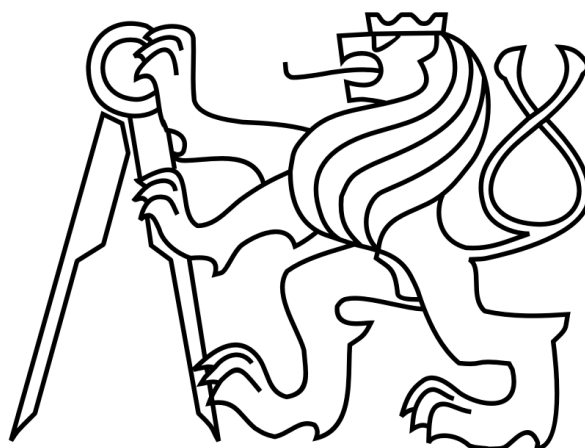


CZECH TECHNICAL UNIVERSITY IN PRAGUE
FACULTY OF ELECTRICAL ENGINEERING
DEPARTMENT OF CONTROL ENGINEERING



Diploma Thesis
Climatic Unit Control

Author: Bc. Jiří Dostál

Supervisor: Ing. Lukáš Ferkl, Ph.D.

Prague, January, 2013

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Jiří Dostál**

Studijní program: Kybernetika a robotika
Obor: Systémy a řízení

Název tématu: **Řízení klimatické jednotky**

Pokyny pro vypracování:


1. Navrhněte a sestrojte systém sběru dat a zasílání řídicích povelů do klimatické jednotky.
2. Seznamte se s metodami identifikace systému. Vytvořte model systému klimatické jednotky.
3. Navrhněte řízení teploty, vlhkosti a osvětlení klimatické jednotky s ohledem na 24 hodinový harmonogram mikroklimatu.
4. Řízení realizujte.

Seznam odborné literatury:

Dodá vedoucí práce

Vedoucí: Ing. Lukáš Ferkl, Ph.D.

Platnost zadání: do konce letního semestru 2012/2013


prof. Ing. Michael Šebek, DrSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan


V Praze dne 2. 12. 2011

Declaration/Prohlášení

I hereby formally declare that the paper submitted is entirely my own and all resources (literature, projects, software etc.) used are cited appropriately with the corresponding bibliographical references provided.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

In Prague, date / V Praze dne: 31.12.2012

Signature / Podpis: 

Acknowledgements

I would like to express my gratitude and thank to my supervisor Ing. Lukáš Ferkl, Ph.D. for his guidance and helpful comments.

My sincere thanks go to my grandfather Ing. Josef Dostál, CSc. for providing much useful advice about vegetation, biology and life in general.

I also want to give thanks to numberless open source developers who indirectly made implementation of all my software components possible by sharing code and thoughts with the general public.

Last but not least, I would like thank to my parents for their continuous support and endless patience.

Thank you!

Abstract

The aim of this thesis is to propose a control of temperature, humidity and illumination in a climatic unit.

Initially, a specific climatic unit - the Klimabox - is described; the control is to be realized on this climatic unit. The first part also lists psychrometric properties of moist air and their calculation.

The communication system necessary for a data acquisition and command signaling is presented in the next part. The acquired data is stored in an external database using a newly created computer program. The program also provides a network communication interface by which a controller is to be connected to the Klimabox.

Further, an identification of the Klimabox dynamic model is explained. The system was first described using the first principles (white-box method); a library of basic dynamic components was created and the model was constructed. The paper further presents black-box identification methods that led to the identification of the Klimabox model suitable for a controller design.

The next part shows the development of a model predictive controller for the climatic unit. The control process was simulated on the identified model and afterwards realized on the real Klimabox.

Finally, a possible future development of the climatic unit and its controller is proposed.

Aim of the thesis

1. Design data acquisition and command signaling system for a climatic unit.
2. Study system identification methods and create a dynamic model of the climatic unit.
3. Develop a controller of temperature, humidity and illumination for the climatic unit.
4. Implement the controller into the climatic unit and perform the control.

Abstrakt

Cílem této práce je navrhnout řízení teploty, vlhkosti a osvětlení klimatické jednotky.

Nejprve je představena klimatická jednotka - Klimabox, na které bude řízení realizováno. A jsou zkoumány fyzikální vlastnosti vlhkého vzduchu a jejich výpočet.

V další kapitole je popsán systém komunikace nutný ke sběru dat a řízení klimatické jednotky. Data jsou ukládána za pomoci nově vytvořeného programu do externí databáze. Program také poskytuje síťové komunikační rozhraní, přes které je Klimabox připojen k regulátoru.

Následující část obsahuje identifikaci dynamického modelu Klimaboxu. Systém byl nejprve popsán na základě fyzikálních zákonů (metodou white-box). K tomuto účelu byla vytvořena knihovna základních dynamických komponent, ze které byl následně model sestrojen. Práce dále popisuje identifikaci Klimaboxu metodami black-box, které vedly k získání modelu použitelného k regulaci.

Cílem další kapitoly je vysvětlit postup návrhu prediktivního regulátoru vhodného k řízení klimatické jednotky. Regulace byla simulována na identifikovaném modelu a následně realizována na skutečném Klimaboxu.

Poslední kapitola se věnuje možnému budoucímu vývoji této klimatické jednotky.

Contents

I. Theory & Background	2
1. Klimabox	3
1.0.1. Brief Klimabox description	3
1.1. History	4
1.1.1. Development phases	4
1.1.2. My preceding work on Klimabox	5
1.2. Use of controlled climate for biological research	6
1.3. Climatic environment for a plant growth	6
1.3.1. Light	7
1.3.2. Temperature	8
1.3.3. Humidity	8
1.3.4. Air circulation and fresh air delivery	9
1.4. Climatic environment for cultivation of other cultures	10
1.4.1. Explantate cultures	10
1.4.2. Micro-bacterial culture	10
1.5. Klimabox construction	10
1.5.1. Klimabox dimensions	10
1.5.2. Block decomposition	10
1.5.3. Light panel	11
1.5.4. Cooling	11
1.5.5. Heating	12
1.5.6. Air humidification	12
1.5.7. Air dehumidification	13
2. Physical background	14
2.1. Heat transfer	14
2.1.1. Thermal conduction	14
2.1.2. Thermal radiation	14
2.1.3. Thermal convection	15
2.2. Psychrometric properties of moist air	15
2.2.1. Atmospheric pressure	16
2.2.2. Saturation water vapor pressure	16
2.2.3. Partial water vapor pressure	17
2.2.4. Humidity ratio	18
2.2.5. Specific volume	18
2.2.6. Absolute humidity	19
2.2.7. Dew point temperature	19
2.2.8. Specific enthalpy of moist air	20
2.2.9. Psychrometric calculation example	21

II. Data acquisition and command signaling	22
3. Klimabox control panel	23
3.1. History	23
3.2. Circuit description	24
3.3. Control panel software	24
3.3.1. Periphery drivers	25
3.3.1.1. Basic libraries	26
3.3.1.1.1. UART	26
3.3.1.1.2. LCD	26
3.3.1.1.3. I ² C	26
3.3.1.1.4. 1-Wire	26
3.3.1.2. Real Time Clock	27
3.3.1.3. SHT11 sensor	27
3.3.1.3.1. Batch processing vs. state machine	27
3.3.1.3.2. Checksum and swap byte	28
3.3.1.3.3. Raw data conversion	28
3.3.2. Data storage	29
3.3.3. Compilation	30
3.3.3.0.4. Compiler switches	30
3.3.3.0.5. Linker switches	30
4. Server	31
4.1. Klimabox daemon	31
4.1.1. Daemonizing a process	32
4.1.2. Periphery drivers	33
4.1.3. Configuration file	34
4.1.4. Logging	34
4.1.5. Future development	34
4.2. Klimabox user interface	35
4.3. Database	35
5. Communications protocol—Klimanet	38
5.1. Klimanet commands	40
5.2. Control commands	40
5.2.1. KNetDumpCmd	40
5.2.2. KNetSetOutputs	40
5.2.3. KNetSetClimPreset	41
5.2.4. KNetSetTempLimits	41
5.2.5. KNetSetDateTime	42
5.2.6. KNetSetLcdBacklight	42
5.2.7. KNetBeep	42
5.2.8. KNetDispText	42
5.2.9. Slave response to the control commands	43
5.3. Request commands	44
5.3.1. Request packet	44
5.3.2. KNetGetTempHumi	44
5.3.2.0.6. Request command	44
5.3.2.0.7. Response	45
5.3.3. KNetGetOutputs	46
5.3.4. KNetGetClimPreset	46
5.3.4.0.8. Request command	46

5.3.4.0.9. Response	47
5.3.5. KNetGetStatus	47
5.3.6. KNetGetDateTime	47
5.4. Slave request commands	48
5.4.1. KNetMeasReadReq	48
5.4.2. KNetClimPresetReadReq	48
5.4.3. KNetRestartOccured	48
5.4.4. KNetDebugMessage	49
5.5. Command summary	49
6. Measurement acquisition	51
6.1. Humidity sensor calibration	51
6.1.1. Calibration in a fixed abs. humidity environment	52
6.1.2. Humidity calibration by fixed points of aqueous salt solutions	53
6.1.3. SHT11 sensor calibration	54
6.1.3.0.10. Calibration in a fixed absolute humidity environment	54
6.1.3.0.11. Calibration by aqueous salt solutions	55
6.2. Conversion of the SHT11 sensor humidity signal output	56
III. Model	60
7. Model introduction	61
7.1. Klimabox model outline	62
7.2. ARX model and least-square estimation	62
8. First principles approach	64
8.1. Identification of Klimabox's components	64
8.1.1. Heating rod identification	64
8.1.2. Cooling coil unit identification	66
8.1.3. Humidifier identification	67
8.1.4. Gas tubes identification	67
8.1.5. Bottom fans identification	68
8.1.6. Chamber body dynamics identification	69
8.2. Klimabox library	70
8.3. Klimabox simulator	73
9. Black-box approach	75
9.1. ARX identification method	75
9.2. Subspace identification method	77
9.2.1. Pseudo-random binary sequence	77
9.2.2. Step tests	81
9.2.3. Model comparison	84
IV. Control	86
10. Controller introduction	87
10.1. Model predictive control	87
11. Klimabox controller	89
11.1. Klimabox dynamic model	89

11.2. Setpoint	90
11.2.1. Setpoint filter	91
11.3. Optimization cost function	92
11.4. Optimization constraints	93
11.5. Klimabox MPC controller simulations	94
11.6. MPC vs. ON-OFF controller	97
11.7. Klimabox MPC controller application	98
12. Future development	101
12.1. LED lamps	101
12.2. Explicit MPC	102
12.3. Control panel connectable to the Ethernet	102
13. Conclusion	104
13.1. Data acquisition and command signaling	104
13.2. Identification	105
13.3. Control	106
13.4. Summary	107
Bibliography	107
A. Contents of the CD attached	I

List of Figures

1.1. Klimabox	3
1.2. Klimabox block composition	4
1.3. Development phases of the Klimabox	5
1.4. Absorption spectrum of Chlorophyll and Carotenoids [15]	7
1.5. Growth chamber.	9
1.6. Air circulation	9
1.7. Light panel	11
1.8. Technology block	12
1.9. Humidifier	13
3.1. Klimabox control panel	23
3.2. Klimabox control panel history	24
3.3. Klimabox control panel hardware block diagram	25
3.4. SHT11 hardware block diagram	27
4.1. Control process - communication scheme	32
4.2. Remote connection to the klimaboxd user interface	35
5.1. Diagram of RS232 signaling	39
5.2. Klimanet frame summary	39
6.1. Relative humidity vs. temperature relationship in a fixed absolute humidity environment	52
6.2. SHT11 sensor being calibrated by potassium carbonate.	54
6.3. SHT11 sensor calibration by constant absolute humidity	55
6.4. SHT11 sensor resolution	55
6.5. SHT11 sensor calibration by saturated salts	56
6.6. Deviation of the integer approximation from the SHT11 relative humidity conversion function.	58
7.1. Generic control scheme	61
7.2. Klimabox model scheme	62
8.1. Energy balance on the heating rod	65
8.2. Heating rod identification	65
8.3. Cooling coil unit identification	66
8.4. Gas tubes heat up dynamics identification	68
8.5. Chamber heat losses identification experiment	69
8.6. Klimabox library	70
8.7. Klimabox library components	72
8.8. Klimabox simulator - heat loss simulation	73
8.9. Klimabox simulator	74
9.1. Klimabox ARX model data fit	76
9.2. Klimabox identification data - PRBS	78

9.3. N4SID_PRBS model data fit and step response.	80
9.4. Klimabox step-test data	82
9.5. N4SID_STEP model data fit and step response.	83
9.6. Model comparison and diagnostics	85
10.1. Model predictive control scheme	87
11.1. Klimabox prediction control scheme	89
11.2. Climatic preset: Pepper - prior to blooming period [7].	90
11.3. Setpoint filter	91
11.4. Klimabox MPC controller simulations	95
11.5. Klimabox ON-OFF controller simulations	96
11.6. MPC vs. ON-OFF controller for tight reference funnel	97
11.7. Klimabox control process communication scheme	98
11.8. Klimabox control (real data)	100
12.1. Absorption spectrum of chlorophyll and carotenoids [15].	101

Part I.

Theory & Background

1. Klimabox

Klimabox is a device generating a controlled climate - a climatic chamber. Climatic chambers are frequently used in biological research; however, they are also useful in the serial production of biological items or industrial products, materials, or electronic device testing.



Figure 1.1.: Klimabox

1.0.1. Brief Klimabox description

Klimabox is a device composed of an isolated chamber as the core element, temperature control equipment (heating and cooling units), humidity control equipment (humidifier, condenser) and light emitting devices.

1. Klimabox

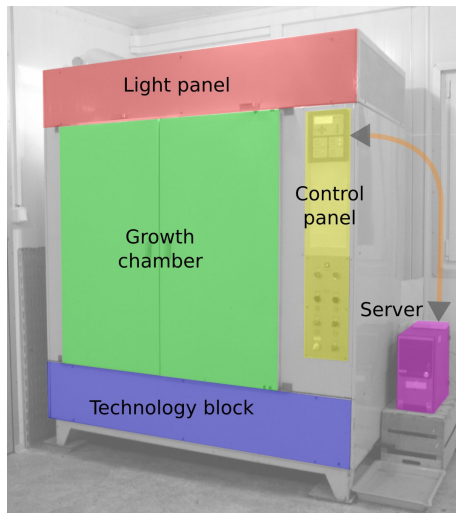


Figure 1.2.: Klimabox block composition

Lights, particularly gas tubes and fluorescent tubes, reside at the top section in a separated light panel, see fig. 1.2. The growth chamber ($1m^3$) in the middle is thermally isolated from the surroundings and its environment is controlled by actuators. The actuators reside in the bottom section in the technology panel.

A thorough Klimabox description is given in section 1.5.

1.1. History

Development of the first Czechoslovakian climatic chamber called “Klimabox” began in 1976 and continued evolving by several stages until 1985. The main inventor and constructor of this device is my grandfather Ing. Josef Dostál, CSc. All the major innovations were later patented in his favor.

This section will give a brief description of the developmental stages until 1985. Klimabox was afterwards no longer manufactured. Further, my prior Klimabox development is presented.

1.1.1. Development phases

Klimabox was not built from scratch into its final shape, instead it evolved through a number of phases. Each phase or sub-phase had at least one prototype.

1. A modified confectioner’s oven for explantate cultures cultivation.
2. A refrigerator equipped with a side wall and ceiling lamp for in-vitro¹ cultures re-cultivation.
3. A newly designed Klimabox with air circulation and:
 - a) with bi-metallic thermometers
 - b) with an electronic temperature control
 - c) with an electronic control of temperature, humidity and illumination (analogue)

¹Cultivation of biologic material in a chemical glass (test tubes, petri etc.)

- d) with presettable daily temperature, humidity and illumination profiles (day/ night settings).
- 4. Klimabox with light panel separated from a growth chamber.
- 5. Micro-controller based digital control.

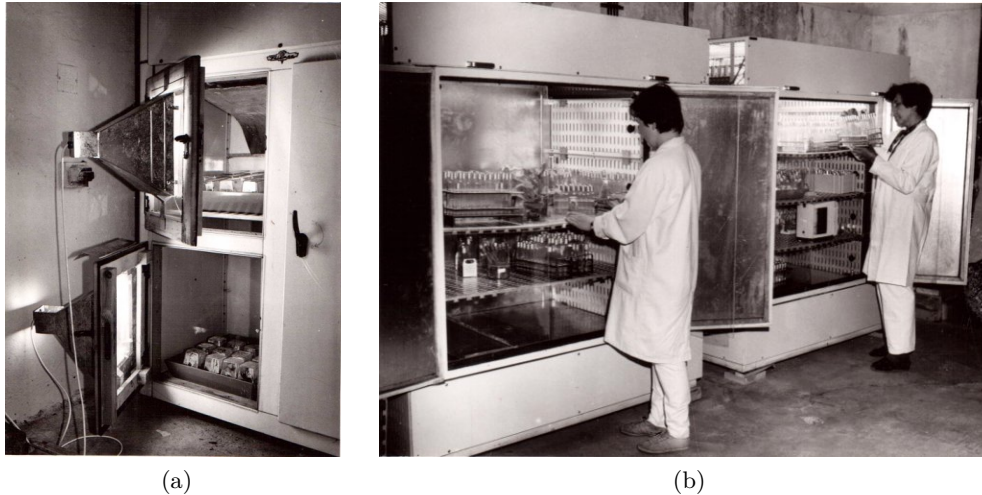


Figure 1.3.: Development phases of the Klimabox: phase 2 (a) and phase 3-c (b)

The major improvement in the Klimabox at this time was the separation of the light sources from the growth chamber at stage 4. Common light sources used for artificial growing were gas tubes that, besides the intense light, also produce a significant amount of heat. This heat had to be drained later by a cooling coil unit (CCU), which led to a not negligible rise of energy consumption. A set of fans and an acrylate transparent deck between the lights and chamber helped to reduce the transferred heat down to 20% of the original.

Development stages 3-4 had only mechanical or analog electronic control. Such methods provide limited control possibilities, e.g., the daily profile consisted of one setting for day and the other for night. The structure of the current controller was outdated and too rigid for improvements so I decided to develop a new digital controller - beginning a new phase of Klimabox development (stage 5).

1.1.2. My preceding work on Klimabox

My grandfather managed to get one Klimabox into our laboratory in winter 2004. In autumn 2004, I started to design a new digital control panel for the Klimabox; it was the main topic of my graduation project [5] at electrical engineering high school. I graduated in 2005 getting an A for a working control panel with LCD, keyboard and software to let users set daily climate profiles.

1. Klimabox

During the work on my bachelor thesis [6], defended in 2009, I managed to integrate the control panel into the Klimabox device and run the first simple on-off control of temperature with it. The most troubling part of my bachelor thesis was to find out how the Klimabox worked and get it working again, unfortunately no plans or circuit schemes were located. And to locate refurbishments for 20-year-old parts was not always an easy task. An AC unit was not present at all and its size had to be estimated by a refrigeration expert. I graduated getting a C.

1.2. Use of controlled climate for biological research

Artificial climate is composed of temperature and humidity profiles, a further light intensity profile, and air change, with an eventual addition of carbon dioxide (CO_2) control. Complementary to a static climate control is its use in time according to biological needs, daily and seasonal biorhythms and ontogenetic² evolution of the biological material placed inside the chamber.

Climatic environment need differs according to the biological material, which could be plants grown in soil or substrates, explantate cultures³ inside various media and cell cultures, either herbal or carnal. Further it also differs for micro-bacterial and fungal cultures of distinct origin. Setting a climatic profile is, however, a job for biology researchers and its exact prior specification is not crucial for controller development. Klimabox also has special use in human and veterinary medicine.

Optimal setting and control of a climatic environment is not an easy task for either a biology researcher or a control engineer.

1.3. Climatic environment for a plant growth

Plants need for their successful growth a rather narrow set of environmental variables. The first necessity is an optimal substrate providing nutrients to roots and nourishing the vegetative parts. Reception of nutrients and transport requires the proper temperature and soil moisture. The temperature of the substrate often differs from the temperature needed for vegetative parts; it is usually lower.

²Exhibited within a life-span of a subject.

³Explantate cultures arise when cultivating isolated parts of a plant under prepared conditions. It is a traditional method of vegetative reproduction.

Control of substrate temperature is rarely necessary. Firstly the heat capacity of the soil acts as a natural delay in temperature changes during a day. Secondly a plant root system has self-thermo-regulation ability.

1.3.1. Light

Light is a prerequisite of photosynthesis and therefore the production of biomass; it is an essential condition of plant growth.

Photosynthesis runs efficiently when the material is being lit with light of wavelengths $380 - 760nm$ (fig. 1.4). Important factors are also irradiation intensity, electromagnetic spectrum and duration of illumination.

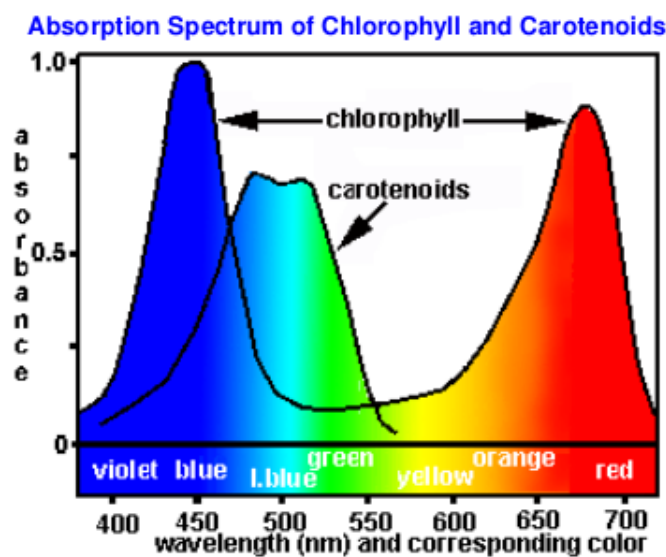


Figure 1.4.: Absorption spectrum of Chlorophyll and Carotenoids [15]

To make the light spectrum appropriate for most plants, Klimabox uses two sources of light - gas tubes and fluorescent tubes. Gas tubes are used for their intense power output and red spectrum, fluorescent tubes are used mainly to shape the spectrum. The light panel produces light with intensity between 2500 and 40000 Lx. Heat made by light sources has to also be taken in account. But as stated in 1.1, Klimabox's separated light panel helps to eliminate up to 80% of the heat from gas tubes.

An economic way of illumination would be LED lamp usage.

1.3.2. Temperature

Temperature is one of the key elements in the on-going chemical reactions. Speed of reaction doubles every 10°C and consequently it affects the speed of cell division and therefore the speed of the plant's growth. Each type of vegetation has from the phylogenetic⁴ perspective distinct temperature requirements. For example PRIMULA VERIS, one of the Czech first spring blooming flowers, if taken to a temperature surpassing 15°C diminishes and dies down. On the other hand, the thermophilic plants, for example, require temperatures that always exceed 25°C. Cardinal temperatures⁵ for a selected number of plants are stated in table 1.1.

Plant	Cardinal temperatures [°C]		
	minimum	optimum	maximum
Barley, oat, rye, wheat	0-5	25-31	31-37
Buckwheat	0-5	25-31	37-44
Hemp	0-5	37-44	44-50
Sunflower	5-10	31-37	37-44
Corn	5-10	37-44	44-50
Pumpkin	10-15	37-44	44-50
Gherkin	15-18	31-37	44-50

Table 1.1.: Cardinal temperatures of selected crop. [7]

There is a close relation between temperature and illumination with respect to the plant cycle. When turning down the light, the temperature should fall too, as is common in nature. Temperature changes in the chamber should be gradual.

1.3.3. Humidity

Photosynthesis can only happen in terms of a significant loss of water due to evaporation although the photosynthesis itself is not a very water-consuming process.

Evaporation from a plant happens mainly by transpiration through leaf pores and its intensity depends on the relative humidity of the air. When humidity is low the transpiration is more severe, as it is when the temperature is high. That is why it is important to maintain good water balance and not let a plant suffer from water deficit. Klimabox should be able to control air humidity in a range of approximately 30 – 95% of relative humidity. Such a range covers the needs of the majority of plants.



Figure 1.5.: Growth chamber.

1.3.4. Air circulation and fresh air delivery

Inner space in Klimabox's chamber (fig. 1.5) can be divided by four perforated aluminum shelves. All four shelves can be used when explantate or micro-bacterial cultures are cultivated. However, when growing plants, the usage of all shelves is limited - mainly because of a low illumination in the lower floors. Young plants are typically placed on the top shelf close to the lamps and as they grow, they are moved down shelf by shelf.

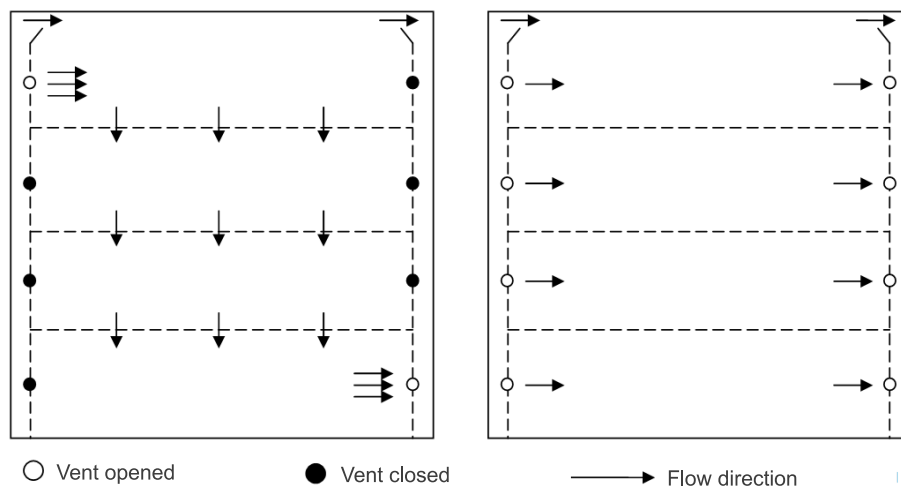


Figure 1.6.: Air circulation

Side panels of the climatic chamber are also perforated. The perforation is adjustable to provide the user with the possibility of setting the air movement throughout the chamber.

⁴Evolution of plant caused by genetic modifications.

⁵Threshold temperature of a vegetative mode of a plant.

1. Klimabox

Two example configurations can be found at figure 1.6.

Fresh air is delivered through holes in the acrylate deck separating the light panel from the chamber. The chamber is also not air-tight.

1.4. Climatic environment for cultivation of other cultures

Environmental needs of other cultures are often very different from those of plants.

1.4.1. Explantate cultures

Explantate cultures are often held in closed translucent containers. Humidity control can be omitted; only temperature and light are controlled.

1.4.2. Micro-bacterial culture

Micro-bacterial cultures often do not require light or humidity control. Temperature control is sufficient.

1.5. Klimabox construction

As it was mentioned before, the goal of this thesis is to control climatic environment in a climatic chamber - Klimabox. To be able to do so, actuators capable of changing climate properties are needed. A description of all actuators installed in Klimabox will be presented in the following sections.

1.5.1. Klimabox dimensions

Klimabox was designed as a device for planting crops under a controlled climate. The actual chamber size is a compromise between technical limitations and growth space. Width of the growth chamber is determined by fluorescent tube length and maximal door size, depth of the chamber is given by maximal reach of a hand inside. Height was determined from the height of fully grown wheat and barley.

1.5.2. Block decomposition

Klimabox is assembled from a light panel, growth chamber and technology panel (fig. 1.2).

1.5.3. Light panel

The light panel (fig. 1.7) is a standalone block separated from the growth chamber. Four gas tubes SHC250W can be found inside the panel. If the tubes were placed inside the growth chamber, the AC unit would have to drain the excessive heat. It would be turned on most of the day period, which would cause a high energy consumption. The placement of gas tubes in the separated panel helps to save up to 80% of consumed energy. The separation element had to be translucent, thermal resistant and not fragile. Glass could not be used due to safety problems, but an acrylate plexiglass was finally found to satisfy all requirements. The separation and the use of acrylate plexiglass as a separating element were the main improvements made on the Klimabox.

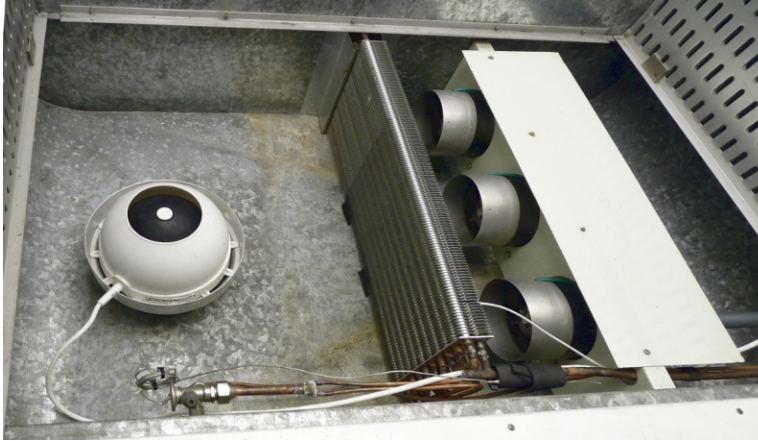
Further, ten fluorescent tubes Tesla 36W are placed in the light panel. A combination of gas tubes and fluorescent tubes gives a more convenient light spectrum.



Figure 1.7.: Light panel

1.5.4. Cooling

Cooling is provided by a cooling control unit (CCU). It is a collection of aggregate, tubing and evaporator connected in one circuit. Aggregate is placed outside the Klimabox body which is particularly useful for maintenance. Noise can be diminished by placing the aggregate into a different room. A refrigerant flowing in the tubes is the one most used nowadays, R134A - a freon ecological substitute. Ozone depletion potential of R134A is an exact zero. Ribboned evaporator is placed inside the Klimabox and can be found in the technology block (fig. 1.8).



(a)



(b)

Figure 1.8.: Technology block: actuators (a), aggregate (b)

1.5.5. Heating

When heating is needed a heating rod is turned on. The heating rod is capable of delivering 400 W and it is also placed in the technology block. The heating rod is hidden behind the fans in figure 1.8.

1.5.6. Air humidification

Humidifiers in general can be categorized as follows:

1. Droplet humidifiers
 - a) rotary
 - b) vacuum tubular
 - c) ultrasonic
 - d) wick capillary evaporative

2. Steam humidifiers

Humidifiers from the first group, sometimes improperly called adiabatic, use some physical principle to disperse water droplets into air. Droplets evaporate suddenly and the latent heat needed for the evaporation causes a temperature reduction.

However, steam humidifiers deliver water into the air already evaporated. The water vapor molecules already possess the phase change energy and the enthalpy (temperature) of air is therefore rising.

Droplet humidifiers are more suitable for climatic chambers. Figure 1.9 shows a rotary humidifier currently in use. An ultrasonic humidifier will be bought in future.



Figure 1.9.: Humidifier

1.5.7. Air dehumidification

Decreasing humidity is the most challenging of all the air manipulations. Absorption dehumidifiers are readily available, they use a water absorption material which once saturated has to be replaced or dried out. This kind of dehumidification would be more appropriate for precise control of air properties, however, an immediately accurate humidity level is not crucial for plants. Use of an absorption dehumidifier would increase the complexity of Klimabox construction and cost so condensation dehumidification is used.

Condensation happens on the CCU's evaporator plates when the evaporator temperature is lower than air's dew point temperature. Although the air is momentarily dehumidified, water molecules are still present in the same number inside the chamber; when the evaporator temperature rises, water drops evaporate again. Water is removed from the Klimabox only when so much condensation occurs that it flows out through a drain. The lack of efficient dehumidification is the biggest limitation of the Klimabox design.

2. Physical background

2.1. Heat transfer

Heat spontaneously flows from a body at higher temperature to a body at lower temperature.

Heat transfer occurs by: conduction, radiation and convection.

2.1.1. Thermal conduction

Conduction is a transfer of heat caused by microscopic diffusion and collisions of particles of some material. The amount of heat transferred by conduction between two spots is proportional to a temperature difference between the spots:

$$\vec{q} = U A \Delta T \text{ [W]} \quad (2.1)$$

where $U \text{ [W/K]}$ is an overall heat transfer coefficient, $A \text{ [m}^2\text{]}$ is a heat transfer area and $\Delta T \text{ [K]}$ is the temperature difference.

2.1.2. Thermal radiation

Heat is transferred through electromagnetic radiation. The heat flow is defined by the Stephan-Boltzmann Law:

$$\vec{q} = \epsilon \sigma (T_h^4 - T_c^4) A \text{ [W]} \quad (2.2)$$

where $\epsilon [-]$ is a surface emissivity (always ≤ 1), $\sigma \text{ [W/m}^2\text{K}^4\text{]}$ is the Stephan-Boltzmann constant, $T_h \text{ [K]}$ is a hot object temperature, $T_c \text{ [K]}$ a cold object temperature and $A \text{ [m}^2\text{]}$ is a heat transfer area.

2.1.3. Thermal convection

Convective heat transfer is a transfer of heat from one place to another by the movement of fluids. The heat flow caused by convection is defined as

$$\vec{q} = f \cdot c_p \Delta T \text{ [W]} \quad (2.3)$$

where $f \text{ [kg/s]}$ denotes a mass flow of the fluid, $c_p \text{ [J/kg·K]}$ is a specific heat of the fluid and $\Delta T \text{ [K]}$ is a temperature difference between the two places.

2.2. Psychrometric properties of moist air

This section explains how to calculate psychrometric properties of moist air in a computational fashion, no Mollier diagram or psychrometric charts are used. The section is ordered to give a step by step guideline to calculate humidity ratio, absolute humidity, specific enthalpy or dew point temperature. Nevertheless, saturation water vapor pressure of the air always has to be calculated first. If the calculations are followed as given in this guideline, all the necessary values will always be ready and formulae may always be solvable without any side calculations.

The following calculation process assumes that the dry bulb temperature and the relative humidity of air are available for measurements.

Thorough information about the psychrometric computation process can be found in [22, 4].

Table 2.1 states constants which are not temperature dependent, and some that vary slightly in a temperature range (263 K – 313 K). The variation is however not significant and does not change the final values critically.

Temperature

The temperature is one of the main characteristics describing the state of moist air; it is measured by thermometer. The first calculation of the whole process is the conversion of the temperature measurements to Kelvins using one of the following formulas:

$$T = T_C + 273.15 \text{ [K]} \quad (2.4)$$

$$T = \frac{5(T_F - 32)}{9} + 273.15 \text{ [K]} \quad (2.5)$$

where $T_C \text{ [°C]}$ is the temperature in degrees Celsius and $T_F \text{ [°F]}$ is the temperature in degrees Fahrenheit.

2. Physical background

Constant	Value	Description
T_0	273.15 K	Absolute temperature at 0 °C
P_{sl}	101325 Pa	Sea level standard atmospheric pressure
L	0.0065 K/m	Altitude temperature lapse rate
T_{sl}	288.25 K	Sea level standard temperature
g	9.80665 m/s ²	Earth-surface gravitational acceleration
M	289.3 · 10 ⁻⁴ kg/mol	Molar mass of dry air
R	8.31447 J/mol·K	Universal gas constant
R_w	461.5 J/kg·K	Gas constant of water vapor
R_a	287.1 J/kg·K	Gas constant of dry air
$c_{p,da}$	1005 J/kg·K	Specific heat capacity of dry air
$c_{p,wv}$	1864 J/kg·K	Specific heat capacity of water vapor
h_{we}	2443 · 10 ³ J/kg	Latent heat of water vaporization

Table 2.1.: Constants for calculations psychrometric properties of moist air

2.2.1. Atmospheric pressure

True barometric air pressure is a variable in psychrometric calculations. It should be measured to obtain precise psychrometric properties of air; however, if no pressure measurements are available, altitude related estimation can be used. The following equation shows an average atmospheric pressure altitude relationship

$$P = P_{sl} \left(1 - L \frac{z}{T_{sl}} \right)^{\frac{g \cdot M}{R \cdot L}} \text{ [Pa]} \quad (2.6)$$

where z [m] is an altitude of the measurement station [14]. All the other variables are constants from table 2.1.

2.2.2. Saturation water vapor pressure

Saturation water vapor pressure is an essential value for most of the psychrometric calculations and therefore its calculation cannot be omitted. Saturation water vapor pressure varies when measured over water or over ice; for the purpose of this paper only saturation water vapor pressure over water will be taken in account.

A large number of formulae arose during the past two centuries, beginning with the Clausius-Clapeyron equation in 1834 which first succeeded in formulating the thermodynamic relationship between pressure and temperature in ideal gas. This was followed by Magnus in 1844, who first evaluated an approximation of saturation water vapor pressure for natural temperature range. This approximation made a base for further corrected forms made by Murray in 1967, Bolton in 1981 and Arden Buck in 1981 [4]. Buck introduced an enhancement factor to the

exponential equation, which improved accuracy in a broader range; he later revised it in 1996, making the approximation very precise in a range -80 to 50°C . Golf and Gratch made in 1946 a very complex and precise model usable for steam calculations. Golf later revisited his formula in 1957 and the latter was recommended for use by the World Meteorological Organization in 1988 as the most precise equation present. Many later refinements have been made to these models. However, the difference in precision is for the purpose of this paper negligible. An illustrative precision comparison of all the mentioned formulae and more can be found in [20].

The saturation water vapor pressure formula from Buck (1996) has been chosen reversibility. The equation for saturation water vapor pressure over water P_{wv}^* is

$$P_{wv}^* = \kappa_w a_w \exp\left(\frac{b_w - T_{db}}{d_w} \cdot \frac{T_{db}}{T_{db} + c_w}\right) [\text{Pa}] \quad (2.7)$$

where the enhancement factor κ_w is

$$\kappa_w = 1 + 10^{-4} \left(7.2 + \frac{P}{100} (0.032 + 5.9 \cdot 10^{-6} T_{db}^2) \right) \quad (2.8)$$

and $P [\text{Pa}]$ is an atmospheric pressure, $T_{db} [^\circ\text{C}]$ is a dry bulb temperature, $a_w = 611.21$, $b_w = 18.678$, and $d_w = 234.5$.

2.2.3. Partial water vapor pressure

Water vapor is one of several gaseous constituents of air. Apart from the principal ones - nitrogen (78.1% of volume) and oxygen (20.9%), there are also argon (0.9%), carbon dioxide (0.03%) and multiple other gases including water vapor (0.07%). Each exerts its own partial pressure and the sum of the pressures makes up the total (barometric) air pressure. Partial water vapor pressure is directly related to relative humidity by the following equation

$$P_{wv} = \frac{\phi_{RH}}{100} P_{wv}^* [\text{Pa}] \quad (2.9)$$

where $\phi_{RH} [\%]$ is the measured relative humidity of air and P_{wv}^* is saturation water vapor pressure calculated in the previous step.

When there is a difference in concentration of water vapor between two points, there will be a corresponding difference in the partial pressures. This will cause a flow of the gas from the point of higher concentration to the lower. When a partial pressure difference exists between two sides of a material, the gas involved will diffuse through the material until the partial

2. Physical background

pressures of that gas are equalized. The rate of flow/diffusion is dependent on the partial pressure difference.

When water evaporates into air, the water molecules in the vapor expand. Air pressure is directly related to the number of gas molecules per m^3 , so water vapor pressure results from the number of water vapor molecules per m^3 . The greater the moisture content of air, the greater the vapor pressure. Thus water vapor pressure is directly related to absolute humidity.

Water vapor pressure also directly affects evaporation rate. If the water vapor pressure in air is already very high, it is more difficult for water molecules to break free from a surface of liquid and enter air as water vapor. The point at which absolutely no evaporation occurs is when the partial water vapor pressure is equal to the saturation water vapor pressure [22].

2.2.4. Humidity ratio

Humidity ratio, also referred to as a specific humidity ratio, is the ratio of water vapor mass m_v to dry air mass m_d in a particular volume V . It can also be expressed using partial water vapor pressure as where P_{da} [Pa] is a pressure of completely dry air and P [Pa] is the barometric pressure of air [21].

However, humidity ratio should not be mistaken for a mixing ratio or specific humidity, which, as used in chemistry, is defined as the ratio of water vapor mass to the total air mass in a system.

2.2.5. Specific volume

Specific volume of moist air per unit mass of dry air and water vapor is an inverted value of moist air density

$$V_S = \frac{1}{\rho_{ma}} \left[\frac{m^3}{kg} \right] \quad (2.10)$$

where ρ_{ma} [kg/m³] is a density of moist air calculated as

$$\rho_{ma} = \frac{P - P_{wv}}{R_a T} + \frac{P_{wv}}{R_w T} = \rho_{da} \frac{1 + W}{1 + W \frac{R_w M}{R}} \left[\frac{kg}{m^3} \right] \quad (2.11)$$

where W [kg/kg] is the humidity ratio derived in the previous step and R , M , R_a , R_w are constants stated in the table 2.1. ρ_{da} [kg/m³] is a density of dry air calculated from the following formula

$$\rho_{da} = \frac{P}{\frac{R}{M} T} \left[\frac{kg}{m^3} \right] \quad (2.12)$$

where P [Pa] is barometric pressure of air and T [K] is air temperature.

2.2.6. Absolute humidity

Absolute humidity is the amount of water vapor present in a unit volume of moist air. It can also be referred to as a water vapor concentration in the air mixture and the relation is

$$\varrho_{AH} = \frac{m_{wv}}{V_{ma}} = W \cdot \rho_{ma} \left[\frac{\text{kg}}{\text{m}^3} \right] \quad (2.13)$$

Absolute humidity changes with air temperature or pressure changes; this can be inconvenient for chemical engineering calculations. As a result, absolute humidity is defined in chemical engineering as mass of water vapor per unit mass of dry air, also known as the mass mixing ratio, which is more rigorous for heat and mass balance calculations. Mass of water vapor per unit volume as defined in the equation 2.13 would be then defined as a volumetric humidity. There can be a potential confusion in this term. In this paper absolute humidity refers to the so-called volumetric humidity defined in the equation 2.13.

2.2.7. Dew point temperature

A dew point temperature (sometimes called simply dew point) is the temperature below which the water vapor present in air starts to condense into liquid water. Such a temperature can be calculated using an inverse saturation water vapor pressure equation 2.7. It provides a temperature when the actual partial water vapor pressure becomes equal to the saturation water vapor pressure. The dew point temperature is expressed as

$$T_{dp} = \frac{d_w}{2} \left(b_w - s - \sqrt{(b_w - s)^2 - 4 \frac{c_w s}{d_w}} \right) [^\circ\text{C}] \quad (2.14)$$

where

$$s = \ln \left(\frac{P_w}{100 \cdot \kappa_w} \right) - \ln(a_w) \quad (2.15)$$

enhancement factor κ_w is

$$\kappa_w = 1 + 10^{-4} \left(7.2 + \frac{P}{100} \left(0.032 + 5.9 \cdot 10^{-6} T_{db}^2 \right) \right)$$

and $a_w = 6.1121$, $b_w = 18.678$, $c_w = 257.14$ and $d_w = 234.5$.

2. Physical background

Air temperature and dew point temperature together is an alternative definition of a psychrometric state of air.

2.2.8. Specific enthalpy of moist air

An enthalpy of moist air includes an enthalpy of dry air, a sensible heat, and an enthalpy of evaporated water, a latent heat. Specific enthalpy is formulated as

$$h = h_{da} + W \cdot h_{wv} \left[\frac{\text{J}}{\text{kg}} \right] \quad (2.16)$$

where W is the humidity ratio calculated previously, h_{da} is the sensible heat, h_{wv} is the latent heat. The sensible and latent heat are expressed as

$$h_{da} = c_{p,da} \cdot T_{db} \left[\frac{\text{J}}{\text{kg}} \right] \quad (2.17)$$

$$h_{wv} = c_{p,wv} \cdot T_{db} + h_{we} \left[\frac{\text{J}}{\text{kg}} \right] \quad (2.18)$$

where T_{db} [°C] is the air dry bulb temperature and $c_{p,da}$, $c_{p,wv}$, h_{we} are constants that can be found in the table 2.1.

2.2.9. Psychrometric calculation example

A stock-house is situated 364 m above sea level. Relative humidity and temperature of air are measured, although atmospheric pressure is not. What is the absolute humidity ϱ_{AH} and dew point temperature T_{dp} , when the relative humidity measures is $\phi_{RH} = 64\%$ and the temperature is $T_{db} = 21.3^\circ\text{C}$?

1. Atmospheric pressure (sec. 2.2.1): $L = 364\text{ m}$; $P = P_{sl} \left(1 - L \frac{z}{T_{sl}}\right)^{\frac{g \cdot M}{R \cdot L}} = 97.021 \cdot 10^3\text{ Pa}$
2. Saturation water vapor pressure(sec. 2.2.2): $T_{db} = 21.3^\circ\text{C}$; $P_{wv}^* = \kappa_w a_w \exp\left(\frac{b_w - T_{db}}{d_w} \cdot \frac{T_{db}}{T_{db} + c_w}\right) = 2543.7\text{ Pa}$
3. Partial water vapor pressure(sec. 2.2.3): $\phi_{RH} = 64\%$; $P_{wv} = \frac{\phi_{RH}}{100} P_{wv}^* = 1628\text{ Pa}$
4. Specific humidity(sec. 2.2.4): $W = 0.62198 \frac{P_{wv}}{P - P_{wv}} = 10.6 \cdot 10^{-3}\text{ kg}_{wv}/\text{kg}_{da}$
5. Density of moist air(sec. 2.2.5): $\rho_{ma} = \rho_{da} \frac{1+W}{1+W \frac{R_w M}{R}} = 1.1406\text{ kg}/\text{m}^3$
6. Absolute humidity(sec. 2.2.6): $\varrho_{AH} = W \cdot \rho_{ma} = 12.107 \cdot 10^{-3}\text{ kg}/\text{m}^3 = \underline{12.11\text{ g}/\text{m}^3}$
7. Dew point temperature(sec. 2.2.7): $T_{dp} = \frac{d_w}{2} \left(b_w - s - \sqrt{(b_w - s)^2 - 4 \frac{c_w s}{d_w}}\right) = \underline{14.22^\circ\text{C}}$

Part II.

Data acquisition and command signaling

3. Klimabox control panel

The klimabox control panel is a microelectronic device designed and manufactured for use in a climatic chamber unit. Users of Klimabox are able to manage climatic chamber settings, monitor its current state and operate outputs on the control panel. Sensors, part of the control panel hardware, are the first link in the chain of the whole control process, whereas the microprocessor's output circuits are the very end of the chain. Further it provides communication with the computer.

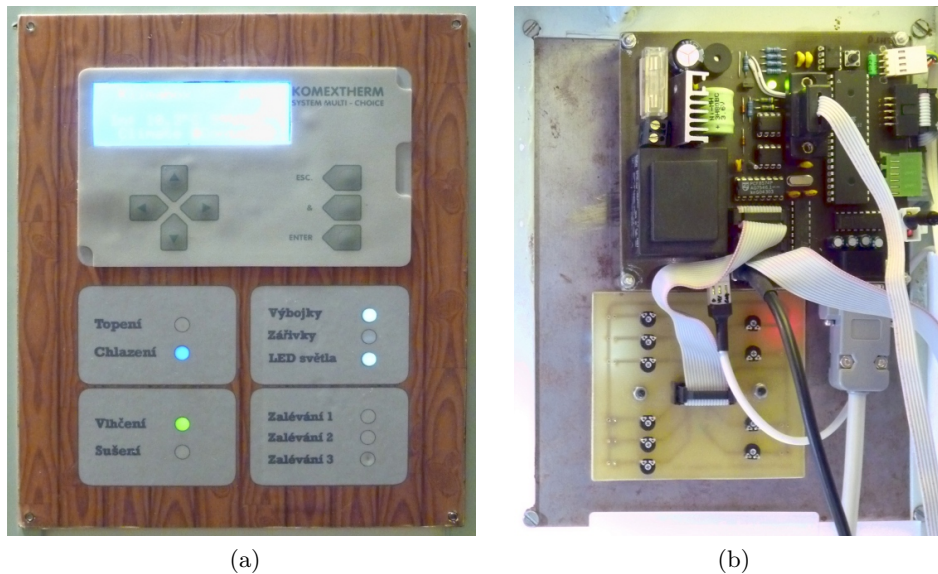


Figure 3.1.: Klimabox control panel front face (a) and its electronics (b)

3.1. History

The idea of developing a new control panel for the klimabox arose in October 2004 stimulated by the upcoming graduation project at high school. The old control panel, picture 3.2 (a), was an analog electronic device having two sets of rotary handlers, one for day temperature, the other for night temperature. Temperature was set by a dry bulb temperature handler, the relative humidity by a wet bulb temperature handler. In other words, the climate inside

3. Klimabox control panel

the chamber could be set only for two time intervals and an operator had to convert his/her climatic preset into dry bulb/wet bulb temperatures. There I saw a space for improvement, a space for a new digital control panel.

3.2. Circuit description

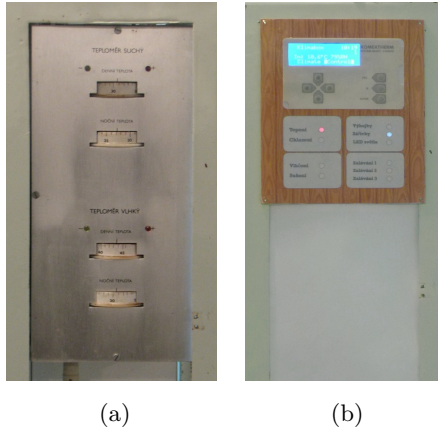


Figure 3.2.: Klimabox control panel history.
Old control panel from 1985 (a)
and the new one from 2009 (b)

The control panel hardware, schematically plotted at fig. 3.3, was designed from scratch. Nevertheless it was inspired by an evaluation kit, which I built previously. A core element of the device is an Atmel AVR 8-bit RISC microcontroller ATmega 32 sitting in the middle of the motherboard. A coupled sensor of temperature and relative humidity SHT11 is connected directly to microcontroller ports. External EEPROM and Real Time Clock circuit are connected by an I²C bus. LCD display, keyboard, RS-232 bonding circuit and output power amplifier complete the list

of important connected devices. A thorough description of the hardware design and its functionality can be found in my graduation project technical documentation [5] or my bachelor thesis [6].per

Later on, in 2012, another SHT11 sensor for laboratory air measurements was added and a DS18S20 thermometer was attached to a cooling coil unit evaporator. Both additional sensors were added to aid both system identification and controller.

3.3. Control panel software

Right after building the hardware in 2005, I needed to learn ANSI C for micro-controllers. The microchip software is what really makes the device alive, but it also is the most complicated and time-consuming part of the project. The panel software in 2005 was capable of communicating with SHT11 sensor, it also provided a simple menu for users to set climate preset parameters. It was written in a free limited version of the commercial CodeVision[®] software, which provides some closed source peripheral drivers in the package. The code optimization of the compiler

software was on the other hand rather poor and the complete code finally consumed the whole 94% of the internal program memory.

A simple 3-state controller was implemented in 2009, when the control panel was also connected for the first time to the actual climatic chamber. It was finally possible to plot first control trajectories, but the microcontroller software was consuming 97% of the program memory.

However, the communication protocol proposed in my bachelor thesis was still left to be implemented. Such a protocol, loss and error free, was necessary for the more sophisticated control that I mean to deploy in this Master's thesis. Therefore it was inadmissible to continue developing the old poorly written code and it was only possible to re-implement the whole software from the beginning.

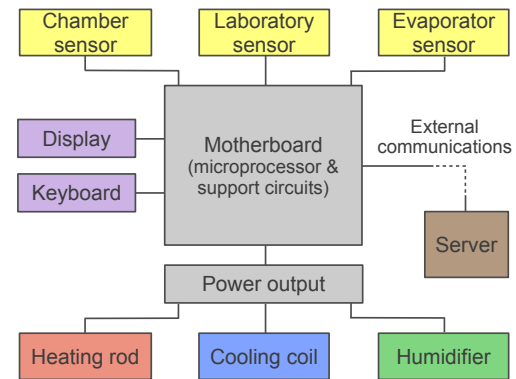


Figure 3.3.: Klimabox control panel hardware block diagram

New klimabox control panel software implementation was done using an open source GNU GCC compiler, and started in 2010. All the old functionality and many new features have been realized since then; only this time aiming for code lightness and lucidity. The total code size now is 74.7% and the software is much more feature rich and reliable.

The following sections will show the important parts of the control panel software such as periphery drivers, measurement storage, screen routines and the communication protocol.

The basic routines are obtained from an open source AVR C library [3]. Makefile template was obtained from the open source WinAVR project.

3.3.1. Periphery drivers

Even though the microchip is the most important element of the whole panel, it is the peripherals which specialize the device for use with the klimabox. Temperature and humidity sensor SHT11 records the air properties and the real time clock circuit looks after accurate timing. Communication with user is shown on an LCD display; the user's feedback enters through a keyboard. UART/RS-232 interface realizes the communication with the computer.

3. Klimabox control panel

3.3.1.1. Basic libraries

This section contains low-level peripheral drivers description. Many of the basic drivers are from [8].

3.3.1.1.1. UART UART, an acronym of Universal Asynchronous Receiver/Transmitter, is a piece of microcomputer hardware responsible for serial communication. Standard input and output streams are redirected into the low level `UART_GETCHAR` and `UART_PUTCHAR` functions; doing this, a programmer can use the usual `PRINTF/SCANF` function to write/read the UART data. Low level UART handling functions are based on Joerg Wunsch's library.

The high level communications protocol is thoroughly described in chapter 5.

3.3.1.1.2. LCD Basic routines for interfacing an HD44780U-driver-based text LCD display are based on Peter Fleury's LCD library [8]. Optional display text wrapping and scrolling was implemented later. The HD44780 LCD driver also contains space for 8 user-defined characters. They were used to extend the in-built ANSI character map by the Code page 852 characters; LCD can now display Czech letters properly. This functionality may not be very well exploited yet, but it may be useful in future projects.

3.3.1.1.3. I²C I²C (Inter-Integrated Circuit) is a multi-master serial single-ended communication bus invented by Phillips that is used to attach low-speed peripherals to a motherboard. ATmega32 microcomputer has an in-built hardware support (called TWI) for I²C bus. Needless to say, such a piece of hardware would make the communication much easier and safer but I was not aware of this when I was designing the panel hardware. The I²C bus wires are not connected to the right TWI I/O pins, instead they are connected to generic I/O pins so a software emulation of the bus had to be used. I²C routines are based on Peter Fleury's assembler I²C library [8].

3.3.1.1.4. 1-Wire DS18S20 thermometer is connected using a 1-wire bus. It is a device communication bus system designed by Dallas Semiconductors that provides low-speed data, signaling, and power over a signal wire. 1-wire is similar in concept to I²C only with lower data rates and longer range; that makes it perfect for use in sensor nets. The entire 1-wire communication is performed by Peter Dannegger's 1-wire library with an extension for DS18x20 sensor family by Martin Thomas Thomas [18].

3.3.1.2. Real Time Clock

Accurate time is kept by a real time clock (RTC) circuit PCF8583. The baseline communication protocol is the I²C but any high-level C routines could be found. A complete service library had to therefore be built. The library contains functions to set and retrieve time, date, status register and alarm settings.

PCF8583 has a status register in which, apart from other information, there is a second and minute change flag. Reading the status register and counting the seconds programmatically, instead of reading the whole time register, saves I²C bus traffic and computation time. Status register is read every 2 ms. The most convenient way of receiving a new-second-flag would of course be using an interruption; this approach, however, requires another wire between RTC and processor and the PCB made at high school does not offer such a wiring.

There is also 236 bytes of unused RAM memory in the chip that has been used for data storage, see section 3.3.2.

3.3.1.3. SHT11 sensor

The SHT11, from a Swiss company Sensirion®, is a factory pre-calibrated temperature and humidity digital sensor. It does not need a single external component to use it in combination with an AVR microcontroller.

It has a 2-wire interface that, unfortunately, does not conform with I²C standards. The sensor provides checksums to ensure integrity

of all measurement data. The Sensirion way of CRC8 implementation is however not standard either; they use CRC8 with all bytes bit-reversed.

The communication library is based on Guido Socher's implementation of the Sensirion protocol [16]. It was further refined by Jaroslav Pajskr, however, many features were still left to implement.

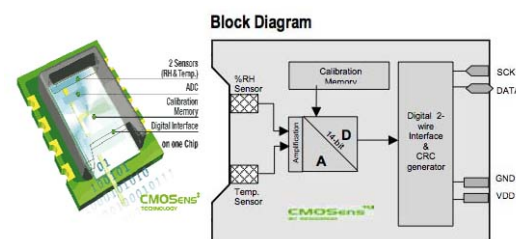


Figure 3.4.: SHT11 hardware block diagram

3.3.1.3.1. Batch processing vs. state machine SHT11 uses a standard double integration A/D converter that outputs a number of pulses counted during the conversion period. It takes about 300 ms to perform one high precision measurement.

3. Klimabox control panel

Batch reading is a sequence of instructions that once executed returns a temperature and humidity reading at the end. Although it is the simplest way to retrieve data, the microprocessor does nothing but wait during the sensor's conversion time. Instead of doing something useful, the main process is trapped in a for-loop and repeatedly checks for the end of conversion. Completion of one measurement blocks the processor for approximately 0.6 s which can lead to losses of non-interrupt events, e.g., keyboard strokes.

State machine reading, an idea originated by Guido Socher [16], offers a way to overcome the previously mentioned issues. Once the measurement function is invoked, it just sends the measurement start sequence to the sensor, increments a state variable and returns back to the main routine. Every time the main routine enters the measurement function afterwards, it reads the state variable and it already knows that the conversion has started. It only checks for the end of conversion and if the conversion is not complete yet, it runs back into the main cycle again.

Two SHT11 sensors are repeatedly read in the main routine; the use of the state machine readings was therefore a must.

3.3.1.3.2. Checksum and swap byte CRC8 checksum, as implemented by Sensirion, requires to bit-reverse all the bytes, i.e. bit 7 swaps with bit 0, bit 6 with bit 1 etc. Every time the microprocessor reads a measurement (temperature and humidity separately) three bytes are received and CRC is computed for them. It means the microprocessor has to bit-reverse byte six times per one complete measurement.

A first hand solution to a byte bit-reversing function would be a for-loop, where the bit would swap iteratively. However, the execution of such a function lasts a long 67 cycles. After some searching, an assembler routine taking only 15 cycles was found (at www.avrfreaks.net). Another 7 clock-cycles are saved each time by inlining the assembler function into the C code - the call/return overhead is removed.

It was also noticed that Sensirion uses the same CRC polynomial (0x8C, initialized with 0x00) as Maxim Instruments. Computation of this routine has already been optimized in the `_CRC_IBUTTON_UPDATE` function in the `CRC16.H` header from the AVR-LIBC library. A code size was saved and some speed was gained by using this inline assembler function.

3.3.1.3.3. Raw data conversion As mentioned before the sensor returns a number of pulses counted during the double-integral conversion. This raw datum is later converted to give a meaningful output. The relative humidity and temperature values are calculated in times-ten

format, i.e., the temperature value 245 means 24.5 °C. This format allows reasonable precision while enabling all the calculations to be done only by integer arithmetics, hence saving a huge amount of code size by excluding the float arithmetics libraries. Detailed description on conversions is given in section 6.

3.3.2. Data storage

Measurements are taken at every measurement interval and therefore the klimabox control panel has to be able to store them. To do so efficiently a special measurement storage structure was developed. The structure contains the following items:

1. Temperature values (chamber, laboratory and evaporator temperature) are stored in a previously mentioned times-ten format (245 means 24.5 °C) and offset by 200 to give an unsigned number. Variable size is 10 bits, which is enough for a range -20 to 82.4°C .
2. Relative humidity values (chamber, laboratory) are also stored in a times-ten format. Variable size is 10 bits giving a range $0 - 100\% \text{RH}$.
3. Number of measurement intervals passed since the last data transfer to a computer. Instead of saving the entire measurement time, which would take at least 22 bits, the measurement time is stored just as the number of intervals passed since one reference time leaving the possibility to squeeze the information into just 8 bits. Up to 256 measurements can be stored.
4. A 1 bit output state variable per every output channel (heating, cooling, moistening, etc.), 9 bits in total.

All data in the structure take up 67 bits, i.e., 9 bytes are necessary to store one measurement.

Each measurement is converted into this structure and later stored into one of the following memory places:

1. First 50 measurements are stored into a reserved memory block (450 bytes) in the microchip's RAM memory.
2. Next 26 measurements will fill out the free RAM memory region of the RTC circuit (236 bytes).
3. Further 28 measurements can be stored in the external EEPROM memory circuit (256 bytes).
4. If that is not enough, up to 152 extra measurements can be held in the microchip's internal EEPROM memory.

3. Klimabox control panel

The server is regularly reading every new measurement right after it is taken. It means that normally only the chip's internal RAM memory gets used in the storage process. In a case of server maintenance or fault the measurements gradually also fill the other memories. When the number of stored measurements exceeds a specified threshold the control panel starts sending read-measurement-requests (command 5.4.1) to the server. All the measurements are transferred and saved in the server's database when the server comes back to life.

Climate daily profile is stored in the chip's internal EEPROM memory.

3.3.3. Compilation

The whole program is compiled under UNIX by a GNU GCC compiler extension for Atmel AVR microcontrollers, the AVR-GCC. There is also an alternative for Windows OS users called WinAVR. Both require a makefile to specify data and options for the compiler. The following paragraph contains a list of options, which were found to play a significant role in the compilation process.

3.3.3.0.4. Compiler switches

-s optimize for size

-fpack-struct packs all structure members together to reduce occupied size. For example, if a structure contains eight bit variables, it is stored in only one byte instead of eight bytes.

-ffunction-section instructs the compiler to place each function (including static ones) in its own code section. During the section linking, every function reference causes the function container section to be linked into the program. As a result, only functions that are referenced at least once from the main or interrupt routines are linked into the final program; thus reducing code size when generic libraries with many extra functions are used (e.g. math library).

-fdata-sections analogous to the previous switch only for data variables (global or static).

3.3.3.0.5. Linker switches

-Wl,-gc-sections instructs the linker to remove unused sections; often used together with **-FFUNCTION-SECTION** and **-FDATA-SECTIONS** switches.

-Wl,-relax by default, the linker links functions using a full CALL statement, which is wasteful when two functions are near each other. Relaxation saves code size by replacing CALL statements with a smaller RCALL whenever possible.

4. Server

Klimabox is positioned in a laboratory which is a standalone building in the middle of our garden. The place is not exactly habitable so a technique of how to work with the klimabox from the warmth of home was developed.

Klimabox was supplied with a computer, more accurately called a server for the lack of monitor, keyboard or mouse. The Klimabox control panel, its digital heart, is connected to the server in two ways; first of all by a programmer which allows the changing of the micro-controller's program remotely. The second connection is realized by a serial link (RS-232) between the two sites.

The server played an important role in the development of the project and plays a vital role in the control process itself. Figure 4.1 shows the server (the applications running on it) standing as an intermediate between the klimabox and its controller.

Two main applications running on the server will be described in this chapter; a klimabox daemon and the database used. The klimabox daemon (sec. 4.1) is a program that encloses the communication with klimabox into a simpler form, allowing it to be remotely accessible, and takes care of data storage. The data are stored in an external MySQL database whose details are explained in section 4.3.

Part of the development also took place in the USA where the Klimabox's remote accessibility really earned its name.

4.1. Klimabox daemon

Klimabox daemon, called KLIMABOXD, is a klimabox driver application running in the background in the klimabox connected computer. It is a program written to envelope the basic klimabox communication and bring a simple uniform control interface for end-user applications.

4. Server

Klimaboxd is a C application compiled in GCC compiler under Linux OS and it was developed from scratch for the purpose of this thesis.

The following sections will explain important and interesting parts of the klimaboxd implementation: process daemonization, peripheral drivers, configuration file, activity logs and possible ways of future development. The user interface - the main feature klimaboxd provides - is described in section 4.2.

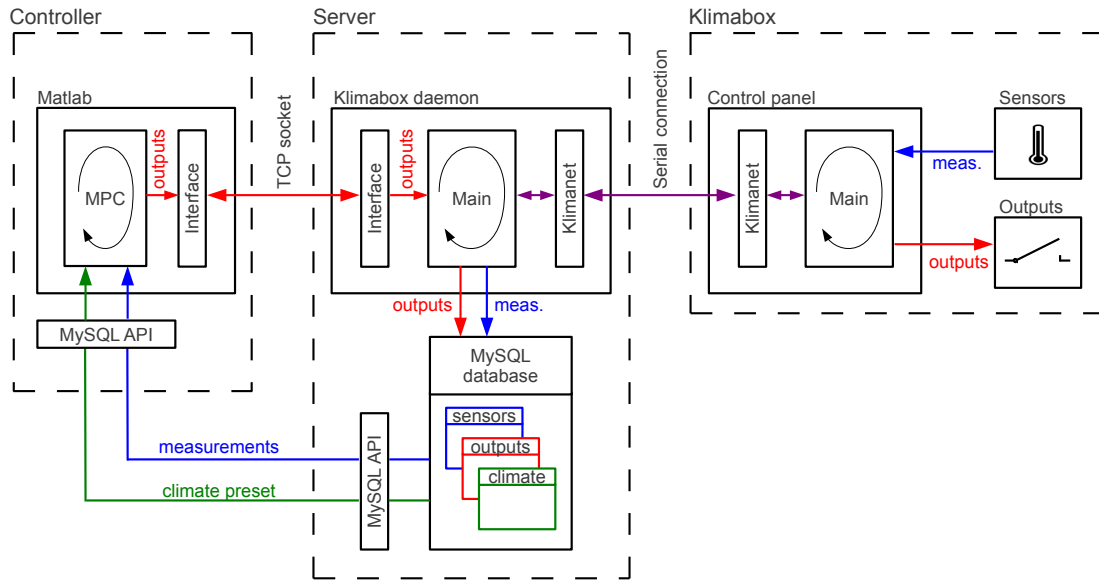


Figure 4.1.: Control process - communication scheme

4.1.1. Daemonizing a process

A daemon process is just like any other process, the only difference is that it does not have a controlling terminal. This is a major aspect of creating a daemon process and it is achieved by a simple technique:

1. The program is started from the terminal (child process of the terminal) and has its own process ID (parent process). The process hierarchy looks like

TERMINAL → PARENT PROCESS

2. Child process is created from within the above parent process using `FORK()`. The process hierarchy now looks like

TERMINAL → PARENT PROCESS → CHILD PROCESS

3. Parent process is terminated and therefore the child process becomes an orphan. The process hierarchy is now

$$\emptyset \rightarrow \text{CHILD PROCESS (orphan)}$$

4. Orphan processes are taken over by the INIT process, i.e., it is no longer related to the terminal process. The process hierarchy becomes

$$\text{INIT} \rightarrow \text{CHILD PROCESS}$$

5. Close input, output and error streams of the new process.

The program is now running detached from the initial terminal. The terminal is freed and can be closed down without any concerns; the daemon process will continue operating flawlessly. To run the klimaboxd as daemon use the “-d” switch at command line.

When klimaboxd is used as daemon all notification/debug messages are issued to a SYSLOG¹ - concretely into the the user log accessible on Unix OS by default at “/var/log/user.log”.

In addition, to assure a proper exit on daemon process closure (most probably by invoking a KILL command) a SIGTERM signal action is registered and appropriate exit function is performed before quitting.

4.1.2. Periphery drivers

Klimabox daemon accesses a variety of computer peripheries.

As said previously, the computer communicates with the klimabox control panel by the serial port. What was however not mentioned is that the Klimabox device itself can be remotely turned on and off using a computer’s parallel port. Klimaboxd has to handle this periphery as well.

Database is accessed using MySQL database C connector.

Finally the klimaboxd user interface is connectable exclusively using TCP sockets - the means of inter-process network communications.

¹Syslog is a standard for computer data logging.

4.1.3. Configuration file

The daemon's behavior is altered through a klimaboxd configuration file. The following parameters are to be specified in the file:

- Verbose - true/false
- Debug level - specifies detail level of notification message generated due to the following
 - 0: errors only
 - 1: urgent
 - 2: system non-periodic
 - 3: system periodic and acknowledgments
 - 4: detailed
- Serial port address, baudrate - for example “/dev/ttyUSB0”, 38400
- Parallel port address - for example “/dev/parport0”
- Database server, user and password
- User interface TCP port, password and maximum number of contemporary connections
- Log file location

4.1.4. Logging

Klimaboxd logs its activity according to the setting from the configuration file. The amount and detail of log issued is adjusted by a debug level described in section 4.1.3.

The log can be written down into a log file (e.g. “./log.txt”). In a daemon mode, the log is also recorded in SYSLOG.

4.1.5. Future development

Klimaboxd is prepared to connect to the Matlab engine. This feature was implemented during early stages of development as one of the possible ways to connect the controller to the klimabox. The user interface was later implemented instead, but future development will probably reincarnate a direct connection to the Matlab engine.

Declaring an controller optimization problem in a optimization language and connecting straight to an optimization solver could be another plausible way of future development.

4.2. Klimabox user interface

Klimabox daemon program - klimaboxd - provides a user interface connectable locally at the server as well as remotely over a network. Whilst the first option is useful when setting up or debugging the connection to the Klimabox, the second option - the remote accessibility - is what really brings value to the klimaboxd daemon. TCP sockets are used for the connection whether from localhost² or from a remote network location. Thanks to the klimaboxd the climatic chamber is operable from any Internet-connected computer.

The remote accessibility however brings security issues and connection handling duties. To prevent anyone connecting to the Klimabox an access to the interface is password protected (password is to be defined in the configuration file, sec. 4.1.3). A remote manual connection to the klimaboxd user interface using Telnet client is presented at figure 4.2.

```
jiri@jiri-laptop:~$ telnet klima-server 11112
Trying 192.168.1.100
Connected to klima-server.
Escape character is '^]'.
    Welcome to Klimaboxd (v. 1.5)
Enter password:
Welcome. For help type 'help'
:
```

Figure 4.2.: Remote connection to the klimaboxd user interface

User or end application, controller for example, controls the Klimabox using a defined set of commands; the commands are listed in table 4.1. The first four commands control the behavior of the user interface itself, the rest is purely klimabox-related. The table also shows a list of admissible parameters (in curly brackets).

Verbosity of the user interface can be altered by a VERBOSE command. Users will value text output (verbose on), whereas applications profit more from numbered responses (verbose off).

Any activity at the klimaboxd user interface is logged into the database (for more information on database structure see section 4.3).

4.3. Database

A database is a structured collection of data. In the case of the Klimabox, MySQL, an open source database management system was used, mostly because it easy to deploy on a

²In computer networking, localhost means basically “this computer”.

4. Server

Command	Short	Description
help	h	Display table of commands.
quit exit	q	Close the connection.
verbose {on off}	-	Verbose output or code response. (def.: on)
shortCmd {on off}	-	Enable short versions of commands. (def.: off)
powerKlimabox {on off}	k	Turn the Klimabox on/off.
getStatus	s	Get runtime status of the Klimabox.
lastMeasurement	m	Display the latest measurement received.
getOutputs	p	Display actual state of outputs.
setOutput {cool,heat,humi,ft1,ft2,gt1,gt2,tf,bf}	i	Turn the specified klimabox actuator on.
clearOutput {cool,heat,humi,ft1,ft2,gt1,gt2,tf,bf}	o	Turn the specified klimabox actuator off.
setMeasurementInterval {#sec.}	d	Set new measurement interval.
setClimatePreset	c	Load climate preset into the Klimabox.
setBacklight {#sec.}	l	Turn the Klimabox's LCD back-light on.
setBeep {#10 ms}	b	Turn the Klimabox's piezo-siren on.
setText {"text"}	t	Display text on Klimabox's LCD.

Table 4.1.: Klimabox user interface commands

Linux server, it has connector libraries in C programming language and because it is well documented; it is also the world's most used database management system. Database role is graphically pictured at 4.1.

A database (KLIMABOX_DB) was created to hold the klimabox runtime information, historical measurement data, historical output data, climatic preset data and the klimanet (communications protocol, sec. 5) and user interface log messages. The data is stored in the following tables:

1. SENSORS - contains temperature and relative humidity of the chamber and laboratory for each measurement time as well as temperature of the cooling coil evaporator and measurement interval.
2. OUTPUTS - contains binary power state of heater, cooling coil unit and humidifier as well as the state of fluorescent tubes, gas tubes and top and bottom fans.
3. PRESET - accommodates climatic preset data combined from temperature, relative humidity and fluorescent and gas tubes setting for every half an hour.
4. DIGITEMP - holds laboratory and outside temperature measured by extra thermometers connected directly to the server.
5. CONTROLLER - contains historic controller statistics.
6. KNET_LOG - is a table listing complete detailed klimanet communication history.

7. UI_LOG - contains full user interface activity log.

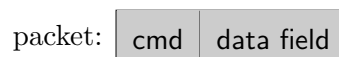
The database is remotely accessible using any MySQL client, however, some created users like MATLAB_USER are not privileged to do more than select, update or insert data in the database.

5. Communications protocol—Klimanet

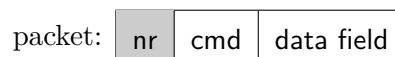
Communications between the computer and the klimabox control panel is provided by a communications protocol called KLIMANET. It provides a transparent error-free connection between two ends of a communication line. The computer (server) is always the master device on the klimanet and directs the traffic. The klimabox daemon (sec. 4.1) together with klimanet creates a virtual periphery from the klimabox device; klimabox is then easily connectable over the network. To allow such a functionality the communication protocol has to be reliable, fast, error-free and feature rich.

Klimanet is a multilayer protocol with the following layers according to the OSI model:

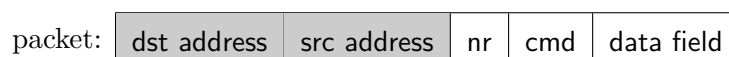
Application layer provides high-end commands for applications to use. This layer defines how the data are organized into packets; the first byte is always a code specification of command, all the following bytes are organized commands data. The exact structure of all commands is described in the sections 5.1, 5.3, 5.4.



Transport layer provides communications between applications ensuring that all data packets arrived and that they arrived in the correct order. To achieve that a packet number is attached to the packet. The receiving side sends back an acknowledgment packet with the same packet number after every successful reception. In the case of packet loss, the sending side is able to detect the failure and retransmits the lost packet.



Network layer incorporates software addressing the communication. The packet from the transport layer is given a header with a source and a destination address. The klimanet protocol can currently operate up to 255 devices.



Data link layer makes the communication error-free. The packet from the previous layer is added with a start character to the front and check-sum, stop character and some void bytes to the end. The check-sum used is a 16 bit Cyclic Redundancy Check (CRC). The check-sum is divided into two bytes where the least significant byte (lsb) is transmitted first and the most significant byte (msb) second. Such a data object is called a frame. All devices use the same shared bus, there is therefore no need for physical addressing in the frame.



Physical layer transmits and receives data link frames over the hardware circuits and wires to all devices on the bus. The hardware standard used is the RS-232 (two ends only), although when a need for multipoint communications emerges the physical connection could be switched to RS-485, USB, etc. The signal specification of the RS-232 standard is described at the figure 5.1.

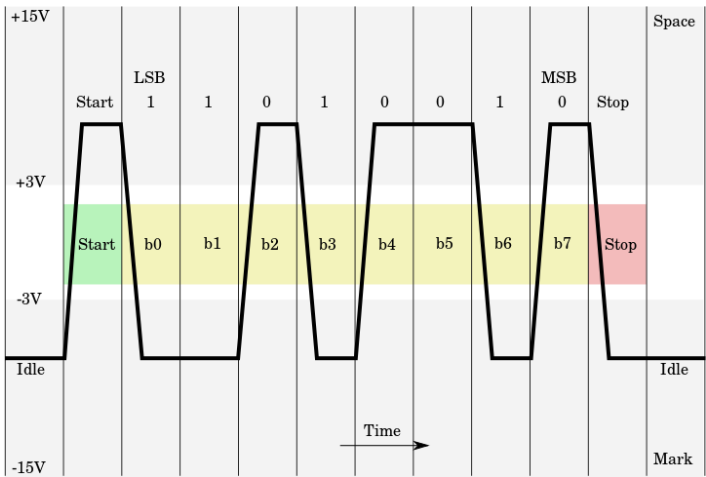


Figure 5.1.: Diagram of RS232 signaling. Credits: commons.wikimedia.com

All communications have request/response style. Every data transfer over klimanet is initiated by a master packet and the slave takes an action upon it. This doctrine can only be violated by special slave request packets that are used to communicate asynchronous events of the klimabox control panel.



Figure 5.2.: Klimanet frame summary

5.1. Klimanet commands

Klimanet provides a set of commands that allow communication with the klimabox control panel. Three major groups can be derived from the command purposes:

- Control commands - setting values into slave devices,
- Request commands - reading information from slave devices,
- Slave request commands - slave communicates an asynchronous event to the master.

Unless stated otherwise, all variables used in command structures are one byte long (8 bits). The $[xB]$ symbol placed behind a variable specifies the byte length of the variable. For example, $[2B]$ denotes a variable to be 2 bytes long (16 bits).

5.2. Control commands

Commands described in this section set values into the Klimabox control panel. Slave reacts to this type of command by a uniform response described in section 5.2.9.

5.2.1. KNetDumpCmd

Command used only to test the klimanet functionality. Its minimal size makes it easier to spot the packet on oscilloscope when resolving hardware problems.

command	data field
0x00	—

5.2.2. KNetSetOutputs

Klimabox control panel actuators are commanded through this instruction.

command	data field									
0x11	ON_OFF	COOL	HEAT	HUMD	FT1	FT2	GT1	GT2	TF	BF

ON_OFF When set to 1, the outputs whose corresponding variable is set to one will be turned on. Setting **ON_OFF** to 2 will result in turning off the corresponding outputs.

COOL Cooler output, **HEAT** Heater, **HUMD** Humidifier,

FT1 Fluorescent tubes - first half, **FT2** Fluorescent tubes - second half,

GT1 Gas tubes - first half, **GT2** Gas tubes - second half,

TF Top section fan, **BF** Bottom section fan.

5.2.3. KNetSetClimPreset

A command serving to load climate presets from the computer into the control panel. Climatic preset data contain four channels (temperature, humidity, fluorescent tubes, gas tubes). Each channel is transferred separately and in a special format because of a maximal packet size issue. The temperature channel, for example, contains 48 values (setting for every half an hour in one day) and its transfer would take 96 bytes when sending uncompressed which would exceed the size of klimanet buffers in the control panel. The temperature is however only a 10 bit value, so a compression can be applied. The compression technique developed lies in concatenating variables bitwise and dealing them into 8 bit chunks for transmission. The stream of data is decompressed in the same fashion on the reception side.

command		data field				
0x12		CHNL	SCV	NCV	VSZ	... DATA STREAM ...

CHNL Channel: 0,1,2,3 corresponds to the temperature, humidity, fluorescent tubes and gas tubes respectively

SCV Index of the first climate value to be transfered

NCV Number of climate values transfered

VSZ Variable size in bits. If a 10bit variables are compressed, value of VSZ is set to 10

DATA STREAM Compressed data stream

5.2.4. KNetSetTempLimits

KNetSetTempLimits is command that instructs the control panel to change its hard temperature limits accordingly.

command	data field	
0x13	TEMP_BOTTOM [2B]	TEMP_TOP [2B]

TEMP_BOTTOM The temperature under which the cooler is turned off automatically (signed integer [2B], lsb first)

TEMP_TOP The temperature over which the heating is turned off automatically (signed integer [2B], lsb first)

5.2.5. KNetSetDateTime

Set date and time defined by *YYYY*, *MM*, *DD*, *HH*, *MM* and *SS* variables into the klimabox control panel. Note that all stored measurements get lost when this command is issued; it is recommended to use KNetGetTempHumi command previously.

command	data field					
0x14	YYYY [2B]	MM	DD	HH	MM	SS

YYYY Year (unsigned integer [2B], lsb first)

MM Month, *DD* Day, *HH* Hour, *MM* Minute, *SS* Second

5.2.6. KNetSetLcdBacklight

This command controls LCD back-light on the control panel.

command	data field
0x15	N_SEC

N_SEC Number of seconds to leave the back-light on. When set to zero back-light is turned off immediately

5.2.7. KNetBeep

Via this command the piezo-siren on the control panel is turned on.

command	data field
0x16	N_10MSEC

N_10MSEC Number of 10 ms to beep.

5.2.8. KNetDispText

Display ASCII string in the x, y position on the LCD.

command	data field		
0x17	X	Y	STR

X column position on LCD, starting with 0 on the left side. Maximum value is 19

Y row position on LCD, starting with 0 at the top. Maximum value is 3

STR Set of characters in Latin 2 - Code page 852 coding

5.2.9. Slave response to the control commands

Every time a control command is issued a response is expected. The response, to be identifiable, preserves the packet number and the command number. If the response does not agree with the issued control command a synchronization procedure is performed. The response message can either report an acknowledgement (ACK) or a negative acknowledgement (NAK). ACK and NAK characters have the following codes

Name	Character	Description
ACK	0x06	Acknowledgement - command received
NAK	0x15	Negative acknowledgement - command rejected

ACK is responded when the packet number has expected value (last packet number plus one), command number is recognized, CRC is fine and the data format matches the specified command. An ACK response looks like

packet num.	command	data field
the same #	the same cmd	ACK

If one of the previously listed checks is not satisfied, a NAK response is responded together with an `error_code` specifying what the error is. The NAK response has the following format:

packet num.	command	data field	
the same #	the same cmd	NAK	error_code

and the `error_code` indicates one of the following scenarios:

Error name	error code	Description
UNKNOWN_CMD	0xFF	Command number was not recognized. No action taken.
ERR_SYNC	0xFE	Packet number is wrong. Synchronization is performed.
CRC_FAIL	0xFD	Cyclic redundancy check (CRC) failed. The same packet is sent again.
CMD_ERROR	0xFB	Data accompanying the command does not fit the command requirements. No action taken.
CMD_EXEC_ER	0xFA	Execution of the command failed. Command dependent actions are performed.

5.3. Request commands

Request commands serve for reading information from the klimabox control panel. The communication has the following layout: Server issues a request packet (5.3.1) and the control panel responds with the data requested.

5.3.1. Request packet

Unless stated otherwise, all requests packets have the same format. It contains a request command number (cmd) and no data in the data field. Such a request packet initiates every data transfer from the klimabox control panel.

command	data field
cmd	—

5.3.2. KNetGetTempHumi

This command serves for saving temperature and humidity measurements along with the current state of outputs and its measurement time to the server's database. When a KNetGetTempHumi command is issued all the measurements stored in the panel are transferred to the server. To see how the control panel stores measurements see the section 3.3.2.

5.3.2.0.6. Request command Request packet of this command is not a standard one. A new measurement interval is sent along in the packet.

command	data field
0x20	INT [2B]

INT Set new measurement interval in seconds. Applicable range of the measurement intervals is 2 s – 1 h. (unsigned integer [2B], lsb byte first)

Note: New interval is set after the last pending measurement is sent. New measurement will occur when the modulus of the day time in seconds and the measurement interval is zero. For example, at 11:41:13 am a new measurement interval - 30 seconds - is sent, the next measurement will be performed at 11:41:30 am.

5.3.2.0.7. Response Two cases can arise when requesting measurements:

1. There are no measurements to read yet. The response will have zero bytes in the data field. It will simply look like this:

command	data field
0x20	—

2. At least one measurement is available. In this case, full 15 byte response is transmitted. The response has the following form (each variable is 2B long)

command	data field								
0x20	T_IN	T_LAB	T_COOL	RH_IN	RH_LAB	#INT	INT	OUT	MORE

T_IN Climatic chamber temperature in °C times 10, offset by 200, e.g., 454 stands for 25.4°C. (unsigned integer [2B], lsb first)

T_LAB Laboratory temperature in °C times 10, offset by 200. (unsigned integer [2B], lsb first)

T_COOL Cooling coil unit evaporator temperature in °C times 10, offset by 200. (unsigned integer [2B], lsb first)

RH_IN Climatic chamber relative humidity in % times 10, e.g., 606 stands for 60.6%. (unsigned integer [2B], lsb first)

#INT Number of measurement intervals passed since the last complete clearance of control panel measurement storage (usually since the last successfully proceeded KNetGetTempHumi command). (unsigned integer [2B], lsb first)

INT Measurement interval used for the measurements. (unsigned integer [2B], lsb first)

5. Communications protocol—Klimanet

OUT Two bytes containing state of all outputs at the measurement time. (unsigned integer [2B], lsb first)

MORE More-data-to-read flag. If set to one, more measurements can be read from the control panel. When the last remaining measurement is being send the flag is set to zero. (most significant bit of **OUT**)

5.3.3. KNetGetOutputs

This command returns the current state of all control panel outputs. If the corresponding output bit is set to one the output is turned on, otherwise it is turned off.

command		data field								
0x21		COOL	HEAT	HUMD	FT1	FT2	GT1	GT2	TF	BF

COOL Cooler, **HEAT** Heater, **HUMD** Humidifier,

FT1 Fluorescent tubes - first half, **FT2** Fluorescent tubes - second half,

GT1 Gas tubes - first half, **GT2** Gas tubes - second half,

TF Top section fan, **BF** Bottom section fan

5.3.4. KNetGetClimPreset

Command used for retrieving climatic preset from the klimabox control panel.

5.3.4.0.8. Request command Not a standard one again, the request command specifies a channel and time range required.

command		data field		
0x22		CHNL	START_SET	#SETS

CHNL Channel of climate preset of interest

START_SET Index of the variable to start the transfer with

#SETS Number of variables demanded

5.3.4.0.9. Response The response from the control panel is in structure identical to the KNetSetClimPreset command (5.2.3).

command	data field					
0x22	CHNL	SCV	NCV	VSZ	... DATA STREAM ...	

CHNL Channel: 0,1,2,3 corresponds to the temperature, humidity, fluorescent tubes and gas tubes respectively

SCV Index of the first climate value to be transfered

NCV Number of climate values transfered

VSZ Variable size in bits. If 10bit variables are compressed, value of *VSZ* would be 10

DATA STREAM Compressed data stream

5.3.5. KNetGetStatus

Command is used to retrieve status of the control panel. It is also used to check the klimabox power state (klimabox turned on - control panel responds). The response will most likely be appended with other status variables in future.

command	data field			
0x23	#MEAS [2B]	INT [2B]	TEMP_TOP [2B]	TEMP_BOTTOM [2B]

#MEAS Number of measurements stored in the control panel (unsigned integer [2B], lsb first)

INT Actual measurement interval (unsigned integer [2B], lsb first)

TEMP_TOP The upper hard limit temperature (signed integer [2B], lsb first)

TEMP_BOTTOM The lower hard limit temperature (signed integer [2B], lsb first)

5.3.6. KNetGetDateTime

Get date and time from the klimabox control panel.

command	data field					
0x24	YYYY [2B]	MM	DD	HH	MM	SS

YYYY Year (unsigned integer [2B], lsb first),

MM Month, *DD* Day, *HH* Hour, *MM* Minute, *SS* Second

5.4. Slave request commands

There are a few asynchronous events, mainly caused by user interaction with the control panel, where the initiation of the communication has to be done by the klimabox control panel (slave device). Special request packets are dispatched from the slave, causing the master to perform the requested action on the slave. All the slave commands are identified by the packet number 0xFF. Note that every slave request command causes restart of the packet numbering.

5.4.1. KNetMeasReadReq

Measurements are stored in the control panel data storage until they are pulled over to the server side. When the number of stored measurements exceeds a certain threshold the KNetMeasReadReq is issued to request the data transfer.

packet number	command	data field
0xFF	0x30	—

5.4.2. KNetClimPresetReadReq

When the user changes the climatic preset on the klimabox control panel this event is communicated to the server to save the recent changes into the database.

packet number	command	data field
0xFF	0x32	CHNLS

CHNLS Variable whose least significant four bits correspond to a change in each climate channel. The temperature channel corresponds to the bit 0, humidity to the bit 1, fluorescent tubes to the bit 2 and gas tubes to the bit 3.

5.4.3. KNetRestartOccured

When the klimabox control panel is reset, the source of the reset is stored. The source is later retrieved when the control panel comes back to life and a KNetRestartOccured slave command is issued during the initialization.

packet number	command	data field
0xFF	0x34	SOURCE

SOURCE Reset source code gains one of the following values

Source name	Code	Description
POWER_ON	0x00	Regular power outage occurred.
EXTERNAL	0x01	Voltage on the chip's reset pin dropped.
BROWN_OUT	0x02	Supply voltage dropped to an insufficient level.
WATCH_DOG	0x03	Watchdog counter overflow.
JTAG	0x04	Reset issued by JTAG.

5.4.4. KNetDebugMessage

A command used for control panel debugging purposes.

packet number	command	data field
0xFF	0x3F	TEXT

TEXT Debug message

5.5. Command summary

Klimanet is a multipoint communication protocol that embodies three groups of commands. Control commands (sec. 5.2) load data into the klimabox control panel, request commands (sec. 5.3) retrieve data from the control panel and finally the control panel uses slave request commands (sec. 5.4) to communicate asynchronous event to the server. Summary of all commands is given in table 5.1.

Table 5.1.: Klimanet command summary

Name	Code	Description
Control commands		
KNetDumpCmd	0x00	Testing command
KNetSetOutputs	0x11	Set output states
KNetSetClimPreset	0x12	Load climate preset
KNetSetTempLimits	0x13	Set hard temperature limits
KNetSetDateTime	0x14	Set date and time
KNetSetLcdBacklight	0x15	Turn on LCD back-light
KNetBeep	0x16	Beep
KNetDispText	0x17	Display text on LCD
Request commands		
KNetGetTempHumi	0x20	Retrieve measurements
KNetGetOutputs	0x21	Get actual output states
KNetGetClimPreset	0x22	Retrieve climatic preset
KNetGetStatus	0x23	Get status
KNetGetDateTime	0x24	Get date and time
Slave request commands		
KNetMeasReadReq	0x30	Read measurements request
KNetClimPresetReadReq	0x32	Read climate preset request
KNetRestartOccured	0x34	Restart information message
KNetDebugMesssage	0x3F	Debug message

6. Measurement acquisition

Taking measurements of physical phenomena of interest is one of the basic requirements of feedback control. An accurate sensor in the correct place has crucial impact on the control quality, not mentioning its influence on system identification. Good positioning is a matter of the control theory explained in part III, whereas the accuracy is a matter of calibration and precise linearization and that is going to be explained in this chapter.

The relative humidity sensor used in klimabox is the SHT11 sensor previously mentioned in sec. 3.3.1.3. The humidity sensor readings were found to be highly unreliable during the process of klimabox system identification. Even though the sensor is claimed to be factory calibrated I decided to check the calibration on my own.

This section will explain the process of calibration of a generic humidity sensor and its application on the temperature and humidity sensor SHT11. It will further explain the process of linearization of the sensor's characteristic in a convenient manner for low performance microcomputers.

6.1. Humidity sensor calibration

Calibration of relative humidity sensors is not an easy task to perform in the home environment. However, several ways of calibration were proposed and realized.

The first idea was to use another humidity sensor, compare the results and see if there was a difference. A psychrometer, an old humidity measurement technique, was used as the second sensing element. A psychrometer has two bulbs: one wet and one dry. After a period of time, the water on the wet bulb evaporates causing the wet bulb to go cooler than the dry bulb. The rate of evaporation is relative humidity dependent and so is the temperature difference between the two bulbs. A psychrometric conversion of the temperature difference gives the relative humidity level in the area.

This approach turned out to be unreliable due to various factors. The wet bulb is covered with a gauze sock through which water rises from a reservoir. To give a quicker dynamic

6. Measurement acquisition

response the wet bulb should be fanned. A balance between rising water and drying by the fan has to be established. It was however unclear whether the wet bulb temperature was influenced more by the temperature of water in the tank or it was being over-dried by the fan. The balance was not found and the homemade psychrometer was therefore not suitable for humidity sensor calibration.

The second approach I came up with, was to fix absolute humidity in some environment and then by changing temperature vary the relative humidity. Absolute humidity is an amount of water vapor in a specific volume of air, thus, in a closed container where the relative humidity is lower than 100%, the absolute humidity is constant. A correction characteristic was obtained for the sensor, however, there was no reference point to attach the correction characteristics to. A relative humidity calibration can be achieved, but the right absolute humidity is still left unknown. This approach was also tried and it is further explained in section 6.1.1.

To create an environment with a known relative humidity level became the next goal. It was found that various salt solutions have the ability to produce a well defined humidity level above their surface. This approach finally gave the answer to the sensor's calibration and it is explained in the section 6.1.2.

6.1.1. Calibration by varying relative humidity in a constant absolute humidity environment

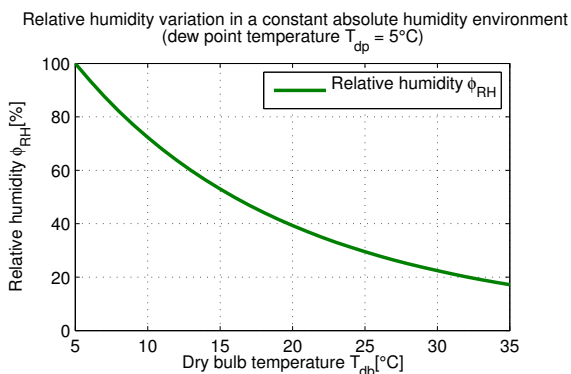


Figure 6.1.: Relative humidity vs. temperature relationship in a fixed absolute humidity environment

A relative humidity trajectory for a changing temperature can be calculated within a fixed absolute humidity environment. Absolute humidity is a partial water vapor pressure multiplied by a constant and divided by absolute temperature. As a function of RH and T it is described as follows

$$\varrho_{AH} = a \cdot 216.17 \cdot \frac{\phi_{RH}}{100} \cdot \frac{1}{T_{db} + 273.15} \cdot e^{\frac{bT}{T+c}} \quad (6.1)$$

where a , b and c are coefficients of the Magnus

formula over water.

Now stating the relative humidity to be a function of temperature

$$\phi_{RH} = f(T_{db})$$

substituting it to 6.1 and letting the absolute humidity be constant

$$\frac{d}{dT}(\varrho_{AH}) = 0$$

the following differential equation is derived:

$$\dot{f}(T_{db}) = f(T_{db}) \cdot \left(\frac{1}{T_{db} + 273.15} - \frac{bc}{(T_{db} + c)^2} \right) \quad (6.2)$$

By solving this differential equation, relative humidity as a temperature function is obtained and it is characterized as follows:

$$\phi_{RH}|_{\varrho_{AH}=const.} = f(T_{db}) = c_n \cdot (T_{db} + 273.15) \cdot e^{\frac{bc}{T_{db}+c}} \quad (6.3)$$

where c_n is a normalization constant dependent on the absolute humidity level. The relation between relative humidity and temperature is plotted at figure 6.1.

The results of the theory applied to the SHT11 sensor are described in section 6.1.3.

6.1.2. Humidity calibration by fixed points of aqueous salt solutions

The common methods of controlling the humidity accurately use either a humidity generator or the equilibration of a closed space with a chemical system which produces the desired equilibrium vapor pressure.

Humidity generators tend to be expensive and complex whereas equilibrium with chemical systems that provide fixed points is a relatively inexpensive and simple method of humidity control. Among the chemical systems used for this purpose, the aqueous salt solutions were found to be cheap enough to get. Several salt solutions along with their fixed points are presented in a table 6.1.

Since a given saturated salt solution provides only one relative humidity level at any desired temperature, a different relative humidity level must be achieved by selecting another appropriate salt [1].

The solution is put into a flat-bottomed vessel and water is added. The amount of added water should be small enough not to dissolve the salt solution, the desired state is to have

6. Measurement acquisition

Salt	Formula	Saturation relative humidity at 20 °C[%RH]
Lithium Bromide	$LiBr$	6.61
Lithium Chloride	$LiCl$	11.31
Potassium Acetate	CH_3CO_2K	23.11
Magnesium Chloride	$MgCl_2$	33.07
Potassium Carbonate	K_2CO_3	43.16
Magnesium Nitrate	$Mg(NO_3)_2$	54.38
Potassium Iodide	KI	69.90
Sodium Chloride	$NaCl$	75.47
Ammonium Sulfate	$(NH_4)_2SO_4$	81.34
Potassium Chloride	KCl	85.11
Potassium Nitrate	KNO_3	94.62

Table 6.1.: Relative humidity level fixed points of various saturated salt solutions [1].

damp looking crystals. The sensor is then placed inside the vessel near the salt surface and the vessel is sealed. After a period of time, an equilibrium is achieved leaving the relative humidity inside at a fixed level. The relative humidity level is also temperature dependent; the temperature of the vessel should be kept constant during the calibration.

This calibration process applied to the SHT11 sensor is shown in section 6.1.3.

6.1.3. SHT11 sensor calibration



Figure 6.2.: SHT11 sensor being calibrated by potassium carbonate.

Hardware details of the sensor implementation can be found in sec. 3.3.1.3. This section gives the results of two calibration processes applied to the SHT11 sensor.

6.1.3.0.10. Calibration in a fixed absolute humidity environment

When the theory developed in sec. 6.1.1 was applied to the SHT11 sensor it seemed that the sensor was not measuring properly (figure 6.3 shows the results). However, when seen from a distance two months later, the reason why the method

gave such wrong results was that the temperature of the experiment was incidentally brought below the dew point. Among other reasons, it could also happen that the sensor was not fanned enough (it was not fanned at all) and the dynamic response of the sensor was then very slow. Although the theory is correct, no further measurements were performed to observe

it. A different calibration method was performed instead and it is described in the following paragraph.

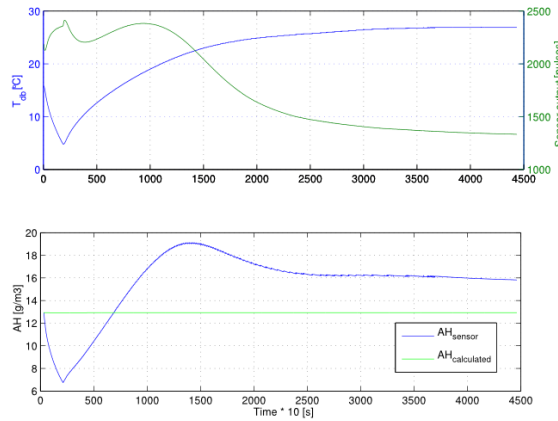


Figure 6.3.: SHT11 sensor calibration by constant absolute humidity. The system was incidentally brought to saturation at the beginning which is one of the reasons why this method failed to give reasonable results.

6.1.3.0.11. Calibration by aqueous salt solutions The calibration process described in section 6.1.2 was applied to the SHT11 sensor for four different salt solutions, namely: magnesium chloride, potassium carbonate, sodium chloride and potassium nitrate. The sensor being calibrated by potassium carbonate is captured at photograph 6.2.

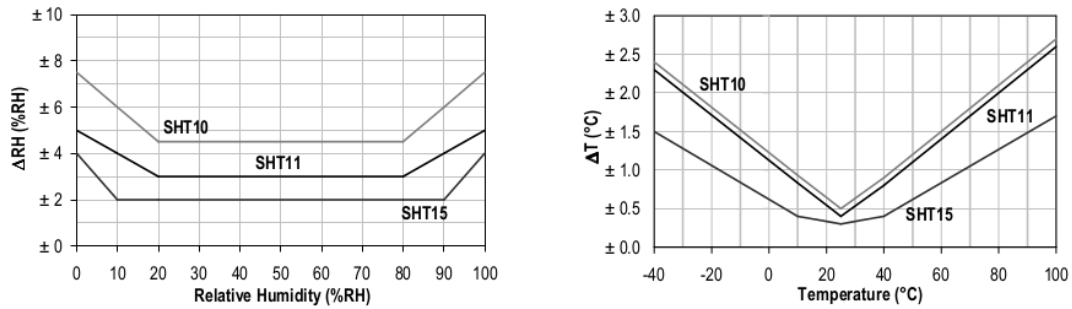


Figure 6.4.: SHT11 sensor resolution. Credits: sensirion.com

The results of all four calibration procedures are given in figure 6.5. The temperature was held at 20 °C by a thermostat; it effected the calibration procedure and periodic waves caused by the temperature control can be seen in the graphs. SHT11 sensor relative humidity resolution is $\pm 3\%$ RH. The calibration experiment gave the deviation to be between $+1.6\%$ RH and $+4.2\%$ RH for the measured range. Under the circumstances in which the calibration was performed, the deviation is not seen to be severely exceeding the sensor resolution. Therefore the sensor can be stated to measure the relative humidity correctly and no corrections to the characteristics are necessary.

6. Measurement acquisition

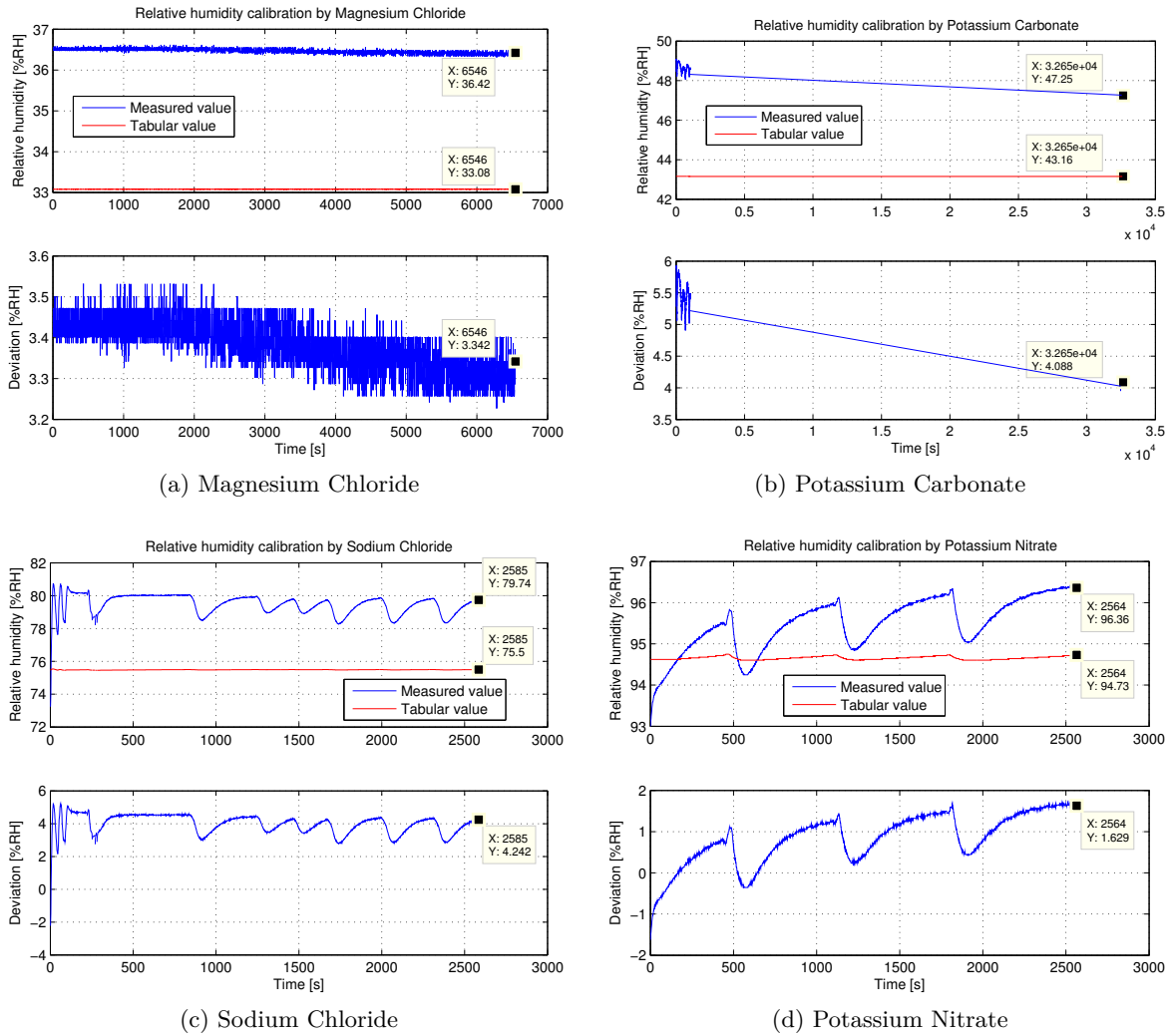


Figure 6.5.: SHT11 sensor calibration by saturated salts at 20 °C. Note that the waves in relative humidity response are caused by temperature control.

Two different calibration methods were performed. The second method gave rational results leaving a message that the sensor was measuring correctly. The issue in relative humidity measurements had therefore be somewhere else! It was learnt that the sensor has to be adequately fanned to give a reasonable dynamic response. The SHT11 sensor response time to relative humidity changes is 5 – 20 s when fanned and 2 – 10 minutes when left in still air. A fan was installed to deliver appropriate air flow to the sensor and since then no flaw in sensor readings has been spotted.

6.2. Conversion of the SHT11 sensor humidity signal output

A correction formula has been recommended by the manufacturer to compensate for non-linearity of the humidity sensor. The full accuracy of the sensor is obtained and the humidity

6.2. Conversion of the SHT11 sensor humidity signal output

readout (number of pulses h) is converted using the following formula

$$RH_{true} = c_3 h^2 + c_2 h + c_1 + C_{RH}(T_{db}) \quad (6.4)$$

where $C_{RH}(T_{db})$ is a temperature correction, h is the humidity sensor output in range 100 – 3340 pulses and c_1, c_2, c_3 are coefficients from a sensor's data-sheet.

For temperatures significantly different from 25 °C the humidity signal requires temperature compensation. The temperature correction corresponds roughly to 0.12 %RH/°C @ 50 %RH, but the exact equation is

$$C_{RH}(T_{db}) = (T_{db} - 25) \cdot (t_1 + t_2 \cdot h) \quad (6.5)$$

where t_1, t_2 are coefficients given in the data-sheet.

Even though these calculations are simple to calculate for modern computers, when it comes to 8bit microchips, the simplicity is no longer so apparent. First of all, such a calculation requires floating point operations (operations with decimal numbers) and thus it requires use of the code heavy floating point algebra libraries. Secondly, it requires 7 floating point multiplications and 5 additions to compute, which at microchip speeds is not negligible.

The correction formula was therefore approximated by a piece-wise linear function with coefficients calculated in the following manner. The most probable range of operation (30 – 100 %RH) was divided into 4 regions (accuracy would not be satisfied for fewer regions) and least-square estimate of the shallow quadratic function 6.4 was found. The parameters of the linear function approximation were found as follows:

$$\theta(r) = \begin{bmatrix} \theta_1(r) \\ \theta_2(r) \end{bmatrix} = \begin{bmatrix} h(r) & 1 \end{bmatrix}^+ \cdot RH_{true}(r) \quad (6.6)$$

where θ is a vector of the identified parameters, r is a region and $[\]^+$ denotes pseudo-inversion.

The linear approximation is now found in form

$$RH_{apr.}(r) = h(r) \cdot \theta_1(r) + \theta_2(r) \quad (6.7)$$

Although the formula now requires only one multiplication and one addition, it still has to be calculated using floating point algebra. For the sake of code lightness and speed the, generally rational, parameters θ were for every region substituted by integer parameters k, a, b that satisfy the following equations:

$$RH_{apr.int.} = \frac{k \cdot h + a}{b} \quad (6.8)$$

$$k = \text{int16} \left(\frac{\theta_1}{\theta_2} a \right) \quad (6.9)$$

$$b = \text{int16} \left(\frac{a}{\theta_2} \right) \quad (6.10)$$

$$(k \cdot \max(h) + a) < 2^{16} \quad (6.11)$$

$$e = RH_{true} - \text{int16}(RH_{apr.int.}) \quad (6.12)$$

$$J = \arg \min_a E \{ e' \cdot e \} \quad (6.13)$$

Which in other words mean that the parameters k, a, b were found in a way so that the final estimation minimizes the mean square error of the estimate.

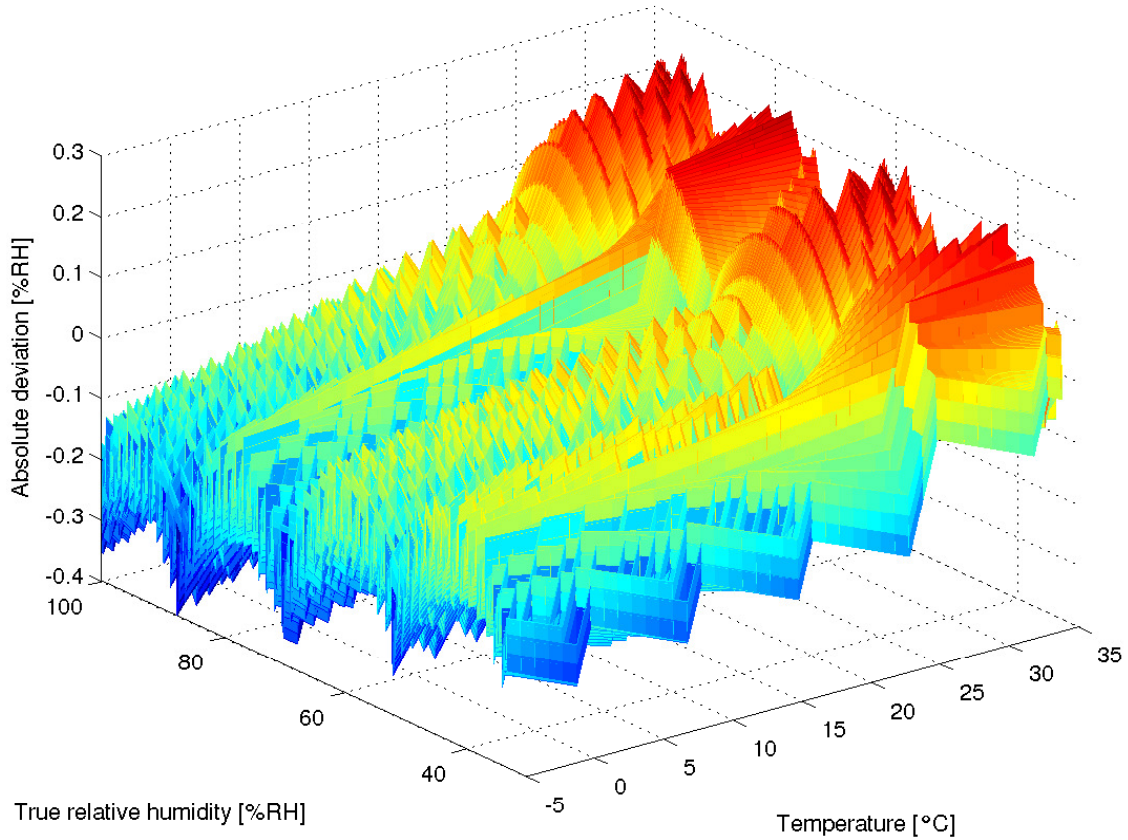


Figure 6.6.: Deviation of the integer approximation from the SHT11 relative humidity conversion function.

The SHT11 relative humidity temperature correction equation 6.5 was rewritten into the

following approximate non-floating point calculation

$$C_{RH,int.}(T_{db}) = \left[([T_{db} - 25] \curvearrowright 1) \cdot \left(\frac{2000 + 16 \cdot h}{100} \right) \right] \curvearrowright 10 \quad (6.14)$$

where $\curvearrowright N$ denotes a bitwise shift to right by N positions.

The relative humidity is then converted from the readout as

$$\phi_{RH} = RH_{apr.int.}(r) + C_{RH,int.} \quad (6.15)$$

A deviation of the integer approximation to the original conversion function is plotted in figure 6.6. The maximal absolute error of the approximation is a surprisingly small value $\pm 0.4\%$ RH.

The whole calculation was finally simplified into a few instructions for finding the corresponding region, three integer multiplications, two integer divisions and three integer additions. Integer operations are generally much faster than floating point ones, bitwise shifting is a super fast one cycle operation. The speed gained by this integer approximation is significant. Floating point library is not used so a big portion of program memory was saved (appr. 14%). A look-up table with parameters for each region takes only 96 bytes.

Part III.

Model

7. Model introduction

A model of a dynamic system is a basic controller design element and understanding the nature of the controlled system is one of the fundamental requirements in the control theory. The mathematical model of the system embodies the knowledge gained about the system; it describes the static behavior of the system, but for the most part it describes the system's dynamics. The model, in reality, is usually not recognized without performing a system identification.

There are three main system identification principles:

1. White-box modeling. The model of the system is constructed purely from the physical insight - it is based on first principles.
2. Grey-box modeling. Semi-physical model, model structure is known, but some parameters are not.
3. Black-box modeling. Non-parametric identification procedure with no prior knowledge about structure or parameters of the system.

Generally, the more physical insight is used, the better the model is obtained. One extreme is the white-box model, where the physical laws govern the model behavior and all parameters are known a priori. The other extreme is the black-box model, where no prior information is used. The black-box modeling methods give statistical models of the system.

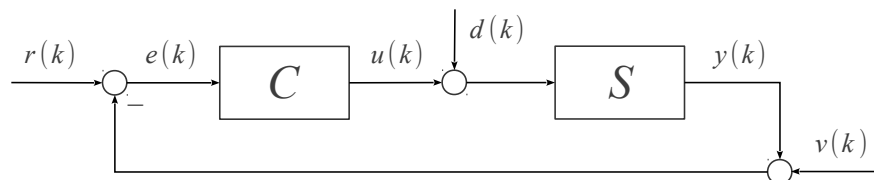


Figure 7.1.: Generic control scheme: C is a controller, S is a controlled system, $r(k)$ is a set reference at the time instant k , $e(k)$ is a control error, $u(k)$ is a set of controllable inputs, $d(k)$ is a set of disturbance inputs, $y(k)$ is a set of outputs and $v(k)$ is an additive output/measurement error.

7. Model introduction

System identification steps paving the path for a klimabox mathematic model will be described within this part of the thesis. Identification of the climatic chamber was approached from two sides in parallel: chapter 8 explains the first-principles approach to the Klimabox identification, the black-box techniques are presented in chapter 9.

The model used for the controller design is the one obtained by the black-box approach.

7.1. Klimabox model outline

Klimabox is a good example of a dynamic system. Eight inputs are entering the system and three outputs are leaving it. The outline of the real system and the model is the same and it is pictured in figure 7.2.

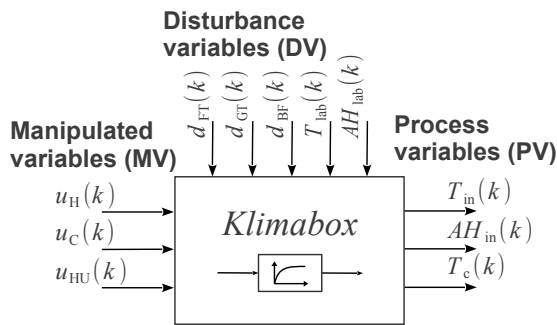


Figure 7.2.: Klimabox model scheme

A set of inputs that are about to be manipulated by a controller are called MANIPULATED INPUTS. The heating rod input u_H belongs there as well as the CCU input u_C and the humidifier input u_{HU} .

The inputs that enter the system but over which the controller has no power form a group of DISTURBANCE INPUTS. The fluorescent tubes (u_{FT}), the gas tubes (u_{GT}) and

the bottom fans (u_{BF}) are set according to their daily preset (they are not bonded to the temperature and humidity control) and they therefore are the disturbance inputs. The laboratory environment (T_{lab} , AH_{lab}) also brings disturbance to the Klimabox.

Outputs of the Klimabox and the model are the chamber air temperature T_{in} , the absolute humidity of the chamber air AH_{in} (q_{in}) and the cooling coil unit evaporator temperature T_c , altogether called PROCESS VALUES.

7.2. ARX model and least-square estimation

Autoregressive model with exogenous input, abbreviated as ARX, is a system of the form

$$a(d)y(d) = b(d)u(t) + e(t) \quad (7.1)$$

The discrete output $y(d)$ is equal to

$$y(t) = \frac{b(d)}{a(d)}u(t) + \frac{1}{a(d)}e(t) \quad (7.2)$$

which in compact form can be written as

$$\mathbf{y} = \mathbf{Z}\boldsymbol{\theta} + \mathbf{e} \quad (7.3)$$

where \mathbf{y} is a vector of all output data, \mathbf{Z} denotes a regressor, $\boldsymbol{\theta}$ is a vector of unknown variables and \mathbf{e} a vector of noise. Note that ARX models are only suitable for SISO (Single Input Single Output) or MISO (Multiple Input Single Output) systems.

The parameters $\boldsymbol{\theta}$ can be found by minimizing multiple different cost functions. The best cost function for the system identification purposes seems to be the sum estimation error squares. The cost function can be expressed as

$$J = \mathbf{e}^T \mathbf{e} = (\mathbf{y} - \mathbf{Z}\boldsymbol{\theta})^T (\mathbf{y} - \mathbf{Z}\boldsymbol{\theta}) \quad (7.4)$$

The optimal vector of parameter minimizes the cost function

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \{J\} = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{y} \quad (7.5)$$

The model found is a least-squares estimate obtained from the data. The estimation is calculated from the data as whole, therefore it is called batch processing.

The estimation can also be done a recursive manner, which gives a maximum likelihood estimate. More information in [11, 9].

8. First principles approach

This chapter will explain the process of creation of a Klimabox white-box model.

8.1. Identification of Klimabox's components

Klimabox has many dynamic elements inside its body, most notably the actuators. The heating rod takes some time to heat up, and it does not affect the absolute humidity. The cooling coil unit, on the other hand, alters the level of both the temperature and the temperature. Although the humidifier seems to change nothing but the humidity, it affects the temperature too. The gas tubes radiate a portion of their heat into the chamber; more importantly though, they heat up the whole laboratory room. The bottom fans stir the air inside the chamber, which seems not to have an effect on the chamber climate; it, however, does change the temperature. The only actuator where no significant effect on the chamber climate was detected is the fluorescent lamps.

8.1.1. Heating rod identification

The heating rod is a metal component that delivers heat into the climate chamber; the absolute humidity is not affected.

The heating element is a resistive metal wire supplied by electric current; the supply current was measured to be $i_h = 1.57 \text{ A @ } 230 \text{ V/50 Hz}$. The current is heating up the rod, whereas convection is cooling it down; the heating rod temperature dynamics (scheme 8.1) is described by the following formula:

$$\dot{T}_h = \frac{1}{C_h} (u \cdot i_h - UA(T_h - T_{env})) \quad (8.1)$$

The system has significant first order dynamics and so it will be modeled. The first parameter to be estimated is the heat transfer coefficient U . It can be estimated by turning the heating rod on for a long period of time to let the rod temperature stabilize (the dynamics diminish). Now, from the law of energy conservation, it can be stated that the input power equals the output power:

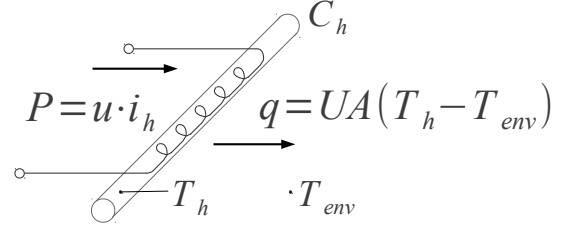


Figure 8.1.: Energy balance on the heating rod

$$u \cdot i_h = UA(T_h - T_{env}) \quad (8.2)$$

from where the heat transfer coefficient UA can be determined. The experiment was performed and the heating rod temperature settled at 233°C while the environment temperature was at 25°C leaving the heat transfer coefficient to be

$$UA = \frac{230 \cdot 1.57}{233 - 25} = 1.73 \text{ W/K} \quad (8.3)$$

The heat capacity of the heating rod C_h was however still unknown and it was going to be found from the dynamic response of the heating rod. A step response was measured and an ARX model was derived. The estimation gave the following model:

$$\dot{T}_h = K_t(T_h - T_{env}) + K_u \cdot u_{on/off} = 0.0064(T_h - T_{env}) + 1.3311 \cdot u_{on/off} \quad (8.4)$$

The heat capacity was identified to be

$$C_h = \frac{u \cdot i_h}{K_u} = 271.3 \text{ J/K} \quad (8.5)$$

which is equal to the heat capacity of 0.7 kg of pure copper. The heating rod cannot be weighted separately, but its weight was approximated to be a similar weight. Another check of the model can be performed by computing the heat transfer coefficient UA from the estimated K_t parameter as $UA = K_t \cdot C_h = 1.73 \text{ W/K}$, which agrees with

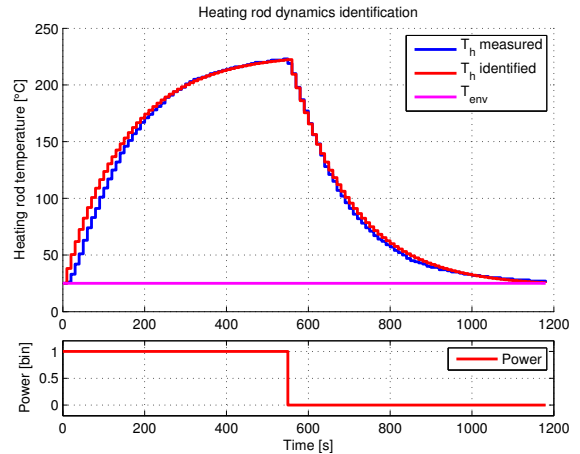


Figure 8.2.: Heating rod identification

8. First principles approach

the previously obtained value.

8.1.2. Cooling coil unit identification

The cooling coil unit (CCU) is the most complicated actuator working in the klimabox. The CCU is a circuit of components, where the compressor is pumping the refrigerant through the piping and the thermal expansion valve (TXV) into the evaporator. Once the CCU is on, the evaporator cools down instantly; the thermal expansion valve will close down the supply of refrigerant into the evaporator as soon as its temperature gets close to the super-heating temperature set on the TXV. The TXV valve is however not a temperature controller - it is a temperature driven pressure controller. Figure 8.3 shows temperature measurements on the CCU. The TXV super-heating temperature was set to around 7°C , which can be seen from the temperature difference between the chamber temperature and the evaporator temperature (plotted in orange color).

The CCU can alter the chamber environment rapidly, it is therefore important to model the evaporator temperature accurately to obtain an accurate klimabox model.

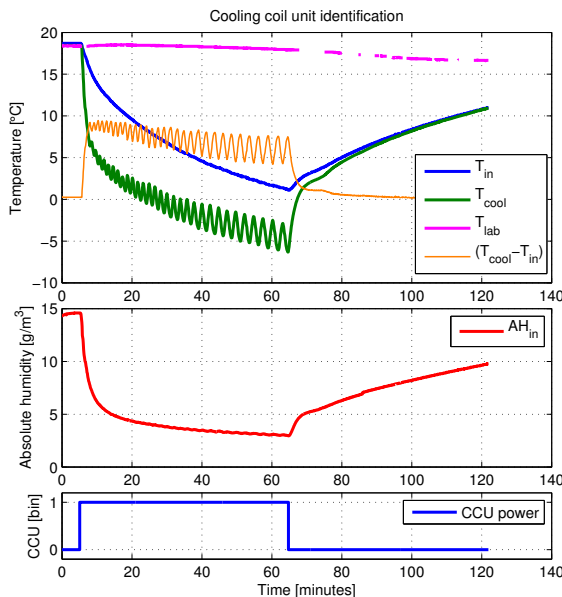


Figure 8.3.: Cooling coil unit identification

temperature gets below the water freezing point (0°C), water drops formed on the evaporator previously begin to freeze giving out its latent heat of melting ($0.334 \cdot 10^6 \text{ J/kg}$), which heats up the evaporator again. Once the CCU is turned off, the evaporator heats up and all the mentioned processes happen in reverse order.

There are complex processes happening on the evaporator. There is the TXV-controlled flow of refrigerant evaporating inside the evaporator's body cooling it down; the R134A refrigerant requires $0.217 \cdot 10^6 \text{ J/kg}$ to evaporate at its boiling point -26.1°C . The evaporator's surface temperature is likely to cool down below the chamber's air dew point quickly; air blowing around the evaporator's ribbons gets cooled down and becomes humidity saturated, which leads to condensation. As water condensates it loses its latent heat ($2.270 \cdot 10^6 \text{ J/kg}$) heating up the evaporator's body in return. When the evaporator's tem-

The rate of condensation/freezing is highly non-linear and cannot be obtained easily from the laws of physics. The model of evaporator temperature would not be accurate; placing a thermometer onto the evaporator turned out to be the best option in terms of accuracy and simplicity.

The effect of the CCU on the chamber air humidity is closely related to the model of evaporator temperature. Air humidity is absorbed onto the evaporator's surface, when the CCU is running; and it evaporates back, when the CCU is turned off. There are only two ways water can escape from the chamber:

1. through a drain in the chamber bottom, when the amount of condensed water is high enough; the amount of condensate is however not known.
2. by water vapor pressure equalization with the laboratory environment; there is no control over the laboratory environment though.

The limited capability of effective drying of the chamber indicates difficulties in the control process. The model of the CCU was not found.

8.1.3. Humidifier identification

The humidifier used in Klimabox is an adiabatic type humidifier that atomizes the water into very small particles and blows them into air. The device affects absolute humidity as well as temperature. There is no significant dynamic behavior; the system is a zero order dynamic system.

The amount of water transferred into air was measured as a decrease of weight of the humidifier water tank over a period of time. The water flow caused by the humidifier is

$$f_{w, hum.} = 0.04 \text{ g/s} \quad (8.6)$$

Air temperature is affected by the latent heat the water needs to evaporate. The heat flow caused by the humidifier is

$$q_{hum.} = -f_{w, hum.} \cdot h_{l2g} = -98 \text{ W} \quad (8.7)$$

8.1.4. Gas tubes identification

The gas tubes affect the temperature inside the chamber only by radiation; the absolute humidity is not affected. The gas tubes take some time to heat up; the radiation heat transfer

8. First principles approach

is not immediate. The dynamics of the gas tubes body temperature has been measured indirectly by two thermometers, one shaded and the other unshaded. The difference between the two temperatures reveals the time dynamics of the lamp. The identification experiment results are presented in figure 8.4; time dynamics of the gas tubes were modeled by a first order transfer function and the time constant is $T_{gt} = 233$ s.

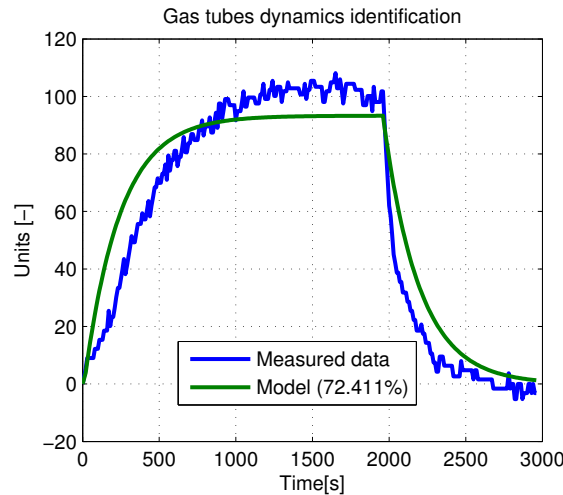


Figure 8.4.: Gas tubes heat up dynamics identification

Heat flow caused by the gas tubes is to be identified later.

8.1.5. Bottom fans identification

The technology block incorporates three fans that provide air movement inside the klimabox and mix the air to be consistent throughout the chamber. The average velocity of air movement inside the chamber was measured by an anemometer¹ to be $v_a = 1.07$ m/s. The air mass movement was calculated to be $f_a = 0.237$ m³/s.

The fans also act as a heating element. The heat flow generated equals the the electrical power entering the fans:

$$q_{bf} = P_{bf} = u \cdot i_{bf} = 230 \cdot 0.5 = 115 \text{ W} \quad (8.8)$$

This heat source is however a disturbance, because bottom fans are not used for control - they are turned on permanently. There are no dynamics in the heat flow; the bottom fans are a zero order dynamic system.

¹A device measuring wind speed.

8.1.6. Chamber body dynamics identification

The climatic chamber inside the klimabox is isolated from the laboratory environment by a pair of metal walls and a layer of extruded polystyrene. There are conduction heat losses to the laboratory that can be calculated as follows:

$$q_{loss} = U_{loss} A_{wall} (T_{in} - T_{lab}) \quad (8.9)$$

The heat transfer coefficient U_{loss} can be determined quite easily, it took some time to figure it out though. A heat source is switched on inside the chamber and it is left on until the chamber temperature stabilizes. Figure 8.5 shows such an experiment performed; note that two heat sources were applied, the heating rod and the bottom fans.

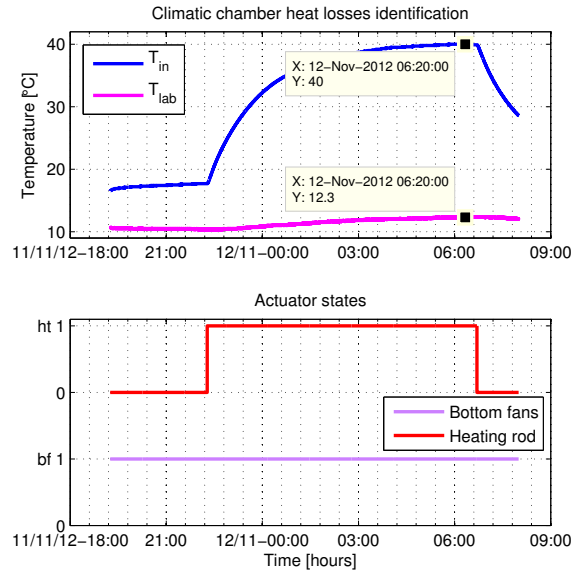


Figure 8.5.: Chamber heat losses identification experiment

The area of all chamber walls was calculated to be $A_{wall} = 7.2 \text{ m}^2$. The loss heat transfer coefficient is

$$U_{loss} = \frac{P_{bf} + P_{ht}}{A_{wall} (T_{in} - T_{lab})} \bigg|_{\frac{dT_{in}}{dt} \rightarrow 0} = 2.17 \text{ W/m}^2 \cdot \text{K} \quad (8.10)$$

which is the same heat loss that would occur if a 1.8 cm thick layer of EPS² was enveloping the chamber completely. The real insulation thickness is $\sim 2.5 \text{ cm}$, but there are a lot of thermal bridges and leaks, which lowers the insulation effect so the identified heat loss coefficient makes sense. The loss heat transfer coefficient was found and checked.

Estimating the dynamics of the wall and branches inside the chamber is tricky because it is a

²Expanded polystyrene.

8. First principles approach

system with spread parameters. There is possibly an infinite number of states, so a reasonable simplification has to be used. The dynamics of the chamber walls was modeled by a number of logically separated parts and the result is presented in section 8.2.

8.2. Klimabox library

A library of dynamic models of each dynamic element present in the Klimabox was created. Figure 8.6 shows the full Klimabox library made in SIMULINK®; it includes dynamic models of the chamber, duct, ventilation, air leakage, insulated and bare walls, heating rod, cooling coil unit, humidifier, gas tubes, bottom fans, and laboratory environment connectors and it also contains some additional support components.

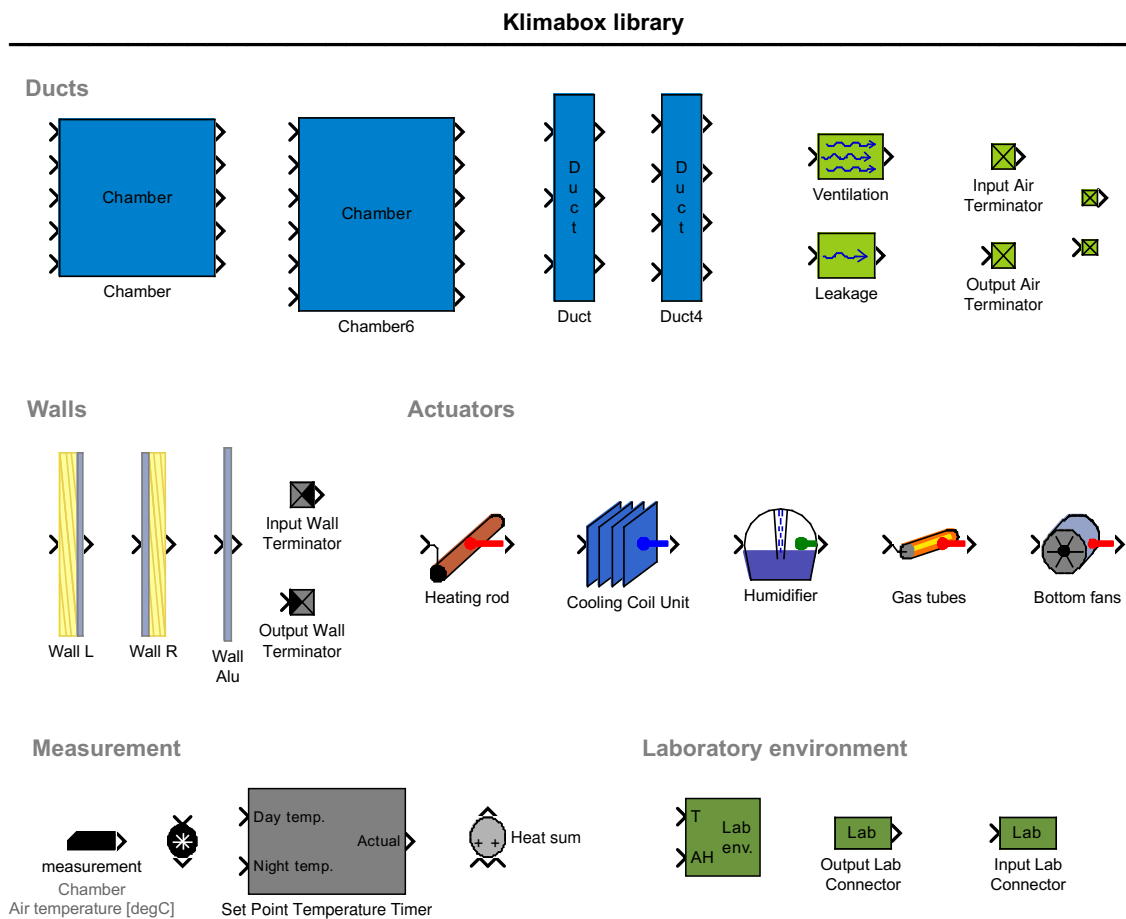


Figure 8.6.: Klimabox library

The components are modeled by the first principles method. Each component also has a list of parameters defining its physical properties.

For example, the wall model (figure 8.7a) is parametrized by the

- wall width L , height H and thickness d

- insulation thickness on input side $d_{ins,i}$ and output side $d_{ins,o}$
- thermal conductivity of wall material k and insulation material k_{ins}
- surface heat transfer coefficient input side h_i and output side h_o
- volumetric specific heat capacity of wall material c_V and insulation $c_{V,ins}$

The wall temperature is a state variable that only changes when there is a heat flow in or out of the wall. The heat flows are only possible by conduction and the heat flow on the input side is modeled as:

$$q_i = U_i A_{wall} (T_{inp.} - T_{wall}) \quad (8.11)$$

where

$$\begin{aligned} U_i &= \left(\frac{1}{h_i} + \frac{d}{2k} + \frac{d_{ins,i}}{k_{ins}} \right)^{-1} \\ A_{wall} &= L \cdot H \end{aligned}$$

The output-side heat flow q_o is calculated in a similar fashion; both heat flow vectors are assumed to point towards the wall. The wall temperature is then calculated from the first law of thermodynamics as:

$$\frac{dT_{wall}}{dt} = \frac{1}{C} (q_i + q_o) \quad (8.12)$$

where

$$C = A_{wall} (d \cdot c_V + (d_{ins,i} + d_{ins,o}) c_{V,ins})$$

There is a zero humidity flow in or out of the wall - the wall does not accumulate humidity.

All the other components are modeled in the same way by the physical laws that describe them. Three other component block diagrams are for illustration presented in figure 8.7. The heating rod model (8.7b) reflects the equation described in section 8.1.1. The duct model (8.7c) shows that the duct is nothing more than just a heat and humidity accumulator; the chamber model has exactly the same internal structure as the duct model.

Block diagram 8.7d shows the cooling coil unit model. A thermal model including the thermo-regulation valve controller was created; however, the humidity processes happening on the evaporator surface (described in sec. 8.1.2) were not modeled. The humidity flow regarding the CCU was not found.

8. First principles approach

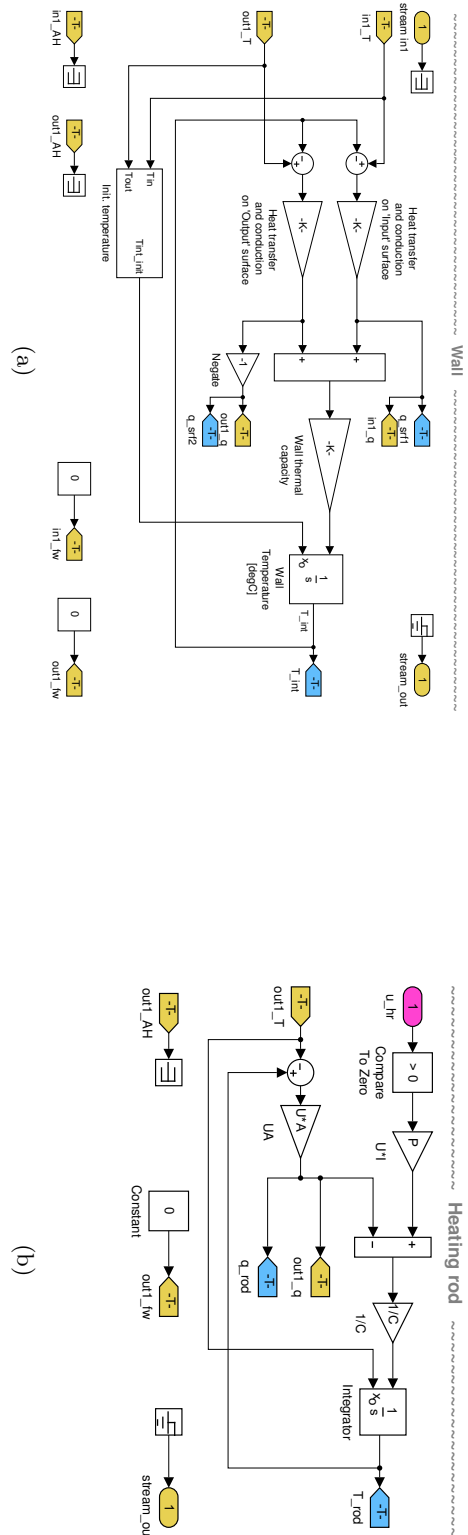


Figure 8.7.: Klimabox library components: wall (a), heating rod (b), duct/chamber (c) and unfinished model of cooling coil unit (d)

8.3. Klimabox simulator

A dynamic simulator of the Klimabox device (figure 8.9) was constructed using the components from the Klimabox library (sec. 8.2). The Klimabox simulator exhibits correct thermodynamic and psychrometric behavior when simulated without the CCU, the CCU model was not found. Figure 8.8 shows the heat loss simulation, simulated at similar conditions to those that occurred when the heat loss coefficient was identified (figure 8.5) in the real system. Responses of both the simulator and the real system are comparable.

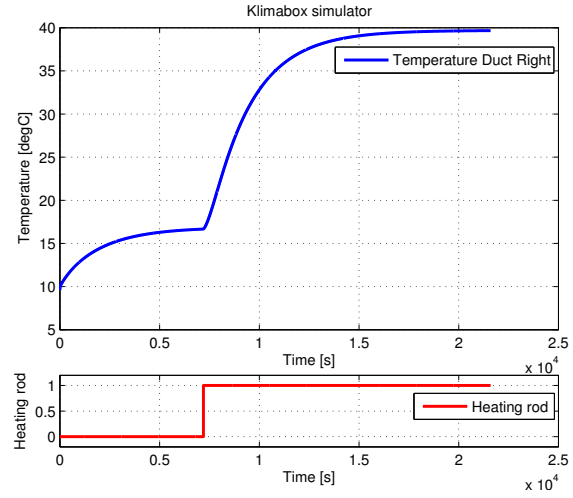


Figure 8.8.: Klimabox simulator - heat loss simulation

The model of the CCU was not derived and therefore one fundamental dynamic element is missing for a complete Klimabox model. This missing component put an end to the development of the simulator and the simulator was not used for controller design. However once finished, the Klimabox simulator can provide a model of great accuracy.

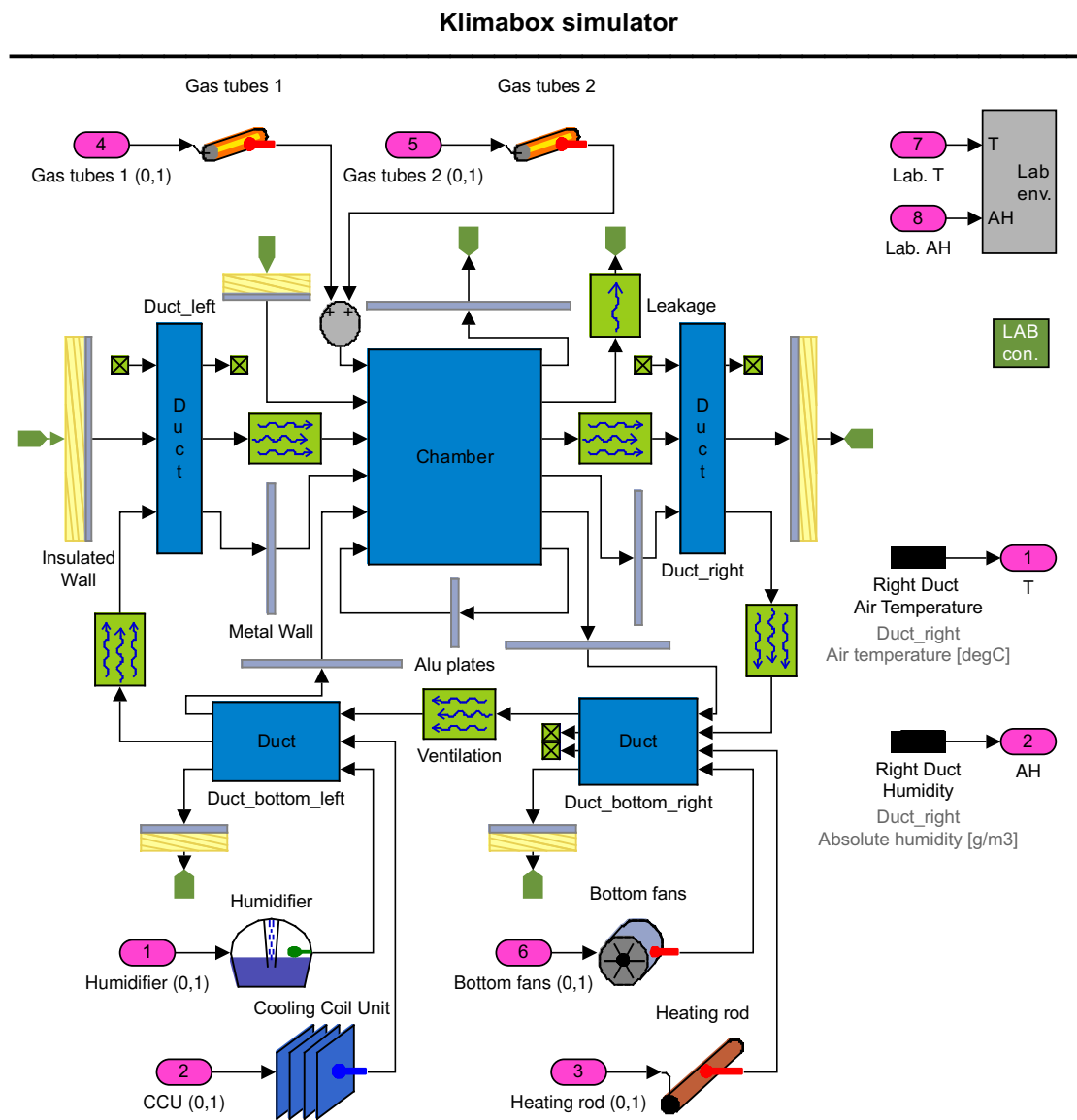


Figure 8.9.: Klimabox simulator

9. Black-box approach

The black-box system identification is a purely data-driven statistic modeling tool. There are black-box identification techniques for both time domain and frequency domain, but only the time domain methods are going to be used for system identification of the Klimabox device in this thesis.

An internal structure of models derived by the black-box methods does not follow any physical properties of the system, instead it is chosen to provide the best statistical fit. Some methods, like subspace identification methods, can recognize the model order automatically while others (e.g. ARX) require to be provided with a model order by an engineer.

The ARX method and the subspace identification method applied on the Klimabox are going to be described in the following section.

Note that all the klimabox actuators are treated as zero order systems (no dynamics).

9.1. ARX identification method

The first black-box identification method used for the klimabox identification was the ARX method. The ARX model and the estimation of its parameters is briefly described in section 7.2, more information can be found in [11, 9].

Two sets of data were measured while feeding the klimabox actuators with a pseudo-random binary sequence (PRBS). The first set was used for the model identification and the second for the model validation.

The ARX was one of the first methods used to obtain the overall klimabox model. In the early stages of identification the measured outputs were the chamber temperature T_{in} , the chamber absolute humidity AH_{in} , the laboratory temperature T_{lab} and also an outside temperature T_{out} . It turned out that the laboratory vs. outside dynamics identification is not necessary, because the laboratory temperature measurements are available all the time.

9. Black-box approach

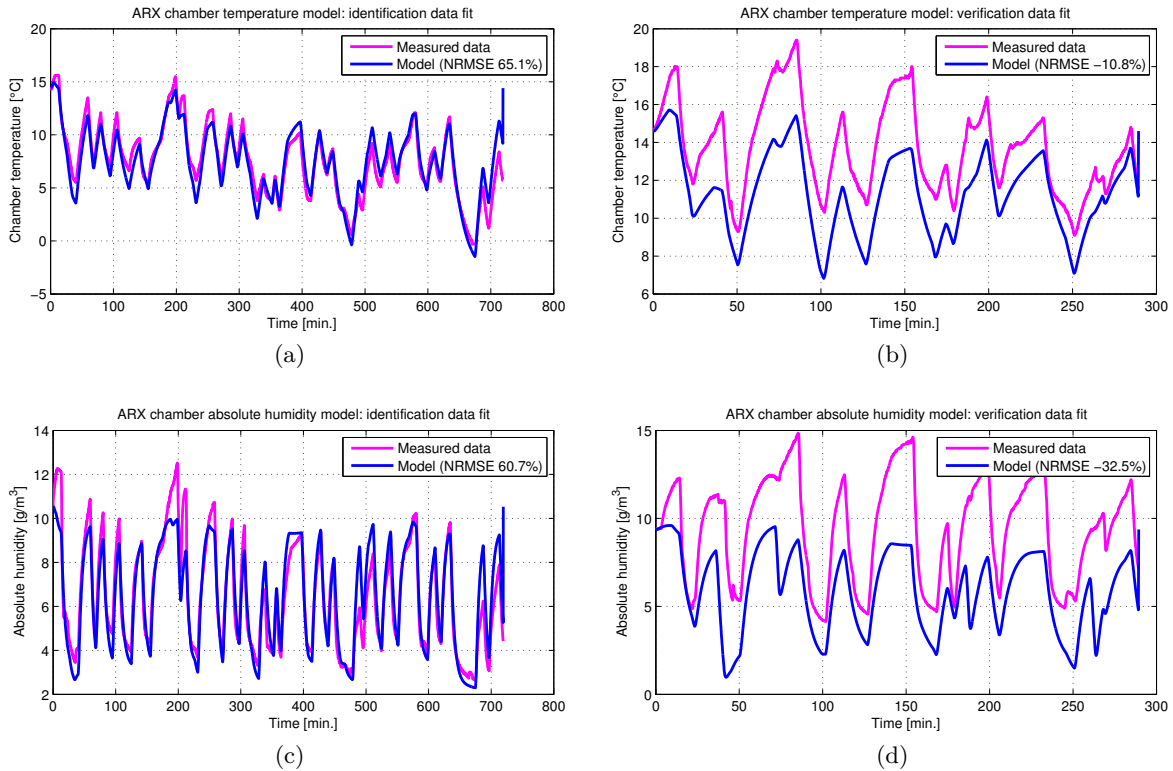


Figure 9.1.: Klimabox ARX model data fit

The ARX model is only suitable for the identification of single output systems; however, assuming that chamber the temperature and absolute humidity are independent, the ARX model for both outputs can be derived separately. Two fifth-order models were identified and concatenated to make the final model.

Figure 9.1 shows the identification data fit of the model together with the validation data fit. The model shows a promising fit on the identification data (65% for the temperature and 60% for the absolute humidity); however, the validation data reveal the model failing. The validation data show that the model preserves the trends, but there is a significant drift in the output values right from the beginning. The drift is not caused by the model estimation, there is a noticeable lack of information in the data.

The identification data lacked important information, like the evaporator temperature T_c and the laboratory absolute humidity AH_{lab} ; the Klimabox was therefore equipped with two additional sensors to provide the information.

The new evaporator temperature T_c is also a model output, but it is correlated with the chamber temperature T_{in} . The outputs are no longer separable and the ARX method cannot be used anymore. The klimabox identification using the subspace methods will be described in the following section.

9.2. Subspace identification method

Subspace identification methods offer an alternative to the classic recursive prediction error minimization methods (e.g. ARX); they utilize geometric projection and linear algebra, instead of numerical minimization algorithms. The subspace identification methods are used to identify parameters (matrices) of a linear time-invariant state space model from the input/output data. The name “subspace” denotes the fact that the model parameters are obtained from a row and a column subspaces of a certain matrix that is formed from the input/output data. Typically the column subspace is used to extract information about the model itself and the row subspace is used to obtain Kalman filter state sequences for the model Pavel Trnka [13]. More information can be found in Katayama [10], Ljung [11], van Overschee and de Moor [19].

Two different identification experiments were performed. Excitation of the system by a pseudo-random binary sequence was tried first and the result are described in section 9.2.1; secondly the system was excited by a series of step tests and this variation is described in section 9.2.2.

9.2.1. Pseudo-random binary sequence

An identification experiment was designed such that it randomly switches on and off all the controllable inputs and part of the disturbance inputs. A pseudo-random binary sequence (PRBS) was used to excite the actuators; the PRBS guaranties that all possible combinations of the actuators will be passed down to the actuators. Figure 9.2 shows the identification data together with the verification data; note that the density of the PRBS differs for the two data sets.

A N4SID (Numerical Algorithm for Subspace State Space System Identification) modification of the subspace method was used for the model identification. The model (called N4SID-PRBS here) has three states T_{in} , AH_{in} and T_c that correspond to the actual model outputs; that is convenient, especially when the model is in a canonical form, because the effect of each input can be examined directly from the model’s coefficients. Results of the identification are presented in figure 9.3. The model trajectories were obtained by simulating the model with the same input data as the real system was fed with. The NRMSE (Normalized Root Mean Square Error) fit is also stated in the figure; an average fit for the identification data is $\sim 70\%$, the validation data show an average fit of about 50%.

9. Black-box approach

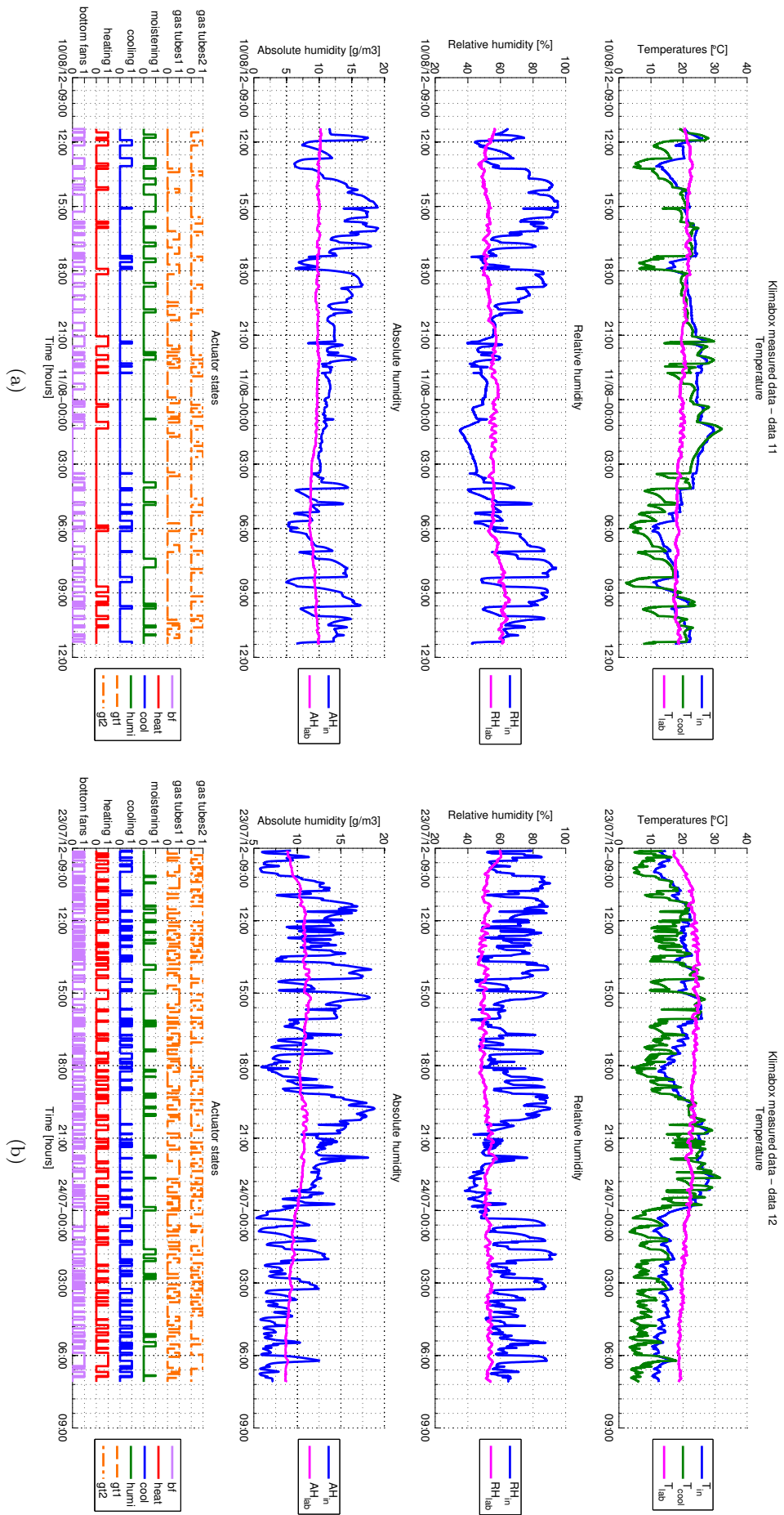
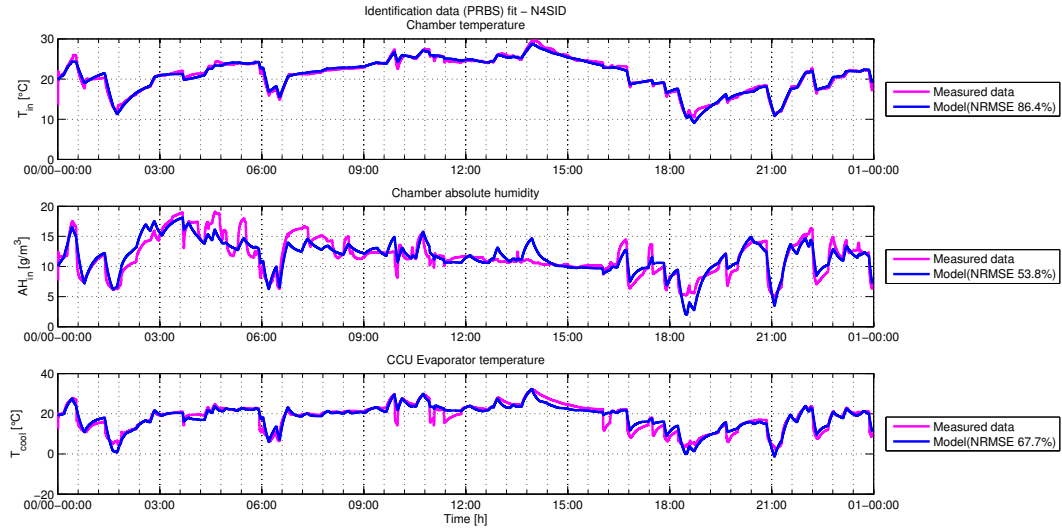


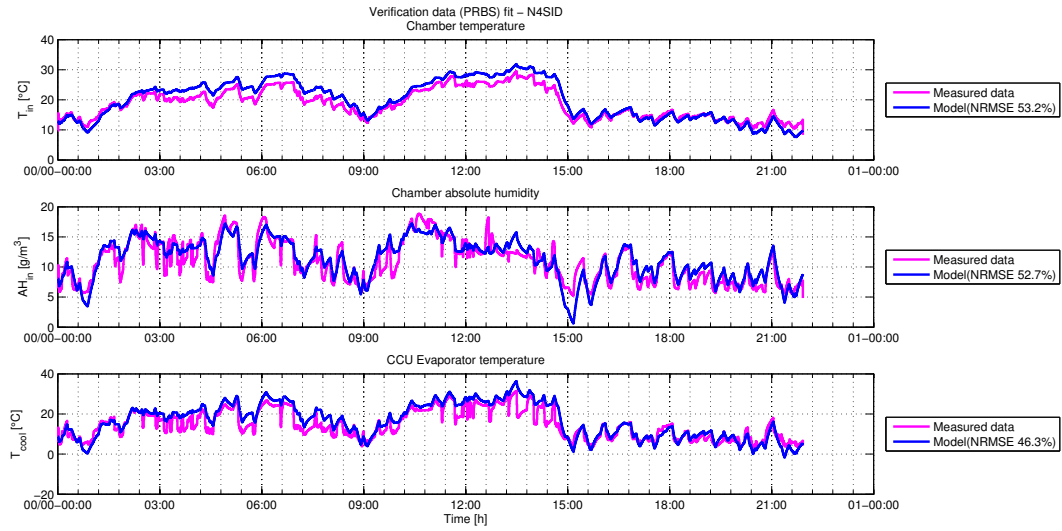
Figure 9.2.: Klimabox identification data (PRBS): identification data (a) and validation data (b).

Despite the rather poor fit, the model shows a strange dynamic behavior in the step responses (figure 9.3c). There is an unreasonable dynamics present in the step response from the CCU (input u_c) to the absolute humidity AH_{in} . Such dynamics are just not present in the klimabox, so the model is not good. The problem is in the model estimation or even a step before in the identification experiment design. A series of step tests were performed and they are going to be described in the following section.

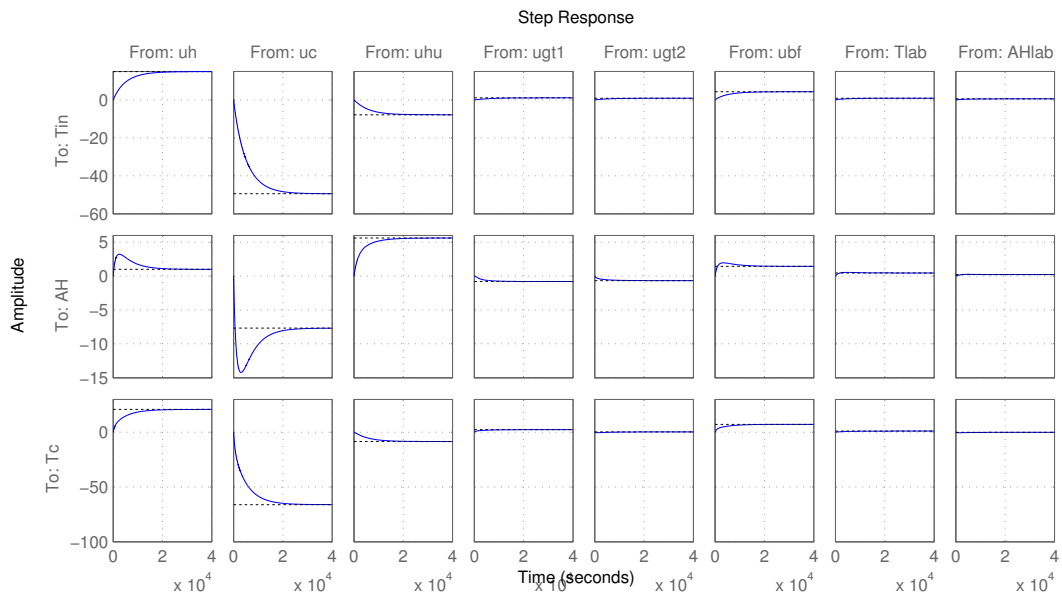
9. Black-box approach



(a)



(b)



(c)

Figure 9.3.: N4SID_PRBS model data fit and step response: the identification data fit (a), the validation data fit (b) and the step responses (c).

9.2.2. Step tests

Making the input sequence random, as it is in the case of PRBS, is useful when no prior knowledge is available about the system; while on the other hand, when the structure of the model is recognized, the identification experiment can be designed more effectively.

The identification experiment was designed so that the gathered data contain the same information that would be needed for a piece-by-piece Klimabox identification (as in sec. 8). Figure 9.4 shows the identification and the validation data obtained during the step tests.

The same N4SID algorithm was used to derive a new state space model (called N4SID_STEP). An average NRMSE model fit for the identification data is $\sim 80\%$; the validation data show an average fit about 70% .

9. Black-box approach

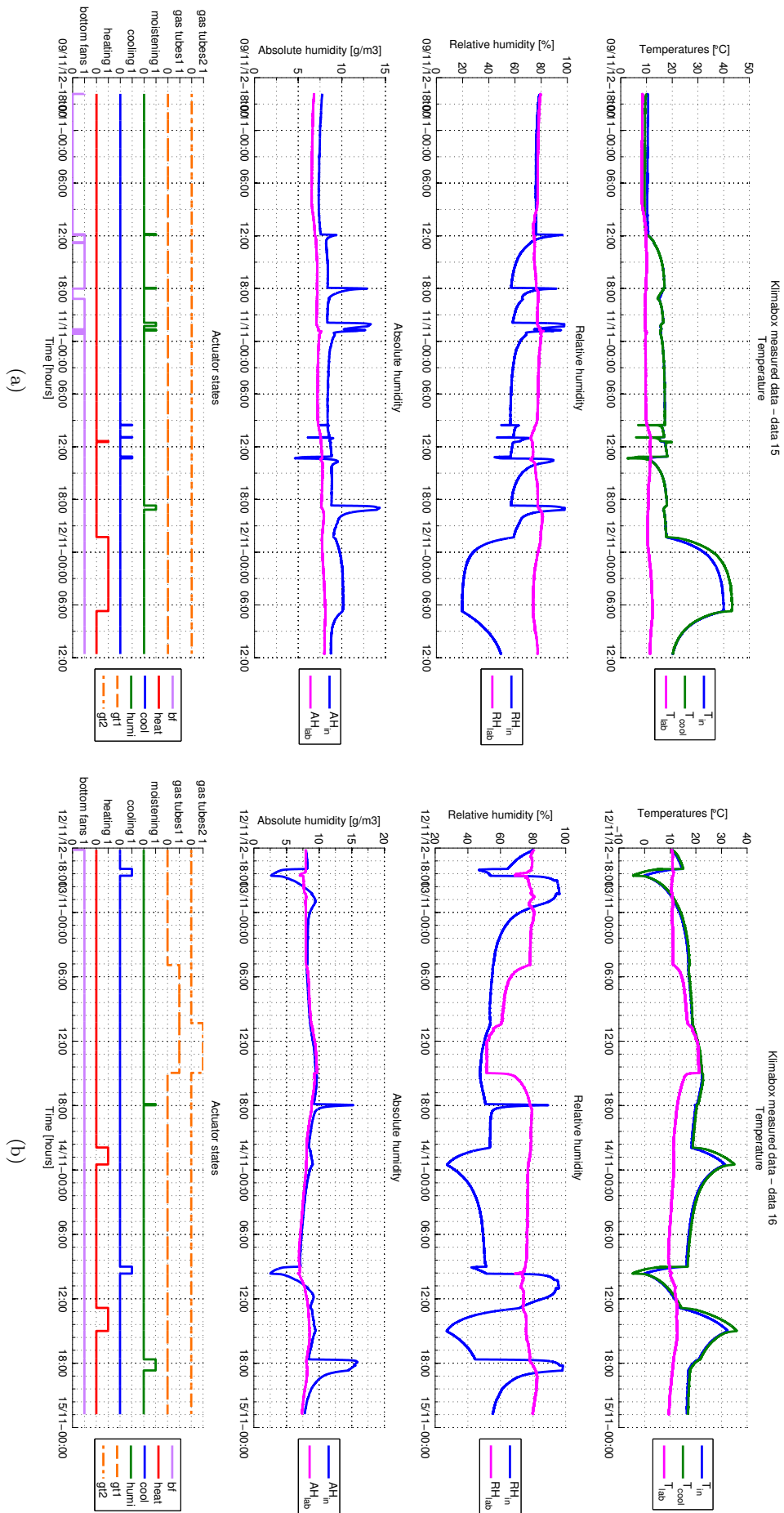
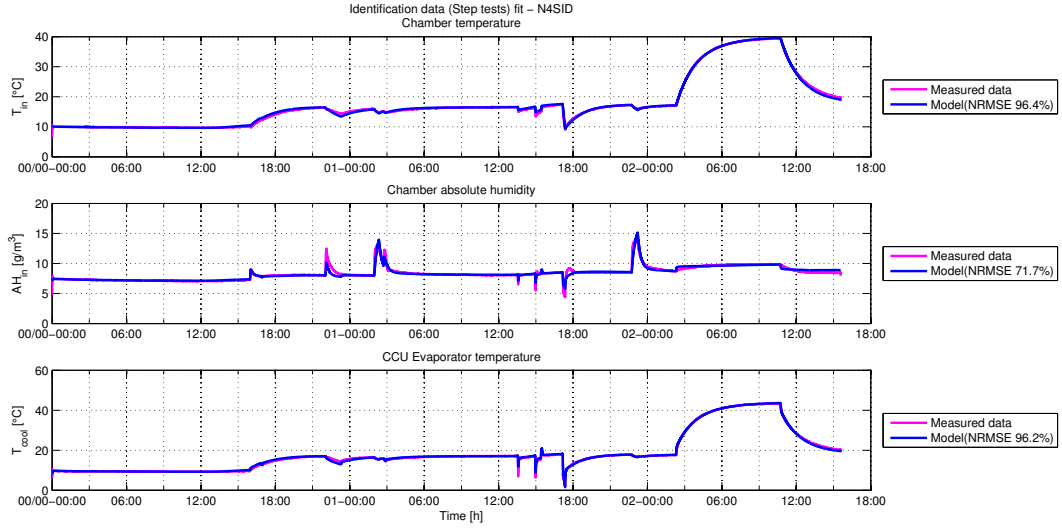
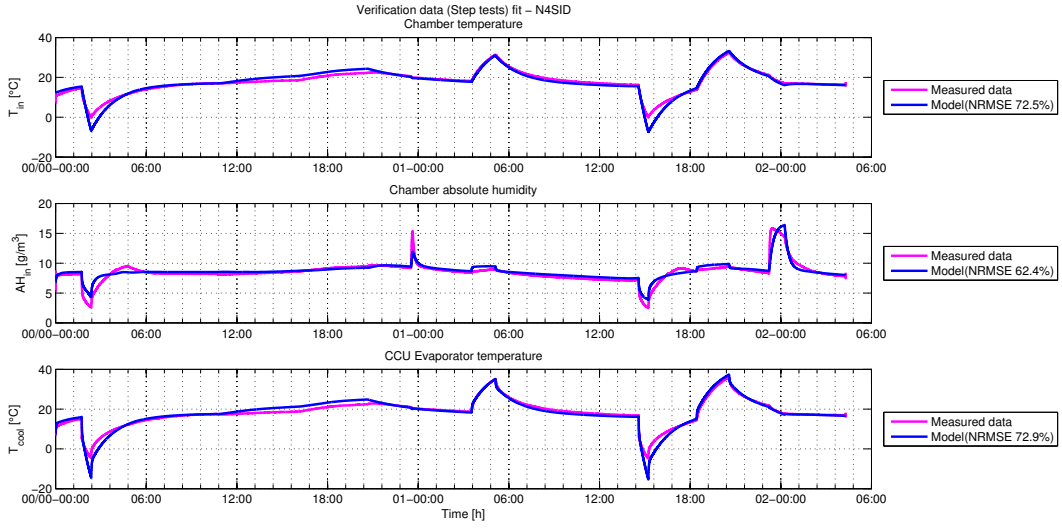


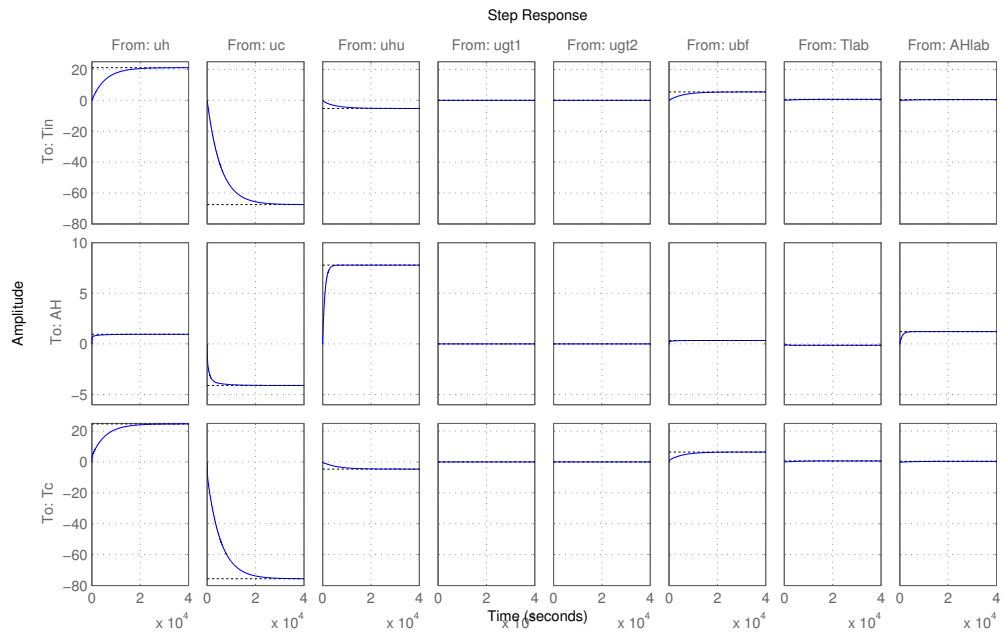
Figure 9.4.: Klimabox step-test data: identification data (a) and validation data (b).



(a)



(b)



(c)

Figure 9.5.: N4SID_STEP data fit and step response: the identification data fit (a), the validation data fit (b) and the step responses (c).

9.2.3. Model comparison

The N4SID_PRBS model shows 15% poorer fit to the data and it exhibits unreasonable dynamic behavior. The N4SID_STEP model has better fit and it has reasonable step responses.

A further diagnostics of the models was performed to prove their validity from different angles. Figure 9.6a displays a simulation of the N4SID_PRBS model, where all the actuators are turned off; it shows a significant drift of the temperature and the absolute humidity over time and the evaporator temperature has really unexpected trajectory. On the contrary, the simulation of the N4SID_STEP model (figure 9.6b) shows the drift and the unhealthy dynamics diminished.

Figure 9.6c shows a spectral density of the estimation error of the two models. The N4SID_STEP model estimation error spectral density is closer to that of white noise.

The mistake in creation of the N4SID_PRBS model occurred right at the identification experiment design; the identification data lacked steady state information and the identification algorithm was therefore fitting the model at higher frequencies. The density of the PRBS signal was too high.

The N4SID_STEP model identification data also have more samples and as the subspace identification methods are guaranteed to minimize the estimation error for the number of samples reaching infinity, the fit should also theoretically be better.

The N4SID_STEP model was chosen for the controller design.

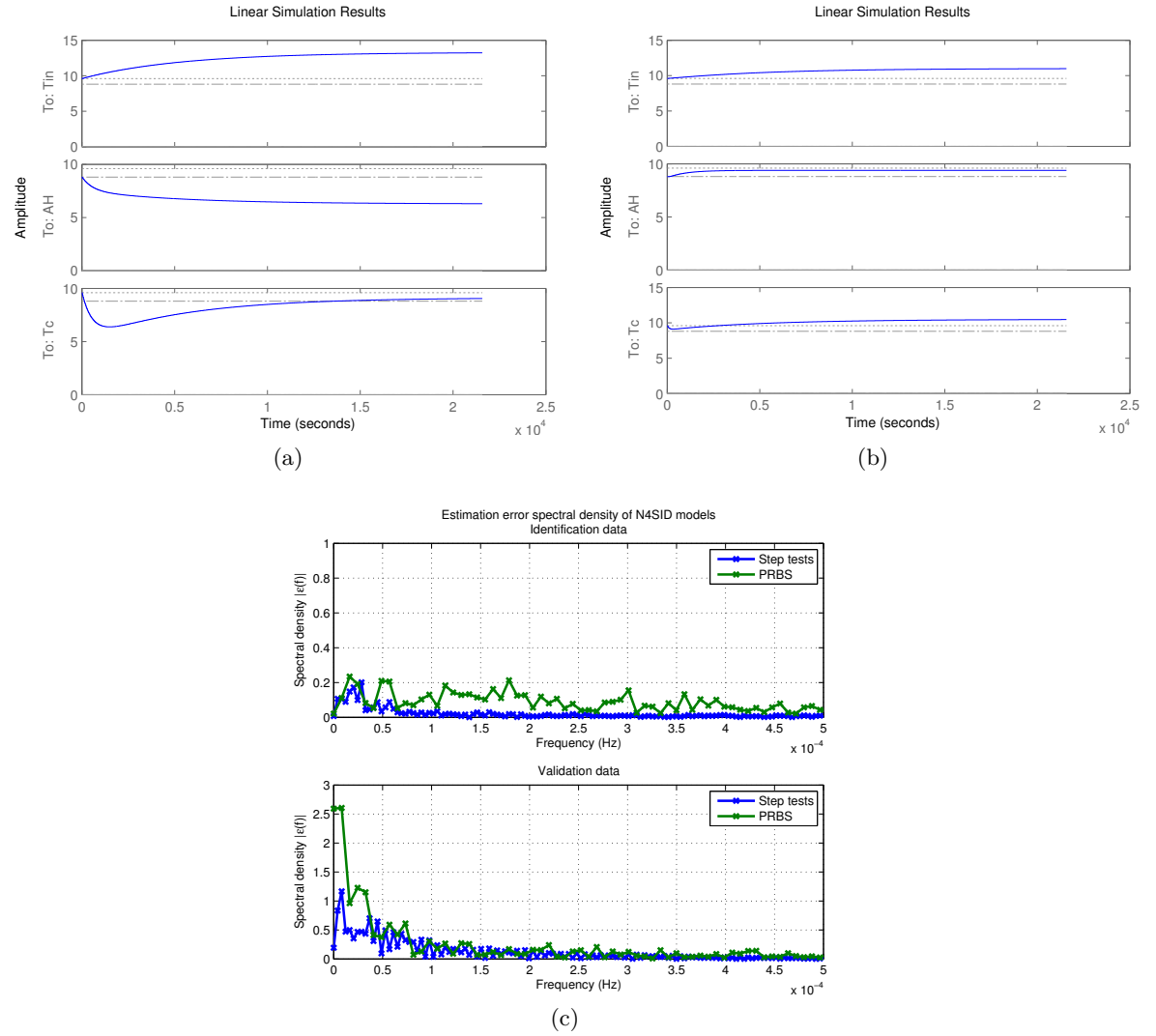


Figure 9.6.: Model comparison and diagnostics: Figure (a) shows the N4SID.PRBS model simulated with all inputs are turned off, figure (b) shows the same simulation for the N4SID.STEP model. Figure (c) presents a comparison of the estimation error spectral density of the two models.

Part IV.

Control

10. Controller introduction

A controller is an elementary part of the Klimabox device. It makes it possible to create the controlled climate inside the chamber. The controllers used in the Klimabox so far were feedback bang-bang (on-off) controllers with hysteresis realized either by analogous electronics (1985) or by a digital controller (2009). While the on-off controller is simple to realize and works well when controlling only temperature it does not provide energy efficient actuator-safe control and it fails for two dimensional systems (temperature and humidity).

According to the European Energy Efficiency Plan 2020, the total consumption of electricity is planned to drop by 20% [2]. Energy efficient control techniques could offer one of the tools to achieve this ecological goal.

A new optimization controller for the Klimabox was designed, the process of creating it and the results are going to be described within this part of the paper.

10.1. Model predictive control

Model predictive control, or MPC, is an advanced control system technique used in industry since the 1980s. MPC controllers, unlike the PID controllers, predict the change in system states/outputs and they can therefore optimize the future control actions. MPC controllers depend on a dynamic model of the controlled system.

Every sampling time a cost optimal control actions are calculated for the entire prediction horizon and only the first control action is applied. The next sampling time, the actual state of the system is measured, the control action is again calculated for the whole prediction horizon

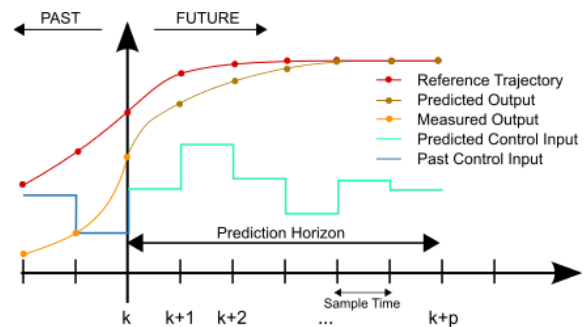


Figure 10.1.: Model predictive control scheme.

Credits: wikipedia.org

10. Controller introduction

and just like before, only the first value is applied. This principle is called “receding horizon mechanism” and it gives the otherwise open-loop control technique a feedback character, making the control method stable and robust. A schematic picture of the process is present in figure 10.1.

Important elements of the MPC are:

- a dynamic model of the controlled system
- an optimization cost function
- optimization constraints

A general optimization problem to be solved by MPC is defined as:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & \gamma(x) \leq 0 \\ & x \in \mathbb{R}^n, f: \mathbb{R}^n \rightarrow \mathbb{R}, \gamma: \mathbb{R}^n \rightarrow \mathbb{R}^m \end{aligned} \tag{10.1}$$

and it states that the goal is to minimize a general function $f(x)$ while the unknown x is subject to the constraints $\gamma(x) \leq 0$.

11. Klimabox controller

The environment in the climatic chamber, the temperature and the absolute humidity, is a dynamic system having multiple inputs (some real, some binary) and multiple outputs (real). Although this is a quite a complex scenario for the “classic” control methods, it fortunately poses no difficulties to the MPC controller.

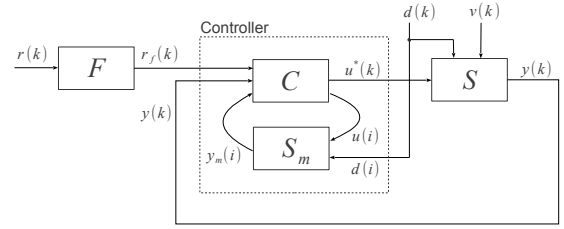


Figure 11.1.: Klimabox prediction control scheme

The control law is governed by the following minimization statement:

$$\begin{aligned} U^* &= \arg \min_{u_t, \dots, u_{t+N-1}} J(u, x, r) \\ \text{subject to} \quad & \text{constraints } \gamma(u, x, r) \end{aligned} \quad (11.1)$$

The solution to the minimization problem is a vector of optimal control actions taken upon the prediction horizon

$$U^* = \begin{bmatrix} u^*(0)' & u^*(1)' & \dots & u^*(N-1)' \end{bmatrix}'$$

from which only the first control action $u^*(0)$ is applied to the klimabox’s actuators; the next sampling time the whole minimization process is computed again. The optimization cost function $J(u, x, r)$ used for the klimabox controller is described in section 11.3; the constraints $\gamma(u, x, r)$ are described in section 11.4.

11.1. Klimabox dynamic model

A model used for the controller design is the N4SID_STEP model derived in sec. 9.2.2. The model inputs are divided for the control purposes into three subsets (figure 7.2):

11. Klimabox controller

1. Controllable inputs $u = \begin{bmatrix} u_h & u_c & u_{hu} \end{bmatrix}'$ (heating rod, CCU and humidifier)
2. Predictable disturbance inputs $d = \begin{bmatrix} u_{gt1} & u_{gt2} & u_{bf} \end{bmatrix}'$ (gas tubes, bottom fans)
3. Measured disturbance inputs $v = \begin{bmatrix} T_{lab} & AH_{lab} \end{bmatrix}'$ (laboratory temperature and absolute humidity)

The state update equation is then written as:

$$x(k+1) = Ax(k) + B_c u(k) + B_d d(k) + B_v v(0) \quad (11.2)$$

$$y(k) = Cx(k) \quad (11.3)$$

The predictable disturbance inputs are known into the future and the optimization process can count with their effect over the prediction horizon. Nevertheless, the measured disturbance inputs are not known into the future and the optimization algorithm treats them as constants during the whole prediction horizon.

11.2. Setpoint

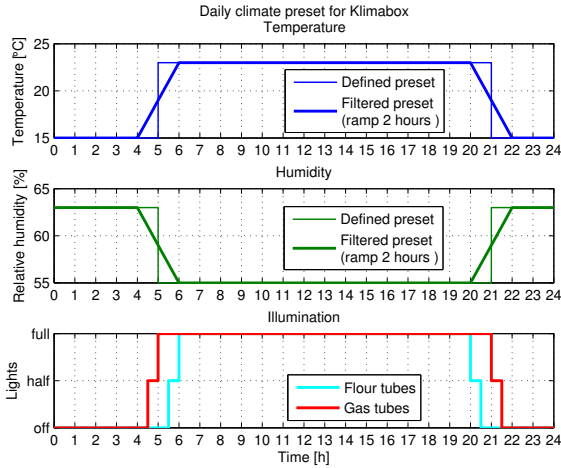


Figure 11.2.: Climatic preset: Pepper - prior to blooming period [7].

The setpoint (reference) $r(k) = \begin{bmatrix} r_{up}(k)' & r_{down}(k)' \end{bmatrix}'$ is part of the information that is input by researchers through the klimabox control panel, or alternatively it can be input directly to the server klimabox database. The chamber climate is defined by the temperature and relative humidity levels in half-an-hour intervals for one day. Climatic preset can also be defined for longer time periods to span for various phases of the biologic material growth such as for example germination, growth, prior to blooming period, blooming period and maturation.

An example climatic preset established for the pepper plant in its prior to blooming period is stated in figure 11.2 (this preset is used for controller simulations and for the real device testing). The figure shows the defined preset, which is a set of discrete values spanned by 30

minutes. The preset level changes in jumps and such a rapid changes, especially in the case of temperature, are not desired in the chamber as they do not occur in nature either. The climatic preset is filtered in the controller to provide smooth transitions.

11.2.1. Setpoint filter

The setpoint can be filtered by many various filters, a window width parameter has to be however stated for all of them. The window width refers to a time period in which the reference level transition must occur.

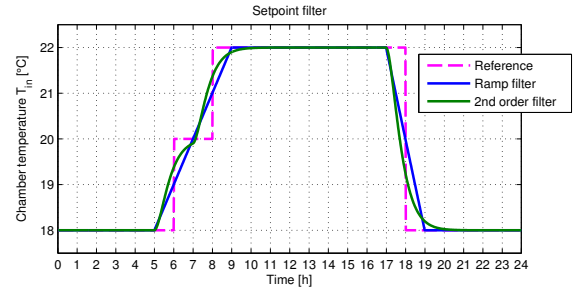


Figure 11.3.: Setpoint filter

One way of filtering the climate preset can be a simple ramping by a non-causal centered moving average filter described by:

$$r_f(k) = \frac{1}{w} \sum_{i=-w/2}^{w/2} r(k+i) \quad (11.4)$$

where w is the window width. Such a filter smooths the setpoint by a number of linear ramps and the effect can be seen in figure 11.3. The figure also shows the reference filtered by a non-causal second order filter with critical damping; the filter is described by the following system matrices:

$$A_f = \begin{bmatrix} 0 & 1 \\ -\frac{1}{T_0^2} & -\frac{2}{T_0} \end{bmatrix}, B_f = \begin{bmatrix} 0 \\ \frac{1}{T_0^2} \end{bmatrix}, C_f = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (11.5)$$

where T_0 is a time constant computed so that the transition settles to the 95% of the reference value within the window-width time. The filter state update function is:

$$x(k+1) = A_f x(k) + B_f r\left(k + \frac{w}{2}\right) \quad (11.6)$$

$$r_f(k) = C_f x(k) \quad (11.7)$$

11.3. Optimization cost function

The optimization cost function reflects the requirements laid upon the klimabox controller - the reference tracking, the energy efficiency and the actuator safe run. The cost function is

$$\begin{aligned}
 J(U, u(-1), x(0)) &= \sum_{k=0}^{N-1} \{J_y(k) + J_u(k) + J_{du}(k)\} \\
 J_y(k) &= (y(k) - z(k))' Q_z (y(k) - z(k)) \\
 J_u(k) &= u(k)' R u(k) \\
 J_{du}(k) &= (u(k) - u(k-1))' R_d (u(k) - u(k-1))
 \end{aligned} \tag{11.8}$$

where $U = \begin{bmatrix} u(0)' & u(1)' & \cdots & u(N-1)' \end{bmatrix}'$; $u \in \{0, 1\}^m$ is the sequence of the future binary inputs which is being optimized.

The reference tracking is ensured by the quadratic cost $J_y(k)$, where $y(k) = \begin{bmatrix} T_{in} & AH_{in} \end{bmatrix}'$ is the vector of the outputs at a time instant k , $z(k)$ is a soft constraint variable (described in section 11.4) representing the reference funnel and the weighting matrix Q_z is

$$Q_z = \begin{bmatrix} 1200 & 0 \\ 0 & 100 \end{bmatrix} \tag{11.9}$$

It is seen, that more importance has been put on the temperature control so that when the humidity reference cannot be reached, the controller still tracks the temperature.

The cost function $J_u(k)$ ensures that the energy consumption is being treated as well; the weighting matrix

$$R = \delta \begin{bmatrix} 1.57 & 0 & 0 \\ 0 & 4.5 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \tag{11.10}$$

contains the power consumptions in amperes of the heating rod, the CCU and the humidifier on the diagonal (from the top respectively) and δ is a scaling parameter.

The maintenance costs on the three actuators are represented by the cost function $J_{du}(k)$. Each change on the inputs is penalized by the following weighting matrix

$$R_d = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 400 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11.11}$$

The matrix depicts the fact that while the heating rod and the humidifier inputs can be freely changed, the CCU input should be planned not to vary too much; the reason is, an excessive switching would damage the CCU's compressor. The current state of inputs $u(-1)$ is fed into the cost function so that the change from the current state is also penalized.

11.4. Optimization constraints

Among many types of constraints that can enter the general optimization process, four constraint types are present in the klimabox controller: member constraints, hard equality constraints, hard inequality constraints and soft constraints.

The member constraints define a subset of values from which a variable value is drawn. All the manipulated input in the klimabox controller are binary, which is mathematically described as:

$$u(k) \in \{0, 1\}^m \quad (11.12)$$

, which also makes the optimization problem a mixed-integer NP-hard problem.

The hard equality constraints in the MPC are the system state update equations:

$$x(k+1) = A_c x(k) + B_c u(k) + B_d d(k) + B_v v(0) \quad (11.13)$$

$$y(k) = C_c x(k) \quad (11.14)$$

The hard inequality constraints in the klimabox controller define the state and output boundaries making the minimization process simpler.

$$-30 \leq x(k) \leq 50 \quad (11.15)$$

$$-30 \leq y(k) \leq 50 \quad (11.16)$$

The soft constraint variable $z(k)$ makes it possible to optimize the output within sliding boundaries, such as reference (setpoint) funnels etc. The following soft constraint defines the reference funnel for the klimabox controller:

$$r_{down}(k) \leq z(k) \leq r_{up}(k) \quad (11.17)$$

where $r_{down}(k)$, $r_{up}(k)$ are the lower and the upper reference levels respectively.

11.5. Klimabox MPC controller simulations

The controller was realized in Matlab[®] and the Yalmip[®] optimization language Löfberg [12]. The optimization solver used was the IBM's CPLEX[®] solver showing better performance than the MOSEK[®] optimization solver.

Many various simulations have been performed to find the right settings of the weighting matrices Q_z , R , R_d , the prediction horizon N and the sampling time T_s .

The prediction horizon length is closely related to the MPC computation time; the prediction horizon for the Klimabox controller was identified to span maximally 15 minutes, but to be $N = 20$ at maximum. Even though it seems that the MPC calculation can last up to the length of the sampling time, the control action based on the one-sampling-period-old data would be outdated; the faster the control action is applied the better. One Klimabox controller calculation for the prediction horizon $N = 20$ takes $0.2 - 3$ s, which is an admissible performance.

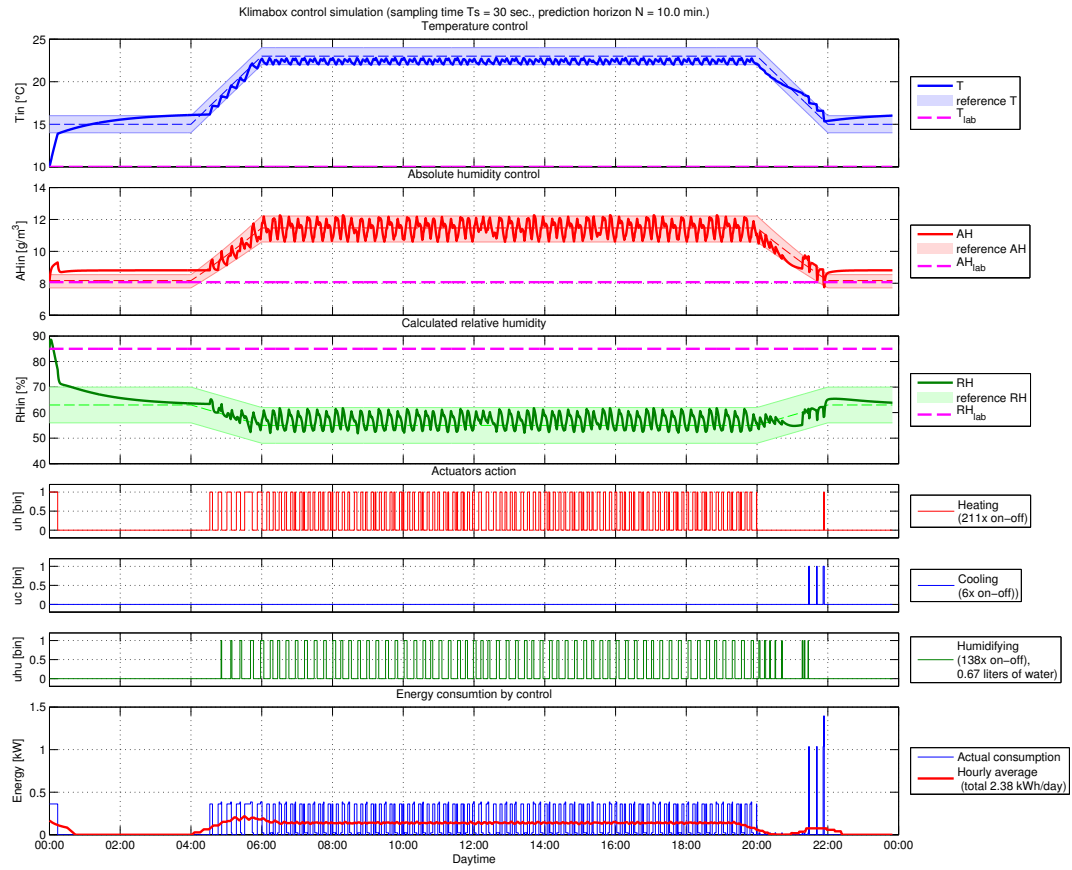
Four sampling times (10 s, 30 s, 60 s, 90 s) were available to choose from. The sampling time 10 s gives the maximum prediction horizon 3.3 minutes, which is not long enough. The sampling time 90 s on the other hand is too long to obtain reasonable control precision - the actuator states does not change for the length of the sampling time and 90 seconds is too long.

Figure 11.4 shows the klimabox control simulation of one day for the climatic preset defined in figure 11.2. The laboratory temperature and the absolute humidity affects the simulation severely; two scenarios were assumed - the approximate summer and winter conditions.

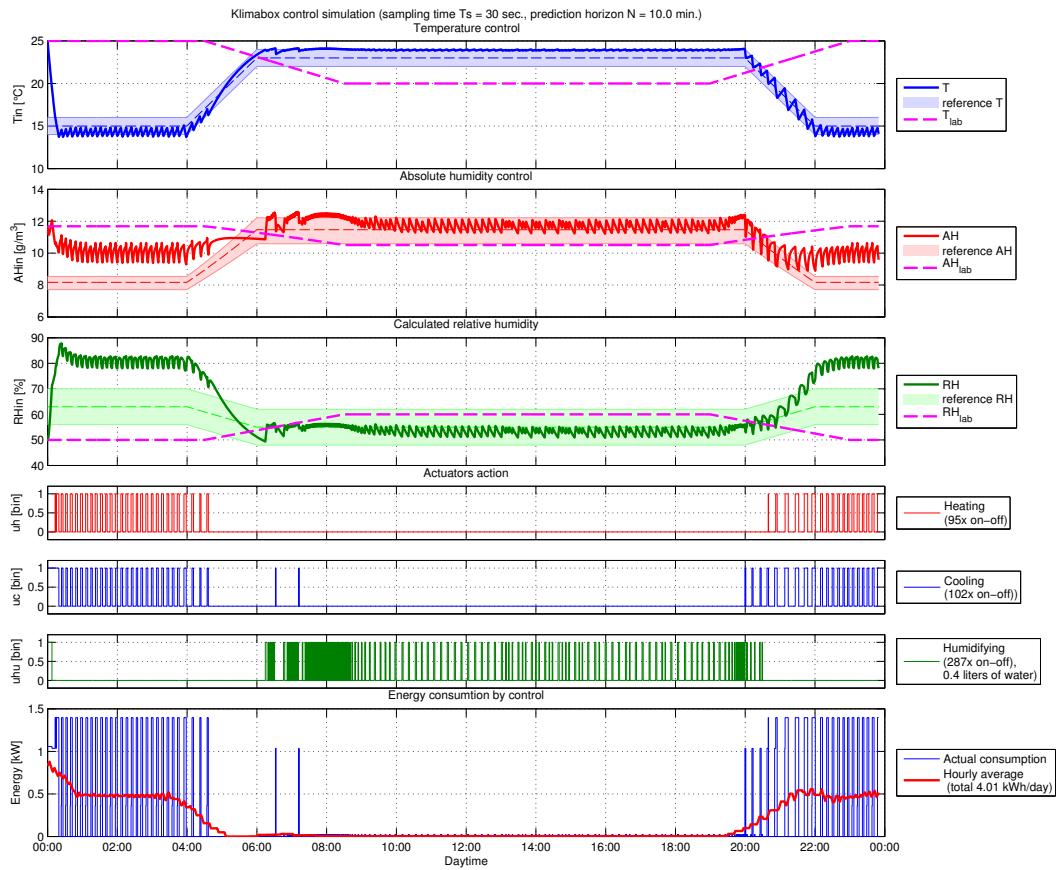
The winter simulation 11.4a shows that the outputs can be driven within the reference funnels while turning the CCU on just three times. A daily average energy consumption by the control was estimated to be 2.4 kWh/day .

The summer simulation 11.4b reveals the limited ability of the Klimabox to dry the chamber's air over a long period of time; the water vapor pressure equalization rate is just too high for the controller to keep up. Unless the absolute humidity reference is not achieved, the temperature is still controlled without major deviations. The CCU was turned on 51 times and that also had impact on the energy consumption caused by the control - leaving it to be 4 kWh/day (illumination not included).

11.5. Klimabox MPC controller simulations



(a)



(b)

Figure 11.4.: Klimabox MPC controller simulations: in winter conditions (a), in summer conditions (b)

11. Klimabox controller

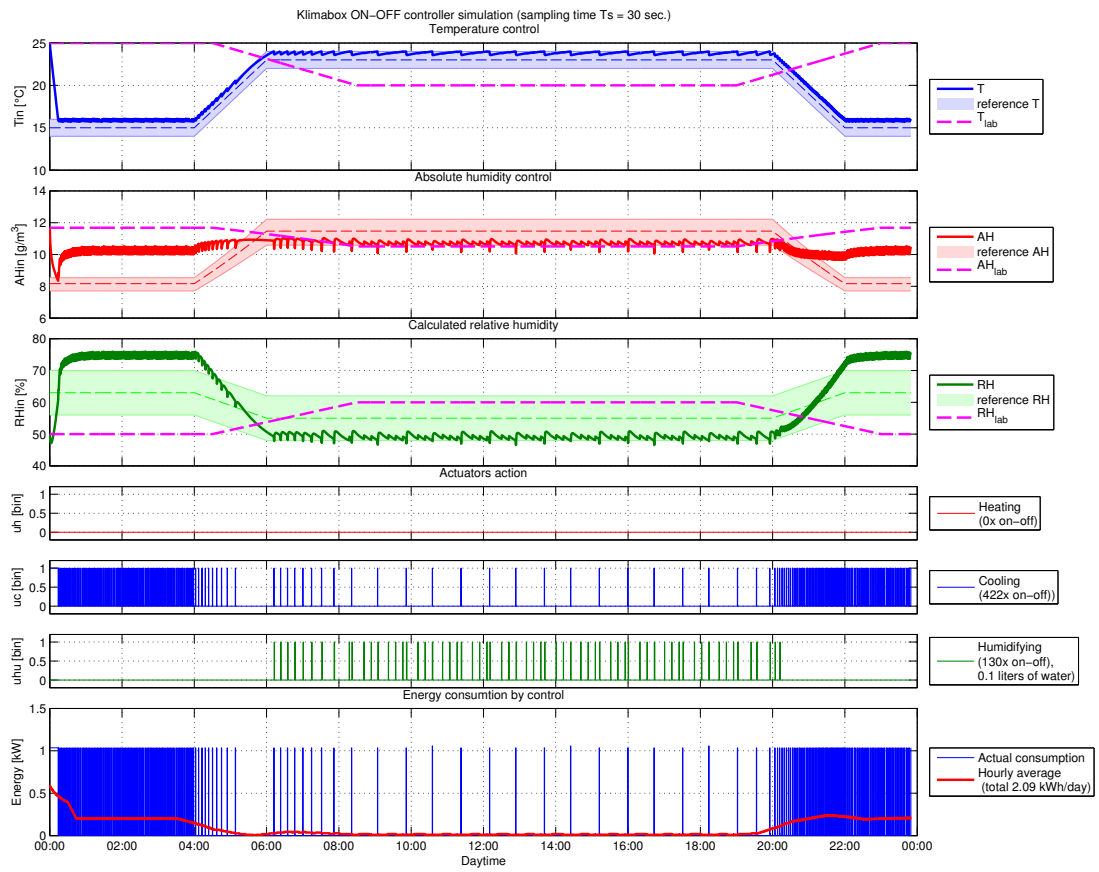
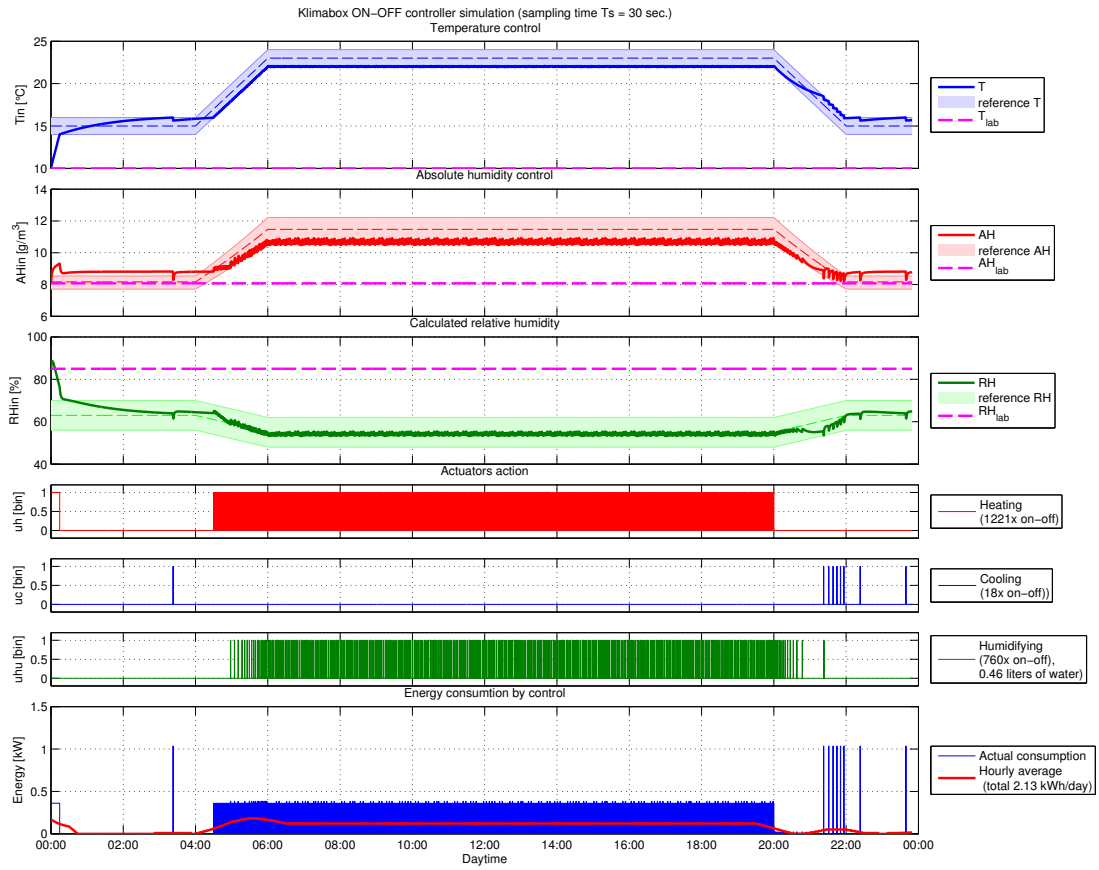


Figure 11.5.: Klimabox ON-OFF controller simulations: in winter conditions (a), in summer conditions (b)

11.6. MPC vs. ON-OFF controller

Figure 11.5 shows the same two simulations as simulated for the MPC in previous section, but this time the controller is a bang-bang (ON-OFF) controller. The ON-OFF controller simply turns on the heater, when the temperature gets under the lower bound of the reference funnel and it turns the heater off if the temperature returns into the reference funnel. So it does with the CCU and the temperature climbing over the reference funnel. When the absolute humidity is too low the controller turns on the humidifier; there is however no control action to decrease the absolute humidity, when it gets over the reference funnel.

Figure 11.5a shows the ON-OFF controller simulation in winter conditions. The control performance comparable to the one of the MPC; the “summer” simulation (11.5b), on the other hand, uncovers the ON-OFF controller weaknesses. The controller is unable to dry the air in the chamber (although the MPC is not very successful either) and it switched over the CCU 422 times, which compared to the 102 times for the MPC is more than 4 times more to achieve the same reference trajectory.

When the reference funnel is tight, the ON-OFF controller’s action on one edge of the reference funnel drives the output over the other edge generating the opposite control action - that causes constant oscillations as the simulation shows in figure 11.6. The MPC simulation of the same situation displays the controller’s ability to predict the system behavior into the future and therefore saving the actuators and the energy. The mean temperature output is at the reference level, although in the case of the ON-OFF controller it is not.

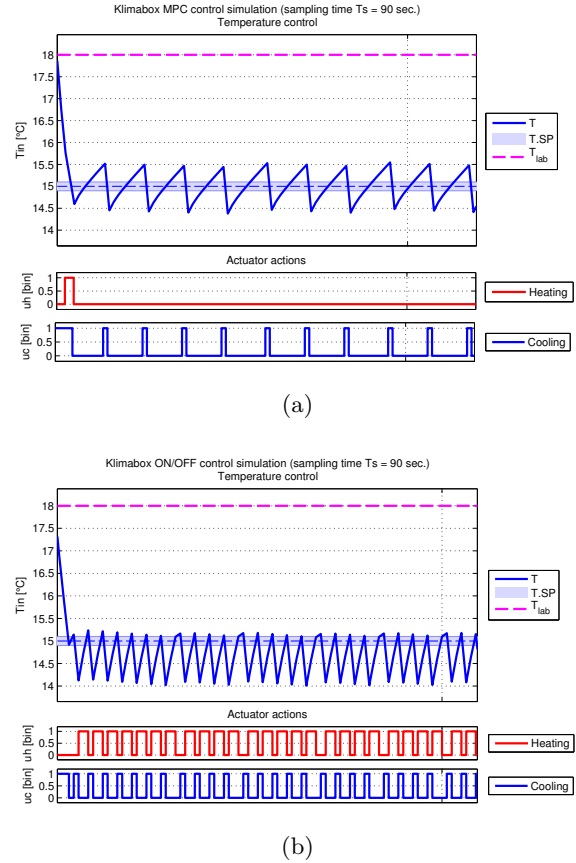


Figure 11.6.: MPC vs. ON-OFF controller for tight reference funnel: MPC (a), ON-OFF (b).

11.7. Klimabox MPC controller application

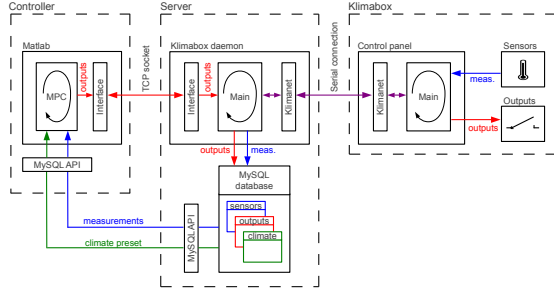


Figure 11.7.: Klimabox control process communication scheme

The klimabox controller simulations had served to tune the controller parameters and once the controller was tuned, it was time to try it for real. The communication scheme 11.7 shows the flow of information through the different parts the communication chain (part II) during the control process. Needless to say, it was worth the effort make the communication lines robust and lossless. After

some synchronization tweaks, everything was finally ready to launch the controller.

A control with the sampling time 60s was performed first; having the actuators turned on for at least 60s however resulted in some oscillations occurring due to the prediction errors. The prediction error cannot add up that high using the sampling period of 30s and a smaller output corrections can be achieved too. The sampling time 30s was selected to serve the best for the Klimabox control purposes.

Figure 11.8 displays real climate trajectories in the climatic chamber; the climate preset used is the one presented in figure 11.2, i.e., the one designed for the prior-to-blooming period of the pepper plant. Figure 11.8a shows the environment obtained with the MPC controller in a setting 1, having the maintenance costs R_d set to

$$\text{setting 1: } R_d = \begin{bmatrix} \text{cost } du_{\text{heat.}} & 0 & 0 \\ 0 & \text{cost } du_{\text{cool.}} & 0 \\ 0 & 0 & \text{cost } du_{\text{hum.}} \end{bmatrix} = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 10 \end{bmatrix}$$

The controller succeeds to control the temperature and the absolute humidity; the temperature with a greater importance than the absolute humidity (defined by the weighting matrix Q_z in 11.9). It used the cooling coil unit (CCU) for 74 minutes - turning it on 37 times. The night transition was however not very smooth and the number times the CCU was used was determined to be unreasonably high.

The the maintenance costs were adjusted to be

$$\text{setting 2: } R_d = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 400 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

which, as opposed to the setting 1, favors the usage of the heating rod and the humidifier before the CCU. Figure 11.8b shows the climate chamber environment being controlled with the controller in setting 2 (the laboratory conditions were very similar the ones of the previous experiment). The controller, this time, used the CCU about 2.5 times less; the CCU was turned on for 28 minutes and switched over $28x$. The comparison between the two figures also reveals that in the second case the humidifier was turned much more frequently; it was switched on for 6.5 time longer. The reason is not the humidity control as one would expect, but surprisingly the humidifier is used to control the temperature. Instead of turning the CCU on, the chamber is cooled down by the evaporating water droplets from the humidifier. The controller is here turning a weakness into a strength by exploiting the fact that the absolute humidities of the laboratory and the chamber equalize quickly.

Figure 11.8b also shows that at approximately 10:00 AM the humidifier's tank ran out of water. The energy consumption caused by controlling the temperature and absolute humidity was calculated to be 1.7 kWh/day ; the real energy consumption was not measured.

A controlled climate in the Klimabox was achieved.

11. Klimabox controller

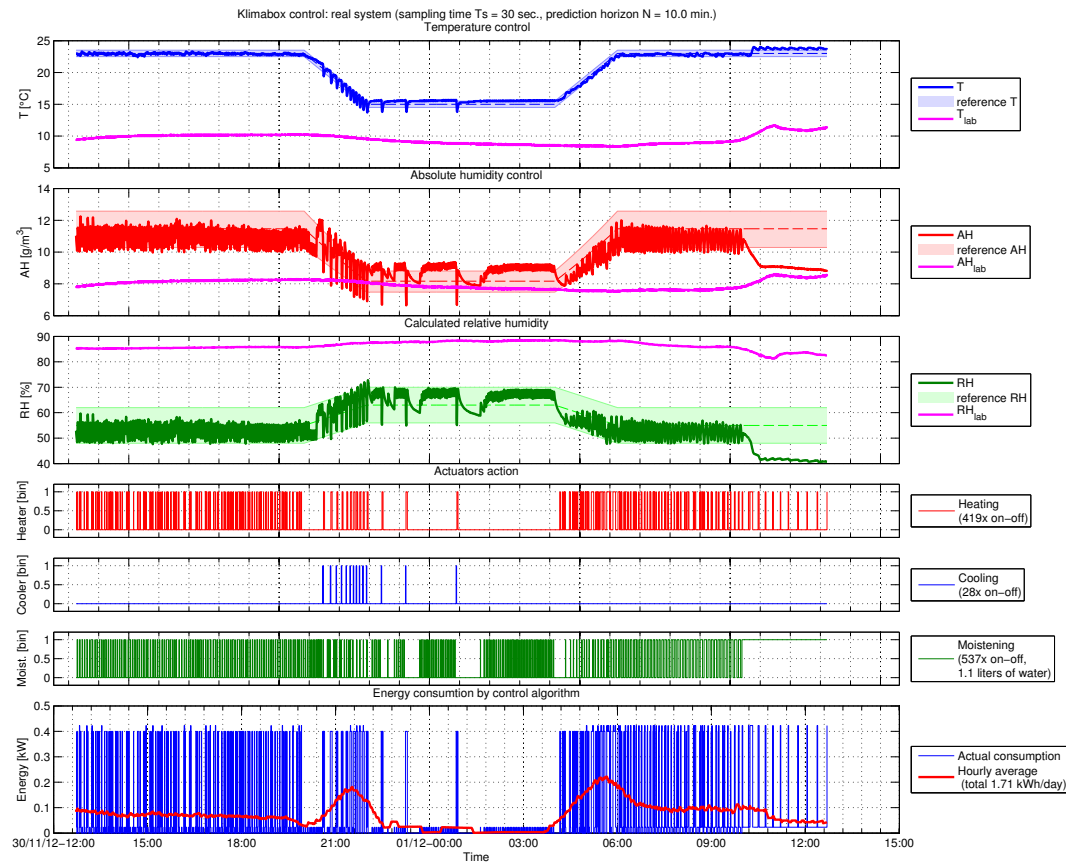
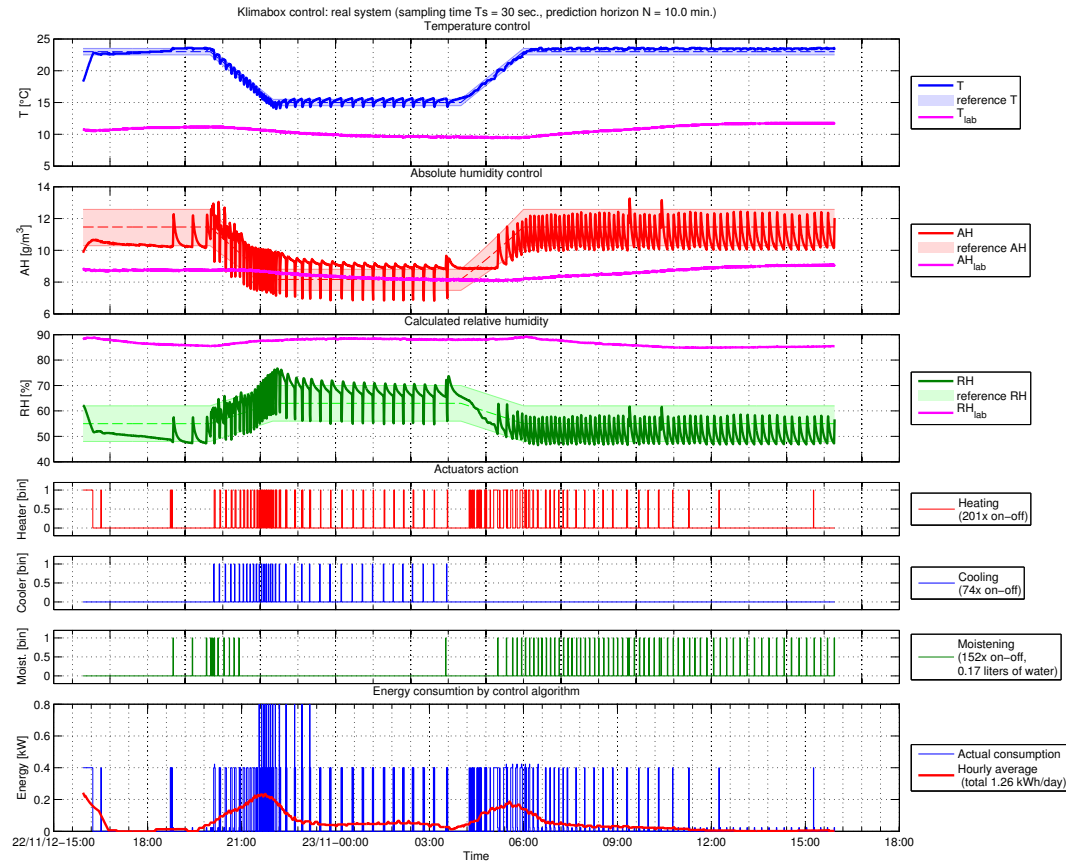


Figure 11.8.: Klimabox control (real data): setting 1 (a), setting 2 (b).

12. Future development

A possible future development is proposed in this section.

12.1. LED lamps

The chlorophyll spectrum (figure 12.1) peaks at 450 nm and 680 nm, there is no light absorption in the green band. LED diode is a source of monochromatic light, i.e., it emits light of just one frequency and using it could save energy.

There has been a big development in commercialization of LED grow lamps during the past few years. The LED grow lamps can be constructed so that they emit light in such wavelengths that correspond to the chlorophyll spectrum peaks (more description in section 1.3.1). The LED lamps save energy by emitting light only in the wavelengths that vegetation can absorb (the blue and red bands).

The LED lamp producers claim that LED lamps save 75% of the consumed energy while having the same light intensity in the red and blue band as the HPS lamps (gas tubes).

In the daily preset used for the klimabox controller testing (figure 11.2), the gas tubes are turned on for 17 hours consuming 17 kWh/day, but most of the power is turned into heat instead of light. Replacing the gas tubes by the LED lamps would result in a reduced energy consumption of only 4.25 kWh/day.

Further saving can be achieved by pulsing the light fluxes from the LED lamps. A paper by Tennessen et al. [17] states that when the light is emitted for only 1% of the duty cycle at frequency 5 kHz, the light utilization efficiency

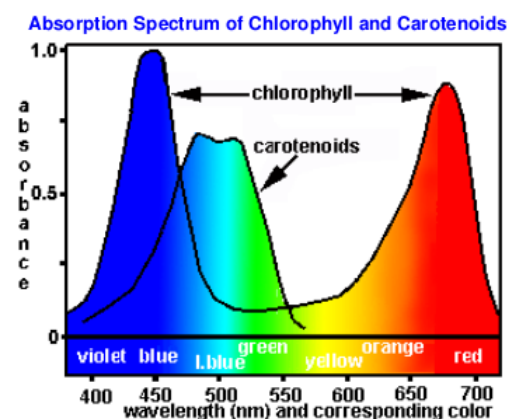


Figure 12.1.: Absorption spectrum of chlorophyll and carotenoids [15].

12. Future development

by the tested plant was 27 times higher than the utilization of a continuous light flux of the same emitted power. LED lamps are capable of flickering at high frequencies and what more, the LED diodes can be overdriven, when turned on only for short impulses. The LED lamps driven this way would theoretically save $\sim 95\%$ of the energy with the same speed of the vegetation growth. The total energy consumption by the lamps in my test case would be just 0.16 kWh/day providing the same growth as the gas tubes (17 kWh/day).

However, it has to be taken into account that the pulsed spectral shaped light is only suitable for a plant growing. It can significantly reduce energy consumption when the Klimabox is used for plant production, but the current light sources should stay in place to fulfill the multipurpose character of the device.

12.2. Explicit MPC

The Klimabox is now controlled from a connected computer, where the MPC is computed on-line every sampling time. Microcontrollers nowadays are not capable of computing the optimization problem on-line, there is however a way of precomputing the controller off-line a storing it as a look-up table; it is called the explicit MPC.

The control action is computed for the whole admissible state space. As the control action is likely to have a similar control action for a nearby states, the state space is clustered into numerous patches (polytopes), where each one is gathering all the neighboring states with the similar control action. As a result, the control action is a piece-wise linear function of the state, which can be stored in a look-up table. Reading the lookup table and applying the control action causes no problems to the microcontroller.

It is however complicated to compute the explicit controller in reasonable time. Further simplifications to the Klimabox model has to be made, but once the explicit controller is calculated, the Klimabox will no longer rely on a helping computer!

12.3. Control panel connectable to the Ethernet

I already have an Ethernet module that is connectable to the current klimabox control panel. The idea is to connect the Klimabox straight to network (the Internet), so that the data processing or the controller calculations can be done off-site in any network connected computer; the computer will no longer have to stay next to the Klimabox and one computer

could serve numerous Klimaboxes. Another advantage of the Ethernet module is that the control panel can be powered over the network cable.

13. Conclusion

The main goal of this project was to design and realize a controller capable of controlling the environment inside a climatic unit. The controller was designed for a specific device - the KLIMABOX (described in chapter 1); the realized process is however applicable to any device incorporating a controlled climate, such as ovens, greenhouses or warehouses. It was desired to control the temperature, humidity and illumination inside the chamber efficiently according to the daily climate preset and with respect to the limitations of the Klimabox's actuators.

First, the psychrometric properties of moist air were studied. It was noted that, while the temperature and relative humidity are mutually highly dependent variables, the absolute humidity varies with the temperature only minimally and this uncoupling is very convenient for the control process. The absolute humidity is also pressure dependent, but as the pressure changes influence the laboratory environment the same as the climatic chamber, it does not affect the control process. The psychrometric properties of moist air and their calculations are described in section 2.2. The main variables describing the state of moist air for control purposes were selected to be the temperature and the absolute humidity.

13.1. Data acquisition and command signaling

The microcontroller-based Klimabox control panel, described in chapter 3, was fitted with two new sensors and it was connected to the computer by the serial connection. The necessity of bilateral communication over the serial line led to the design and implementation of a new communications protocol KLIMANET (chapter 5); the idea had already been proposed in my bachelor thesis Dostal [6]. The microcontroller software was completely rewritten, aiming for code lightness, to be able to accommodate the Klimanet routines in the microcontroller's program memory. The main microcontroller's program now provides many new features such as data storage, better handling of the peripherals, the communication protocol, etc., while being 22% smaller than the previous one; implementation details are described in section 3.3.

A new computer program called KLIMABOX DAEMON was programmed to cope with the Klimanet on the other side of the serial cable - in the server. The Klimabox daemon program, described in section 4.1, takes care of reading the sensor data from the Klimabox control panel and storing it in the database (4.3). It also provides the TCP connectable interface (section 4.2) for the Klimabox remote operation. The controller is connected to the Klimabox by this interface. The information flow is schematically pictured in figure 4.1.

The humidity sensor's reliability was questioned during the first measurements. Three humidity calibration methods were proposed and realized, as described in section 6.1; the results of the calibration by aqueous salt solutions proved that the SHT11 relative humidity sensor was measuring correctly. The reason why the early measurement were spoiled, was the lack of appropriate fanning of the sensor; extra sensor-fanning equipment was installed into the Klimabox. Also a refined way of conversion of the sensor readouts using only integer algebra was proposed (section 6.2); the maximal deviation from the manufacturer's formula is only $\pm 0.4\%$ RH, whereas the speed gained and the code size saved is significant for the microcontroller.

13.2. Identification

An identification of the Klimabox dynamic model was approached by the first principles and the black-box methods. The first principles approach, described in chapter 8, was realized first, giving an idea of the dynamic processes happening in the Klimabox. A Simulink library of the Klimabox elementary dynamic components was developed (section 8.2) and the Klimabox simulator was built from the components (section 8.3). Although the model simulates the temperature dynamics well, a model of effects of the cooling coil unit (CCU) on the absolute humidity was not finished. A model of complex non-linear processes regarding the CCU's evaporator, described in 8.1.2, was not found. The evaporator temperature was however found to be an important state variable having a great effect on the chamber's climate; a thermometer was added to the evaporator.

An identification by two black-box identification methods was performed to avoid modeling of the CCU. The ARX models of the chamber temperature and absolute humidity were found as described in section 9.1. Although the method supports only the SISO and MISO systems, a model of a MIMO system can be derived, when the outputs are decoupled. The newly added evaporator temperature model output is however no longer decoupled from the other two outputs; the system became a pure MIMO system and it was identified by the subspace

13. Conclusion

identification method instead.

The model found by the subspace identification algorithm N4SID (section 9.2) shows the average NRMSE fit to be 82% for the identification data and 70% for the validation data. The identification experiments are described in sections 9.2.1 and 9.2.2. The model derived by the subspace identification shows a good fit and reliable static and dynamic behavior; the model is also simple enough (only 3 states) for the controller optimization calculations. The model derived in section 9.2.2 was selected for the controller.

13.3. Control

The model predictive controller (MPC), realized in Matlab[®] and the Yalmip[®] optimization language Löfberg [12], was implemented; the optimization solver used was the IBM's CPLEX[®] solver. The optimization cost function, described in section 11.3, reflects the control requirements laid upon the controller - the reference tracking, energy efficiency and respect the actuators' operation limits. The optimization constraints (section 11.4), among other constraints, define the actuator's inputs to be binary, which makes the optimization a mixed-integer NP-hard problem. The sampling period 30 s was selected for a reasonable trade-off between the prediction horizon length and the actualization period of the actuators' states.

The climate preset (setpoint), defined for every 30 minutes in a day, is filtered by the non-causal centered moving average filter described in section 11.2. The discrete jumps in the reference are replaced by gradual linear transitions; the length of the transitions can be altered through the Klimabox control panel.

The simulations of the Klimabox controller (section 11.5) show that the controller tracks the reference and that it exploits the entire reference funnel to save unnecessary switching of the CCU whenever possible. They also show the limited ability of the Klimabox actuators to decrease the absolute humidity inside the chamber; the water vapor pressure equalization rate is very high and there is no efficient way to dry the air. Although the rapid water vapor pressure equalization is a weakness of the Klimabox design, it was turned by the MPC controller into a strength; as can be clearly seen from the real control data (figure 11.8), the controller uses the humidifier to cool down the chamber's air and thus it saves the CCU switching. When the water droplets from the humidifier evaporate, the latent heat of phase change is extracted from the surrounding air, which causes the temperature reduction.

The real control data were obtained in winter, when the laboratory temperature was lower than the preset climate temperature. It shows the calculated energy consumption of the heating rod, the CCU and the humidifier together to be $\sim 1.5 \text{ kWh/day}$; the control energy consumption in summer is expected to be $\sim 4 \text{ kWh/day}$ due to the necessity of cooling. It is however still only 15% of the energy consumed by the Klimabox in total. For a daily illumination time of 16 hours the power used for the illumination is 21.6 kWh/day . Most of the energy is however converted into heat instead of light and that is why the usage of LED lamps was proposed (section 12.1). If the Klimabox was used for plant production, the pulsed LED lamps would theoretically save $\sim 95\%$ of the energy for the illumination, consuming only 0.2 kWh/day while providing the same carbon production in the vegetation. An educated usage of the LED lamps would reduce the Klimabox energy consumption more than any controller could ever achieve by energy efficient control of the actuators.

The main benefit of the MPC, aside from the capability of controlling both the temperature and the humidity, turned out to be the minimization of the actuator switching. Excessive CCU switching would cause damage to the compressor and it would increase the noise level of the Klimabox operation.

13.4. Summary

The necessity of the duplex communication with the real Klimabox device led to the creation of various software tools; they allowed the Klimabox to be operable over the Internet. The Klimabox model was identified using the subspace identification method and the model predictive controller was designed and applied. A better way of controlling a climatic unit was found!

The realization of this project provided me with a great way of learning the modern control techniques in practice and the Klimabox is now ready for the continuation of my grandfather's biological research.

Bibliography

- [1] Relative Humidity Sensor Rh-BTA. Technical report, Vernier Software & Technology, Beaverton, 2007.
- [2] Energy 2020 - A Strategy for Competitive, Sustainable and Secure Energy. Technical report, European Commission, 2010.
- [3] Avr-libc User Manual [online; accessed 06/11/10], 2012. URL www.nongnu.org/avr-libc/.
- [4] Arden Buck. New equations for computing vapor pressure and enhancement factor. *Journal of Applied Meteorology*, 1981. URL http://www.public.iastate.edu/~bkh/teaching/505/arden_buck_sat.pdf.
- [5] Jiri Dostal. *Automatické ovládaní klimatické komory*. High school graduation project, 2005.
- [6] Jiri Dostal. *Regulace klimatické jednotky*. Bachelor thesis, Czech Technical University in Prague, 2009.
- [7] Josef Dostal. Vývoj československého klimatizovaného boxu. Technical report, Výzkumný ústav rostlinné výroby Praha 6 - Ruzyně, 1985.
- [8] Peter Fleury. ACR-GCC Libraries [online; accessed 25/10/11]. URL <http://jump.to/fleury>.
- [9] Vladimír Havlena and Jan Stecha. *Moderní teorie řízení*. Edicní středisko CVUT, Prague, 1999.
- [10] Tohru Katayama. *Subspace Methods for System Identification*. Springer, London, 2005. ISBN 9781852339814.
- [11] Lennart Ljung. *System Identification: Theory for the User*, volume 11 of *Prentice-Hall information and system sciences series*. Prentice Hall, 1987. ISBN 0138816409. doi: 10.1016/0005-1098(89)90019-8. URL <http://linkinghub.elsevier.com/retrieve/pii/0005109889900198>.

- [12] J Löfberg. YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In *Proceedings of the CACSD Conference*, pages 284–289. IEEE, IEEE, 2004. ISBN 0780386361. doi: 10.1109/CACSD.2004.1393890. URL <http://control.ee.ethz.ch/~joloef/yalmip.php>.
- [13] Pavel Trnka. *Subspace Identification Methods*. Doctoral thesis, Czech Technical University in Prague, 2007.
- [14] Portland State Aerospace Society. A Quick derivation relating altitude to air pressure. Technical report, 2004. URL <http://www.psas.pdx.edu>.
- [15] D.W. Reed. *Horticulture - Science and Practices*. Pearson Custom Publishing, Boston, 1999.
- [16] Guido Socher. Temperature and humidity measurements with the AVR web-server [online; accessed 24/05/12]. URL <http://tuxgraphics.org/electronics/200709/avr-webserver-sensirion-humidity-temperature.shtml>.
- [17] DJ Tennessen, RJ Bula, and TD Sharkey. Efficiency of photosynthesis in continuous and pulsed light emitting diode irradiation. *Photosynthesis research*, 1995. URL <http://www.springerlink.com/index/N520108G8T85U811.pdf>.
- [18] Martin Thomas. DS18x20 with AVR [online; accessed 12/06/12]. URL http://www.siwawi.arubi.uni-kl.de/avr_projects/tempsensor/index.html.
- [19] Peter van Overschee and Bart de Moor. *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Kluwer Academic Publishers, 1996.
- [20] Holger Vömel. Saturation vapor pressure formulations [online; accessed 25/10/12], 2011. URL <http://cires.colorado.edu/~voemel/vp.html>.
- [21] Luther R. Wilhelm. Numerical Calculation of Psychrometric Properties in SI Units. *Transactions of the ASAE*, pages 318–325, 1976.
- [22] WMO. Guide to meteorological instruments and methods of observation. 7(8):681, 2008. URL <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Guide+to+Meteorological+Instruments+and+Methods+of+Observation#0>.

A. Contents of the CD attached

The CD attached contains an electronic version of this thesis.