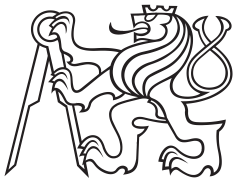
Diploma Thesis



Czech Technical University in Prague

Faculty of Electrical Engineering
Department of Cybernetics

Distributed Predictive Control of Buildings

Bc. Jan Hauser

Supervisor: Ing. Jiří Dostál
Field of study: Cybernetics and Robotics
Subfield: Robotics
May 2017

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

DIPLOMA THESIS ASSIGNMENT

Student: Bc. Jan H a u s e r

Study programme: Cybernetics and Robotics

Specialisation: Robotics

Title of Diploma Thesis: Distributed Predictive Control of Buildings

Guidelines:

1. Study thermodynamics, modelling of buildings and model predictive control methods.
2. Develop a centralized model predictive control for a building heating problem.
3. Develop a distributed (coordinated) model predictive control for the same problem.
4. Test the aforementioned techniques using a given building simulator and compare their properties and performance.

Bibliography/Sources:

[1] Borrelli F., Bemporad A., and Morari M. - Predictive control for linear and hybrid systems – Cambridge University Press 2011.
[2] Samar S., Boyd. S. - Distributed Estimation via Dual Decomposition – EEC 2007.
[3] Endel P. – Distributed Predictive Control – CTU 2012.

Diploma Thesis Supervisor: Ing. Jiří Dostál

Valid until: the end of the winter semester of academic year 2017/2018

L.S.

prof. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, September 29, 2016

Acknowledgements

Foremost, I would like to express my sincere thanks to my supervisor Ing. Jiří Dostál for his excellent guidance, improvement proposals and helpful comments.
I also want to give thanks to numberless open source developers who indirectly made implementation of all my software components possible by sharing code and thoughts with the general public.
Finally, my great gratitude goes to my family and classmates for their support and endless patience.

Declaration

I hereby formally declare that I worked the presented thesis independently. I quoted all used resources (literature, projects, software etc.) in accord with Methodical instructions about ethical principles for writing academic thesis.

In Prague, 24. May 2017

Signature: .....

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, 24. May 2017

Podpis: .....

Abstract

The aim of this thesis is to introduce thermal principles of a building, apply these principles into a model, design a building distributed control and compare it with a centralised control approach.
The building model analysis is based on electric-thermal analogy and deals with the thermal influences in building. Concept of a building zone is introduced and algorithm for concatenation of these zones is defined in order to prepare building for centralized control design.
Furthermore, centralized model predictive control theory and examples with simulations are included for comparison with other control approaches. The main objective of this thesis is dedicated to distributed control. Decomposition structures are presented as a background for distributed mechanism. Development of distributed model predictive control algorithms involves resource allocation problem as well as local zone control design.
Above all, the distributed control algorithms are challenged against centralized control approach. Comparison of these algorithms is achieved using objective functions of each control optimalization problem.

Keywords: distributed predictive control, building model, dual decomposition

Supervisor: Ing. Jiří Dostál

Abstrakt

Tato diplomová práce se zabývá definicí tepelných vztahů v budově, následným sestavením modelu, návrhem distribuovaného řízení a v neposlední řadě jeho porovnáním s řízením centralizovaným.
Analýza modelu budovy je postavena na elektro-termální analogii a zahrnuje všechny hlavní tepelné operace v budově. Zároveň je představen pojem zóny budovy a na něj navazuje algoritmus spojování těchto zón. Rozdělení konceptu budovy do zón je základním principem pro umožnění realizace návrhu distribuovaného řízení.
Dále práce představuje centralizované prediktivní řízení spolu s příklady a simulacemi, které poskytují možnost srovnání. Hlavním cílem práce je ovšem návrh řízení distribuovaného. V této problematice jsou nejprve představeny dekompoziční metody a jejich užití. Samotný vývoj algoritmů distribuovaného prediktivního řízení přináší hned několik úskalí, která jsou rozebrána. Jedním z nich je například problém s přerozdělením zdrojového tepla mezi zóny budovy, dalším problémem je návrh lokálních zónových regulátorů.
V závěru práce přichází nejpodstatnější část srovnání navržených algoritmů na modelových situacích. K porovnání slouží tzv. ztrátová funkce, kterou počítá každý z navržených řídících algoritmů v rámci optimalizace.

Klíčová slova: distribuované prediktivní řízení, model budovy, duální dekompozice

Překlad názvu: Distribuované prediktivní řízení budov

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation of the Thesis

The main motivation of this thesis is to compare and implement various building control techniques. The thesis is mainly devoted to the centralized and distributed model predictive control techniques application on buildings. Furthermore, one of the points is also to specify and illustrate the advantages and disadvantages of these control techniques. Computational effort, for instance, is one of the biggest issues. The decomposition methods understanding is also important in order to guarantee distributed control optimality.

1.2 Contribution of the Thesis

At first, a modelling part is introduced with all the building zone types which are based on [1]. Both a simpler R1C0 and a more complex R2C1 building zone models are introduced and used in the main control part of this thesis. The building zone model describes one simple unit of a large scale building system. A building zones concatenation algorithm is defined to create building model based on these simple units.

A large scale system control techniques are compared and fully described in the control part. Centralized and distributed model predictive control of a building creates the core of this theses. Moreover, building thermal control is dependent on a heat resource, which is ordinaly limited. Resource allocation algorithms are presented in order to solve control optimization despite this limitation. Last but not least, the decomposition methods and their practical usage in distributed methods is presented. Of course, the example simulations of the mentioned issues creates also a very important part.

1.3 Outline of the Thesis

Chapter 1 is an introductory part of the thesis, where the model predictive control techniques are briefly described. Motivation and contribution are mentioned in this chapter as well.

Chapter 2 describes the modelling part of this thesis. The thermal model of building is derived. Also a general algorithm for building zones merging is defined in this chapter. Models defined in this chapter are used in all the following control parts.

Chapter 3 presents the control part of this thesis. At first, the theory of model predictive control and it's robust extension are presented. The simple one zone model control with it's optimization part is described. Furthermore, the centralized and decentralized MPC with applications are defined in this chapter. All these advanced control techniques uses the previously presented control and model mechanisms.

Chapter 4 shows the implementations of presented control algorithms and analyses their objective functions in order to compare them. The centralized MPC is an object of comparison as well as the exact iterative distributed MPC with resource allocation algorithm using Nash optimality and the robust distributed MPC with dual decomposition based resource allocation algorithm.

Chapter 5 is the summarization part of this thesis. The possible future work is also outlined here.

1.4 Model Predictive Control Techniques

Model predictive control is a popular control strategy for design of a high performance model-based process control systems because of its ability to handle multi-variable interactions, constraints on control (manipulated) inputs and system states, and optimization requirements in a systematic manner [16]. This performance is widely used as a comfortable method to handle engineering problems. This control design method is used in many variations. As the system scale increases the control techniques performance differs due to the communication space and computational complexity.

A centralized control solution, where all subsystems' interactions are considered, is a traditional option [8]. This approach has many advantages and disadvantages. As the word "centralized" prompts, the control approach handles the whole system from one central point, where the control action is computed. This approach produces the best result but at a high price. Unfortunately, there are also many inconveniences like computational effort or higher risk of failure. Society dependency on large-scale systems grows and the centralized approach is gradually evaluated as an inappropriate method to handle these systems. A very good example of such a systems is a road-traffic system, an electricity network or also a building system.

During passing from the centralized to decentralized control approach the scale of the systems is fragmented. A system is divided into the separate subsystems. There is no information shared between the subsystems, what also comes with a lot of complications. The main complications are considered when the large interactions appears and the system stability is at risk. There is also a poor systemwide control performance if the subsystems interact significantly [18].

Here the emphasize on distributed approach comes into account. The system is again fragmented into small subsystems, whose control is assigned to a certain number of controllers. The subsystems and controllers creates agents. The controllers share information important to handle subsystems interactions. The subsystems are dynamically coupled. The distributed technique also offers very good performance in the robustness and fault-tolerance area. Computational requirements decrease as the problem is broken into smaller parts. If one of the agents records an error, it does not mean the defect for the other agents. In a centralized manner, if one system's section fails, whole the system is affected. Nevertheless, there are also some drawbacks that has to be taken into account. As denoted above, the centralized technique offers better performance. Performance loss happens due to the system fragmentation.

Off course, the role of a coordinator cannot be also ommited. The coordinator represents an algorithm or a higher level structure sometimes also called master problem. This higher level structure can be designed in many fractions through the subsystems, or it can be just one represented by one coordinator.

Chapter 2

Building Thermal Model

There are many approaches to thermal modelling of a building. A thermal-electric analogy is used to visualize the termal model of a building in this thesis. The full physical description of the technique of thermal (RC) circuits is described in [1] and here it is just briefly reminded.

A one floor building model with a couple of neighbouring rooms will be taken as a reference building model through the thesis.

2.1 Physical Foundations

Mainly it is the Fourier's law (2.1) describing the heat conduction

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

where $\mathbf{q}[\mathrm{W} \cdot \mathrm{m}^{-2}]$ is a local heat flux density , $k[\mathrm{W} \cdot \mathrm{m}^{-1} \cdot \mathrm{K}^{-1}]$ is a thermal conductivity and $\nabla T[\mathrm{K} \cdot \mathrm{m}^{-1}]$ is a temperature gradient . This law can be rewritten in infinitesimal limit

$$\dot{Q} = U \cdot A \cdot \triangle T,$$

where $\dot{Q}[\mathrm{W}]$ is a heat flow rate, $A[m^2]$ is a cross-sectional surface area , $\triangle T[\mathrm{K}]$ is a temperature difference between the ends and $U[\mathrm{W} \cdot \mathrm{m}^{-2} \cdot \mathrm{K}^{-1}]$ is a heat conductance , which is reciprocal to a heat resistance R $[\mathrm{m}^2 \cdot \mathrm{K} \cdot \mathrm{W}]$.

Next very important equation is the definition of the object's heat capacity

$$C\dot{T} = \dot{Q},$$

where $C[\mathrm{W} \cdot \mathrm{K}^{-1}]$ stands for the total thermal capacity.

2.2 One-Zone Building Model (R1C0)

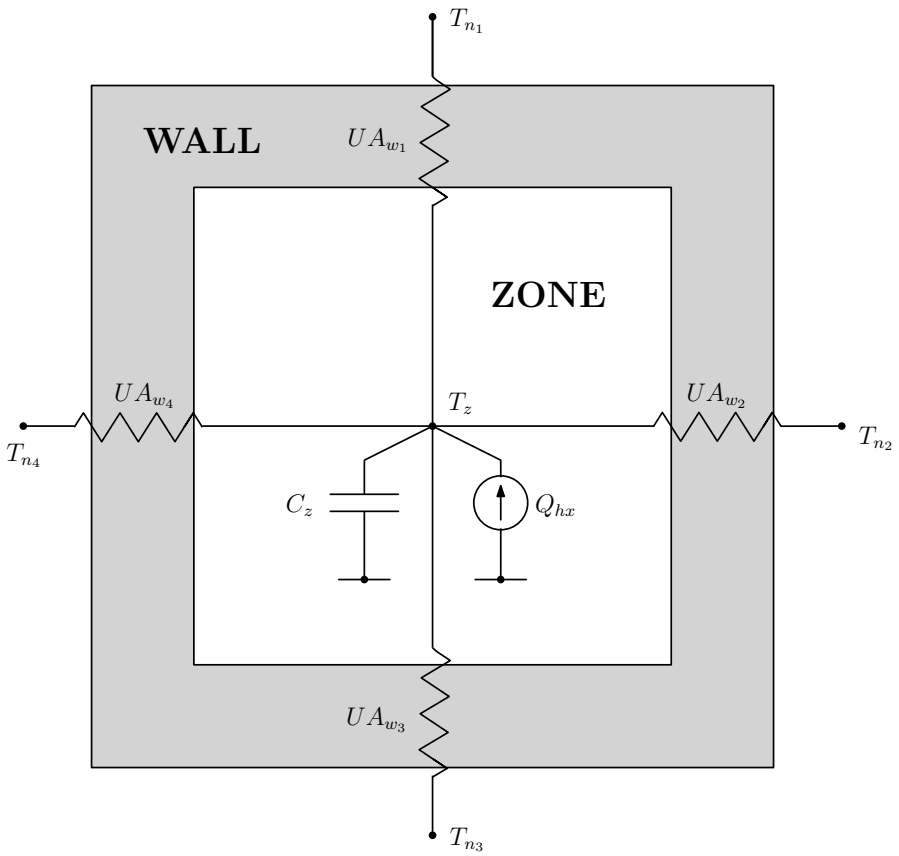


Figure 2.1: Scheme of the R1C0 model of one-zone. The R1C0 stands for a simplified model of wall, which is described by the resistance and capacitance. This example demonstrates a zone with four neighbouring zones. [1]

The one-zone R1C0 thermal model with n neighbouring zones is taken from [1] and its state space description is as follows

$$\begin{aligned} \dot{T} &= AT + \mathbf{B}\mathbf{u} \\ y &= CT + \mathbf{D}\mathbf{u}, \end{aligned} \tag{2.2}$$

where $A \in \mathbb{R}$, $\mathbf{B} \in \mathbb{R}^{1\times n+1}$, $C \in \mathbb{R}$ and $\mathbf{D} \in \mathbb{R}^{1\times n+1}$ are given as

$$A = -\frac{1}{C_z}\sum_{i=1}^{n} h_i, \quad \mathbf{B} = \begin{bmatrix} \frac{1}{C_z} & \frac{h_1}{C_z} & \dots & \frac{h_n}{C_z} \end{bmatrix} \tag{2.3}$$

$$C = 1, \qquad \mathbf{D} = \mathbf{0} \tag{2.4}$$

and

$$h_i = U_i \cdot A_{w_i},$$

where C_z is the zone thermal capacitance, h_i is the total i-th wall conductance, $U_i A_{w_i}$ is the i-th wall conductance multiplied by the wall's surface area.

The state $T \in \mathbb{R}$ and the inputs $\mathbf{u} \in \mathbb{R}^{n+1\times 1}$ are defined as follows

$$T = T_z, \quad \mathbf{u} = \begin{bmatrix} Q_{hx} \\ T_{n_1} \\ \vdots \\ T_{n_n} \end{bmatrix}$$

The state T_z stands for the zone temperature, inputs $T_{n_{1\dots n}}$ are the neighbouring zones temperatures or ambient (outside) temperatures T_{amb} and the input Q_{hx} is a heat flow from a heat exchanger.

2.3 One-Zone Building Model (R2C1)

The one-zone R2C1 model with n neighbouring zones is briefly described by the following state space model:

$$\begin{aligned} \dot{\mathbf{T}} &= \mathbf{AT} + \mathbf{Bu} \\ y &= \mathbf{CT} + \mathbf{Du}, \end{aligned}$$

where $\mathbf{A} \in \mathbb{R}^{p\times p}$, $\mathbf{B} \in \mathbb{R}^{p\times p}$, $\mathbf{C} \in \mathbb{R}^{1\times p}$ and $\mathbf{D} \in \mathbb{R}^{p\times p}$, $p = n + 1$, are given as

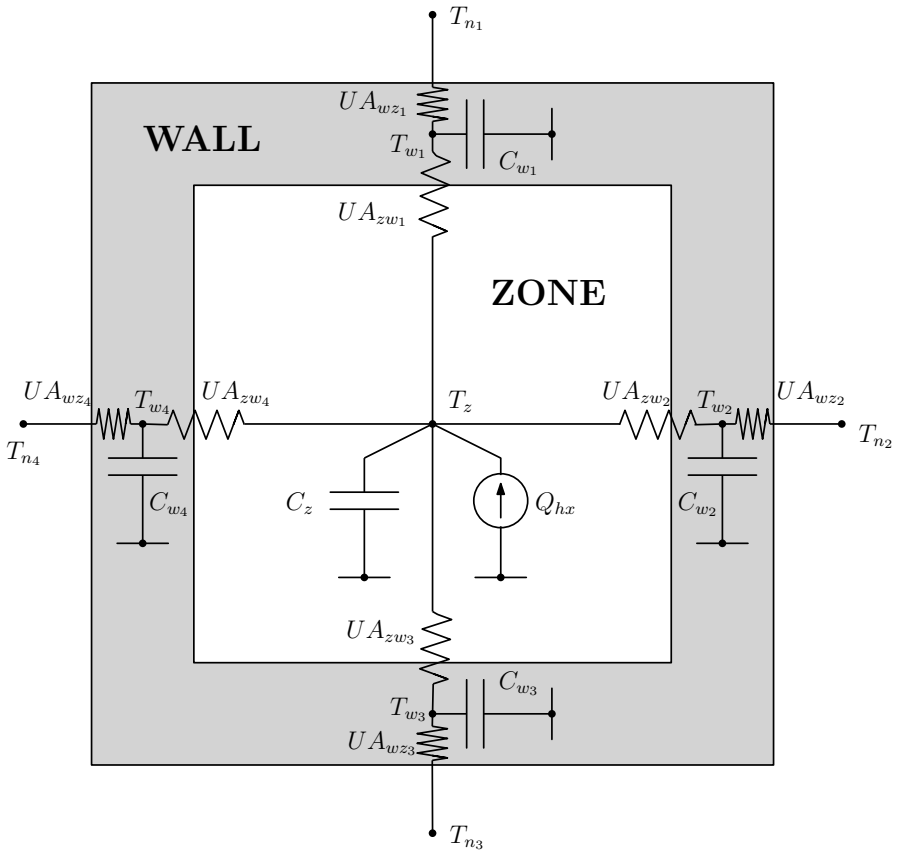


Figure 2.2: Scheme of the one-zone R2C1 model. This model is a more complex version of the previously presented R1C0 model. [1]

$$\mathbf{A} = \begin{bmatrix} -\sum_{i=1}^{4} \frac{h_{zw_i}}{C_z} & \frac{h_{zw_1}}{C_z} & \cdots & \frac{h_{zw_n}}{C_z} \\ \frac{h_{zw_1}}{C_{w_1}} & -\frac{h_{zw_1}+h_{wz_1}}{C_{w_2}} & \mathbf{0}^{\mathrm{T}} & 0 \\ \vdots & \mathbf{0} & \ddots & \vdots \\ \frac{h_{zw_n}}{C_{w_n}} & 0 & \cdots & -\frac{h_{zw_n}+h_{wz_n}}{C_{w_n}} \end{bmatrix} \tag{2.5}$$

(2.6)

$$\mathbf{B} = \begin{bmatrix} \frac{1}{C_z} & 0 & \cdots & 0 \\ 0 & \frac{h_{wz_1}}{C_{w_1}} & \mathbf{0}^{\mathrm{T}} & 0 \\ \vdots & \mathbf{0} & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{h_{wz_n}}{C_{w_n}} \end{bmatrix} \tag{2.7}$$

(2.8)

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}, \quad \mathbf{D} = 0$$

and

$$\begin{aligned} h_{zw_i} &= U_i \cdot A_{zw_i} \\ h_{wz_i} &= U_i \cdot A_{wz_i}, \end{aligned}$$

where C_{w_i} is a wall capacitance. h_{zw_i} (or h_{wz_i}) parameters are the zone-wall (or wall-zone) crossing resistances, $U_i A_{zw_i}$ (or $U_i A_{wz_i}$) is the i-th zone-wall (or wall-zone) conductance multiplied by the wall's surface area. Meaning of a zone-wall (wall-zone respectivelly) indication is that the wall is divided into layers, see Fig. 2.2.

The state vector $\mathbf{T} \in \mathbb{R}^{p\times 1}$, the inputs $\mathbf{u} \in \mathbb{R}^{p\times 1}$ and the output $y \in \mathbb{R}$ are defined as follows

$$\mathbf{T} = \begin{bmatrix} T_z \\ T_{w_1} \\ \vdots \\ T_{w_n} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} Q_{hx} \\ T_{n_1} \\ \vdots \\ T_{n_n} \end{bmatrix}, \quad y = T_z,$$

where T_z is the zone temperature and $T_{w_{1\dots n}}$ are the zone walls temperatures.

2.4 Four-Zone Building Model (R1C0)

A four-zone model is described in [1] and the motivation of this section is to describe such model in a state space form using the system matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ from the one-zone R1C0 model.

Four simple zones from sec. 2.2 are connected here to create one global model, which is later used for centralised control approach.

Consider the state matrix $\mathbf{A}$ and the four-zone model scheme from Fig. 2.3. The $\mathbf{A}^{big}$ matrix contains these $\mathbf{A}_{1\dots 4}$ matrices on the diagonal. Parameters describing the shared parameters of walls are transfered from matrices $\mathbf{B}$ to proper positions in matrix $\mathbf{A}^{big}$ to match the states from the state vector $\mathbf{T}$.

A general algorithm is described in sec. 2.6.

The zone adjacency matrix is created here. Row and column numbers of this matrix are equal to the zone numbers.

$$\mathcal{A} = \{a_{i,j}\}, \; a_{i,j} = \begin{cases} 1 & \text{zone } i \text{ and zone } j \text{ share common wall} \\ 0 & \text{otherwise} \end{cases}, \; i,j = 1,\dots m,$$

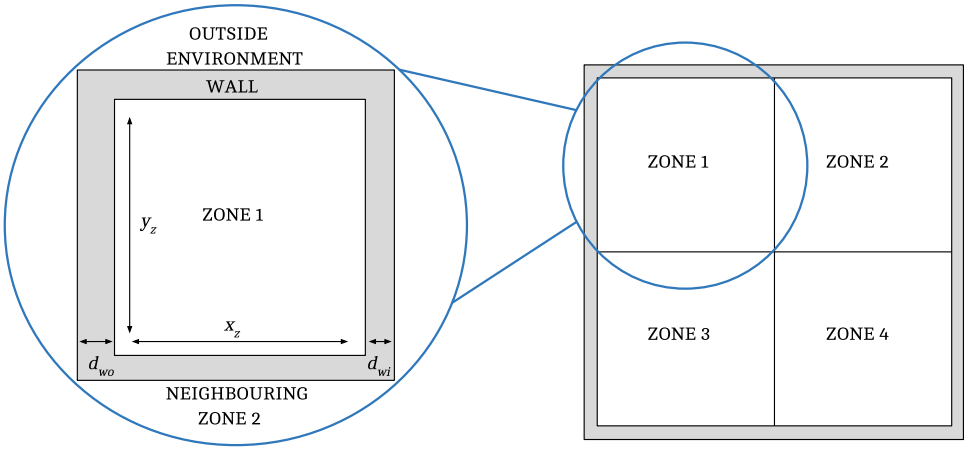


Figure 2.3: Four-zone scheme for thermal model of building.

where m is the total number of zones. More specifically for the four-zone model

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}. \tag{2.9}$$

The input, state, output matrices $\mathbf{A} \in \mathbb{R}^{4\times4}$, $\mathbf{B} \in \mathbb{R}^{4\times5}$, $\mathbf{C} \in \mathbb{R}^{4\times4}$ and matrix $\mathbf{D} \in \mathbb{R}^{4\times5}$ are defined as follows

$$\mathbf{A}^{big} = \begin{bmatrix} -\frac{h_1}{C_{z_1}} & \frac{h_{1,2}}{C_{z_1}} & \frac{h_{1,3}}{C_{z_1}} & 0 \\ \frac{h_{2,4}}{C_{z_2}} & -\frac{h_2}{C_{z_2}} & 0 & \frac{h_{2,3}}{C_{z_2}} \\ \frac{h_{3,1}}{C_{z_3}} & 0 & -\frac{h_3}{C_{z_3}} & \frac{h_{3,2}}{C_{z_1}} \\ 0 & \frac{h_{4,1}}{C_{z_4}} & \frac{h_{4,4}}{C_{z_4}} & -\frac{h_4}{C_{z_4}} \end{bmatrix}$$

$$\mathbf{B}^{big} = \begin{bmatrix} \frac{1}{C_{z_1}} & 0 & \dots & 0 & \frac{h_{1,1}+h_{1,4}}{C_{z_1}} \\ 0 & \frac{1}{C_{z_2}} & \ddots & \vdots & \frac{h_{2,1}+h_{2,2}}{C_{z_2}} \\ \vdots & \ddots & \frac{1}{C_{z_3}} & 0 & \vdots \\ 0 & \dots & 0 & \frac{1}{C_{z_4}} & \frac{h_{4,2}+h_{4,3}}{C_{z_4}} \end{bmatrix}$$

$$\mathbf{C}^{big} = \mathbf{I}_4, \quad \mathbf{D}^{big} = \mathbf{0}$$

and

$$\begin{aligned} h_{i,j} &= U_{i,j} \cdot A_{w_{i,j}} \\ h_i &= \sum_{j=1}^{4} h_{i,j}, \end{aligned}$$

where $h_{i,j}$ is the total conductance of j-th wall, i-th zone.

And for the state vector $\mathbf{T} \in \mathbb{R}^{4\times 1}$ and the inputs $\mathbf{u} \in \mathbb{R}^{5\times 1}$ stands

$$\begin{aligned} \mathbf{T} &= \begin{bmatrix} T_{z_1} \\ T_{z_2} \\ T_{z_3} \\ T_{z_4} \end{bmatrix} \\ \mathbf{u} &= \begin{bmatrix} Q_{hx_1} & Q_{hx_2} & Q_{hx_3} & Q_{hx_4} & T_{amb} \end{bmatrix}^{\mathrm{T}}. \end{aligned} \tag{2.10}$$

The symmetries in matrices are notable. Both the matrices $\mathbf{A}^{big}, \mathbf{B}^{big}$ have the zones sequentially ordered on their diagonals in the zone numbered order.

2.5 Four-Zone Building Model (R2C1)

Now consider a more complicated situation, when four zones of the R2C1 thermal model are connected together.

The zone adjency matrix $\mathcal{A}$ is equal to (2.9).

The wall adjacency matrix $\mathcal{A}^{wall}$ is defined here.

$$\mathcal{A}^{wall} = \{a_{i,j}\}, \; a_{i,j} = \begin{cases} 1 & \text{wall } i \text{ and wall } j \text{ interact} \\ 0 & \text{otherwise} \end{cases}, \; i,j = 1,\ldots,s$$

where s equals to the size of matrix $\mathbf{A}^{big}$ defined below. The value of s can be also represented as the sum of total number of zones and total number of neighbours for each zone. This matrix is constructed by taking into account the walls which are shared between the zones. Simply said, it points out the zones common walls and walls parameters positions in the matrix $\mathbf{A}^{big}$. The $\mathcal{A}^{wall}$ elements positions are strictly reflected to the matrix $\mathbf{A}^{big}$, where 1 stands for the position containing a wall parameter and 0 describes the position without a wall parameter. See the full explanation in remark 2.1.

Now it is easier to describe the state space model matrices $\mathbf{A}^{big} \in \mathbb{R}^{20\times 20}$, $\mathbf{B}^{big} \in \mathbb{R}^{20\times 5}$, $\mathbf{C}^{big} \in \mathbb{R}^{4\times 20}$ and $\mathbf{D}^{big} \in \mathbb{R}^{4\times 5}$.

The diagonal parts of matrix $\mathbf{A}^{big}$ are matrices $\mathbf{A}_{1\ldots 4}$ from (2.5). It follows the process presented in previous section and generally defined in sec. 2.6.

The shared wall related parts of the matrix $\mathbf{A}^{big}$ are generated using the wall adjacency matrix $\mathcal{A}^{wall}$ as follows

$$\begin{aligned}
\mathbf{A}^{big}(3,6) &= \frac{h_{wz_{1,2}}}{C_{w_{1,2}}} = \frac{h_{zw_{2,4}}}{C_{w_{2,4}}};\ \mathbf{A}^{big}(4,11) = \frac{h_{wz_{1,3}}}{C_{w_{1,3}}} = \frac{h_{zw_{3,1}}}{C_{w_{3,1}}} \\
\mathbf{A}^{big}(10,1) &= \frac{h_{wz_{2,4}}}{C_{w_{2,4}}} = \frac{h_{zw_{1,2}}}{C_{w_{1,2}}};\ \mathbf{A}^{big}(9,16) = \frac{h_{wz_{2,3}}}{C_{w_{2,3}}} = \frac{h_{zw_{4,1}}}{C_{w_{4,1}}} \\
&\text{etc.}
\end{aligned}$$

and rest of the state space describtion is defined as follows,

$$\mathbf{B}^{big}(1:5,1:5) = \begin{bmatrix} \frac{1}{C_{z_1}} & 0 & 0 & 0 & 0 \\ 0 & \cdots & \cdots & 0 & \frac{h_{wz_{1,1}}}{C_{w_{1,1}}} \\ \vdots & \ddots & & \vdots & 0 \\ \vdots & & \ddots & \vdots & 0 \\ 0 & \cdots & \cdots & 0 & \frac{h_{wz_{1,4}}}{C_{w_{1,4}}} \end{bmatrix}$$

$$\mathbf{B}^{\mathbf{big}}(6:10,1:5) = \begin{bmatrix} 0 & \frac{1}{C_{z_2}} & 0 & 0 & 0 \\ 0 & \cdots & \cdots & 0 & \frac{h_{wz_{2,1}}}{C_{w_{2,1}}} \\ \vdots & \ddots & & \vdots & \frac{h_{wz_{2,2}}}{C_{w_{2,2}}} \\ \vdots & & \ddots & \vdots & 0 \\ 0 & \cdots & \cdots & 0 & 0 \end{bmatrix}$$

$$\text{etc.}$$

$$\begin{aligned}
\mathbf{C}^{big}(k,l) &= 1;\ \mathbf{D}^{big} = \mathbf{0} \\
&\quad k = 1,2,3,4;\ l = 5k-4
\end{aligned}$$

$$\begin{aligned}
h_{zw_{i,j}} &= U_{i,j} \cdot A_{zw_{i,j}} \\
h_{wz_{i,j}} &= U_{i,j} \cdot A_{wz_{i,j}},
\end{aligned}$$

where the $\mathbf{A}^{big}$ indexes identification is fully described in remark 2.1.

The state vector $\mathbf{T} \in \mathbb{R}^{20\times 1}$ is described below. The input vector $\mathbf{u} \in \mathbb{R}^{5\times 1}$ is cut of the non necessary neighbouring zones temperatures which are represented in vector $\mathbf{u}$ by environment temperature T_{amb} and are replaced in the state vector $\mathbf{T}$ by the zone temperatures. Finally, the output vector $\mathbf{y} \in \mathbb{R}^{4\times 1}$ includes just the zones' temperatures.

$$\begin{aligned} \mathbf{T} &= \begin{bmatrix} T_{z_1} & T_{w_{1,1}} & T_{w_{1,2}} & T_{w_{1,3}} & T_{w_{1,4}} & T_{z_2} & \dots & T_{w_{4,4}} \end{bmatrix}^{\mathrm{T}} \\ \mathbf{u} &= \begin{bmatrix} Q_{hx_1} & Q_{hx_2} & Q_{hx_3} & Q_{hx_4} & T_{amb} \end{bmatrix}^{\mathrm{T}} \\ \mathbf{y} &= \begin{bmatrix} T_{z_1} & T_{z_2} & T_{z_3} & T_{z_4} \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{2.11}$$

Remark 2.1. The first 5×20 submatrix of $\mathcal{A}^{wall}$ demonstrates the Z_1 zone relations. The first zone interacts with the T_{z_2}, T_{z_3} states (temperatures of zones Z_2 and Z_3 respectivelly) according to the four zones scheme Fig. 2.3. These zones are the neighboures of zone Z_1 according to the adjency matrix (2.9). The wall shared between the zones Z_1 and Z_2 is called $W_{1,2}$. The wall placed bethween the zones Z_1 and Z_3 is $W_{1,3}$.

The states T_{z_2}, T_{z_3} are on the 6th and 11th rows in the state matrix $\mathbf{T}$ (which is multiplied by the 6th and 11th columns in the matrix $\mathcal{A}^{wall}$ in the operation $\mathcal{A}^{wall} \cdot \mathbf{T}$). The walls $W_{1,2}$, $W_{1,3}$ temperatures $T_{w_{1,2}}$, $T_{w_{1,3}}$ are on the 3rd and 4th rows in the state matrix $\mathbf{T}$. Therefore the wall parameters positions for this matrix equals to the positions $\mathcal{A}^{wall}(3,6)$ and $\mathcal{A}^{wall}(4,11)$. The relations parameters will be placed exactly on these positions in the final $\mathbf{A}^{big}$ matrix.

$$\mathcal{A}^{wall} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1}^{1\times p} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1}^{1\times p} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}^{1\times p} \\ \mathbf{1}^{1\times p} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1}^{1\times p} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}^{1\times p} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1}^{1\times p} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1}^{1\times p} & \mathbf{0} \end{bmatrix}, \tag{2.12}$$

where

$$\mathbf{1}^{1\times p} = \begin{bmatrix} 1 & \mathbf{0}^{1\times p-1} \end{bmatrix}.$$

2.6 Global Building Model Concatenation Algorithm

This section presents the general algorithm for constructing a global state space model from the local zone models and their interconnections. The algorithm was formulated by generalization of the procedures presented in the previous modelling sections.

It is necessary to define a concept of an extended adjacency matrix $\mathcal{A}^{ext}$ because of the generalization.

$$\mathcal{A}^{ext} = \{a_{i,j}\}, \; a_{i,j} = \begin{cases} W_{i,j} & \text{zone } i \text{ and zone } j \text{ interact} \\ 0 & \text{otherwise} \end{cases}, \; i,j = 1,\ldots,m,$$

where m is the total number of zones. This matrix is basically the adjacency matrix like in (2.9) but the ones are replaced with the wall identifiers $W_{i,j}$. There are just two possible values: $\{0,1\}$ in the original adjacency matrix. The extended adjacency matrix takes these values: $\{0, W_{i,j}\}$, where $i,j \in \mathbb{Z}$ are the appropriate indexes. See remark 2.3.

It is also necessary to create the vector of states as in (2.11). See remark 2.2.

Algorithm 1 Global building model concatenation algorithm.

1. Generate the extended adjacency matrix $\mathcal{A}^{ext}$ based on the model scheme.
2. For all zones Z_i, $i = 1, \ldots, m$ identify the one-zone model matrix $\mathbf{A}_i, \mathbf{B}_i$, where m equals to the number of zones.
3. Place matrices $\mathbf{A}_i$ in the index order on the diagonal of matrix $\mathbf{A}^{big}$ from top to bottom. See remark 2.4.
4. For each matrix $\mathbf{B}_i$.
   a. Go through the matrix $\mathcal{A}^{ext}$ and for each non-zero element $W_{i,j}$ get a column from matrix $\mathbf{B}_i$ which belongs to the T_{n_j} state. The index j is equal to the neighbour with temperature T_{n_j} which is connected to zone Z_i through the wall identified $W_{i,j}$. Name the selected column as '**c**' for later reference. See remark 2.5.
   b. Erase **c** from matrix $\mathbf{B}_i$.
   c. Select a column from $\mathbf{A}^{big}$ which corresponds to the state T_{z_k}, where k is the column of $W_{i,j}$ in the matrix $\mathcal{A}_{ext}$. Replace this column with the column **c**. Remark 2.6
   d. Place the matrix $\mathbf{B}_i$,without the erased columns, to the diagonal of matrix $\mathbf{B}^{big}$. See remark 2.7.
5. Generation of matrix $\mathbf{C}^{big}$ depends on the selected outputs, $\mathbf{D}^{big} = 0$.

Remark 2.2. The vector of states is created by perturbation of the zone models in the following order

$$\mathbf{T}_i = \begin{bmatrix} T_{z_i} \\ T_{w_{i,1}} \\ T_{w_{i,2}} \\ \vdots \\ T_{w_{i,r}} \end{bmatrix}; \mathbf{T} = \begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \\ \cdots \\ \mathbf{T}_m \end{bmatrix},$$

where $i = 1, ..., m$, m is equal to the number of zones and r is the number of walls shared with neighbouring zones for zone Z_i.

Remark 2.3. The extended adjacency matrix creation for the four zones R2C1 model is presented in this example. The non-oriented graph representing the four-zone R2C1 model in sec. 2.4 was defined as follows.

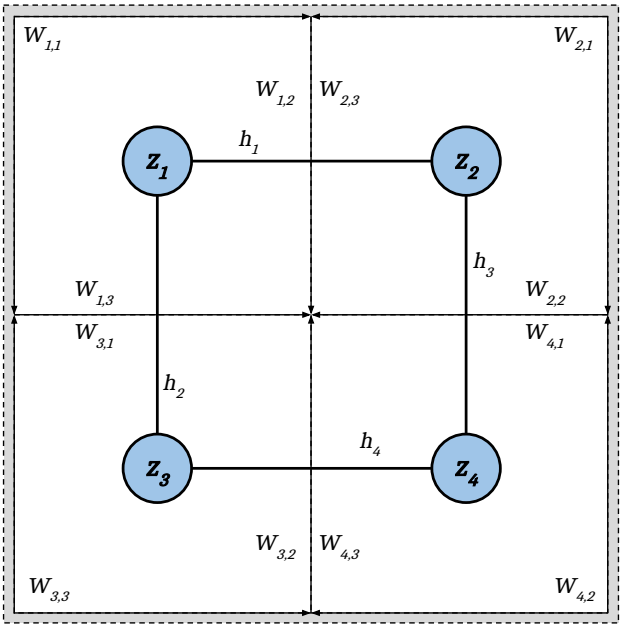


Figure 2.4: The four-zone R2C1 model non-oriented graph. This graph serves to simplify the extended adjacency matrix $\mathcal{A}^{ext}$ (2.13) definition. The wall indexes were redefined so that each zone wall belongs to different neighbouring zone, outside environment respectivelly. This fact implies that the wall identifiers $W_{1,1}$, $W_{2,1}$, $W_{3,3}$, $W_{4,2}$ stands for the whole wall areas neighbouring with the outside environment.

The extended adjacency matrix example based on Fig. 2.4 is presented below.

$$\mathcal{A}^{ext} = \begin{bmatrix} 0 & W_{1,2} & W_{1,3} & 0 \\ W_{2,3} & 0 & 0 & W_{2,2} \\ W_{3,1} & 0 & 0 & W_{3,2} \\ 0 & W_{4,1} & W_{4,3} & 0 \end{bmatrix} \tag{2.13}$$

This matrix represents wall-zone relations. For example, the zone Z_1 has two neighbours, zone Z_2 and zone Z_3. It shares with zone Z_2 the wall $W_{1,2}$ and with zone Z_3 the wall $W_{1,3}$.

Remark 2.4. The $\mathbf{A}_i$ placement into the matrix $\mathbf{A}^{big}$ is presented in this remark. Let's have a matrix $\mathbf{A}_1$ of size 4×4 and matrix $\mathbf{A}_2$ of size 3×3. Matrix $\mathbf{A}^{big}$ could be created using even more one-zone matrices but for purpose of this remark consider just these two.

Place the matrices $\mathbf{A}_1, \mathbf{A}_2$ into the diagonal of the matrix $\mathbf{A}_{big}$. The final matrix $\mathbf{A}^{big}$ is filled, before taking the next steps of the Alg. 1, see (2.14).

$$\mathbf{A}^{big} = \begin{bmatrix} A_{1_{11}} & \dots & A_{1_{14}} & & & & \\ \vdots & \ddots & \vdots & & \mathbf{0} & & \\ A_{1_{41}} & \dots & A_{1_{44}} & & & & \\ & & & A_{2_{11}} & A_{2_{12}} & A_{2_{13}} & \\ & \mathbf{0} & & A_{2_{21}} & A_{2_{22}} & A_{2_{23}} & \\ & & & A_{2_{31}} & A_{2_{32}} & A_{2_{33}} & \\ & & & & & & \ddots \end{bmatrix} \tag{2.14}$$

Remark 2.5. Let's have a wall identifiers $W_{1,2}$ from (2.13). According to the Alg. 1, look for the column in $\mathbf{B}_i$ which belongs to the T_{n_2} state.

Selecting such column is described below. The blue color marked column belongs to the blue color marked state.

$$\begin{bmatrix} \frac{1}{C_z} & 0 & 0 & 0 \\ 0 & \frac{h_{wz_1}}{C_{w_1}} & 0 & 0 \\ 0 & 0 & \frac{h_{wz_2}}{C_{w_2}} & 0 \\ 0 & 0 & 0 & \frac{h_{wz_3}}{C_{w_2}} \end{bmatrix} \cdot \begin{bmatrix} Q_{hx} \\ T_{n_1} \\ T_{n_2} \\ T_{n_3} \end{bmatrix}$$

The selected column equals to $\mathbf{c} = \begin{bmatrix} 0 & 0 & \frac{h_{wz_2}}{C_{w_2}} & 0 \end{bmatrix}^{\mathrm{T}}$.

Remark 2.6. Let's have the matrix $\mathbf{A}^{big}$ and $W_{1,2}$ from the previous remarks. According to the Alg. 1, look for the column which corresponds to the state T_{z_2}. This column will be replaced with column '$\mathbf{c}$' from previous step. The index k equals to 2 in this remark, because $W_{1,2}$ was taken from the second column of the matrix $\mathcal{A}^{ext}$.

The vector of states is according to Fig. 2.4

$$\mathbf{T} = \begin{bmatrix} T_{z_1} & T_{n_{1,1}} & T_{n_{1,2}} & T_{n_{1,3}} & T_{z_2} & \dots \end{bmatrix}^{\mathrm{T}}.$$

This implies that the column to be replaced equals to the fifth column in the matrix $\mathbf{A}^{big}$, because this column corresponds to the state T_{z_2}. The matrix $\mathbf{A}^{big}$ then looks as follows.

$$\mathbf{A}^{big} = \begin{bmatrix} A_{1_{11}} & \dots & A_{1_{14}} & c_1 & & & \\ \vdots & \ddots & \vdots & \vdots & & \mathbf{0} & \\ A_{1_{41}} & \dots & A_{1_{44}} & c_4 & & & \\ & & & A_{2_{11}} & A_{2_{12}} & A_{2_{13}} & \\ & \mathbf{0} & & A_{2_{21}} & A_{2_{22}} & A_{2_{23}} & \\ & & & A_{2_{31}} & A_{2_{32}} & A_{2_{33}} & \\ & & & & & & \ddots \end{bmatrix}$$

Remark 2.7. Placing of the non-square matrices to a diagonal is described here. Assume again just two matrices $\mathbf{B}_1 \in \mathbb{R}^{4\times3}$, $\mathbf{B}_2 \in \mathbb{R}^{3\times5}$ and the matrix $\mathbf{B}^{big}$ of temporarily unknown size. A filled matrix $\mathbf{B}^{big}$ is created by the direct sum of the matrices $\mathbf{B}_1, \mathbf{B}_2$

$$\begin{aligned}\mathbf{B}^{big} &= \mathbf{B}_1 \oplus \mathbf{B}_2 \\ &= \begin{bmatrix} B_{1_{11}} & B_{1_{12}} & B_{1_{13}} & & & & \\ \vdots & & \vdots & & & \mathbf{0} & \\ B_{1_{41}} & \dots & B_{1_{43}} & & & & \\ & & & B_{2_{11}} & B_{2_{12}} & \dots & B_{2_{15}} \\ & \mathbf{0} & & B_{2_{21}} & & \ddots & B_{2_{25}} \\ & & & B_{2_{31}} & \dots & \dots & B_{2_{35}} \end{bmatrix}\end{aligned}$$

and more generally for m matrices

$$\mathbf{B}^{big} = \bigoplus_{i=1}^{m} \mathbf{B}_i.$$

Chapter 3

Building Control

All the control algorithms presented in this chapter are based on model predictive control (MPC). This advanced optimalization control principle is used in this thesis in a several ways, the centralised and distributed ways of control, the decentralised approach is just mentioned. These MPC types are theoretically described and then also compared. Each of them has its advantages and disadvantages. For example the centralised control unit will always struggle with the high computational requirements, the distributed controller will always worse handle the subsystems interactions and so on.

The first section introduces the MPC principles as well as the robust MPC in sec. 3.1. The second sec. 3.2 presents the centralised building control method. The distributed methods of building control are presented in the third sec. 3.3.

3.1 One-Zone Control

A control of one building zone is described in this section. These one zone principles are later used in the whole building control. It is the first step to control each zone separately without considering the zone interactions.

This section also serves as an introduction to to the he model predictive control (MPC) and robust model predictive control. Then the single zone model controller design is described as well.

3.1.1 Model Predictive Control

Model predictive control is a control method based on an exact model and system states prediction. Future control steps are optimized using the predicted states in a prediction horizon. This control approach has one big advantage in comparison to other control

methods, it is capable to incorporate constraints and handle multi-variable interactions. [11]

MPC control law is computed in real time, what means that the optimization process to choose optimal input vector $\mathbf{u}(t)$ is repeated every time step. [12]

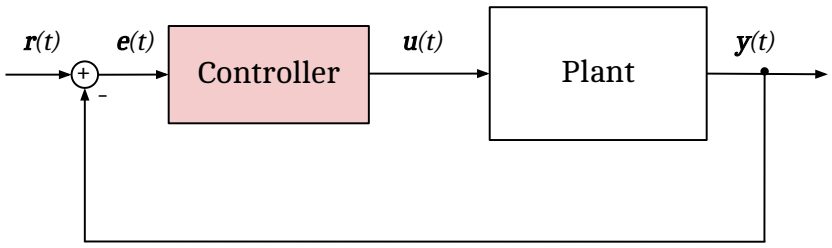


Figure 3.1: Classical feedback control design which solves dominant issues in frequency domain. The issue is for example the disturbance rejection.

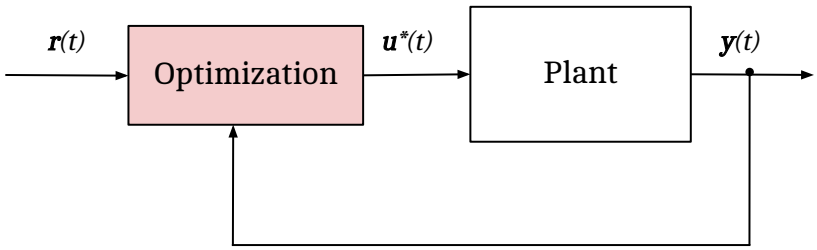


Figure 3.2: Model predictive control design. This control design solves the control issues mainly in the time domain.

The sections below describe the MPC design steps. Fig. 3.3 also shows all these steps and MPC principle itself. There are shown the predicted outputs $\mathbf{y}(k)$ which are created using optimized manipulted inputs $\mathbf{u}(k)$ in the figure.

System Definition

MPC is discrete-time control technique, which implies the discrete time state space model as follows

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_\mathrm{d}\mathbf{x}(k) + \mathbf{B}_\mathrm{d}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}_\mathrm{d}\mathbf{u}(k), \end{aligned}$$

where $\mathbf{A}_d, \mathbf{B}_\mathrm{d}, \mathbf{C}$ and $\mathbf{D}_\mathrm{d}$ are discrete state matrices. Process of discretization is done in this thesis using the zero-order hold method with appropriate sample time.

Constraints

All the physical systems has some kind of limitations (constraints). There are many types of them to be opperated, the limits on manipulated inputs, their rates of change or the limits on controlled output. An example of input constraints are actuator limits. Performance constraint example is the maximal overshoot or the maximal reference violation bounds. The temperature/pressure limit as the safety constraint is also worth mentioning.

The constraints can be cast into two cathegories, soft and hard constraints divided with respect on their inviolability. A hard constraint is boundary which cannot be broken despite the optimization problem infeasibility. On the other hand the soft constraint allows a violation, what is reflected directly to the optimization process by increasing of a floating cost. This is fully described in the following section, where the optimization problem is defined.

Optimization Problem

The MPC optimization problem definition is the most important part of this brief introduction to model predictive control. Simple theoretic situation is presented in this section. Let's define optimization problem with simple input soft constraint and output hard constraint.

The objective function (or cost function) is defined in a following way,

$$\begin{aligned} \mathbf{J}\left(\mathbf{x}(k), \mathbf{u}(k)\right) &= \mathbf{Q}\left\|\mathbf{C}\mathbf{x}(N) - \mathbf{s}_y(N)\right\| + \\ &\quad + \sum_{k=1}^{N-1} \mathbf{Q}\left\|\mathbf{C}\mathbf{x}(k) - \mathbf{s}_y(k)\right\| + \mathbf{R}\left\|\mathbf{u}(k)\right\|, \end{aligned} \tag{3.1}$$

where $\mathbf{Q} \geq 0, \mathbf{R} \geq 0$ are matrices of appropriate sizes, $\mathbf{x}(t)$ is a a state vector, $\mathbf{u}(t)$ is an input vector and $\mathbf{s}_y$ is an output soft constraint variable.

The optimization problem can be then defined as follows,

$$\min_{\mathbf{u}(k)} \quad \mathbf{J}\left(\mathbf{x}(k), \mathbf{u}(k)\right) \tag{3.2}$$

$$\begin{aligned} \text{subject to} \quad & \mathbf{x}(k+1) = \mathbf{A}_{\mathrm{d}}\mathbf{x}(k) + \mathbf{B}_{\mathrm{d}}\mathbf{u}(k) \\ & y(k) = \mathbf{C}\mathbf{x}(k) \\ & y_{lb} \leq \mathbf{s}_y(k) \leq y_{ub}, \text{ for } k = 1, \ldots, N \\ & 0 \leq \mathbf{u}(k) \leq u_{ub}, \text{ for } k = 0, 1, \ldots, N-1, \end{aligned} \tag{3.3}$$

where u_{ub}, y_{ub} and y_{lb} are the boundaries for both the hard input and soft output constraints.

Note that the hard state/output constraint are rarely used in real applications. The state/output constraint are typically softened by using a slack variable. The reason is that they may cause infeasibility of the optimization problem.

Move Blocking. One of the effective way to reduce the computational effort is called move blocking [12]. This method is based on the fixation of the manipulates variables over the time intervals in the prediction horizon. Degrees of freedom in the optimization problem are then reduced.

Two types of horizonts are used for move blocking. The control horizon N_C is a type of horizon on which the manipulated inputs $\mathbf{u}(k)$ are optimalized. Fixed manipulated input as the last optimized one $\mathbf{u}(k+N_C)$ is used for all the steps after control horizon N_C. The prediction horizon N_P is the horizon MPC is calculating the predicted outputs on, but where manipulated inputs are fixed. Denote that $N_P > N_C$. See remark 3.1 and Fig. 3.3.

Remark 3.1. Move blocking example is presented here. Instead os solving control law for the whole vector $\mathbf{u}^* = \begin{bmatrix} \mathbf{u}_0^{*\mathrm{T}} & \ldots & \mathbf{u}_{N-1}^{*\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{N \times m}$, it can optimized only the optimal vector $\mathbf{u}^* = \begin{bmatrix} \mathbf{u}_0^{*\mathrm{T}} & \ldots & \mathbf{u}_{M-1}^{*\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{M \times m}$, where m is the number of manipulated inputs, size of each vector $\mathbf{u}_i^{*T}$ respectivelly. Then to calculate the actual manipulated input $\mathbf{u} = (\mathbf{T} \otimes \mathbf{I}_m) \cdot \mathbf{u}^*$ where $\otimes$ stands for the Kronecker product and $\mathbf{T} \in \mathbb{R}^{N \times M}$ is called blocking matrix with $M < N$. $\mathbf{T}$ is a matrix of zeros and ones where each row contains exactly one non-zero element.

For the system with move blocking $\mathbf{u}_1 = \mathbf{u}_2 = \mathbf{u}_3$, $m = 1$ and $N = 4$ the matrices look as follows,

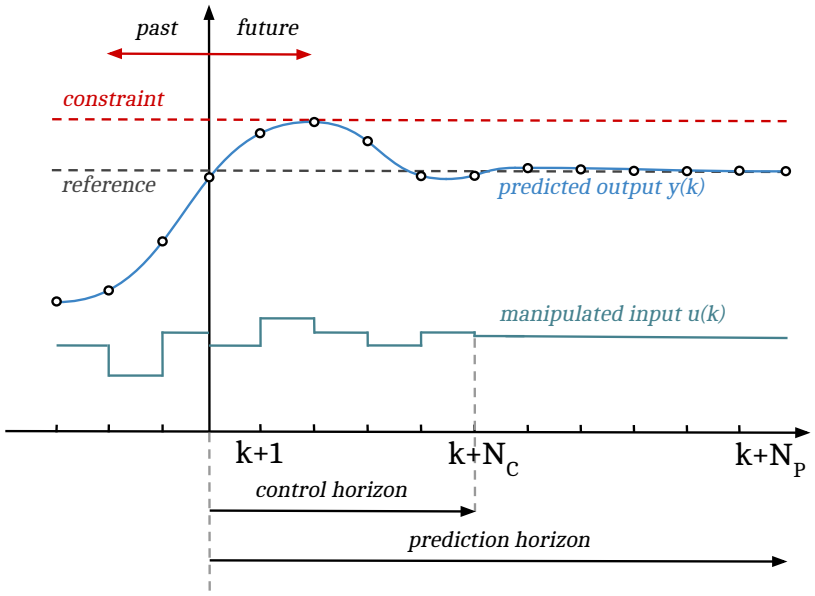


Figure 3.3: Model predictive control strategy [13]. This picture shows the main principle of MPC strategy. Two types of prediction horizonts are used in the picture to enable using of move blocking technique. It is important to say, that this shows the prediction loop, it represents one actual step in time. Usually only the first optimized input $\mathbf{u}(k)$ is used in one step in time. There is also shown the upper constraint for the control output y_{ub}.

$$\mathbf{u}^* = \begin{bmatrix} u_0^{*\mathrm{T}} & u_1^{*\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \ \mathbf{T} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

and the $N = 4$ degrees of freedom problem was reduced to $M = 2$ degrees of freedom optimization problem. [15]

3.1.2 Robust Model Predictive Control

Robust MPC [9], also called tube based MPC, is a type of MPC which is appropriate for the perturbed stochastic systems, see apendix. A. Stochastic systems are characteristic by presence of uncertainty in a model. Fig. 3.4 shows the robust MPC principles.

System Definition

The stochastic systems suitable for robust MPC design has the following discrete definition

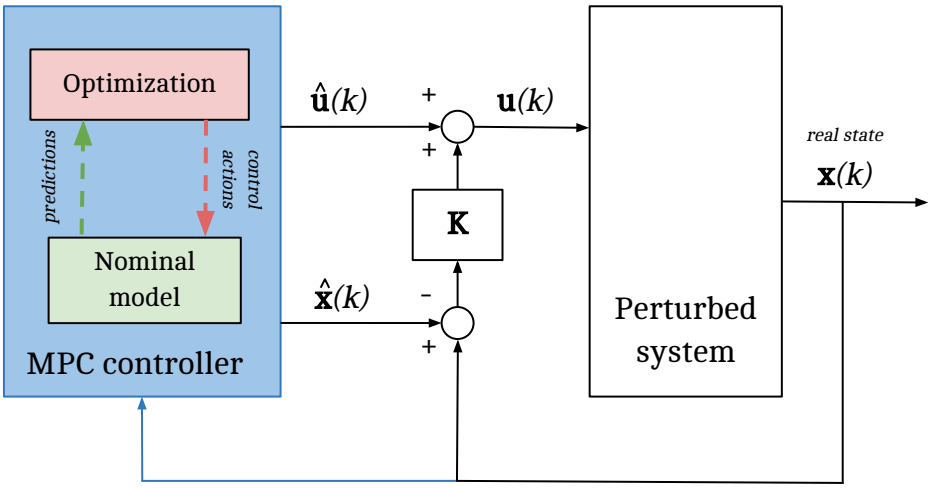


Figure 3.4: Robust MPC scheme, using real state feedback from the perturbed system. This scheme ilustrate the principles described in sec. 3.1.2.

$$\mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k) + \mathbf{B}_d\mathbf{u}(k) + \mathbf{w}(k),\ \ \mathbf{w}(k) \in \mathbb{W},$$

where $\mathbf{w}(k)$ is a bounded perturbation and $\mathbb{W} = \{\mathbf{w} : w_{lb} \leq \mathbf{w} \leq w_{ub}\}$ is a box uncertainty model. The goal of robust MPC is to design a controller that provides worst case optimality and constraints satisfaction for all possible distrurbances $\mathbf{w}(k)$ from $\mathbb{W}$.

A nominal model of this pertrubed system is

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}_d\hat{\mathbf{x}}(k) + \mathbf{B}_d\hat{\mathbf{u}}(k)$$

and the control law is defined as

$$\mathbf{u}(k) = \hat{\mathbf{u}}(k) + \mathbf{K}\left(\mathbf{x}(k) - \hat{\mathbf{x}}(k)\right),$$

where $\mathbf{K}$ will result from solving an optimization problem stated later.

Robust Positively Invariant Set

At first, some intuitive introduction for the positively invariant sets is presented [14]. These sets can be represented using the function trajectory. Once the function trajectory enters the positively invariant set, it will never leave it again. Mathematically representation is like follows

$$x(0) \in \mathbb{X} \Rightarrow x(k) \in \mathbb{X}, \text{ for } \forall k > 0.$$

Next step is to define the robust positively invariant (RPI) set $\mathbb{Z}$.

Note that

$$\begin{aligned} \mathbf{z}(k) &= \mathbf{x}(k) - \hat{\mathbf{x}}(k) \\ \mathbf{z}(k+1) &= (\mathbf{A}_d + \mathbf{B}_d\mathbf{K})\,\mathbf{z}(k) + \mathbf{w}(k), \end{aligned}$$

which is independent of $\hat{\mathbf{u}}(k)$.

If $\mathbf{A}_d + \mathbf{B}_d\mathbf{K}$ asymptotically stable, then there exists a RPI set $\mathbb{Z}$ such that, if $z(t) \in \mathbb{Z}$ and $\mathbf{w}(k) \in \mathbb{W}$ for $\forall k \geq t$, then

$$\mathbf{z}(t+i) \in \mathbb{Z}, \text{ for } \forall i \geq 0,$$

what can be also represented as

$$(\mathbf{A}_d + \mathbf{B}_d\mathbf{K})\mathbb{Z} \oplus \mathbb{W} \subseteq \mathbb{Z},$$

where $\oplus$ is a Minkowski sum of spaces.

It means that if there exists the set $\mathbb{Z}$ such that some $\mathbf{z}$ is in this set, $\mathbf{A}_d + \mathbf{B}_d\mathbf{K}$ is asymptotically stable and $\mathbf{w}$ lies in the uncertain bounded set $\mathbb{W}$, then every future value of $\mathbf{z}$ also lies in the set $\mathbb{Z}$.

Tightened Constraints

Using this RPI set $\mathbb{Z}$ the tightened constraints $\hat{\mathbb{X}}, \hat{\mathbb{U}}$ can be defined. Note the input constraints

$$\begin{aligned} \mathbf{x} &\in \mathbb{X} \subseteq \mathbb{R}^n \\ \mathbf{u} &\in \mathbb{U} \subseteq \mathbb{R}^m, \end{aligned}$$

then

$$\begin{aligned} \hat{\mathbf{x}} &\in \hat{\mathbb{X}},\ \hat{\mathbb{X}} = \mathbb{X} \ominus \mathbb{Z} \\ \hat{\mathbf{u}} &\in \hat{\mathbb{U}},\ \hat{\mathbb{U}} = \mathbb{U} \ominus \mathbf{K}\mathbb{Z}, \end{aligned}$$

where $\ominus$ is a Minkowski difference.

Nominal Model Invariant Set

The invariant set $\hat{\mathbb{X}}^f \subseteq \hat{\mathbb{X}}$, is the positively invariant terminal set for the nominal model, if $\hat{\mathbf{x}}(k) \in \hat{\mathbb{X}}^f$ and if the nominal state $\hat{\mathbf{x}}$ evolves as

$$\hat{\mathbf{x}}(k+1) = (\mathbf{A}_d + \mathbf{B}_d\mathbf{K})\hat{\mathbf{x}}(k),$$

then it also stands

$$\begin{aligned} \hat{\mathbf{x}}(k+i) &\in \hat{\mathbb{X}}^f \\ \mathbf{K}\hat{\mathbf{x}}(k+i) &\in \hat{\mathbb{U}} \end{aligned}$$

for all $i \geq 0$.

Optimization Problem

The robust cost function is defined in a similar fashion as (3.1)

$$\mathbf{J}\left(\hat{\mathbf{x}}(k), \hat{\mathbf{u}}(k)\right) = \mathbf{Q}\left\|\hat{\mathbf{x}}(N)\right\| + \sum_{k=1}^{N-1} \mathbf{Q}\left\|\hat{\mathbf{x}}(k)\right\| + \mathbf{R}\left\|\hat{\mathbf{u}}(k)\right\|,$$

where $\mathbf{Q} \geq 0, \mathbf{R} \geq 0$ are cost matrices of appropriate sizes. Here are not included any constraints in this description in order to emphasize robust control specific properties, so the cost function is slightly simplier than in the previous section.

The robust MPC optimization problem is defined as,

$$
\begin{aligned}
\min_{\hat{\mathbf{u}}(k)} \quad & \mathbf{J}\left(\hat{\mathbf{x}}(k), \hat{\mathbf{u}}(k)\right) \\
\text{subject to} \quad & \hat{\mathbf{x}}(k+1) = \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\hat{\mathbf{u}}(k) \\
& \hat{\mathbf{x}}(k) \in \hat{\mathbb{X}}, \text{ for } k = 1, ..., N \\
& \hat{\mathbf{u}}(k) \in \hat{\mathbb{U}}, \text{ for } k = 1, ..., N-1 \\
& \hat{\mathbf{x}}(k+N) \in \hat{\mathbb{X}}^f \\
& \mathbf{x}(k) = \hat{\mathbf{x}} \oplus \mathbb{Z}.
\end{aligned} \tag{3.4}
$$

Note that $\hat{\mathbf{x}}(t)$ is an argument of the optimization problem and the last constraint simply says that the nominal trajectory $\hat{\mathbf{x}}(k)$ depends on the pertrubed trajectory $\mathbf{x}(k)$. This fact includes an effect of uncertainty $\mathbf{w}$.

The result of the optimization problem is the optimal nominal input sequence

$$\hat{\mathbf{u}}^*(k : k+N-1|k) = \hat{\mathbf{u}}^*(k|k), ..., \hat{\mathbf{u}}^*(k+N-1|k),$$

and the nominal state

$$\hat{\mathbf{x}}^*(k|k).$$

The robust MPC control input is defined using these optimised results.

$$\mathbf{u}^*(k) = \hat{\mathbf{u}}^*(k|k) + \mathbf{K}\left(\mathbf{x}(k) - \hat{\mathbf{x}}^*(k|k)\right),$$

where $\mathbf{K}$ is any gain, which stabilizes $\mathbf{x} \in \mathbb{X}$.

Algorithm 2 Robust MPC Algorithm to solve the robust control case.

Input: Initial trajectory $\hat{\mathbf{x}}(0) = x_0$ at time step $k = 0$
Output: Control input $\mathbf{u}(k)$

1. Set $k + 1 \rightarrow k$
2. Solve the nominal MPC optimization problem (3.4).
3. Apply the robust control input $\mathbf{u}^*(k) = \hat{\mathbf{u}}^*(k|k) + \mathbf{K}\left(\mathbf{x}(k) - \hat{\mathbf{x}}^*(k|k)\right)$ to the real system.
4. Go back to step 1.

3.1.3 One-Zone Control (R1C0)

Finally, definition of a one-zone R1C0 building control is presented here. See sec. 2.2 for model definition. Every subsection describes firstly the MPC approach and secondly the robust MPC design, because the control techniques' definitions varies just slightly.

There are also several constraints specification options, which are clarified here in this chapter. The controlled outputs and manipulated inputs constraints come into consideration. The reason why to use the input contraints is fact, that the heat source is bounded.

Zone Model Refinement

The one-zone (R1C0) state space system is defined in (2.2). There are just couple of refinements.

The MPC design is a discrete-time control method, so at first it is necessary to discretize the created model.

At first, we split appart manipulated inputs and measurable disturbances

$$\begin{aligned} \mathbf{T}(k+1) &= \mathbf{A}_d\mathbf{T}(k) + B_d u(k) + \mathbf{B}_{d_v}\mathbf{v}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{T}(k) \\ & k = 1, \ldots, N. \end{aligned} \tag{3.5}$$

This refined system is used in the MPC definition, $u(k)$ now defines the only manipulated input and $\mathbf{v}(k)$ defines all the measurable disturbances. Also note that output-input matrix $\mathbf{D} = 0$ and so it was neglected in the definition above. The input vector, disturbance vector and matrices $B, \mathbf{B}_v$ are redefined as follows,

$$u = Q_{hx}, \mathbf{v} = \begin{bmatrix} T_{n_1} \\ \vdots \\ T_{n_4} \end{bmatrix}$$

$$B = \frac{1}{C_z}, \mathbf{B}_v = \begin{bmatrix} \frac{h_1}{C_z} & \ldots & \frac{h_4}{C_z} \end{bmatrix},$$

where the matrices B and $\mathbf{B}_v$ are countinuous forms of the matrices $B_d, \mathbf{B}_{d_v}$ from (3.5) and the other values are already defined in sec. 2. The state vector $\mathbf{T}$ and matrices $\mathbf{A}_d, \mathbf{C}$ remain unchanged.

For the robust MPC, the system is again slightly different as the uncertainty has to be defined

$$\begin{aligned} \mathbf{T}(k+1) &= \mathbf{A}_d\mathbf{T}(k) + B_d u(k) + B_{d_v} v(k) + \mathbf{B}_{d_w}\mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{T}(k) \\ & \quad k = 1, \ldots, N. \end{aligned}$$

Denote that $\mathbf{w}(k)$ is the bounded uncertainty vector and it includes all the neighbouring zones which are also controlled. The neighbouring zones' temperatures are for this zone unpredictable and are suitable candidates to be taken as uncertainties. The environment neigbouring zone temperature is not necessarily taken as uncertainty.

In a specific case, when the $T_{n_1} = T_{env}$ the system matrices would be redefined as follows,

$$u = Q_{hx}, v = T_{env},$$

$$\mathbf{w} = \begin{bmatrix} T_{n_2} \\ T_{n_3} \\ T_{n_4} \end{bmatrix}, \; T_{n_{i_{lb}}} \leq T_{n_i} \leq T_{n_{i_{ub}}} \text{ for } i = 2, 3, 4$$

$$B = \frac{1}{C_z}, B_v = \frac{h_1}{C_z}, \mathbf{B}_w = \begin{bmatrix} \frac{h_2}{C_z} & \frac{h_3}{C_z} & \frac{h_4}{C_z} \end{bmatrix},$$

where $\mathbf{B}_w$ is a continuous form of the matrix $\mathbf{B}_{d_w}$.

Zone Constraints Specification

As noted above, controlled output soft constraints are used in the zone control definition to allow zone temperature to vary freely around reference. Also the manipulated input is soft constrainted not to risk infeasibility.

The cost function with both these constraints included is defined as follows,

$$\mathbf{J}\left(\mathbf{x}(k), \mathbf{u}(k)\right) = \mathbf{Q}\left\|\mathbf{C}_d\mathbf{x}(k+N) - \mathbf{s}_y(k+N)\right\| + \tag{3.6}$$

$$+ \sum_{i=k}^{k+N-1} \mathbf{Q}\left\|\mathbf{C}_d\mathbf{x}(i) - \mathbf{s}_y(i)\right\| + \mathbf{R}\left\|\mathbf{u}(i)\right\| + \mathbf{q}\left\|\mathbf{u}(i) - \mathbf{s}_u(i)\right\|,$$

where s_y and s_u are soft constraints variables which are bounded and $\mathbf{q}$ is weight matrix of appropriate size. The reason why the input is weighted twice is to divide the costs for the input constraint and input behaviour. Denote also that this one zone control optimization definition is slightly changed once the distributed controller is introduced.

3.2 Centralized Control

This section introduces the centralized control approach and its application to a building. Model of building and its controller together creates a single system-controller unit. The system includes all inputs, outputs and model interactions, which are handled by the controller [9].

3.2.1 Centralized Model Predictive Control

Centralized approach which uses the model predictive control is presented in this section. See sec. 3.1.1 for MPC definition. Concept of a large scale system, which includes all the smaller subsystems and their interactions and which is handled by one controller, is shown in a centralized MPC scheme in Fig. 3.5.

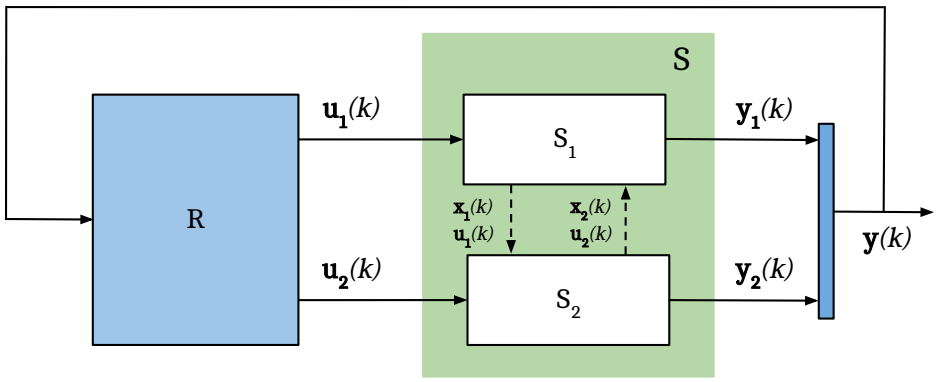


Figure 3.5: Centralized model predictive control principle of one system and one central controller is shown in this figure. Note that the system scale is one of the biggest centralized design effort, which leads to impact on computational complexity.

This approach has also it's limitations, especially for large scale MIMO systems, due to the computational complexity. The model definition is then complicated and if one system unit (subsystem) collapses then it influences the whole system.

General system definition and optimization problem specification can be taken from sec. 3.1.1.

3.2.2 Building Centralized Control (R1C0)

There was presented the one-zone control design in sec. 3.1.3. The centralized approach differs from the one-zone design mainly in scale. This section discusses an example of centralized MPC, which is designed to control the four-zone R1C0 model of building. For model definition see sec. 2.4.

Building System Refinement

Following matrix changes need to be done using the same methodic as in sec. 3.1.3,

$$
\begin{aligned}
\mathbf{u} &= \begin{bmatrix} Q_{hx_1} & Q_{hx_2} & Q_{hx_3} & Q_{hx_4} \end{bmatrix}^{\mathrm{T}} \\
v &= T_{amb} \\
\mathbf{B}^{big} &= \begin{bmatrix} \frac{1}{C_{z_1}} & 0 & \cdots & 0 \\ 0 & \frac{1}{C_{z_2}} & \ddots & \vdots \\ \vdots & \ddots & \frac{1}{C_{z_3}} & 0 \\ 0 & \cdots & 0 & \frac{1}{C_{z_4}} \end{bmatrix} \\
\mathbf{B}_v^{big} &= \begin{bmatrix} \frac{h_{1,1}+h_{1,4}}{C_{z_1}} \\ \frac{h_{2,1}+h_{2,2}}{C_{z_2}} \\ \vdots \\ \frac{h_{4,2}+h_{4,3}}{C_{z_4}} \end{bmatrix}.
\end{aligned}
$$

Partioning of the manipulated inputs $\mathbf{u}$ and the measurable disturbance v is presented above. Denote that measurable disturbance v equals to T_{amb}, what is the ambient (outside) temperature. The other state space matrices remain unchanged except of necessary discretization using the zero-order-hold method.

The robust approach is not suitable for centralised control design, since in our case the environment temperatures does not involve uncertainty.

Building Constraints Specification

The building zone constraints specification does not require any change in constraints formulation from the one-zone control case. For constraints formulation see sec. 3.1.3. The only difference is in the scale and the number of boundaries.

Each zone in the four-zone building model needs its own reference which implies that each zone also needs its own soft output constraints i.e. $y_{1_{lb}}, y_{2_{ub}}, ..., y_{4_{lb}}, y_{4_{ub}}$. The input constraint u_{ub} just acquires the appropriate value which corresponds to the four-zone heat exchanger capacity i.e. the constraint simply offers to set the four times higher then for the one-zone case.

3.3 Distributed Control

It is necessary to introduce the decomposition structures [7] in the very beggining of the distributed control design studies. The decomposition approach shows how to split the large scale system into the number of subsystems. System decomposition results the distributed or decentralized algorithm usage [9]. Here in this chapter only the distributed control design is presented.

Handling of the building zone interactions is the most complicated effort in a control design. Many of these distributed solutions turn into the iterative algorithm approach.

3.3.1 Decomposition

Decomposition is a general approach of solving a problem by breaking it up into smaller ones and solving each separately, either in parallel or sequentially [6, 7, 8]. Basically, decomposition methods describe solution of relations between subsystems. Two main approaches of decomposition are taken into account. The first one is called separable or trivially parallelizable, this approach is for systems where decomposition can be solved in one step. Subsystems are not dynamically coupled, but the objective function includes subsystems variables. The objective function can be represented as

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) = f_1(x_1) + ... + f_n(x_n) \\ \text{subject to} \quad & x_i \in C_i, \ i = 1, ...N, \end{aligned}$$

where C_i is a constraints set for a i-th subproblem.

The second case and the more interesting one is, when the subsystems are dynamically coupled like in Fig. 3.6. Then subsystems interact and any problem cannot be solved independently. For this case mainly the primal and dual decomposition methods are used. [7] There is a complicating variable, which increase the problem's complexity. Both the primal and dual decomposition objective functions are similar and look as follows,

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) = f_1(x_1, y) + ... + f_n(x_n, y) \\ \text{subject to} \quad & x_i \in C_i, \ i = 1, ...N, \end{aligned} \tag{3.7}$$

where y is a complicating variable.

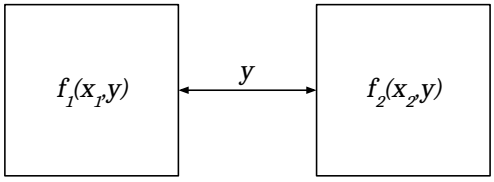


Figure 3.6: This scheme shows dynamically coupled systems. These systems are coupled with a complicating variable y. Since they share this complicating variable, they are not fully separable.

Primal Decomposition

This decomposition method is based on fixing the complicating variable $\mathbf{y}$. Then the problem (3.7) for $n = 2$ splits and decomposition is possible.

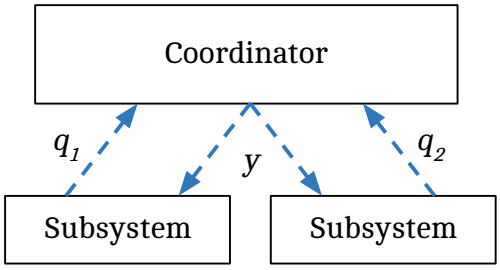


Figure 3.7: This scheme shows the primal decomposition structure. Each subystems sends the sensitivity $\mathbf{q}_i$ of local optimum to complicating variable to the coordinator. Coordinator checks the stop condition and optionaly responses with the updated complicating variable $\mathbf{y}$.

Consider two subproblems ϕ_1, ϕ_2 with objective functions f and optimization problems

$$\begin{aligned} \phi_1^*(\mathbf{y}) &= \underset{\mathbf{x}_1}{\text{minimize}}\ f(\mathbf{x}_1, \mathbf{y}) \\ \phi_2^*(\mathbf{y}) &= \underset{\mathbf{x}_2}{\text{minimize}}\ f(\mathbf{x}_2, \mathbf{y}). \end{aligned}$$

Also consider the master problem ϕ with objective function and optimization problem

$$\phi^*(\mathbf{y}) = \underset{\mathbf{y}}{\text{minimize}}\ \phi_1(\mathbf{y}) + \phi_2(\mathbf{y}).$$

An iterative algorithm is used to solve this problem. In every iteration the subproblems

ϕ_1 and ϕ_2 needs to be solved to find values $\phi_1^*(\mathbf{y})$ and $\phi_2^*(\mathbf{y})$ and its gradients or subgradients according to iterative method.

The following procedure describes using a subgradient method. See also Fig. 3.7. At first, solve the subproblems. Then calculate the subgradients of the local optima about the complicating variable $\mathbf{y}$.

$$\mathbf{q}_i = \frac{\partial \phi_i^*(\mathbf{x}_i^*, \mathbf{y})}{\partial \mathbf{y}} \tag{3.8}$$

Then the master problem subgradient is

$$\mathbf{q} = \sum_i \mathbf{q}_i.$$

Then update the complicating variable $\mathbf{y}$ by

$$\mathbf{y}_{k+1} = \mathbf{y}_k - \alpha_k \mathbf{q},$$

where $\alpha_k > 0$ is a step size in descent. The step size definition methods are briefly summarized in sec. 3.3.2. The whole algorithm is summarized in Alg. 3.

Algorithm 3 Primal decomposition

Input: Initial estimate of $\mathbf{y}_0$
Output: Optimal value $\mathbf{x}^*$, value of $\mathbf{y}_k$ at the k-th iteration step

1. Distribute $\mathbf{y}_k$ to subsystems.
2. Optimize each subproblem and return the sensitivity of each local optimum to complicating variable.

$$\mathbf{q}_i = \frac{\partial \phi_i^*(\mathbf{x}_i^*, \mathbf{y}_k)}{\partial \mathbf{y}_k}$$

3. Coordinator updates the complicating variable $\mathbf{y}_k$

$$\mathbf{y}_{k+1} = \mathbf{y}_k - \alpha_k \sum_i \mathbf{q}_i$$

4. Continue to 1. if the stop condition is not passed.

$$\left| \sum_i \mathbf{q}_i \right| < \varepsilon$$

Dual Decomposition

A dual decomposition of the problem (3.7) is schematically depicted in Fig. 3.8.

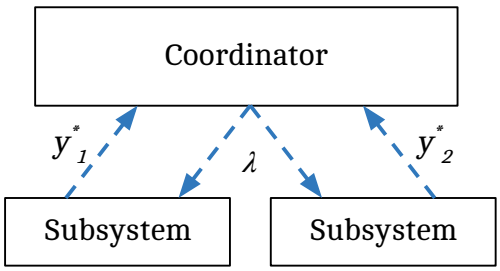


Figure 3.8: The dual decomposition scheme is presnted to compare it with the primal decomposition one. Here each of the subsystems sends the local optimal value of $\mathbf{y}_i^*$ to the coordinator. Then the stop condition is checked and the coordinator optionaly responses with an update of a shadow price $\boldsymbol{\lambda}$.

For a specific case when $n = 2$, the problem can be expressed as follows

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(\mathbf{x}) = f_1(\mathbf{x}_1, \mathbf{y}_1) + f_2(\mathbf{x}_2, \mathbf{y}_2) \\ \text{subject to} \quad & \mathbf{y}_1 = \mathbf{y}_2. \end{aligned}$$

It means that each local system has it's local copy of complicating variable $\mathbf{y}$ and the consensus is the consistency constraint that requires that all the local copies of $\mathbf{y}$ are equal. The objective function is now separable.

Define the Lagrangian

$$\begin{aligned} L(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2, \boldsymbol{\lambda}) &= f_1(\mathbf{x}_1, \mathbf{y}_1) + f_2(\mathbf{x}_2, \mathbf{y}_2) + \boldsymbol{\lambda}^{\mathrm{T}}(\mathbf{y}_1 - \mathbf{y}_2) \\ &= L_1(\mathbf{x}_1, \mathbf{y}_1, \boldsymbol{\lambda}) + L_2(\mathbf{x}_2, \mathbf{y}_2, \boldsymbol{\lambda}), \end{aligned}$$

which is separable and each subsystem solves it's part

$$\begin{aligned} g_1(\boldsymbol{\lambda}) &= \underset{\mathbf{x}_1\mathbf{y}_1}{\text{minimize}}\ L_1(\mathbf{x}_1, \mathbf{y}_1, \boldsymbol{\lambda}) \\ &= \underset{\mathbf{x}_1\mathbf{y}_1}{\text{minimize}}\ f_1(\mathbf{x}_1, \mathbf{y}_1) + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{y}_1 \\ g_2(\boldsymbol{\lambda}) &= \underset{\mathbf{x}_2\mathbf{y}_2}{\text{minimize}}\ L_2(\mathbf{x}_2, \mathbf{y}_2, \boldsymbol{\lambda}) \\ &= \underset{\mathbf{x}_2\mathbf{y}_2}{\text{minimize}}\ f_2(\mathbf{x}_2, \mathbf{y}_2) - \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{y}_2. \end{aligned}$$

These subproblem optimas $g_1(\boldsymbol{\lambda}), g_2(\boldsymbol{\lambda})$ define the master problem and objective function

$$\max_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda}} g_1(\boldsymbol{\lambda}) + g_2(\boldsymbol{\lambda}),$$

which can be solved by gradient or subgradient methods similarly to the primal decomposition formulation (3.8)

$$\begin{aligned} \frac{\partial g_1(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} &= \mathbf{y}_1^* \\ \frac{\partial g_2(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} &= -\mathbf{y}_2^* \\ \frac{\partial g(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} &= \mathbf{y}_1^* - \mathbf{y}_2^*. \end{aligned}$$

This solution is used to update the new iteration of shadow prices $\boldsymbol{\lambda}$ as follows

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

See the full algorithm description below.

Algorithm 4 Dual decomposition

Input: Initial estimated value of $\boldsymbol{\lambda}_0$

Output: Optimal values of $\mathbf{y}_1^*, \mathbf{y}_2^*$, value of $\boldsymbol{\lambda}_k$ at the k-th iteration step

1. Distribute $\boldsymbol{\lambda}_k$ to subsystems.
2. Optimize each dual subproblem and return the complicating variable $\mathbf{y}_1^*, \mathbf{y}_2^*$ of each local optimum.
3. Coordinator updates the shadow prices $\boldsymbol{\lambda}_k$.

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

4. Continue to 1. if the stop condition is not passed.

$$|\mathbf{y}_1^* - \mathbf{y}_2^*| < \varepsilon$$

General Decomposition Structures

A general decomposition method is described in this section. Systems with only two subsystems were described above, however large scale systems require more general decomposition methods [7].

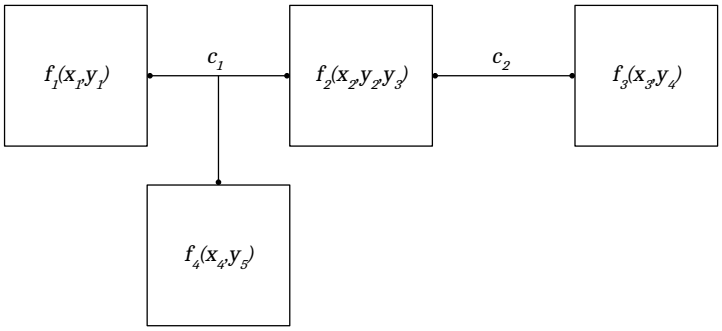


Figure 3.9: More complicated decomposition structure is shown in this figure. Different subsystems share different complicating variables using coupling constraints. e.g. coupling constraint c_1 requires that the complicating variables y_1, y_2, y_5 are equal.

There is an example of a more complex system in Fig. 3.9. Coupling constraint c_1 is a constraint that requires that the complicating variables y_1, y_2, y_5 are equal. Further, the coupling constraint c_2 requires equality of the complicating variables y_3, y_4. The associated problem can be written as

$$\begin{aligned} \underset{\mathbf{x},\mathbf{y}}{\text{minimize}} \quad & f_1(\mathbf{x}_1,\mathbf{y}_1) + f_2(\mathbf{x}_2,\mathbf{y}_2,\mathbf{y}_3) + f_3(\mathbf{x}_3,\mathbf{y}_4) + f_4(\mathbf{x}_4,\mathbf{y}_5) \\ \text{subject to} \quad & \mathbf{y}_1 = \mathbf{y}_2 = \mathbf{y}_5,\ \mathbf{y}_3 = \mathbf{y}_4. \end{aligned}$$

The decomposition general approach is based on the graph theory. Have K subsystems, each of them has public variables $\mathbf{y}_i \in \mathbb{R}^{p_i}$, private variables $\mathbf{x}_i \in \mathbb{R}^{n_i}$, objective function $f_i : \mathbb{R}^{n_i} \times \mathbb{R}^{p_i}$ and local constraints $C_i \subseteq \mathbb{R}^{n_i} \times \mathbb{R}^{p_i}$.

Next step is to describe the subsystems coupling by public variables. This coupling also corresponds to a net in the decomposition structure. Collect all the public variables into one vector variable $\mathbf{y} = (\mathbf{y}_1, ..., \mathbf{y}_k) \in \mathbb{R}^p$, where $p = p_1 + ... + p_k$ is the total number of public variables, for instance p_1 stands for the total number of public variables of the first subsystem. Now two notations need to be distinguished, $(\mathbf{y})_i$ describes the i-th component of $\mathbf{y}$, for $i = 1, ..., p$, and $\mathbf{y}_i$ is the portion of $\mathbf{y}$ associated with subsystem i.

The coupling constraints are expressed by

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

where $\mathbf{z} \in \mathbb{R}^N$ stands for the common value of the public variables for N nets. The vector $\mathbf{z}$ is also called as the vector of net variables. $\mathbf{E} \in \mathbb{R}^{p \times N}$ is described as follows

$$\mathbf{E}_{ij} = \{e_{i,j}\};\ e_{i,j} = \begin{cases} 1 & (\mathbf{y})_i \text{ is in net } j \\ 0 & \text{otherwise} \end{cases}.$$

Each row of the matrix $\mathbf{E}$ is associated with the different subsystem i. The problem of N nets is described using the following expresion

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{N} f_i(\mathbf{x}_i, \mathbf{y}_i) \\ \text{subject to} & \mathbf{y}_i = \mathbf{E}_i \mathbf{z}, \qquad i = 1, ..., N, \end{array} \tag{3.9}$$

where $\mathbf{E}_i \in \mathbb{R}^{p_i \times N}$ is the partioning of the rows of $\mathbf{E}$ into blocks associated with the different subsystems. Denote that matrix $\mathbf{E}_i$ is a $0-1$ matrix, which maps the vector of net variables $\mathbf{z}$ into the public variables $\mathbf{y}_i$ of subsystem i.

Next step is to briefly describe the primal and dual decomposition using this general notation.

Primal Decompostion. The vector of net variables $\mathbf{z}$ and the public variables $\mathbf{y}_i = \mathbf{E}_i \mathbf{z}$ for all nets are fixed at each iteration, which makes the problem separable. Then the local objective functions are defined as

$$\phi_i(\mathbf{y}_i) = \underset{x_i}{\text{minimize}}\ f_i(\mathbf{x}_i, \mathbf{y}_i).$$

The problem defined in (3.9) can be expressed as

$$\text{minimize}\ \phi(\mathbf{z}) = \text{minimize} \sum_{i=1}^{K} \phi_i(\mathbf{E}_i \mathbf{z}). \tag{3.10}$$

Both the master and local problems can be solved by the gradient or subgradient methods.

$$\begin{aligned} \mathbf{q}_i &= \frac{\partial \phi_i(\mathbf{x}_i^*, \mathbf{y}_i)}{\partial \mathbf{y}_i} \\ \mathbf{q} &= \sum_{i=1}^{N} \mathbf{E}_i^{\mathrm{T}} \mathbf{q}_i \in \partial \phi(\mathbf{z}) \end{aligned}$$

See the algorithm describing all the steps of primal decomposition method.

Algorithm 5 Primal decomposition

Input: Initial estimated value of $\mathbf{z}_0$

Output: Optimal value $\mathbf{x}^*$, value of $\mathbf{z}_k$ at the k-th iteration step

1. Distribute net variables to subsystems.

$$\mathbf{y}_i = \mathbf{E}_i \mathbf{z}_k, \;\; i = 1, ..., N$$

2. Separately optimize each subproblem (3.10) and compute gradient.

$$\mathbf{q}_i = \frac{\partial \phi_i(\mathbf{x}_i^*, \mathbf{y}_i)}{\partial \mathbf{y}_i}, \;\; i = 1, ..., N$$

3. Collect and sum the subproblems sensitivities to get gradient of the master problem.

$$\mathbf{q} = \sum_{i=1}^{N} \mathbf{E}_i^{\mathrm{T}} \mathbf{q}_i$$

4. Update the net variables.

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \alpha_k \mathbf{q}$$

Dual Decomposition. The first step is to form the problem (3.9) using the Lagrangian.

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

where $\boldsymbol{\lambda} \in \mathbb{R}^{\mathrm{T}}$ is the vector of the Lagrange multipliers (shadow prices) associated with $\mathbf{y} = \mathbf{E}\mathbf{z}$ and $\boldsymbol{\lambda}_i$ is the subvector of $\boldsymbol{\lambda}$ associated with the subsystem i.

The master problem with it's consensus can be defined by minimizing the Lagrangian over $\mathbf{z}$.

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \quad & g(\boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda}} \; \textstyle\sum_{i=1}^{N} g_i(\boldsymbol{\lambda}_i) \\ \text{subject to} \quad & \mathbf{E}^{\mathrm{T}} \boldsymbol{\lambda} = 0, \end{aligned}$$

where the subproblems optimal solutions $g_i(\boldsymbol{\lambda}_i)$ are defined as

$$g_i(\boldsymbol{\lambda}_i) = \underset{\mathbf{x}_i \mathbf{y}_i}{\text{minimize}}\ f_i(\mathbf{x}_i, \mathbf{y}_i) + \boldsymbol{\lambda}_i^{\mathrm{T}} \mathbf{y}_i.$$

This problem can be solved by subgradient method. [1, 7, 8] The projection operator onto the feasible set $\mathbf{E}^{\mathrm{T}}\boldsymbol{\lambda} = 0$ is affine and gives an iteration

$$\boldsymbol{\lambda}_{k+1} \quad = \quad \boldsymbol{\lambda}_k + \alpha_k \left(\mathbf{I} - \mathbf{E}(\mathbf{E}^{\mathrm{T}}\mathbf{E})^{-1}\mathbf{E}^{\mathrm{T}} \right) \mathbf{y}, \tag{3.11}$$

which is possible to be interpreted by the following steps. $\hat{\mathbf{z}} = (\mathbf{E}^{\mathrm{T}}\mathbf{E})^{-1}\mathbf{E}^{\mathrm{T}}\mathbf{y}$ gives the average of the public variables $\mathbf{y}$ over each net. Then they are substracted from the corresponding public variables $\mathbf{g} = \mathbf{y} - \mathbf{E}\hat{\mathbf{z}}$ to form a projected subgradient.

Therefore, each subproblem has its own copy of the public variables and Lagrange multipliers (shadow prices). After the subsystems optimizes the local problems, the coordinator compares the public variables and updates the shadow prices. The consensus is to effort the consistency between the appropriate public variables according to the nets distribution. Note that each net tries to reach the consistency of it's related public variables. See also the algorithm below, which describes all the steps of general dual decomposition.

Algorithm 6 Dual decomposition

Input: Initial estimated value of $\boldsymbol{\lambda}_0$ satisfying $\mathbf{E}^{\mathrm{T}}\boldsymbol{\lambda} = 0$ (e.g. $\boldsymbol{\lambda} = \mathbf{0}$)
Output: Optimal value of $\mathbf{y}$, value of $\boldsymbol{\lambda}_k$ at the k-th iteration step

1. Distribute $\boldsymbol{\lambda}_k$ to subsystems.
2. Optimize each dual subproblem separately. Solve $g_i(\boldsymbol{\lambda}_i)$ and return the optimal $\mathbf{y}_i$.
3. Coordinator computes the average of public variables over each net.

$$\hat{\mathbf{z}} = (\mathbf{E}^{\mathrm{T}}\mathbf{E})^{-1}\mathbf{E}^{\mathrm{T}} y$$

4. Coordinator updates prices on public variables.

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \alpha_k \left(\mathbf{y} - \mathbf{E}\hat{\mathbf{z}} \right)$$

5. Continue to 1. if the stop condition (consistency constraint residual condition) is not passed.

$$\begin{aligned} r \quad &= \quad \|\mathbf{y} - \mathbf{E}\hat{\mathbf{z}}\|_2 \\ r \quad &< \quad r_{THRES} \end{aligned}$$

Also note that there is a stop condition which is defined by the consistency between the corresponding public variables

$$r = \|g\|_2 = \|\mathbf{y} - \mathbf{E}\hat{\mathbf{z}}\|_2 .$$

It is also called the consistency constraint residual.

3.3.2 Step Size Rules

Here is a brief introduction into the rules used to choose the step size in the iterative algorithms with the subgradient methods. The convergence analysis and more step size rules can be found in [7, 8].

Constant Step Size

The simpliest case is to use the constant step size, where $\alpha_k = \alpha$ and $\alpha > 0$. An important fact is that α is independent on the iteration index k.

Non-Summable Diminishing Step Sizes

Step size has to satisfy couple of following conditions in this nonsummable diminishing case

$$\alpha_k \geq 0, \ \sum_{k=1}^{\infty} \alpha_k = \infty, \ \lim_{k\to\infty} \alpha_k = 0$$

A typical example of this step size rule is

$$\alpha_k = \frac{a}{\sqrt{k}}$$

where $a > 0$.

Algorithm by Nesterov

Nesterov gradient method or so called Nesterov's algorithm is fully described in [1, 8]. It is an advanced step size rule. The main principle is to reduce the amount of computation at each step using the point (state) extrapolation. It also uses some of the basic step sizes rules to choose the factor α. See the full algorithm 7.

Algorithm 7 The Nesterov's algorithm of gradient evaluation in the extrapolated point.
Input: Actual and previous points $\mathbf{x}_k$, $\mathbf{x}_{k-1}$.
Output: Next step point $\mathbf{x}_{k+1}$.

1. Calculate previous state direction vector $\mathbf{v}_k$

$$\mathbf{v}_k = \mathbf{x}_k - \mathbf{x}_{k-1}$$

2. Initialize the damping factor ε_k

$$\varepsilon_k = \frac{k-1}{k+2}$$

3. Extrapolate the current point

$$\bar{\mathbf{x}}_k = \mathbf{x}_k + \varepsilon_k \mathbf{v}_k$$

4. Make gradient step from the extrapolated point

$$\mathbf{x}_{k+1} = \bar{\mathbf{x}}_k + \alpha_k \nabla f(\bar{\mathbf{x}}_k)$$

3.3.3 Distributed Model Predictive Control

This section deals with distributed system control. Such systems consist of subsystems without any dependency specification, see Fig. 3.10. Term distributed means a system divided into smaller subsystems, which do not exclude interaction. Each subsystem has its controller and the locally calculated information exchange between them is permitted. The transmition topology is of two types. It can be the fully connected transmission network (all-to-all communication) or the partially connected transmission network (neighbour-to-neigbour communication). Next very important fact is about how the information is transmitted. There are again two approaches of the information exchange, an iterative approach and a non-iteravtive approach. If we talk about the iterative manner, the information is exchanged more than once during one sample time. On the other hand, there is just one information transmission during one sample time with non-iterative information exchange approach.

If we talk about distributed MPC, it means that each local zone is controlled by it's own MPC. Since all the subsystems are similiar, it is possible to have one controller definition used in all the subsystems, what simplifies the design process. Each of these local zone MPCs solves the optimization problems or so called local subproblems. In some cases there is also some large scale problem, which have to be solved by some kind of supervising unit. Here come into play the decomposition structures. The supervising units are called coorinators and their task is to solve the master problem.

Moreover, one of the similar structures is decentralised expression, which stands for

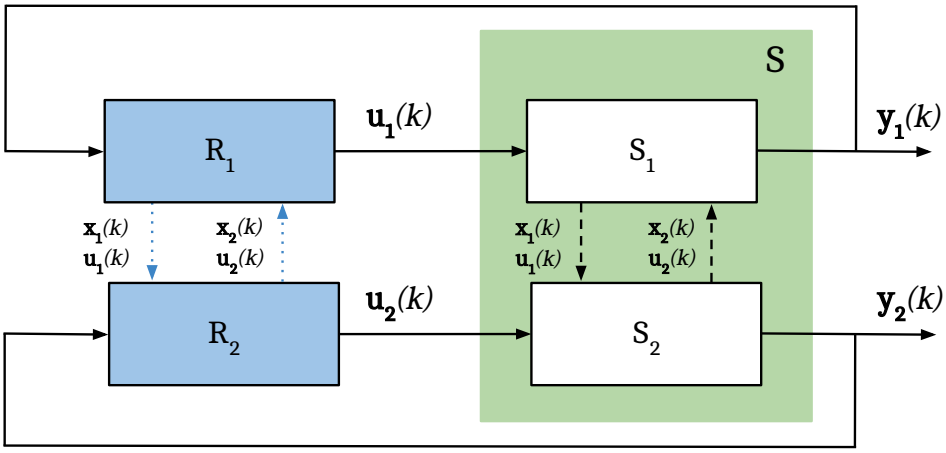


Figure 3.10: Distributed control scheme of a distributable system. The distributed stands here for the fact, that the system is possible to divide into smaller units, not necessarily independent. This is the main difference against the decentralised approach, the subsystems can also interract between each other. The information (predicted state and manipulated inputs) is callculated locally and transmitted among the local regulators. Again, each subsystem has its own local controller.

the system consisting of strictly independent subsystems as denoted in [8], see Fig. 3.11. It can be also represented as each subsystem having a local controller designed to control $\mathbf{x}_i(k)$ using input $\mathbf{u}_i(k)$ independently of $\mathbf{x}_j(k)$ and $\mathbf{u}_j(k)$, where $i \neq j$.[9]

3.3.4 Building Distributed Control (R1C0)

Building control impels to use the distributed solution. The reason is that each building zone has very similiar physical description and behaviour what can be very helpful during the control desing.

Two main problems were solved during the control design. The first one was the local zone MPC solution i.e. definition of the cost function, settings of the appropriate constraints and level of communication with other zones. The second problem was the resource allocation problem [16], where coordinator needs to fairly distribute it's limited heat resource between the zone subsystems.

Local Zone MPC

Theory of the two main approaches used in the local zone MPC is presented in sec. 3.1. These two control approaches were used to design the appropriate control mechanism. Both the algorithms are based on an iterative process. If we talk about the local zone

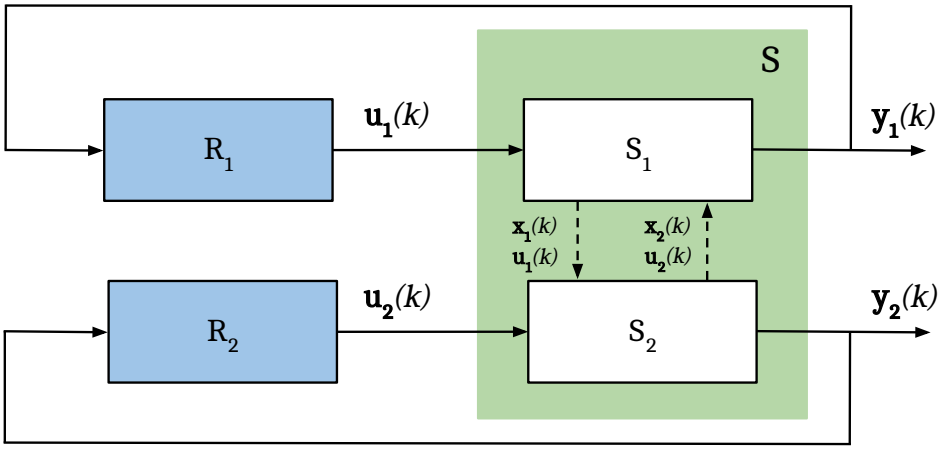


Figure 3.11: Decentralised control scheme of a splitted system. The decentralised means that some system is divided into smaller subsystems and interactions are neglected. The main advantage of this independence is very small impact if one of the subsystems collapse. The control engineer also does not have to solve the problem of subsystems' communication and ther interactions, since they are independent. Interactions can be also neglected and designated as a model error. There could be seen in the picture that each the subsystem has its own local controller.

MPC, the main point is to optimize handling of the zones neighbours interactions. See Alg. 8 and 9. These algorithms also do not solve the resource allocation problem which is solved within the optimalization as the superior problem.

Exact Zone Interactions. The first approach can be described like a model when each zone updates it's neighbours agents with the information about it's temperature prediction in each iteration step. It basically means that all the zone temperatures are kept actual for all the neighbouring zones. The stop condition of this iteration algorithm is based on the difference between the two consecutive steps of the neighbours predicted temperatures.

Algorithm 8 Exact zone interactions algorithm representing the high level of agent communication.

Input: Initial values of the neighbouring zones temperatures
$\mathbf{y}^{0*}_{n\neq l}(k) = \begin{bmatrix} y^{0*}_{n\neq l}(k) & \dots & y^{0*}_{n\neq l}(k+N) \end{bmatrix}$ for each zone l.
Output: The optimized control vector $\mathbf{u}^*_l(k) = \begin{bmatrix} u^*_l(k) & \dots & u^*_l(k+N) \end{bmatrix}$.

1. Set iteration index $q = 0$. Keep constant sampling time k.
2. Each zone l of m zones with n_m neighbours simultaneously solves the optimization problem (3.1,3.2) to obtain $\mathbf{y}^{q*}_l(k) = \begin{bmatrix} y^{q*}_l(k) & \dots & y^{q*}_l(k+N) \end{bmatrix}$ using the information from it's neighbours $\mathbf{y}^{q-1*}_{n\neq l}(k)$, where $n = 1,\dots,m;\ l = 1,\dots,m$ and index $n \neq l$ describes ommiting of zone l from $\mathbf{y}^*_{n\neq l}$.

$$\begin{aligned} \mathbf{x}^{q*}_l(k+i+1) &= \mathbf{A}_d\mathbf{x}^{q*}_l(k+i) + \mathbf{B}_d u^{q*}_l(k+i) + \mathbf{B}_{d_v}\mathbf{v}^q_l(k+i) \\ y^{q*}_l(k+i) &= \mathbf{C}_{\mathrm{d}}\mathbf{x}^{q*}_l(k+i),\ l = 1,\dots,m \end{aligned}$$

   The neighbours temperatures $\mathbf{y}^{q-1*}_{n\neq l}(k)$ are handled in the model as disturbances $\mathbf{v}^q_l(k) = \begin{bmatrix} y^{q-1*}_1(k) & \dots & y^{q-1*}_{n_m}(k) \end{bmatrix}^{\mathrm{T}}$. The environment neighbours temperatures are also included in the disturbance matrix. Note that the information about neighbours' temperatures is one iteration step behind the new prediction.
3. Each zone l checks the stop condition.

$$\left|\mathbf{y}^{q*}_l(k) - \mathbf{y}^{q-1*}_l(k)\right| \leq \varepsilon,\ \varepsilon \geq 0,\ l = 1,\dots,m$$

   a. Condition passed. Continue to step 5.
   b. Condition not passed. Set $q = q+1$ and $\mathbf{y}^{q-1*}_l(k) = \mathbf{y}^{q*}_l(k)$. All zones exchange this information with their neighbours and go back to step 2.
4. Apply the first step of the optimalized manipulated input $\mathbf{u}^{q*}_l(k)$ where $l = 1,\dots,m$.
5. Move horizont to the next sampling time $k = k+1$ and go to step 1.

Robust Zone Interactions . The second approach is based on the robust MPC behaviour from sec. 3.1.2. The zone predicted outputs were exchanged in every iteration in the prevoius algorithm, here the zone communication happens only at each sample time. The robust approach is used to compensate the weak zone interactions using the neighbours' predictions uncertainties.

Algorithm 9 Robust based zone interaction algorithm using the predictions with uncertainty.

Input: Initial values of the neighbouring zones temperatures
$\mathbf{y}^*_{n\neq l}(k) = \begin{bmatrix} y^*_{n\neq l}(k) & \dots & y^*_{n\neq l}(k+N) \end{bmatrix}$ for each zone.
Output: The optimized control vector $\mathbf{u}^*_l(k) = \begin{bmatrix} u^*_l(k) & \dots & u^*_l(k+N) \end{bmatrix}$.

1. Keep constant sampling time k.
2. Each zone l of the m zones with n_m neighbours simultaneously solves the robust optimization problem from sec. 3.1.3 to obtain $\mathbf{u}^*_l(k)$ and $\mathbf{y}^*_l(k) = \begin{bmatrix} y^*_l(k) & \dots & y^*_l(k+N) \end{bmatrix}$ using the information from it's neighbours $\mathbf{y}^*_{n\neq l}(k-1)$, where $n = 1,\dots,m,\ l = 1,\dots,m$ and index $n \neq l$ describes ommiting of zone l from $\mathbf{y}^*_{n\neq l}$.

$$\begin{aligned} \mathbf{x}^*_l(k+1) &= \mathbf{A}_d\mathbf{x}^*_l(k) + \mathbf{B}_d u^*_l(k) + \mathbf{B}_{d_v}\mathbf{v}_l(k) + \mathbf{B}_{d_w}\mathbf{w}_l(k) \\ y^*_l(k) &= \mathbf{C}_\mathrm{d}\mathbf{x}^*_l(k),\ l = 1,\dots,m,\ k = 1,\dots,N \end{aligned}$$

where $\mathbf{v}_l(k)$ includes only measurable disturbances like environment temperature. Vector $\mathbf{w}_l(k)$ includes the neighbours temperatures' predictions $\mathbf{y}^*_{n\neq l}(k-1)$ as disturbances with uncertainty. The uncertainty δw is bounded and grows with the sampling time.

$$\begin{aligned} \mathbf{w}_l(k) &= \begin{bmatrix} y^*_1(k-1) & \dots & y^*_{n_m}(k-1) \end{bmatrix}^\mathrm{T} + p(k)\cdot\delta w \\ p(k) &= k\cdot q,\ q > 0 \\ |\delta w| &\leq r,\ r > 0, \end{aligned}$$

where uncertainty increasing with sampling time is expressed using a parameter $p(k)$.
Also note that the information about neighbours' predictions is one sample time behind the new zone prediction.

3. Apply the first step of the optimalized manipulated input $\mathbf{u}^*_l(k)$ where $l = 1,\dots,m$.
4. Move horizont to the next sampling time $k = k+1$ and all zones exchange the predicted output information $\mathbf{y}^*_l(k)$ with their neighbours and go back to step 1.

Resource Allocation

Building distributed MPC solves the resource allocation problem when the local zones units require more resource (heat) than available. Heat has to be distributed between the zones fairly with respect to the reference tracking.

Nash Optimality. Distributed MPC algorithm using the Nash optimality is described in [17].

Assume a system with m zones. Each zone has the local MPC with the cost function $\mathbf{J}_l$ defined in sec. 3.1.3 and each zone has it's own local decision variable $\mathbf{u}_l(k) = \begin{bmatrix} u_l(k) & \dots & u_l(k+N) \end{bmatrix}$, $l = 1, \dots, m$. These local decision variables are iteratively optimized to get $\mathbf{u}_l^*(k)$. Denote that the manipulated input (decision variable) in each local cost function is bounded using the soft upper constraints $\mathbf{u}_{l_{ub}}(k) = \begin{bmatrix} u_{l_{ub}}(k) & \dots & u_{l_{ub}}(k+N) \end{bmatrix}$.

The principle used here is that each local soft input constraint is updated in each iteration until the check condition is passed, respectively until the Nash equilibrum is reached. The input constraint update is performed using an information from the neighbouring zones. The algorithm is described step by step below.

The control desicion set $\mathbf{u}(k) = \begin{bmatrix} \mathbf{u}_1(k) & \dots & \mathbf{u}_m(k) \end{bmatrix}$ is said to be Nash optimal (reaches the Nash equilibrum) if

$$\mathbf{J}_l\left(\mathbf{x}(k), \mathbf{u}_l^q(k), \mathbf{u}_{n\neq l}(k)\right) \leq \mathbf{J}_l\left(\mathbf{x}(k), \mathbf{u}(k), \mathbf{u}_{n\neq l}(k)\right), \; n = 1, \dots, m,$$

where q is one iteration step.

Basically, the control set $\mathbf{u}(k)$ is Nash optimal if there does not exist any value of $\mathbf{J}_l$ lower than the value of $\mathbf{J}_l$ obtained using the $\mathbf{u}_l^q(k)$.

Algorithm 10 Resource allocation algorithm using the Nash optimality.

Input: Resource upper limit U_{max}. Initial value of input soft constraints $\mathbf{u}^0_{l_{ub}}(k) = \begin{bmatrix} u^0_{l_{ub}}(k) & \dots & u^0_{l_{ub}}(k+N) \end{bmatrix}$.

Output: The optimized control vector $\mathbf{u}^*_l(k) = \begin{bmatrix} u^*_l(k) & \dots & u^*_l(k+N) \end{bmatrix}$.

1. Set iteration index $q = 0$. At constant sampling time k, each of the m zones (subsystems) local MPC makes initial estimation of $\mathbf{u}^{q*}_l(k)$ using the soft constraint $\mathbf{u}^0_{l_{ub}}(k)$. Each zone sends estimated local decision variable to all the other zones.

$$\mathbf{u}^{q*}_l(k) = \begin{bmatrix} u^{q*}_l(k) & \dots & u^{q*}_l(k+N) \end{bmatrix}, \; l = 1, \dots, m$$

2. Each zone updates its input soft constraint

$$\mathbf{u}_{l_{ub}}(k) = U_{max} - \sum_{l=1}^{m} \mathbf{u}^{q*}_{n \neq l}(k), \; l, n = 1, \dots, m$$

   and solves the optimization problem with the cost function $\mathbf{J}_l$ from sec. 3.1.3 simultaneously to obtain $\mathbf{u}^*_l(k)$.

3. Each zone checks if its iteration condition is satisfied.

$$\left| \mathbf{u}^{q*}_l(k) - \mathbf{u}^{q-1*}_l(k) \right| \leq \varepsilon, \; l = 1, \dots, m, \; \varepsilon \geq 0$$

   a. Condition passed. Continue to step 4.

   b. Condition not passed. Set $q = q + 1$ and $\mathbf{u}^{q-1*}_l(k) = \mathbf{u}^{q*}_l(k)$. All zones exchange this information and go back to step 2.

4. Apply the first step of the optimalized manipulated input $\mathbf{u}^{q*}_l(k)$ where $l = 1, \dots, m$.

5. Move horizon to the next sampling time $k = k + 1$ and go to step 1.

Dual Decomposition. The second approach defines the resource allocation as a master problem in a dual decomposition structure. Role of a coordinator solving the master problem is introduced here. Every zone represents one of the subproblems. Dual decomposition is described in sec. 3.3.1 and using this approach the resource allocation problem was solved in an iterative way.

Consider m subsystems with an appropriate cost function $\mathbf{J}_l$ described below in Alg. 11. The local optimization variable is the control input $\mathbf{u}_l(k) = \begin{bmatrix} u_l(k) & \dots & u_l(k+N) \end{bmatrix}$, $l = 1, \dots, m$. This variable is in every iteration step sent to the coordinator. The coordinator updates the shadow prices $\boldsymbol{\lambda}$ using the Nesterov's algorithm 3.3.2. These prices are distributed to the local zones and reflected to their cost functions. For local zones MPC problem solution, see the Alg. 8, 9.

Algorithm 11 Resource allocation algorithm using the dual decomposition approach.
Input: Initial value of $\boldsymbol{\lambda}_0$. Resource upper limit U_{max}.
Output: The optimized control vector $\mathbf{u}_l^*(k) = \begin{bmatrix} u_l^*(k) & \dots & u_l^*(k+N) \end{bmatrix}$.

1. Set iteration index $q = 0$. Keep constant sampling time k.
2. The coordinator distributes $\boldsymbol{\lambda}_q$ to zones.
3. Each of the m zones solves the optimization problem from sec. 3.1.3 to obtain $\mathbf{u}_l^{q*}(k)$. Each zone uses the extended cost function which reflects the distributed shadow prices.

$$\mathbf{J}_l = \mathbf{J}_l + \sum_{k=1}^{N-1} \boldsymbol{\lambda} \left\| \mathbf{u}_l(k) \right\|, \; l = 1, \dots, m$$

   And send these optimal values $\mathbf{u}_l^{q*}(k)$ to the coordinator
4. Coordinator updates the shadow prices $\boldsymbol{\lambda}$ using the Nesterov algorithm 3.3.2.

$$\boldsymbol{\lambda}_{q+1} \quad = \quad \bar{\boldsymbol{\lambda}}_q - \alpha_q \left(U_{max} - \sum_{l=1}^{m} \mathbf{u}_l^{q*}(k) \right)$$

   where α_q is a step size. The step size rules are described in sec. 3.3.2.
5. Coordinator checks the stop condition.

$$|\boldsymbol{\lambda}_{q+1} - \boldsymbol{\lambda}_q| \leq \varepsilon, \; \varepsilon \geq 0$$

   a. Condition passed. Continue to step 6.
   b. Condition not passed. Set $q = q + 1$ and $\boldsymbol{\lambda}_q = \boldsymbol{\lambda}_{q+1}$.
6. Apply the first step of the optimalized manipulated input $\mathbf{u}_l^{q*}(k)$ where $l = 1, \dots, m$.
7. Move horizont to the next sampling time $k = k + 1$ and go to step 1.

Chapter 4

Control Results

This chapter presents the final results of this thesis. It is based on all the previous chapters and uses the concepts introduced therein.

At first, the fundamental one-zone building control simulations of two types are shown, see the one-zone models description in sec. 2.2 and 2.3. Moreover, the simulation benchmark is presented and all the control approaches descibed in sec. 3 are tested on it with a specific setup. That is to say, benchmark uses centralized and distributed approach. Also various resource allocation algorithms from sec. 3.3.4 are presented in the simulations. Furthemore, comparison of the step size rules for the dual decomposition concept from sec. 3.3.2 is simulated and commented. Objective functions of the control optimization algorithms are used for comparative analysis.

Denote that the sampling time $T_s = 15\text{min}$ is used for all the simulations in this thesis. Computations and simulations were performed using MATLAB® & SIMULINK® R2016a. Control optimization algorithms were performed using YALMIP Version 19-Sep-2015. [2, 3]

4.1 Benchmark Definition

The five-zone building model from Fig. 4.1 was defined as the building benchmark for all the simulations below. However, some basic one-zone simulations are also presented in the following section as the introduction for the five-zone benchmark simulations.

Five-zone building benchmark consists of the four neighbouring building zones which surround one central zone. This central zone is located in a center of the benchmark. This zone distribution was selected to attain strong zone interactions during the simulations. In order to simplify benchmark model and computational difficulty, the benchmark uses the simple R1C0 model for each zone. See sec. 2.2 for further information about this zone model.

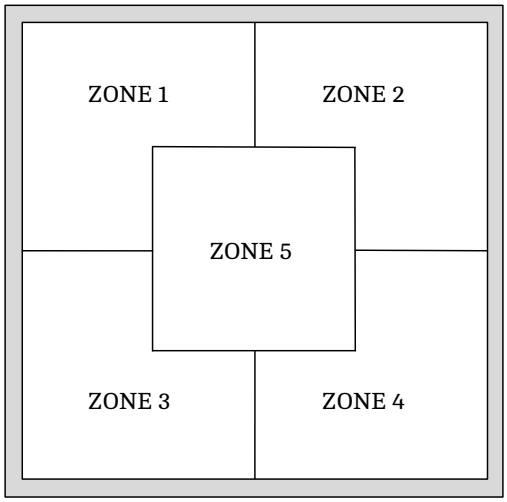


Figure 4.1: This scheme represents the the five-zone building benchmark profile. One central zone surronded by the four neighbouring zones was selected to properly simulate interactions. between zones

Benchmark initial conditions are defined as follows. Zone initial temperatures are $T_{z_1} = 15°\text{C}$, $T_{z_2} = 16°\text{C}$, $T_{z_3} = 15°\text{C}$, $T_{z_4} = 17°\text{C}$, $T_{z_5} = 15°\text{C}$. The ambient (outside temperature) is defined as $T_{amb} = -12°\text{C}$. The heat resource Q, which equals to input vector u, is softly bounded $u_{min} < u < u_{max}$, where $u_{min} = 0$ and u_{max} value is case specific. The output is also soft constrainted $r - \varepsilon < y < r + \varepsilon$, where $\varepsilon = 0.5°\text{C}$ and r stands for a reference. Each benchmark control approach also uses different resource distribution algorithm, what is explained in each appropriate section in particular.

The optimization problem setup is equal for the one-zone examples as well as for all the five-zone building benchmark cases (unless otherwise stated). All the cases use the objective function presented in (3.6) which is minimalized on the prediction horizon $N = 50$ as described in sec. 3.1.1. The costs Q, R, q have the following values, $Q = 1 \cdot 10^7$, $R = 1$, $q = 4 \cdot 10^5$. The centralized MPC optimization problem specific details can be found in sec. 3.2.2.

4.2 Simulations

There are presented building zones and benchmark simulations in this section, firstly, one-zone MPC of two types; secondly, benchmark centralized MPC; then, benchmark exact interactions distributed MPC with resource distribution using Nash optimality. Finally, benchmark robust distributed MPC with resource allocation based on dual decomposition is presented.

4.2.1 One-Zone MPC

As mentioned above, the beggining of this section deals with the one-zone model control. The simple R1C0 one-zone optimization problem from sec. 3.1.3 is solved and presented below. Also the same problem, but for the more complex one-zone R2C1 model is solved here in this section. Both the models situations are presented in Fig. 4.2. This one-zone concept is equivalent to the decentralised control approach.

These simulations represents the behaviour of one-zone surrounded by the environment with the ambient (environment) temperature $T_{amb} = -12°\text{C}$, which represents the only neighbour. The zone initial temperature is $T_z = 15°\text{C}$ and it tracks the reference r using the soft constraint $r - \varepsilon < y < r + \varepsilon$, where $\varepsilon = 0.5°\text{C}$. The zone's wall initial temperature $T_w = 6°\text{C}$ has to be specified for R2C1 zone control. Heat resource Q represents the input vector u and is bounded using the soft constraint $u_{min} < u < u_{max}$, where $u_{min} = 0\text{W}$, $u_{max} = 3450\text{W}$.

4.2.2 Benchmark: Centralized MPC

This control part is based on sec. 3.2, especially on sec. 3.2.2, where the model and optimization refinement are presented. However, the five-zone building benchmark, described in sec 4.1, is used for a simulation here. The MPC optimization problem for each specific zone of the benchmark is defined the same way as in the sec 4.2.1 above.

Fig. 4.3, 4.4 and 4.5 show three five-zone building benchmark simulations with centralized MPC. Each of the three cases uses different resource limitation. For the first resource limit stands $u_{max} = 9\text{kW}$, for the second $u_{max} = 8\text{kW}$ and for the third $u_{max} = 6\text{kW}$. The soft constraint $\sum_i u_i < u_{max}$, which is equal to the one defined in 4.1, is used here. These figures show the resource distribution and resource zone allocation through the prediction horizon as well. The first case presents a situation when enough resource is available. On the other hand the third case shows a situation in resource shortage. The most significant inter-zone interactions can be seen in the zone 5 temperature predictions. In the end of zone 5 simulation the temperature is still increasing and violating the output soft constraint, despite not using the heat resource, $u_5^* = 0$ respectivelly.

4.2.3 Benchmark: Exact Interactions Distributed MPC

The centralised-distributed conversion is described in sec. 3.3. This control approach uses the same benchmark as the previous centralized one. The difference is, that the distributed approach is used here. Basically, the zones are split into five subsystems. Inter-zone interactions are achieved by the communication between these subsystems, what also implies the distributed approach.

The exact interactions algorithm, described in sec. 3.3.4, deals with the subsystems (local zones) communication. Algorithm with the Nash optimality is used here for resource distribution, see sec. 3.3.4.

The same initial conditions, as in sec. 4.2.2 above, are used here. The soft constraint upper bound of heat resource Q (input vector u respectivelly) is defined as follows, $u_{max} = 9\text{kW}$ and it is distributed between the zones using the Nash optimality algorithm mentioned above.

The distributed approach with exact interactions algorithm is presented in Fig. 4.6. Firstly, from the local zone MPCs point of view, local MPCs use the exact interactions algorithm. Secondly, each zone simulation contains the resource allocation with a variable resource limit created by the resource distribution algorithm. It brings a big advantage, which is notable f.e. in the zone 5 simulation. Zone 5 requires minimal amount of heat to reach the reference temperature, which results in more available heat resource for the other zones. Other inter-zone interactions are also shown in the simulation. Solution presented in this section was reached after only 5 iterations, what is achieved due to sufficient amount of available heat resource.

4.2.4 Benchmark: Robust Distributed MPC

This control approach uses the distributed MPC principles as well as the previous one. The five-zone building benchmark is used again, see sec. 4.1.

The difference from the previous approach is in the local zone MPC problem handling. In this example, the robust based MPC design from sec. 3.3.4 is used for the local zones. Uncertainty influence on the local zone robust MPCs is illustrated later in this section. Resource allocation problem is here solved using dual decomposition based algorithm described in sec. 3.3.4. Dual decomposition technique is based on an iterative approach.

The input soft constraint upper bound is again case specific as mentioned above. Three cases are presented for robust distributed MPC in Fig. 4.7, 4.8 and 4.9, where the input upper bound value is $u_{max} = 9\text{kW}$ for the first case, $u_{max} = 8\text{kW}$ for the second case and $u_{max} = 6\text{kW}$ for the third case. The last case has heat resource soft bounded significantly more than the previous cases to show the dual decomposition technique advantages. Figures show the resource distribution through the zones on the right-bottom simulations, this demonstrates also the soft constraint behaviour. It was slightly violated to reach the optimal solution in the third case. Actual impact of this violation would show up in the real simulations.

The local zones use robust MPCs as a control approach, what implies the uncertainty in the system, see sec. 3.3.4. Fig. 4.10 is related to the first case robust distributed MPC simulation and shows the neighbouring zones uncertainties for zone 1. The neighbouring zones uncertainties increase with the prediction horizon. The actual prediction temperatures from Fig. 4.7 fits into the uncertain area for zone 1 neighbouring zones from Fig. 4.10.

Significantly, the optimization problem definition differs from the previous examples. The reason is the resource distribution with dual decomposition technique. See the local zone objective function specification in Alg. 11. Values of the cost matrices differ as well, $Q = 1 \cdot 10^5$, $R = 0.01$ and cost matrix q is not used in the objective function.

The first presented case solution was achieved in 191 iteration steps, the second case in 435 iteration steps and the third case in 3961 iteration steps, using the stop condition value $\varepsilon = 0.1$ and the algorithm by Nesterov from sec. 3.3.2 as the advanced step size rule. The specific α value used in Nesterov's algorithm was $\alpha_k = \frac{1}{k}$.

Other α values were tested with Nesterov's algorithm as well. The results are presented in sec. 4.3, see Fig. 4.11.

4.3 Evaluation of the Results

The algorithms for five-zone building benchark simulated above are evaluated and compared in this section. Firstly, the centralized MPC from sec. 4.2.2. Secondly, the exact interactions distributed MPC using resource distribution algorithm with Nash optimality, see sec. 4.2.3. Thirdly, the robust distributed MPC using dual decomposition based resource distribution algorithm, simulations are presented in sec. 4.2.4. Another key point to evaluate here is the Nesterov's algorithm for various step sizes (α values respectivelly).

4.3.1 Benchmark Control Evaluation

This section evaluates results of the three algorithms mentioned above. Comparison of the algorithms is made by using an uniform cost function. Optimization problems of all the three control algorithms are set using MPC problems in a various definitions. The centralised MPC was shown as well as the exact interactions distributed and robust distributed MPC.

A problem is, that each of the three optimization problems objective function is slightly different. Denote that objective function of each specific case cannot be compared with the others due to different costs and different specific terms handling the resource distribution. With respect to this fact, after the analysis, only the common parts of all three objective functions were used to make results evaluation. For each algorithm was calculated the uniform objective function for purpose of the algorithms comparison. The uniform objective function J_u is defined as follows,

$$J_u = \sum_{1}^{N} \left(\mathbf{Q} \left\| \mathbf{y} - \mathbf{s}_y \right\| + \mathbf{R} \left\| \mathbf{u} \right\| \right),$$

Algorithm	Objective	Resource Limit u_{max}[W]		
-	-	9000	8000	6000
Centralised MPC	$J_{u_1}[-] \cdot 10^9$	0.1449	0.7381	3.0777
Exact Distributed MPC	$J_{u_2}[-] \cdot 10^9$	0.2367	infeasible	infeasible
Robust Distributed MPC	$J_{u_3}[-] \cdot 10^9$	0.1853	1.3088	1.5988

Table 4.1: This table compares three algorithms used for five-zone building benchmark control. Firstly, the centralized MPC algorithm is represented by J_{u_1}. Secondly, the exact interactions distributed MPC using resource distribution algorithm with Nash optimality has the uniform objective J_{u_2}. Thirdly, objective function J_{u_3} stands for the robust distributed MPC using dual decomposition based resource distribution algorithm.

where the term containing cost matrix $\mathbf{Q}$ stands for the reference tracking penalization and the second term containing the $\mathbf{R}$ cost matrix describes a resource price. For further information about this objective function see sec. 3.1.1. The cost matrices got following values, $Q = 1 \cdot 10^7$, $R = 1$ and the uniform objective function is calculated for whole prediction horizon $N = 50$.

Three simulation cases were presented in the previous sec. 4.2. Each simulation setup differs from the others by available resource amount. All the three simulation cases are compared in a following Tab. 4.1, which shows the uniform objective function values for each algorithm. Centralized approach provides generally the best results. It is manifested through the lowest values of the uniform objective function J_{u_1}. However, the robust MPC approach with dual decomposition algorithm for resource distribution provides the best result in the last case with $u_{max} = 6$kW. Moreover, the centralized MPC was able to find a solution in this case with very low resource amount available as well. See Fig. 4.9 and 4.5.

4.3.2 Step Size Analysis for Algorithm by Nesterov

Nesterov's algorithm, described in 3.3.2, was used and analyzed with development of the dual decomposition based resource distribution algorithm, see sec. 4.2.4. Resource distribution algorithm is described in 3.3.4.

The five-zone building benchmark simulation from Fig. 4.9 uses with Nesterov's algorithm the step size (α value respectivelly) rule $\alpha_k = \frac{1}{k}$, but four more basic step size rules were selected according to [8]. All the four step size rules were applied for Nesterov's algorithm for this five-zone building benchmark setup with $u_{max} = 6$kW. They were compared from the perspective of obective function optimal value convergence. What's more, the number of iterations is also important fact. See Fig. 4.11 where the step size comparison is presented with detailed look into the first 1000 iterations.

From the presented data, all the used step sizes convergence is acceptable and they are heading to the same optimal value of the objective function J. Difference is only in the speed of convergence, number of iterations respectivelly. The step size rules $\alpha_k = \frac{1}{k}$ and $\alpha_k = \frac{1}{1000+k}$ were found as the most suitable ones for our case with Nesterov's algorithm

and the first one was used in the simulation from sec. 4.2.4.

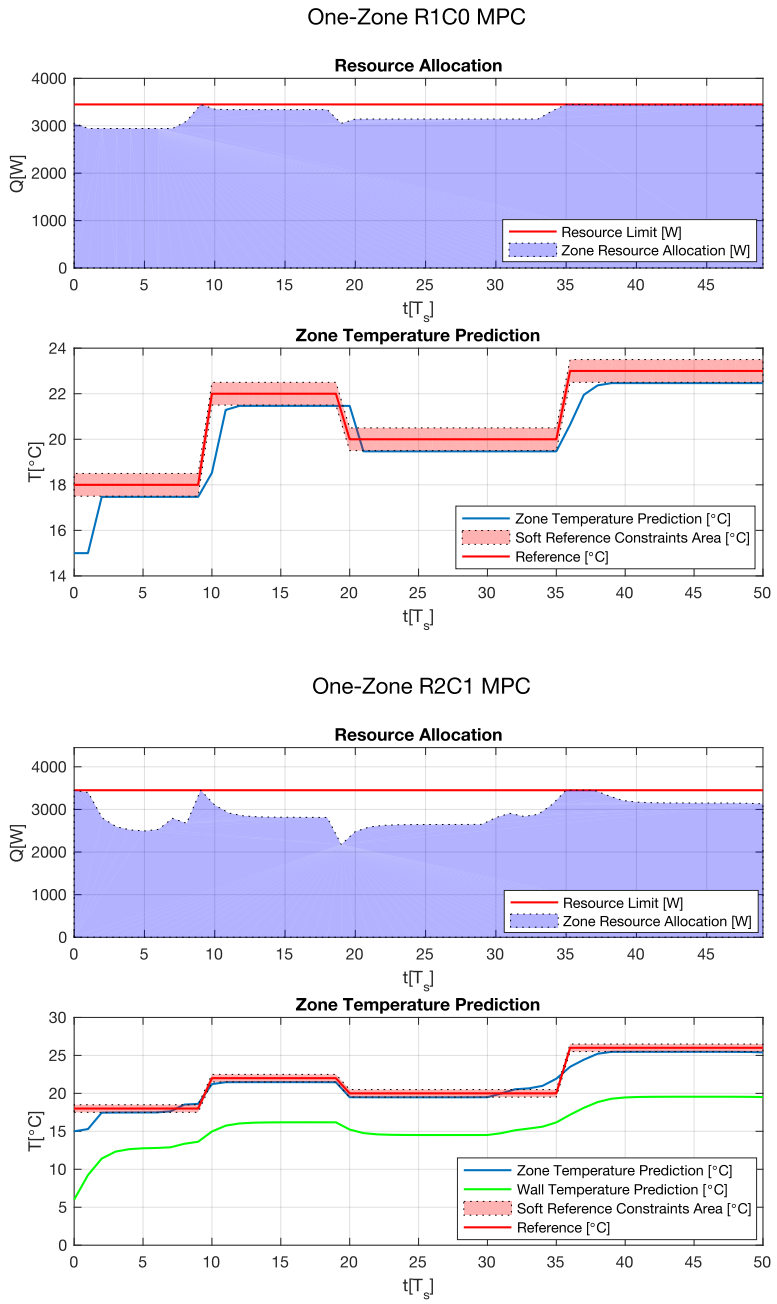


Figure 4.2: This figure shows the one-zone building MPC design for two model types. Picture on the top demonstrates the simplified one-zone R1C0 model MPC. The one-zone R2C1 model MPC is presented in the bottom part of figure . Both the simulations uses the same initial setup. Initial zone temperature $T_z = 15°\text{C}$, input soft constraint $0\text{W} < u < 3450\text{W}$, output soft constraint $r - 0.5°\text{C} < y < r + 0.5°\text{C}$ and for R2C1 zone also initial wall temperature $T_w = 6°\text{C}$. Three reference changes are simulated to present the control accuracy and input soft constraint influence. The reference tracking delay is shown there, where zone requires more resource then available. This input shortage leads to the input resource limit sliding.

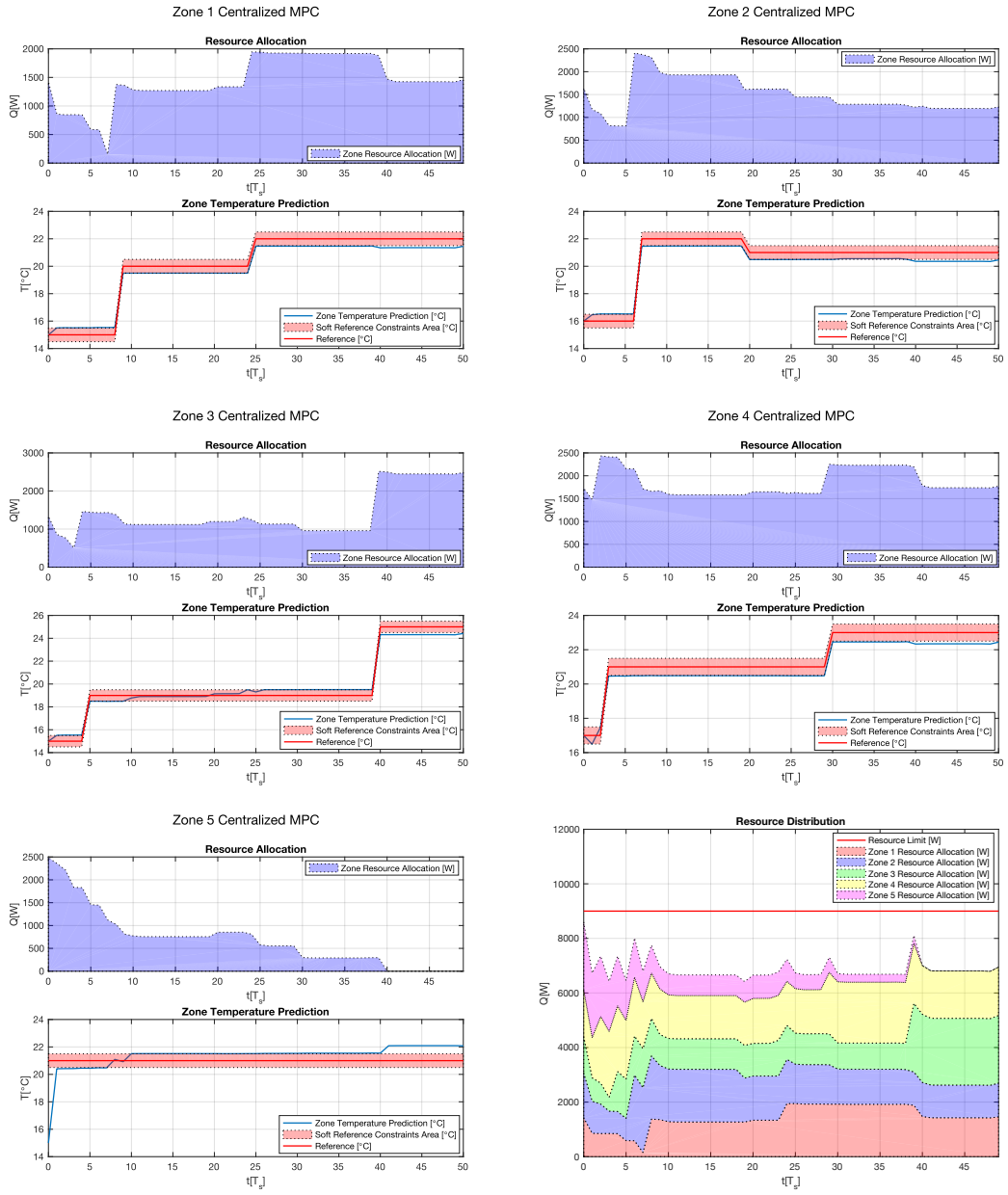


Figure 4.3: All the zones temperature predictions and resource allocations using the centralized MPC are presented in this figure. Simulations are shown from zone 1 to zone 5 from top pictures to bottom ones. Each zone simulation contains the temperature prediction and optimalized input u_i^* represented by heat resource Q. Finally, resource distribution is presented on the right-bottom picture. Initial conditions are defined in the benchmark specification sec. 4.1. Inter-zone interactions have the biggest impact on the zone 5. The predicted temperature of zone 5 is increasing, despite not using input heat $u_5^* = 0$. The bottom-left simulation shows the zone 5 data.

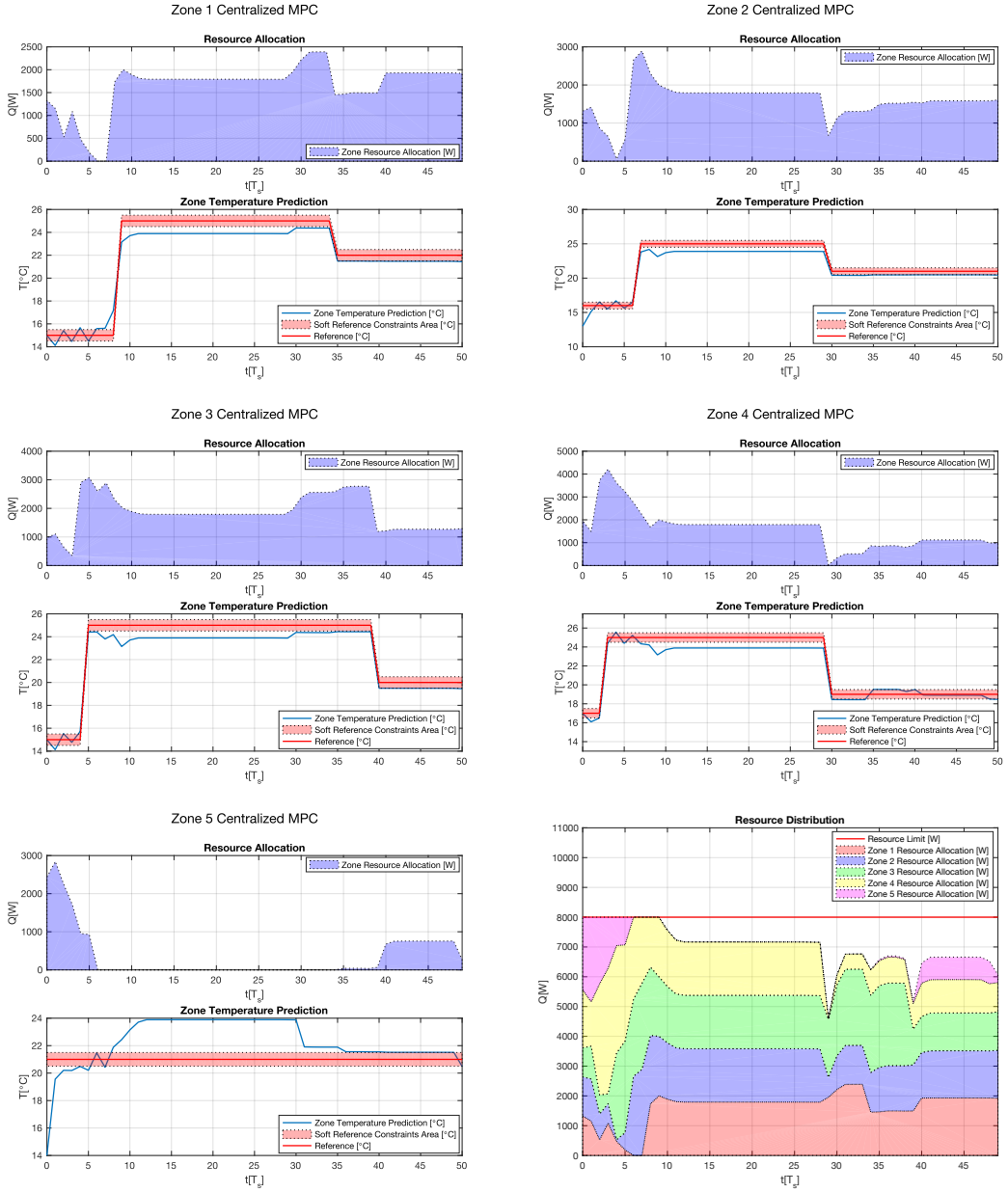


Figure 4.4: This figure presents second case for centralized MPC, where resource limit $Q_{max} = 8\text{kW}$ is used for the input constraint $\sum_i Q_i < Q_{max}$. Heat Q represents the input u. Initial conditions are defined in the benchmark specification sec. 4.1. The benchmark zone are shown from zone 1 to zone 5 from top to bottom of this figure. Resource distribution plot is situated in the bottom-right position. This case shows situation when resource requirements are high in the beggining of simulation and then they are gradually decreasing.

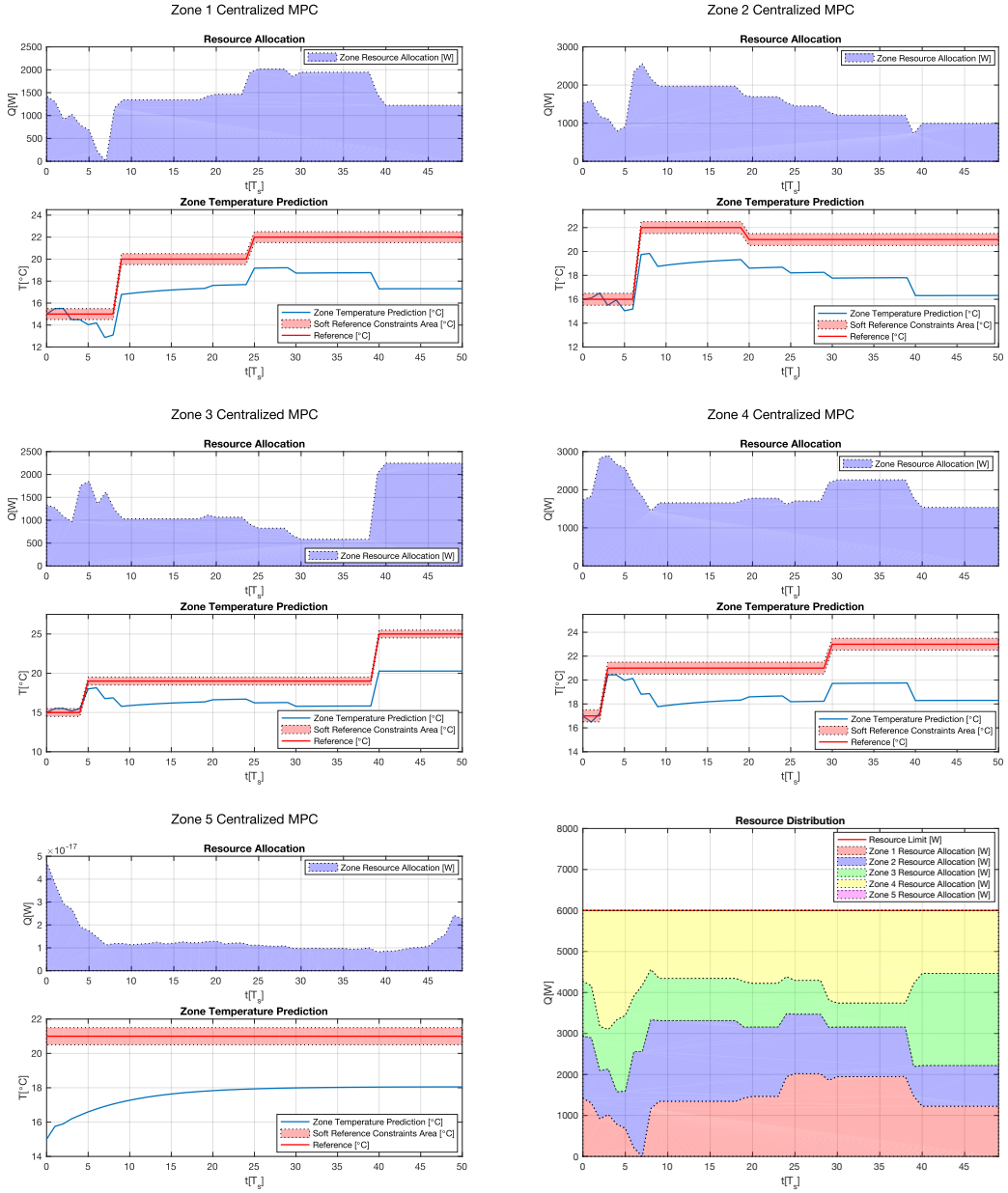


Figure 4.5: This figure shows the third case for centralized MPC algorithm of five-zone building benchmark. Resource limitation was increased even more than in previous case presented in Fig. 4.4 and for input upper bound stands $Q_{max} = 6\text{kW}$. All the other initial conditions are described in sec. 4.1. Zones are presented from top to bottom in numerical order with resource distribution plot on the bottom-right position. This case simulates centralised MPC behaviour in situation of resource shortage.

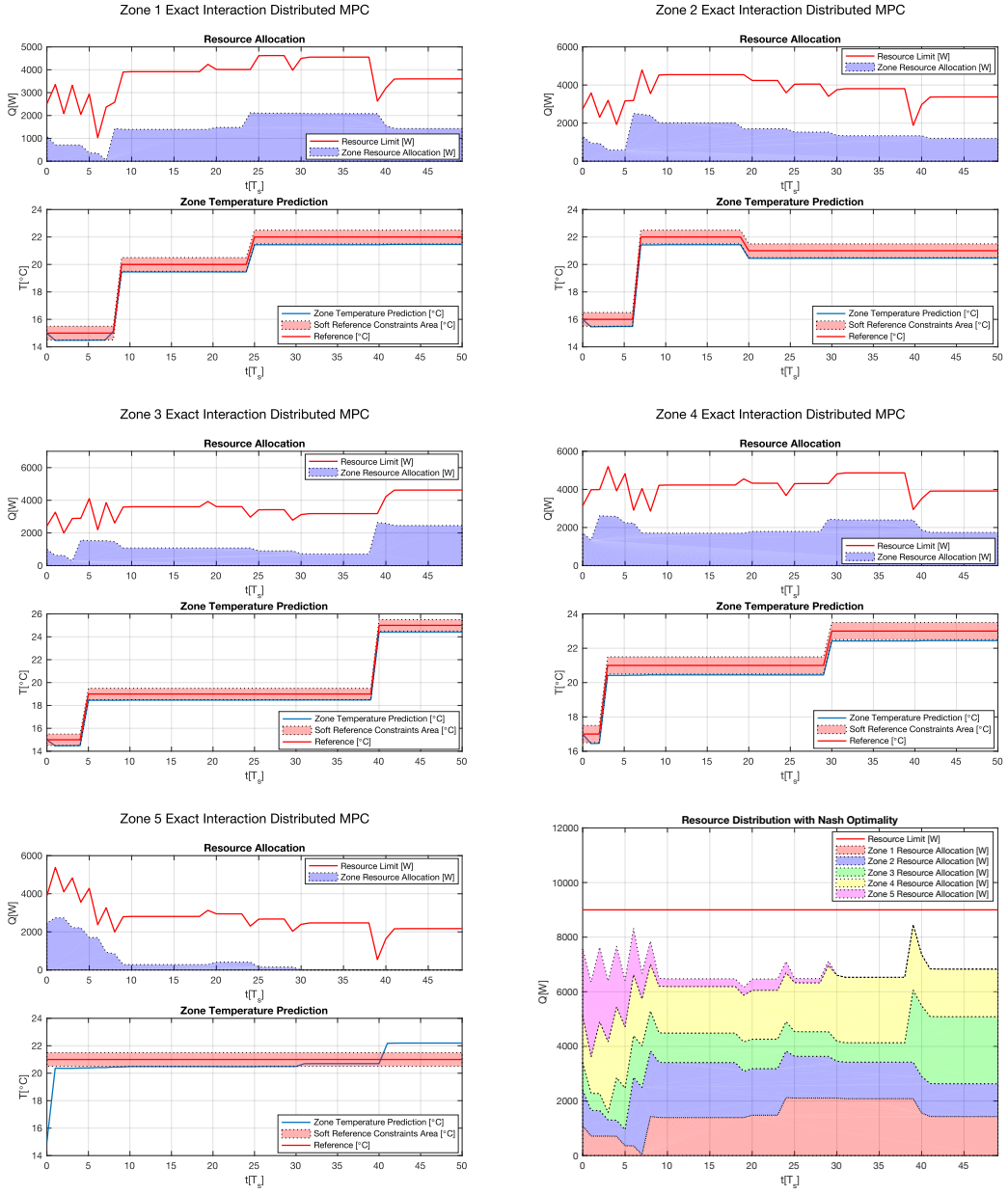


Figure 4.6: The five-zone benchmark using the exact interactions distributed MPC control approach is presented in this figure. Zones 1 to 5 are shown from top to bottom of this figure. Resource distribution solution using the Nash Optimality is shown as the bottom-right simulation. Inter-zone interactions are well observable in resource allocation part of each zone simulation. Zones reduce the interactions influence using their inputs what yields the appropriate reference tracking. An exception is the fifth zone, where input reached the lower bound and zone temperature violated the output soft constraint.

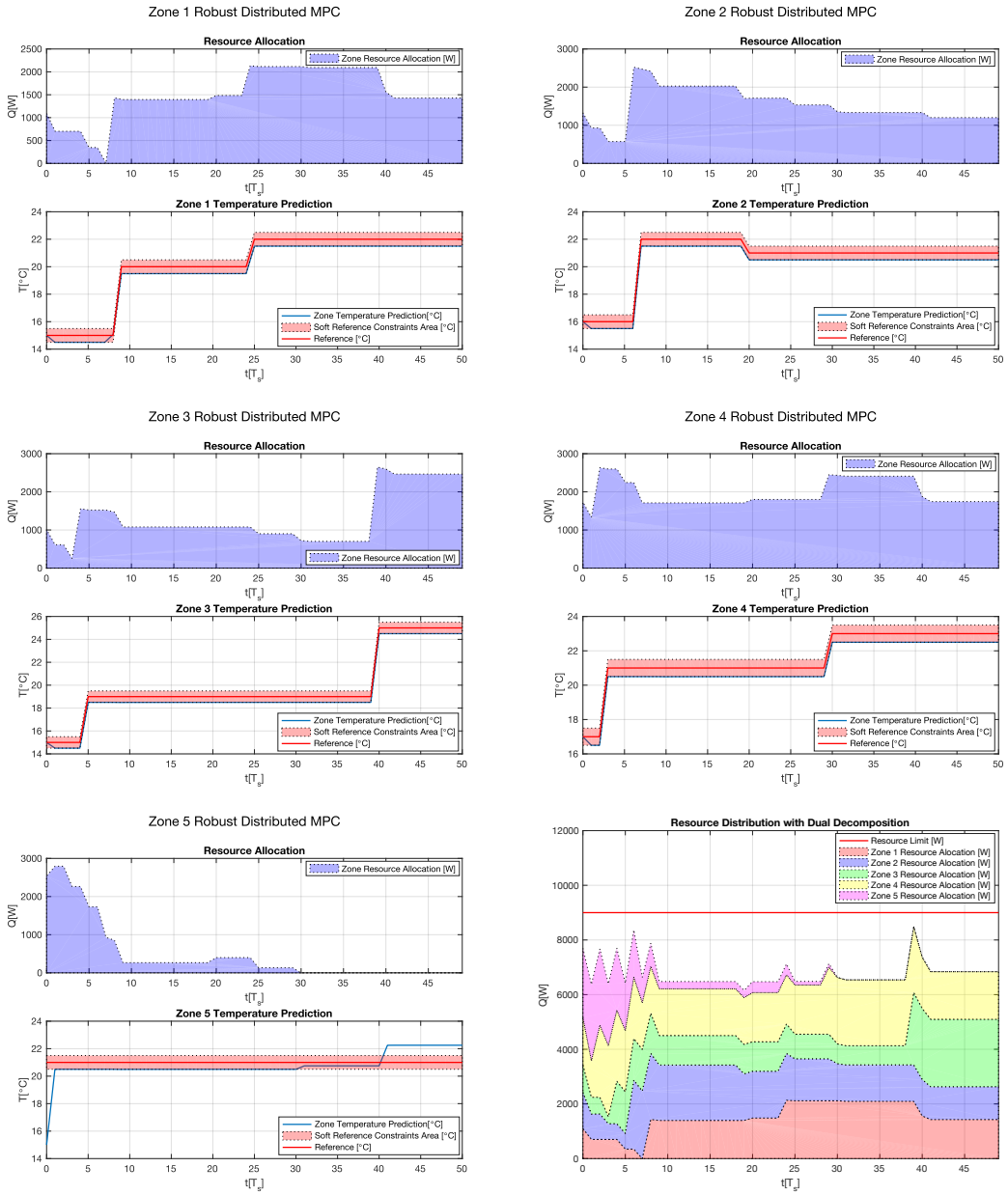


Figure 4.7: Simulation of the robust distributed MPC approach with dual decomsposition based resource distribution algorithm applied on the five-zone building benchmark is shown in this figure. As well as in the previous examples, five zones are presented from top to bottom of this figure. Each zone simulation contains a temperature prediction part and a resource allocation part. The last right-bottom simulation presents the resource distribution algorithm based on dual decomposition technique. Heat resource was bounded with value $u_{max} = 9\text{kW}$. This resource amount is enough to control each zone temperature properly. This solution was achieved in 191 iterations using the Nesterov's algorithm for step size rule with a specific value $\alpha_k = \frac{1}{k}$. The stop condition value is as follows, $\varepsilon = 0.1$.

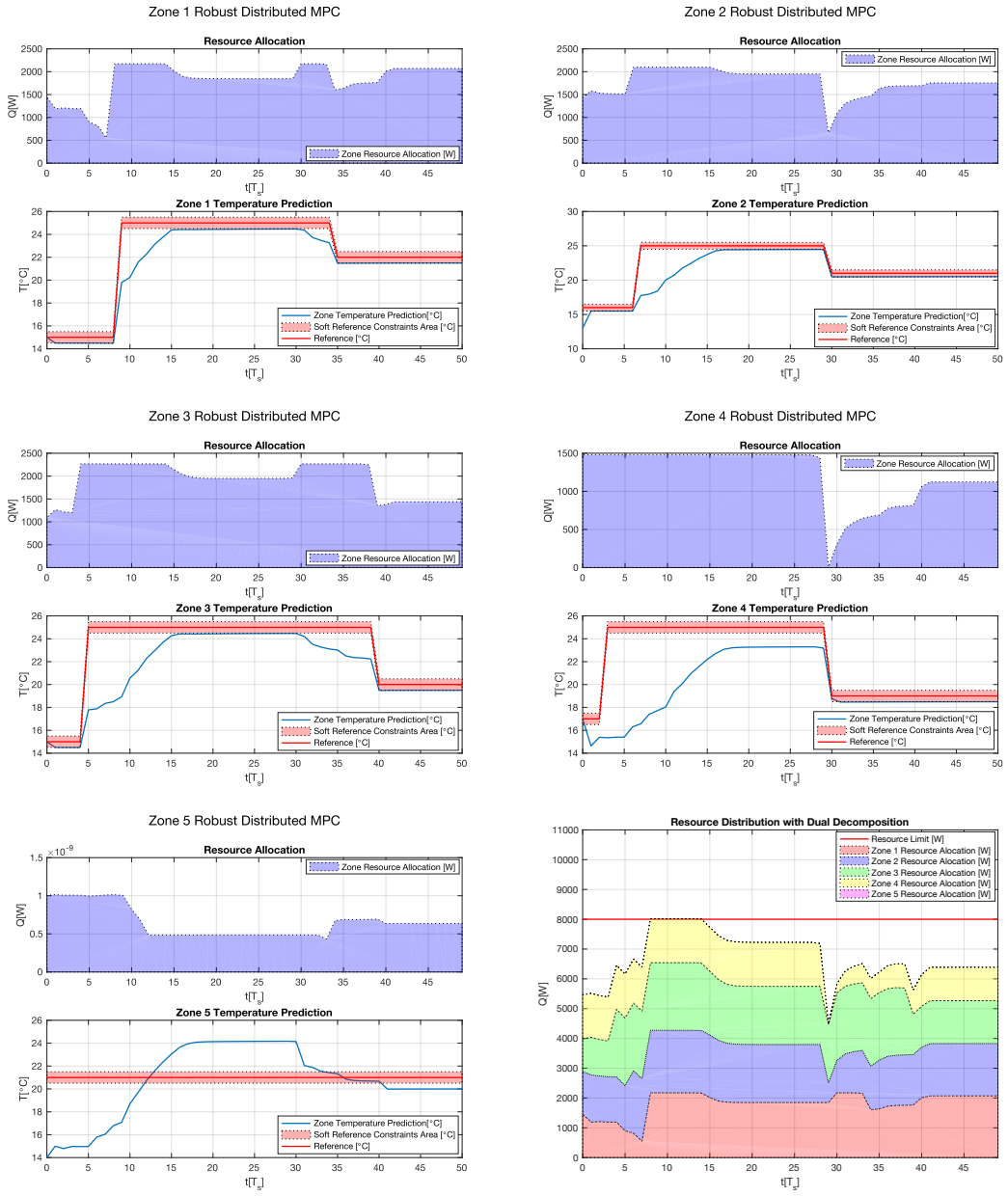


Figure 4.8: The second simulation case for the robust distributed MPC approach is presented in this figure. The five zones are presented from top to bottom of this figure in numerical order. Right-bottom simulation presents the resource distribution with input upper limit $u_{max} = 8$kW. Different zone temperature references were set in comparison to the other robust distributed MPC simulations due to deliberate fluctuation of resource requirements. This solution was achieved in 435 iterations using the Nesterov's algorithm for step size rule as well as the other robust distributed MPC cases.

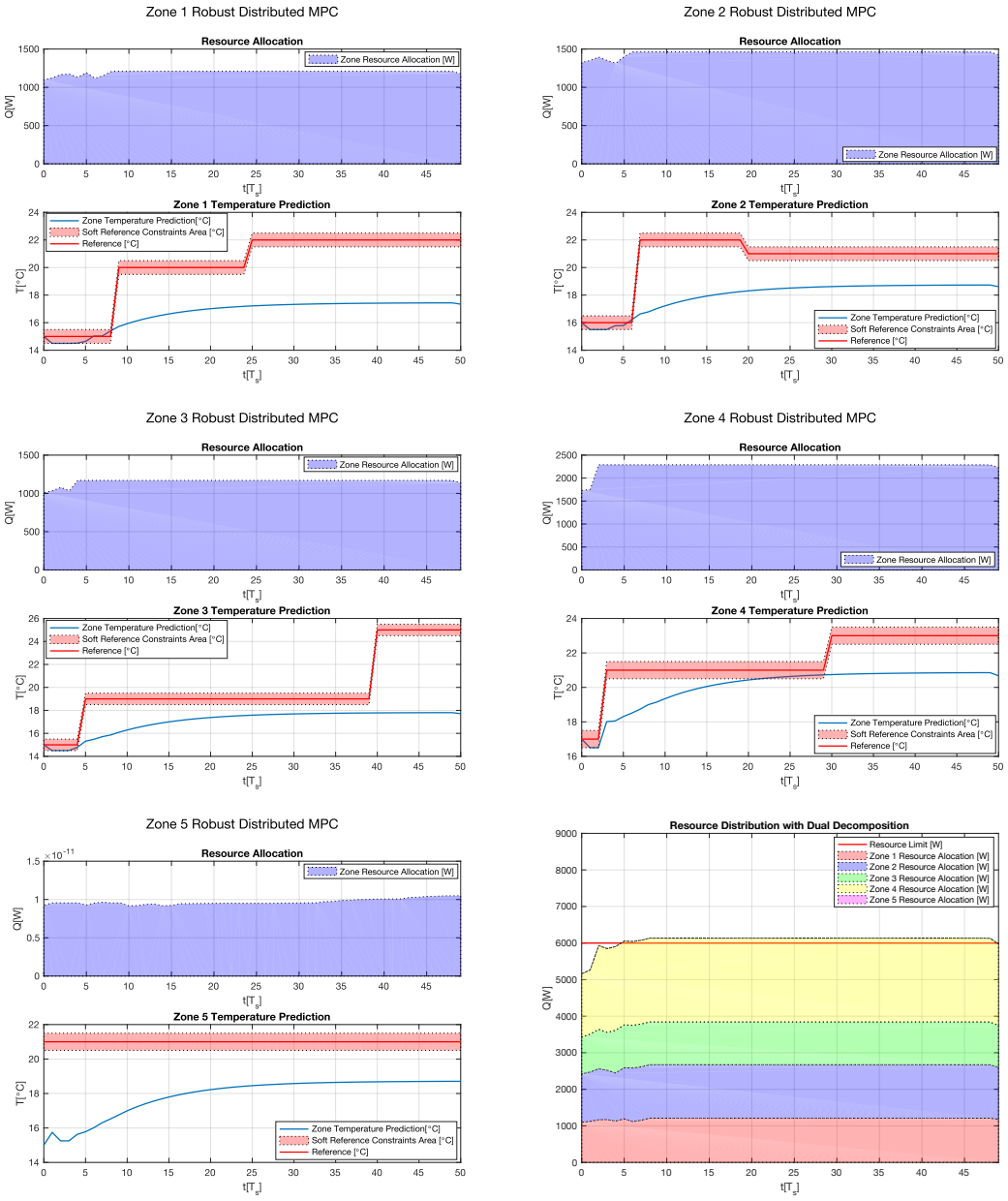


Figure 4.9: The third robust distributed MPC simulation with dual decomsposition based resource distribution algorithm is shown in this figure. The five-zone building benchmark zones simulations are presented from top to bottom of this figure. Simulation of the resource distribution algorithm based on dual decomposition technique is situated in the bottom-roght position and presents behaviour for input upper limit $u_{max} = 6\text{kW}$. However, this resource limitation cause inefficient reference tracking, it is interesting to compare it to the centralized approach, see Fig. 4.5. This solution was achieved in 3961 iteration steps.

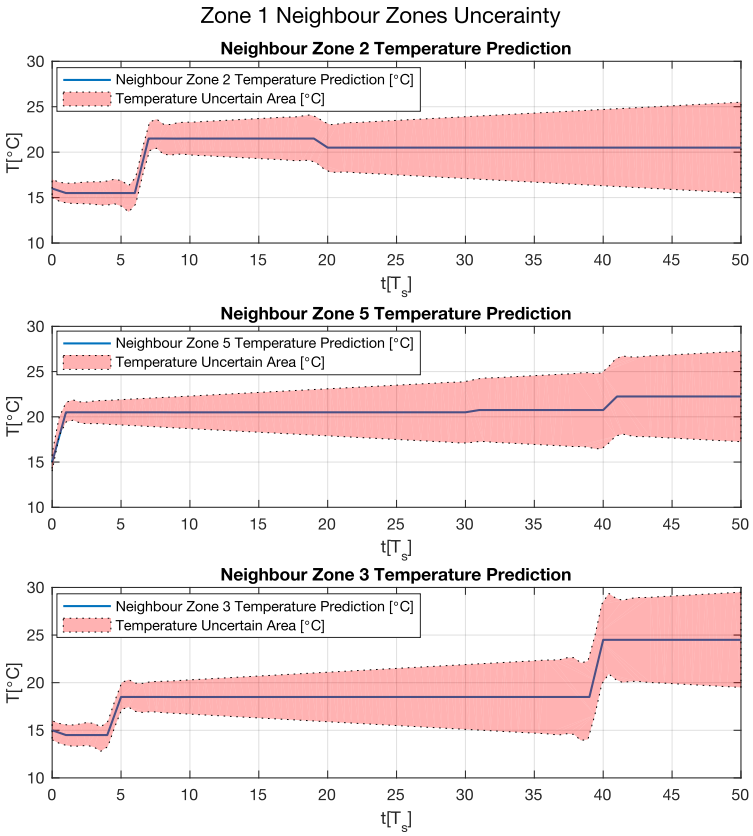


Figure 4.10: This figure shows the local robust MPC uncertainty behaviour for zone 1 from the five-zone building benchmark using the distributed MPC approach with dual decomposition based resource distribution algorithm from sec. 4.2.4. See the increasing neighbour zone temperature prediction uncertainty. This uncertainty increases with the prediction horizon. This figure is also comparable to the optimalized temperature predictions of zones 2,3 and 5 after 191 iterations from Fig. 4.7. These predictions may differ due to the robust interaction algorithm behaviour, where the predictions are communicated between zones each time step, not iteration step.

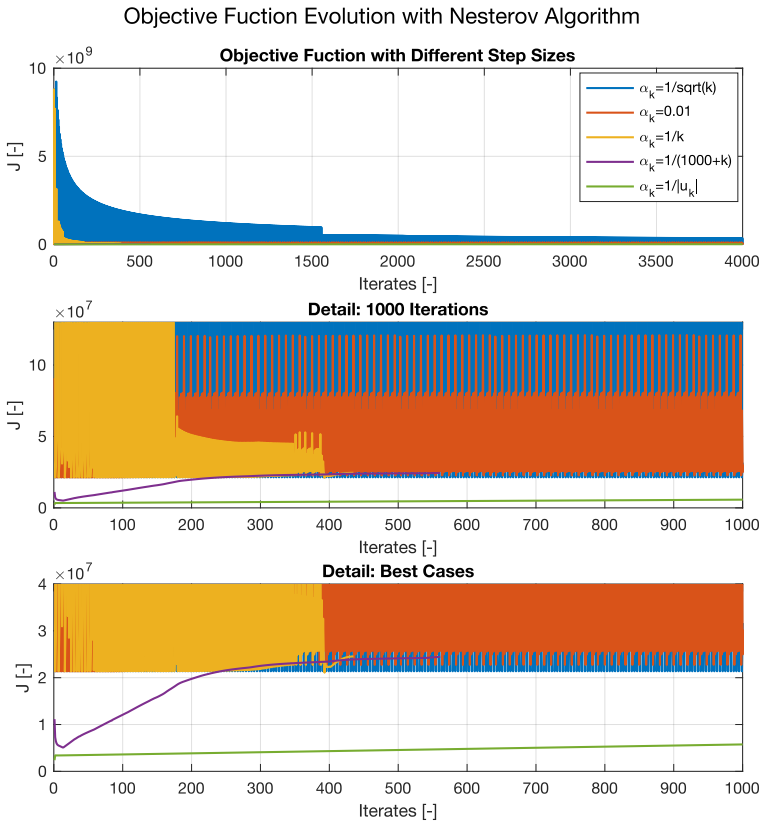


Figure 4.11: This figure presents the step sizes used in Nesterov's algorithm for one specific dual decomposition problem, which is defined in sec. 4.2.4. It was used for the robust distributed MPC with dual decomposition based resource distribution algorithm. All the step sizes converge to the same optimal value of J, they only differ in the speed of convergence (number of iterations respectivelly). Note that $\alpha_k = \frac{1}{k}$ or $\alpha_k = \frac{1}{1000+k}$ belongs to the faster step sizes, the first one finished in 435 iterations and the second one in 561 iteration steps as shown in the bottom best cases detail plot. On the other hand, $\alpha_k = \frac{1}{\sqrt{k}}$ or $\alpha_k = \frac{1}{|u_k|}$ belongs to the slower ones.

Chapter 5

Conlusion

The aim of this thesis was to analyze distributed predictive control for buildings. Generally, to introduce predictive control of large scale systems and compare the centralized a distributed control approaches.

At first, the building simulation model was introduced in chap. 2. The modelling part uses the thermal-electric analogy and shows a building model from a zone based perspective. The building model is divided into smaller zones which interacts. Two types of zone models are presented. One of them considers no wall capacity and is called R1C0 model, it's name is derived from the electric analogy. The other zone thermal model is called R2C1 and cosiders also wall capacity. Furthermore, this chapter presents the zone merging techniques with the final output in a form of a global building model concatenation Alg. 1.

Second, chap. 3 describes the necessary control theory for the various types of MPC. This chapter explains basic MPC, robust MPC and their centralized and distributed versions, each with an application in building control. The advantages and disadvantages of each control approach are discussed. Moreover, the distributed control approach of sec. 3.3 introduces the principles of decomposition. Both the primal and dual decomposition techniques are described, master problem and subproblems theory are exlained and coordinator role is mentioned. At last, the step size rules are presented. Another key point to bring out are the distributed control algorithms which are defined at the end of this chapter. One of the distributed control problems is the resource allocation, another one is the local zone control problem. This thesis offers two algorithms for solving the resource allocation problem, the Nash optimality Alg. 10 and the dual decomposition Alg. 11. Both the algorithms use iterative approach. The local zone control problem solution is of two types as well, the first one is the exact zone interactions Alg. 8 and the second one is the robust zone interactions Alg. 9. All the presented algorithms are then used in simulation examples to demonstrate their applications.

Thirdly, applications of the control mechanisms are presented in chap. 4. This chapter is opened by simple one-zone building simulations. Then the more complex five-zone

building benchmark is presented and it's setup is mentioned. With this in mind, three five-zone building benchmark model situations are simulated with intent to compare defined control algorithms from previous chapter. The main goal of these simulations is to show the control challenges, which have to be taken into account. Resource shortage, for instance, is one of them. The uniform objective function J_u was used as an comparison factor for all three algorithms, results are presented in Tab. 4.1. The centralized algorithm provides the best results and was evaluated as the most reliable. However, system complexity grows significantly with this algorithm even for the five-zone building benchmark. The distributed algorithms were considered as more interesting ones from a scientific point of view. Their usage requires understansing of the interative techniques as well as the decomposition structures. The dual decomposition technique was proven as more efficient than the exact interactions one for optimization of the resource distribution. Robust MPC together with stochastic systems theory is interesting extension for normal MPC approach and uncertainty plays a major role in handling of the communication noise and the other defects of zone interactions.

The more complicated and more complex R2C1 building zone model application can be taken as the purpose of a future work. The building zone interactions could be more desired and simulations could get closer to real applications. Moreover, the distributed control approach could handle more complex decomposition techniques such as subnets analysis. Next challenging enhancement would be more detailed analysis of high level step size rules as Nesterov's in order to optimalize the computational time of distributed algorithms using iterative approach.

Personally, I found implementation of distributed algorithms as a very interesting experience. Understanding of the iterative approaches and the decomposition structures is also beneficial for my personal development. I would like to point out that optimization of the distributed algorithms is time consuming operation which requires patience but the results are worth it.

Bibliography

[1] BÄUMELT, T., *Distributed Building Identification*, Thesis, Czech Technical Unversity in Prague, 2016, Prague: FEE ČTU, 2016, Print.

[2] LÖFBERG, J., *YALMIP : A Toolbox for Modeling and Optimization in MATLAB.* In Proceedings of the CACSD Conference, 2004, Taipei, Taiwan.

[3] LÖFBERG, J., *Automatic robust convex programming.* Optimization methods and software, 2012, Taylor & Francis.

[4] STURZENEGGER, D. C. T., *Model Predictive Building Climate Control - Steps Towards Practice* [online]. (n.d.): 437-62. Model Predictive Building Climate Control - Steps Towards Practice, 2014-01-01 [retrieved 2017-05-24]. Accessible from: http://e-collection.library.ethz.ch/

[5] TARAU A. N., DE SCHUTTER B., HELLENDOORN J., *Centralized, decentralized, and distributed model predictive control for route choice in automated baggage handling systems* [online]. Control Engineering and Applied Informatics, Special Issue on Distributed Control in Networked Systems, vol. 11, no. 3, pp. 24–31, 2009-01-01 [retrieved 2017-05-24], Accessible from: http://www.dcsc.tudelft.nl/~bdeschutter/pub/rep/09_024.pdf

[6] TRNKA, P., *Distributed Optimization and Model Predictive Control.* Honeywell Prague Laboratory, 2012. Prague: Honeywell.

[7] BOYD, S., *et al. Notes on Decomposition Methods* [online]. Stanford : Stanford University, 2007-02-12 [retrieved 2017-05-24]. Accessible from: http://www.stanford.edu/class/ee364b/notes/decomposition_notes.pdf.

[8] ENDEL, P., Distributed Predictive Control, *Thesis, Czech Technical University in Prague, 2012, Prague: FEE ČTU.*

[9] FARINA, M., FERRARI TRACATE, G., *Decentralized and distributed control* [online]. Graduate School on Control, Sup´elec, France, 2012-01-01 [retrieved 2017-05-24]. Available from: http://www.eeci-institute.eu/GSC2012/Photos-EECI/EECI-GSC-2012-M4/

[10] HAVLENA, V., ŠTECHA J., *Moderní teorie řízení.* Ediční středisko ČVUT, 1999, Prague.

[11] PATWARDHAN, S.C., *A Gentle Introduction to Model Predictive Control (MPC) Formulations based on Discrete Linear State Space Models.* Dept. of Chemical Engineering, I. I. T. Bombay, Powai, Mumbai.

[12] ZEILINGER, M., JONES, C., BORRELLI, F., MORARI, M., *Model Predictive Control* [online]. UC Berkeley, 2014-10-01 [retrieved 2017-05-24]. Accessible from: http://www.mpc.berkeley.edu/mpc-course-material

[13] BEMPORAD, A., MORARI, M., *Robust Model Predictive Control: A Survey.* Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH), Physikstrasse 3, CH-8092 Zürich, Switzerland.

[14] MACIEJOWSKI, J., *Model Predictive Control.* Cambridge University Engineering Department, EECI, TU Berlin, 2016-03-18.

[15] CAGIENARDA, R., GRIEDERA, P., KERRINGANB, E. C., MORARI M., *Move blocking strategies in receding horizon control.* Journal of Process Control (Volume 17, Issue 6, July 2007), Pages 563–570.

[16] MAESTRE, J. M., NEGENBORN, R. R., *Distributed Model Predictive Control Made Easy.* National Technical University of Athens, Athens, Greece, ISBN 978-94-007-7006-5.

[17] GIOVANINI, L., BALDERUD, J., *Game Approach To Distributed Model Predictive Control.* IET Control Theory & Applications (Volume: 5, Issue: 15, October 13 2011), Pages: 1729 - 1739, ISSN 1751-8652

[18] VENKAT, A. N., HISKENS, I. A.,RAWLINGS, J. B., WRIGHT, S. J., *Distributed MPC Strategies With Application to Power System Automatic Generation Control.* IEEE Transactions on Control Systems Technology (Volume: 16, Issue: 6, Nov. 2008), Pages: 1192 - 1206, ISSN 1558-0865.

Appendix A

Stochastic Systems

This chapter is a brief introduction into the stochastic system analysis [9]. The stochastic system is a system type which is influenced by the deterministic inputs and by the random (stochastic) inputs as well. The linear stochastic system's state equations can be derived from the well known deterministic system's state equations. This determinstc description has to be extended by the non-predictable or non-measurable parts of the system, these parts are also called the stochastic processes. After taking into consideration these random stochastic inputs, we inroduce the linear stochastic system described below by the state equations.

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{v}(t) \\ y(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) + \mathbf{e}(t), \end{aligned} \tag{A.1}$$

where $\mathbf{v}(t), \mathbf{e}(t)$ are called the process noise and the measurement noise.

Consider now that these random sequences are created by the random variables with the same probability distribution. Take into account also that these random variables are independent on the previous and actual states and inputs, they have zero mean value

$$\mathcal{E}\left\{ \begin{bmatrix} \mathbf{v}(t) \\ \mathbf{e}(t) \end{bmatrix} \right\} = 0, \tag{A.2}$$

and the covariance matrix is

$$\operatorname{cov}\left\{ \begin{bmatrix} \mathbf{v}(t_1) \\ \mathbf{e}(t_1) \end{bmatrix}, \begin{bmatrix} \mathbf{v}(t_2) \\ \mathbf{e}(t_2) \end{bmatrix} \right\} = \mathcal{E}\left\{ \begin{bmatrix} \mathbf{v}(t_1) \\ \mathbf{e}(t_1) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}(t_2) \\ \mathbf{e}(t_2) \end{bmatrix}^{\mathrm{T}} \right\} = \begin{bmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{S}^{\mathrm{T}} & \mathbf{R} \end{bmatrix} \delta(t_1 - t_2),$$

where

$$\delta(t_1 - t_2) = \begin{cases} 1 & t_1 = t_2 \\ 0 & t_1 \neq t_2 \end{cases} .$$

This stochastic process (sequence) is called the white noise.

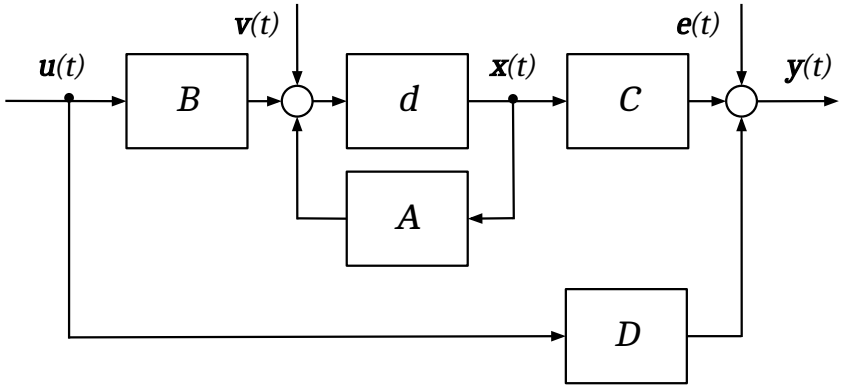


Figure A.1: The linear stochastic system model. Both the deterministic and stochastic parts are included in this model design. The stochastic processes $\mathbf{v}(t), \mathbf{e}(t)$ are represented as the white noise.

The bright insight into the stochastic systems states and outputs evolution in time is presented here as well. The states and outputs mean values and covariance matrices are used in this description.

$$\begin{aligned} \mu_x(t) &= \mathcal{E}\{\mathbf{x}(t)\} \\ \mu_y(t) &= \mathcal{E}\{\mathbf{y}(t)\} \end{aligned}$$

The mean value operator $\mathcal{E}$ is linear which implies

$$\begin{aligned} \mathcal{E}\{\mathbf{x}(t+1)\} &= \mathcal{E}\{\mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{v}(t)\} = \mathbf{A}\,\mathcal{E}\{\mathbf{x}(t)\} + \mathbf{B}\mathbf{u}(t) + \mathcal{E}\{\mathbf{v}(t)\} \\ \mathcal{E}\{\mathbf{y}(t+1)\} &= \mathcal{E}\{\mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) + \mathbf{e}(t)\} = \mathbf{C}\,\mathcal{E}\{\mathbf{x}(t)\} + \mathbf{D}\mathbf{u}(t) + \mathcal{E}\{\mathbf{e}(t)\}\,. \end{aligned}$$

It is possible to say with respect to (A.2) that the mean values of state and output change the same way as the states and output of deterministic system themselves.

$$\begin{aligned} \mu_x(t+1) &= \mathbf{A}\mu_x(t) + \mathbf{B}\mathbf{u}(t) \\ \mu_y(t+1) &= \mathbf{C}\mu_y(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \tag{A.3}$$

Next step is to derive the state covariance matrix evolution and the state-output covariance matrix. The state deviation from it's mean value is defined as

$$\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \mu_x(t)$$

and the state covariance matrix

$$\mathbf{P}_x = \mathrm{cov}\left\{\mathbf{x}(t)\right\} = \mathcal{E}\left\{\tilde{\mathbf{x}}(t)\tilde{\mathbf{x}}^{\mathrm{T}}(t)\right\}.$$

After the equations (A.1), (A.3) subtraction.

$$\begin{aligned} \tilde{\mathbf{x}}(t+1) &= \mathbf{A}\tilde{\mathbf{x}}(t) + \mathbf{v}(t) \\ \tilde{\mathbf{y}}(t) &= \mathbf{C}\tilde{\mathbf{x}}(t) + \mathbf{e}(t) \end{aligned}$$

The following equation is not using the mean value operator and considers the actual and previous states of noise independence.

$$\begin{aligned} \mathcal{E}\left\{\tilde{\mathbf{x}}(t+1)\tilde{\mathbf{x}}^{\mathrm{T}}(t+1)\right\} &= \mathcal{E}\left\{\mathbf{A}\tilde{\mathbf{x}}(t)\tilde{\mathbf{x}}^{\mathrm{T}}(t)\mathbf{A}^{\mathrm{T}} + \mathbf{A}\tilde{\mathbf{x}}(t)\mathbf{v}^{\mathrm{T}}(t) + \mathbf{v}(t)\tilde{\mathbf{x}}^{\mathrm{T}}(t)\mathbf{A}^{\mathrm{T}} + \mathbf{v}(t)\mathbf{v}^{\mathrm{T}}(t)\right\} \\ &= \mathbf{A}\mathbf{P}_x\mathbf{A}^{\mathrm{T}} + \mathbf{Q} \end{aligned}$$

and analogous operation for the state-output covariance

$$\begin{aligned} \mathcal{E}\left\{\tilde{\mathbf{x}}(t+1)\tilde{\mathbf{y}}^{\mathrm{T}}(t)\right\} &= \mathcal{E}\left\{\mathbf{A}\tilde{\mathbf{x}}(t)\tilde{\mathbf{x}}^{\mathrm{T}}(t)\mathbf{C}^{\mathrm{T}} + \mathbf{A}\tilde{\mathbf{x}}(t)\mathbf{e}^{\mathrm{T}}(t) + \mathbf{v}(t)\mathbf{C}^{\mathrm{T}}\tilde{\mathbf{x}}^{\mathrm{T}}(t) + \mathbf{v}(t)\mathbf{e}^{\mathrm{T}}(t)\right\} \\ &= \mathbf{A}\mathbf{P}_x\mathbf{C}^{\mathrm{T}} + \mathbf{S} \end{aligned}$$

$$\begin{aligned} \mathcal{E}\left\{\tilde{\mathbf{y}}(t)\tilde{\mathbf{y}}^{\mathrm{T}}(t)\right\} &= \mathcal{E}\left\{\mathbf{C}\tilde{\mathbf{x}}(t)\tilde{\mathbf{x}}^{\mathrm{T}}(t)\mathbf{C}^{\mathrm{T}} + \mathbf{C}\tilde{\mathbf{x}}(t)\mathbf{e}^{\mathrm{T}}(t) + \mathbf{e}(t)\mathbf{C}^{\mathrm{T}}\tilde{\mathbf{x}}^{\mathrm{T}}(t) + \mathbf{e}(t)\mathbf{e}^{\mathrm{T}}(t)\right\} \\ &= \mathbf{C}\mathbf{P}_x\mathbf{C}^{\mathrm{T}} + \mathbf{R}. \end{aligned}$$

The results are summarized by the following equations

$$\mathbf{P}_x(t+1) = \mathbf{A}\mathbf{P}_x(t)\mathbf{A}^{\mathrm{T}} + \mathbf{Q} \tag{A.4}$$

$$\mathrm{cov}\left\{\begin{bmatrix} \mathbf{x}(t+1) \\ \mathbf{y}(t) \end{bmatrix}\right\} = \begin{bmatrix} \mathbf{A}\mathbf{P}_x(t)\mathbf{A}^{\mathrm{T}} + \mathbf{Q} & \mathbf{A}\mathbf{P}_x(t)\mathbf{C}^{\mathrm{T}} + \mathbf{S} \\ \mathbf{C}\mathbf{P}_x(t)\mathbf{A}^{\mathrm{T}} + \mathbf{S}^{\mathrm{T}} & \mathbf{C}\mathbf{P}_x(t)\mathbf{C}^{\mathrm{T}} + \mathbf{R} \end{bmatrix}. \tag{A.5}$$

In a case when $\mathbf{v}(t), \mathbf{e}(t)$ are Gaussian noises and the initial state $\mathbf{x}(0)$ has also the Gaussian distribution, then the state and output of the system is also the Gaussian process and is characterized by the moments (A.3) and (A.5).

Also when the initial state $\mathbf{x}(0)$ has the moments $\mathcal{E}\{\mathbf{x}(0)\} = \mu_x(0)$ and $cov\{\mathbf{x}(0)\} = \mathbf{P}_x(0)$ then the solution of the equations (A.3), (A.4) can be explicitly expressed as

$$\begin{aligned} \mu_x(t) &= \mathbf{A}^t \mu_x(0) + \sum_{\tau=0}^{t-1} \mathbf{A}^{t-\tau-1} \mathbf{B}\mathbf{u}(\tau) \\ \mathbf{P}_x(t) &= \mathbf{A}^t \mathbf{P}_x(0) \left(\mathbf{A}^{\mathrm{T}}\right)^t + \sum_{\tau=0}^{t-1} \mathbf{A}^\tau \mathbf{Q} \left(\mathbf{A}^{\mathrm{T}}\right)^\tau . \end{aligned}$$

This definition using the series allows us to study the steady state if the matrix $\mathbf{A}$ is stable (i.e. $\mathrm{eig}(\mathbf{A}) < 0$). The steady state solution for the covariance state matrix is

$$\mathbf{P}_x = \lim_{t \to \infty} \mathbf{P}_x(t),$$

which suits the Lyapunov equation

$$\mathbf{P}_x = \mathbf{A}\mathbf{P}_x\mathbf{A}^{\mathrm{T}} + \mathbf{Q}.$$

The steady state output covariance matrix solution is then

$$\mathbf{P}_y = \mathbf{C}\mathbf{P}_x\mathbf{C}^{\mathrm{T}} + \mathbf{R}.$$

Appendix B

Contents of CD Attached

The CD attached to this thesis contains directories as follows:

- `master_thesis` - contains a pdf version of thesis
- `matlab_codes` - contains all the MATLAB m-files related to thesis