air, and Zone 3 should acknowledge its willingness to release such air, and correspondingly accommodate its inlet opening.

To systematically resolve conflicts like the one just described, we introduce coordinating logic variables (cf. (Heymann et al., 1998)). Without loss of generality, we regard such variables as state variables which take values in the binary set and whose derivatives are constantly equal to zero. Their values are reset from time to time by the event-based controller to be specified below. For Zone 1, the logic variables are

$$\sigma_{12}^{(1)}, \quad \sigma_{21}^{(1)},$$

for Zone N,

$$\sigma_{N-1,N}^{(N)}, \quad \sigma_{N,N-1}^{(N)},$$

115

and for each zone $i \neq 1, N$,

$$\sigma_{i-1,i}^{(i)}, \quad \sigma_{i,i-1}^{(i)}, \quad \sigma_{i,i+1}^{(i)}, \quad \sigma_{i+1,i}^{(i)}.$$

Each one of the logic variables takes values in the set $\{0, 1\}$. If σ is the vector in which all the logic variable are stacked, we have $\sigma \in \{0, 1\}^{4(N-1)}$. The logic variables $\sigma_{i-1,i}^{(i)}$, $\sigma_{i,i-1}^{(i)}, \sigma_{i,i+1}^{(i)}, \sigma_{i+1,i}^{(i)}$ are set by zone i. Loosely speaking, if $\sigma_{i-1,i}^{(i)} = 0$, this means that zone i does not want to accept air flow from zone $i - 1$. On the contrary, if $\sigma_{i-1,i}^{(i)} = 1$, the zone is willing to accept air flow from Zone $i - 1$. Note that $\sigma_{i-1,i}^{(i)} = 1$ does not necessarily imply that flow will occur from zone $i - 1$ to i, i.e. not necessarily $Q_{i-1,i} \neq 0$, as this depends on whether or not zone $i-1$ is willing to provide air to zone i. Similarly for the other logic variables. The rules followed to set the logic variables to a new value and when this should take place is discussed in the next section. Furthermore, for each zone, we introduce "cumulative" variables, which are related to the amount of internal flow that the neighboring zones are willing to exchange in either one of the two directions. Such variables are recursively defined as follows:

$$\begin{aligned}
\gamma_N^+ &= 0 \\
\gamma_i^+ &= (\gamma_{i+1}^+ + 1)\sigma_{i,i+1}^{(i)} \cdot \sigma_{i,i+1}^{(i+1)}, \quad i = N - 1, \ldots, 1,
\end{aligned} \tag{9.9}$$

and

$$\begin{aligned}
\gamma_1^- &= 0 \\
\gamma_i^- &= (\gamma_{i-1}^- + 1)\sigma_{i-1,i}^{(i-1)} \cdot \sigma_{i-1,i}^{(i)}, \quad i = 2, \ldots, N.
\end{aligned} \tag{9.10}$$

9.3 Design of the Controllers

In this section we introduce the controllers. For the sake of simplicity, first we derive here them under the assumption that the constraints on the flow balance are fulfilled. Then, in the next section, we show that this assumption is actually verified by the controllers. The controllers we are interested in are able to maintain the state of the system within the so-called *thermal region*:

$$F = \Pi_{i=1}^N F_i,$$

where $F_i = \{x_i : x_i \in [x_i^m, x_i^M]\}$, $x_i^m = T_i^m - T_{amb} \geq 0$, $x_i^M = T_i^M - T_{amb} > x_i^m$, for all the times, for any initial vector state, and under the action of any admissible disturbance $W \in \mathcal{W}$. This kind of controllers are referred to as safety controllers in (Lygeros et al., 1999) and our design follows the indications given therein. The controllers here enjoy important features. First, they take into account the constraints imposed by the neighbor zones. In doing so, they are able to guarantee flow exchange among zones when all the zones are willing to carry out this action, while they avoid the raise of conflicts when the actions carried out by neighbor zones are not compatible with each other.

116

9.3.1 Design Procedure

The problem is that of designing a controller which guarantees the state x_i which describes the evolution of the temperature of zone i to belong to F_i, the projection on the x_i-axis of the thermal region F, for all the times. Following (Lygeros et al., 1999), the problem is addressed by formulating the two game problems:

$$
\begin{aligned}
J_i^{1*}(x,t) &= \max_{U(\cdot)\in\mathcal{U}} \min_{W(\cdot)\in\mathcal{W}} J_i^1(x,U(\cdot),W(\cdot),t) \;, \\
J_i^{2*}(x,t) &= \max_{U(\cdot)\in\mathcal{U}} \min_{W(\cdot)\in\mathcal{W}} J_i^2(x,U(\cdot),W(\cdot),t) \;,
\end{aligned}
\tag{9.11}
$$

where the value functions

$$
\begin{aligned}
J_i^1(x,U(\cdot),W(\cdot),t) &= \ell_i^1(x(0)) := x_i(0) - x_i^m \;, \\
J_i^2(x,U(\cdot),W(\cdot),t) &= \ell_i^2(x(0)) := -x_i(0) + x_i^M \;,
\end{aligned}
$$

represent the cost of a trajectory $x(\cdot)$ which starts from x at time $t \leq 0$, evolves according to the equations (9.5) under the action of the control $U(\cdot)$ and the disturbance $W(\cdot)$. Clearly, $F_i = \{x \; : \; \ell_i^j(x) \geq 0 \text{ for } j = 1, 2\}$. In (Lygeros et al., 1999), the set of safe sets is defined as $\{x \; : \; J_i^{j*}(x) := \lim_{t\to-\infty} J_i^{j*}(x,t) \geq 0\}$, where the function $J_i^{j*}(x,t)$, $j = 1, 2$, is found by solving the Hamilton-Jacobi equation

$$
\begin{aligned}
-\frac{\partial J_i^{j*}(x,t)}{\partial t} &= \min\left\{0, H_i^{j*}\left(x, \frac{\partial J_i^{j*}(x,t)}{\partial x}\right)\right\} \\
J_i^{j*}(x,0) &= \ell_i^j(x) \;,
\end{aligned}
\tag{9.12}
$$

$H_i^{j*}(x,p)$, the optimal Hamiltonian, is computed through the point-wise optimization problem

$$
H_i^{j*}(x,p) = \max_{U\in\mathbf{U}} \min_{W\in\mathbf{W}} H_i^{j*}(x,p,U,W) \;,
\tag{9.13}
$$

and

$$
H_i^{j*}(x,p,U,W) = p^T f(x,U,W) \;.
$$

Notice that, by (9.12), at each time $J_i^{j*}(x,t)$ is non decreasing. Hence, if $J_i^{j*}(x) \geq 0$, then $J_i^{j*}(x,0) \geq 0$ as well, i.e. $\ell_i^j(x) \geq 0$. In other words, as expected, the set of safe states $\{x \; : \; J_i^{j*}(x) \geq 0\}$ is included in the set $\{x \; : \; \ell_i^j(x) \geq 0\}$.

9.3.2 Controllers

To the purpose of designing the controllers, it is convenient to characterize the maximal controlled invariant set contained in F (Lygeros et al., 1999), i.e. the largest set of initial conditions for the state variables for which there exist control actions which maintain the state within F no matter what the admissible disturbance acting on the system is. The system being bilinear, it is not difficult to show that the maximal controlled invariant set coincides with F (see Proposition 9.3.1).

117

The controllers we design below at any time guarantee the controlled temperature in each zone to be either increasing or decreasing. Given that, for each $i = 1, \ldots, N$, the local controller has access at any time to the temperatures x_{i-1}, x_i, x_{i+1} (to x_i, x_{i+1} if $i = 1$, and to x_{i-1}, x_i if $i = N$), and to the coordinating variables, and that it also knows whether the zone is in the "cooling down" or "heating up" mode, the values for the coordinating logic variables and controls are chosen so as to enforce the maximizing controller $U(\cdot)$ for the game $J_i^{1*}(x, t)$, if the zone is heating up, or for the game $J_i^{2*}(x, t)$, if the zone is cooling down, and to take into account the additional constraints imposed by the logic variables of the neighboring zones. In the calculations which lead to the design of the controller, we adopt the notation $Q_{in,i}^M$ to denote the value [2]

$$Q_{in,i}^M := Q_{out,i}^M + \sum_{j=1}^{\gamma_i^-} Q_{out,i-j}^M + \sum_{j=1}^{\gamma_i^+} Q_{out,j+i}^M . \tag{9.14}$$

Remark. Depending on the values of γ_i^-, γ_i^+, which in turn depend on the values taken

by the coordinating logic variables, the variable $Q_{in,i}^M$ can represent different values. All the possible values for $Q_{in,i}^M$ obtained from (9.14) plus the zero value define the set Δ_i introduced in section 9.2. Note that the $Q_{in,i}^M$'s depend on the values of the maximal fan capacity of all the zones. If such a knowledge is not available, it is possible to avoid it by proper redefinition of the γ_i^-'s and γ_i^+'s. We do not pursue it here, as it would require a more cumbersome notation.

We now introduce, for each Zone i, the controller which is able to handle the conflicting scenarios. To this purpose we need to explicitly take into account the conditions at the neighbor zones, namely temperatures and logic variables. As a result, for each Zone i, we precisely characterize the optimal controller which satisfies the game problems (9.11). Furthermore, by construction, whenever the neighbor zones agree on the actions to carry out (and this can be assessed from the values taken by the coordinating logic variables), warm air is redirected from zones which are cooling down to zones which are heating up and are at lower temperatures. At the same time, the zones which are heating up collaborate with the neighbor zones which are cooling down to increase the amount of outflow. The controller is summarized in table 9.1.

---

[2] In the sums, if $\gamma_i^+ = 0$ ($\gamma_i^- = 0$), then $\sum_{j=1}^{\gamma_i^+} Q_{out,j+i}^M = 0$ ($\sum_{j=1}^{\gamma_i^-} Q_{out,i-j}^M = 0$).

Mode	Zone 1 *Cooling Down*					
		$Q_{out,1}$	$Q_{in,1}$	u_1	$\sigma_{12}^{(1)}$	$\sigma_{21}^{(1)}$
1	Under any condition	$Q_{out,1}^M$	$Q_{in,1}^M$	0	1	0
	Zone 1 *Heating Up*					
2	$x_1 \geq x_2$	0	0	u_1^M	0	0
3	$x_1 < x_2$ $\sigma_{21}^{(2)} = 0$	0	0	u_1^M	0	1
4	$x_1 < x_2$ $\sigma_{21}^{(2)} = 1$	$Q_{out,1}^M$	0	u_1^M	0	1

Table 9.2: Summary of the control law for Zone 1.

Mode	Zone i *Cooling Down*							
		$Q_{out,i}$	$Q_{in,i}$	u_i	$\sigma_{i-1,i}^{(i)}$	$\sigma_{i,i-1}^{(i)}$	$\sigma_{i,i+1}^{(i)}$	$\sigma_{i+1,i}^{(i)}$
1	Under any condition	$Q_{out,i}^M$	$Q_{in,i}^M$	0	0	1	1	0
	Zone i *Heating Up*							
2	$x_{i-1} \leq x_i \wedge x_i \geq x_{i+1}$	0	0	u_i^M	0	0	0	0
3	$x_{i-1} > x_i \wedge x_i \geq x_{i+1}$ $\sigma_{i-1,i}^{(i-1)} = 0$	0	0	u_i^M	1	0	0	0
4	$x_{i-1} > x_i \wedge x_i \geq x_{i+1}$ $\sigma_{i-1,i}^{(i-1)} = 1$	$Q_{out,i}^M$	0	u_i^M	1	0	1	0
5	$x_{i-1} \leq x_i \wedge x_i < x_{i+1}$ $\sigma_{i+1,i}^{(i+1)} = 0$	0	0	u_i^M	0	0	0	1
6	$x_{i-1} \leq x_i \wedge x_i < x_{i+1}$ $\sigma_{i+1,i}^{(i+1)} = 1$	$Q_{out,i}^M$	0	u_i^M	0	1	0	1
7	$x_{i-1} > x_i \wedge x_i < x_{i+1}$ $\sigma_{i-1,i}^{(i-1)} = 0 \wedge \sigma_{i+1,i}^{(i+1)} = 0$	0	0	u_i^M	1	0	0	1
8	$x_{i-1} > x_i \wedge x_i < x_{i+1}$ $\sigma_{i-1,i}^{(i-1)} = 1 \wedge \sigma_{i+1,i}^{(i+1)} = 0$	$Q_{out,i}^M$	0	u_i^M	1	0	1	0
9	$x_{i-1} > x_i \wedge x_i < x_{i+1}$ $\sigma_{i-1,i}^{(i-1)} = 0 \wedge \sigma_{i+1,i}^{(i+1)} = 1$	$Q_{out,i}^M$	0	u_i^M	0	1	0	1
10	$x_{i-1} > x_i \wedge x_i < x_{i+1} \wedge x_{i-1} \geq x_{i+1}$ $\sigma_{i-1,i}^{(i-1)} = 1 \wedge \sigma_{i+1,i}^{(i+1)} = 1$	$Q_{out,i}^M$	0	u_i^M	1	0	1	0
11	$x_{i-1} > x_i \wedge x_i < x_{i+1} \wedge x_{i-1} < x_{i+1}$ $\sigma_{i-1,i}^{(i-1)} = 1 \wedge \sigma_{i+1,i}^{(i+1)} = 1$	$Q_{out,i}^M$	0	u_i^M	0	1	0	1

Table 9.1: Summary of the control law for Zone i. Transitions from the "Cooling Down" mode to the "Heating Up" modes are triggered only if the clause $x_i \geq x_i^M$ is verified. Similarly, converse transitions occur only if the clause $x_i \leq x_i^m$ holds true.

Mode		Zone N Cooling Down				
		$Q_{out,N}$	$Q_{in,N}$	u_N	$\sigma^{(N)}_{N-1,N}$	$\sigma^{(N)}_{N,N-1}$
1	Under any condition	$Q^M_{out,N}$	$Q^M_{in,N}$	0	0	1
		Zone N Heating Up				
2	$x_{N-1} < x_N$	0	0	u^M_N	0	0
3	$x_{N-1} \geq x_N$ $\sigma^{(N-1)}_{N-1,N} = 0$	0	0	u^M_N	1	0
4	$x_{N-1} \geq x_N$ $\sigma^{(N-1)}_{N-1,N} = 1$	$Q^M_{out,N}$	0	u^M_N	1	0

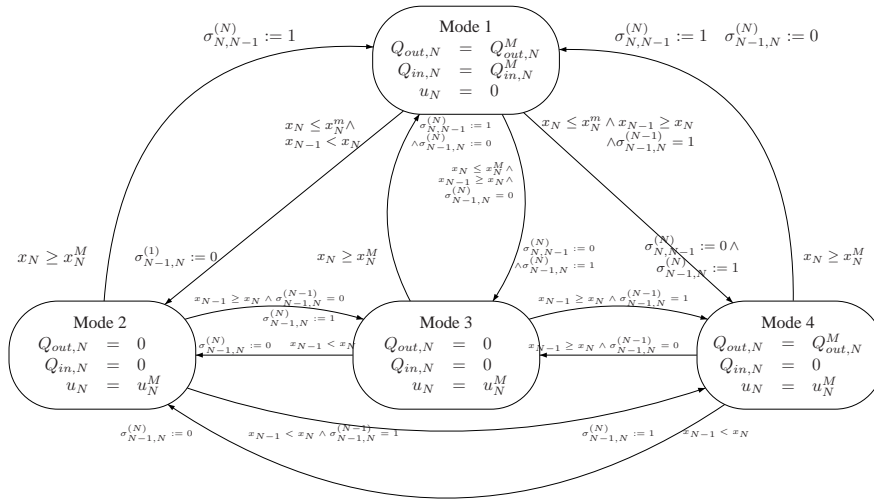Table 9.3: Summary of the control law for Zone N.

Figure 9.2: The event-based controller for Zone N.

For the special cases $i = 1, N$, the controller simplifies, as it can be seen in Table 9.2 and, respectively, 9.3. It then becomes very easy to represent the behavior of the switched controller by a graph (see Figures 9.3, and, respectively, 9.2).
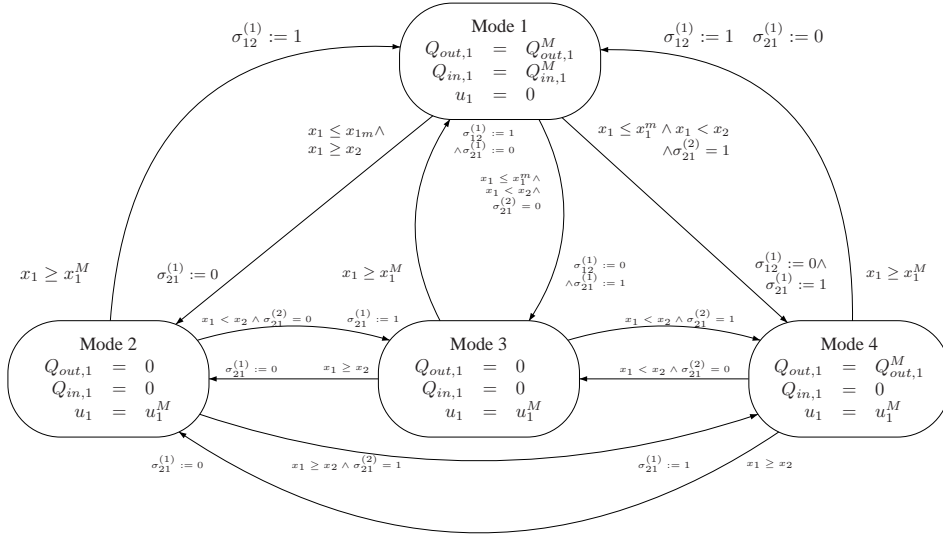
Figure 9.3: The graph represents the event-based controller for Zone 1. When the edge has two labels the first one represents the guard, the second one (denoted by :=) the reset. When no reset is present, then all the variables remain unchanged upon the transition. All the guards at the edges linking the three states at the bottom should include the clause $\neg x_1 \geq x_1^M$, which is omitted for the sake of simplicity. For each mode (discrete state) only three representative values are indicated. The remaining values can be derived from Table 9.2 and allow to obtain the continuous-time model associated with the discrete state. Thus, for instance, if the zone is in Mode 1, then it is evolving according to the equation $V_1 \dot{x}_1 = -\sum_{j=1}^{\gamma_1^+} Q_{out,j+1}^M x_1 - Q_{out,1}^M x_1 + w_{T1}$, with $\gamma_1^+ = (\gamma_2^+ + 1)\sigma_{12}^{(1)}\sigma_{12}^{(2)}$, and $\sigma_{12}^{(1)} = 1$, $\sigma_{21}^{(1)} = 0$.

Proposition 9.3.1 *Suppose that*

$$Q_{i,i+1} = \sum_{j=1}^{\gamma_i^+} Q_{out,j+i}^M \quad i = 1, 2, \ldots, N-1 \,,$$

$$(9.15)$$

$$Q_{i,i-1} = \sum_{j=1}^{\gamma_i^-} Q_{out,i-j}^M \quad i = 2, \ldots, N-2, N-1 \,,$$

and let, for each $i = 1, 2, \ldots, N$,

$$u_i^M > -w_{Ti}^m \,, \qquad (9.16)$$

121

and

$$Q_{out,i}^M x_i^M - w_{iT}^M > 0 \;, \tag{9.17}$$

hold. Then, for each $i = 1, 2, \ldots, N$, the maximal controlled invariant set of F_i coincides with F_i itself, the controller described in table 9.1 if $i \neq 1, N$ (respectively, in table 9.2 if $i = 1$ and in table 9.3 if $i = N$), renders F_i invariant and is the maximizing controller of the game problems (9.11).

Proof: See Appendix 9.7. □

Remark. A few observations are in order:

- Loosely speaking, the condition (9.15) amounts to require the flow balance fulfilled for each zone. We shall verify in the next section that this results in no loss of generality, for the controller is actually capable of guaranteeing the fulfillment of such constraint.

- Non strict inequalities in (9.16) and (9.17) suffice for F_i to be the maximal controlled invariant set. However, having them fulfilled with strict inequalities guarantees the temperature of each zone to be either decreasing or increasing.

- A few additional words about conditions (9.16) and (9.17) are in order. Although they may appear restrictive, they are very frequently encountered in practice. In fact, for many applications, the disturbance w_{Ti}^M is nonnegative, that is thermal dispersion is largely dominated by internally generated disturbance heat (this is especially true in livestock buildings or otherwise overcrowded closed environments). Hence, (9.16) is already satisfied with $u_i^M = 0$. (Notice that in this case, having a heating strategy still makes sense, in order to rapidly steer the temperature to a large enough value at which it is safe to let in fresh air from outside to increase the indoor air quality.) Furthermore, we shall see in the next section that it is not always possible to use warm air originated from neighbor zones to heat up the temperature of a zone. This is the case for instance when neighbor zones are themselves heating up. In such cases, the zone must be able to heat up only relying on its own heating system, and therefore condition (9.16) is necessary for the problem to be feasible.
  On the other hand, having exceedingly high indoor temperature can be harmful and must be avoided at any cost. Hence, it is mandatory to equip a room with a fan which is able to steer away the temperature from the dangerous limit, so that (9.17) is satisfied. Notice that, as for condition (9.16), even for condition (9.17) a smaller safety set does not help to relax these requirements. Indeed, (9.16) is independent of the state, while (9.17) is such that if it holds for any x_i which is inside the thermal region, then it is *a fortiori* true for $x_i = x_i^M$.

In the next section, we show that the controllers introduced here are actually feasible controllers, meaning that the flow balance (9.3) is fulfilled for each zone. In other words, it will become clear that the condition (9.15) is actually guaranteed by our design of the controller.

9.4 Feasibility of the controllers

The main obstacle to prove the feasibility of the controllers investigated in the previous section comes from the fact that the dynamics of each zone is closely intertwined with those of the neighbor zones and that the number of zones are arbitrarily large. Nevertheless, we can exploit the topology of the system, namely the configuration according to which the zones are positioned, to approach the problem by an inductive argument. In particular, we shall characterize conditions under which the flow balance is fulfilled for the first 2 zones. Then we shall proceed by showing the conditions under which, assuming that the flow balance is fulfilled up to Zone i, the flow balance is fulfilled even for Zone $i + 1$, and concluding the argument considering the zones $N - 1$ and N.

To make the statements below more concise we introduce the following definition:

Definition 9.4.1 *Let i be an integer such that $2 \leq i \leq N - 1$. Zones $1, 2, \ldots, i$ are said to* conditionally *satisfy the flow balance (9.3), if (9.3) is fulfilled for Zones $1, 2, \ldots, i$ provided that:*

- *If Zone i is in Mode 1, 4 or 8,*

$$
Q_{i,i+1} = \sum_{j=1}^{\gamma_i^+} Q_{out,\,j+i}^M \ , \tag{9.18}
$$

with $\gamma_i^+ = (\gamma_{i+1}^+ + 1)\sigma_{i,i+1}^{(i+1)}$, and $Q_{i+1,i} = 0$.

- *If Zone i is in Mode 2, 3, 5, 7 or 10,*

$$
Q_{i,i+1} = 0 \text{ and } Q_{i+1,i} = 0 \ . \tag{9.19}
$$

- *If Zone i is in Mode 6, 9 or 11, $Q_{i,i+1} = 0$ and*

$$
Q_{i+1,i} = \sum_{j=1}^{\gamma_{i+1}^-} Q_{out,\,i+1-j}^M \ , \tag{9.20}
$$

with $\gamma_{i+1}^- = \gamma_i^- + 1$.

In other words, Zones $1, 2, \ldots, i$ are said to *conditionally* satisfy the flow balance if all of them fulfill the flow balance provided that suitable conditions are met involving only the "terminal" Zone i.

In the following statements, we consider the N-zone climate control unit (9.5), (9.2), (9.3), (9.6), (9.8), under the action of the controllers introduced above. Proofs are postponed to the Appendices.

Lemma 9.4.1 *Zones 1 and 2 conditionally satisfy the flow balance (9.3).*

Proof: See Appendix 9.8. □

Then along the lines of the proof of the previous lemma, the following statement can be proven:

Lemma 9.4.2 *For some integer $2 \leq i \leq N - 2$, if Zones $1, 2, \ldots, i$ conditionally satisfy the flow balance (9.3), then also Zones $1, 2, \ldots, i, i + 1$ conditionally satisfy the flow balance (9.3).*

Proof: See Appendix 9.9. □

The arguments above can be finalized to prove that the flow balance is fulfilled for all the zones:

Proposition 9.4.1 *Zones $1, 2, \ldots, N - 1, N$ satisfy the flow balance (9.3).*

Proof: Zones 1 and 2 conditionally satisfy the flow balance (9.3) by Lemma 9.4.1. Then applying repeatedly Lemma 9.4.2, one can prove that actually Zones 1 to N-1 conditionally satisfy the flow balance. This means that, the flow balance is fulfilled for Zones 1 to N-1, provided that conditions (9.18)-(9.20) hold with $i = N - 1$. Below we prove that such conditions are actually true, and that consequently the flow balance is fulfilled even for Zone N. Suppose that Zone $N - 1$ is in Mode 1, 4 or 8. Then, we would like to prove

$$Q_{N-1,N} = \sum_{j=1}^{\gamma_{N-1}^{+}} Q_{out,\, j+N-1}^{M} \, ,$$

with $\gamma_{N-1}^{+} = (\gamma_{N}^{+} + 1)\sigma_{N-1,N}^{(N)}$, and $Q_{N,N-1} = 0$. Bearing in mind the control laws for Zones $N - 1$ and N, we can argue in the following way. When Zone N is in Mode 1 and 2, $\sigma_{N-1,N}^{(N)} = 0$ and we must verify that $Q_{N-1,N} = 0$. In Mode 1, $Q_{out,N} = Q_{out,N}^{M}$ and $Q_{in,N} = Q_{in,N}^{M}$. But, Zone $N - 1$ being in Mode 1, 4 or 8 yields that $\sigma_{N,N-1}^{(N-1)} = 0$ (see Table 9.1), and therefore $Q_{in,N}^{M} = Q_{out,N}^{M}$, which allows to conclude $Q_{N-1,N} = 0$ and the fulfillment of the flow balance for Zone N. In Mode 2, $Q_{out,N} = Q_{in,N} = 0$, and again $Q_{N-1,N} = 0$ and the fulfillment of the

124

flow balance in Zone N are trivially true. When Zone N is in Mode 3 and 4, $\sigma_{N-1,N}^{(N)} = 1$, and we must have $Q_{N-1,N} = Q_{out,N}^{M}$. It is not possible to have Zone N in Mode 3 and Zone $N-1$ in either one of Modes 1, 4 and 8 because of the contrasting requirements on the logic variables $\sigma_{N,N-1}^{(N-1)}$, $\sigma_{N-1,N}^{(N-1)}$ by the two Zones. On the other hand, in Mode 4, that $Q_{N-1,N} = Q_{out,N}^{M}$ is immediately verified because $Q_{out,N} = Q_{out,N}^{M}$ and $Q_{in,N} = 0$.

Suppose now that Zone $N-1$ is in Mode 2, 3, 5, 7 or 10. Then the flow balance for Zone N must be verified with $Q_{N-1,N} = Q_{N,N-1} = 0$. This is immediate if Zone N is in Mode 2 or 3. Suppose now that Zone N is in Mode 1, that is $Q_{out,N} = Q_{out,N}^{M}$ and $Q_{in,N} = Q_{in,N}^{M}$. Since $\gamma_N^- = (\gamma_{N-1}^- + 1)\sigma_{N,N-1}^{(N-1)} \cdot 1$, $\gamma_N^- = 0$ when Zone $N-1$ is in Mode 2, 3 or 10, so that $Q_{in,N}^{M} = Q_{out,N}^{M}$ and the flow balance is proven. On the other hand, the cases in which Zone $N-1$ is in Mode 5 or 7 and Zone N is in Mode 1 are not feasible because of $\sigma_{N-1,N}^{(N)}$, which is required to be 0 by Zone $N-1$ in Modes 5 or 7, and imposed to be 1 by Zone N in Mode 1. Suppose now that Zone N is in Mode 4. This occurs if $\sigma_{N-1,N}^{(N-1)} = 1$, which excludes the possibility for Zone $N-1$ to be in Mode 2, 3, 5, 7. Suppose then that Zone $N-1$ is in Mode 10. But neither this case is possible because we should have $x_{N-1} < x_N$, while Zone N in Mode 4 requires $x_{N-1} \geq x_N$.

Finally, consider the case that Zone $N-1$ is in Mode 6, 9 or 11. As this requires $\sigma_{N,N-1}^{(N)} = 1$, Zone N can only operate in Mode 1, as the other modes impose $\sigma_{N,N-1}^{(N)} = 0$. Furthermore, $\gamma_N^- = \gamma_{N-1}^- + 1$. This implies

$$Q_{in,N} = Q_{out,N}^{M} + \sum_{j=1}^{\gamma_{N-1}^- + 1} Q_{out,N-j}^{M}.$$

As $Q_{out,N} = Q_{out,N}^{M}$, then $Q_{N,N-1} = \sum_{j=1}^{\gamma_{N-1}^- + 1} Q_{out,N-j}^{M}$, and therefore the flow balance is satisfied for Zone N. This ends the proof. $\square$

9.5 Numerical results for a 3-zone climate control unit

In this section we present the outcome of a simulation for the 3-zone case, for which the *thermal regions* are defined as $T_{1m} = T_{3m} = 12\,°\text{C}$, $T_{1M} = T_{3M} = 14\,°\text{C}$, $T_{2m} = 14.5\,°\text{C}$ and $T_{2m} = 16.5\,°\text{C}$. For the convenience of the reader, the simulation is performed in the absolute coordinate system $T_i = x_i - T^{amb}$. The thermal regions picked for the simulation imply that whenever Zone 2 is cooling down and either Zone 1 or Zone 3 is heating up internal flow should occur. We also note that, under the present scenario, the controllers take the form illustrated in figures 9.4-9.6, which point out the event-based nature of the design.

The zone volumes are set to $V_1 = 2000\,m^3$ and $V_2 = V_3 = 1800\,m^3$, the maximum heating capacities to $u_1^M = u_2^M = 2\,°\text{C}m^3 s^{-1}$ and $u_3^M = 3\,°\text{C}m^3 s^{-1}$ and the fan capacities to $Q_{1,out}^{M} = 1.3\,m^3 s^{-1}$, $Q_{2,out}^{M} = 0.9\,m^3 s^{-1}$ and $Q_{3,out}^{M} = 1.2\,m^3 s^{-1}$. The disturbances as well as the ambient temperature are piecewise constant functions of time.
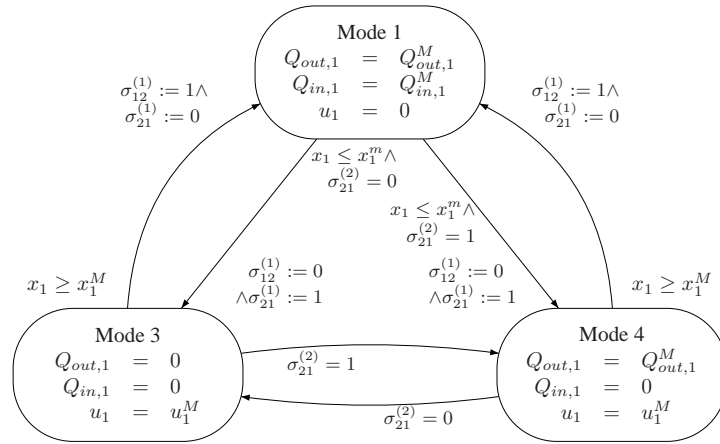
Figure 9.4: Controller for Zone 1 when $x_1^M < x_2^m$.
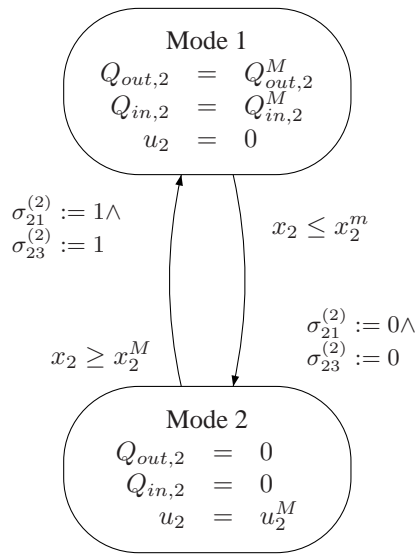
Figure 9.5: Controller for Zone 2 when $x_1^M < x_2^m > x_3^M$.
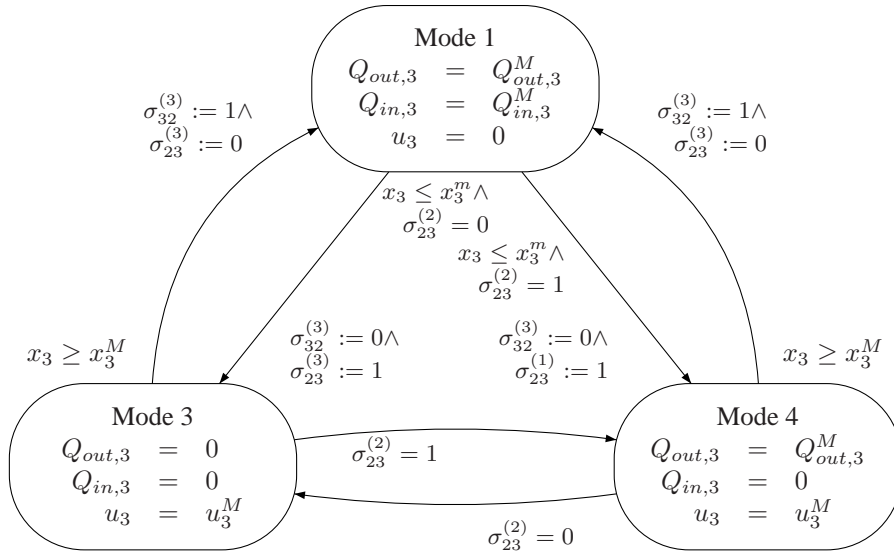
Figure 9.6: Controller for Zone 3 when $x_3^M < x_2^m$.

The initial state is set to $T = [14\,°\text{C}\ 16\,°\text{C}\ 12.5\,°\text{C}]^\text{T}$. Figure 9.7 illustrates the evolution of the three temperatures. Figure 9.8 illustrates the modes at which each controller is operating the considered time horizon. When Zone 2 is in Mode 1 and Zone 1 and/or 3 are in Mode 4, internal flow occurs. The occurrence of internal flow is depicted in Figure 9.9. To further point out the use of internal flow, Figure 9.10 illustrates the time profile for $Q_{2,in}$ and $Q_{2,out}$. While $Q_{2,out}$ is constantly equal to $Q_{2,out}^M = 1.9$ when nonzero, $Q_{2,in}^M$ depends on the cumulative flow variables γ_2^+ and γ_2^- and may take 5 different values.
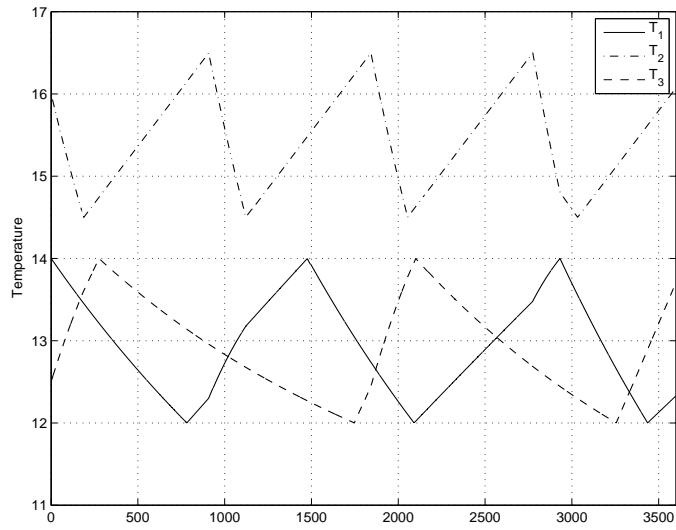
Figure 9.7: Time history of the three temperatures.

Figure 9.8: Evolution of the controller modes for the three controllers.

Figure 9.9: Time history for the internal flow in Zone 2.

Figure 9.10: Time history for the inflow and outflow in Zone 2.

9.6 Conclusion

The paper has discussed a control strategy for a multi zone climate unit capable of maintaining the state within a prescribed (safe) set, by managing the internal flow between zones. The control laws are inherently event-based and distributed, namely they only change action when certain boundari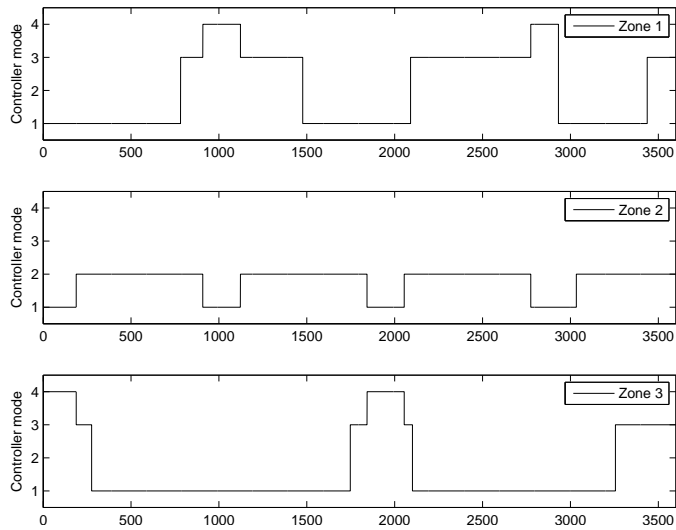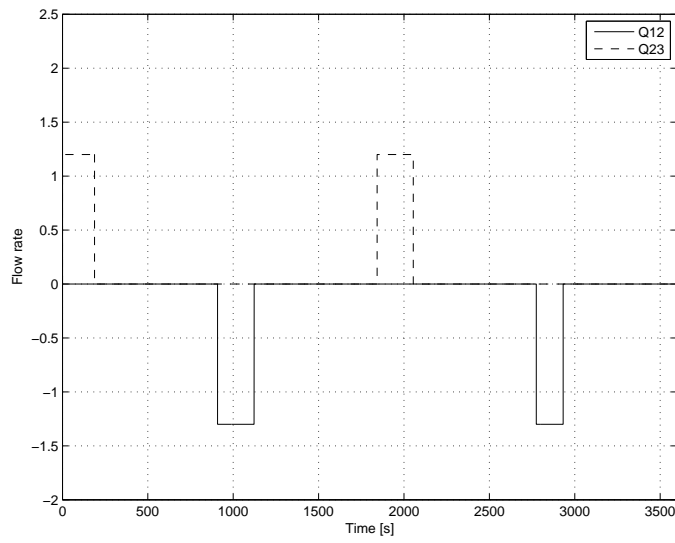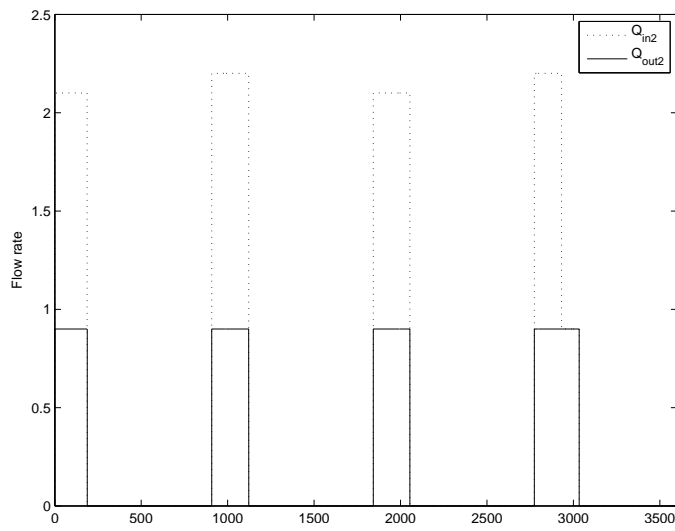es are met and/or when neighboring conditions change, and additionally they only require feedback information originated from neighboring zones. Our motivation for considering the devised control strategy is the possible implementation in a resource constrained environment using wireless battery powered climatic sensors. Hence, we were after a solution to the problem which allowed to reach the control goal by transmitting feedback information only sporadically. We have observed that the controllers take on values in a finite set, thus allowing for a potentially robust information transmission encoded using a finite number of bits. We have showed that the control law handles internal flow efficiently by using warm air or additional ventilation from neighbor zones to heat up or, respectively, cool down, whenever certain conditions are met. The proposed controller works for an arbitrarily large number of zones. The main feature to overcome the complexity of the analysis due to the large-scale nature of the problem consists of exploiting the topology of the system to show that the analysis and design of an N-zone system can always be reduced to the 3-zone case. The method is tailored to interconnected systems, with a time-invariant graph underlying the connections, and described by bilinear differential equations with linear algebraic constraints. This appears to be a framework common to other technological fields. In this respect, we believe that the same approach could prove useful even for other applications with more complex topologies and modeled by differential algebraic equations. On the other end, the system we present could be employed to test other design techniques for networked control systems, or to propose other problems of coordinated control. Finally, it could be interesting to investigate the trade-off between more demanding control objectives, such as set-point regulation, and the complexity of the controller, measured in terms of required data rate of the transmission channel and computational capability of the controller.

9.7 Proof of Proposition 9.3.1

The proof of the first part of the proposition, namely that F_i coincides with its maximal controlled invariant set, is shown by the following two simple lemma.

Lemma 9.7.1 *For any $i = 1, 2, \ldots, N$, if*

$$u_i^M \geq -w_{Tim} \;, \tag{9.21}$$

we have

$$\{x \,:\, J_i^{1*}(x) \geq 0\} = \{x \,:\, \ell_i^1(x) \geq 0\} \,. \tag{9.22}$$

Proof: Let first i be $\neq 1, N$. It is immediately seen that, by (9.3), the objective function of the static game

$$H_i^{1*}\left(x, \frac{\partial J_i^{1*}(x,0)}{\partial x}\right)$$

takes the form

$$Q_{i-1,i}(x_{i-1} - x_i) + Q_{i+1,i}(x_{i+1} - x_i) - Q_{in,i}x_i + u_i + w_{Ti} \,. \tag{9.23}$$

This shows that

$$H_i^{1*}\left(x, \frac{\partial J_i^{1*}(x,0)}{\partial x}\right) \geq u_i^M + w_{Ti}^m \,. \tag{9.24}$$

In fact, the minimizing w clearly requires $w_{Ti} = w_{Ti}^m$, whereas the maximizing U imposes $Q_{in,i} = 0$ and $u_i = u_i^M$, being $x_i \geq 0$. Furthermore, if $x_{i-1} - x_i \geq 0$ (respectively, $x_{i+1} - x_i \geq 0$), then the maximizing U yields $Q_{i-1,i} \geq 0$ ($Q_{i+1,i} \geq 0$), otherwise $Q_{i-1,i} = 0$ ($Q_{i+1,i} = 0$). In any case,

$$Q_{i-1,i}(x_{i-1} - x_i) + Q_{i+1,i}(x_{i+1} - x_i) \geq 0 \,,$$

and this shows (9.24). Since $u_i^M + w_{Ti}^m \geq 0$, we have

$$H_i^{1*}\left(x, \frac{\partial J_i^{1*}(x,0)}{\partial x}\right) \geq 0 \,.$$

One concludes that $J_i^{1*}(x, 0) \geq 0$ implies $J_i^{1*}(x, t) \geq 0$ for all t by (9.12), and hence $J_i^{1*}(x) \geq 0$, i.e. $\{x \,:\, J_i^{1*}(x) \geq 0\} \supseteq \{x \,:\, x_i - x_i^m \geq 0\}$. This proves the thesis for $i \neq 1, N$. In the case $i = 1$ (respectively, $i = N$), (9.23) becomes

$$Q_{21}(x_2 - x_1) - Q_{in,1}x_1 + u_1 + w_{T1} \quad (Q_{N-1,N}(x_{N-1} - x_N) - Q_{in,N}x_N + u_N + w_N)$$

and simple arguments as before show that (9.24) holds even with $i = 1, N$. The rest of the arguments go through exactly as in the case $i \neq 1, N$.

Lemma 9.7.2 *For any* $i = 1, 2, \ldots, N$, *if*

$$Q_{out,i}^M x_i^M - w_{iT}^M \geq 0 \,, \tag{9.25}$$

then

$$\{x \,:\, J_i^{2*}(x) \geq 0\} = \{x \,:\, -x_i + x_i^M \geq 0\} \,. \tag{9.26}$$

Proof: We proceed in the same way as in the previous lemma. In this case,

$$H_i^{2*}\left(x, \frac{\partial J_i^{2*}(x,0)}{\partial x}\right)$$

writes as

$$-Q_{i-1,i}x_{i-1} + Q_{i,i+1}x_i + Q_{out,i}x_i + Q_{i,i-1}x_i - Q_{i+1,i}x_{i+1} - u_i - w_{Ti} \,,$$

131

which immediately yields

$$H_i^{2*}\left(x, \frac{\partial J_i^{2*}(x,0)}{\partial x}\right) \geq V_i^{-1}\left[Q_{out,i}x_i - w_{Ti}^M\right]$$

being $x_{i-1}, x_i, x_{i+1} > 0$, $Q_{i,i+1}, Q_{i,i-1} \geq 0$. $J_i^{2*}(x,0) \geq 0$ then yields $x_i \leq x_i^M$. This and (9.17) imply $J_i^{2*}(x,t) \geq 0$ for all $t \geq 0$ and $J_i^{2*}(x) \geq 0$. The thesis is then immediately concluded. $\square$

It turns out that

$$F_i = \bigcap_{j=1}^{2}\{x \,:\, J_i^{j*}(x) \geq 0\}\,,$$

that is the first part of the thesis. That F_i is invariant under the action of the controller i is also easily verified by checking that, thanks to the definition of the controller and (9.16), (9.17), the velocity vector always points inward F_i whenever x_i is on the boundary of F_i. This also points out that at each time Zone i is either cooling down or heating up. To prove the last part of the proposition, we focus on each one of these mutually exclusive cases.

Zone i is cooling down. In this case, the computation of the maximizing controller $U(\cdot)$ for the game $J_i^{2*}(x,t)$ reduces to the optimization problem

$$\max_{U \in \mathbf{U}} \min_{W \in \mathbf{W}} \left\{-Q_{i-1,i}x_{i-1} + Q_{i,i+1}x_i + Q_{out,i}x_i + Q_{i,i-1}x_i - Q_{i+1,i}x_{i+1} - u_i - w_{Ti}\right\}\,.$$

This admits the solution

$$Q_{i-1,i}^* = Q_{i+1,i}^* = 0\,, \quad Q_{out,i}^* = Q_{out,i}^M\,, \quad u_i^* = 0\,, \quad w_{Ti*} = w_{Ti}^M\,,$$

where $Q_{i,i+1}^*, Q_{i,i-1}^*, Q_{in,i}^*$ are the largest values which satisfy

$$Q_{i,i+1}^* + Q_{out,i}^M + Q_{i,i-1}^* = Q_{in,i}^*\,,$$

and comply to the constraints imposed by the neighbors, embodied by the logic variables $\sigma_{i,i+1}^{(i+1)}$, $\sigma_{i+1,i}^{(i+1)}, \sigma_{i-1,i}^{(i-1)}, \sigma_{i,i-1}^{(i-1)}$. Such largest values are achieved by letting

$$Q_{in,i}^* = Q_{in,i}^M\,, \quad \sigma_{i,i+1}^{(i)*} = 1\,, \quad \sigma_{i,i-1}^{(i)*} = 1\,.$$

In fact, by condition (9.15), the choice above yields that

$$Q_{i,i+1}^* = \begin{cases} \sum_{j=1}^{\gamma_{i+1}^+ +1} Q_{out,j+i}^M & \text{for} \quad \sigma_{i,i+1}^{(i+1)} = 1 \\ 0 & \text{for} \quad \sigma_{i,i+1}^{(i+1)} = 0 \end{cases}$$

and

$$Q_{i,i-1}^* = \begin{cases} \sum_{j=1}^{\gamma_{i-1}^- +1} Q_{out,i-j}^M & \text{for} \quad \sigma_{i,i-1}^{(i-1)} = 1 \\ 0 & \text{for} \quad \sigma_{i,i-1}^{(i-1)} = 0\,. \end{cases}$$

Zone i is heating up. The optimization problem in this case takes the form

$$\max_{U \in \mathbf{U}} \min_{W \in \mathbf{W}} \left\{Q_{i-1,i}(x_{i-1} - x_i) + Q_{i+1,i}(x_{i+1} - x_i) - Q_{in,i}x_i + u_i + w_{Ti}\right\}\,.$$

We have

$$Q^*_{in,i} = 0 , \quad u^*_i = u^M_i , \quad w_{Ti*} = w^m_{Ti} .$$

The optimal values for the remaining control variables can be decided on the basis of the values of x_{i-1}, x_i, x_{i+1} and of the coordinating logic variables set by the neighbors. The cases to be examined are as follows.

- $x_{i-1} \leq x_i \wedge x_i \geq x_{i+1}$. In this case $Q^*_{i-1,i} = Q^*_{i+1,i} = 0$. By (9.3), and $Q^*_{in,i} = 0$ established above, it must also be true $Q^*_{out,i} = Q^*_{i,i-1} = Q^*_{i,i+1} = 0$. Hence, we set

$$\sigma^{(i)*}_{i,i-1} = \sigma^{(i)*}_{i-1,i} = \sigma^{(i)*}_{i+1,i} = \sigma^{(i)*}_{i,i+1} = 0 .$$

This forbids neighbors to draw (respectively, release) air from (to) Zone i.

- $x_{i-1} > x_i \wedge x_i \geq x_{i+1}$. We must have $Q^*_{i+1,i} = 0$, and therefore, by the flow balance,

$$Q^*_{i-1,i} = Q^*_{out,i} + Q^*_{i,i-1} + Q^*_{i,i+1} , \tag{9.27}$$

where all the values in the equation above must be as large as possible (so that, by (9.2), $Q^*_{i,i-1} = 0$) and conform to the constraints imposed by Zone $i - 1$.
If $\sigma^{(i-1)}_{i-1,i} = 0$, then

$$Q^*_{i-1,i} = Q^*_{out,i} = Q^*_{i,i-1} = Q^*_{i,i+1} = 0 ,$$

and we must set

$$\sigma^{(i)*}_{i,i-1} = \sigma^{(i)*}_{i+1,i} = \sigma^{(i)*}_{i,i+1} = 0 .$$

In this case, it is not necessary to set $\sigma^{(i-1)*}_{i-1,i} = 0$, because $\sigma^{(i-1)}_{i-1,i} = 0$ already. In fact, we set $\sigma^{(i)*}_{i-1,i} = 1$, for, if in the meanwhile Zone $i - 1$ happens to switch to a new controller, $\sigma^{(i)}_{i-1,i} = 1$ signals to Zone $i - 1$ that Zone i is willing to draw warm air from it, and this will correctly affect the decision of Zone $i - 1$ regarding which controller to switch.
On the other hand, if $\sigma^{(i-1)}_{i-1,i} = 1$, by (9.27), we must set

$$\sigma^{(i)*}_{i-1,i} = 1 , \quad Q^*_{out,i} = Q^M_{out,i} , \quad \sigma^{(i)*}_{i,i+1} = 1 .$$

In this way,

$$Q^*_{i,i+1} = \begin{cases} \sum_{j=1}^{\gamma^+_{i+1}+1} Q^M_{out,j+i} & \text{for} \quad \sigma^{(i+1)}_{i,i+1} = 1 \\ 0 & \text{for} \quad \sigma^{(i+1)}_{i,i+1} = 0 \end{cases}$$

and

$$Q^*_{i-1,i} = Q^M_{out,i} + Q^*_{i,i+1} ,$$

is at its maximum given the constraints.

- $x_{i-1} \leq x_i \wedge x_i < x_{i+1}$. It is the case symmetric to the previous one and the optimal solution can be immediately derived. If $\sigma^{(i+1)}_{i+1,i} = 0$, then

$$Q^*_{out,i} = \sigma^{(i)*}_{i,i-1} = \sigma^{(i)*}_{i-1,i} = \sigma^{(i)*}_{i,i+1} = 0 .$$

and $\sigma^{(i)*}_{i+1,i} = 1$. If $\sigma^{(i+1)}_{i+1,i} = 1$, then

$$\sigma^{(i)*}_{i,i-1} = 1 , \quad Q^*_{out,i} = Q^M_{out,i} , \quad \sigma^{(i)*}_{i+1,i} = 1 .$$

- $x_{i-1} > x_i \land x_i < x_{i+1}$. We distinguish 5 sub-cases:

  (i) $\sigma_{i-1,i}^{(i-1)} = 0 \land \sigma_{i+1,i}^{(i+1)} = 0$. We must have

  $$Q_{i-1,i}^* = Q_{i+1,i}^* = 0 \ ,$$

  and hence

  $$Q_{i,i-1}^* = Q_{out,i}^* = Q_{i,i+1}^* = 0$$

  as well. To enforce this, we set

  $$\sigma_{i,i-1}^{(i)*} = \sigma_{i,i+1}^{(i)*} = 0 \ .$$

  Even in this case, it is not necessary to set $\sigma_{i-1,i}^{(i)*} = \sigma_{i+1,i}^{(i)*} = 0$, for $\sigma_{i-1,i}^{(i-1)} = \sigma_{i+1,i}^{(i+11)} = 0$, and indeed we set

  $$\sigma_{i-1,i}^{(i)*} = \sigma_{i+1,i}^{(i)*} = 1 \ .$$

  (ii) $\sigma_{i-1,i}^{(i-1)} = 1 \land \sigma_{i+1,i}^{(i+1)} = 0$. This reduces to the case

  $$x_{i-1} > x_i \land x_i \geq x_{i+1} \land \sigma_{i-1,i}^{(i-1)} = 1 \ ,$$

  already examined above.

  (iii) $\sigma_{i-1,i}^{(i-1)} = 0 \land \sigma_{i+1,i}^{(i+1)} = 1$. This reduces to the case

  $$x_{i-1} \leq x_i \land x_i < x_{i+1} \land \sigma_{i+1,i}^{(i+1)} = 1 \ .$$

  (iv) $\sigma_{i-1,i}^{(i-1)} = 1 \land \sigma_{i+1,i}^{(i+1)} = 1 \land x_{i-1} \geq x_{i+1}$. As for (ii), this reduces to the case

  $$x_{i-1} > x_i \land x_i \geq x_{i+1} \land \sigma_{i-1,i}^{(i-1)} = 1 \ .$$

  (v) $\sigma_{i-1,i}^{(i-1)} = 1 \land \sigma_{i+1,i}^{(i+1)} = 1 \land x_{i-1} < x_{i+1}$. As for (iii), this case reduces to

  $$x_{i-1} \leq x_i \land x_i < x_{i+1} \land \sigma_{i+1,i}^{(i+1)} = 1 \ .$$

□

9.8 Proof of Lemma 9.4.1

We consider combinations of modes for Zone 1 and 2. Note that the majority possible cases are not feasible, and hence the analysis is simpler than it could appear. We shall refer to the case in which Zone 1 is in Mode i and Zone 2 is in Mode j as *Case i.j*. For the convenience of the readers, while the first cases will be examined more in detail, we shall proceed faster as we get acquainted with the line of reasoning which underlies the proof. Let us recall that we are to prove that the flow balance (9.3) is fulfilled for $i = 1, 2$, i.e. for Zone 1 and 2, provided that:

- If Zone 2 is in Mode 1, 4 or 8,

$$Q_{23} = \sum_{j=1}^{\gamma_2^+} Q_{out,\,j+2}^M \,, \tag{9.28}$$

with $\gamma_2^+ = (\gamma_3^+ + 1)\sigma_{23}^{(3)}$, and $Q_{32} = 0$.

- If Zone 2 is in Mode 2, 3, 5, 7 or 10,

$$Q_{23} = 0 \text{ and } Q_{32} = 0 \,. \tag{9.29}$$

- If Zone 2 is in Mode 6, 9 or 11, $Q_{23} = 0$ and

$$Q_{32} = \sum_{j=1}^{\gamma_3^-} Q_{out,\,3-j}^M \,, \tag{9.30}$$

with $\gamma_3^- = \gamma_2^- + 1$.

Consider the *Case 1.1*. Then both zones are cooling down and no air exchange is possible. Indeed,

$$\gamma_1^+ = (\gamma_2^+ + 1)\sigma_{12}^{(1)}\sigma_{12}^{(2)} = (\gamma_2^+ + 1) \cdot 1 \cdot 0 = 0 \text{ and } \gamma_2^- = \sigma_{21}^{(1)}\sigma_{21}^{(2)} = 0 \cdot 1 = 0 \,.$$

As a consequence, $Q_{in,1} = Q_{out,1}^M$. As far as Zone 2 is concerned, observe that $Q_{out,2} = Q_{out,2}^M$ and $\gamma_2^+ = (\gamma_3^+ + 1) \cdot 1 \cdot \sigma_{23}^{(3)}$. Hence,

$$Q_{in,2} = Q_{out,2}^M + \sum_{j=1}^{\gamma_2^-} Q_{out,2-j}^M + \sum_{j=1}^{\gamma_2^+} Q_{out,j+2}^M = Q_{out,2}^M + \sum_{j=1}^{\gamma_2^+} Q_{out,j+2}^M \,.$$

We conclude that, as a total of $Q_{out,1}^M + Q_{out,2}^M$ is removed from Zone 1 and 2, and a total inflow of $Q_{out,1}^M + Q_{out,2}^M + \sum_{j=1}^{\gamma_2^+} Q_{out,j+2}^M$ is allowed through the 2 inlets, the flow balance for the 2 zones is fulfilled provided that (9.28) holds. *Case 2.1*. Mode 2 for Zone 1 imposes that $Q_{out,1} = Q_{in,1} = 0$. As before, $\gamma_2^- = 0$, $\gamma_2^+ = (\gamma_3^+ + 1)\sigma_{23}^{(3)}$, $Q_{out,2} = Q_{out,2}^M$ and $Q_{in,2} = Q_{out,2}^M + \sum_{j=1}^{\gamma_2^+} Q_{out,j+2}^M$, we can draw exactly the same conclusion as above. The *Case 3.1* is not feasible because, Zone 2 in Mode 1 imposes $\sigma_{21}^{(2)} = 1$, and this induces an immediate transition of Zone 1 from Mode 3 to 4. On the other hand, it is admissible the *Case 4.1*. As $Q_{out,1} = Q_{out,1}^M$ and $Q_{in,1} = 0$, then $Q_{21} = Q_{out,1}^M$. Consistently, $\gamma_2^- = \sigma_{21}^{(1)}\sigma_{21}^{(2)} = 1$. Additionally, $\gamma_2^+ = (\gamma_3^+ + 1) \cdot 1 \cdot \sigma_{23}^{(3)}$ yields

$$Q_{in,2} = Q_{out,2}^M + \sum_{j=1}^{\gamma_2^-} Q_{out,2-j}^M + \sum_{j=1}^{\gamma_2^+} Q_{out,j+2}^M = Q_{out,2}^M + Q_{out,1}^M + \sum_{j=1}^{\gamma_2^+} Q_{out,j+2}^M \,,$$

that is, again, the flow balance for the 2 zones is fulfilled provided that (9.28) is true.

Case 1.2. $\gamma_1^+ = (\gamma_2^+ + 1)\sigma_{12}^{(1)}\sigma_{12}^{(2)} = (\gamma_2^+ + 1) \cdot 1 \cdot 0 = 0$ yields $Q_{in,1} = Q_{out,1}^M$, with $Q_{out,1} = Q_{out,1}^M$. In Mode 2, $Q_{in,2} = Q_{out,2} = 0$, and $\gamma_2^- = \gamma_2^+ = 0$, therefore for the 2 zones to have the flow balance fulfilled, (9.29) must be true. Under *Cases 2.2* and *3.2* (the former is feasible only if $x_1 = x_2$), $Q_{in,i} = Q_{out,i} = 0$ for both $i = 1, 2$. Hence, for the flow balance in the 2 zones to be fulfilled, (9.29) must hold. The *Case 4.2* is not feasible, as Zone 1 being in Mode 4 requires $\sigma_{21}^{(2)} = 1$ (otherwise, a transition to Mode 3 would occur), while Zone 2 in Mode 2 imposes $\sigma_{21}^{(2)} = 0$.

Case 1.3. This is unfeasible, for Mode 1 imposes $\sigma_{12}^{(1)} = 1$, which would imply a transition out from Mode 3 for Zone 2. Under the *Case 2.3*, $Q_{in,i} = Q_{out,i} = 0$ for both $i = 1, 2$, and hence the flow balance for Zones 1 and 2 is fulfilled provided that (9.29) holds. The two *Case 3.3* and *Case 4.3* are not compatible, as it can not be simultaneously $x_1 > x_2$ and $x_1 < x_2$.

When Zone 2 is in Mode 4, only the *Case 1.4* is possible. We have: $Q_{out,i} = Q_{out,i}^M$, for $i = 1, 2$ and $Q_{in,2} = 0$. Also, $\gamma_1^+ = \gamma_2^+ + 1$, and $\gamma_2^+ = (\gamma_3^+ + 1) \cdot 1 \cdot \sigma_{23}^{(3)}$, imply $Q_{in,1} = Q_{out,1} + Q_{out,2} + \sum_{j=1}^{\gamma_2^+} Q_{out,j+2}^M$, from which we conclude that the flow balance is preserved provided that $Q_{23} = \sum_{j=1}^{\gamma_2^+} Q_{out,j+2}^M$.

When Zone 2 is in Mode 5, only *Cases 1.5* and *2.5-3.5* must be checked. In the former case, $\gamma_1^+ = 0$, and hence $Q_{in,1} = Q_{out,1} = Q_{out,1}^M$. On the other hand $Q_{in,2} = Q_{out,2} = 0$, and hence (9.29) must be true. In the latter two cases, $Q_{in,i} = Q_{out,i} = 0$ for both $i = 1, 2$, and again (9.29) must be true.

Consider now the *Case 1.6*. We have $Q_{out,1} = Q_{out,1}^M$, $\gamma_1^+ = (\gamma_2^+ + 1)\sigma_{12}^{(1)}\sigma_{12}^{(2)} = 0$, and therefore $Q_{in,1} = Q_{out,1}^M$. On the other hand, $Q_{out,2} = Q_{out,2}^M$, $Q_{in,2} = 0$, and $\gamma_2^+ = \gamma_2^- = 0$, whereas $\gamma_3^- = \sigma_{32}^{(2)}\sigma_{32}^{(3)} = 1$, since for Zone 2 to be in Mode 6, it is required $\sigma_{32}^{(3)} = 1$. We conclude that (9.30) must be true. Under *Case 2.6* (which is feasible only if $x_1 = x_2$), it is easily verified that $Q_{32} = Q_{out,1}^M$, $\gamma_2^- = 0$ and $\gamma_3^- = 1$. Hence, $Q_{32} = \sum_{j=1}^{\gamma_3^-} Q_{out,3-j}^M$, that is (9.30). *Case 3.6* is not feasible. We are left with *Case 4.6*. In this case, $Q_{in,i} = 0$ and $Q_{out,i} = Q_{out,i}^M$ for $i = 1, 2$, so that the fulfillment of the flow balance requires $Q_{32} = Q_{out,1}^M + Q_{out,2}^M$. As $\gamma_2^- = 1$ and $\gamma_3^- = (\gamma_2^- + 1) = 2$, (9.30) is immediately proven.

We examine now the *Case 1.7*. This is not feasible, for Mode 1 imposes $\sigma_{12}^{(1)} = 1$, which would cause Zone 2 to switch from Mode 7 to Mode 8. We discard the *Cases 3.7* and *4.7* too, as the requirements on the temperatures x_1, x_2 are contradictory. This is still true for all the cases for which Zone 1 is in Mode 3 or 4 whereas Zone 2 is in mode 7 *to* 11, and these will be ignored in the sequel. The only case to investigate is *Case 2.7*. It is immediately verified that (9.29) must hold.

When Zone 2 is in Mode 8, the only cases to consider are *Cases 1.8* and *2.8*. Under *Case 1.8*, $Q_{out,i} = Q_{out,i}^M$, for $i = 1, 2$, $Q_{in,1} = Q_{in,1}^M$, $Q_{in,2} = 0$, $\gamma_1^+ = \gamma_2^+ + 1$, and $\gamma_2^+ = (\gamma_3^+ + 1) \cdot 1 \cdot \sigma_{23}^{(3)}$. Hence $Q_{in,1} = Q_{out,1}^M + \sum_{j=1}^{\gamma_2^+ + 1} Q_{out,j+1}^M = Q_{out,1}^M + Q_{out,2}^M +$

$\sum_{j=1}^{\gamma_2^+} Q_{out,j+2}^M$, which gives (9.28). *Case 2.8* yields $\gamma_1^+ = \gamma_2^- = 0$, $Q_{out,2} = Q_{out,2}^M$, $Q_{in,2} = 0$, and $\gamma_2^+ = (\gamma_3^+ + 1)\sigma_{23}^{(3)}$, so that (9.28) must hold.

The *Case 1.9* is not feasible because there are opposing requests on $\sigma_{12}^{(1)}$. We examine the only possible case, namely *Case 2.9*. We have $\gamma_1^+ = 0$, $Q_{out,1} = Q_{in,1} = 0$, $Q_{out,2} = Q_{out,2}^M$, $Q_{in,2} = 0$, $\gamma_2^- = 0$, $\gamma_3^- = 1$. The latter 2 equalities in particular allow to verify (9.30).

In the *Case 1.10*, $Q_{out,1} = Q_{out,1}^M$, $Q_{in,1} = Q_{in,1}^M$, with $Q_{in,1}^M$ easily computed from $\gamma_1^+ = (\gamma_2^+ + 1)$, $\gamma_2^+ = (\gamma_3^+ + 1)\sigma_{23}^{(3)}$. Moreover, $Q_{out,2} = Q_{out,2}^M$, $Q_{in,2} = 0$, and as a result the correctness of (9.29) is proven. The *Case 2.10* is not feasible (compare the requirements on $\sigma_{12}^{(1)}$ by the 2 Zones).

We are left with the last feasible case, i.e. *Case 1.11*. Then $Q_{out,1} = Q_{out,1}^M$ and $\gamma_1^+ = 0$, so that $Q_{in,1} = Q_{out,1}^M$. Moreover, $Q_{out,2} = Q_{out,2}^M$, $Q_{in,2} = 0$, $\gamma_2^- = 0$, $\gamma_2^+ = 0$, and $\gamma_3^- = 1$. The validity of (9.30) is immediately inferred.

9.9 Proof of Lemma 9.4.2

We prove that Zones $1, 2, \ldots, i + 1$ conditionally satisfy the flow balance, that is, the flow balance is fulfilled for those zones provided that:

- If Zone $i + 1$ is in Mode 1, 4 or 8,

$$Q_{i+1,i+2} = \sum_{j=1}^{\gamma_{i+1}^+} Q_{out,\,j+i+1}^M \,, \tag{9.31}$$

  with $\gamma_{i+1}^+ = (\gamma_{i+2}^+ + 1)\sigma_{i+1,i+2}^{(i+2)}$, and $Q_{i+2,i+1} = 0$.

- If Zone $i + 1$ is in Mode 2, 3, 5, 7 or 10,

$$Q_{i+1,i+2} = 0 \text{ and } Q_{i+2,i+1} = 0 \,. \tag{9.32}$$

- If Zone $i + 1$ is in Mode 6, 9 or 11, $Q_{i+1,i+2} = 0$ and

$$Q_{i+2,i+1} = \sum_{j=1}^{\gamma_{i+2}^-} Q_{out,\,i+2-j}^M \,, \tag{9.33}$$

  with $\gamma_{i+2}^- = \gamma_{i+1}^- + 1$.

In principal the proof goes very similar to the one of the previous lemma, but we change it to cope with the fact that in the current case the controller mode combinations occur in larger number. Zone $i + 1$ depends on Zone i and $i + 2$ so in total there are $11 \cdot 11 \cdot 11 =$

1331 different modes to consider. Many of these modes are however not feasible i.e. specific mode combinations would immediately cause a transition to a different mode combination. To narrow the number of combinations to consider, it suffices to take into account the possible combinations of coordinating logic variables for Zones i, $i+1$ and $i+2$. Noticing that simultaneous flow into a zone from both neighboring zones is impossible, it is enough to consider the following cases:

1.
$$
\begin{array}{ll}
\sigma_{i,i+1}^{(i)}\sigma_{i,i+1}^{(i+1)} = 0 & \sigma_{i+1,i+2}^{(i+1)}\sigma_{i+1,i+2}^{(i+2)} = 0 \\
\sigma_{i+1,i}^{(i)}\sigma_{i+1,i}^{(i+1)} = 0 & \sigma_{i+2,i+1}^{(i+1)}\sigma_{i+2,i+1}^{(i+2)} = 0
\end{array}
\tag{9.34}
$$

2.
$$
\begin{array}{ll}
\sigma_{i,i+1}^{(i)}\sigma_{i,i+1}^{(i+1)} = 0 & \sigma_{i+1,i+2}^{(i+1)}\sigma_{i+1,i+2}^{(i+2)} = 0 \\
\sigma_{i+1,i}^{(i)}\sigma_{i+1,i}^{(i+1)} = 1 & \sigma_{i+2,i+1}^{(i+1)}\sigma_{i+2,i+1}^{(i+2)} = 0
\end{array}
\tag{9.35}
$$

3.
$$
\begin{array}{ll}
\sigma_{i,i+1}^{(i)}\sigma_{i,i+1}^{(i+1)} = 0 & \sigma_{i+1,i+2}^{(i+1)}\sigma_{i+1,i+2}^{(i+2)} = 1 \\
\sigma_{i+1,i}^{(i)}\sigma_{i+1,i}^{(i+1)} = 0 & \sigma_{i+2,i+1}^{(i+1)}\sigma_{i+2,i+1}^{(i+2)} = 0
\end{array}
\tag{9.36}
$$

4.
$$
\begin{array}{ll}
\sigma_{i,i+1}^{(i)}\sigma_{i,i+1}^{(i+1)} = 0 & \sigma_{i+1,i+2}^{(i+1)}\sigma_{i+1,i+2}^{(i+2)} = 1 \\
\sigma_{i+1,i}^{(i)}\sigma_{i+1,i}^{(i+1)} = 1 & \sigma_{i+2,i+1}^{(i+1)}\sigma_{i+2,i+1}^{(i+2)} = 0
\end{array}
\tag{9.37}
$$

5.
$$
\begin{array}{ll}
\sigma_{i,i+1}^{(i)}\sigma_{i,i+1}^{(i+1)} = 1 & \sigma_{i+1,i+2}^{(i+1)}\sigma_{i+1,i+2}^{(i+2)} = 0 \\
\sigma_{i+1,i}^{(i)}\sigma_{i+1,i}^{(i+1)} = 0 & \sigma_{i+2,i+1}^{(i+1)}\sigma_{i+2,i+1}^{(i+2)} = 0
\end{array}
\tag{9.38}
$$

6.
$$
\begin{array}{ll}
\sigma_{i,i+1}^{(i)}\sigma_{i,i+1}^{(i+1)} = 0 & \sigma_{i+1,i+2}^{(i+1)}\sigma_{i+1,i+2}^{(i+2)} = 0 \\
\sigma_{i+1,i}^{(i)}\sigma_{i+1,i}^{(i+1)} = 0 & \sigma_{i+2,i+1}^{(i+1)}[\sigma_{i+2,i+1}^{(i+2)} = 1
\end{array}
\tag{9.39}
$$

7.
$$
\begin{array}{ll}
\sigma_{i,i+1}^{(i)}\sigma_{i,i+1}^{(i+1)} = 1 & \sigma_{i+1,i+2}^{(i+1)}\sigma_{i+1,i+2}^{(i+2)} = 1 \\
\sigma_{i+1,i}^{(i)}\sigma_{i+1,i}^{(i+1)} = 0 & \sigma_{i+2,i+1}^{(i+1)}\sigma_{i+2,i+1}^{(i+2)} = 0
\end{array}
\tag{9.40}
$$

8.
$$
\begin{array}{ll}
\sigma_{i,i+1}^{(i)}\sigma_{i,i+1}^{(i+1)} = 0 & \sigma_{i+1,i+2}^{(i+1)}\sigma_{i+1,i+2}^{(i+2)} = 0 \\
\sigma_{i+1,i}^{(i)}\sigma_{i+1,i}^{(i+1)} = 1 & \sigma_{i+2,i+1}^{(i+1)}\sigma_{i+2,i+1}^{(i+2)} = 1
\end{array}
\tag{9.41}
$$

Case (i). In this case, it is immediately seen that $\gamma_i^+ = \gamma_i^- = 0$ and $\gamma_{i+1}^+ = \gamma_{i+1}^- = \gamma_{i+2}^- = 0$. The former pair of equalities and the hypothesis that Zone $1, 2, \ldots, i$ conditionally satisfy the flow balance yield that necessarily $Q_{i,i+1} = Q_{i+1,i} = 0$. We distinguish two sub cases: Zone $i+1$ is heating up and Zone $i+1$ is cooling down. If Zone $i+1$

is heating up, then $Q_{in,i+1} = 0$, and since no internal ventilation can occur with neighboring zones, necessarily $Q_{out,i+1} = 0$ as well. This implies $Q_{i+1,i+2} = Q_{i+2,i+1} = 0$. When zone $i+1$ is cooling down the outflow provided by the fan is $Q_{out,i+1} = Q_{out,i+1}^M$. Now (9.34) leads to:

$$Q_{in,i+1} = Q_{out,i+1}^M + \sum_{j=1}^{\gamma_{i+1}^-} Q_{out,i+1-j}^M + \sum_{j=1}^{\gamma_{i+1}^+} Q_{out,j+i+1}^M = Q_{out,i+1}^M \qquad (9.42)$$

and hence $Q_{i+1,i+2} = Q_{i+2,i+1} = 0$. The facts above prove that (9.31) and (9.32) are true. We are left with verifying that Zone $i+1$ can not be in Mode 6, 9, or 11, as this would require $Q_{i+2,i+1} \neq 0$ (see (9.33)). Zone $i+1$ can be in Mode 6, 9, or 11 only if $\sigma_{i+2,i+1}^{(i+1)} \sigma_{i+2,i+1}^{(i+2)} = 1$, which contradicts (9.34). *Case (ii)*. It is easily verified that in this case Zone i can be in Mode 6,9 or 11 only, whereas Zone $i+1$ can be only in Mode 1. By hypothesis, the former fact gives that $Q_{i+1,i} = \sum_{j=1}^{\gamma_{i+1}^-} Q_{out,i+1-j}^M$. Because of (9.35), $\sum_{j=1}^{\gamma_{i+1}^+} Q_{out,j+i+1}^M = 0$, and the inlet is set to:

$$Q_{in,i+1} = Q_{out,i+1}^M + \sum_{j=1}^{\gamma_{i+1}^-} Q_{out,i+1-j}^M$$

Since zone $i+1$ is cooling down with $Q_{out,i+1} = Q_{out,i+1}^M$, this gives $Q_{i+1,i+2} = Q_{i+2,i+1} = 0$. As for the case before, (9.31) and (9.32) trivially hold, while Zone $i+1$ can never be in Mode 6, 9, or 11. *Case (iii)*. With (9.36), no air exchange between Zone i and $i+1$ is possible, and in particular Zone $i+1$ is in Mode 1, i.e. cooling down with $Q_{out,i+1} = Q_{out,i+1}^M$. The inlet is set to:

$$Q_{in,i+1} = Q_{out,i+1}^M + \sum_{j=1}^{\gamma_{i+1}^-} Q_{out,i+1-j}^M + \sum_{j=1}^{\gamma_{i+1}^+} Q_{out,j+i+1}^M = Q_{out,i+1}^M + \sum_{j=1}^{\gamma_{i+1}^+} Q_{out,j+i+1}^M$$

It is promptly verified that (9.31) holds true. *Case (iv)*. As before Zone $i+1$ is operating at Mode 1, with the inflow set to:

$$Q_{in,i+1} = Q_{out,i+1}^M + \sum_{j=1}^{\gamma_{i+1}^-} Q_{out,i+1-j}^M + \sum_{j=1}^{\gamma_{i+1}^+} Q_{out,j+i+1}^M$$

Zone i fulfills (9.20) because it can be in Mode 6, 9 or 11, and hence (9.31) holds. *Case (v)* In this case, Zone i can operate in Modes 1,4 or 8, whereas Zone $i+1$ in Mode 4, 8 or 10. By hypothesis, the former gives $Q_{i,i+1} = \sum_{j=1}^{\gamma_i^+} Q_{out,i+j}^M = Q_{out,i+1}^M$, as $\gamma_{i+1}^+ = 0$.

Zone $i + 1$ in Mode 4, 8 or 10 gives $Q_{out,i+1} = Q_{out,i+1}^M$ and $Q_{in,i+1} = 0$, so that $Q_{i+1,i+2} = Q_{i+2,i+1} = 0$, which is precisely (9.31) or (9.32). *Case (vi)*. From (9.39),

no air is exchanged between zone i and $i+1$. Furthermore, $\sigma_{i+2,i+1}^{(i+1)} \cdot \sigma_{i+2,i+1}^{(i+2)} = 1$ gives that Zone $i + 1$ can be in Mode 6, 9, 10 or 11. At a second sight, we can rule out Mode 10, as Zone $i+1$ is in Mode 10 only if $\sigma_{i,i+1}^{(i)} \sigma_{i,i+1}^{(i+1)} = 1$, which contradicts (9.39). Zone $i + 1$ being in one of these feasible modes gives $Q_{i+2,i+1} = Q_{out,i+1}^M$, which proves (9.33), as $\gamma_{i+2}^- = 1$. *Case (vii)*. Here air flows from zone i to $i + 1$ and from $i + 1$ to

$i + 2$. In particular, Zone i can be in either one of Modes 1,4, 8, whereas Zone $i + 1$ can be either in Mode 4 or 10, with $Q_{out,i+1} = Q_{out,i+1}^M$. Being $\gamma_i^+ = \gamma_{i+1}^+ + 1$, we conclude that

$$Q_{i+1,i+2} = \sum_{j=1}^{\gamma_{i+1}^+} Q_{out,i+1+j}^M$$

with $\gamma_{i+1}^+ = (\gamma_{i+2}^+ + 1)\sigma_{i+1,i+2}^{(i+1)}\sigma_{i+1,i+2}^{(i+2)} = \gamma_{i+2}^+ + 1$, that is (9.31). *Case (viii)*. This

is the exact opposite case of the previous one. Zone i is in Mode 6, 9, or 11, with $\gamma_{i+1}^- = \gamma_i^- + 1$. Zone $i + 1$ is heating up with warm air from zone $i + 2$ so it operates the fan at its maximum $Q_{out,i+1}^M$. In particular, it is in Mode 6 or 9, which implies that (9.33) is fulfilled. This concludes the proof.

Paper 8: Guided Controller Synthesis for Climate Controller Using UPPAAL TIGA

Jan J. Jessen[*]   Jacob I. Rasmussen[†]   Kim G. Larsen[†]
Alexandre David[†]

Abstract

In this paper, we present a complete tool chain for automatic controller synthesis using UPPAAL TIGA and Simulink. The tool chain is explored using an industrial case study for climate control in a pig stable. The problem is modeled as a game, and we use UPPAAL TIGA to automatically synthesize safe strategies that are transformed for input to Simulink, which is used to run simulations on the controller and generate code that can be executed in an actual pig stable provided by industrial partner Skov A/S. The model allows for guiding the synthesis process and generate different strategies that are compared through simulations.

10.1   Introduction

Inevitable parts in a traditional control design cycle are modelling, simulations and synthesis. Modelling often results in non-linear continuous models needing linearization and/or model order reduction in order to be applicable for control, while simulation can implement both original and linearized models. For control synthesis standard linear controllers are verified by design, but the control engineer still needs to perform the step of translating a mathematical description of the controller into an executable application

---

[*]Automation and Control, Aalborg University, Denmark, E-mail:jjj@control.aau.dk
[†]Department of Computer Science, Aalborg University, Denmark, E-mail:illum@cs.aau.dk
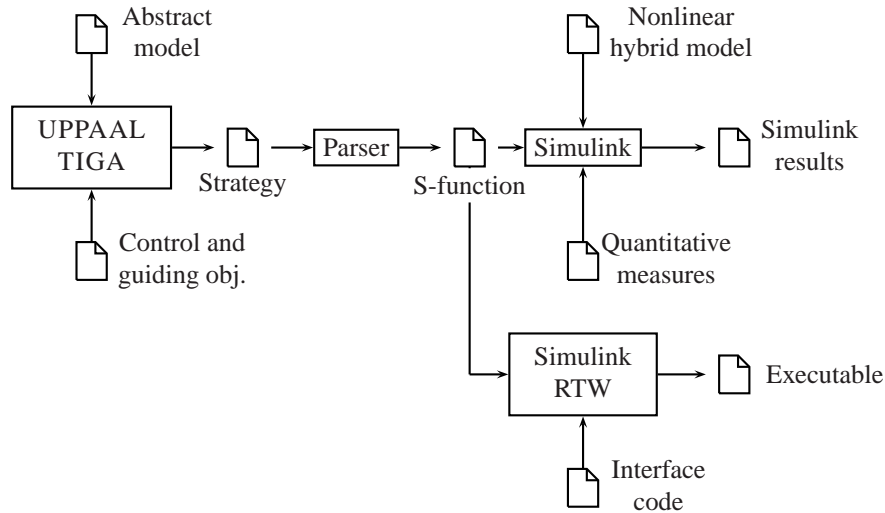
Figure 10.1: Illustration of tool chain for model based control.

that can be run on an embedded platform. Additionally, in the setting of hybrid models controller synthesis itself is a highly non-trivial task.

In this paper, we present a model-based framework for optimal control using the recently developed controller synthesis tool UPPAAL TIGA (Homepage, 2006) in combination with Simulink (SIMULINK, 2007) and Real-Time Workshop (Workshop, 2007) providing a complete tool chain for modeling, synthesis, simulation and automatic generation of production code (see Fig. 10.1). The framework requires that two models of the control problem are provided: An abstract model in terms of a timed game and a complete, dynamic model in terms of a (non-linear) hybrid system. Given the abstract (timed game) model together with logically formulated control and guiding objectives, UPPAAL TIGA automatically synthesizes a strategy which is directly compiled into an S-function representation of the controller. Now using Simulink together with the concrete (dynamic) model, simulation results for additional quantitative aspects of the synthesized controller can be obtained. Alternatively, given interface code for the specific actuators and sensors, Real-Time Workshop allows for the generation of production code implementing the synthesized controller.

The framework is presented through an industrial case study carried out in collaboration with the company Skov A/S specializing in climate control systems used for modern intensive animal production. For such systems it is of extreme importance that the climate control work properly, since a failure can result in the death of entire batches of animals and in turn loss of revenue for the farmer. In this context a *properly* functioning control system should additionally provide a comfortable environment for the animals.

142

In (Persis et al., 2006, 2007), a dynamic model for a pig stable that is both nonlinear and hybrid and a verified stable temperature controller has been presented. We show in this paper that our framework allows for automatic generation of the controller presented in (Persis et al., 2006, 2007), and moreover that our framework makes it straight forward to obtain and implement extended controllers, e.g. by including humidity control.

In Section 2 we describe a dynamic, zone-based climate model for the evolution of temperature in a pig stable. In Section 3 we briefly describe UPPAAL TIGA together with the notions of timed game, control objective and strategies. Section 4 is the main section giving a detailed description of how the climate controller is modelled and synthesized with UPPAAL TIGA. Numerical results are presented in Section 5, and conclusions are given in Section 6.

10.2   Climate Model

In this section, we introduce the dynamic climate model describing the evolution of temperature in a pig stable. The presented model is zone based, a concept where the pig stable is divided into distinct climatic zones, and where the zones interact by exchanging air flow. The idea is illustrated in Fig. 10.2 where a stable is partitioned into N subareas, and where the zones exchange air flow.
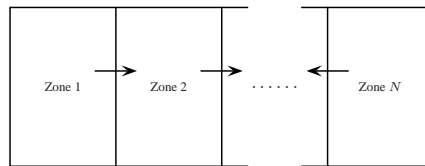

Figure 10.2: N zones.

Though it would be relevant to model temperature, humidity, CO2 and ammonia concentration we initially limit ourselves to modeling only temperature, in order to illustrate the zone concept. It would though be easy to include the disregarded climate parameters since the mixing dynamics are, roughly, identical.

Assumption 10.2.1 *Climatic interdependence between zones is assumed solely through internal air flow.*

With assumption 10.2.1 we thus neglect radiation and diffusion etc. between zones, claiming they are negligible compared to the effect from having internal air flow. Besides internal air flow a zone interact with the ambient environment by activating a ventilator in an exhaust pipe and consequently opening a screen to let fresh air into the building. Air flowing from outside into the ith zone is denoted Q_i^{in} [m³/s], from inside to outside Q_i^{fan} [m³/s]. Air flowing from zone i to $i+1$ is denoted $Q_{i,i+1}$ [m³/s] (air flow is defined

positive from a lower index to a higher index). A stationary flow balance for each zone i is found:

$$Q_{i-1,i} + Q_i^{\text{in}} = Q_{i,i+1} + Q_i^{\text{fan}} \tag{10.1}$$

where by definition $Q_{0,1} = Q_{N,N+1} = 0$. The flow balance is illustrated in Fig. 10.3 using the following definitions: $[x]^+ \triangleq \max(0, x)$, $[x]^- \triangleq \min(0, x)$.
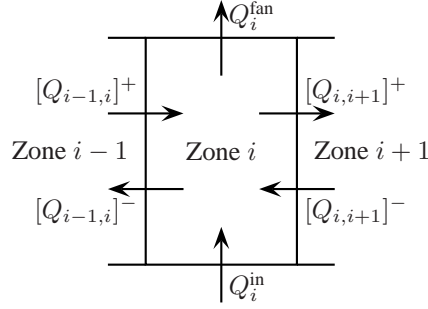

Figure 10.3: Illustration of flows for zone i.

In accordance with (Janssens et al., 2004; Arvanitis et al., 2006) and taking into account the flows leaving/entering the ith zone, the following model for temperature evolution is easily obtained.

$$\frac{dT_i}{dt} V_i = T^{\text{amb}} Q_i^{\text{in}} - T_i Q_i^{\text{fan}} + [Q_{i-1,i}]^+ T_{i-1} - [Q_{i-1,i}]^- T_i$$
$$- [Q_{i,i+1}]^+ T_i + [Q_{i,i+1}]^- T_{i+1} + \frac{u_i^t + W_i^t}{\rho_{\text{air}} c_{\text{air}}} \tag{10.2}$$

where V_i [m³] is the zone volume, T^{amb} [°C] is the ambient temperature, Q_i^{in}, Q_i^{fan} is the inflow and outflow respectively. c_{air} [J/(kg°C)] is the specific heat capacity of air, ρ_{air} [kg/m³] is the air density. u_i^t [J/s] is the controlled heating and W_i^t [J/s] is heat production from the pigs. For the actuator signals maximum values exists $Q_i^{\text{fan}} \in [0, Q_i^{\text{fan},M}]$, $Q_i^{\text{in}} \in [0, Q_i^{\text{in},M}]$, $u_i^t \in [0, u_i^{t,M}]$. The disturbance is not known but bounded $W_i^t \in [W_i^{t,m}, W_i^{t,M}]$.

In (Persis et al., 2006) a temperature controller for the model in (10.2) is presented. The presented controller is a multi-zone controller, i.e., it consists of N individual controllers. The controller is event-based, and only changes its control action when certain boundaries are met or a neighboring zone changes its control action. The controller in (Persis et al., 2006) is designed to solve a two player game theoretic problem following (Lygeros et al., 1999) at each time a state has changed or a change in coordinating variables take place. Each controller maintains a set of coordinating variables δ^i that holds information about the controllers willingness to exchange air flow with the neighboring

144

zones, and only if two neighboring zones agree to the exchange, air will flow between the zones.

The control actions available to controller is the heating u_i^t, opening of the inlets Q_i^{in} and turning on the ventilators Q_i^{in}. The controller has two "modes" heating up and cooling down, and an initial mode set to either one. We remark specifically that opening of the inlet is not enough to force air into the zone. This being a physical system air has to be removed either by operating the ventilator or by having a neighboring zone extract air. The controller operation in zone i is as follows: When heating up ventilation is closed and heating is turned on. If in addition a neighbor zone has warmer air than in the current zone, the controller will inform the neighboring zone's controller that it would like to receive the warmer air. Only if the two zones agree to exchange air will the controller in zone i turn on its ventilation fan extracting warm air from one of the neighbor zones. When cooling down, the heating is turned off, the inlets opened and the ventilation fan is turned on. The controller will in addition inform the two neighbor zones, that it would like to "give away" air thus forcing more fresh air into the zone.

10.3 Timed Games, Control Objectives and Strategies

UPPAAL TIGA is a tool for solving control problems modeled as (networks of) timed game automata (Homepage, 2006). As an example consider the control problem in Fig. 10.4, where a central controller C is to maintain the temperature of two tanks, T_1 and T_2 above some critical minimum level, say $5°$C.

Each tank is modelled as a timed game automaton with location `High` indicating that the temperature in the tank is between $80°$C and $100°$C. Similarly, the `Low` locations indicate a temperature between $10°$C and $15°$C and the `Critical` locations that temperature is below $5°$C. The controller C has the possibility for heating either tank thus lifting (or maintaining) its temperature to the `High` level; the act of heating is modelled as synchronizations on the channels h_1 and h_2. The guards $\mathrm{z} \geq 1$ on the clock z of the controller enforces that heating actions of C are seperated by at least 1 time-unit. The dashed edges in the two tanks represent *uncontrollable* transitions for lowering the temperature (from `High` to `Low` and from `Low` to `Critical`) in a tank in case no heating action of the controller has taken place for a certain time period; e.g. the guard $3 \leq \mathrm{x} \wedge \mathrm{x} \leq 4$ indicates that the temperature in T_1 may drop from `High` to `Low` at any moment between 3 and 4 time-units since the last heating of the tank.

Control purposes are formulated as "`control:  P`", where `P` is a TCTL formula specifying either a safety property, (`A[ ]` φ) or a liveness property (`A<>` φ). Given a control purpose, "`control:  P`", the search engine of UPPAAL TIGA will provide a *strategy* (if any such exists) for the controller under which the behaviour of the model will satisfy `P`. Here a strategy is a function that in any given state of the game informs the controller what to do either in terms of "performing a controllable action" or to "delay". In our tank example of Fig. 10.4 the control purpose may be formulated as "`control: A[] not(`T_1`.Critical or `T_2`.Critical)`". Endeed there is a strategy guaran-
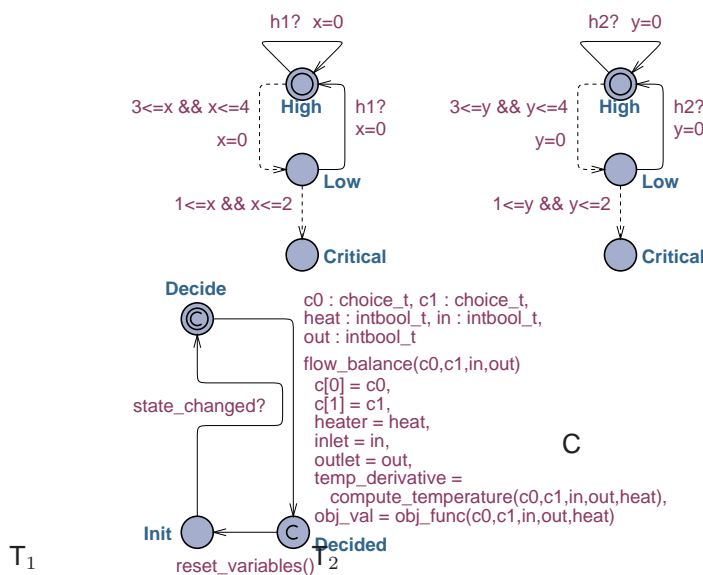
Figure 10.4: Two Tank Temperature Control Problem.

teeing the safety property involved (i.e., the `Critical` temperature level is avoided in both tanks). In the case when the two tanks are both having a `Low` temperature level the strategy provided by UPPAAL TIGA requests the controller to heat T_2 whenever $(2 < y \wedge 1 < z \wedge y \leq x) \vee ((2 < x \wedge 1 < z) \wedge (y < 1 \vee x < y))$. In case $(2 < y \wedge 1 < z \wedge x < 1)$ the strategy suggest to heat T_1. Interestingly, it may be shown (as discovered by UPPAAL TIGA), that for slower controllers (e.g. replacing the guards $z \geq 1$ by $z \geq 2$) no strategy exists which will ensure our control purpose.

UPPAAL TIGA is integrated in the UPPAAL 4.0 framework permitting the use of discrete (shared or global) variables over simple or structured types (arrays and recods) including user-defined types. Functions can be declared using C-like syntax and used in guards and update statements. Edges have an additional select statement as a shorthand notation for all edges that satisfy the statement.

10.4   Modelling

In this section, we give a detailed description of how the climate controller has been modelled in UPPAAL TIGA. We divide the description into a model section and a property section with guiding.

10.4.1 The Models

The compound model consists of three kinds of automata, the neighbor automaton, an auxilliary automaton, and controller automaton. Each of these are described in turn in the following.

Neighbor Automaton

The neighbor model is an automaton with just uncontrollable transitions that can change the observable variables of the neighboring zone. The template for the neighbor automaton is depicted in Fig. 10.5 and is instantiated with a parameter `id` which can take the values 0 and 1 to indicate the left and right neighbor.
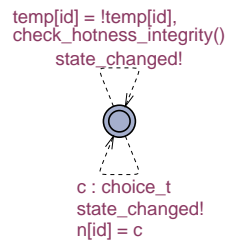

Figure 10.5: Neighbor Automaton.

Each neighbor has a variable `temp` that discretizes the temperature information of the neighbor to either `HOTTER` or `COLDER` than the control zone. Furthermore, there is a variable `n` that holds the values of the interaction variables of the neighbor. The variable n, which can take any of the values `WANT`, `HAVE` and `NEITHER` (encoded as the type `choice_t`), is used to indicate whether the neighbor wants air flow from the control zone, wants to deliver air flow to the control zone, or does not want to exchange air flow with the control zone. To switch the temperature of a neighboring zone, the environment can take the uncontrollable transition at the top of Fig. 10.5. The call to the fucntion `check_hotness_integrity()` on the transition is explained below. The bottom transition uses special UPPAAL TIGA syntax for select statements. This is shorthand notation for the three cases where c takes on any of the values of `choice_t`, i.e., the environment can set the control variables of the neighbor to any kind of desired interaction. Whenever the environment changes an observable variable it synchronizes over the channel `state_changed` with the controller, to allow the controller to change the control strategy. This way we keep a strictly alternating game where the controller reacts every time an observable variable changes value.

147

Auxiliary Automaton

To manage the other observable variables, we introduce an auxiliary automaton that allows the environment to change these variables. The auxiliary automaton is depicted in Fig. 10.6.
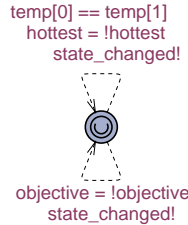

Figure 10.6: Auxiliary Automaton.

The final two observable variables that can change are, first, the variable `objective` which determine whether the control zone should `HEATUP` or `COOLDOWN` (bottom transition of the automaton). The second variable is a result of the discretization of the temperature information. The control zone needs information about which neighbor has hotter air. This is encoded using the Boolean observable variable `hottest` where value 0 indicates the left neighbor is hotter and vice versa for value 1. The environment can change the value of `hottest` on the top transition only when either both zones are either colder or hotter, otherwise the value can become inconsistent with the temperatures of the neighbors. The function call `check_hotness_integrity` is used by the neighbor automaton whenever the temperature changes to guarantee that `hottest` is left in a consistent state.

Controller Automaton

The controller automaton synchronizes with the auxiliary automaton and neighbor automata over the channel `state_changed` whenever an observable variable changes values. Upon synchronization, the controller enters the committed state `Decide` and the setting of the control variables is determined on the transition exiting `Decide`. The controller automaton is depicted in Fig. 10.7

The controller automaton determines the value of five control variables: Two variables for the interaction with the neighbors (`c[0]` and `c[1]`) and one variable for each of the `heater`, `inlet` and `outlet` (the latter three are all Boolean variables). The selection statement on the transition from `Decide` to `Decided` guarantees that all possible settings are considered. The guard statement `flow_balance` guarantees that no inconsistent control state wrt. air flow is considered, i.e., whenever air is flowing out of a zone, air is flowing into the zone (see Algorithm 5) and vice versa. After
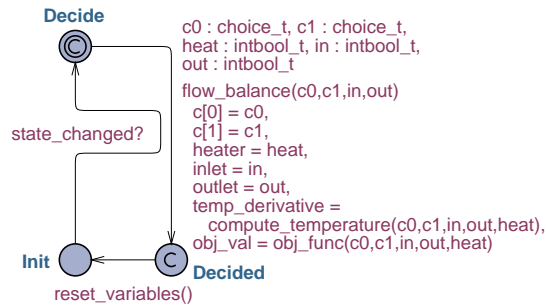
Figure 10.7: Controller Automaton.

updating the control variables the combined impact of the control variables and the observable variables on the temperature of the control zone is computed using the function `compute_temperature`. We refer to `obj_func` later, when we talk about guiding.

Upon entering the committed location `Decided` the transition back to `Init` is taken immediately which resets all the control variables. This is merely a step to minimize the state space since, as we shall see, the effect of the control decision is only important in `Decided`.

Algorithm 5 Procedure to guarantee that the flow balance is satisfied.

---

proc flow_balance(c0,c1,in,out) :   bool

1: bool o = out || (n[0]==WANT && c0==HAVE) || (n[1]==WANT && c1==HAVE)
2: bool i = in || (n[0]==HAVE && c0==WANT) || (n[1]==HAVE && c1==WANT)
3: **return** o == i

Discretization of the Temperature Derivative

Since the model is discretized such that the controller does not know the exact temperature of the neighboring zones, this needs to be reflected in the computed temperature derivative.

We choose to let the different control parameters contribute to the temperature derivative according to the table to the right. The values for the airflows correspond to opening the outlet fan and getting air only from the specific source. Given that the fan capacities are fixed, getting air from multiple sources will share the capacity. E.g., getting air from both (hotter) neighbors

Heater:	5
Inlet:	-7
Hotter neighbor	
- hottest:	2
- coldest:	1
Colder neighbor	
- hottest:	1
- coldest:	2

would yield a contribution of 1 from the hottest and 0.5 from
the coldest, resulting in a total contribution of 1.5. Furthermore,
having multiple sources of outflow increases the inflow contribution proportionally, e.g.,
allowing the inlet to give a total contribution of -21 by opening the outlet and providing
air for both neighbors.

Computing the Temperature Derivative

As we saw above, the fans have a fixed capacity that might be shared among the different
sources of outflow. Since this can result in a non-integral contribution and UPPAAL
TIGA only handles integers, we need to multiply these contributions with an apropriate
factor to guarantee integral values. Since a single source of outflow can be shared among
up to three sources of inflow, we choose a constant `OUT_CONTRIBUTION=6` to denote
the available contribution per outflow source as this can be integrally shared among the
potential inflow sources. This has the added effect that we need to multiply the heater
contribution by six as well, to keep the proportions.

The function for computing the temperature derivative is listed in Algorithm 6. Lines
1 and 2 compute the contributors to air flow in and out of the zone. For outflow, this,
in order, corresponds to 1) is the outlet open, 2) is air flowing from the control zone to
the left zone, and 3) similarly for the right zone. The computation is analogous for air
flowing into the control zone.

The value of `amp` computed in line 3 is the contribution for each inflow given the
total outflow. Now, the return statement computes the total effect of the control decision
by using the table above and the amplifier for each inflow contribution. Note that the
heat contribution is also amplified to keep the the proportions defined above.

The final two negative parts of the contribution are used to indicate that giving air
away cools the zone. These are used as incentives to let the controller offer air when it
wants to cool. The reason is that when the controller is used in all zones we can imagine
the situation when one zone needs to cool and a neighbor want the air to heat up. In
the control situation when neither are interacting, one of the zones need to intiate the
cooperation, and this is accomplished with the given incentives. Note that these values
are negligible in the overall contribution.

10.4.2   The Property

In order to synthesize the controller, we need to specify the property that the resulting
controller should synthesize. An immediate choice would be[1]:

$$
\begin{aligned}
\phi \equiv \texttt{control} : \quad &\texttt{A[]Controller.Decided imply} \\
&\texttt{(objective ? 1 : -1)*temp\_derivative > 0}
\end{aligned}
\tag{10.3}
$$

---

[1]Recall that we switched the sign of the temperature derivative when the objective is to cool down.

Algorithm 6 Algorithm for computing the temperature derivative.

```
proc compute_temperature(c0,c1,in,out,heat) :
int
 1: int outflow = out + (c0==HAVE && n[0]==WANT)+(c1==HAVE
    && n[1]==WANT)
 2: int inflow = in + (c0==WANT && n[0]==HAVE)+(c1==WANT &&
    n[1]==HAVE)
 3: int amp = (outflow * OUT_CONTRIBUTION) / inflow
 4: return OUT_CONTRIBUTION*5*heat
    + amp*(c0==WANT && n[0]==HAVE ? (temp[0]?  (!hottest?
    2:1) :  ( hottest?  -2:-1)) :  0)
    + amp*(c1==WANT && n[1]==HAVE ? (temp[1]?  ( hottest?
    2:1) :  (!hottest?  -2:-1)) :  0)
    + amp*(in ?  -7 :  0)
    - (c0==HAVE) - (c1==HAVE)
```

In other words, invariantly whenever the controller enters `Decided`, the value of `temp_derivative` should be greater than zero when heating is the objective and less than zero when the objective is cooling. However, this property would be satisfied by the simple controller that never interacts with the neighbors and turn on the heater when the objective is heating and opens the inlet and outlet when the objective is cooling.

Guiding

With the property above we can determine whether we can satisfy the main objective or not. Now, we define an objective function called `obj_func` that will guide the controller synthesis process while also satisfying the property above[2]. Given an appropriate objective function, the following property can be used to guide the controller synthesis process:

```
φ ≡ control :  A[] ZC.Decided imply forall (c0 :  choice_t)
      forall (c1 :  choice_t) forall (in :  intbool_t)
      forall (out :  intbool_t) forall (heat :  intbool_t)
      flow_balance(c0,c1,in,out) imply
    obj_val >= obj_func(c0,c1,in,out,heat)
```

In plain words, the property states that it should hold invariantly that whenever the controller makes a decision and enters the location `Decided`, then for all other possible controller choices that satisfy the flow balance, the computed objective function is

---

[2]To satisfy both properties we use conjunction, but do not include the conjuction to simplify properties.

smaller or equal to the choice made. In short, the controller always chooses a configuration of the control variables that maximizes `obj_func` among all valid choices.

The simplest objective function is to use `compute_temperature`, but to compensate for the sign depending of the objective as (10.3) above. This guiding proccess will produce a controller that maximizes (minimizes) the temperature derivative for every control decision. An alternate strategy is to define the objective function over some sort of energy consumption by, e.g., penelizing turning on the heater or fan, thus, optimizing towards energy optimality.

10.4.3 Controlling Humidity

As mentioned in (Persis et al., 2006), the climate controller should, ideally, be extended with the ability to control the humidity in the stable as well. However, the approach outlined in (Persis et al., 2006) makes this extension a tedious strategy, since the increase in oberservable variables creates exponentially more configurations.

Changing our model to accommodate for humidity as well, requires a slight modification to the models along the lines of how the temperature was modelled. Furthermore, the objective function needs to represent the effect of temperature and humidity with a simple value. Note, that neither the controller automaton nor the property changes, as the set of controllable variables remains unchanged.

To discretize the humidity readings of the neighboring zones, analogously to the temperature representation, we introduce a Boolean `humid` variable for each zone and a `morehumid` variable to determine which of the neighbors have air with the highest humidity. Obviously, there are constraints on consistent variable assignments for the three variables in the same way as for the temperature variables.

To incorporate the variables in the model, we need an extra uncontrollable transition in the neighbor automaton, that can change the value of the respective `humid` variable. This is followed by a consistency check on the `morehumid` variable. Moreover, we add two extra uncontrollable transitions to the auxiliary automaton, one to change the `more_humid` variable, and one to change the objective with respect to humidity which is encoded in the variable `decrease_humidity`. These are all the changes needed to the automata.

We incorporate humidity information in the objective function in a similar fashion the temperature model with the exception that humidity only has an upper limit, so the objective is either to decrease the humidity or ignore the humidity. Thus, when `decrease_humidity` has value false, the humidity contributes nothing to the objective function. Otherwise, the contribution is given as in Algorithm 7 where a postive value indicates a decrease in humidity. In the algorithm, `amp` is computed as for the temperature contribution. The positive contributions are from opening the inlet (contribution of 5) and getting less humid air (contribution of 2 for the least humid air and 1 for the most humid). Receiving more humid air from a neighboring zones contributes negatively. Finally, we encourage a zone to interact, if a neighboring zone has less humid

air, even if the zone does not want to interact (first two parts of the sum).

Algorithm 7 Humidity contribution to the objective function

```
1: return (!humid[0] && c0 == WANT ? 1 :  0) + (!humid[1] &&
   c1 == WANT ? 1 :  0)
   + amp*(c0==WANT && n[0]==HAVE ? (humid[0] ?
   (!morehumid?-2:-1):( morehumid?2:1)) :0)
   + amp*(c1==WANT && n[1]==HAVE ? (humid[1] ?  (
   morehumid?-2:-1):(!morehumid?2:1)) :0)
   + amp*(5*in)
```

The objective function is constructed with a weight parameter between the temperature derivative and the humidity parameter with changing the sign of the temperature as above. The weighing can be altered to generate different controllers, which later can be compared in some appropriate fashion as discussed in Section 10.5.

10.5 Results

In this section, we present some numerical results where the controller generated by UPPAAL TIGA has been simulated in Simulink using realistic values for the model (10.2).

The Tool Chain

According to Fig. 10.1, to generate production code for the climate controller of the pig stable, we need to transform the output format of UPPAAL TIGA to input for Simulink. Simulink allows input of so-called S-functions which are user provided C-code that can be used within the Simulink model. We have build a script which takes UPPAAL TIGA strategies as input and delivers S-functions as output. The Simulink model with the S-function can be used to either run simulations of the pig stable, or generate Comedi compliant production code through Real-Time Workshop. The code generation is realized through a Comedi library for Simulink (Jessen et al., 2006b).

Numerical Results

We have synthesized two types of controllers using the models described above. One controlling only the temperature, and one controlling both temperature and humidity. In the first experiment, we synthesized a controller for temperature only as explained in Section 10.4. Due to limited space, we choose not to include the graphs for the experiments, as the synthesized controller is identical to that of, (Persis et al., 2006, 2007). As in (Persis et al., 2006, 2007), the controller behaves well under simulation and keeps the zone temperatures within the given bounds.
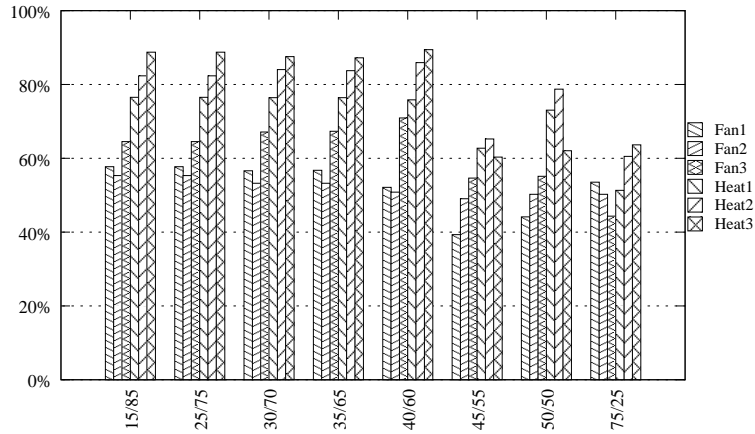
153

Figure 10.8: Active time for heaters and fans for different controllers. The number (x/y) indicate an objective function using x percent temerature contribution and y percent humidity contribution.

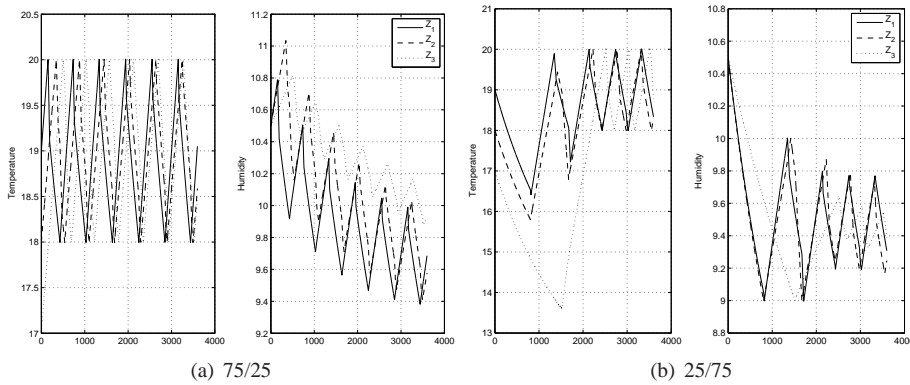(a) 75/25   (b) 25/75

Figure 10.9: Simulation results for temperature and humidity when guiding towards a) 75% temperature and 25% humidity and b) vice versa.

In order to illustrate the guiding specification in UPPAAL TIGA a number of different controllers are simulated in Simulink. A weight is put on the objective function guiding towards temperature or humidity control. The simulation scenario is as follows: The stable is partitioned into 3 zones, and the thermal boundary is set to $[18\ 20]$ and for humidity $[9\ 10]$ for all three zones. The initial conditions are set to $T_1 = 19$, $T_2 = 18$ and $T_3 = 17$, $H_1 = H_2 = H_3 = 11$. All the conducted experiments stear the state to the defined boundaries in finite time, but initially some states are steared away from

154

the boundary. In order to quantify and compare the different controllers the total time when the heat or fan are on is recorded. The result is illustrated in Fig. 10.8. The results show, that the controllers can be divided up into two catagories, one from 0% to 40% temperature guided, and one from 45% to 100%. The controllers in the latter category use less heat and fan capacity than the controllers in the former catagory, indicating that the former are preferred controllers. However, Fig. 10.9 shows how the temperature and humidity are controlled for controllers in both catagories. As it can be seen, the controller with more heat and fan activation (25/75) reaches a stable state faster than the controller with less activity of heaters and fans.[3] Thus, the choice between the controllers is not immediately clear, but the quatifications can be used by the control engineers to make an informed choice.

10.6 Conclusions and Future Work

In this paper, we have presented a complete tool chain for automatic controller synthesis from timed game automata models to production code. For the livestock production case study, the controller synthesis process has enabled, through guiding, to synthesize an identical controller do that of (Persis et al., 2006, 2007). The controller in (Persis et al., 2006, 2007) was synthesized in a tedious manual way, which indicates the importance of a simple automated process. Note that the notion of time was not necessary in modelling our controller, however, we choose UPPAAL TIGA because the tool was available and one the only ones of it's kind.

Furthermore, the model was easily extended to include humidity, which was left as a matter to explore in (Persis et al., 2006, 2007), but never pursued due to the heavy time requirement of the added exponential complexity. With an appropriately defined weighted objective function, UPPAAL TIGA was used to synthesize a controller capable of regulating temperature as well as humidity in a matter of seconds. A number of controllers were synthesized with varying weights between temperature and humidity, and all were able to reach stable temperature and humidity conditions in Simulink simulations. Simulink was further used to track the heat and fan activity for the different controllers, in order to allow for comparison of different controllers. This can be a very effective strategy for differentiating controllers and choosing an appropriate one among a number of controllers satisfying the conditions. As future work, we want to continue conducting experiments in the real life pig stable provided by Skov A/S in order to evaluate the different controllers capacity of controlling temperature as well as humidity in a real life setting.

---

[3]Simulation results for all controllers can be found at the project website, (Web, 2007).

Experimental Results

An experimental facility has been established in the northern part of Jutland - see (Jessen et al., 2006a,b). This chapter presents results from experiments conducted at this facility.

11.1 Game Controller Temperature Results

As described in the enclosed articles the stable is partitioned into three zones, each having individual ventilation components. There are no animals in the stable, instead a central boiler is used to generate heat in the three zones. There are no sensors to quantify the amount of heat delivered to the three zones, only the position of the valves controlling the amount of water to the radiators are known. The conducted experiments are summarized in table 11.1.

Exp	Fan	Pig	Heat	Zone 1	Zone 2	Zone 3
1	1-1-1	0-0-0	10-10-10	8-10	8-10	8-10
2	1-1-1	0-0-0	10-10-10	10-12	10-12	10-12
3	1-2-1	4-4-4	6-6-6	14-16	12-14	14-16

Table 11.1: Configurations for conducted experiments.

There are five ventilation outlets in the experimental facility, and in (Jessen et al., 2006b) it is illustrated that fan 2 and fan 4 are at the zone boundary between zone 1 (3) to zone 2. The fan column in table 11.1 refers to the configuration of fans for the individual zones where 1-1-1 means that fan 2 and fan 4 are inactive in the experiment. Only the valve position is known, the pig column, therefore, refers to the constant offset in valve position for the three zones that represent the pigs' heat in the stable. A value of 0, thus, means "no pigs" present. The valve controlling the amount of water to the

radiators can be set in an arbitrary position between 0 (closed) and 90 (fully open). From the computer controlling the actuators and as described in (Jessen et al., 2006b) a value is set from Matlab between 0 and 10 giving a linear relation to the opening degree. Both the Pig and Heat column refer to this number.

Following (Persis et al., 2006, 2007) heating should be set to maximum whenever the controller has to heat up. When the fan is operated in a zone it is always set to maximum speed attracting either warm air from a neighboring zone (to heat up) or cold air from outside (to cool down). The inlets should be set so that the required flow balance is fulfilled, but how this is accomplished in practice is not clear. Local controllers are implemented around the inlets taking a reference value between 0 and 10 giving a linear relation between closed and fully open (Jessen et al., 2006b). In all the experiments inlets are set to 5 when they only deliver cold ambient to the zone in which they are placed. If more than one zone needs air, inlets are set to a value of 10. The sampling interval is set to 1 [min].

11.1.1   Experiment 1

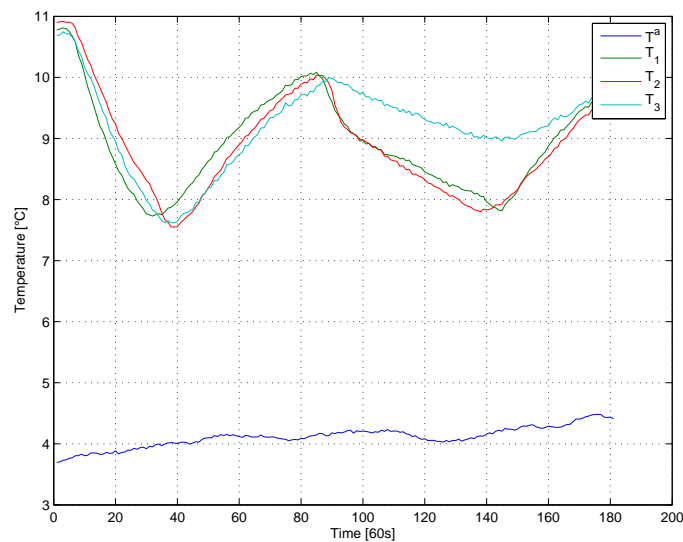Figure 11.1 illustrates the recorded temperature in the three zones for the configuration in table 11.1.

Figure 11.1: Experiment 1. Zone temperature and ambient temperature.

In this experiment the temperature interval is set to $[8 \quad 10]$ °C for the three zones. As figure 11.1 illustrates, the controller seems to fulfill the qualitative requirements: 1) The temperature oscillates between the upper and lower bound, and 2) The temperatures in the three zones are "close" to the *thermal region*. Figure 11.1 shows that there is an undershot when hitting the lower bound. To inspect this in more detail figure 11.2 illustrates a zoom on the time interval $[20 \quad 40]$ for temperature (T), inlet (I) and heating (H) for zone 1, 2 and 3.
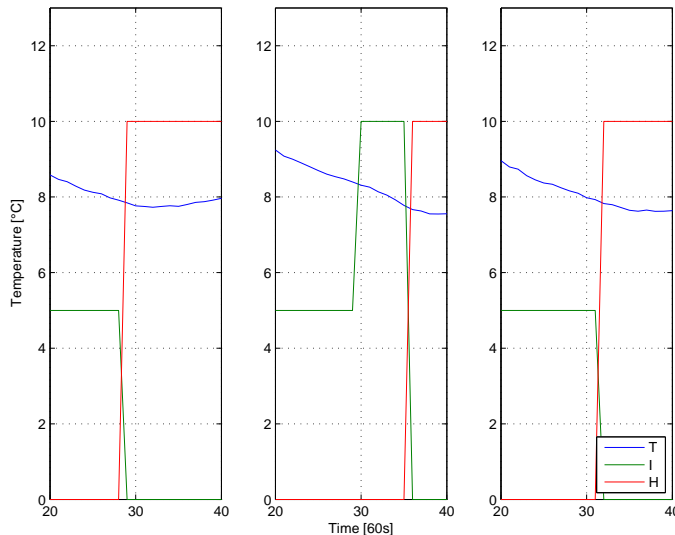

Figure 11.2: Experiment 1. Zoom on time interval from figure 11.1.

As seen in figure 11.2 there is a small delay after the lower bound is hit before heating is turned on. This is caused by the implementation of the controller, which causes a 2 sample delay from measurement to actuation, which explains the undershot. After the lower bound is hit all three zones are heated up, and both zone 1 and zone 2 consequently hit the upper bound and then start to cool down again. In figure 11.1 it is seen that zone 3 does not reach the upper bound but still cools down after approximately 85 minutes. To investigate this in more detail figure 11.3 illustrates a zoom on the time interval $[75 \quad 100]$.

As figure 11.3, shows the temperature in zone 3 "just misses" to hit the upper bound, and after zone 1 and zone 2 are cooled down heating is apparently inadequate to raise the temperature.
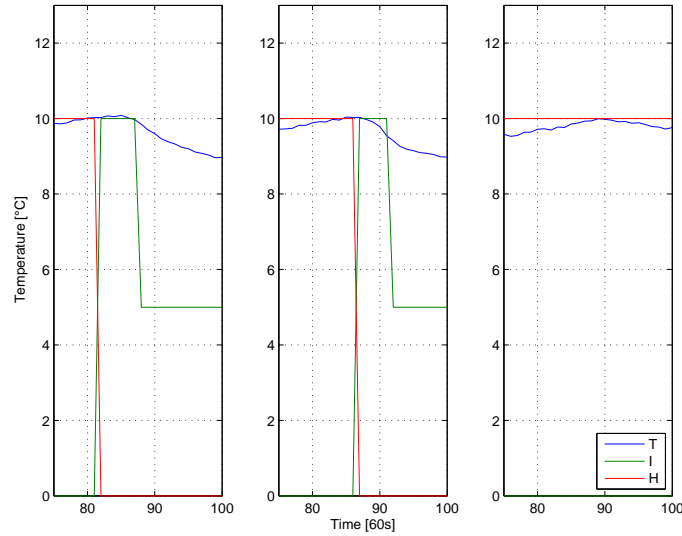
Figure 11.3: Experiment 1. Second zoom on time interval from figure 11.1.

11.1.2 Experiment 2

Experiment 2 is a replica of the configuration in experiment 1, except that the specified temperature interval is changed to $[10 \quad 12]\,^\circ$C. The resulting temperatures for the three zones are illustrated in figure 11.4.

Experiment 2 illustrates the same kind of qualitative behavior from the controller as seen in experiment 1. Figure 11.5 presents a zoom of figure 11.4 from the first time the temperatures are close to the specified lower bound. The three graphs in figure 11.5 are from zone 1, 2 and 3 respectively. The second and third graphs illustrate that approximately after 45 minutes, T_2 is greater than T_3 while the controller in zone 2 is trying to cool down and the controller in zone 3 is trying to heat up. The inlets in zone 2 are consequently opened fully. As the temperature in zone 2 hits the lower bound heating takes place with warm air from zone 1. As seen in the first graph in figure 11.5 the temperature in zone 1 "just misses" to hit the lower bound, and after both zone 2 and zone 3 are heated up the controller in zone 1 is unable to cool down. Zone 1 is actually trying to cool down even though the temperature is rising.
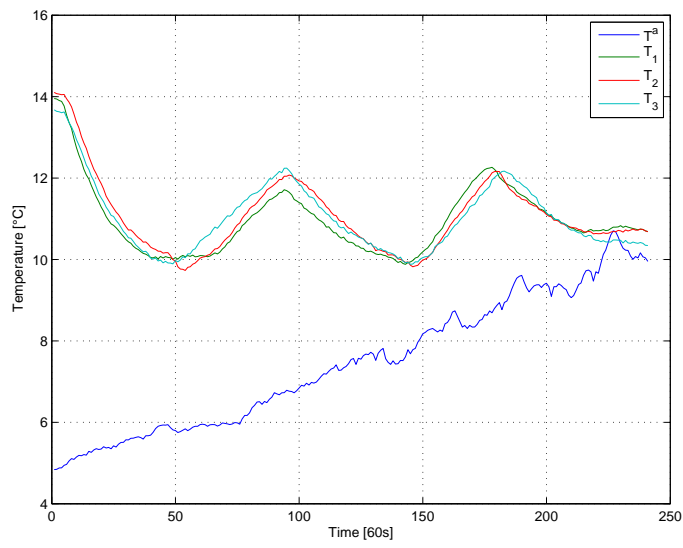
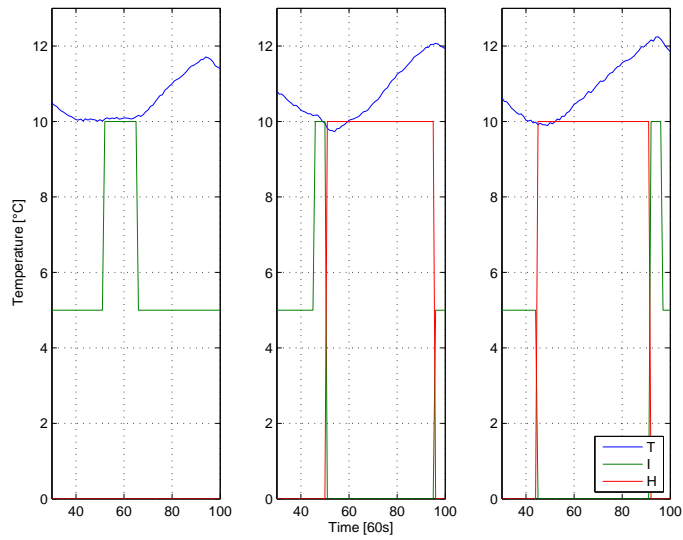Figure 11.4: Experiment 2. Zone temperature and ambient temperature.

Figure 11.5: Experiment 2. Zoom on actuators for first cross with T_m.

161

11.1.3   Experiment 3

In experiment 3 the pigs are set to a constant offset of 4 in all three zones. Subsequently, when heating is turned on it is set to 6, thus, having maximum heating capacity. Figure 11.6 illustrates the outcome of the experiment.
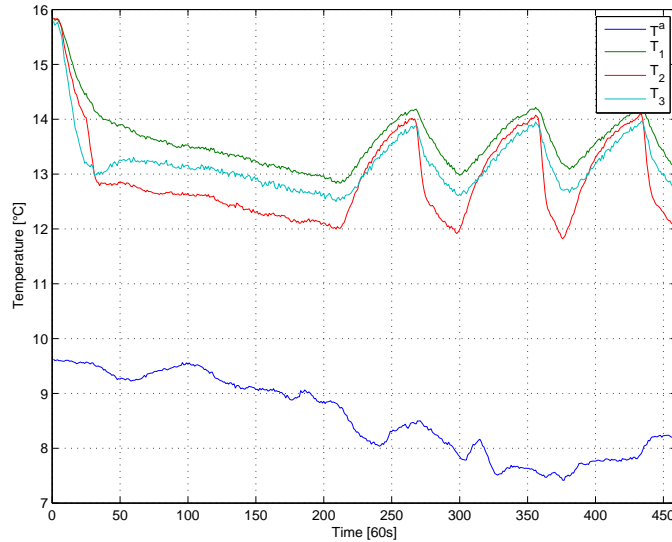
Figure 11.6: Experiment 3. Zone temperature and ambient temperature.

As seen on figure 11.6 zone 1 and zone 3 are unable to maintain the temperature within the thermal region, in fact both zones are trying to heat up during the entire experiment. Zone 2 on the other hand shows the desired behaviour, and from inspecting figure 11.6 it is clear that the controller in zone 2 governs the behavior in zone 1 and zone 3. Though zone 2 influences zone 1 and zone 3, it is clear that it *is* possible to maintain different temperature in the zones, but the requirements given in table 11.1 can not be fulfilled.

11.1.4   Discussion of Game Controller Temperature Results

The temperature controller presented in (Persis et al., 2006, 2007) was originally motivated by the following:

- Zone based control requires more sensors

- Installation costs are high
  - Many safety requirements for cabling

- Battery powered wireless sensors

- Send limited information

- Event based control

The sought solution of the controller is illustrated in figure 11.7 where the red circles illustrate a change in control action.
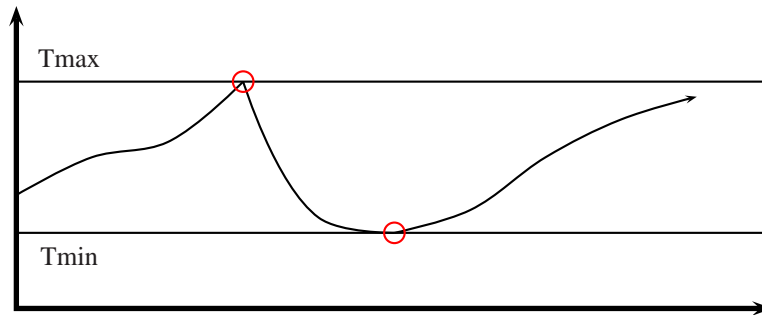

Figure 11.7: Wanted solution for game temperature controller.

Though not all aspects of the presented controller have been verified with experimental results, the presented results indicate that the devised temperature controller has the qualitative property as sought in figure 11.7. It is, however, clear from the experiments, that a total separation of thermal regions is not possible with the current configuration in the experimental facility. The conducted experiments indicate that there are some unmodeled dynamics that influence the experiments. A basic assumption in the enclosed articles is that zones only affect each other by the presence of internal air flow. But from the presented experiments this can not be verified or proved wrong. The key issue here is the presence of internal air flow, for which no detector has been available. This means that for a given configuration of fan and inlet settings it is not known if the desired flow balance actually is fulfilled.

The experiments indicates a strong synchronization for the three zones. This synchronization is caused by the fact that there is not enough heating capacity available in the experimental facility. At present it is not possible to conclude if this phenomena will occur if implemented in a real pig stable.

TWELVE

Conclusion and Recommendations

In this thesis various aspects related to climate control in livestock buildings have been presented along with a survey on literature within climate control/modelling. The enclosed articles cover both development environment for new climate control algorithms, early ideas for performing system identification and a game theoretic approach to climate control.

12.1   Summary of Contributions

- In (Jessen et al., 2006b) it was demonstrated how to build a future climate control system based on readily available hardware and software components (COTS). An experimental facility has been established as a showcase of what a future system can offer. This includes a complete tool chain from simulations using Simulink to actual implementation on the target platform using Real-Time Workshop. The development of the controller also allows remote monitoring and control independent of the client hardware e.g. a mobile phone or PDA (Jessen et al., 2006a).

- A novel model for climate dynamics has been introduced based on the concept of zones. The model is considered both nonlinear and hybrid, and is based on air flow in 2 dimensions, and is as such a simplification of actual flow inside a stable. In (Jessen and Schiøler, 2006c) a simulation algorithm using realistic actuator signals is presented, along with a technique to build a graphical modelling interface in simulink for zone based climate dynamics.

- Under the assumption of well defined internal flows, it has been shown (Jessen and Schiøler, 2006a) how to perform parameter estimation of a hybrid dynamical model. A simple signal vector construction is presented that easily allows inclusion of e.g. humidity modelling or heat loss through the building envelope.

- A verified *safety controller* for temperature is presented yielding a closed loop hybrid and decentralized control law (Persis et al., 2006). The decentralized controller only uses information from neighboring zones and takes values in a finite set. The controller assumes that flows can be set so that the required flow balance is fulfilled, and it is explicitly proven that *all* decentralized controllers jointly fulfill the required flow balance.

- A complete tool chain from abstract modeling in UPPAAL TIGA to both simulations and production ready code via Real-Time Workshop in Simulink has been presented. The tool chain allows to do easy experiments of heuristics for controlling both temperature and humidity.

12.2   Conclusion

The very existence of the constructed experimental facility and the corresponding computer control of the actuators, proves that it is possible to use freely available software (e.g. Linux, Comedi, Apache, MySQL, PHP, ssh) and cheap hardware to build a computer system that can deliver the functionality expected of a future climate control system. In this thesis it was chosen to use Mathworks products as a case for the controller development cycle. But the use of Matlab and Simulink (and Real-Time Workshop) is not necessarily the best choice for developing future climate control systems. Matlab is an excellent tool for rapid prototyping of new algorithms, and with many build-in functions for e.g. control engineering it is often the first choice for solving a given task. Adding the fact that one byes Matlab, good support is naturally expected. However, two critical points should be considered: Price and modelling formalism. Matlab is reputed as being *very* expensive, and for smaller companies this cost could be too high compared to the benefits from using this tool. In the Linux and open source community a Matlab clone like Scilab (website, 2007c) is getting more and more mature with respect to stability, support and user community and with Scicos (website, 2007b) a complete tool chain exists that delivers the same functionality as Matlab/Simulink/RTW. With respect to modelling formalism, Simulink is restricted when modelling e.g. hybrid systems, since the top level of simulation always is in a time domain. With the StateFlow toolbox it is possible to model events, but the level hierarchy can not be changed e.g. as in Ptolemy II (website, 2007a).

The presented model for temperature dynamics was initially constructed with the ambition of having a *simple* model as basis for corresponding simple controllers. It has, however, not been possible to perform experiments to either validate or reject the proposed model. For control engineering purposes models should be *sufficiently accurate* meaning that it should be possible to construct a controller fulfilling certain objectives, based on the given model. The conducted experiments indicate that it is possible to maintain the temperature within the thermal zone, which supports the proposed model. It is, however, necessary to perform more experiments e.g. with a real stable in order

to conclude further on the proposed model. In respect of the experimental facility there are indications that the partitioning into three zones is too much, yielding a middle zone that simply is too small.

The parameter estimation technique was motivated by the idea of having a step response kind of experiment, giving as output all the possible model combinations of the dynamic climate model. Initially, an experiment was conducted leading to the parameters of the model for specific internal flow directions used in the experiment. Due to practical problems with the experimental facility, a second experiment with the same flow direction was never performed, which should have been used to validate the found parameters from the first experiment. Consequently, the step of deducing all possible dynamic models was never taken.

When this project was started it was envisaged that a future climate control system would use a wireless sensor network to inform about both climate as well as animal welfare conditions. This fostered the idea of sending limited information in order to save battery power, which again led to the idea of event based control. The game theoretic approach to temperature control proved to be a sound framework for developing a rule based controller capable of maintaining the temperature within the defined boundaries. But, a drawback of the presented approach is that it uses flow as a control signal instead of actuator signals. Due to limited available heating capacity, it has not been able to conclude if it will be possible to maintain different thermal regions in different zones. Testing in real life pig stables can reveal this. With respect to internal flow direction it is should be noted that controller takes values in a finite set. It should, therefore, be possible to verify that the desired flow directions are fulfilled using e.g. a smoke generator.

In the introduction it was stated that this thesis would only consider temperature and humidity, but humidity has in reality been disregarded. This thesis has primarily dealt with ideas and visions for future climate control systems, hence, the proposed techniques are still in a prototype phase. But as demonstrated in the COTS articles it is straightforward to include humidity both in the development environment of controllers and in the user interface for monitoring purposes. In the ZoneLib article, inclusion of humidity is merely a matter of expanding the vector between zones, and is, thus, left for possible future development of ZoneLib. With respect to the control of both temperature and humidity, it was planed to expand the game theoretic approach to also include humidity as well as temperature. But the time-consuming and cumbersome required work led to the investigation of an automated process for yielding the control rules. The found tool was UPPAAL TIGA, a tool developed at Aalborg University. As presented in the article submitted for CAV 2007, the tool made it possible to perform simulations in which both temperature and humidity is maintained within the defined boundaries. Whether or not it is possible to maintain both temperature and humidity from the rule based controller in a real pig stable, is not known at present time. This is left for future testing to conclude on.

12.3   Recommendations

For further work and possible product implementations the following recommendations are given:

- Perform experiments to validate the feasibility of using ZoneLib. Development of new climate controllers using ZoneLib could possibly reduce development time, since the simulation can use real actuator signals. ZoneLib could also be useful for modelling unconventional building structures. The library should, however, be expanded to include more types of actuators.

- The results from the parameter estimation should be finalized to yield all possible models based on the outcome of one experiment. If a simple sensor could be developed that only had to inform about the direction of the air flow, the assumption of guaranteed internal airflow would no longer be needed. If all possible models were deduced, a controller could choose the model that has the best fit to control the climate after - see e.g. (Jessen, 2006).

- The game theoretic approach to controller synthesis is based on an ideal model with flow as a control signal. If the parameter estimation procedure could be linked to the game controller, this synthesis procedure would automatically give the actuator signals, thus, giving a fully automated controller synthesis.

- Farmers learn that draught is lethal for the pigs and it should be avoided at all times. With the parameter estimation model it is possible to deduce the influence from operating actuators in one zone on a neighboring zone. Thus, a controller can be made that solves a constraint that all actuator signals should yield zero flow between zones.

- If the every day life of the farmer should change in order to manage larger production facilities, the entire production needs to be automated. Before this can happen, welfare sensors are needed as well as the possibility of monitoring the production status. The presented framework is ready for such a system, once it becomes available. The idea of using a database as inter component integrator (or "glue") easily allows for the inclusion of different simultaneous controllers. The climate controllers do not need to know if the reference point for climate (in the database) is set by a farmer or a different control loop.

- This thesis has not dealt with fault detection, but the zone concept facilitates this, using the extra sensors. It is, therefore, recommended to investigate this in more detail and let the result be linked to the computer system, which could automatically inform the farmer, in case of a fault, or the company servicing the ventilation system.

- During a blizzard, a snow scraper truck destroyed a telecommunication box at the roadside to the experimental facility, leaving the entire area without wired

168

communication. In the proposed setup the computer system uses the Internet for remote monitoring, but in the reported incident, communication with the computer was impossible. For a farmer monitoring the climate this could be severe, and it is, therefore, recommended to install e.g. a gprs modem, thus giving an extra communication line from the computer.

BIBLIOGRAPHY

Aaslyng, J. M., Ehler, N., and Jakobsen, L. (2005). Climate control software integration with a greenhouse environment control computer. *Environmental Modelling & Software*, (20):521–527.

Arvanitis, K., Soldatos, A., Daskalov, P., Pasgianos, G., and Sigrimis, N. (2006). Nonlinear robust temperature-humidity control in livestock buidings. *Biosystems Engineering*, 93(1):1–24.

Balluchi, A., Benvenuti, L., Villa, T., Wong-Toi, H., and Sangiovanni-Vincentelli, A. (1999). A case study of hybrid controller synthesis of a heating system. In *Proc. of Hybrid Systems and AI*, Stanford, CA, 7-12.

Barber, E. M. and Ogilvie, J. R. (1982). Incomplete mixing in ventilated airspaces. part i. theoretical considerations. *Canadian Agricultural Engineering*, 24(1):25–29.

Barber, E. M. and Ogilvie, J. R. (1984). Incomplete mixing in ventilated airspaces. part ii. scale model study. *Canadian Agricultural Engineering*, 26(2):189–196.

Bastianelli, D. and Sauvant, D. (1997). Modelling the mechanisms of pig growth. *Livestock Production Science*, (51):97–107.

Berckmans, D., de Moor, M., and de Moor, B. (1993). Linking physical models to system identification in micro-environment control. In *IFAC 12th Triennial World Congress*, pages 1143–1146.

Berckmans, D. and Goedseels, V. (1986). Development of new control technique for the ventilation and heating of livestock buildings. *Journal of Agricultural Engineering Research*, (33):1–12.

Biering-Sørensen, S., Hansen, F. O., Klim, S., and Madsen, P. T. (1994). *Håndbog i struktureret programudvikling*. Teknisk Forlag A/S. In Danish.

Bibliography

Brecht, A. V., Quanten, S., Zerihundesta, T., Buggenhout, S. V., and Berckmans, D. (2005). Control of the 3-d spatio-temporal distribution of air temperature. *International Journal of Control*, 78(2):88–99.

Brockett, R. and Liberzon, D. (2000). Quantized feedback stabilization of linear systems. *IEEE Trans. Aut. Contr.*, 45:1279–1289.

Brohus, H., Frier, C., and Heiselberg, P. (2002). Stochastic single and multizone model of a hybrid ventilated building - a monte carlo approach. Technical report, Aalborg University.

Bucher, R. and Balemi, S. (2006). Rapid controller prototyping with matlab/simulink and linux. *Control Engineering Practice*, 14(2):185–192.

Bygholm, F. (2001). Skov friskluftventiler. type da 1200 og da 1211. prøverapport nr 937. Technical report, Forskningscenter Bygholm. In Danish.

Caponetto, R., Fortuna, L., Nunnari, G., Occhipinti, L., and Xibilia, M. G. (2000). Soft computing for greenhouse climate control. *IEEE Transactions on Fuzzy Systems*, 8(6):753–760.

Chotai, A., Young, P. C., Tych, W., and Lees, M. (1994). Proportional-integral-plus (pip) design for stochastic delta operator systems. In *International Conference on Control*, volume 1, pages 75–80.

cič, B. Z. (1998). Extension software for real-time control system design and implementation with matlab-simulink. *Simulation Practice and Theory*, (6):703–719.

Coelho, J. P., de Moura Oliveira, P. B., and Cunha, J. B. (2005). Greenhouse air temperature predictive control control using the particle swarm optimisation algorithm. *Computers and Electronics in Agriculture*, (49):330–344.

Cunha, J. B., Couto, C., and Ruano, A. E. (1997). Real-time parameter estimation of dynamic temperature models for greenhouse environmental control. *Control Engineering Practice*, 5(10):1473–1481.

Daskalov, P., Arvinitis, K., Pasgianos, G., and Sigrimis, N. (2004). An adaptive nonlinear technique for environmental control of animal buildings. AgEng.

Eker, J. and Malmborg, J. (1999). Design and implementation of a hybrid control strategy. *IEEE Control Systems Magazine*.

Elia, N. and Mitter, S. (2001). Stabilization of linear systems with limited information. *IEEE Trans. Aut. Contr.*, 46(9):1384–1400.

Fagnani, F. and Zampieri, S. (2003). Stability analysis and synthesis for scalar linear systems with a quantized feedback. *IEEE Trans. Aut. Contr.*, 48:1569–1584.

172

Flinders, F. and Oghanna, W. (1997). Simulation of a complex traction pwm rectifier using simulink and the dynamic node technique. In *23rd International Conference on Industrial Electronics, Control and Instrumentation IECON 97*, volume 2, pages 738–743.

Flinders, F., Senini, S., and Oghanna, W. (1993). Mixed electrical and mechanical simulations using dynamic systems analysis packages. In *Proceedings of the 1993 IEEE/ASME Joint Railroad Conference*, pages 87–93.

Gazi, V. and Passino, K. (2005). Stability of a one-dimensional discrete-time asynchronous swarm. *IEEE Trans. Systems, Man, and Cybernetics – Part B*, 35:834–841.

Ghoumari, M. Y. E., Tantau, H. J., and Serrano, J. (2005). Non-linear constrained mpc: Real-time implementation of greenhouse air temperature control. *Computers and Electronics in Agriculture*, 49:345–356.

Goebel, R., Hespanha, J., Teel, A., Cai, C., and Sanfelice, R. (2004). Hybrid systems: generalized solutions and robust stability. In *Proc. of the 6th IFAC Symp. Nonl. Contr. Syst.,* Stuttgart, Germany.

Guerra, G. and Marsch, R. (1997). A commercial off the shelf (cots) solution for engine simulation. In *1997 IEEE Autotestcon Proceedings*, pages 560–567.

Gutman, P. O., Lindberg, P. O., Ioslovich, I., , and Seginer, I. (1993). A non-linear optimal greenhouse control problem solved by linear programming. *Journal of Agricultural Engineering Research*, (55):335–351.

Hansen, J. A. (2004). Intelligent sensory using compact data structures and bayesian networks. In Petterson, P. and Yi, W., editors, *Nordic Workshop on Programming Theory*.

Hansen, J. A., Jessen, J. J., and Petersen, A.-L. S. (2005). Dyrevelfærd starter i computeren. *Aktuel Naturvidenskab*, (2):20–22. In Danish.

Hansen, L. H., Bidstrup, N., Madsen, H., Vadstrup, P., and Holst, J. (1995). A hydraulic simulink model of the heating system in a single family house. In *Nordic Matlab Conference*.

Hespanha, J., Naghshtabrizi, P., and Xu, Y. (2006). A survey of recent results in networked control systems. To appear in the Proc. of IEEE, Special Issue on Networked Control Systems.

Heymann, M., Lin, F., and Meyer, G. (1998). Synthesis and viability of minimally interventive legal controllers for hybrid systems. *Discrete Event Dyna. Syst.: Theory Appl.*, 8:105–135.

Homepage, U. T. (2006). `http://www.cs.aau.dk/~adavid/tiga`.

Bibliography

Inc., T. M. (2006). Matlab. http://www.mathworks.com.

Ishii, H. and Francis, B. (2002). Stabilizing a linear system by switching control with dwell time. *IEEE Trans. Aut. Contr.*, 47:1962–1973.

Jadbabaie, A., Lin, J., and Morse, A. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Aut. Contr.*, 48:692–697.

Janssens, K., Brecht, A. V., Desta, T. Z., Boonen, C., and Berckmans, D. (2004). Modelling the internal dynamics of energy and mass transfer in an imperfectly mixed ventilated airspace. *Indoor Air*, 14(3):146–153.

Jessen, J. J. (2006). Future climate control system for multizone livestock buildings based on system identification. *DANET Newsletter*, pages 1–4.

Jessen, J. J., Nielsen, J. F. D., Schiøler, H., and Jensen, M. R. (2006a). Cots technologies for internet based monitoring of livestock buildings. In Hwang, J.-N., editor, *Proceeding of the 10th IASTED International Conference on Internet and Multimedia Systems and Applications*, pages 10–15. IASTED, ACTA Press.

Jessen, J. J. and Schiøler, H. (2006a). Parameter estimation for zone based climate dynamics. In *The Second IASTED International Conference on Computational Intelligence (CI2006). San Francisco, California, USA*.

Jessen, J. J. and Schiøler, H. (2006b). System identification in a multi zone livestock building. In *International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA06). Sydney, Australia*. Accepted for publication. Withdrawn due to financial issues.

Jessen, J. J. and Schiøler, H. (2006c). Zonelib: A simulink library for modeling zone divided climate dynamics. In *ISCA 19th International Conference on Computer Applications in Industry and Engineering (CAINE2006). Las Vegas, Nevada, USA*.

Jessen, J. J., Schiøler, H., Nielsen, J. F. D., and Jensen, M. R. (2006b). Cots technologies for integrating development environment, remote monitoring and control of livestock stable climate. In *Proceedings 2006 IEEE International Conference on Systems, Man, and Cybernetics*.

Johansson, K., Speranzon, A., and Zampieri, S. (2005). On quantization and communication topologies in multi-vehicle rendezvous. In *Proc. of the 16th IFAC World Congress,* Prague, Czech Republic.

Koerkamp, P. W. G. G., Metz, J. H. M., Uenk, G. H., Phillips, V. R., Holden, M. R., Sneath, R. W., Short, J. L., White, R. P., Hartung, J., Seedorf, J., Schröder, M., Linkert, K. H., Pedersen, S., Takai, H., Johnsen, J. O., and Wathes, C. M. (1998). Concentrations and emissions of ammonia in livestock buildings in nortern europe. *J. agric. Engng Res.*, 70:79–95.

174

Lay, D. C. (1997). *Linear Algebra and its Applications*. Addison Wesley, 2 edition.

Liberzon, D. and Hespanha, J. (2005). Stabilization of nonlinear systems with limited information feedback. *IEEE Trans. Aut. Contr.*, 50:910–915.

Lin, Z., Francis, B., and Maggiore, M. (2005). Necessary and sufficient graphical conditions for formation control of unicycles. *IEEE Trans. Aut. Contr.*, 50:121–127.

Linker, R., Gutman, P. O., and Seginer, I. (1999). Robust controllers for simultaneous control of temperature and co2 concentrations in greenhouses. *Control Engineering Practice*, (7):851–862.

Linker, R., Gutman, P. O., and Seginer, I. (2002). Observer-based robust failure detection and isolation in greenhouses. *Control Engineering Practice*, (10):519–531.

Ljung, L. (1987). *System Identification - Theory for the User*. Prentice-Hall Information and System Science Series. Prentice Hall.

Lygeros, J., Tomlin, C., and Sastry, S. (1999). Controllers for reachability specifications for hybrid systems. *Automatica*, 35:349–370.

Mamdami, E. and Assilina, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7:1–13.

Martinez, S., Bullo, F., Cortes, J., and Frazzoli, E. (2005). On synchronous robotic networks. Part I and II. In *Proc. of the 44th IEEE Conf. Dec. Contr. and Europ. Contr. Conf. ECC'05,* Sevilla, Spain.

Montgomery, D. C. (2001). *Design and Analysis of Experiments*. John Wiley & Sons, INC., 5th edition.

Moor, M. D. and Berckmans, D. (1996). Building a grey box model to model the energy and mass transfer in an imperfectly mixed fluid by using experimental data. *Mathematics and Computers in Simulation*, 42:233–244.

Morcos, M. M., Mowry, J. A., and Heber, A. J. (1994). A solid-state speed controller for capacity motors driving ventilation fans. *IEEE Transactions on Industry Applications*, 30(3):656–664.

Moughan, P. J., Verstegen, M. W. A., and Visser-Reyneveld, M. I., editors (1995). *Modelling Growth in the Pig*. Wageningen Pers.

Nair, G., Evans, R., Mareels, I., and Moran, W. (2004a). Topological feedback entropy and nonlinear stabilization. *IEEE Trans. Autom. Contr.*, 49(9):1585–1597.

Nair, G., Evans, R. J., and Caines, P. E. (2004b). Achievable rate regions for decentralised stabilisation. In *Proc. of the 16th Int. Symp. on Mathematical Theory of Networks and Systems,* Katholieke Universiteit Leuven, Leuven, Belgium.

175

Bibliography

Nair, G. N. and Evans, R. J. (2003). Exponential stabilisability of finite-dimensional linear systems with limited data rates. *Automatica*, 39:585–593.

Nešić, D. and Teel, A. (2004). Input to state stability of networked control systems. *Automatica*, 40:2121–2128.

Ngo, D. and Harris, M. (2001). A reliable infrastructure based on cots technology for affordable space application. In *IEEE Proceedings*, volume 5 of *Aerospace Conference*, pages 2435–2441.

Nielsen, B. and Madsen, H. (1998). Identification of a linear continuous time stochastic model of the heat dynamics of a greenhouse. *Journal of Agricultural Engineering Research*, (71):249–256.

Ottenheimer, A. (2006). phplot. http://sourceforge.net/projects/phplot/.

Pasgianos, G. D., Arvanitis, K., Polycarpou, P., and Sigimis, N. (2003). A nonlinear feedback technique for greenhouse environmental control. *Computers and Electronics in Agriculture*, (40):153–177.

Pedersen, S., Morsing, S., m, J. S. S., and Jacobsen, L. (2004). Simulation of indoor climate in animal houses using cigr 2002 heat production. In *Internation Symposium of the CIGR Section II in Évora, Portugal*.

Pedersen, S. and Saalvik, K. (2002). Climatization of animal houses. Technical report, International Commission of Agricultural Engineering, Section II.

Pedersen, S., Takai, H., Johnsen, J. O., Metz, J. H. M., Koerkamp, P. W. G. G., Uenk, G. H., Phillips, V. R., Holden, M. R., Sneath, R. W., Short, J. L., White, R. P., Hartung, J., Seedorf, J., Schröder, M., Linkert, K. H., and Wathes, C. M. (1998). A comparison of three balance methods for calculating ventilation rates in livestock buildings. *J. agric. Engng Res.*, 70:25–37.

Persis, C. D. (2005). n-bit stabilization of n-dimensional nonlinear systems in feedforward form. *IEEE Trans. Aut. Contr.*, 30:299–311.

Persis, C. D. and Isidori, A. (2004). Stabilizability by state feedback implies stabilizability by encoded state feedback. *Systems & Control Letters*, 53:249–258.

Persis, C. D., Jessen, J. J., Izadi-Zamanabadi, R., and Schiøler, H. (2006). Internal flow management in a multi-zone climate control unit. In *Invited paper in the session* Networked Control Systems *2006 CCA/CACSD/ISIC*.

Persis, C. D., Jessen, J. J., Izadi-Zamanabadi, R., and Schiøler, H. (2007). A distributed control algorithm for internal flow management in a multi-zone climate unit. *International Journal of Control*. Accepted for publication. To appear.

176

Piñón, S., Camacho, E. F., Kuchen, B., and Peña, M. (2005). Constrained predictive control of a greenhouse. *Computers and Electronics in Agriculture*, 49:317–329.

Pressman, R. S. (1986). *Software engineering: a practitioner's approach (2nd ed.).* McGraw-Hill, Inc., New York, NY, USA.

Quaranta, G. and Mantegazza, P. (2001). Using matlab-simulink rtw to build real time control applications in user space with rtai-lxrt. In *Realtime Linux Workshop Milano 2001*.

Quijano, N., Gil, A., and Passino, K. (2005). Experiments for dynamic resource allocation, scheduling, and control. *IEEE Control Systems Magazine*, 25:63–79.

Schauberger, G., Piringer, M., and Petz, E. (2000). Steady-state balance model to calculate the indoor climate of livestock buildings, demonstrated for finishing pigs. *International Journal of Biometeorology*, (43):154–162.

Schmidt, E. (1989). *Properties of water and steam in SI-Units*. Springer-Verlag Berlin Heidelberg New York.

Scleef, D. (2006). Comedi. www.comedi.org.

Seedorf, J., Hartung, J., Schröder, M., Linkert, K. H., Phillips, V. R., Holden, M. R., Sneath, R. W., Short, J. L., White, R. P., Pedersen, S., Takai, H., Johnsen, J. O., Metz, J. H. M., Koerkamp, P. W. G. G., Uenk, G. H., and Wathes, C. M. (1998). Concentration and emissions of airborne endotexins and microorganisms in livestock buildings in northern europe. *J. agric. Engng Res.*, 70:97–109.

SIMULINK (2007). `http://www.mathworks.com/products/simulink/`.

Swarop, D. and Hedrick, J. (1996). String stability of interconnected systems. *IEEE Trans. Aut. Contr.*, 41:349–357.

Tatikonda, S. and Mitter, S. (2004). Control under communication constraints. *IEEE Trans. Aut. Contr.*, 49:1056–1068.

Taylor, C. J., Leigh, P., Price, L., Young, P. C., Vranken, E., and Berckmans, D. (2004). Proportional-integral-plus (pip) control of ventilation rate in agricultural buildings. *Control Engineering Practice*, 12(2):225–233.

van der Schaft, A. and Schumacher, H. (2000). *An introduction to hybrid dynamical systems*. LNCIS. Springer, London.

Web, P. (2007). `http://www.cs.aau.dk/~illum/automatic_control/`.

website (2007a). Ptolemyii homepage. `http://ptolemy.berkeley.edu/ptolemyII/`.

Bibliography

---

website (2007b). Scicos homepage. `http://www.scicos.org/`.

website (2007c). Scilab homepage. `http://www.scilab.org/`.

Wong, W. and Sung, C. (1996). Robust convergence of low-data-rate distributed controllers. *IEEE Trans. Aut. Contr.*, 41:349–357.

Workshop, R.-T. (2007). `http://www.mathworks.com/products/rtw/`.

Yale, W. (1997). Data warehouse tools. In *MILCOM 97 Proceedings*, volume 2, pages 764–767.

Young, P., Price, L., Berckmans, D., and Janssens, K. (2000). Recent developments in the modelling of imperfectly mixed airspaces. *Computers and Electronics in Agriculture*, 26(3):239–254.

Zaheer-uddin, M., Patel, R., and Al-Assadi, S. (1993). Design of decentralized robust controllers for multizone space heating systems. *IEEE Transactions on Control Systems Technology*, 1:246–261.

Zerihun, T., Brecht, A. V., Quanten, S., Buggenhout, S. V., Meyers, J., Baelmans, M., and Berckmans, D. (2005). Modeling and control of heat transfer phenomena inside a ventilated air space. *Energy and Buildings*, 37:777–786.