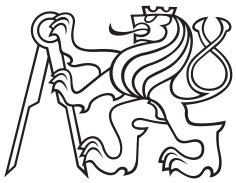
Master Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Control Engineering

Distributed Building Identification

Bc. Tomáš Bäumelt

Supervisor: Ing. Jiří Dostál
Field of study: Cybernetics and Robotics
Subfield: Systems and control
May 2016

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Tomáš Bäumelt**

Studijní program: Kybernetika a robotika
Obor: Systémy a řízení

Název tématu: **Distribuovaná identifikace budov**

Pokyny pro vypracování:

1. Vytvořte nebo upravte převzatý simulační model chování budovy. Při návrhu dbejte na to, aby model co nejvíce odpovídal skutečnosti.
2. Nastudujte algoritmy síťového konsensu a distribuované optimalizace. Popište výhody a nevýhody nalezených algoritmů a vyberte vhodné algoritmy pro distribuovanou identifikace modelu budovy pomocí komunikujících zařízení měřících tepelný tok každého výměníku tepla.
3. Implementujte vybrané algoritmy distribuované identifikace a porovnejte jejich vlastnosti při použití na simulátoru budovy.
4. Modifikujte vámi vybraný algoritmus tak, aby byl vhodný pro běh na embedovaném zařízení s jádrem ARM Cortex M3; zejména s ohledem na paměťovou náročnost a omezenou rychlost komunikace. Možnými modifikacemi může být např. použití iterativního identifikace pro snížení paměťové náročnosti, využití iterací rozložených v čase pro snížení výpočetní náročnosti a multi-parametrické optimalizace pro snížení komunikační náročnosti.

Seznam odborné literatury:

[1] Boyd, S., & Vandenberghe, L. (2004). Convex optimization. Cambridge, UK: Cambridge University Press
[2] Boyd, S., & Gorinevsky, D. (2007). Distributed Estimation via Dual Decomposition. In European Control Conference (pp. 1511-1516)
[3] Terelius, H. (2010). Distributed Multi-Agent Optimization via Dual Decomposition. (Master thesis). KTH
[4] Wan, P., Lemmon, M.D. (2009). Event-triggered distributed optimization in sensor networks. In Information Processing in Sensor Networks (pp.49-60)

Vedoucí: Ing. Jiří Dostál

Platnost zadání: do konce letního semestru 2016/2017

L.S.

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 15. 10. 2015

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Ing. Jiří Dostál for the continuous support, his patience and immense knowledge. His guidance helped me in all the time of writing this thesis.

Besides my supervisor, my sincere thanks also goes to Ing. Jiří Řehoř for his priceless help in grey-box modelling.

Last but not the least, I also would like to thank my family and my girlfriend who supported me throughout writing this thesis and my life in general.

Declaration/Prohlášení

I hereby formally declare that I wrote the presented thesis on my own and cited all the used information sources in compliance with the Methodical instructions about the ethical principles for writing academic theses.

In Prague, May 27, 2016.

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.

V Praze dne 27. května 2016

.....

Abstract

This master thesis deals with building identification. At first, two different building thermal continuous-time LTI models are created using a technique of RC thermal circuits. These models are then used for simulations and data generation for identification process.

The main objective of this thesis is to propose and implement an algorithm for distributed building identification. The identification problem of the overall building is decomposed using so called decompositions methods, which results in solving small local identification problems with additional constraints on parameters of mutual walls. The local identification processes measured data (temperatures and heat flows of heat exchangers) and utilizes grey-box modelling and the Least Squares or Maximum Likelihood Estimation.

Finally, the proposed algorithm is tested on a 4-zone building benchmark model for various configurations.

Keywords: Building identification, Decomposition methods, Continuous-time Grey-box modelling, Linear systems.

Supervisor: Ing. Jiří Dostál

Abstrakt

Tato diplomová práce se zabývá identifikací budov. Nejprve jsou vytvořeny pomocí RC obvodů dva teplotní modely budovy jako lineární časově neproměnné systémy spojitého času. Tyto modely budou poté použity během simulací a také jako generátory dat pro identifikační proces.

Hlavním cílem této práce je navrhnout a implementovat algoritmus pro distribuovanou identifikaci budov. Identifikace budovy jako celku je rozložena pomocí takzvaných dekompozičních metod na menší lokální identifikační problémy v jednotlivých místnostech, kde si musí být rovny parametry, které popisují společné zdi mezi místnostmi. Lokální identifikace používá jako data naměřené teploty vzduchu v místnostech a tepelný tok výměníku. Identifikace je založena na grey-box modelování a odhaduje parametry pomocí nejmenších čtverců nebo maximálně věrohodného odhadu.

Nakonec je navržený algoritmus otestován na čtyřzónovém modelu budovy s několika různými nastaveními.

Klíčová slova: Identifikace budov, Dekompoziční metody, Grey-box modelování ve spojitém čase, Lineární systémy.

Překlad názvu: Distribuovaná identifikace budov

Contents

Figures

Tables

Chapter 1

Introduction

Heating or more generally HVAC (+ Ventilation and Air Conditioning) systems form an integral part of modern buildings. Their function is to ensure a good environment for occupants inside the buildings. As for the heating, this good environment is represented by a certain air temperature range in the building. Therefore, the temperature is permanently controlled to remain within the desired range.

However, an energy consumption when not intelligent HVAC systems are used can form up to 30 % of the overall building consumption [1]. The costs of heating may then amount to relatively high sums. Fortunately, modern HVAC systems can decrease the consumption and costs by 20-30 % [2], therefore they are being applied increasingly.

On the other hand, a correct (optimal) function of these systems is dependent on accurate dynamic building models which are often difficult to obtain [3]. The identification of the building is thus a crucial part for all the intelligent systems which utilize model-based methods. Nevertheless, this is associated with several challenging issues.

The first problem is the experimental data are often insufficient for the building identification. The second problem is that the buildings are modelled as large-scale MIMO[1] systems with a lot of parameters to be identified. Therefore, it is very demanding to estimate all the parameters correctly.

This thesis just deals with the building identification and brings an approach how to overcome the complexity of the identification problem. The approach utilizes decomposition methods to distribute the large identification problem to smaller identifications of individual rooms (zones). The local identification problem is based on grey-box modelling where the model structure is attained from physical principles and parameters are estimated.

1.1 Motivation of the thesis

This thesis was motivated by a need to obtain a proper thermal building model which will serve for advanced heating control methods, e.g. Model Predictive Control. The use of decomposition for the overall building identification problem arises from two aspects. For one thing, the decomposition brings a complexity reduction of the large-scale problem. For another, a nature of the problem is basically distributed. The data acquisition is performed by sensors measuring a heat flow of heat exchangers and

[1] Multiple Input Multiple Output

by wireless temperature sensors that are placed in every zone.

Therefore the idea is to carry out an identification of each zone separately using its own measured data. Furthermore, the parameters describing a mutual wall of two adjacent zones must be adjusted to attain a consensus. This will be ensured by a distributed identification algorithm which forms a main implementation part of this thesis.

1.2 Contribution of the thesis

At first, we give a brief introduction to a so called RC (building) modelling. Using these technique we create two thermal models of a building for simulation and identification purposes. One model (R2C1) achieves higher fidelity and is primarily intended for simulations and "experimental" data generation. The other model (R1C0) which is simpler serves mainly as a grey-box model template for the identification.

The main part of this thesis is devoted to implementation of the algorithm for a distributed building identification. The identification problem leads to an optimization problem which is thanks to duality transformed to the dual problem and then decomposed using a dual decomposition method. The identification problem is thus distributed into single zones whose agents communicates only with their adjacent agents.

1.3 Previous work

As for the building modelling, the technique of RC circuits is widely used. Zone air is always being associated with a certain capacity, a difference is only in the wall modelling. In the simpler case the walls are modelled with one resistance (conductance), this approach is utilized in [2, 4–6]. A wall can be also modelled having its own inner capacity and two resistances describing the heat flow transfers between air and the wall. This more complex case is considered in [3].

The identification of the building model is performed several ways (algorithms). The traditional ones uses a grey-box modelling:

- input-output - pbARMAX [2, 5], PEM [3, 4],
- state space description - subspace methods [1].

Another important part which will be exploited is convex optimization [7] and decomposition methods [8] which were for instance applied in [9] for distributed predictive control.

1.4 Outline of the thesis

Chapter 1 is an introductory part where motivation for thesis and its contribution are mentioned. Previous work along with used methods on a field of building identification is also mentioned. Chapter 2 is devoted to creation of a thermal building model. Chapter 3 is just a short introduction to mathematical optimization and duality. Chapter 4 presents decompositions methods and algorithms for optimization problems, both in a simple and general hypergraph form.

Chapter 5 derives and presents all the aspects needed to create the main identification algorithm. Chapter 6 presents results of the distributed identification applied on a benchmark 4-zone building. Identification was performed for various settings and the results are compared each other. The last chapter 7 summarizes achieved results and propose possible future development.

Remark 1.1. All computations and simulations were performed using MATLAB® & SIMULINK® R2015b.

Chapter 2

Thermal model of a building

The main objective of this thesis is to identify a thermal model of a building from a given data set. Thus the first thing we need is a good simulator of the building to provide the desired data for identification purposes.

As a building we consider only one floor which is created by several rooms neighbouring one another. Therefore only horizontal heat transfers among rooms and with outside environment are considered.

An elementary thermodynamics of the room is modelled by using a technique of thermal circuits. They consist of thermal sources, conductances and capacitances which model the reality in an acceptable way.

There are several physical laws which are important to derive a system of differential equations.

2.1 Physical foundations

The technique of thermal (RC) circuits is based on linear fundamental laws of thermodynamics.

2.1.1 Conductance

Firstly, it is Fourier's law also known as the law of heat conduction. It is expressed by

$$\mathbf{q} = -k\nabla T \tag{2.1}$$

where $\mathbf{q}$ is a local heat flux density $[\mathrm{W} \cdot \mathrm{m}^{-2}]$, k is a thermal conductivity (also known as K-value) $[\mathrm{W} \cdot \mathrm{m}^{-1} \cdot \mathrm{K}^{-1}]$ and ∇T is a temperature gradient $[\mathrm{K} \cdot \mathrm{m}^{-1}]$. The law (2.1) can be rewritten for a homogeneous material of one-dimensional geometry as

$$\frac{\Delta Q}{\Delta t} = -kA\frac{\Delta T}{\Delta x} \tag{2.2}$$

where $\frac{\Delta Q}{\Delta t}$ is a scalar heat flow [W], A is a cross-sectional surface area, ΔT is a temperature difference between the ends and Δx is a distance between the ends. If we denote

$$U = \frac{k}{\Delta x} = \frac{1}{R} \tag{2.3}$$

where U is the conductance (or U-value) [$\mathrm{W} \cdot \mathrm{m}^{-2} \cdot \mathrm{K}^{-1}$], R is the resistance and is reciprocal to conductance, however it is more common to work with conductances in thermodynamics modelling.

Now we can write Fourier's law as

$$\frac{\Delta Q}{\Delta t} = UA\,(-\Delta T) \tag{2.4}$$

and in infinitesimal limit we obtain

$$\dot{Q} = UA\,(-\Delta T) \tag{2.5}$$

which is useful for describing dynamical properties in thermal circuits. If the material consists of n layers with the same A and $\dot{Q}$ but different conductances U_i the total conductance U is the sum of their reciprocal values similarly to resistors in electrical domain

$$\frac{1}{U} = \frac{1}{U_1} + \cdots + \frac{1}{U_n}. \tag{2.6}$$

2.1.2 Heat capacity

The second important element in thermal circuits is a capacitance which is characterized by heat capacity (thermal capacity). The heat capacity is defined as a ratio of the amount of heat transferred to an object and the resulting temperature increase of the object

$$C = \frac{Q}{\Delta T}. \tag{2.7}$$

Heat capacity is assumed either at constant pressure (C_p) or constant volume (C_V). Usually we work with specific heat capacity which is related to unit mass of a material

$$c_p = \frac{C_p}{m}. \tag{2.8}$$

Finally, by differentiating (2.7) with respect to time we obtain a well-known equation of object's temperature evolvement

$$\dot{T} = \frac{1}{C}\dot{Q}. \tag{2.9}$$

Remark 2.1. The heat flow $\dot{Q}$ is usually denoted by other symbols, e.g. q is used in [2] and Φ is used in [4]. We will use a notation with an omitted dot above, therefore just Q.

2.2 Zone model (R2C1)

Now having relevant equations (2.5) and (2.9) for the conductance U or capacity C, we set a simple model of the zone-wall heat transfer and then a model of the whole zone.

2.2.1 Zone-wall model

The heat transfer trough a wall from one zone to another can be represented as a simple RC circuit. Since the wall is modelled by two conductances (resistances) and one capacity, the model is denoted with an abbreviation R2C1.

In the Fig. 2.1 we can see that the zone air is heated primarily by a heat flow Q_{hx} from a heat exchanger (radiator), which represents a controllable input, and secondarily the zone is heated (or cooled) by heat transfers through walls. The zone air has a capacity

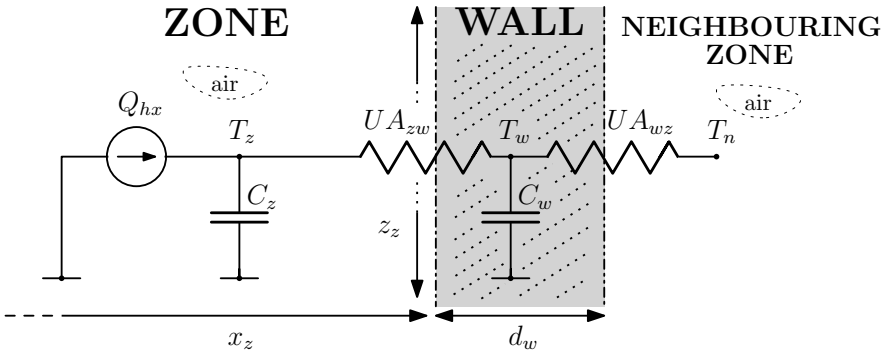


Figure 2.1: Zone-wall circuit

C_z and its temperature is T_z.

The heat flow propagates from the zone through a series of conductances UA_{zw}, partly is storied in the wall on account of a capacity C_w, hence influences the wall temperature T_w. Finally, a part of the heat flow propagates again through the series of conductances UA_{wz} and influences a temperature T_n in the neighbouring zone.

The direction of the heat flow is completely determined by a temperature difference between two particular nodes. To derive a system of ordinary differential equations describing a given thermal circuit, we use a rule similar to Kirchhoff's (junction) law

$$\sum_{i=1}^{n} Q_i = 0. \tag{2.10}$$

The rule (2.10) says that at any node (junction) in a thermal circuit, the algebraic sum of n heat flows flowing to the node is equal to zero. Therefore we can immediately write equations for the circuit 2.1

$$\begin{aligned} Q_{hx} &= C_z\dot{T}_z + UA_{zw}\left(T_z - T_w\right), \\ UA_{zw}\left(T_z - T_w\right) &= C_w\dot{T}_w + UA_{wz}\left(T_w - T_n\right). \end{aligned} \tag{2.11}$$

Rewriting equations (2.11) we obtain a state space model

$$\begin{aligned} \dot{T}_z &= -\frac{1}{C_z}UA_{zw}T_z + \frac{1}{C_z}UA_{zw}T_w + \frac{1}{C_z}Q_{hx}, \\ \dot{T}_w &= -\frac{1}{C_w}\left(UA_{zw} + UA_{wz}\right)T_w + \frac{1}{C_w}UA_{zw}T_z + \frac{1}{C_w}UA_{wz}T_n, \\ y &= T_z. \end{aligned} \tag{2.12}$$

2.2.2 Complete zone model

Now we approach to formation of the whole zone model. We assume having for instance a building that consists of four zones (rooms). The zone is bounded by four walls and its thermal circuit is depicted in the Fig. 2.2.

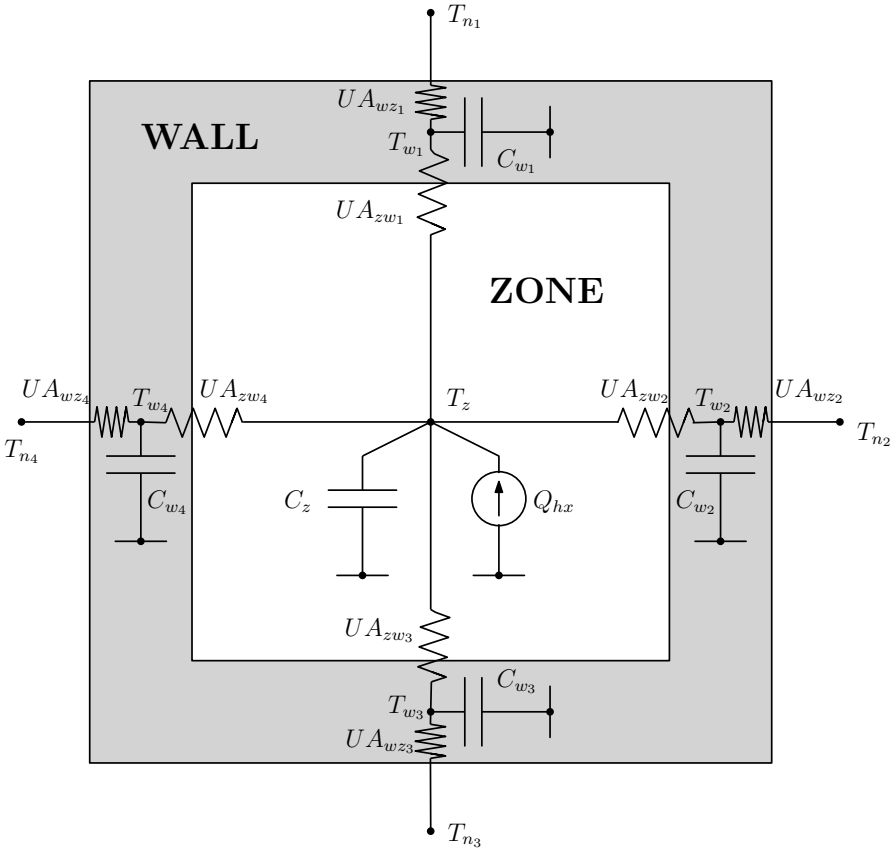


Figure 2.2: Thermal circuit of a zone

Based on this circuit we can write state space equations for the whole zone model

$$\begin{aligned}
\dot{T}_z &= -\frac{1}{C_z}\left(UA_{zw_1}+UA_{zw_2}+UA_{zw_3}+UA_{zw_4}\right)T_z \\
&\quad +\frac{1}{C_z}\left(UA_{zw_1}T_{w_1}+UA_{zw_2}T_{w_2}+UA_{zw_3}T_{w_3}+UA_{zw_4}T_{w_4}\right)+\frac{1}{C_z}Q_{hx}, \\
\dot{T}_{w_i} &= -\frac{1}{C_{w_i}}\left(UA_{zw_i}+UA_{wz_i}\right)T_{w_i}+\frac{1}{C_{w_i}}UA_{zw_i}T_z+\frac{1}{C_{w_i}}UA_{wz_i}T_{n_i}, \quad i=1,\ldots,4, \\
y &= T_z.
\end{aligned} \tag{2.13}$$

The state space model (2.12) is a continuous-time LTI system, hence it can be expressed

in matrix form

$$\begin{aligned} \underbrace{\begin{bmatrix} \dot{T}_z \\ \dot{T}_{w_1} \\ \vdots \\ \dot{T}_{w_4} \end{bmatrix}}_{\dot{\mathbf{T}}} &= \mathbf{A}_c \underbrace{\begin{bmatrix} T_z \\ T_{w_1} \\ \vdots \\ T_{w_4} \end{bmatrix}}_{\mathbf{T}} + \mathbf{B}_c \underbrace{\begin{bmatrix} Q_{hx} \\ T_{n_1} \\ \vdots \\ T_{n_4} \end{bmatrix}}_{\mathbf{u}}, \\ \mathbf{y} &= \mathbf{C}_c\mathbf{T} + \mathbf{D}_c\mathbf{u}, \end{aligned} \tag{2.14}$$

where

$$\begin{aligned} \mathbf{A}_c &= \begin{bmatrix} -\frac{UA_{zw}}{C_z} & \frac{UA_{zw_1}}{C_z} & \dots & \frac{UA_{zw_4}}{C_z} \\ \frac{UA_{zw_1}}{C_{w_1}} & -\frac{UA_{zw_1}+UA_{wz_1}}{C_{w_1}} & \mathbf{0}^{\mathrm{T}} & 0 \\ \vdots & \mathbf{0} & \ddots & \vdots \\ \frac{UA_{zw_4}}{C_{w_4}} & 0 & \dots & -\frac{UA_{zw_4}+UA_{wz_4}}{C_{w_4}} \end{bmatrix}, \quad \mathbf{B}_c = \begin{bmatrix} \frac{1}{C_z} & 0 & \dots & 0 \\ 0 & \frac{UA_{wz_1}}{C_{w_1}} & \mathbf{0}^{\mathrm{T}} & 0 \\ \vdots & \mathbf{0} & \ddots & \vdots \\ 0 & 0 & \dots & \frac{UA_{wz_4}}{C_{w_4}} \end{bmatrix}, \\ \mathbf{C}_c &= \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}, \quad \mathbf{D}_c = \mathbf{0} \end{aligned} \tag{2.15}$$

and

$$UA_{zw} = \sum_{i=1}^{4} UA_{zw_i}. \tag{2.16}$$

2.3 Simplified zone model (R1C0)

Due to slow dynamics of the heat transfer through a wall we can represent the wall by just one conductance (see Fig. 2.3). The wall is then modelled by one conductance (resistance) and no capacity, therefore we denote the model with an abbreviation R1C0. A circuit of the whole zone is depicted in the Fig. 2.4.

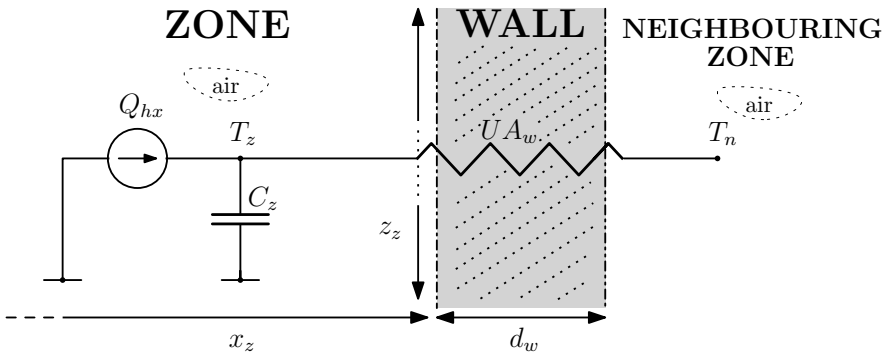


Figure 2.3: Zone-wall circuit (R1C0)

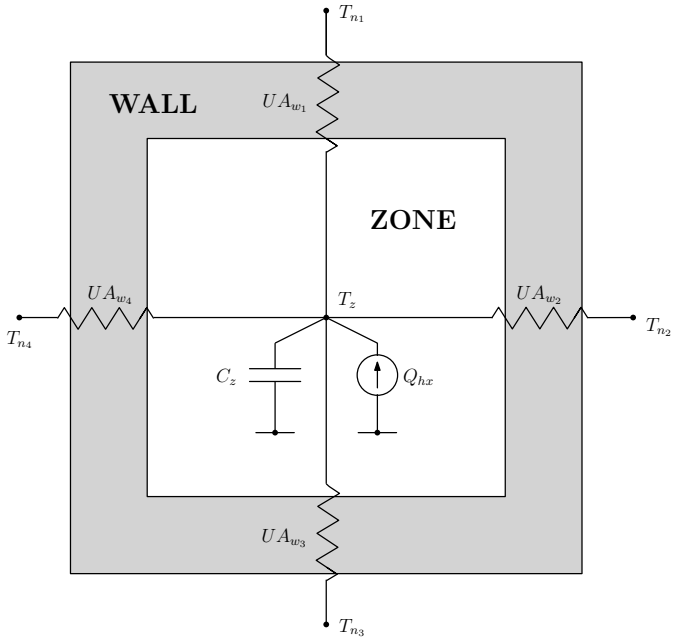


Figure 2.4: Simplified thermal circuit of a zone

A corresponding state space model is given by

$$\begin{aligned} \dot{\mathbf{T}}(t) &= \mathbf{A}_c\mathbf{T}(t) + \mathbf{B}_c\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}_c\mathbf{T}(t) + \mathbf{D}_c\mathbf{u}(t), \end{aligned} \tag{2.17}$$

where $\mathbf{A}_c \in \mathbb{R}$, $\mathbf{B}_c \in \mathbb{R}^{1\times 5}$, $\mathbf{C}_c \in \mathbb{R}$ and $\mathbf{D}_c \in \mathbb{R}^{1\times 5}$ are given by

$$\begin{aligned} \mathbf{A}_c &= -\frac{1}{C_z}\sum_{i=1}^{4} UA_{w_i}, \quad \mathbf{B}_c = \begin{bmatrix} \frac{1}{C_z} & \frac{UA_{w_1}}{C_z} & \dots & \frac{UA_{w_4}}{C_z} \end{bmatrix}, \\ \mathbf{C}_c &= 1, \quad \mathbf{D}_c = \mathbf{0}. \end{aligned} \tag{2.18}$$

The state vector $\mathbf{T}(t) \in \mathbb{R}$ is given by

$$\mathbf{T}(t) = T_z(t) \tag{2.19}$$

and $\mathbf{u}(t) \in \mathbb{R}^{5\times 1}$

$$\mathbf{u}(t) = \begin{bmatrix} Q_{hx} \\ T_{n_1} \\ \vdots \\ T_{n_4} \end{bmatrix}(t) \tag{2.20}$$

is an input vector where $T_{n_1}, \dots, T_{n_4}$ are temperatures behind the walls (i.e. temperatures in the neighbouring zone or outside the building).

2.4 Building model

Our example building consists of four zones (rooms) placed horizontally each other at one floor. A sketch of the building is in the Fig. 2.5. Each zone is modelled by either the

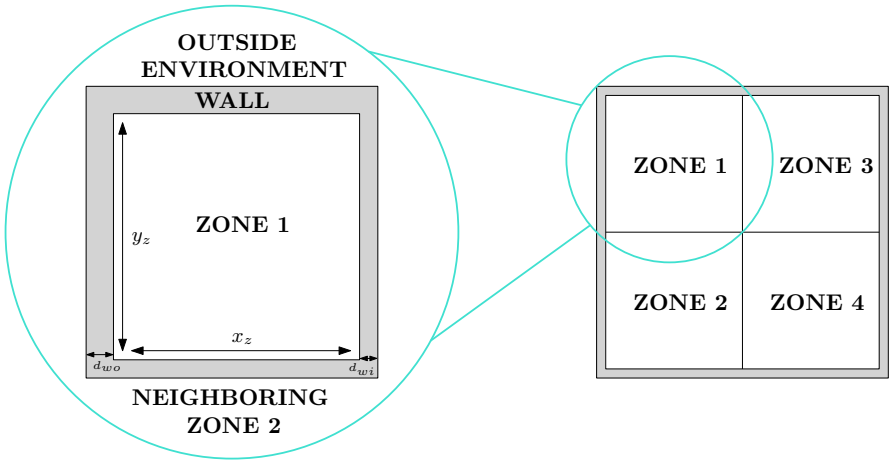


Figure 2.5: Simple 2x2 building

R2C1-model (2.14, 2.15) or the R1C0-model (2.17, 2.18). The whole building model is composed of single zone models which share wall models if they neighbour each other.

For simulation purposes there were chosen concrete constants for the building model (see Tab. 2.1). The SIMULINK scheme of the overall building model is in the Fig. 2.6.

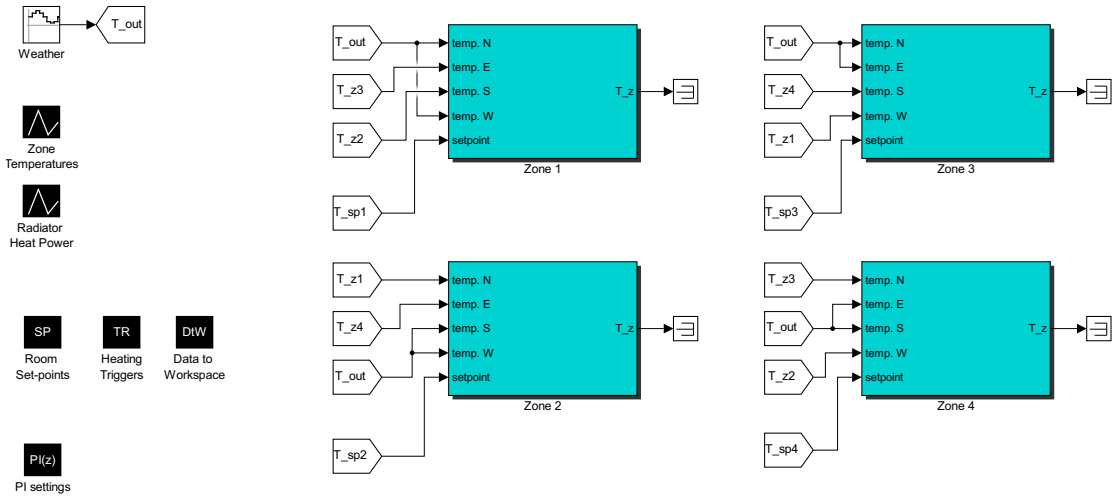


Figure 2.6: SIMULINK scheme of the 2x2 building model

description	symbol	value
zone x-dimension	x_z	4.8 m
zone y-dimension	y_z	4.8 m
zone z-dimension	z_z	2.6 m
zone air density	ρ_z	$1.29\ \mathrm{kg \cdot m^{-3}}$
zone air specific heat capacity	c_z	$1005\ \mathrm{J \cdot kg^{-1} \cdot K^{-1}}$
zone air thermal conductivity	k_z	$0.024\ \mathrm{W \cdot m^{-1} \cdot K^{-1}}$
outside wall width	d_{wo}	0.3 m
inside wall width	d_{wi}	0.15 m
brick density	ρ_w	$1922\ \mathrm{kg \cdot m^{-3}}$
brick specific heat capacity	c_w	$840\ \mathrm{J \cdot kg^{-1} \cdot K^{-1}}$
brick thermal conductivity	k_w	$0.8\ \mathrm{W \cdot m^{-1} \cdot K^{-1}}$

Table 2.1: Building constants

2.4.1 Discretization

The continuous-time systems (2.14) and (2.17) were discretized, since it is typical to observe and control them in discrete-time intervals. A sampling period T_s was chosen as

$$T_s = 60\ \mathrm{s}, \tag{2.21}$$

which is sufficient for thermal systems like these due to their slow dynamics.

We consider a sampled output $\mathbf{y}_k = \mathbf{y}(kT_s)$ for $k \in \mathbb{N}$ and simultaneously sampled input which is zero-order hold, i.e.

$$\mathbf{u}(t) = \mathbf{u}_k, \quad t \in \langle kT_s, (k+1)T_s \rangle . \tag{2.22}$$

A discretized state space system is then given by

$$\begin{aligned} \mathbf{T}_{k+1} &= \mathbf{A}(T_s)\mathbf{T}_k + \mathbf{B}(T_s)\mathbf{u}_k, \\ \mathbf{y}_k &= \mathbf{C}\mathbf{T}_k + \mathbf{D}\mathbf{u}_k. \end{aligned} \tag{2.23}$$

The quadruplet of discrete-time system matrices $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ is determined by

$$\mathbf{A}(T_s) = \exp\left(\mathbf{A}_c T_s\right), \tag{2.24a}$$

$$\mathbf{B}(T_s) = \int_0^{T_s} \exp\left(\mathbf{A}_c s\right) \mathrm{d}s \cdot \mathbf{B}_c, \tag{2.24b}$$

$$\mathbf{C} = \mathbf{C}_c, \tag{2.24c}$$

$$\mathbf{D} = \mathbf{D}_c. \tag{2.24d}$$

A computation of the matrix integral in (2.24b) can be demanding but according to [10] the matrices $\mathbf{A}$ and $\mathbf{B}$ could be computed simultaneously using an augmented matrix exponential as

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \exp\left(\begin{bmatrix} \mathbf{A}_c & \mathbf{B}_c \\ \mathbf{0} & \mathbf{0} \end{bmatrix} T_s \right). \tag{2.25}$$

2.5 Heating control

Each zone in the building is to be heated to a desired temperature T_z. Hence there was designed a simple feedback control for each zone to achieve that.

We assume following technical (saturation) limits for the manipulated input $\mathbf{u}(1) = Q_{hx}$ in each zone

$$Q_{hx} \in \langle 0, 2000 \rangle \text{ W}. \tag{2.26}$$

Based on the single zone model neglecting interactions among zones there was designed a discrete-time PI controller with a transfer function

$$P(z) = k_p + k_i \frac{T_s}{z-1}, \tag{2.27}$$

where

$$k_p = 300, \quad k_i = 0.3. \tag{2.28}$$

The PI controller (2.27) was applied with a proper anti-windup to recover from saturation. The general scheme of the system is in the Fig. 2.7 where disturbances $d(k)$ and measurement noise $v(k)$ are also depicted (their description will be given later at (6.10) and (6.12)). The azure block **G** denotes the discretized system (2.23). Signals $r(k)$ and $e(k)$ represent a reference (setpoint) temperature and error, respectively. The SIMULINK scheme of the zone model with the closed-loop heating is in the Fig. 2.8.

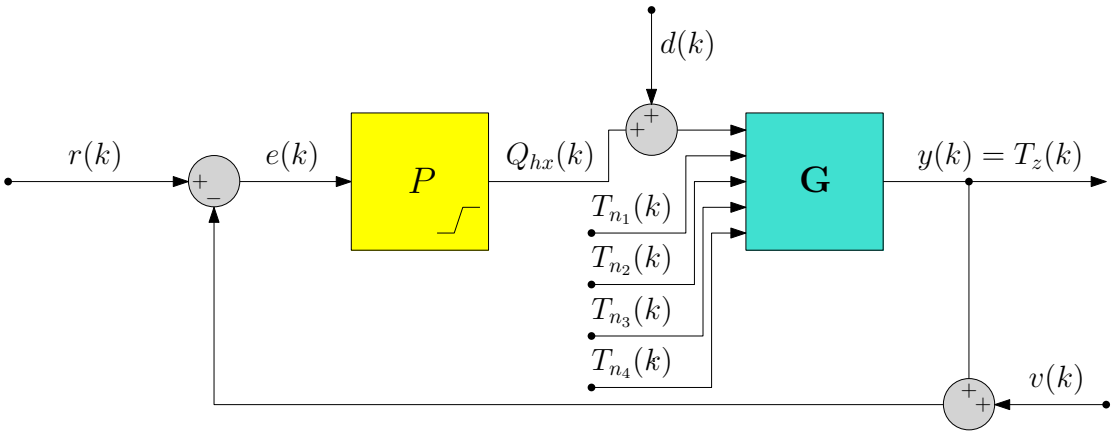


Figure 2.7: One zone closed-loop system with disturbances and measurement noise

Now we are able to perform a closed-loop simulation of for instance the R1C0 building. We simulate a one-day heating of the building. The reference tracking of zone temperatures is in the Fig. 2.9, where T_{z_i} is a zone air temperature, T_{sp_i} is a zone setpoint and T_{out} is a temperature of outside environment. Manipulated heat powers of heat exchangers are in the Fig. 2.10. A unit of a time axis is the sampling period T_s, therefore the time axis is in minutes.

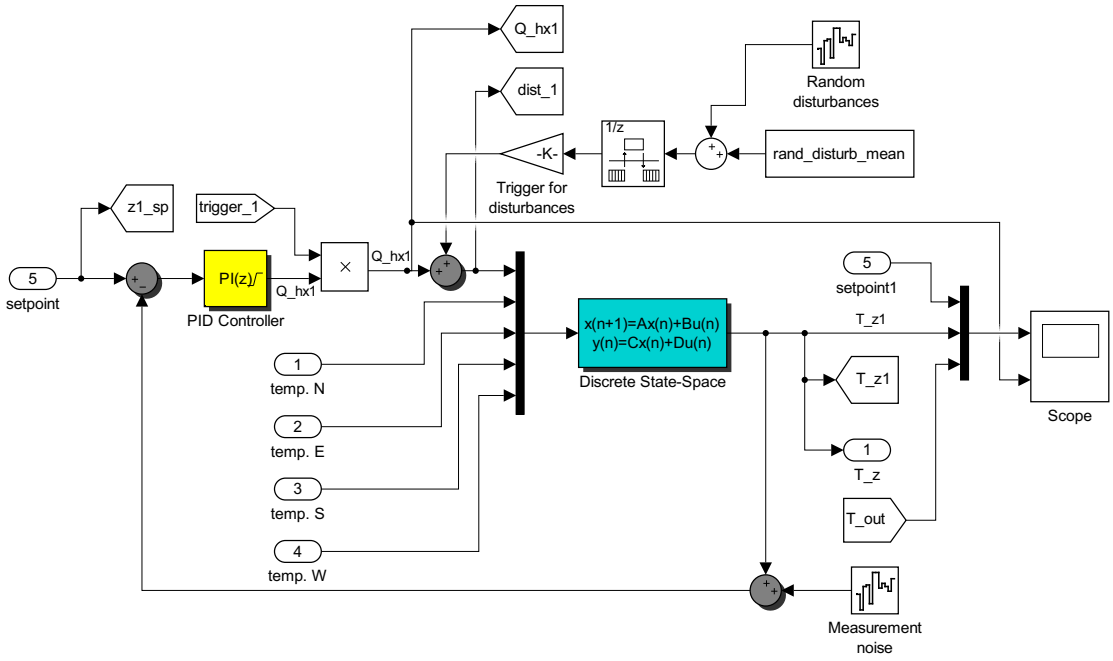


Figure 2.8: SIMULINK scheme of the zone model with closed-loop heating

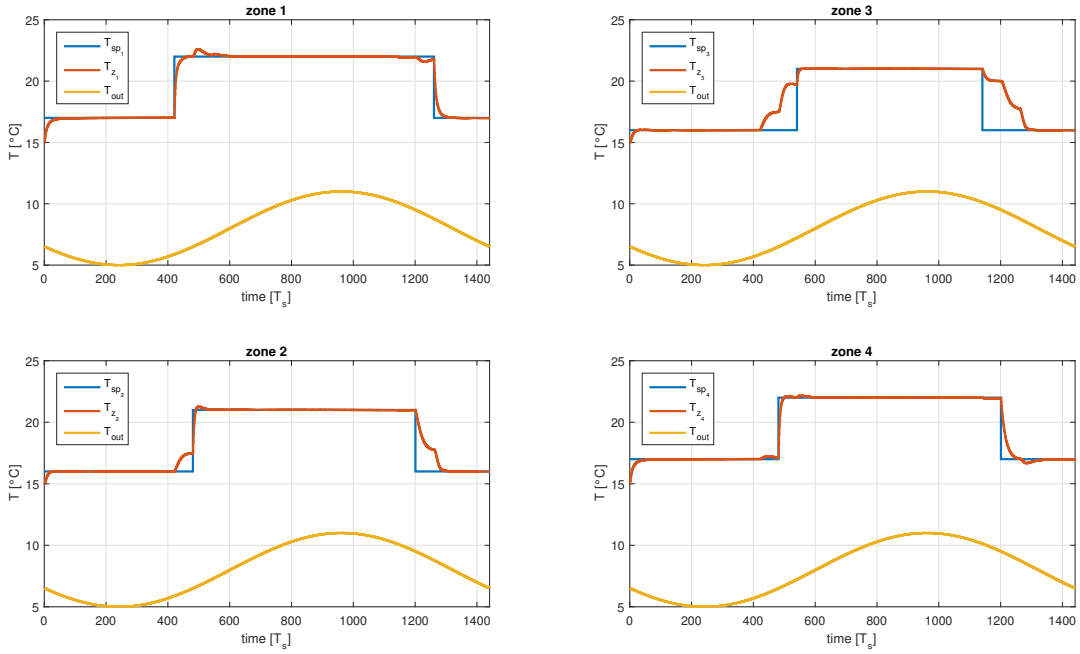


Figure 2.9: Reference tracking - air temperatures in building zones (R1C0)

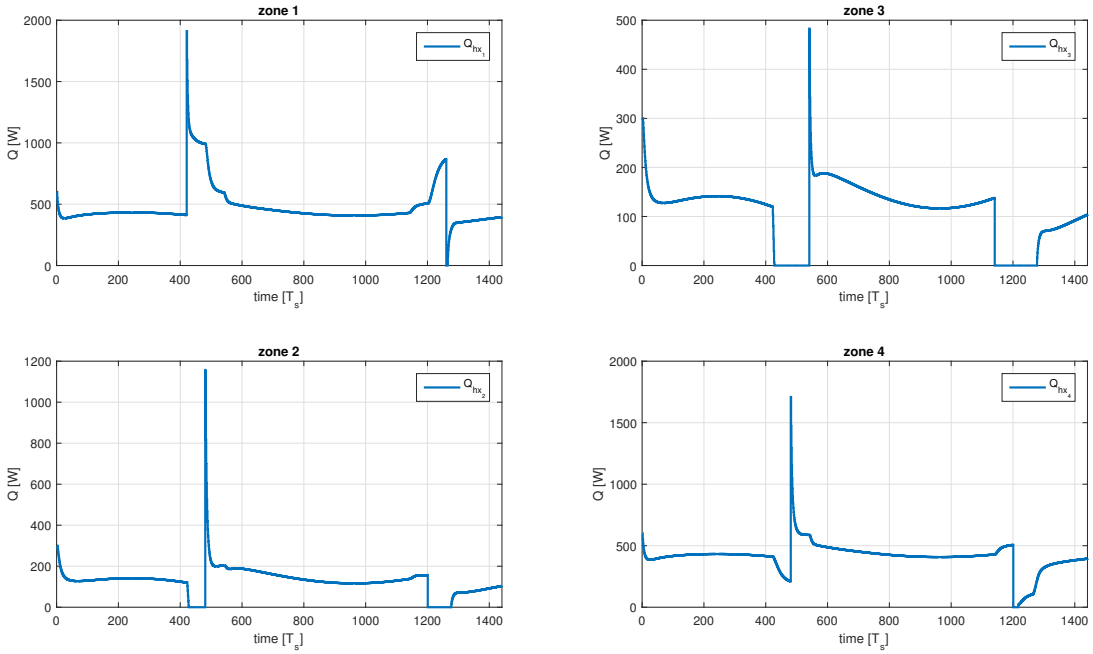


Figure 2.10: Manipulated heat powers in building zones (R1C0)

Chapter 3

Mathematical optimization

3.1 Introduction to optimization

Optimization is a mathematical discipline dealing with solving optimization problems. The solution of such problems has to be optimal among all feasible solutions. The standard form for an optimization problem can be formulated as in [7]

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & f_0(\mathbf{x}) \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, p. \end{aligned} \tag{3.1}$$

The vector $\mathbf{x} = \begin{bmatrix} x_1, \ldots, x_n \end{bmatrix}^{\mathrm{T}}$ is the optimization variable of the problem and the function $f_0 : \mathbb{R}^n \to \mathbb{R}$ is called the objective (or loss, cost) function whose value is to be minimized. The functions $f_i : \mathbb{R}^n \to \mathbb{R},\ i = 1, \ldots, m$, are the inequality constraint functions and $h_i : \mathbb{R}^n \to \mathbb{R},\ i = 1, \ldots, p$, are the equality constraint functions. The set of feasible points

$$\mathcal{D} = \{\mathbf{x} \,|\, f_i(\mathbf{x}) \leq 0,\ i = 1, \ldots, m;\ h_i(\mathbf{x}) = 0,\ i = 1, \ldots, p\} \tag{3.2}$$

is called the domain of the optimization problem.

A vector $\mathbf{x}^*$ is called optimal, a solution or minimizer of the problem (3.1), if it has the smallest objective value among all vectors satisfying the constraints. Thus for any $\mathbf{x}$ that satisfy the constraints, we have

$$f_0(\mathbf{x}) \geq f_0(\mathbf{x}^*). \tag{3.3}$$

The optimal value $f_0(\mathbf{x}^*)$ to the problem (3.1) is thus the maximum lower bound for the objective function f_0 in the domain $\mathcal{D}$

$$p^* = f_0(\mathbf{x}^*) = \inf \{f_0(\mathbf{x}) \,|\, \mathbf{x} \in \mathcal{D}\}. \tag{3.4}$$

3.2 Duality in optimization

Let's assume the domain $\mathcal{D}$ of the problem (3.1) to be nonempty. Then we can form a so called Lagrangian by adding weighted sums of constraints to the objective function

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i h_i(\mathbf{x}), \tag{3.5}$$

where λ_i is a Lagrange multiplier associated with the i-th $f_i(\mathbf{x}) \leq 0$ inequality constraint and similarly ν_i is a Lagrange multiplier associated with the i-th $h_i(\mathbf{x}) = 0$ equality constraint. The vectors $\boldsymbol{\lambda} \in \mathbb{R}^{m \times 1}$ and $\boldsymbol{\nu} \in \mathbb{R}^{p \times 1}$ are called dual variables associated with the problem (3.1).

The Lagrange dual function is defined as the minimal value of the Lagrangian (3.5) over all $\mathbf{x} \in \mathcal{D}$

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x} \in \mathcal{D}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x} \in \mathcal{D}} \left(f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^{p} \nu_i h_i(\mathbf{x}) \right). \tag{3.6}$$

An important property of the dual function g is its concavity, even when the problem (3.1) is not convex.

The dual function yields lower bounds on the optimal value p^* of the problem (3.1)

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \leq p^* \tag{3.7}$$

for any $\boldsymbol{\lambda} \succeq 0$ and any $\boldsymbol{\nu}$ (see [7]).

Now a natural question arises: What is the best lower bound on p^* obtained from the dual function g?

We can find this bound by solving another optimization problem

$$\begin{aligned} &\text{maximize} && g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \\ &\text{subject to} && \boldsymbol{\lambda} \succeq 0. \end{aligned} \tag{3.8}$$

This problem is called the Lagrange dual problem associated with the original problem (3.1) which is called in this context as the primal problem.

The Lagrange dual problem (3.8) is always a convex optimization problem, since the objective dual function is concave and the constraints are convex [7].

The optimal value of the dual problem

$$d^* = q(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) \tag{3.9}$$

is, by definition, the best lower bound on p^*. Hence, it brings an important inequality

$$d^* \leq p^* \tag{3.10}$$

which is called weak duality. The difference of the optimal values $p^* - d^*$ is called the optimal duality gap. If it reduces to zero, then

$$d^* = p^* \tag{3.11}$$

and so called strong duality holds. See [7] for details and conditions for the strong duality.

Chapter 4

Decomposition methods

In this chapter, we present an overview of decomposition methods used for solving optimization problems. The idea of these methods is that some mathematical problems can be at first decomposed to smaller (easier to solve) subproblems and then solved independently each other. Finally, the solution to the original problem is assembled from subproblems solutions.

There are several advantages of solving smaller subproblems than one big problem. Firstly, the subproblems can be solved in parallel which comprises time savings. Secondly, if a complexity of the original problem grows faster than linearly, then solving subproblems also mean significant computational savings. Another benefit of parallelization is the application in multi-agent systems, where the decomposition methods yields a distributed optimization algorithm [11].

4.1 Separable problems

A problem is called separable (or trivially parallelizable), if the subproblems are completely independent, thus they can be solved in parallel. The separable problem has a form of

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f_0(\mathbf{x}) = f_1(x_1) + \cdots + f_n(x_n) \\ \text{subject to} \quad & x_i \in \mathcal{C}_i, \quad i = 1, \ldots, n, \end{aligned} \tag{4.1}$$

where $\mathcal{C}_i$ is a constraint set for the i-th subproblem, presumably described by linear equalities and convex inequalities.

The problem (4.1) is separable since the subproblems

$$\begin{aligned} \underset{x_i}{\text{minimize}} \quad & f_i(x_i) \\ \text{subject to} \quad & x_i \in \mathcal{C}_i \end{aligned} \tag{4.2}$$

can be solved independently of each other. This is the simplest kind of problems that can be decomposed. A situation is more complicated when some variables are common to more subproblems or some constraints involve variables from more subproblems.

4.2 Complicating variable

The subproblems are often called subsystems. If there are interconnections among these subsystems, then the interconnections represent a variable which is common to those subsystems.

Consider a simple unconstrained problem

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad f_0(\mathbf{x}) = f_1(\mathbf{x}_{1\dots r}, \mathbf{y}) + f_2(\mathbf{x}_{r+1\dots n-s}, \mathbf{y}), \tag{4.3}$$

where the vector of optimization variables is given by

$$\mathbf{x} = \left[\mathbf{x}_{1\dots r}^{\mathrm{T}}, \mathbf{x}_{r+1\dots n-s}^{\mathrm{T}}, \mathbf{y}^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{4.4}$$

The variable $\mathbf{y} \in \mathbb{R}^{s\times 1}$ is called a complicating or coupling variable. From the view of single subsystems, the variables $\mathbf{x}_{1\dots n-s}$ are called private or local variables and $\mathbf{y}$ is called a public variable. A graphical representation of the system (4.3) is in the Fig. 4.1.

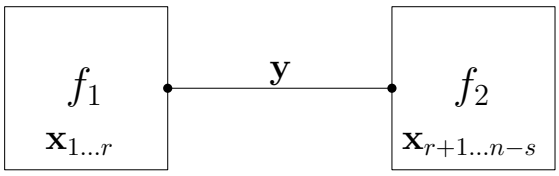


Figure 4.1: System with a complicating variable

4.3 Primal decomposition

The simplest way how to deal with a complicating variable $\mathbf{y}$ in (4.3) is by fixing its value and then solve subproblems which have become separable.

This method is called primal decomposition due to a manipulation with the primal variables. The solutions to separable subproblems are expressed as a function of public variables

$$\begin{aligned} \Phi_1(\mathbf{y}) &= \min_{\mathbf{x}_{1\dots r}} \quad f_1(\mathbf{x}_{1\dots r}, \mathbf{y}), \\ \Phi_2(\mathbf{y}) &= \min_{\mathbf{x}_{r+1\dots n-s}} \quad f_1(\mathbf{x}_{r+1\dots n-s}, \mathbf{y}). \end{aligned} \tag{4.5}$$

Defining

$$\Phi(\mathbf{y}) = \Phi_1(\mathbf{y}) + \Phi_2(\mathbf{y}), \tag{4.6}$$

we can express the solution to the original problem (4.3) with the complicating variable as

$$\min_{\mathbf{x}} f_0(\mathbf{x}) = \min_{\mathbf{y}} \left(\min_{\mathbf{x}_{1\dots n-s}} \Phi_1(\mathbf{y}) + \Phi_2(\mathbf{y}) \right) = \min_{\mathbf{y}} \Phi(\mathbf{y}). \tag{4.7}$$

This optimization problem is called a master or coordinator problem in primal decomposition. If the original problem is convex, then so is the master problem [9, 11]. Therefore,

the master problem can be solved by a gradient method (in case of differentiable functions) or subgradient method.

Let's denote the gradients (sensitivities to the complicating variable $\mathbf{y}$) of (4.5) as

$$\begin{aligned} \mathbf{g}_1(\mathbf{y}) &= \frac{\partial \Phi_1(\mathbf{y})}{\partial \mathbf{y}}, \\ \mathbf{g}_2(\mathbf{y}) &= \frac{\partial \Phi_2(\mathbf{y})}{\partial \mathbf{y}}, \\ \mathbf{g}(\mathbf{y}) &= \frac{\partial \Phi(\mathbf{y})}{\partial \mathbf{y}} = \mathbf{g}_1(\mathbf{y}) + \mathbf{g}_2(\mathbf{y}). \end{aligned} \tag{4.8}$$

The gradient (descent) method solves the master problem (4.7) by an iteration

$$\mathbf{y}_{k+1} = \mathbf{y}_k - \alpha_k \underbrace{(\mathbf{g}_1(\mathbf{y}_k) + \mathbf{g}_2(\mathbf{y}_k))}_{\mathbf{g}(\mathbf{y}_k)}, \tag{4.9}$$

where $\alpha_k > 0$ is an appropriate step size in descent.

Now, we are able to write an algorithm (see Alg. 1) for the primal decomposition method solving the problem (4.7). An illustration of the algorithm is in the Fig. 4.2.

Algorithm 1 Primal decomposition

Input: Initial estimate $\mathbf{y}_0$

Output: Estimate $\mathbf{y}_K$ at K-th iteration

1: **for** $k = 0$ **to** K **do**
   `% Solve subproblems and return their subgradients`
2: Solve $\Phi_1(\mathbf{y}_k)$ and return $\mathbf{g}_1(\mathbf{y}_k)$
3: Solve $\Phi_2(\mathbf{y}_k)$ and return $\mathbf{g}_2(\mathbf{y}_k)$
   `% Update the complicating variable`
4: $\mathbf{y}_{k+1} = \mathbf{y}_k - \alpha_k \left(\mathbf{g}_1(\mathbf{y}_k) + \mathbf{g}_2(\mathbf{y}_k)\right)$
5: **end for**

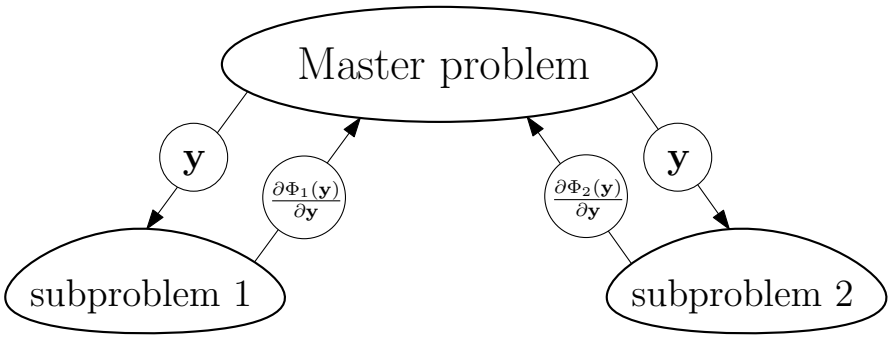


Figure 4.2: Primal decomposition algorithm

4.4 Dual decomposition

The second method how to decompose the problem (4.3) is based on duplicating the complicating variable $\mathbf{y}$. This approach utilizes the strong duality and converts the primal problem to the dual one using Lagrange multipliers.

By introducing local copies $\mathbf{y}_1$ and $\mathbf{y}_2$ of the complicating variable $\mathbf{y}$, we obtain a problem with an additional so called consistency condition

$$\begin{aligned} \underset{\mathbf{x}_{1\dots n-s},\mathbf{y}_1,\mathbf{y}_2}{\text{minimize}} \quad & f_0(\mathbf{x}_{1\dots n-s},\mathbf{y}_1,\mathbf{y}_2) = f_1(\mathbf{x}_{1\dots r},\mathbf{y}_1) + f_2(\mathbf{x}_{r+1\dots n-s},\mathbf{y}_2) \\ \text{subject to} \quad & \mathbf{y}_1 = \mathbf{y}_2. \end{aligned} \tag{4.10}$$

Now, let's form a Lagrangian of the problem (4.10)

$$L(\mathbf{x}_{1\dots n-s},\mathbf{y}_1,\mathbf{y}_2,\boldsymbol{\lambda}) = f_1(\mathbf{x}_{1\dots r},\mathbf{y}_1) + f_2(\mathbf{x}_{r+1\dots n-s},\mathbf{y}_2) + \boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{y}_1 - \mathbf{y}_2). \tag{4.11}$$

The dual problem is now separable and subproblems can be solved independently on each other

$$\begin{aligned} &\min_{\mathbf{x}_{1\dots r},\mathbf{y}_1} \quad f_1(\mathbf{x}_{1\dots r},\mathbf{y}_1) + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{y}_1, \\ &\min_{\mathbf{x}_{r+1\dots n-s},\mathbf{y}_2} \quad f_2(\mathbf{x}_{r+1\dots n-s},\mathbf{y}_2) - \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{y}_2. \end{aligned} \tag{4.12}$$

The optimal values of these subproblems are $q_1(\boldsymbol{\lambda})$ and $q_2(\boldsymbol{\lambda})$. The associated dual (master) problem is

$$\max_{\boldsymbol{\lambda}} \quad q(\boldsymbol{\lambda}) = q_1(\boldsymbol{\lambda}) + q_2(\boldsymbol{\lambda}). \tag{4.13}$$

This master problem can be solved with a gradient or subgradient method. The gradients of the subproblems are

$$\begin{aligned} \frac{\partial q_1(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} &= \mathbf{y}_1^*, \\ \frac{\partial q_2(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} &= -\mathbf{y}_2^*, \\ \frac{\partial q(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} &= \mathbf{y}_1^* - \mathbf{y}_2^*. \end{aligned} \tag{4.14}$$

Now, having corresponding gradients, we can write an iteration for the master problem (4.13) solution

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha_k(\mathbf{y}_1^* - \mathbf{y}_2^*). \tag{4.15}$$

Similarly to a primal decomposition case, we present an algorithm for the dual decomposition method (see Alg. 2). An illustration of the algorithm is in the Fig. 4.3.

Algorithm 2 Dual decomposition

Input: Initial estimate $\boldsymbol{\lambda}_0$

Output: Estimate $\boldsymbol{\lambda}_K$ at K-th iteration

1: **for** $k = 0$ **to** K **do**

`% Solve subproblems and return their subgradients`

2: Solve $q_1(\boldsymbol{\lambda}_k)$ and return $\mathbf{y}_1^*$

3: Solve $q_2(\boldsymbol{\lambda}_k)$ and return $-\mathbf{y}_2^*$

`% Update the dual variable`

4: $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha_k \left(\mathbf{y}_1^* - \mathbf{y}_2^*\right)$

5: **end for**

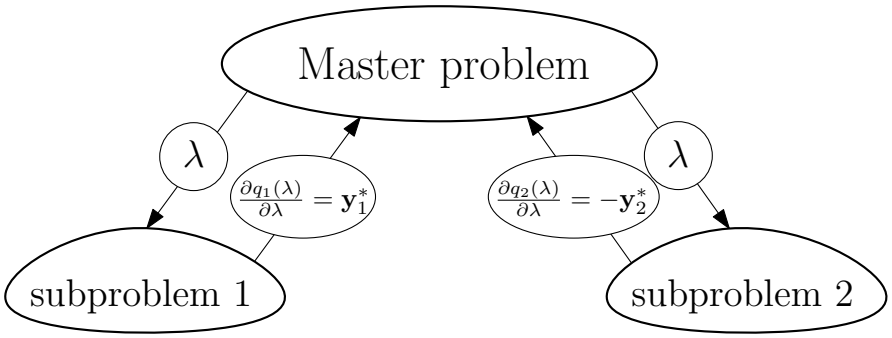


Figure 4.3: Dual decomposition algorithm

4.5 Hypergraph representation

The decomposition algorithms in previous sections assume there are only two subsystems (subproblems), though in general a situation can be far more complicated with a lot of subsystems and complicating variables.

At first, we introduce a notation which is based on the graph theory. A graph $G(V, E)$ consists of a set of nodes (vertices) V and a set of edges $E \subseteq V \times V$. A hypergraph $H(X, E_h)$ differs from a graph in the fact that a hyperedge from the set E_h can connect any number of vertices.

The structure of a problem to be decomposed can be represented generally by a hypergraph H. The nodes of this hypergraph represent the subsystems which have their own objectives, local variables and constraints. The hyperedges (or nets) of H are associated with complicating (public) variables.

Let's suppose, our problem is represented by an undirected graph $G(V, E)$. A number of subsystems is $N = |V|$ and a number of edges $n_z = |E|$ corresponds to the number of consistency constraints. We introduce a vector $\mathbf{z} \in \mathbb{R}^{n_z \times 1}$ which gives the common values of the public variables in each consistency constraint [8].

Now, we can express all consistency constraints in the matrix form as

$$\mathbf{y} = \mathbf{E}\mathbf{z}, \tag{4.16}$$

where $\mathbf{y} \in \mathbb{R}^{2n_z \times 1}$ is a vector of public variables from all subsystems and the matrix

$\mathbf{E} \in \mathbb{R}^{2n_z \times n_z}$ is defined as

$$\mathbf{E}_{ij} = \begin{cases} 1 & \mathbf{y}_i \text{ is in a constraint } j \\ 0 & \text{otherwise.} \end{cases} \tag{4.17}$$

The matrix $\mathbf{E}$ specifies the set of consistency constraints for the given subsystem interaction [8]. Using the hypergraph representation, the original optimization problem consisting of N subsystems can be expressed as

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N} f_i(\mathbf{x}_i, \mathbf{y}_i) \\ \text{subject to} \quad & \mathbf{y}_i = \mathbf{E}_i \mathbf{z}, \quad i = 1, \ldots, N, \end{aligned} \tag{4.18}$$

where $\mathbf{E}_i$ is a matrix built from rows of the matrix $\mathbf{E}$ which are associated with an i-th subsystem. Similarly, $\mathbf{x}_i$ are private and $\mathbf{y}_i$ are public variables associated to the i-th subsystem.

4.5.1 Primal decomposition

The primal decomposition method distributes an element of variable $\mathbf{z}$ to subsystems connected by the corresponding hyperedge. The distribution to an i-th subsystem is ensured by equalities $\mathbf{y}_i = \mathbf{E}_i \mathbf{z}$. Each subsystem then optimizes its local objective function

$$\Phi_i(\mathbf{y}_i) = \min_{\mathbf{x}_i} \; f_i(\mathbf{x}_i, \mathbf{y}_i) \tag{4.19}$$

and sends a computed gradient $\frac{\partial \Phi_i(\mathbf{y}_i)}{\partial \mathbf{y}_i}$ to the master. The global problem

$$\min_{\mathbf{z}} \; \Phi(\mathbf{z}) = \sum_{i=1}^{N} \Phi_i(\mathbf{y}_i) \tag{4.20}$$

is solved by the master which sums up received gradients and update public variables. The whole process is described by an Algorithm 3.

4.5.2 Dual decomposition

Firstly, we have to find a dual problem to the original primal problem hypergraphly represented (4.18). The Lagrangian is given by

$$\begin{aligned} L(\mathbf{x}_{1,\ldots,N}, \mathbf{y}, \mathbf{z}, \boldsymbol{\lambda}) &= \sum_{i=1}^{N} f_i(\mathbf{x}_i, \mathbf{y}_i) + \boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{y} - \mathbf{E}\mathbf{z}) \\ &= \sum_{i=1}^{N} \left(f_i(\mathbf{x}_i, \mathbf{y}_i) + \boldsymbol{\lambda}_i^{\mathrm{T}} \mathbf{y}_i \right) - \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{E}\mathbf{z}. \end{aligned} \tag{4.21}$$

Algorithm 3 Primal decomposition - hypergraph form

Input: Initial estimate of net variables $\mathbf{z}^0$
Output: Estimate $\mathbf{z}^K$ at K-th iteration

1: **for** $k = 0$ **to** K **do**
2: **for** $i = 1$ **to** N **do**
`% Distribute net variables to subproblems`
3: $\mathbf{y}_i^k = \mathbf{E}_i \mathbf{z}^k$
`% Solve a subproblem and return its subgradient`
4: Solve $\Phi_i(\mathbf{y}_i^k)$ and return $g_i(\mathbf{y}_i^k)$
5: **end for**
`% Calculate the subgradient of the master problem`
6: $g(\mathbf{y}^k) = \sum_{i=1}^{N} \mathbf{E}_i^{\mathrm{T}} g_i(\mathbf{y}_i^k)$
`% Update the vector of net variables`
7: $\mathbf{z}^{k+1} = \mathbf{z}^k - \alpha^k g(\mathbf{y}^k)$
8: **end for**

A Lagrange dual function to the primal problem is obtained by minimizing the Lagrangian (4.21)

$$q(\boldsymbol{\lambda}) = \min_{\mathbf{x}_{1,\ldots,N},\mathbf{y},\mathbf{z}} \sum_{i=1}^{N} \left(f_i(\mathbf{x}_i, \mathbf{y}_i) + \boldsymbol{\lambda}_i^{\mathrm{T}} \mathbf{y}_i \right) - \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{E} \mathbf{z}. \tag{4.22}$$

The dual function becomes separable if it is first minimized over $\mathbf{z}$. This results in the condition $\mathbf{E}^{\mathrm{T}}\boldsymbol{\lambda} = 0$, which states that the sum of Lagrange multipliers over each net is zero. The subproblems can be now solved separately

$$q_i(\boldsymbol{\lambda}_i) = \min_{\mathbf{x}_i,\mathbf{y}_i} \; f_i(\mathbf{x}_i, \mathbf{y}_i) + \boldsymbol{\lambda}_i^{\mathrm{T}} \mathbf{y}_i, \quad i = 1, \ldots, N, \tag{4.23}$$

and the dual master problem has a form

$$\begin{aligned} \underset{\boldsymbol{\lambda}}{\text{maximize}} \quad & q(\boldsymbol{\lambda}) = \sum_{i=1}^{N} q_i(\boldsymbol{\lambda}_i) \\ \text{subject to} \quad & \mathbf{E}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{0}. \end{aligned} \tag{4.24}$$

Since there is the constraint in this master problem (4.24), the solution can be reached via a so called projected subgradient method. The projection operator P onto feasible set $\mathbf{E}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{0}$ is affine and gives an iteration

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \alpha^k \underbrace{\left(\mathbf{I} - \mathbf{E} \left(\mathbf{E}^{\mathrm{T}} \mathbf{E} \right)^{-1} \mathbf{E}^{\mathrm{T}} \right)}_{P} \mathbf{y}^{k*}. \tag{4.25}$$

This iteration update can be interpreted as follows - at first, the average values of local copies of public variables are computed

$$\hat{\mathbf{z}}^k = \left(\mathbf{E}^{\mathrm{T}} \mathbf{E} \right)^{-1} \mathbf{E}^{\mathrm{T}} \mathbf{y}^{k*} \tag{4.26}$$

and then are subtracted from corresponding public variables

$$\mathbf{g}^k = \mathbf{y}^{k*} - \mathbf{E}\hat{\mathbf{z}} \tag{4.27}$$

to form a projected subgradient $\mathbf{g}^k$ [9, 11]. The idea of the dual decomposition is such that each subproblem has its own copy of the public variables and corresponding Lagrange multipliers (prices). After the subsystems find their optima, the public variables are compared and dual variables (prices) are updated.

The master problem makes an effort to reach the consistency between local copies of the public variables. The procedure is described by an Algorithm 4.

Algorithm 4 Dual decomposition - hypergraph form

Input: Initial dual variable vector $\boldsymbol{\lambda}^0$ satisfying $\mathbf{E}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{0}$ (e.g. $\boldsymbol{\lambda} = \mathbf{0}$)
Output: Estimate $\boldsymbol{\lambda}^K$ at K-th iteration (and primal variables $\mathbf{y}^*$)

1: **for** $k = 0$ **to** K **do**
2: **for** $i = 1$ **to** N **do**
`% Distribute multipliers (prices) to subproblems`
3: $\boldsymbol{\lambda}^k \rightarrow \boldsymbol{\lambda}_i^k$
`% Solve a subproblem and return its subgradient`
4: Solve $q_i(\boldsymbol{\lambda}_i^k)$ and return $\mathbf{y}_i^{k*}$
5: **end for**
`% Compute average of public variables over each net`
6: $\hat{\mathbf{z}}^k = \left(\mathbf{E}^{\mathrm{T}}\mathbf{E}\right)^{-1}\mathbf{E}^{\mathrm{T}}\mathbf{y}^{k*}$
`% Update the dual variables (prices)`
7: $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \alpha^k\left(\mathbf{y}^{k*} - \mathbf{E}\hat{\mathbf{z}}^k\right)$
`% Compute the norm of consistency constraint residual`
8: $r^k = \left\|\mathbf{y}^{k*} - \mathbf{E}\hat{\mathbf{z}}^k\right\|_2$
9: **if** $r^k < r_{\mathrm{THRES}}$ **then**
10: **break**
11: **end if**
12: **end for**

The norm of the projected subgradient computed during iterations in the Alg. 4

$$r^k = \left\|\mathbf{g}^k\right\|_2 = \left\|\mathbf{y}^{k*} - \mathbf{E}\hat{\mathbf{z}}^k\right\|_2 \tag{4.28}$$

measures a consistency between corresponding local copies of public variables. It is called the consistency constraint residual and is used as a stopping criterion (r_{THRES}) for the outer loop in the Alg. 4.

Chapter 5

Distributed identification

This chapter constitutes a main implementation part of this thesis. The theory presented in previous chapters will be now applied to identify a benchmark 4-zone building in the distributed manner.

5.1 Introductory problem description

The goal is to identify a building model in order to be used subsequently e.g. for an advanced heating control. A such advanced control method is for example MPC[1] or its distributed counterpart dMPC. However, these methods are dependent on a precise model to function properly.

The input (experimental) data for identification are these quantities

- heat flows of heat exchangers in each zone - $Q^i_{hx},\ i = 1, \ldots, N$,
- air temperatures of each zone - $T^i_z,\ i = 1, \ldots, N$,
- outside (ambient) temperature - T_{out}.

The data for the i-th zone are arranged as follows

$$\begin{aligned} \mathbf{u}^i_k &= \begin{bmatrix} Q^i_{hx}(k) & T^i_{n_1}(k) & \ldots & T^i_{n_4}(k) \end{bmatrix}^{\mathrm{T}}, \quad y^i_k = T^i_z(k), \\ \mathcal{D}^i_K &= \left\{ \mathbf{u}^i_1, \ldots, \mathbf{u}^i_K, y^i_1, \ldots, y^i_K \right\}, \end{aligned} \tag{5.1}$$

where $T^i_{n_j},\ j = 1, \ldots, 4$ are temperatures behind the i-th zone's walls, therefore either a temperature in the neighbouring zone or outside temperature.

The identification of a building model as a whole is a straightforward approach and can be powerful for small buildings. Although, the complexity of the identification problem grows faster than linearly for an increasing number of zones (rooms).

Thus, we approach to this problem by decomposing it into single zone identifications whilst attempting to reach a consensus between overlapping parts of single zones (i.e. mutual walls).

There are three main identification approaches according to usage of physical insight

[1] Model Predictive Control

1. White-box modelling
   - The model of a system is based completely on physical principles which determine both the model structure and parameter values.
2. Grey-box modelling
   - A semi-physical/statistical approach where the model structure is given by physical principles but some parameters may not be known.
3. Black-box modelling
   - A purely statistical approach where neither the model structure nor the parameters are known.

Since, we have knowledge of the model structure at our disposal (see (2.14) or (2.17)) and numeric values of parameters are to be determined from statistical processing of the measured (simulated) data, we utilize the grey-box modelling.

5.2 Grey-box modelling

A structure of our grey-box model (abbreviated GBM) is given by single zone models R2C1 (2.14) or R1C0 (2.17). Hence, we will not state a general form of GBM (can be found in [12]) but a slightly modified form which is completely convenient for our case.

Let's write a parametrised continuous-time LTI[2] state space model

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}_c(\boldsymbol{\theta})\mathbf{x}(t) + \mathbf{B}_c(\boldsymbol{\theta})\mathbf{u}(t), \\ y(t) &= \mathbf{C}_c\mathbf{x}(t), \end{aligned} \tag{5.2}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is a state vector, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is an input vector and $y(t) \in \mathbb{R}$ is an output vector. The matrices $\mathbf{A}_c(\boldsymbol{\theta}) \in \mathbb{R}^{n\times n}$ and $\mathbf{B}_c(\boldsymbol{\theta}) \in \mathbb{R}^{n\times n_u}$ are parametrised whereas $\mathbf{C}_c \in \mathbb{R}^{1\times n}$ is given a priori.

Finally, $\boldsymbol{\theta} \in \mathbb{R}^{n_p}$ denotes a vector of parameters which arise from physical principles. The parameter vector $\boldsymbol{\theta}$ is formed as

- R2C1 (according to the Fig. 2.2)

$$\boldsymbol{\theta} = \begin{bmatrix} C_z & UA_{zw_1} & \dots & UA_{zw_4} & C_{w_1} & \dots & C_{w_4} & UA_{wz_1} & \dots & UA_{wz_4} \end{bmatrix}^{\mathrm{T}}, \tag{5.3}$$

- R1C0 (according to the Fig. 2.4)

$$\boldsymbol{\theta} = \begin{bmatrix} C_z & UA_{w_1} & \dots & UA_{w_4} \end{bmatrix}^{\mathrm{T}}. \tag{5.4}$$

There is also a reasonable physical constraint for all parameters

$$\boldsymbol{\theta} \succ \mathbf{0}, \tag{5.5}$$

i.e. all parameters must be positive.

[2] Linear Time Invariant

5.2.1 Discrete-time GBM

Since the data used for identification are discrete-time measured sequences, there is a need to transform a continuous-time GBM to its discrete-time counterpart. Considering the assumption (2.22) and T_s being a sampling period, we obtain

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}(\boldsymbol{\theta}, T_s)\mathbf{x}_k + \mathbf{B}(\boldsymbol{\theta}, T_s)\mathbf{u}_k, \\ y_k &= \mathbf{C}\mathbf{x}_k. \end{aligned} \tag{5.6}$$

Matrices $\mathbf{A}$ and $\mathbf{B}$ can be computed as (2.25) and $\mathbf{C} = \mathbf{C}_c$.

5.2.2 Output error GBM

Both the continuous (5.2) and discrete-time GBM (5.6) are deterministic models. However, identification data are always affected by additive noise.

Let's extend our discrete-time GBM (5.6) with a simplified stochastic part representing a measurement noise at the output

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}(\boldsymbol{\theta}, T_s)\mathbf{x}_k + \mathbf{B}(\boldsymbol{\theta}, T_s)\mathbf{u}_k, \\ y_k &= \mathbf{C}\mathbf{x}_k + e_k. \end{aligned} \tag{5.7}$$

The measurement noise $e_k \in \mathbb{R}$ is a Gaussian white noise described by

$$e_k \sim \mathcal{N}(0, R), \quad \operatorname{cov}\{e_k, e_{k+i}\} = 0, \ i \neq 0, \tag{5.8}$$

where $R > 0$ denotes a variance of e_k.

5.3 Parameter estimation

Now, having the parametrised GBM, we want to estimate optimal values of parameters using the data (5.1). There exist several estimation methods and the most common are Least Square Estimation (LSE) and Maximum Likelihood Estimation (MLE).

5.3.1 Least square estimation

The LSE method is based on a natural endeavour to minimize the sum of squared errors e_k^2 in (5.7). The optimization problem can be expressed as

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \ V(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\mathcal{E}}(\boldsymbol{\theta})^{\mathrm{T}}\boldsymbol{\mathcal{E}}(\boldsymbol{\theta}), \tag{5.9}$$

where $\boldsymbol{\mathcal{E}}(\boldsymbol{\theta}) \in \mathbb{R}^{K\times 1}$ is an error vector

$$\boldsymbol{\mathcal{E}}(\boldsymbol{\theta}) = \mathbf{Y} - \hat{\mathbf{Y}}(\boldsymbol{\theta}) \tag{5.10}$$

formed from measured outputs

$$\mathbf{Y} = \begin{bmatrix} y_1 & \dots & y_K \end{bmatrix}^{\mathrm{T}} \tag{5.11}$$

and their predictions

$$\hat{\mathbf{Y}}(\boldsymbol{\theta}) = \begin{bmatrix} \hat{y}_1(\boldsymbol{\theta}) & \dots & \hat{y}_K(\boldsymbol{\theta}) \end{bmatrix}^{\mathrm{T}}. \tag{5.12}$$

The predicted outputs are calculated using the discrete-time GBM (5.6) for the given $\boldsymbol{\theta}$

$$\begin{aligned} \hat{\mathbf{x}}_{k+1}(\boldsymbol{\theta}) &= \mathbf{A}(\boldsymbol{\theta})\hat{\mathbf{x}}_k(\boldsymbol{\theta}) + \mathbf{B}(\boldsymbol{\theta})\mathbf{u}_k, \\ \hat{y}_k(\boldsymbol{\theta}) &= \mathbf{C}\hat{\mathbf{x}}_k(\boldsymbol{\theta}). \end{aligned} \tag{5.13}$$

5.3.2 Maximum likelihood estimation

The MLE method searches for a maximizer $\boldsymbol{\theta}^*$ of the likelihood function

$$\mathcal{L}(\boldsymbol{\theta}|\mathcal{D}^K) = p(\mathbf{Y}|\boldsymbol{\theta}, \mathbf{U}) = \prod_{k=1}^{K} p(y_k|\mathcal{D}_{k-1}, \mathbf{u}_k, \boldsymbol{\theta}). \tag{5.14}$$

For our case, where $y_k \in \mathbb{R}$, the Gaussian probability density function reduces to

$$p(y_k|\mathcal{D}_{k-1}, \mathbf{u}_k, \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi R}} \exp\left(-\frac{\epsilon_k^2}{2R}\right). \tag{5.15}$$

Hence according to [12], the maximization of (5.14) is equivalent to minimizing the logarithm of an estimated variance $\hat{R}$

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \; V(\boldsymbol{\theta}) = \frac{1}{2} \log \underbrace{\left(\frac{1}{N}\boldsymbol{\mathcal{E}}(\boldsymbol{\theta})^{\mathrm{T}}\boldsymbol{\mathcal{E}}(\boldsymbol{\theta})\right)}_{\hat{R}}. \tag{5.16}$$

Both the LSE (5.9) and MLE (5.16) problems can be solved by an iterative descent gradient method.

5.4 Formulation of the identification problem

We formulate the building identification as an optimization problem and use earlier presented dual decomposition method (see 4.5.2) to distribute the problem into local identifications of each zone with a supervising master (global) problem that ensures consistency between the parameters of shared model parts (i.e. wall parameters).

At first, we derive some statements for a simpler 2-zone R1C0 building (see Fig. 5.1), since the problem is more transparent. The identification problem of this building can be formulated as

$$\begin{aligned} &\underset{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2}{\text{minimize}} \; V_1(\boldsymbol{\theta}_1) + V_2(\boldsymbol{\theta}_2) \\ &\text{subject to } \boldsymbol{\theta}_i \succ \mathbf{0}, \quad i = 1, 2, \end{aligned} \tag{5.17}$$

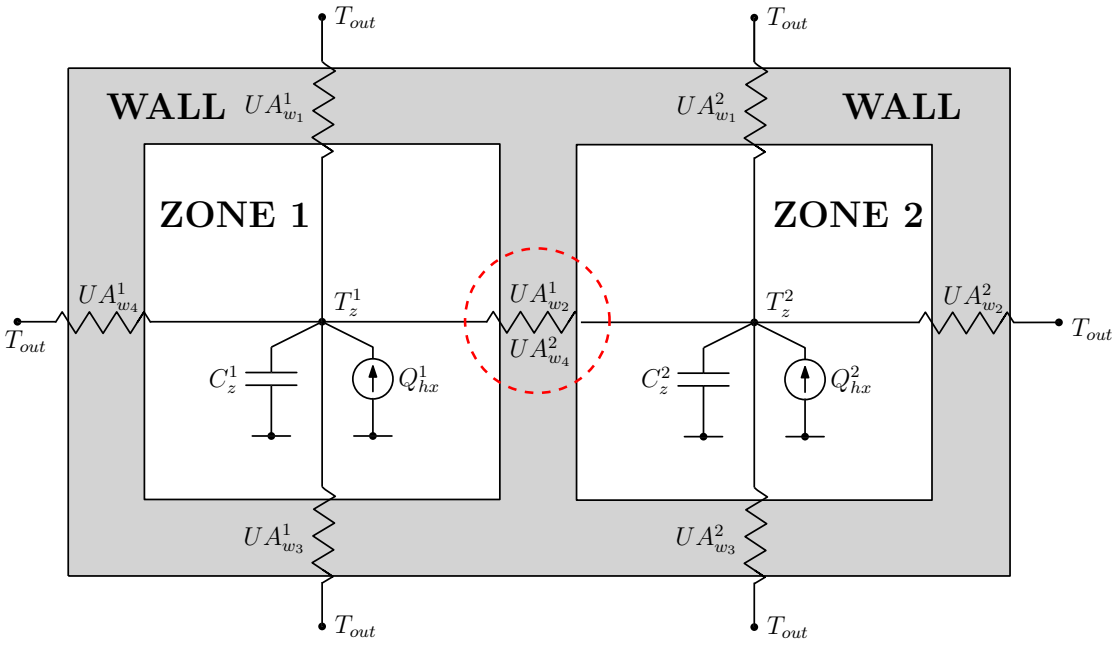


Figure 5.1: A two-zone R1C0 model with a red circled complicating variable

where $V(\cdot)$ is an objective function of either LSE (5.9) or MLE (5.16). The parameter vector $\boldsymbol{\theta}_i$ is defined as at (5.4). It is obvious the problem (5.17) cannot be trivially separated, since there is a complicating variable (red written) in $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$

$$\begin{aligned} \boldsymbol{\theta}_1 &= \begin{bmatrix} C^1_z & UA^1_{w_1} & UA^1_{w_2} & UA^1_{w_3} & UA^1_{w_4} \end{bmatrix}^{\mathrm{T}}, \\ \boldsymbol{\theta}_2 &= \begin{bmatrix} C^2_z & UA^2_{w_1} & UA^2_{w_2} & UA^2_{w_3} & UA^2_{w_4} \end{bmatrix}^{\mathrm{T}}. \end{aligned} \tag{5.18}$$

The complicating variable (a conductance of the mutual wall) is marked with a red dashed circle in the Fig. 5.1. Hence, the optimized parameter values $\boldsymbol{\theta}^*_1(3)$ and $\boldsymbol{\theta}^*_2(5)$ must not be different, but need to satisfy

$$\boldsymbol{\theta}^*_1(3) = \boldsymbol{\theta}^*_2(5) \quad \text{or} \quad UA^{1*}_{w_2} = UA^{2*}_{w_4}. \tag{5.19}$$

In a general case, when a building consists of more than two zones (e.g. our 4-zone model), the situation gets more complicated and we need to distinguish public variables from private ones in the parameter vector $\boldsymbol{\theta}$.

Therefore, we introduce a vector of public variables $\boldsymbol{\theta}_p \in \mathbb{R}^{p\times 1}$, $p < n_p$ and define a matrix $\mathbf{S} \in \mathbb{R}^{p\times n_p}$ which selects public variables from the parameter vector

$$\boldsymbol{\theta}_p = \mathbf{S}\boldsymbol{\theta}. \tag{5.20}$$

5.4.1 Graph representation utilization

Before we formulate the global identification problem, we investigate its crucial part, namely the equality $\mathbf{y} = \mathbf{E}\mathbf{z}$ stated at (4.16).

Let's focus for a while on determination and a meaning of the matrix $\mathbf{E}$ and vector $\mathbf{z}$. They have already been formally defined at (4.16) and (4.17) but we want them to be computed algorithmically. This can be truly reached, since we anticipate their structure is somehow associated with a hypergraph representation of our building identification problem.

Again, consider the benchmark 4-zone building. The zones are represented by nodes and public variables by edges (see Fig. 5.2). The vector $\mathbf{z}$ is formed as

$$\mathbf{z} = \begin{bmatrix} z_1 & \dots & z_{n_z} \end{bmatrix}^{\mathrm{T}}, \tag{5.21}$$

where $n_z = |E|$ is the number of edges in the graph $G(V, E)$, i.e. number of public variables.

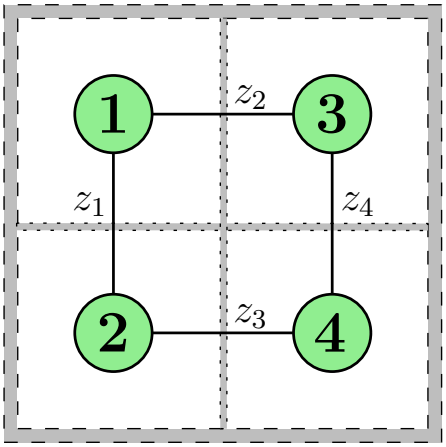


Figure 5.2: Graph representation of the 4-zone building: nodes represent zones (rooms) and edges their public variables.

The graph G can be also expressed with a so called adjacency matrix $\mathbf{A}_G \in \mathbb{R}^{N \times N}$ which is defined as

$$\mathbf{A}_{Gij} = \begin{cases} 1 & \text{there is an edge between the } i\text{-th and } j\text{-th node} \\ 0 & \text{otherwise.} \end{cases} \tag{5.22}$$

Hence, the graph G of the 4-zone building in the Fig. 5.2 has a following adjacency matrix

$$\mathbf{A}_G = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}. \tag{5.23}$$

Another form of the graph representation is an incidence matrix $\mathbf{I}_G \in \mathbb{R}^{N \times n_z}$ which gives relations between nodes and edges as follows

$$\mathbf{I}_{Gij} = \begin{cases} \pm 1 & \text{the node } i \text{ and edge } z_j \text{ are incident} \\ 0 & \text{otherwise.} \end{cases} \tag{5.24}$$

The sign of ones in $\mathbf{I}_G$ is given by the edges' direction. Even though our graph G is undirected, we declare the direction of edges in a sense the edge is directed from a lower numbered node to a higher numbered one. This will be later utilized in matrix $\mathbf{E}$ forming.

The incidence matrix related to our concrete graph G is given by

$$\mathbf{I}_G = \begin{bmatrix} -1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{5.25}$$

Now, we assemble the matrix $\mathbf{E}$ using the incidence matrix $\mathbf{I}_G$:

1. We go sequentially through the rows of $\mathbf{I}_G$ and search for nonzero elements.
2. Found nonzeros are placed stepwise at preserved column positions but independent rows of $\mathbf{E}$.
3. We "blow up" the matrix $\mathbf{E}$ by substituting ones for identity matrices $\mathbf{I}_{n_{pw}}$ and minus ones for left to right flipped identity matrices $\hat{\mathbf{I}}_{n_{pw}}$.

Using this procedure we obtain the matrix $\mathbf{E}$ for the 4-zone R1C0 model

$$\mathbf{E}_{R1C0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.26}$$

5.5 Global identification problem

At this time, we are able to use a dual decomposition method in the hypergraph form (4.5.2) and make the building identification distributed. Firstly, let's write a Lagrangian for the 4-zone building ($N = 4$)

$$\begin{aligned} L(\boldsymbol{\theta}_{1,\dots,N}, \mathbf{z}, \boldsymbol{\lambda}) &= \sum_{i=1}^{N} V_i(\boldsymbol{\theta}_i) + \boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{y} - \mathbf{E}\mathbf{z}) \\ &= \sum_{i=1}^{N} \left(V_i(\boldsymbol{\theta}_i) + \boldsymbol{\lambda}_i^{\mathrm{T}} \boldsymbol{\theta}_{p_i} \right) - \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{E}\mathbf{z}, \end{aligned} \tag{5.27}$$

where the vector $\mathbf{y}$ is built from all public variables $\boldsymbol{\theta}_{p_i}$, $i = 1, \dots, N$ in a sense that follows a structure of the graph G (Fig. 5.2) and the matrix $\mathbf{E}$ (5.26).

The dual problem can be now split into local identification problems

$$q_i(\boldsymbol{\lambda}_i) = \min_{\boldsymbol{\theta}_i} \; F_i(\boldsymbol{\theta}_i) = V_i(\boldsymbol{\theta}_i) + \boldsymbol{\lambda}_i^{\mathrm{T}} \boldsymbol{\theta}_{p_i}, \quad i = 1, \dots, N, \tag{5.28}$$

and the global identification problem has a form (as mentioned at (4.24))

$$\begin{aligned} \underset{\boldsymbol{\lambda}}{\text{maximize}} \quad & q(\boldsymbol{\lambda}) = \sum_{i=1}^{N} q_i(\boldsymbol{\lambda}_i) \\ \text{subject to} \quad & \mathbf{E}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{0}. \end{aligned} \tag{5.29}$$

This master problem can be solved via standard iterative gradient methods

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^{k} + \alpha^{k}\left(\mathbf{y}^{k*} - \mathbf{E}\hat{\mathbf{z}}^{k}\right), \tag{5.30}$$

the last thing we need to determine is a step size α^k.

5.5.1 Step size rules

We give just a brief description of several significant rules that are being used for the global problem solution. All these step size rules including their convergence analysis etc. are thoroughly described in [9, 11].

1. **Constant step size**

   The simplest way how to choose a step size is to fix it for a concrete value, that is $\alpha^k = \alpha,\ \alpha > 0$.

2. **Variable - nonsummable diminishing step size**

   The sequence of steps is described by

   $$\alpha^k \geq 0, \quad \lim_{k\to\infty} \alpha^k = 0, \quad \sum_{k=1}^{\infty} \alpha^k = \infty. \tag{5.31}$$

   Specifically, we used this step size

   $$\alpha^k = \frac{a}{\sqrt{k}}, \quad a > 0. \tag{5.32}$$

3. **Advanced rules - Nesterov's algorithm**

   This algorithm presented by Nesterov in 1983 is based on heavy balls gradient methods but the gradient is evaluated at an extrapolated point. Unlike the standard gradients methods with a rate of convergence of order $1/k$ after k steps, Nesterov's algorithm converges with rate of order $1/k^2$ (for detailed description see [9, 13]).

5.6 Local identification problem

In this section, we focus on detailed formulation and solution search of the local problem which covers identification of one zone.

This optimization problem arises from (5.28) with an extra constraint on parameters (5.5)

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{minimize}} \quad & F(\boldsymbol{\theta}) = V(\boldsymbol{\theta}) + \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\theta}_p \\ \text{subject to} \quad & \boldsymbol{\theta} \succ \mathbf{0}. \end{aligned} \tag{5.33}$$

Firstly, assume $V(\boldsymbol{\theta})$ being the LSE cost function (see (5.9))

$$V(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\mathcal{E}}(\boldsymbol{\theta})^{\mathrm{T}}\boldsymbol{\mathcal{E}}(\boldsymbol{\theta}) = \frac{1}{2}\left(\mathbf{Y} - \hat{\mathbf{Y}}(\boldsymbol{\theta})\right)^{\mathrm{T}}\left(\mathbf{Y} - \hat{\mathbf{Y}}(\boldsymbol{\theta})\right), \tag{5.34}$$

the vector of predictions $\hat{\mathbf{Y}}(\boldsymbol{\theta}) \in \mathbb{R}^{K\times 1}$ is given by

$$\hat{\mathbf{Y}}(\boldsymbol{\theta}) = \left(\mathbf{C}\hat{\mathbf{X}}(\boldsymbol{\theta})\right)^{\mathrm{T}} = \hat{\mathbf{X}}(\boldsymbol{\theta})^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}, \tag{5.35}$$

where the matrix of state predictions $\hat{\mathbf{X}}(\boldsymbol{\theta}) \in \mathbb{R}^{n\times K}$ is formed as

$$\hat{\mathbf{X}}(\boldsymbol{\theta}) = \begin{bmatrix} \hat{\mathbf{x}}_1(\boldsymbol{\theta}) & \dots & \hat{\mathbf{x}}_K(\boldsymbol{\theta}) \end{bmatrix} \tag{5.36}$$

and finally the state predictions $\hat{\mathbf{x}}_k(\boldsymbol{\theta}),\ k = 1, \dots, K$ are evaluated using the dicrete-time GBM 5.13.

Hence, the cost function in (5.33) can be expressed using (5.20) as

$$F(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\mathcal{E}}(\boldsymbol{\theta})^{\mathrm{T}}\boldsymbol{\mathcal{E}}(\boldsymbol{\theta}) + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{S}\boldsymbol{\theta}. \tag{5.37}$$

To minimize this cost function $F(\boldsymbol{\theta})$ we can use again one of gradient descent methods. The crucial part of these methods is evaluation of the cost function's gradient. This can be carried out either by its numerical approximation or by a computation using an exact analytical derivation. We proceed with the second option and express the gradient analytically, though its derivation is neither easy nor straightforward.

Remark 5.1. The local optimization problem (5.33) was solved by an interior-point algorithm of `fmincon` function from the Optimization Toolbox™.

5.6.1 Gradient computation

A gradient (or generally a Jacobian matrix) of the cost function actually represents its sensitivity to parameter changes. Let's denote it as $\mathbf{J}(\boldsymbol{\theta}) \in \mathbb{R}^{1\times n_p}$ and differentiate $F(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}^{\mathrm{T}}$

$$\mathbf{J}(\boldsymbol{\theta}) = \frac{\partial F(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\mathrm{T}}} = -\boldsymbol{\mathcal{E}}(\boldsymbol{\theta})^{\mathrm{T}}\frac{\partial \hat{\mathbf{Y}}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\mathrm{T}}} + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{S}. \tag{5.38}$$

We can see the red written term in (5.38) is essential for the gradient computation and needs to be further derived (the steps of derivation follow [12]). This matrix $\frac{\partial \hat{\mathbf{Y}}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\mathrm{T}}} \in \mathbb{R}^{K\times n_p}$ represents output parameter sensitivity and using (5.35) we obtain

$$\frac{\partial \hat{\mathbf{Y}}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\mathrm{T}}} = \frac{\partial}{\partial \boldsymbol{\theta}^{\mathrm{T}}}\left(\mathbf{C}\hat{\mathbf{X}}(\boldsymbol{\theta})\right)^{\mathrm{T}} \simeq \frac{\partial \hat{\mathbf{X}}(\boldsymbol{\theta})^{\mathrm{T}}}{\partial \boldsymbol{\theta}^{\mathrm{T}}}\mathbf{C}^{\mathrm{T}}, \tag{5.39}$$

where the term $\frac{\partial \hat{\mathbf{X}}(\boldsymbol{\theta})^{\mathrm{T}}}{\partial \boldsymbol{\theta}^{\mathrm{T}}} \in \mathbb{R}^{K \times n \times n_p}$ would be a "cubic matrix" generally. Hence, we must differentiate the output sensitivity vector $\hat{\mathbf{Y}}(\boldsymbol{\theta})$ with respect to individual parameters separately

$$\frac{\partial \hat{\mathbf{Y}}(\boldsymbol{\theta})}{\partial \theta_i} = \frac{\partial \hat{\mathbf{X}}(\boldsymbol{\theta})^{\mathrm{T}}}{\partial \theta_i}\mathbf{C}^{\mathrm{T}} = \boldsymbol{\mathcal{X}}_i(\boldsymbol{\theta})^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}, \quad i = 1, \ldots, n_p \tag{5.40}$$

and then assemble it as

$$\frac{\partial \hat{\mathbf{Y}}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\mathrm{T}}} = \begin{bmatrix} \boldsymbol{\mathcal{X}}_1(\boldsymbol{\theta})^{\mathrm{T}}\mathbf{C}^{\mathrm{T}} & \ldots & \boldsymbol{\mathcal{X}}_{n_p}(\boldsymbol{\theta})^{\mathrm{T}}\mathbf{C}^{\mathrm{T}} \end{bmatrix}. \tag{5.41}$$

The matrix $\boldsymbol{\mathcal{X}}_i(\boldsymbol{\theta}) \in \mathbb{R}^{n \times K}$ is formed from state sensitivities to a parameter θ_i

$$\boldsymbol{\mathcal{X}}_i(\boldsymbol{\theta}) = \begin{bmatrix} \boldsymbol{\xi}_1^i(\boldsymbol{\theta}) & \ldots & \boldsymbol{\xi}_K^i(\boldsymbol{\theta}) \end{bmatrix}, \quad \boldsymbol{\xi}_k^i(\boldsymbol{\theta}) = \frac{\partial}{\partial \theta_i}\hat{\mathbf{x}}_k(\boldsymbol{\theta}). \tag{5.42}$$

We need express state sensitivities $\boldsymbol{\xi}_k^i(\boldsymbol{\theta})$. To achieve that we utilize the discrete-time GBM predictor

$$\begin{aligned} \boldsymbol{\xi}_{k+1}^i(\boldsymbol{\theta}) &= \frac{\partial}{\partial \theta_i}\hat{\mathbf{x}}_{k+1}(\boldsymbol{\theta}) = \frac{\partial}{\partial \theta_i}\left(\mathbf{A}(\boldsymbol{\theta}) \cdot \hat{\mathbf{x}}_k(\boldsymbol{\theta})\right) + \frac{\partial \mathbf{B}(\boldsymbol{\theta})}{\partial \theta_i}\mathbf{u}_k \\ &= \frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_i}\hat{\mathbf{x}}_k(\boldsymbol{\theta}) + \mathbf{A}(\boldsymbol{\theta})\underbrace{\frac{\partial \hat{\mathbf{x}}_k(\boldsymbol{\theta})}{\partial \theta_i}}_{\boldsymbol{\xi}_k^i(\boldsymbol{\theta})} + \frac{\partial \mathbf{B}(\boldsymbol{\theta})}{\partial \theta_i}\mathbf{u}_k \\ &= \mathbf{A}_i(\boldsymbol{\theta})\hat{\mathbf{x}}_k(\boldsymbol{\theta}) + \mathbf{A}(\boldsymbol{\theta})\boldsymbol{\xi}_k^i(\boldsymbol{\theta}) + \mathbf{B}_i(\boldsymbol{\theta})\mathbf{u}_k. \end{aligned} \tag{5.43}$$

The equation (5.44) can be understood as an extended discrete-time LTI system for the state prediction $\hat{\mathbf{x}}(\boldsymbol{\theta})$ and state sensitivity $\boldsymbol{\xi}^i(\boldsymbol{\theta})$. Let's write this extended system in a matrix form

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{x}}_{k+1} \\ \boldsymbol{\xi}_{k+1}^i \end{bmatrix} &= \begin{bmatrix} \mathbf{A}(\boldsymbol{\theta}) & \mathbf{0} \\ \mathbf{A}_i(\boldsymbol{\theta}) & \mathbf{A}(\boldsymbol{\theta}) \end{bmatrix}\begin{bmatrix} \hat{\mathbf{x}}_k \\ \boldsymbol{\xi}_k^i \end{bmatrix} + \begin{bmatrix} \mathbf{B}(\boldsymbol{\theta}) \\ \mathbf{B}_i(\boldsymbol{\theta}) \end{bmatrix}\mathbf{u}_k, \quad i = 1, \ldots, n_p, \\ \hat{y}_k &= \begin{bmatrix} \mathbf{C} & \mathbf{0}^{\mathrm{T}} \end{bmatrix}\begin{bmatrix} \hat{\mathbf{x}}_k \\ \boldsymbol{\xi}_k^i \end{bmatrix}, \\ \begin{bmatrix} \hat{\mathbf{x}}_0 \\ \boldsymbol{\xi}_0^i \end{bmatrix} &= \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{0} \end{bmatrix}. \end{aligned} \tag{5.44}$$

Using this extended system (5.44) we are able to compute both state predictions and state sensitivities. The last thing remaining to be determined are derivatives of discrete-time matrices $\mathbf{A}_i(\boldsymbol{\theta})$ and $\mathbf{B}_i(\boldsymbol{\theta})$.

We can apply the approach (2.25) again, this time for the extended system (5.44)

$$\left[\begin{array}{cc|c} \mathbf{A}(\boldsymbol{\theta}) & \mathbf{0} & \mathbf{B}(\boldsymbol{\theta}) \\ \mathbf{A}_i(\boldsymbol{\theta}) & \mathbf{A}(\boldsymbol{\theta}) & \mathbf{B}_i(\boldsymbol{\theta}) \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{I} \end{array}\right] = \exp\left(\left[\begin{array}{cc|c} \mathbf{A}_c(\boldsymbol{\theta}) & \mathbf{0} & \mathbf{B}_c(\boldsymbol{\theta}) \\ \mathbf{A}_{c_i}(\boldsymbol{\theta}) & \mathbf{A}_c(\boldsymbol{\theta}) & \mathbf{B}_{c_i}(\boldsymbol{\theta}) \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right] T_s\right), \quad i = 1, \ldots, n_p, \tag{5.45}$$

where

$$\mathbf{A}_{c_i}(\boldsymbol{\theta}) = \frac{\partial \mathbf{A}_c(\boldsymbol{\theta})}{\partial \theta_i} \quad \text{and} \quad \mathbf{B}_{c_i}(\boldsymbol{\theta}) = \frac{\partial \mathbf{B}_c(\boldsymbol{\theta})}{\partial \theta_i} \tag{5.46}$$

are derivatives of continuous-time matrices with respect to individual parameters.

We remark that a ubiquitous parameter vector $\boldsymbol{\theta}$ in parentheses within this whole section emphasizes the fact that all the quantities are functions of $\boldsymbol{\theta}$. In particular, this means for them to be evaluated every time again for new parameter values during iterations of the local optimization problem (5.33).

5.6.2 Maximum likelihood estimation case

In the previous section, we assumed $V(\boldsymbol{\theta})$ being a LSE cost function (5.34). Now, we will also present a local cost function $F(\boldsymbol{\theta})$ and its gradient $\mathbf{J}(\boldsymbol{\theta})$ for the MLE case.

The estimation cost function is given by (5.16)

$$V(\boldsymbol{\theta}) = \frac{1}{2} \log \left(\frac{1}{N} \boldsymbol{\mathcal{E}}(\boldsymbol{\theta})^{\mathrm{T}} \boldsymbol{\mathcal{E}}(\boldsymbol{\theta}) \right) \tag{5.47}$$

and the overall local cost function of (5.33) is

$$F(\boldsymbol{\theta}) = \frac{1}{2} \log \left(\frac{1}{N} \boldsymbol{\mathcal{E}}(\boldsymbol{\theta})^{\mathrm{T}} \boldsymbol{\mathcal{E}}(\boldsymbol{\theta}) \right) + \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{S} \boldsymbol{\theta}. \tag{5.48}$$

A gradient of this cost function is then given by

$$\mathbf{J}(\boldsymbol{\theta}) = \frac{\partial F(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\mathrm{T}}} = - \left(\boldsymbol{\mathcal{E}}(\boldsymbol{\theta})^{\mathrm{T}} \boldsymbol{\mathcal{E}}(\boldsymbol{\theta}) \right)^{-1} \boldsymbol{\mathcal{E}}(\boldsymbol{\theta})^{\mathrm{T}} \frac{\partial \hat{\mathbf{Y}}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\mathrm{T}}} + \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{S} \tag{5.49}$$

and we can see a crucial role plays the red written term again (thus can be computed as stated at5.6.1).

5.6.3 Parameter normalization

The real values of the parameter $\boldsymbol{\theta}$ to be estimated are numerically in a very different scale. For example, the capacities C calculated using the values in the Tab. 2.1 are in a range $\sim 10^4 - 10^7$ J $\cdot$ K^{-1}, on the other hand conductances UA are in a range $\sim 10^0 - 10^2$ W $\cdot$ K^{-1}.

These different scales of optimal parameter values significantly worsen numerical properties and stability of optimization procedures. To eliminate this adverse issue we normalize the parameters by their true values (or by their expected scales in real situations). Thus, we define normalized parameters as

$$\boldsymbol{\theta}_n = \mathbf{N}^{-1} \boldsymbol{\theta} \tag{5.50}$$

and vice versa a return to the denormalized parameters

$$\boldsymbol{\theta} = \mathbf{N} \boldsymbol{\theta}_n, \tag{5.51}$$

where $\mathbf{N} \in \mathbb{R}^{n_p \times n_p}$ is a diagonal normalizing matrix with scale values on the main diagonal. Hence, our optimization variable is no longer $\boldsymbol{\theta}$ directly but its normalized companion $\boldsymbol{\theta}_n$.

Since, we now optimize over $\boldsymbol{\theta}_n$, our objective function $F(\boldsymbol{\theta}_n)$ is so evaluated. Furthermore, we need to express the gradient as a function of $\boldsymbol{\theta}_n$ as well

$$\mathbf{J}(\boldsymbol{\theta}_n) = \frac{\partial F(\boldsymbol{\theta}_n)}{\partial \boldsymbol{\theta}_n^{\mathrm{T}}} = \frac{\partial F(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\mathrm{T}}} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\theta}_n^{\mathrm{T}}} = \frac{\partial F(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\mathrm{T}}} \mathbf{N} = \mathbf{J}(\boldsymbol{\theta})\mathbf{N}, \tag{5.52}$$

where the gradient $\mathbf{J}(\boldsymbol{\theta})$ is computed as proposed at 5.6.1. Finally, we remark that an optimal normalized parameter vector which we will search for during identification ought to be

$$\boldsymbol{\theta}_n^* = \mathbf{1}, \tag{5.53}$$

if the matrix $\mathbf{N}$ contains true initial values of the system which is to be identified.

5.7 Modification for embedded usage

When embedded usage is concerned (e.g. with the ARM Cortex M3 core), there is a need to modify (optimize) the whole identification algorithm the way it would take as less both global and local iterations as possible and thus respect a limited speed of communication and limited memory size.

At first, let's focus on global iterations. It is obvious a decrease of global iterations reduces communication load among zones' agents. A number of global iterations mainly depends on the used step size rule (for the given precision threshold r_{THRES}). We will carry out several identification processes for various step sizes in the chapter 6 and it will be seen that a proper step size (rule) can reduce the number of global iterations manyfold. Another matter is a choice of the precision threshold which is sufficient probably even when being less strict than $r_{\mathrm{THRES}} = 10^{-5}$ as presented at (6.9).

At second, we concentrate on local problem iterations. The local optimization problem (5.33) is solved by an iterative gradient descent algorithm which needs an initial guess of the optimization variable $\boldsymbol{\theta}_n^0$. We set this initial guess as tenfold scale values, i.e.

$$\boldsymbol{\theta}_n^0 = \mathbf{N} \left[10 \dots 10\right]^{\mathrm{T}}. \tag{5.54}$$

The i-th local optimization subproblem is being solved again with every next k-th global iteration which supplies a corresponding price vector $\boldsymbol{\lambda}_i^k$ to the i-th local agent. Hence, when the next local optimization is to be performed, it starts again with the initial guess (5.54) - this approach is called a cold start.

However, we can notice a local objective function in (5.33) is being modified just slightly (by $\boldsymbol{\lambda}_i^k$) with every new k-th global iteration. Therefore, we utilize this fact and use a found minimizer $\boldsymbol{\theta}_n^{*k-1}$ as a so called hot start initial guess for the k-th local optimization. An example of comparison between a number of iterations performed within a local optimization for the cold and hot start is in the Fig. 5.3.

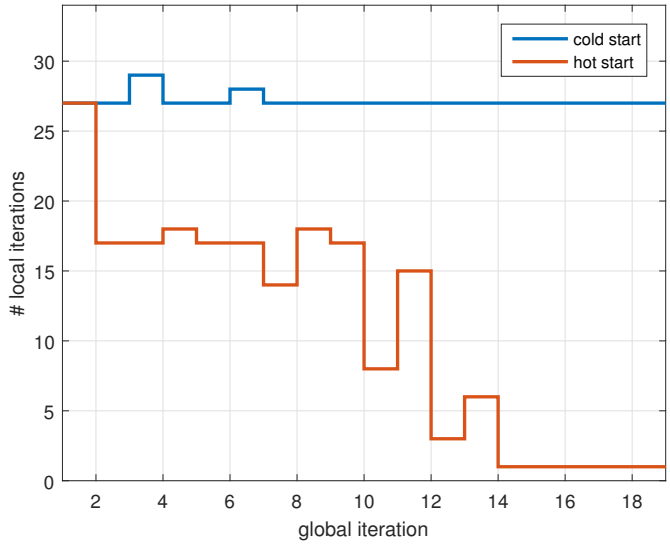


Figure 5.3: Comparison of the number of local iterations taken for the cold and hot start.

The total number of local iterations for the presented case is

$$\text{cold start}: \ \#_{L_c} = 516, \quad \text{hot start}: \ \#_{L_h} = 200. \tag{5.55}$$

Therefore, we can see the total number of local iterations $\#_L$ reduces more than twice when the hot start is being used.

In previous sections, we presented the LSE (5.9) and MLE (5.16) method. The identification results which will be presented in the chapter 6 were obtained both for the LSE and MLE. However, the LSE generally outperformed the MLE in most cases and thus we can say is most suitable for embedded usage.

Another thing, which could be utilized in embedded usage, is the analytically computed gradient. Nevertheless, the time of the identification process for analytically and numerically approximated gradient were almost the same, even though the analytically one usually took a few iterations less then numerically computed gradient.

Chapter 6

Identification results

This chapter presents main results of the building identification. We carry out several identification processes with different settings, that is:

- Data generator: either R1C0 (2.17) or R2C1 (2.14) building model,
- Additive input heat flow disturbances: we (do or do not) burden the data generator with positive biased heat flow disturbances at the input, i.e. we simulate there are other heat flow sources in a building that are unknown to the grey-box model,
- Global problem: various step size rules as proposed at 5.5.1,
- Local problem: various estimation methods - LSE (5.9) or MLE (5.16).

Afterwards, we compare identification results of all these settings according to following criteria:

- $\#_G$ - number of iterations of the global problem - suppose the global algorithm took g iterations till it reaches a desired precision threshold (6.9)

$$\#_G = g. \tag{6.1}$$

- $\#_L$ - total number of local iterations in all subproblems, i.e. the number is added up across all the global iterations and local subproblems

$$\#_L = \sum_{i=1}^{g} \sum_{j=1}^{4} l_{ij}, \tag{6.2}$$

  where l_{ij} is the number of iterations in a j-th subproblem under the i-th iteration of the global problem.

- t [s] - time of the whole identification process - starts running with the first iteration and stops after the g-th iteration of the global problem.
- Goodness of fit and parameters - will be introduced at a section 6.2.

Remark 6.1. All computations were performed using a dual core CPU Intel® Core™ i5-5200U.

6.1 Rich data importance

Measured data represent an input of the identification process. Although, the data cannot be arbitrary, because even a state-of-the-art identification algorithm fails without appropriate data.

The data supplied to the algorithm must be rich in contents, that is to contain as much information as possible. It would be ideal to excite all the modes a system has. Nevertheless, this is unreachable in practice for building identification, since we can hardly heat zones up to arbitrary temperature, especially not in presence of inhabitants.

Another restriction is a constraint on heat flow of heat exchanger (for our case stated as (2.26)). In addition, we have to realize the thermal dynamics of a building is very slow, therefore we need a long data sequence (approximately hundreds to thousands samples) to be sufficiently informative.

As proposed in [3] good excitation signals could be square waves corresponding to changing set point temperatures, step or multi-sinusoidal inputs and random Gaussian signals. An amount of information in data is closely related to the term *data-dependent identifiability* which can be expressed with a Fisher information matrix (for details see [3]).

In all cases that will be presented afterwards, we will use a ONE-DAY data acquisition while applying the closed-loop heating control. There is always a night and day set point temperature where the night one is naturally lower.

6.2 Validation of identified models

After a model is identified, it is convenient to measure its "quality" with respect to provided data, e.g. goodness of fit etc. This can be performed by several ways focusing on different aspects.

6.2.1 Goodness of fit

This can be provided by several so called metrics and among the most used belong these:

- Normalized Root Mean Square Error which is a relative metric and is defined as

$$fit_{\mathrm{NRMSE}} = \left(1 - \frac{\|\mathbf{Y} - \hat{\mathbf{Y}}\|_2}{\|\mathbf{Y} - \frac{1}{K}\sum_{k=1}^{K} y_k\|_2}\right) 100 \quad [\%], \tag{6.3}$$

- Mean Absolute Error which gives an average of the absolute errors

$$fit_{\mathrm{MAE}} = \frac{1}{K}\sum_{k=1}^{K} |y_k - \hat{y}_k| \quad [-], \tag{6.4}$$

- Mean Squared Error which gives an average of the squared errors

$$fit_{\mathrm{MSE}} = \frac{1}{K}\|\mathbf{Y} - \hat{\mathbf{Y}}\|_2^2 \quad [-]. \tag{6.5}$$

6.2.2 Goodness of parameters

At first, we focus on public parameters. The local copies of public parameters must satisfy the constraint (4.16) and an overall goodness of this constraint satisfaction is expressed by the consistency constraint residual (4.28). However, to show this goodness per parameter we must introduce another metric to verify a consensus between corresponding pairs of public variable. Hence, we present a residual public variable error defined as

$$r_{z_i} = |\mathbf{y}_k^* - \mathbf{y}_l^*|, \quad i = 1, \ldots, n_z, \tag{6.6}$$

where indices k and l satisfy

$$\mathbf{E}_{ki} = 1 \wedge \mathbf{E}_{li} = 1. \tag{6.7}$$

In other words, the residual (6.6) measures an absolute error between identified wall parameters of two adjacent zones.

At second, we focus on all parameters, i.e. both private and public ones. We express their optimality to initial parameter values of the data generator as

$$e_{\theta_i} = \frac{|\theta_i^* - \theta_i|}{\theta_i}, \quad i = 1, \ldots, n_p, \tag{6.8}$$

naturally this assumes we have knowledge of the initial values.

6.2.3 Precision threshold

The building identification problem is solved using the dual decomposition method. An implementation of this method is based on the Alg. 4 and as a stopping criterion serves the consistency constraint residual (4.28) which was chosen as

$$r_{\text{THRES}} = 10^{-5}. \tag{6.9}$$

This precision threshold on public variables equality is a reasonable trade-off between precision and a number of iterations of the global problem.

6.3 R1C0 data generator

The building models presented in the chapter 2 - both R1C0 (2.17) and R2C1 (2.14) - were utilized as data generators for an identification process. In order to simulate real conditions, we burden measured zone temperatures with additive white Gaussian noise that has properties

$$e_k \sim \mathcal{N}(0, 0.01T_s), \quad \text{cov}\{e_k, e_{k+i}\} = 0, \; i \neq 0, \tag{6.10}$$

where the variance corresponds to a standard deviation of 0.1 K for continuous-time measure.

At first, we use the R1C0 model as a data generator and so as the grey-box model to be identified. Therefore, optimal values of normalized vector parameter which we will search for

$$\boldsymbol{\theta}_{n_i}^* = \begin{bmatrix} C_z & UA_{w_1} & UA_{w_2} & UA_{w_3} & UA_{w_4} \end{bmatrix}^{\mathrm{T}}, \quad i = 1, \ldots, 4. \tag{6.11}$$

should be as close as possible to ones (as mentioned at (5.53)).

6.3.1 Disturbance-free case

There were carried out several identification processes. At the global algorithm we applied either a fixed step size or variable step size (for details see 5.5.1) while in the local problem we were using the LSE or MLE method.

The results are presented in a Tab. 6.1 and Tab. 6.2, respectively. A table 6.5 shows the goodness of fit for both the disturbance-free and disturbance case which will be described afterwards. We can see the fit_{NRMSE} is very good which is not surprising as our grey-box model R1C0 corresponds exactly to the data generator.

A group of figures 6.4 - 6.3 documents the results of identification using fixed step size and LSE. A Fig. 6.4 shows nicely the behaviour of the global algorithm which forces public variables of adjacent zones to be equal. Thus, the consistency constraint residual decreases which is shown in the Fig. 6.5b (in logarithmic scales). The next Fig. 6.1 depicts a correspondence between the measured data from the generator and identified model. Another part of data, which are measured heat flows of heat exchangers are in the Fig. 6.2. The last figure of this group (6.3) presents errors defined at (6.8) between original parameters of the generator and estimated ones obtained via identification.

LSE	**step size**	**# iterations Global**	**Local**	**time [s]**
fixed	0.001	37	1456	41.1
	0.002	19	885	25.8
	0.003	12	639	18.27
variable	0.002	105	1992	62.7
	0.005	17	698	20.1
	0.010	14	777	21.6

Table 6.1: Dependence of the number of iterations on various step sizes (R1C0 data, LSE, disturbance-free case).

MLE	**step size**	**# iterations Global**	**Local**	**time [s]**
fixed	0.2	17	851	64.4
	0.3	11	608	44.1
	0.4	21	998	76.3
variable	0.5	15	2245	72.1
	0.7	9	1414	46.3
	0.8	11	1622	52.3

Table 6.2: Dependence of the number of iterations on various step sizes (R1C0 data, MLE, disturbance-free case).

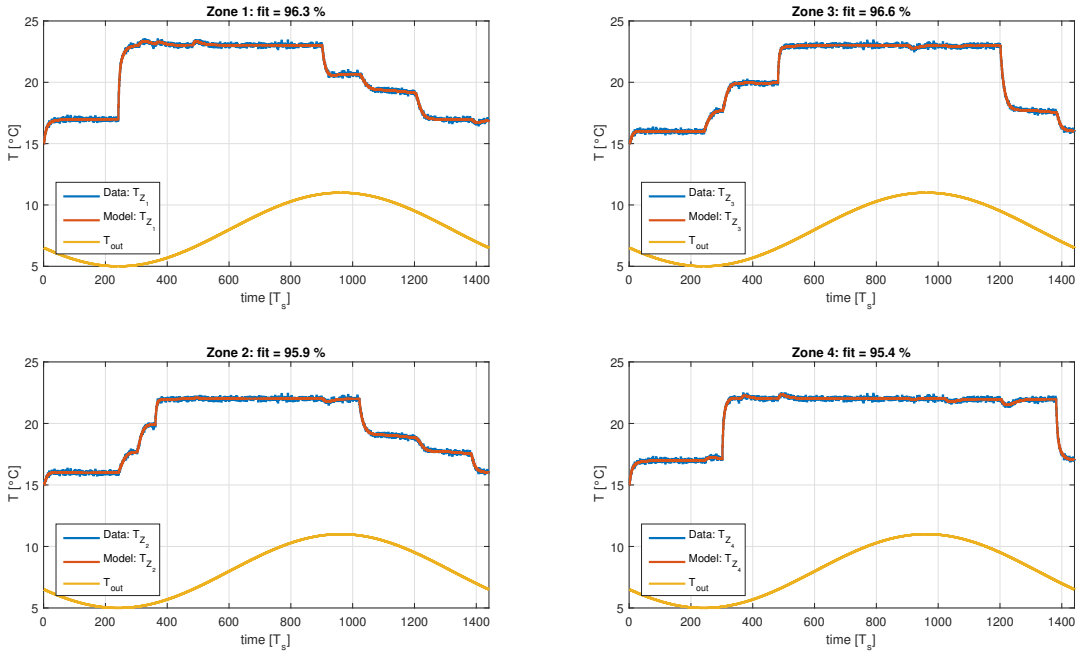


Figure 6.1: Comparison of measured output data and identified model simulation (air temperatures in zones) - R1C0 without disturbances.

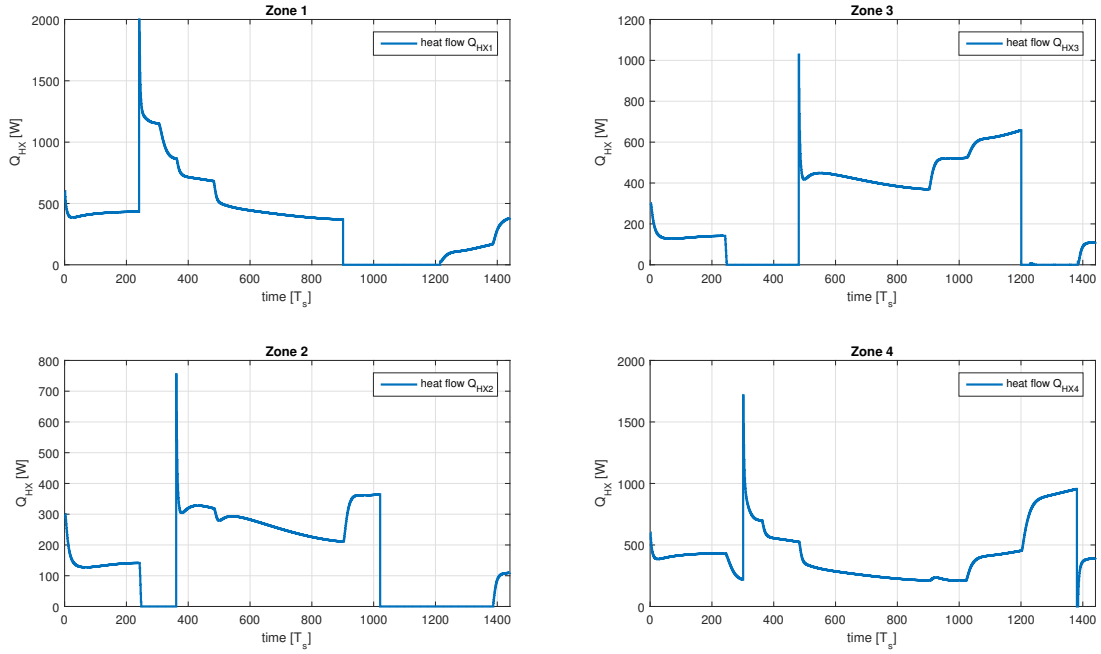


Figure 6.2: Measured input data - heat flows of heat exchangers (R1C0 without disturbances).

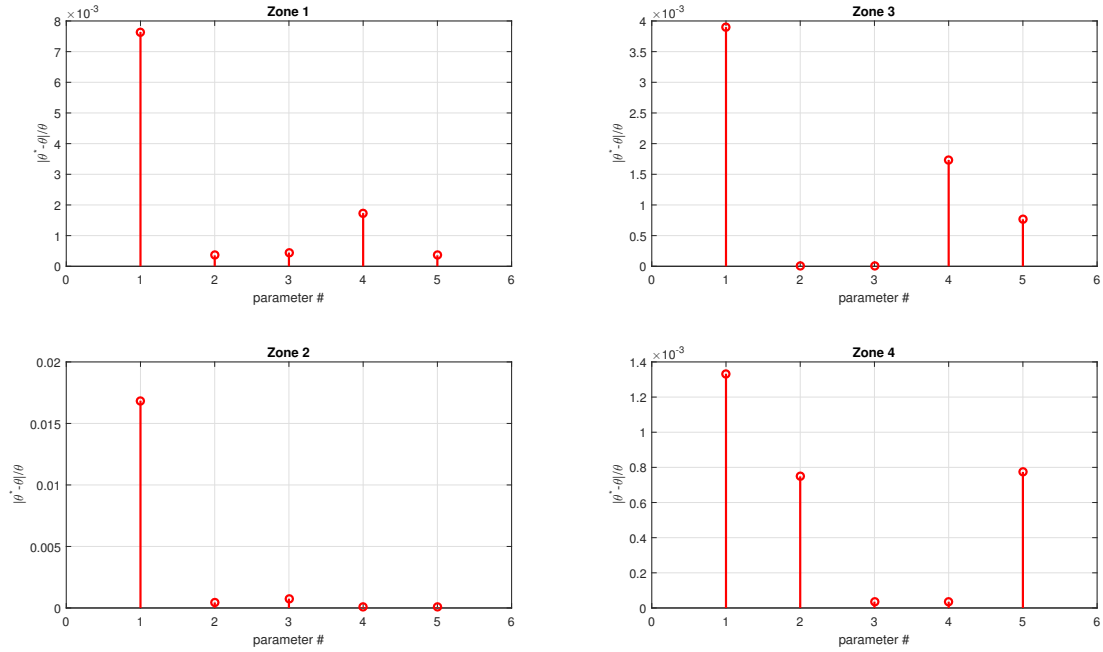


Figure 6.3: Error of individual θ_i parameters with respect to initial values - R1C0 without disturbances.

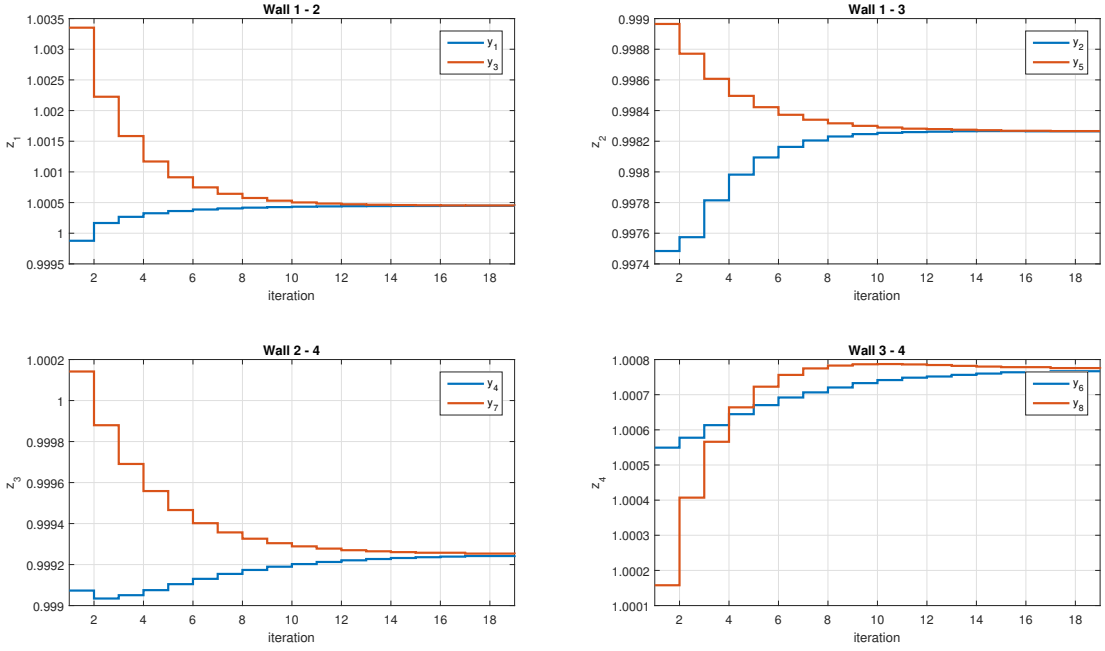
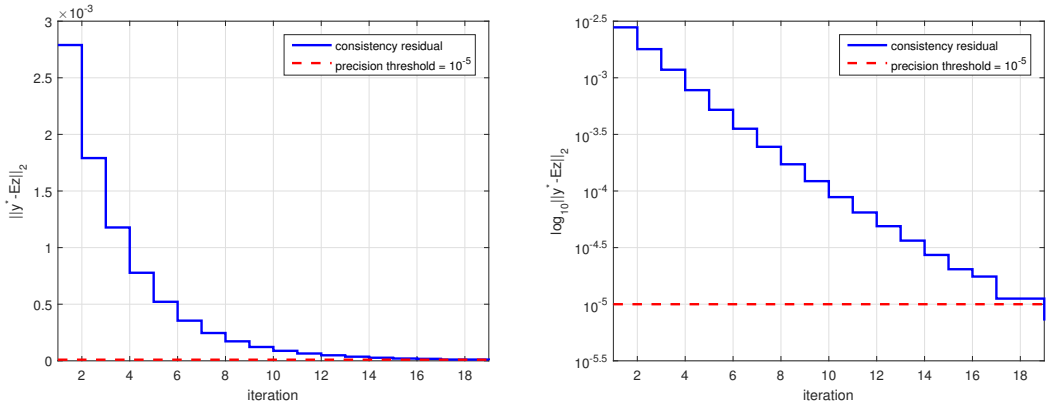


Figure 6.4: Convergence of public variables for a FIXED step size (R1C0 without disturbances, LSE).



(a) : linear scale **(b) :** logarithmic scale

Figure 6.5: Decrease of consistency constraint residual for a FIXED step size (R1C0 without disturbances).

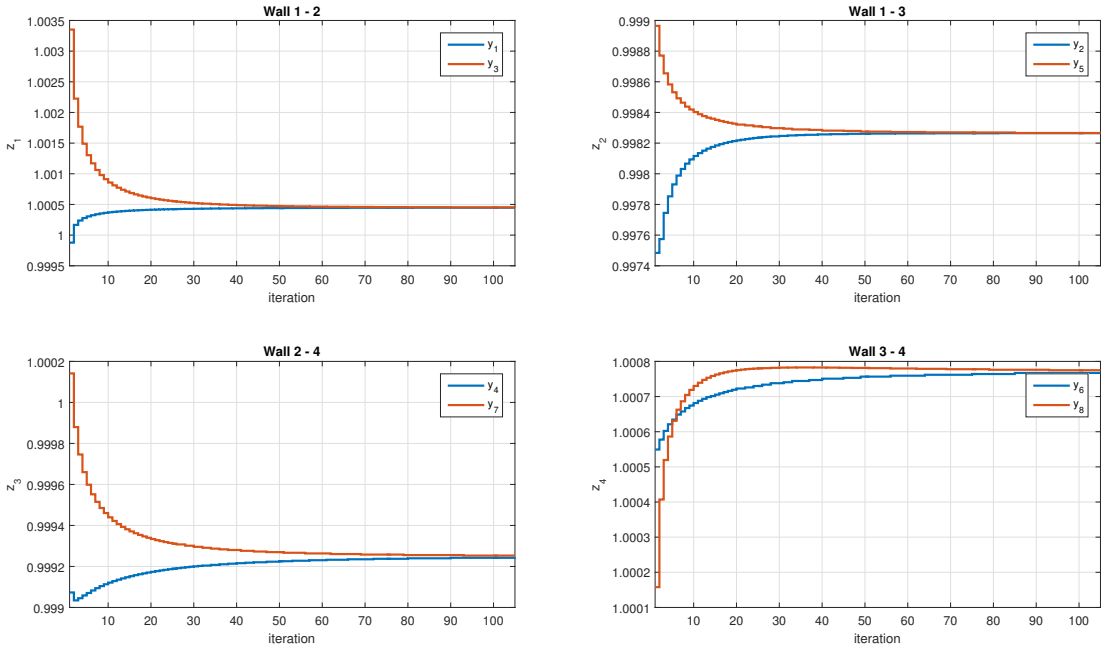
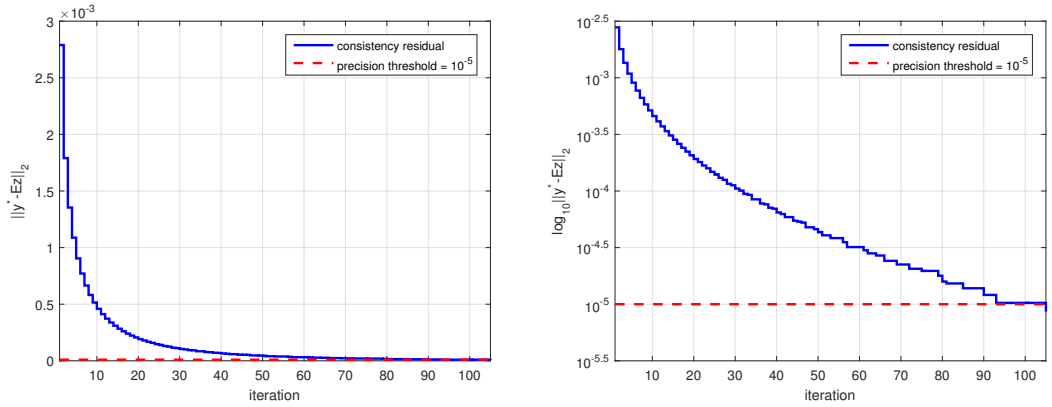


Figure 6.6: Convergence of public variables for a VARIABLE step size (R1C0 without disturbances, LSE).

6.3.2 Disturbance case

In the previous section, we were identifying the building model where only measurement data was burdened with a Gaussian white noise (6.10). Now, we will try to more approximate real conditions and let other heat sources affect a temperature in the zones. Since these additional heat flow sources are unknown to our grey-box model, we call



(a) : linear scale **(b) :** logarithmic scale

Figure 6.7: Decrease of consistency constraint residual for a VARIABLE step size (R1C0 without disturbances).

them disturbances. These disturbances simulate for instance occupancy of zones as one person approximately has a heat power ~ 80 W, furthermore a lot of electric appliances and devices produce heat as well. Hence, we simulate these disturbances as a biased white Gaussian noise

$$d_k \sim \mathcal{N}(50, 30^2 T_s), \quad \text{cov}\{d_k, d_{k+i}\} = 0, \; i \neq 0, \tag{6.12}$$

where the mean value of heat flow is 50 W and the variance corresponds to a standard deviation of 30 W for a continuous-time case. The general scheme of the system with disturbances and measurement noise is in the Fig. 2.7. Now, we perform several identifications with different settings - again, results are summarized in tables 6.3, 6.4 and the goodness of fit is presented in a Tab. 6.5.

LSE	step size	# iterations Global	# iterations Local	time [s]
fixed	0.001	34	1309	40.2
	0.002	26	1351	38.2
	0.003	17	939	26.6
variable	0.002	200	5200	162.4
	0.005	29	1279	37.4
	0.010	∞	∞	∞

Table 6.3: Dependence of the number of iterations on various step sizes (R1C0 data, LSE, disturbance case). Note the algorithm diverges for the variable step size $\alpha_k = 0.01/\sqrt{k}$ unlike the disturbance-free case.

MLE	step size	# iterations Global	Local	time [s]
fixed	0.05	52	2149	149.7
	0.08	31	1363	91.2
	0.1	24	1090	70.35
variable	0.1	193	7559	588.9
	0.2	45	1947	132.0
	0.3	19	961	57.28

Table 6.4: Dependence of the number of iterations on various step sizes (R1C0 data, MLE, disturbance case).

DIS-TURB.	zone	Goodness of fit NRMSE [%]	MAE [-]	MSE [-]	r_z	wall
×	1	96.29	0.0789	0.0099	$3 \cdot 10^{-6}$	1
	2	95.92	0.0789	0.0099	$2 \cdot 10^{-6}$	2
	3	96.59	0.0789	0.0099	$6 \cdot 10^{-6}$	3
	4	95.36	0.0789	0.0099	$7 \cdot 10^{-6}$	4
✓	1	89.01	0.2316	0.0851	$4 \cdot 10^{-6}$	1
	2	88.26	0.2297	0.0790	$1 \cdot 10^{-6}$	2
	3	89.25	0.2499	0.0953	$6 \cdot 10^{-6}$	3
	4	86.12	0.2418	0.0887	$6 \cdot 10^{-6}$	4

Table 6.5: Goodness of fit with(out) disturbance presence (for R1C0 data). Both LSE and MLE give the same fit for all step sizes, the difference is only in the number of iterations. The part right to a double line contains absolute errors of corresponding pairs public variables.

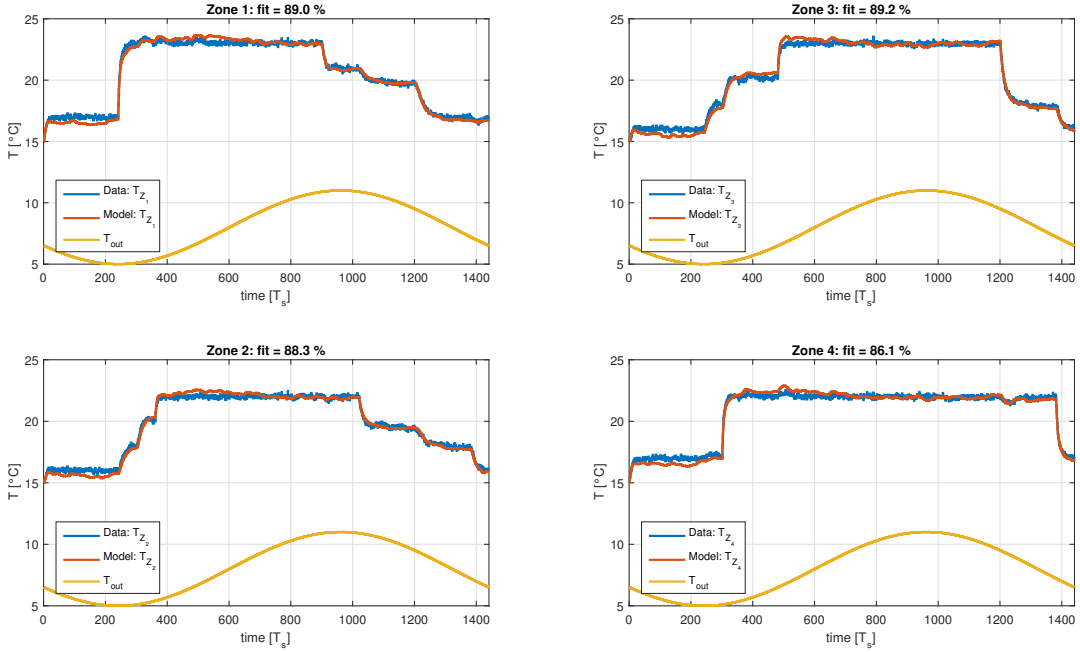


Figure 6.8: Comparison of measured output data and identified model simulation (air temperatures in zones) - R1C0 with disturbances.

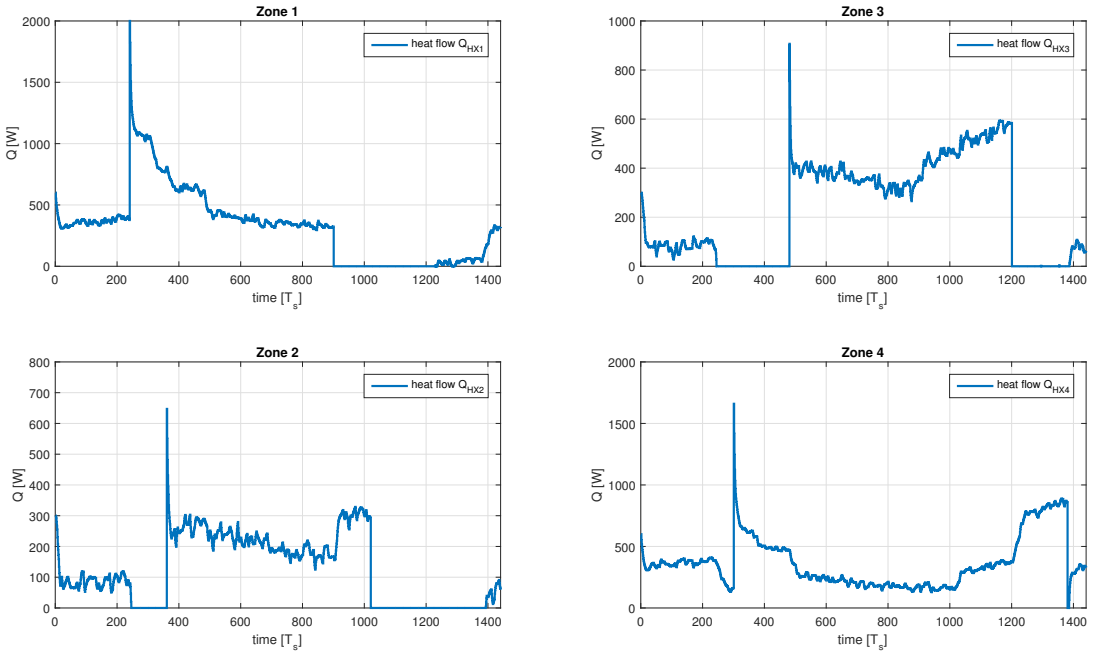


Figure 6.9: Measured input data - heat flows of heat exchangers (R1C0 with disturbances).

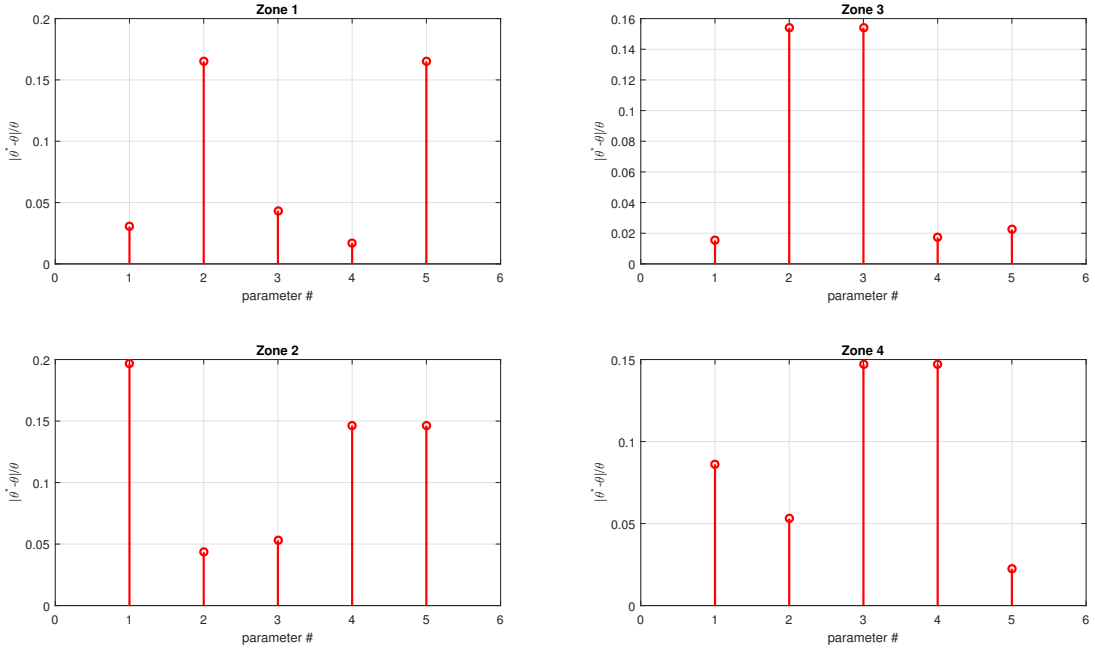


Figure 6.10: Error of individual θ_i parameters with respect to initial values - R1C0 with disturbances.

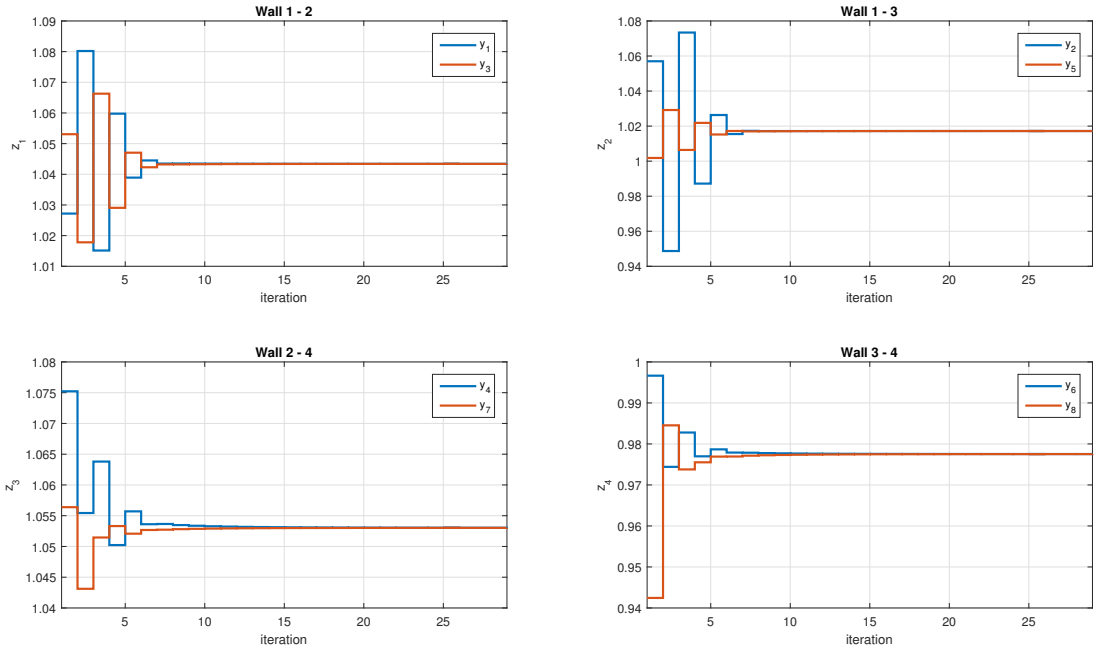
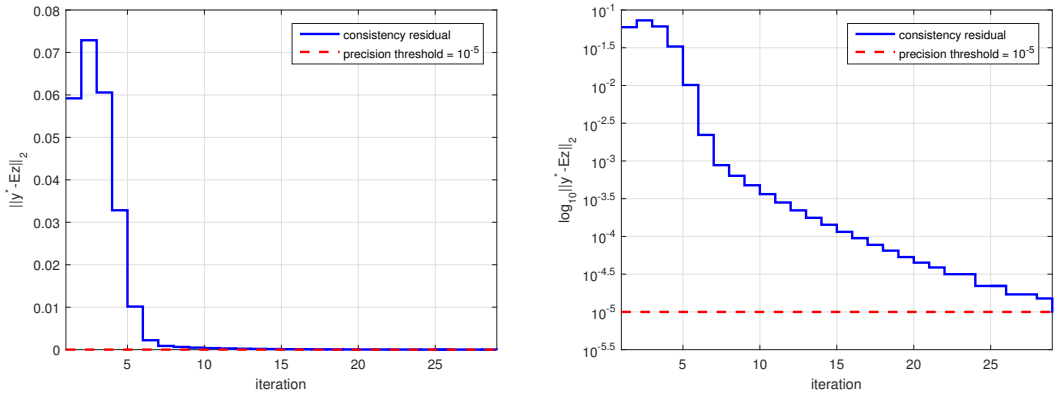


Figure 6.11: Convergence of public variables for a VARIABLE step size (R1C0 with disturbances, LSE).



(a) : linear scale **(b) :** logarithmic scale

Figure 6.12: Decrease of consistency constraint residual for a VARIABLE step size (R1C0 with disturbances).

6.4 R2C1 data generator

At second, we will simulate a situation when we try identifying a system with our assumed grey-box model but this grey-box is a simplification of the real system, i.e. neglects low significant states (with poor dynamics) etc. We will use the R2C1 model as a data generator but identify the R1C0 grey-box model. Therefore, optimal values of normalized

parameter vector which we will search for will no longer be exactly ones. However, we can still verify a correctness of the identified model as we calculate a goodness of fit.

6.4.1 Disturbance-free case

At first again, we begin with the simpler case where no disturbances are taken into account. The results computed using LSE method and various step sizes for the global problem are presented in a Tab. 6.6. For this time, we do not present all the figures as in previous cases since they do not differ so much from the disturbance case figures which will be shown in the next subsection. A goodness of fit for this case is presented along with the disturbance case in a Tab. 6.8.

LSE	step size	# iterations Global	# iterations Local	time [s]
fixed	0.0005	55	1970	54.8
	0.001	27	1096	34.9
	0.0012	24	1085	29.9
variable	0.001	209	3279	106.7
	0.002	47	1350	38.9
	0.003	47	1672	48.6

Table 6.6: Dependence of the number of iterations on various step sizes (R2C1 data, LSE, disturbance-free case).

6.4.2 Disturbance case

Now, we will try to identify the system burdened with disturbances introduced at (6.12). We remark this is the worst case for the identification process, since the real system (building) distinguishes from the grey-box model applied within the identification, in addition there are unmeasured disturbances which affect the air temperature in zones. Therefore this case approximates most of all a real situation for the building identification.

LSE	step size	# iterations Global	# iterations Local	time [s]
fixed	0.0005	47	1893	53.3
	0.0008	38	1542	43.0
	0.001	22	1049	28.9
variable	0.001	164	3183	97.0
	0.002	25	947	28.0
	0.0025	45	1650	48.1

Table 6.7: Dependence of the number of iterations on various step sizes (R2C1 data, LSE, disturbance case).

DIS-TURB.	zone	Goodness of fit NRMSE [%]	MAE [-]	MSE [-]	r_z	wall
×	1	83.80	0.3223	0.1873	$4 \cdot 10^{-6}$	1
	2	82.29	0.3334	0.2100	$3 \cdot 10^{-6}$	2
	3	82.89	0.4134	0.2738	$3 \cdot 10^{-7}$	3
	4	75.04	0.3981	0.3981	$8 \cdot 10^{-6}$	4
✓	1	80.62	0.3828	0.2592	$4 \cdot 10^{-6}$	1
	2	80.00	0.3680	0.2559	$5 \cdot 10^{-6}$	2
	3	80.52	0.4596	0.3445	$6 \cdot 10^{-6}$	3
	4	71.53	0.4479	0.3742	$11 \cdot 10^{-6}$	4

Table 6.8: Goodness of fit both with and without disturbance presence (for R2C1 data, LSE). The right part of the table contains absolute errors of corresponding pairs public variables.

Results showing a convergence rate for various step sizes are expressed in a Tab. 6.7. The goodness of fit which is presented in a Tab. 6.8 is mostly about 80 % (fit_{NRMSE}) and can be seen in the Fig. 6.13. The corresponding measured input data, i.e. heat flows of heat exchangers, are depicted in the Fig. 6.14. A figure 6.15 shows a convergence of public variables for a variable step size $\alpha_k = \frac{0.002}{\sqrt{k}}$ and the last figure 6.16 depicts a decrease of the consistency constraint residual.

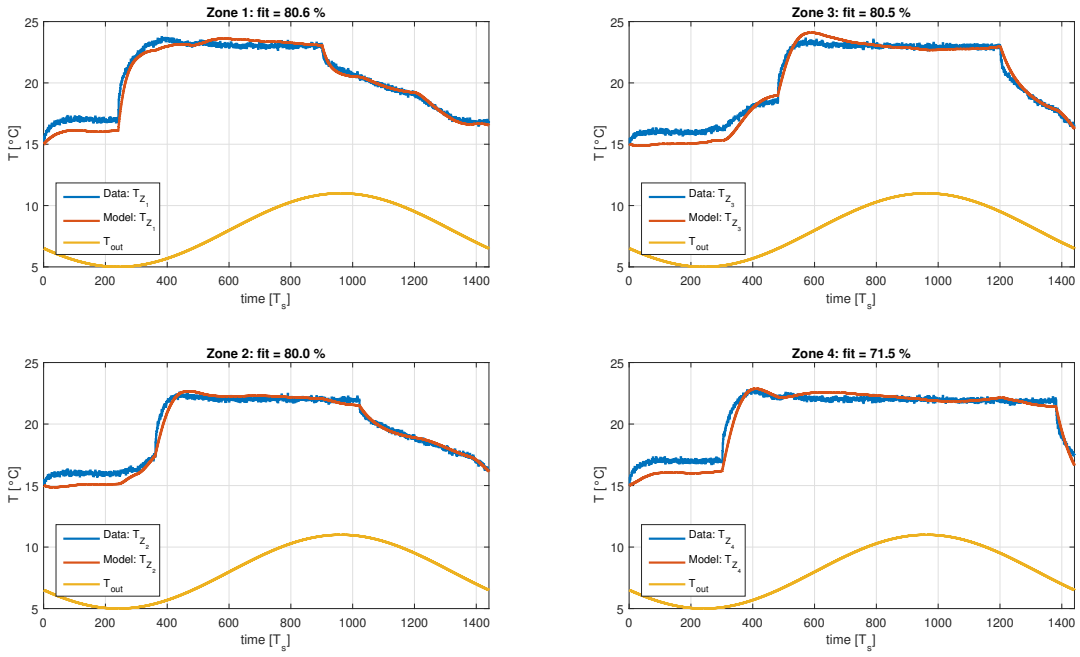


Figure 6.13: Comparison of measured output data and identified model simulation (air temperatures in zones) - R2C1 with disturbances.

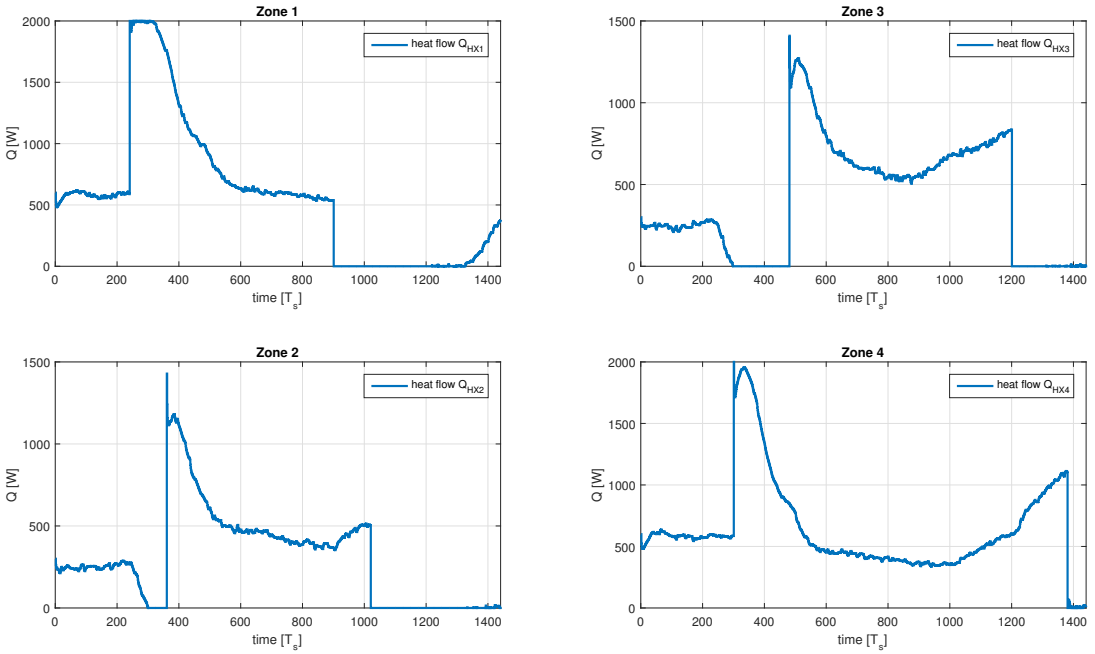


Figure 6.14: Measured input data - heat flows of heat exchangers (R2C1 with disturbances).

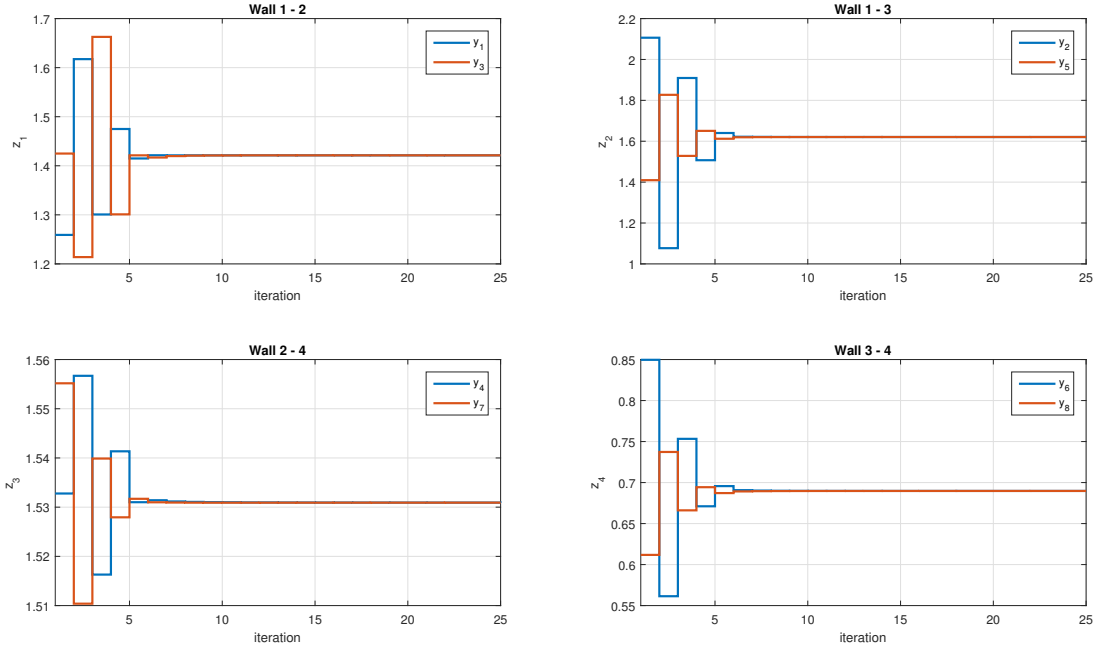
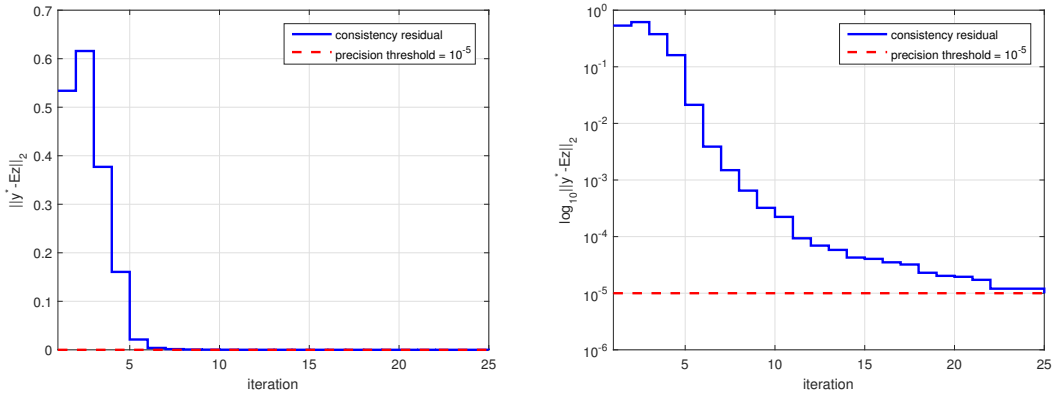


Figure 6.15: Convergence of public variables for a VARIABLE step size (R2C1 with disturbances, LSE).



(a) : linear scale **(b) :** logarithmic scale

Figure 6.16: Decrease of consistency constraint residual for a VARIABLE step size (R2C1 with disturbances).

6.5 Summary of the results

There were carried out several identification processes with different settings in this chapter. Now, we give a brief summary to the attained results.

The best results were obtained when our assumed grey-box model perfectly matched up with the data generator (R1C0 case) and no disturbances affected the zones. Although, the results are impressive, this is rather an ideal case mostly far away from real conditions. Much more important is the R2C1 case when the grey-box is just approximation of a real more complex system. An identification of the building is then a true challenge and verifies a robustness of the whole algorithm.

As for the step sizes, the fixed step size rule is generally a bit faster and behaves more "aggressive" but its step must be smaller than the initial one in the variable step size rule and also is unable to recover even from a little increase of the consistency constraint residual at the beginning of the global algorithm.

The results for the LSE and MLE method are quite similar, though the LSE needs mostly less time to reach desired precision threshold of the global problem. On the other hand, the MLE method performs often less global iterations compared to the LSE.

Chapter 7
Conclusion

7.1 Thermal model of a building

The main goal of this thesis was to design an algorithm for the distributed building identification. That is an algorithm which would identify a building in the distributed manner. Before designing and implementing the algorithm itself, we achieved a smaller goal which was to create a proper simulation model of a building. There were created two of them: one with higher fidelity (R2C1) suitable for simulation purposes and the second simpler (R1C0) rather being a grey-box template, though used as a simulator and data generator as well.

There was also designed a simple discretized PI closed-loop heating control, since identification data acquisition is usually executed while closed-loop control is being applied due to safe issues.

7.2 Distributed identification

After we presented important theoretical parts concerning duality in optimization and decomposition methods, the next goal (the greatest of this thesis actually) was to think up and implement an algorithm for distributed identification. This algorithm is based on the dual decomposition method that has several advantages compared to the primal decomposition. At first, local sensitivities to global optimization variables (= prices) can be expressed easily straight away, since they are directly equal to optimized public variables. At second, for larger buildings almost all zone parameters are public (except for zone capacity). In case of the primal decomposition, this would mean a significant increase of global iterations (i.e. increase of communication among agents) because the public variables are fixed at local problems.

Therefore, we decided to use the dual decomposition method. It splits the overall building identification problem into local ones in each zone. The next step was thus to formulate a local identification problem. We utilized our prepared grey-box model R1C0 and expressed the problem as a (Nonlinear) Least Squares or Maximum Likelihood Estimation. The local problem was solved using a gradient method. In order not to rely only on a numerically approximated gradient provided by the `fmincon` solver, we implemented our own computation of the analytical gradient. The identification process took roughly the same time for both gradients, even though sometimes the

analytical gradient outperformed the numerical one by several less iterations. Thinking the numerical gradient computation is highly optimized, we can consider this as a good result.

After implementing the whole algorithm, we tested its function on a benchmark 4-zone building model. Firstly, the test was carried out on the R1C0 building model to demonstrate and verify a proper function of the algorithm. In this ideal case the goodness of fit reached about 96 %. Secondly, we used the R2C1 model as a data generator and again successfully identified the building even when it was burdened with disturbances simulating an occupancy. The goodness of fit achieved mostly about 80 % in this case.

7.3 Future development

Now, we propose a possible future development. If our proposed algorithm is to be used in an embedded device, the performance could be further increased by following enhancements. Firstly, a faster gradient computation would shorten a time needed for local optimization. Another suggested enhancement relates to the global algorithm. Application of some advanced step size rule as Nesterov's would most likely also decrease an overall computational time.

The last possible enhancement we propose is probably the most challenging. The identification in our algorithm is now being performed as batch processing of the entire measured data which is quite demanding. The major breakthrough might be reached if the data were processed iteratively with every new measurement and estimation of parameters would just have been updated.

Bibliography

[1] E Toffoli G Baldan and G Albertin L Schenato. Thermodynamic Identification of Buildings using Wireless Sensor Networks. 2004.

[2] Francesco Scotton. Physics-based modeling and identification for HVAC systems. (August):1404–1409, 2013.

[3] Clarence Agbi, Zhen Song, and Bruce Krogh. Parameter identifiability for multi-zone building models. *Proceedings of the IEEE Conference on Decision and Control*, pages 6951–6956, 2012.

[4] Klaus Kaae (Dtu) Andersen, Henrik (Dtu) Madsen, and Lars H. (Risø) Hansen. Modelling the heat dynamics of a building using stochastic differential equations. *Energy and Buildings*, 31(1):13–24, 2000.

[5] Siyu Wu and Jian Qiao Sun. A physics-based linear parametric model of room temperature in office buildings. *Building and Environment*, 50:1–9, 2012.

[6] Clarence Agbi and Bruce Krogh. Decentralized identification of building models. *Proceedings of the American Control Conference*, pages 1070–1075, 2014.

[7] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*, volume 25. Cambridge University Press, 2004.

[8] Sikandar Samar, Stephen Boyd, and Dimitry Gorinevsky. Distributed estimation via dual decomposition. *Proc. European Control Conference*, pages 1511–1516, 2007.

[9] Petr Endel. Distributed Predictive Control. *Control Systems, ...*, 2012.

[10] Lennart Ljung. *System Identification (2Nd Ed.): Theory for the User*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.

[11] K A N Terelius Master and Degree Project Stockholm. Distributed Multi-Agent Optimization via Dual Decomposition. 2010.

[12] Jiří Řehoř and Vladimír Havlena. Maximum likelihood estimation of LTI continuous-time grey-box models. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 19:3739–3744, 2014.

[13] Y Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Kluwer Academic Publishers, 2004.

Appendix A

Contents of CD attached

The CD attached to this thesis contains following directories:

- `thesis_text` - contains a pdf file of this thesis,
- `ml_codes` - contains MATLAB m-files and SIMULINK *.mdl models.