SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

Reg. No.: FCHPT-19990-50907

Model Predictive Control with Applications in Building Thermal Comfort Control

DISSERTATION THESIS

2017 **Ing. Ján Drgoňa**

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

Reg. No.: FCHPT-19990-50907

Model Predictive Control with Applications in Building Thermal Comfort Control

DISSERTATION THESIS

Study programme:	Process Control
Study field:	5.2.14. Automation
Department:	Institute of Information Engineering, Automation, and Mathematics
Thesis supervisor:	doc. Ing. Michal Kvasnica, PhD.

2017 **Ing. Ján Drgoňa**

Slovak University of Technology in Bratislava
Faculty of Chemical and Food Technology
Institute of Information Engineering, Automation

DISSERTATION THESIS TOPIC

Author of the thesis:	Ing. Ján Drgoňa
Study programme:	Process Control
Study field:	5.2.14. Automation
Registration number:	FCHPT-19990-50907
Student's ID: :	50907
Thesis supervisor:	doc. Ing. Michal Kvasnica, PhD.
Title of the thesis:	Model Predictive Control with Applications in Building Thermal Comfort Control
Language of thesis:	English
Date of entry:	02. 07. 2012
Date of submission:	01. 06. 2017

Ing. Ján Drgoňa
Solver

prof. Ing. Miroslav Fikar, DrSc.
Head of department

prof. Ing. Miroslav Fikar, DrSc.
Study programme supervisor

Acknowledgments

In the first place, I would like to express my special appreciation and thanks to my PhD supervisor prof. Michal Kvasnica. I would like to thank you for constant encouragement in my research, for maintaining a high motivation and for extraordinary patience with myself. I will always remember those challenging and exciting times we spent together in front of the whiteboard while explaining your ideas and sharing your immense knowledge. I could not have imagined having a better advisor and mentor for my PhD study.
I would like to genuinely thank prof. Miroslav Fikar, the head of the department, for his helpfulness, for maintaining the high standards and for an excellent working environment. My thank also go to prof. Alajos Mészáros and prof. Monika Bakošová for their openness and support during my studies. I thank my fellow colleagues, co-authors and friends, namely Martin Klaučo, Juraj Holaza, Bálint Takács, Martin Jelemenský, Juraj Oravec, Deepak Ingole, Ayush Sharma, Filip Janeček, Martin Kalúz and Richard Valo for the stimulating discussions, for the long evenings we were working together before deadlines, and for all the fun we have had during our studies. Many thanks also go to the whole department, specifically to technical staff for their tireless and hidden but vital work. I would like to also thank prof. Boris Rohál-Il'kiv and colleagues from Faculty of Mechanical Engineering, as well as prof. Danica Rosinová from Faculty of Electrical Engineering, for inspiring conversations during our numerous shared seminars on selected topics.
My sincere thanks also go to prof. Lieve Helsen who provided me with an opportunity to join their "sysi" group at KU Leuven as visiting researcher. I can say without any hesitation that I could not join a better team. Also, I thank all my friends at KU Leuven for creating scientifically stimulating and very friendly environment. In particular, I am grateful to Damien Picard for fruitful collaboration, refreshing ideas from different perspectives and for making my research visit at KU Leuven such a success.
And last, but not least, let me also thank all my friends and family in my mother's tongue. Chcel by som sa poďakovať všetkým mojim priateľom za ich podporu, priateľstvo a inšpiráciu na riešenie, alebo náhľad do mnohých praktických ako aj filozofických problémov. A v neposlednom rade by som chcel z celého srdca poďakovať mojim rodičom, starým rodičom, setre a všetkým členom rodiny za ich silnú podporu a starostlivosť počas celého môjho štúdia, ako aj počas celého života. Ďakujem.

Abstract

This thesis deals with applications of model predictive control (MPC) on the building climate control problems. Many studies have proved that building sector can significantly benefit from replacing the current practice rule-based controllers (RBC) for more advanced control strategies like MPC. Despite this intensive research, the application of the MPC in practice is still in its early stages. This is mainly because the MPC requires an accurate controller model of the building envelope and its heating, ventilation and air conditioning (HVAC) systems. However, the necessary level of the model complexity to obtain a good MPC performance remains a priori unknown, and no systematic method is available. This thesis introduces such systematical investigation of the required controller model complexity necessary to obtain the optimal control performance for a given building. Moreover, the optimization-based control algorithms, like MPC, impose increased hardware and software requirements, together with more complicated tuning and error handling capabilities required from the commissioning staff. This problem is tackled in this thesis, by two ways. First, it is shown how the explicit solutions can be synthesized even for the MPC formulations taking into account uncertainties in the weather predictions. The main bottleneck of this approach, however, are its limitations only to the problems of modest complexity. This drawback is further eliminated by introducing a versatile framework for synthesis of simple, yet well-performing control strategies that mimic the behaviour of optimization-based controllers, also for the large scale multiple-input-multiple-output (MIMO) control problems which are common in the building sector. The idea is based on devising simplified control laws learned from MPC by exploiting the powers of multivariate regression algorithms and dimensionality reduction techniques. The main advantage of the proposed methods stems from their easy implementation even on low-level hardware without the need for advanced software libraries.

Abstrakt

Táto práca sa zaoberá aplikáciami preditívneho riadenia (MPC) na problematiku riadenia tepelnej pohody v budovách. Mnohé štúdie ukázali, že nahradenie aktuálne používaných regulátorov založených na pravidlách (RBC) pokročilými metódami riadenia môže výrazne prispieť k energetickým úsporám a zvýšenému komfortu obyvateľov. Aj napriek tomuto intenzívnemu výskumu aplikácia MPCv praxi je ešte stále v počiatočnom štádiu. A to hlavne z toho dôvodu, pretože MPC vyžaduje presný matematický model budovy a jej vykurovacích, chladiacih a klimatizačných (HVAC) systémov. Potrebná úroveň zložitosti modelu na dosiahnutie dobrého výkonu MPC však zostáva a priori neznáma a na toto určenie nie je k dispozícii žiadna systematická metóda. Táto práca predstavuje systematickú štúdiu zložitosti predikčného modelu potrebného na dosiahnutie optimálneho správania sa regulátora pre konkrétnu budovu. Druhou prekážkou aplikácie algoritmov prediktívneho riadenia sú zvýšené požiadavky na hardvér a softvér a to hlavne z dôvodu nutnosti riešenia optimalizačných problémov v reálnom čase. Pokročilé algoritmy riadenia navyše vyžadujú špeciálne vyškolených pracovníkov schopných ladiť a odstraňovať poruchy pri zavádzaní tejto technológie do praxe. Tento problém je riešený v tejto práci dvomi spôsobmi. V prvom prípade skúmame možno zostrojiť takzvané explicitné riešenia aj pre formulácie MPC ktoré zohľadňujú neistoty predpovedí počasia. Takéto riešenia sú najprv predpočítané v režime off-line a v on-line režime potom umožňujú jednoduchú a výpočtovo efektívnu implementáciu preditívneho riadenia aj na zariadeniach s obmedzeným výpočtovým výkonom. Hlavnou prekážkou tohto prístupu sú však striktné obmedzenia na zložitosť riešeného problému, vyjadrené počtom parametrov. Táto nevýhoda je v druhom prístupe prekonaná zostrojením regulátorov ktoré napodobňujú správanie prediktívneho regulátora cez využitie algoritmov strojového učenia. Konkrétne ide o využitie algoritmov mnohorozmerovej regresie a techník redukcie zložitosti. Takéto regulátory sú navyše zostrojiteľné aj pre zložité problémy s viacerými vstupmi a výstupmi (MIMO) a s veľkým množstvom parametrov, ktoré sú bežné pri riadení budov. Hlavná výhoda navrhovaných metód spočíva v ich ľahkej implementácii aj na lacných zariadeniach bez potreby pokročilých softvérových knižníc.

Notation

Mathematical Symbols

x_k	state variable at k-th time step
u_k	input variable at k-th time step
d_k	disturbance variable at k-th time step
y_k	output variable at k-th time step
ω	random variable
ξ	vector of parameters in control context / features in machine learning context
$\mathbb{N}_a^b$	set of integers $\{a, a+1, \ldots, b\}$
$\mathbb{N}^n$	column vector of integer values of length n
$\mathbb{R}^n$	column vector of real values of length n
$\mathbb{R}^{n \times m}$	matrix of real values of n-rows and m-columns

Abbreviations

Optimization

LP	linear programming
QP	quadratic programming
MIP	mixed integer programming
MILP	mixed integer linear programming
MIQP	mixed integer quadratic programming
mpP	multi parametric programming
mpLP	multi parametric linear programming
mpQP	multi parametric quadratic programming
mpMILP	multi parametric mixed integer linear programming

Functions

PWA	piecewise affine function
PWQ	piecewise quadratic function
PWC	piecewise constant function

Systems

ODE	ordinary differential equation
I/O	input - output
SISO	single input - single output system
SIMO	single input - multiple output system
MISO	multiple input - single output system
MIMO	multiple input - multiple output system
LTI	linear time-invariant
SSM	state space model
MOR	model order reduction
ROM	reduced order model
HSV	hankel singular values

Controllers

RBC	rule based controller
PID	proporcional integral derivative
LQR	linear quadratic regulator
LQE	linear quadratic estimator
LQG	linear quadratic gaussian
MPC	model predictive control
RHC	receding horizon control
OSF-MPC	off-set free model predictive control
S-MPC	standard model predictive control
MPHC	model predictive heuristic control
MAC	model algorithmic control
DMC	dynamic matrix control
QDMC	quadratic dynamic matrix control
IDCOM	identification and command
IDCOM-M	identification and command - multiple input/output
HIECON	hierarchical constraint control
SMCA	setpoint multivariable control architecture
SMOC	shell multivariable optimizing controller
PCT	predictive control technology
RMPCT	robust model predictive control technology
CEMPC	certainty equivalence model predictive control

Building modeling and control

HVAC	heating ventilation air conditioning
BAS	building automation system
BACS	building automation and control system
ISE	indoor temperature simulink engineering
BES	building energy simulation
IDEAS	Integrated District Energy Assessment by Simulation
ACH	maximum volume air change per hour
U-value	overall heat transfer coefficient

List of Figures

List of Tables

Contents

Chapter 1

Introduction

The beginning is the most important part of the work.

Plato, The Republic

1.1 Building Control Overview and Challenges

The total energy consumed in heating, cooling, ventilation and air-condition (HVAC) systems in commercial and residential buildings nowadays account for 40 % of global energy consumption (Parry et al. 2007). In Europe, this figure is reported to be as high as 76 %. Any reduction of energy demand thus has a huge effect, which goes hand-to-hand with reduction of greenhouse gases and overall level of pollution.

Two major ways can be followed to lower the energy consumption of HVAC systems for buildings (McQuiston et al. 2005). One option is to focus on a better physical construction, for instance by using better insulations, or by devising an energy-friendly structure of the building. An obvious downside of these approaches is that they require significant resources and are mainly applicable only to newly-constructed buildings.

The second principal way is to improve the efficacy of HVAC control systems (Levermore 2000). Various control methods are nowadays available to achieve this goal. They range from use of the classical PID and state-feedback controllers (Canbay et al. 2004), through methods based on artificial intelligence concepts such as fuzzy systems (Hamdi and Lachiver 1998), neural networks (Kusiak and Xu 2012), machine learning (Liu and Henze 2007), multi-agent control systems (Dounis and Caraiscos 2009), up to the application of the optimal control algorithms (Ma et al. 2009, Široký et al. 2011). The advantage of the latter class is that the various performance criteria can be rigorously stated as an optimization problem, leading to a best possible performance. Numerous studies reported that advanced optimization-based HVAC control could significantly reduce the energy consumption and mitigate emissions of greenhouse gases, see e.g. Castilla et al. (2014), Gyalistras et al. (2010),

Roth et al. (2002). However, currently, the majority of the building control strategies still adopts only simple rule-based logic with limited energy savings capabilities (Aghemo et al. 2013, Mechri et al. 2010).

One of the control methods exploiting the full potential of the building's HVAC systems is model predictive control (MPC) (Maciejowski 2002). The high performance of MPC is achieved by accounting for minimization of consumed energy and maintaining high comfort standards while taking into account technological restrictions, weather forecasts and building dynamics. In MPC, control inputs that minimize a certain objective function (which accounts for consumption of energy and maximization of thermal comfort) subject to the constraints are computed by solving a corresponding optimization problem at each sampling instant. In recent years, many energy efficient MPC approaches have been reported for control of the HVAC systems (Ma et al. 2012a, Oldewurtel et al. 2010; 2012, Van Schijndel et al. 2008, Váňa et al. 2014, Široký et al. 2011).

Despite this intensive research efforts, the transfer of this technology to the commercial sector is still in its early stages, mainly because of the following four reasons as pointed out by Cigler et al. (2013). First, the accurate yet simple building model is required. However, to obtain such well-performing model with a minimum of effort is difficult and time-consuming task (Li and Wen 2014). Second, the design and tuning of MPC controllers are challenging, because the commission engineers are usually not trained to set up such complex control systems based on numerical optimization. Moreover, contrary to the industrial applications of MPC, buildings are not operated with on-site engineers monitoring and supervising the functioning of the employed control system. For these reasons, there is a significant requirement in this field for a simple implementation of the control algorithms without the loss of their high energy efficient performance (Domahidi et al. 2014). Third, there is also an strong need for data availability and processing power as the computation of MPC control actions for complex systems can be easily based on hundreds or even thousands of parameters. These variables are provided either by direct real-time measurements from the network of sensors, by external services like weather forecasts, or by state estimators. All of this makes the implementation of MPC even more challenging. And fourth, the on-line solution of the corresponding optimization problem and the extensive data processing impose considerable challenges on hardware and software infrastructure, which is not a standard in today's buildings.

Although some approaches for a fast and simple on-line implementation of MPC for building control applications have been suggested previously (Ma et al. 2011; 2012b), the task remains very challenging, especially when using existing control hardware, such as programmable logic controllers (PLC). There are two main difficulties. First, such a simple hardware provides only limited computational capabilities with a limited amount of memory storage (typically in the range of kilobytes). Secondly, most PLCs do not allow the control algorithm to be implemented in high-level languages. As a result, implementation of the complex, optimization-based control algorithms on a simple hardware is cumbersome (Huyck et al. 2012).

Here the ambition of this thesis is to tackle the last three challenges from the list of issues by constructing a simple yet well-performing control policy which offers a smooth

implementation suitable for a low-level hardware. One option for achieving this goal so is to calculate the explicit representation of the MPC feedback law (Bemporad et al. 2002, Borrelli 2003). For a rich class of MPC problems, the explicit solution takes a form of a piecewise affine (PWA) function defined over a polyhedral domain of the parametric space. Obtaining the optimal control input then reduces to a mere function evaluation. Such a task can be easily performed even by a simple hardware (Kvasnica et al. 2010b). The fundamental limitation of explicit MPC solution, however, is that the complexity of the computed PWA control law grows exponentially with the dimensionality of the parametric space imposed by prediction horizon and number of variables. Therefore it can be applied only on the hardware with storage capacity large enough to accommodate the PWA function. However, this is usually not a realistic assumption, since the size of explicit MPC solutions can easily exceed several megabytes even for the systems with low complexity, making it infeasible for complex building control problems with several thousand parameters.

The alternative way to tackle this problem is to employ the approximations of the MPC solution. The central idea here is not new. In fact, a variety of approximate explicit MPC solutions has been proposed in the literature (Alessio and Bemporad 2009). In general, there are two groups of approaches. First, the geometric methods which are based on efficient polyhedral partitioning of the state space (Bemporad and Filippi 2001, Grieder and Morari 2003, Pannocchia et al. 2007). And the second group are the data driven function approximation methods. The earliest work in this area was based on neural networks Parisini and Zoppoli (1995), while more recent works based on e.g. PWA (Bemporad et al. 2010), polynomial (Kvasnica et al. 2010a), and nonlinear (Domahidi et al. 2011) function approximations, or wavelet interpolations (Summers et al. 2009) have been reported as well.

One of the first attempts for the approximation of MPC laws in the building control context was introduced by Coffey (2013). This method is based on linear interpolations of MPC solutions generated for a grid of selected parameters. However, the complexity of such grid approach is increasing exponentially with the number of parameters, which strongly limits its applicability to large scale problems. Other researchers (Domahidi et al. 2014, Le et al. 2014, May-Ostendorp et al. 2011) used classification algorithms for extracting the simple decision rules from MPC employing logical control actions. An approach based on piecewise linear mixing architecture approximating the continuous control laws was proposed in Baldi et al. (2015). However, all the approaches listed above were developed and tested only on problems with modest complexity, usually with single (continuous or binary) control variable, and with only dozens of parameters.

1.2 Goals and Contributions of the Thesis

The academic goals of the thesis were summarized as follows:

- Comprehensive research in the field of building automation, and relevance evaluation of the integration of the MPC strategies in modern intelligent buildings.

- Development of efficient MPC strategies, tier algorithmic formulations, analysis and simulation studies on various building control problems.

- Experimental validation of developed algorithms on laboratory devices, or real world buildings with an integrated building automation system.

The first two goals were further elaborated in this thesis. The contribution is hence twofold. The first contribution is the investigation of the influence of the controller model accuracy on the performance evaluation of the building climate controllers. The results presented in this thesis are the first to systematically assess the performance of MPC using controller models of different orders without relying on system identification but using linearization and model order reduction techniques instead. This means that each reduced order model is *the best possible linear representation* of the building with that given number of states as each remaining state is optimally chosen by the model order reduction technique. This study shows that the minimum number of states of the controller model necessary to obtain the optimal control performance is higher than typical orders used in black- and grey-box methods. The most significant results of the author related to the evaluation of the integration of the MPC strategies in modern buildings are:

- Picard, D., **Drgoňa, J.**, Helsen, L., Kvasnica, M.: Impact of the Controller Model Complexity on MPC Performance Evaluation for Buildings. Energy and Buildings, 2017. (after 1st review, IF: 2.973)

- Picard, D., **Drgoňa, J.**, Helsen, L., Kvasnica, M.: Impact of the controller model complexity on MPC performance evaluation for building climate control. In The European Conference on Computational Optimization, Leuven, Belgium, vol. 4, 2016.

The second contribution of this thesis is the development of various advanced MPC strategies for buildings control applications. The problem of the uncertainty in weather prediction was tackled by means of stochastic control. This thesis further introduces a compact methodology for the construction of the simple suboptimal MPC-like control strategies for building control applications by using advanced machine learning algorithms. The focus is given on a creation of the systematic and universal framework applicable to a variety of large-scale building control problems while providing valuable insights into the selection of relevant features and appropriate type of the approximation model. In particular, the added value of this thesis lies in devising computationally tractable MPC approximations for complex building control problems for multiple-input-multiple-output (MIMO) systems with hundreds or even thousand of parameters. The most significant results of the author related to the development of efficient MPC strategies suitable for building control applications are:

- **Drgoňa, J.**, Picard, D., Helsen, L., Kvasnica, M.: Approximate Model Predictive Control for Complex Building Control Problems via Machine Learning, Applied Energy, 2017. (Submitted, IF: 5.746)

- **Drgoňa, J.**, Klaučo, M., Kvasnica, M.: MPC-Based Reference Governors for Thermostatically Controlled Residential Buildings. In IEEE Conference on Decision and Control (CDC), Osaka, Japan, vol. 54, pp. 1334-1339, 2015.
- Klaučo, M., **Drgoňa, J.**, Kvasnica, M., Di Cairano, S.: Building Temperature Control by Simple MPC-like Feedback Laws Learned from Closed-Loop Data. In Preprints of the 19th IFAC World Congress Cape Town, South Africa, pp. 581-586, 2014.
- **Drgoňa, J.**, Kvasnica, M., Klaučo, M., Fikar, M.: Explicit Stochastic MPC Approach to Building Temperature Control. In IEEE Conference on Decision and Control (CDC), Florence, Italy, pp. 6440-6445, 2013.
- **Drgoňa, J.**, Kvasnica, M.: Comparison of MPC Strategies for Building Control. In Proceedings of the 19th International Conference on Process Control, Slovak University of Technology in Bratislava, Štrbské Pleso, Slovakia, pp. 401-406, 2013.

The author has also participated in research covering other areas of the control design, however, that results are not elaborated in this thesis. Specifically, advances in the area of explicit MPC has been made by exploring the applicability of the region-less explicit MPC strategy. Results were published in:

- **Drgoňa, J.**, Klaučo, M., Janeček, F., Kvasnica, M.: Optimal control of a laboratory binary distillation column via regionless explicit MPC. *Computers & Chemical Engineering*, ISSN: 0098-1354, vol. 96, pg. 139–148, 2017. (IF: 2.581)
- **Drgoňa, J.**, Janeček, F., Klaučo, M., Kvasnica, M.: Regionless Explicit MPC of a Distillation Column. In IEEE 2016 European Control Conference (ECC), Aalborg, Denmark, pp. 1568-1573, 2016.

Furthermore, the author made contributions in the various process control applications, namely control of the laboratory devices like distillation column, continuous stirred-tank reactor and fuel cell plant. The results of this research were or will be published in:

- Holaza, J., Klaučo, M., **Drgoňa, J.**, Oravec, J., Kvasnica, M. and Fikar M.: MPC-Based Reference Governor Control of a Continuous Stirred-Tank Reactor. Computers & Chemical Engineering, 2017. (after 1st review, IF: 2.581)
- **Drgoňa, J.**, Takáč, Z., Horňák, M., Valo, R., Kvasnica, M.: Fuzzy Control of a Laboratory Binary Distillation Column, Accepted to the 21st International Conference on Process Control, Slovakia, 2017.
- Ingole, D., **Drgoňa, J.**, Kalúz, M., Klaučo, M., Bakošová, M., Kvasnica, M.: Model Predictive Control of a Combined Electrolyzer-Fuel Cell Educational Pilot Plant, Accepted to the 21st International Conference on Process Control, Slovakia, 2017.
- Ingole, D., **Drgoňa, J.**, Kalúz, M., Klaučo, M., Bakošová, M., Kvasnica, M.: Explicit Model Predictive Control of a Fuel Cell. In The European Conference on Computational Optimization, Leuven, Belgium, vol. 4, 2016.

- Sharma, A., **Drgoňa, J.**, Ingole, D., Holaza, J., Valo, R., Koniar, S., Kvasnica M.: Teaching Classical and Advanced Control of Binary Distillation Column. In 11th IFAC Symposium on Advances in Control Education, Bratislava, Slovakia, vol. 11, pp. 348-353, 2016.
- **Drgoňa, J.**, Klaučo, M., Valo, R., Bendžala, J., Fikar, M.: Model Identification and Predictive Control of a Laboratory Binary Distillation Column. In Proceedings of the 20th International Conference on Process Control, Slovak Chemical Library, Štrbské Pleso, Slovakia, 2015.

Full publication list of the author can be found in Appendix A.

1.3 Outline of the Thesis

This thesis is structured into two parts. The first part provides the theoretical backgrounds of the thesis and the second part introduces the theoretical contributions and investigates the simulation case studies. The theoretical part starts with the necessary mathematical background introducing the fundamental concepts of sets, functions, optimization, probability and statistic in Chapter 2. The introduction to the history and features of the model predictive control is given in Chapter 3. The second part of the thesis presents the various MPC strategies and their evaluation on simulation case studies. Chapter 4 describes the building modeling as a crucial part of the successful application of the MPC and introduces two particular models which are used in the simulation case studies for both, as controller models and emulating the building's behaviour. Chapter 5 discusses the building control nuances and defines the control objectives. Further, different types of the building climate controllers (BCC) are introduced in Chapter 5. Namely, a traditional rule-based-controller (RBC), a proportional-integral-derivative controller (PID), a deterministic MPC in standard form (S-MPC) and with an off-set free approach (OSF-MPC), stochastic formulation of the MPC problem and finally the approximate MPC controllers synthesized via machine learning algorithms. Chapter 6 introduces the application of proposed advanced MPC strategies formulated in the previous Chapter. The performance of the explicit stochastic MPC is studied in Section 6.1. The regression trees based approximations of the MPC are studied in Section 6.2.2. Section 6.3 investigates the impact of the controller model complexity on MPC performance evaluation for buildings and serves as the starting point for the next case study dealing with the approximate MPC for complex building control problems by using deep time delay neural networks, which is given in Section 6.4. Finally, Chapter 7 concludes the thesis and addresses the future research topics emanating from the studies presented in this thesis.

Part I

Theory

Chapter 2

Mathematical Background

Mathematics are well and good but nature keeps dragging us around by the nose.

Albert Einstein

2.1 Terminology and Definitions on Sets and Functions

This section provides basic terminology and definitions necessary for understanding following Sections 2.2, 2.3 and Chapter 3. Proofs for this section can be found in references e.g. Berger (1987), Grunbaum (2000), Schneider and Eberly (2003), Webster (1995), Weisstein (2014).

2.1.1 Sets

Definition 2.1.1 (ϵ-ball) *Or open n-dimensional ϵ-ball $\in \mathbb{R}^n$ around a given central point x_c is a set defined as*

$$\mathcal{B}_\epsilon(x_c) = \{X \in \mathbb{R}^n : \|x - x_c\|\} \tag{2.1}$$

where the radius $\epsilon > 0$ and $\| \bullet \|$ stands for any vector norm. □

Definition 2.1.2 (Neighborhood) *The neighborhood of a set $\mathcal{S} \subseteq \mathbb{R}^n$ is a set $\mathcal{N}(\mathcal{S})$ and $\mathcal{S} \subseteq \mathcal{N}(\mathcal{S}) \subseteq \mathbb{R}^n$ such that for each $s \in \mathcal{S}$ there exists n-dimensional ϵ-ball with $\mathcal{B}_\epsilon(s) \subseteq \mathcal{N}(\mathcal{S})$.* □

Definition 2.1.3 (Closed Set) *A set $\mathcal{S} \subseteq \mathbb{R}^n$ is closed if every point x which is not a member of $\mathcal{S}$ has a neighborhood disjoint from $\mathcal{S}$, or shortly*

$$\forall x \notin \mathcal{S} \; \exists \epsilon > 0 : \mathcal{B}_\epsilon(x) \cap \mathcal{S} = \emptyset \tag{2.2}$$

□

Definition 2.1.4 (Bounded Set) *A subset $\mathcal{S}$ of a metric space (M, μ) is bounded if it is contained in a ball $\mathcal{B}_r(\bullet)$ of finite radius r, i.e. if there $\exists x \in M$ and $r > 0$ such that $\forall s \in \mathcal{S}$, we have $\mu(x, s) < r$, or shortly $\mathcal{S} \subseteq \mathcal{B}_r(s)$* □

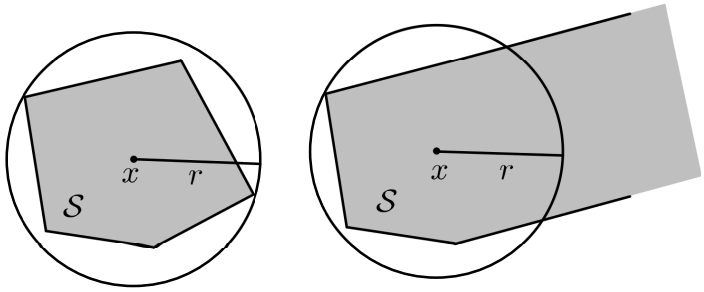


Figure 2.1: Bounded set (left) contained in a ball $\mathcal{B}_r(x)$ and unbounded set (right) uncontained in its entirety inside a ball $\mathcal{B}_r(x)$.

Definition 2.1.5 (Compact Set) *A set $\mathcal{S}$ is compact if it is closed and bounded.* □

Definition 2.1.6 (Null Set) *Let X be a measurable space, let μ be a measure on X, and let N be a measurable set in X. If μ is a positive measure, then N is null (zero measure) if its measure $\mu(N)$ is zero. If μ is not a positive measure, then N is $\mu-$ null if N is $|\mu|-$ null, where $|\mu|$ is the total variation of μ. Equivalently if every measurable subset $A \subseteq N$ satisfies $\mu(A) = 0$. For signed measures, this is stronger than simply saying that $\mu(N) = 0$. For positive measures, this is equivalent to the definition given above.* □

The empty set is always a null set, it is unique set having no elements, its size or cardinality is zero. For empty set we use common notations including $\varnothing$, and $\emptyset$. Graphical comparisson of feasible (non-empty) and infeasible (empty) sets is shown in Fig. 2.2.

Definition 2.1.7 (Convex Set) *A set $\mathcal{S} \subseteq \mathbb{R}^n$ is convex if for any two points $x_1, x_2 \in \mathcal{S}$ and parameter λ, with $0 \leq \lambda \leq 1$ following must hold*

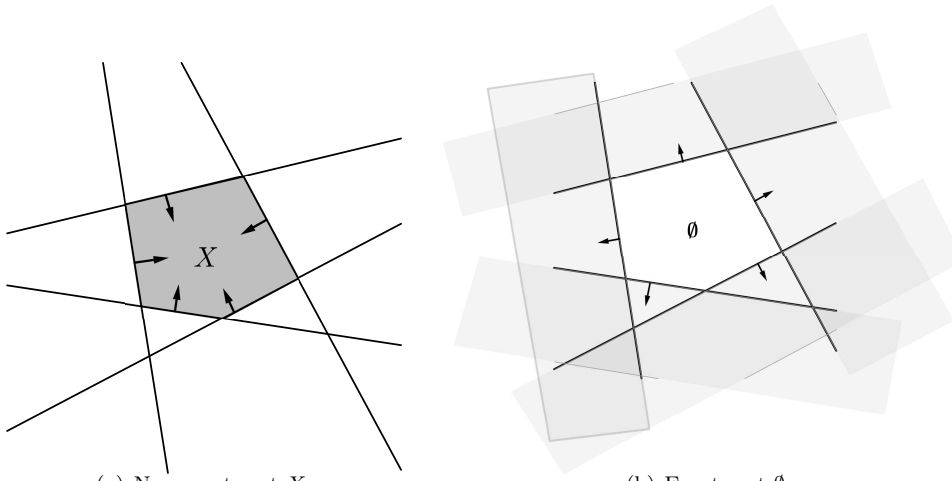
$$\lambda x_1 + (1 - \lambda) x_2 \in \mathcal{S} \tag{2.3}$$

In other words the line segment connecting any pair of points x_1, x_2 from $\mathcal{S}$ must lie entirely within $\mathcal{S}$. □

Definition 2.1.8 (Convex hull) *A convex hull of a finite set of points $\mathcal{V} = (v_1, \ldots, v_M)$, where $v_i \in \mathbb{R}^n$, $\forall i \in \mathbb{N}_1^M$, is the smallest convex set containing $\mathcal{V}$ defined as*

$$conv(\mathcal{V}) = \{\textstyle\sum_i \lambda_i v_i \ : \ \lambda \geq 0, \sum_i \lambda_i = 1\}. \tag{2.4}$$

□



(a) Non-empty set X. (b) Empty set $\emptyset$.

Figure 2.2: Non-empty and Empty set, constructed by intersection of 5 hyperplanes, represented by lines and their corresponding direction vectors.

Definition 2.1.9 (Set Collection) *A set $\mathcal{S} \subseteq \mathbb{R}^n$ is called a set collection if it is a collection of finite number of n-dimensional sets $\mathcal{S}_i$, i.e.*

$$\mathcal{S} = \{\mathcal{S}_i\}_{i=1}^{N_\mathcal{S}} \tag{2.5}$$

where $dim(\mathcal{S}_i) = n$ and $\mathcal{S}_i \subseteq \mathbb{R}^n$, for $i \in \mathbb{N}_1^{N_\mathcal{S}}$ with $N_\mathcal{S} < \infty$. A set collection of sometimes also referred to as family of sets. □

Definition 2.1.10 (Set Partition) *A collection of sets $\{\mathcal{S}_i\}_{i=1}^{N_\mathcal{S}}$ is a partition of a set $\mathcal{S}$ if $\mathcal{S} = \cup_{i=1}^{N_\mathcal{S}} \mathcal{S}_i$ and $\mathcal{S}_i \cap \mathcal{S}_j$ for all $i \neq j$, where $i, j \in \mathbb{N}_1^{N_\mathcal{S}}$* □

2.1.2 Functions

Definition 2.1.11 (Affine Function) *Let $f : \mathcal{S} \mapsto \mathbb{R}$ be real-valued function with $\mathcal{S} \in \mathbb{R}^n$, than function f acting on a vector x is affine, if it is of the form*

$$f(x) = Fx + g \tag{2.6}$$

Where multiplication of vector x by matrix $F \in \mathbb{R}^n$ represents a linear map, and addition of vector $g \in \mathbb{R}$ represents translation. Alternatively (Schneider and Eberly 2003) function f is called affine function or affine map, if and only if for every family $\{(x_i, \lambda_i)\}_{i \in I}$ of weighted points in $\mathcal{S}$, such that $\sum_{i \in I} \lambda_i = 1$ we have

$$f\left(\sum_{i \in I} \lambda_i x_i\right) = \sum_{i \in I} \lambda_i f(x_i) \tag{2.7}$$

In other words, f preserves center of mass. □

An affine transformation or affine map (from the Latin, affinis, "connected with") between two vector spaces is the composition of two functions a linear transformation or linear map, followed by a translation as shown in Definition 2.1.11. From geometrical point of view, these are precisely the functions that map straight lines to straight lines (Gallini 2014). Due to these properties affine functions play a vital role in mathematical optimization.

Definition 2.1.12 (Piecewise Affine Function) *Let $f_{PWA} : \mathcal{S} \mapsto \mathbb{R}$ be real-valued function with $\mathcal{S} \in \mathbb{R}^n$ than function f_{PWA} is a piecewise affine (PWA), if $\{\mathcal{S}_i\}_{i=1}^{N_\mathcal{S}}$ is a set partition of $\mathcal{S}$, with total number of partitions $N_\mathcal{S}$ and*

$$f_{PWA}(x) = F_i x + g_i, \qquad \forall x \in \mathcal{S}_i \tag{2.8}$$

Where $F_i \in \mathbb{R}^n$, $g_i \in \mathbb{R}$. □

Definition 2.1.13 (Piecewise Quadratic Function) *Let $f_{PWQ} : \mathcal{S} \mapsto \mathbb{R}$ be real-valued function with $\mathcal{S} \in \mathbb{R}^n$ than the function f_{PWQ} is piecewise quadratic (PWQ), if $\{\mathcal{S}_i\}_{i=1}^{N_\mathcal{S}}$ is a set partition of $\mathcal{S}$, with total number of partitions $N_\mathcal{S}$ and*

$$f_{PWQ}(x) = x^T E_i x + F_i x + g_i, \qquad \forall x \in \mathcal{S}_i \tag{2.9}$$

Where $E_i \in \mathbb{R}^{n \times n}$, $F_i \in \mathbb{R}^n$, $g_i \in \mathbb{R}$. □

Definition 2.1.14 (Convex Function) *Let $f : \mathcal{S} \mapsto \mathbb{R}$ be real-valued function, where $\mathcal{S} \in \mathbb{R}^n$ is nonempty convex set. Than the function f is convex on set $\mathcal{S}$ if for any two optimization variables $x_1, x_2 \in \mathcal{S}$, with parameter $0 \leq \lambda \leq 1$ following is true*

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2), \tag{2.10}$$

□

One special types of convex functions are called norms, which are assigning positive values representing lenghts to all non-zero vectors. Therefore tey are very useful for representation of distances between objects in vector spaces.

Definition 2.1.15 (Vector p-Norm) *The general notion of vector p-norm for vector $x \in \mathbb{R}^n$ or shortly $\|x\|_p$ is defined as*

$$\|x\|_p = \left(\sum_i |x|^p \right)^{1/p} \tag{2.11}$$

and holds following properties

- $\|x\|_p > 0$,
- $\|x\|_p = 0 \Leftrightarrow x = 0$,

- $\|cx\|_p = |c|\|x\|_p$, $\forall c \in \mathbb{R}$,
- $\|x_1 + x_2\|_p = \|x_1\|_p + \|x_2\|_p$.

□

Particular types of vector p-norms can be defined as follows.

Definition 2.1.16 (Vector 1-Norm) *Also called Manhattan norm. Computed as a sum of absolute values of vector's elements.*

$$\|x\|_1 = \sum_i^n |x_i|. \tag{2.12}$$

□

Definition 2.1.17 (Vector 2-Norm) *Also called Euclidean norm, representing shortest distance in euclidean space.*

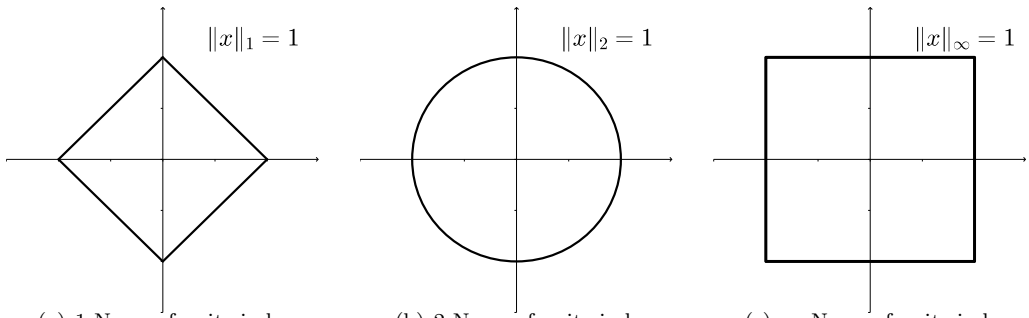
$$\|x\|_2 = \sqrt{\sum_i^n x_i^2} \tag{2.13}$$

□

Definition 2.1.18 (Vector ∞-Norm) *Computed as maximum absolute value of vector's elements.*

$$\|x\|_\infty = \max_{1 \le i \le n} |x_i| \tag{2.14}$$

□



(a) 1-Norm of unit circle. (b) 2-Norm of unit circle. (c) ∞-Norm of unit circle.

Figure 2.3: Illustrations of unit circles in different norms.

2.1.3 Polytopes

Are special types of sets, acting as the backbone of mathematical optimization, in this section will be provided some basic definitions on polytopes.

Definition 2.1.19 (Hyperplane) *Hyperplane $\mathcal{P} \in \mathbb{R}^n$ is a set in a form*

$$\mathcal{P} = \{x \in \mathbb{R}^n \ : a_i^T x = b_i\}, \tag{2.15}$$

Where $a_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$, $\forall i \in \mathbb{N}_1^m$. □

Definition 2.1.20 (Half-space) *Half-space $\mathcal{P} \in \mathbb{R}^n$ is a set in a form*

$$\mathcal{P} = \{x \in \mathbb{R}^n \ : a_i^T x \leq b_i\}, \tag{2.16}$$

Where $a_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$, $\forall i \in \mathbb{N}_1^m$. □

Definition 2.1.21 (Polyhedron) *Polyhedron $\mathcal{P} \in \mathbb{R}^n$ is the intersection of finite number of half-spaces, and can be compactly defined as follows*

$$\mathcal{P} = \{x \in \mathbb{R}^n \ : \ Ax \leq b\}, \tag{2.17}$$

where matrixes $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ are representing collection of intersecting affine half-spaces. Polyhedron also holds properties of a convex and closed set. □

Definition 2.1.22 (Polytope) *Set $\mathcal{P} \in \mathbb{R}^n$ is called a polytope if it is a bounded polyhedron.* □

Definition 2.1.23 (Polytope Representation) *in general there are two types of polytope representations, defined as*

- *$\mathcal{V}$-polytope $\mathcal{P} \subset \mathbb{R}^n$ is a convex hull of finite point set $\mathcal{V} = \{v_1, \ldots, v_M\}$, for $v_i \in \mathbb{R}^n$, $\forall i \in \mathbb{N}_1^M$, representing vertices of the polytope*

$$\mathcal{P} = \{x \ : \ x = \textstyle\sum_i^M \lambda_i v_i, \ 0 \leq \lambda_i \leq 1, \ \sum_i^M \lambda_i = 1\}, \tag{2.18}$$

- *$\mathcal{H}$-polytope is a bounded intersection of finite number half-spaces*

$$\mathcal{P} = \{x \in \mathbb{R}^n \ : \ Ax \leq b\}, \tag{2.19}$$

  where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$.

□

2.2 Mathematical Optimization

A mathematical optimization is an important tool in making decisions, and in analyzing physical systems applied in wide variety of scientific fields of study. Namely economics, operations research, electrical, chemical, mechanical and finally control engineering as the primary concern of this thesis. More comprehensive insight into the rich topic of mathematical optimization the reader can find in references such as Boyd and Vandenberghe (2004).

2.2.1 Taxonomy of Optimization

To provide a taxonomy of optimization is a tough task because of dense multiple connections between its subfields. One such comprehensive perspective focused mainly on the subfields of deterministic optimization problems with a single objective function can be found online in NEOS (2014) and is shown in Fig. 2.4.

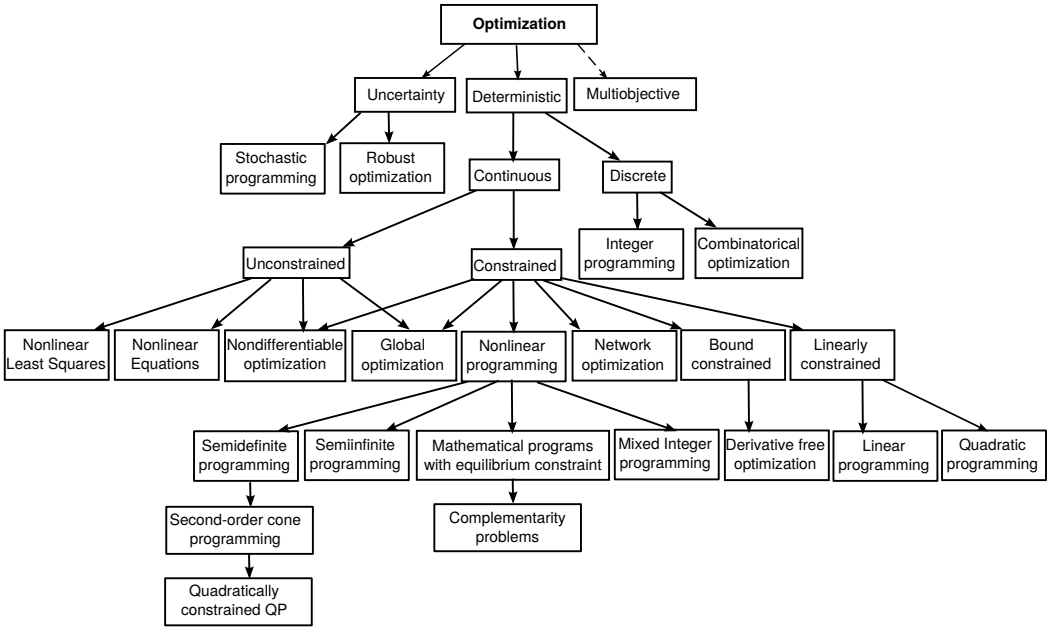


Figure 2.4: Classification of Optimization problems (NEOS 2014).

A wide collection of available optimization software and solvers organized by problem type can also be found on NEOS Server web-pages (NEOS-software 2014, NEOS-solvers 2014), together with valuable information about the group of algorithms, listed alphabetically or by problem type (NEOS-algorithms 2014).

2.2.2 Constrained Optimization

Constrained optimization is the process of optimizing an objective function w.r.t. some variables in the presence of constraints on those variables. Constrained optimization problems can be furthered classified according to the nature of the constraints (e.g., linear, nonlinear, convex) and the smoothness of the functions (e.g., differentiable or non-differentiable) NEOS (2014). For further reading see, e.g., references in Bertsekas (1996).

Standard Optimization Problem

In mathematical optimization terminology, a standard optimization problem is a fundamental notion representing the problem of finding the best solution among the group of all

possible and feasible solutions. The standard form of continuous constrained optimization problem is defined as follows (Boyd and Vandenberghe 2004).

$$J^* = \min_x f_0(x) \tag{2.20a}$$

$$\text{s.t. } g_i(x) \leq 0, \qquad i \in \mathbb{N}_1^m \tag{2.20b}$$

$$h_j(x) = 0, \qquad j \in \mathbb{N}_1^p \tag{2.20c}$$

Objective function also called *cost function* (2.20a), representing the first part of the problem (2.20). It is a real valued function with its domain $f_0 : \mathbb{R}^n \mapsto \mathbb{R}$, which to each optimized variable $x = (x_1, x_2, \ldots, x_n)^T$ assigns concrete real value $f_0(x)$ and which overall value has to be minimized during optimization. Maximization problem can be treated by the negation of the objective function.

Variables or the unknowns x are the components of the system which are being optimized and for which we want to find corresponding values. They can represent a broad range of quantities of the optimization problem, e.g.the amount of consumed resources or the time spent on each activity, whereas in data fitting, the variables would be the parameters of the model.

Constraints are representing an admissible set of values for optimized variables x, for which is given optimization problem feasible. In general, there are two types of constraints, inequality constraints defined as (2.20b), and equality constraints defined as (2.20c), merged by a notion of constraints set $\mathcal{S}$. More clarified classification of constraints for practical needs can be found in the documentation for MATLAB $Optimization Toolbox^{TM}$ (Mathworks 2014), listed with increasing complexity and required computing power from top to bottom:

- *Bound Constraints*, representing lower and upper bounds on individual components: $x \leq U$ and $x \geq L$.
- *Linear Equality and Inequality Constraints*, where $g_i(x)$ and $h_i(x)$ has a linear form.
- *Nonlinear Equality and Inequality Constraints*, where $g_i(x)$ and $h_i(x)$ has a non-linear form (e.g. integer constraints).

In most of the optimization problems the constraints satisfaction is mandatory, this kind of constraints which must be held during whole optimization procedure are also called *hard constraints*. However in some optimization problems can appear constraints which are preferred but not required to be satisfied, this kind of non-mandatory constraints are known as *soft constraints*, which are unique in having some additional *slack variables* that are penalized in the objective function.

Feasible region also called feasible set, search space, or solution space is the set of all possible values of the optimization variables x of a problem (2.20) that satisfy the problem's constraints. It can be perceived as an initial set of all candidate solutions to the problem before the set of candidates has been reduced by the optimization procedure. A *candidate solution*, therefore, must be unconditionally a member of the feasible set for a given problem.

Definition 2.2.1 (Feasible Set) *of problem* (2.20) *is defined as:*

$$X = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i \in \mathbb{N}_1^m, \quad h_j(x) = 0, j \in \mathbb{N}_1^p\} \tag{2.21}$$

A point x is said to be feasible for problem (2.20) *if it belongs to the feasible set X.* □

In general, a *feasible set* can be considered to be *bounded* in the sense of Definition 2.1.4 if it is in a certain sense of a finite size, or it can be regarded as *unbounded* if it contains points which values goes to infinity at least in one direction, as shown in Fig. 2.1. The problem with unbounded feasibility sets are that there may or may not be an optimum, with dependence on the objective function specifications. Thus a unique solution to the problem may not exist.
Difficulties also appear in the case, if there are no intersection of the problem's constraints, therefore there are no points that satisfy all the constraints simultaneously. Thus the feasible region is considered to be the null set in the sense of Def. 2.1.6, i.e. the case when the problem has no solution and is said to be *infeasible*.
Process of finding such a point in the feasible region is called *constraint satisfaction* and it is a crucial condition for finding the solution of constrained optimization problems.

Solution of an optimization problem is computed optimal value of the objective function, usually denoted by J^* or J^{opt}. As a solution is often considered also a minimizer (a vector which achieves that value), usually denoted as x^* or x^{opt}, if exists.
When the objective function is not provided, the problem (2.20) is being called a *feasibility problem*. Meant that we are just interested in determining the problem's feasibility, or in other words to find a feasible point. By convention, the cost function $f_0(x)$ is set to a constant $c \in \mathbb{R}$, to reflect the fact that we are indifferent to the choice of a point x as long as it is feasible.

Definition 2.2.2 (Optimal set of solutions) *of problem* (2.20) *is defined as the set of feasible points for which the objective function achieves the optimal value:*

$$X^* = \{x \in \mathbb{R}^n : f_0(x) = J^*, \quad g_i(x) \leq 0, i \in \mathbb{N}_1^m, \quad h_j(x) = 0, j \in \mathbb{N}_1^p\} \tag{2.22}$$

A standard notation for the optimal set is via the argmin notation:

$$X^* = \arg\min_{x \in \mathbb{X}} f_0(x) \tag{2.23}$$

A point x is said to be optimal if it belongs to the optimal set. If the problem is infeasible,

the optimal set is considered empty by convention. Thus existence of optimal points is not necessary. □

In mathematical optimization from a theoretical point of view, of view the notion of the optimal solution is crucial. However, for practical reasons, there has been established a weaker notion of suboptimal solution of the problem, representing points which are very close to the optimum. This is because most of the practical algorithms are iterative and are only able to compute suboptimal solutions, and never reach true optimality.

Definition 2.2.3 (Suboptimality) *more specifically the ϵ-suboptimal set is defined as*

$$X^{\epsilon} = \{x \in \mathbb{R}^n : f_0(x) = J^* + \epsilon, \quad g_i(x) \leq 0, i \in \mathbb{N}_1^m, \quad h_j(x) = 0, j \in \mathbb{N}_1^p\} \tag{2.24}$$

Where any point x in the ϵ-suboptimal set is termed ϵ-suboptimal and denoted x^{ϵ}. □

Constrained Optimization Problems Classes: Based on types of constraints $\mathcal{S}$ and objective function $f_0(x)$, constrained optimization covers a large number of subfields for which specialized algorithms are available, we will name some of the most important classes.

- *Bound Constrained Optimization*, where the constraints are only in the form of lower and upper bounds on the variables.
- *Linear Programming*, the objective function as well as all the constraints are linear functions.
- *Quadratic Programming*, the objective function is quadratic and the constraints are linear functions.
- *Semidefinite Programming*, the objective function is linear and the feasible set is the intersection of the cone of positive semidefinite matrices with an affine space.
- *Nonlinear Programming*, the objective function or at least some of the constraints are nonlinear functions.

In following sections, we will investigate differences between convex and non-convex optimization problems and their properties.

2.2.3 Convex Optimization

The optimization problem in a standard form (2.20) is called a convex optimization problem if:

- the objective function $f_0(x)$ is convex in the sense of Definition 2.1.14
- the constraint set $\mathcal{S}$ is convex in the sense of Definition 2.1.7

Convex problems are very popular and preferred in comparison with non-convex problems, due to their several advantages:

- Any local optimum is naturally also a global optimum, what guarantees that the global minimum of objective function will also be found.
- If there can not be found any global optimum, the problem can be labelled as infeasible.
- Convex problems are in contrast with non-convex problems easily solved, with a wide variety of suitable solvers.

However, the practical problems often exhibit non-convex properties. Hence convex problems are not always suitable framework for solutions to real-world problems, what is the main drawback of them. But where they can be applied, they used to be extremely efficient. In the following text, we will introduce two basic types of convex optimization problems with linear constraints, namely linear programming (LP) and quadratic programming (QP).

Linear Programming

problem is a convex optimization problem, which has a linear objective function (2.25a) with continuous real variables x subject to linear constraints (2.25b), (2.25c), and can be in general formulated as follows.

$$J^* = \min_x \quad c^T x \tag{2.25a}$$

$$\text{s.t.} \quad Ax \leq b \tag{2.25b}$$

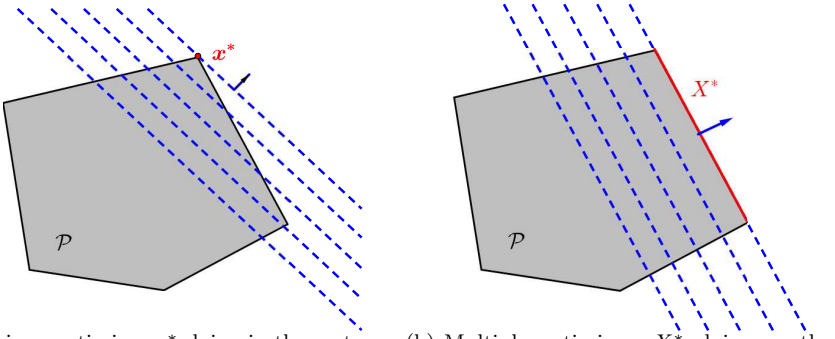
$$A_{\text{eq}} x = b_{\text{eq}}, \tag{2.25c}$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $A_{\text{eq}} \in \mathbb{R}^{p \times n}$ and $b_{\text{eq}} \in \mathbb{R}^p$. Hence the feasible region (see Def. 2.2.1) of such a problem is a convex polyhedron (see Def. 2.1.21), i.e. a region in multidimensional space, whose boundaries are formed by hyperplanes (see Def. 2.1.19) and whose corners are vertices.

Solution Properties: LP can be geometrically interpreted as searching for an optimum x^* of a linear objective function over a given polyhedral region $\mathcal{P}$. This procedure can result in several different scenarios.

1. *Feasible problem*, with value of the objective function $-\infty < J^* < \infty$, and two possible results:
   (a) Unique optimizer x^*, representing a single point.
   (b) Multiple optimizers X^*, representing a set of points $x^* \in \mathbb{R}^m$.
2. *Infeasible problem*, with value of the objective function $J^* = \pm\infty$, due to two reasons:
   (a) Polyhedral region $\mathcal{P}$ is an empty set, and $J^* = \infty$.
   (b) Polyhedral region $\mathcal{P}$ is unbounded set in direction of minimization of the objective function, and $J^* = -\infty$.

Graphical demonstrations of different optimization results of LP problem on two-dimensional space are shown in Fig. 2.5. Where Fig. 2.5(a) represents a unique solution x^*, which lies in a vertex of the region $\mathcal{P}$. And situation Fig. 2.5(b), when the objective function is parallel to one of the constraints with resulting multiple solutions X^* of equal quality, which lies on the edge of the region $\mathcal{P}$.



(a) Unique optimizer x^*, lying in the vertex of polyhedron $\mathcal{P}$.

(b) Multiple optimizers X^*, lying on the edge of polyhedron $\mathcal{P}$.

Figure 2.5: Different types of feasible solutions of LP. Where $\mathcal{P}$ represents constraints set, objective function is represented by dashed blue lines with its direction vector, and optimizers are depicted as a red dot for x^*, or red line for X^* respectively.

The strength of LP problems lies in their relative simplicity with a comparison to other classes of optimization problems, what allows the existence of a wide variety of solvers, allowing solving LP problems efficiently even for a large number of variables.

Quadratic Programming

problem is a convex optimization problem, which has a quadratic objective function (2.26a) with continuous real variables x subject to linear constraints (2.26b), (2.26c), and can be in general formulated as follows.

$$J^* = \min_x \tfrac{1}{2} x^T H x + q^T x + c \tag{2.26a}$$

$$\text{s.t.} Ax \leq b \tag{2.26b}$$

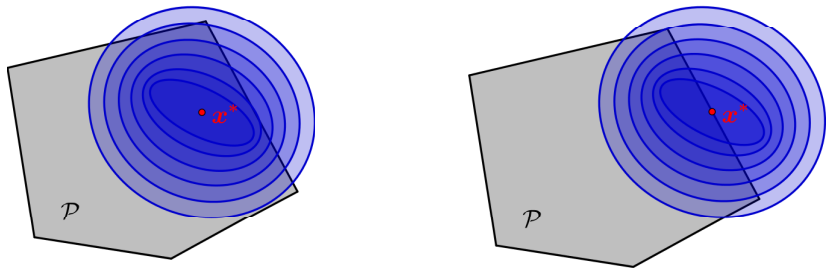
$$A_{\text{eq}} x = b_{\text{eq}}, \tag{2.26c}$$

where $x \in \mathbb{R}^n$, $H \in \mathbb{R}^{n \times n}$, $q \in \mathbb{R}^n$, $c \in \mathbb{R}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $A_{\text{eq}} \in \mathbb{R}^{p \times n}$ and $b_{\text{eq}} \in \mathbb{R}^p$. The difficulty of solving the QP problem depends largely on the nature of the matrix H. If matrix $H = H^T \succ 0$ is positive semidefinite on the feasible set, the resulting problem is convex QP and can be solved in polynomial time. On the other hand if matrix H is indefinite the resulting problem is non-convex QP, which means that the objective function may have more than one local minimizer, and the problem is NP-hard.

Solution Properties: QP can be geometrically interpreted as searching for a optimum x^* of a quadratic objective function over a given polyhedral region $\mathcal{P}$. This procedure can result in two scenarios.

1. *Feasible problem*, with the value of the objective function $-\infty < J^* < \infty$, and unique optimizer x^*.
2. *Infeasible problem*, with the value of the objective function $J^* = \infty$, caused by empty polyhedron $\mathcal{P}$.

In contrast with solutions of LP problems, the solution of QP problem if it is feasible, always results in unique optimizer x^*, due to the quadratic shape of the objective function, as demonstrated in Fig 2.6.



(a) Unique optimizer x^*, lying inside a polyhedron $\mathcal{P}$.

(b) Unique optimizer x^*, lying on the edge of polyhedron $\mathcal{P}$.

Figure 2.6: Uniqueness of feasible solutions of QP problems. Where $\mathcal{P}$ represents constraints set, objective function with its gradient is depicted by blue ellipses and optimizer x^* is given as a red dot.

There have been done in-depth research about solutions and properties of QP problems; some can be found e.g. in Abrams and Ben Israel (1969), Beale and Benveniste (1978), Best and Kale (2000), De Angelis et al. (1997).

2.2.4 Non-convex Optimization

Non-convex optimization problems are simply all problems which are not convex, i.e. either the objective function or constraints of such problems are not convex. Because there does not exist unique approach for optimization algorithm selection, the structure of general non-convex problem must be examined first. And subsequently, an appropriate method for a particular problem class can be selected. For our purposes, the important class of constrained nonlinear programming is called mixed integer programming (MIP), containing both, continuous and discrete variables. Deeper view inside a class of MIP problems can be found e.g. in Nemhauser and Wolsey. (1988), Schrijver. (1984), Wolsey (1998).

Mixed Integer Linear Programming

(MILP) is an non-convex optimization problem which has a linear objective function (2.27a) with continuous real variables x and integer variables δ subject to linear constraints (2.27b), (2.27c), and can be in general formulated as follows.

$$J^* = \min_{x,\delta} \quad c^T x + d^T \delta \tag{2.27a}$$

$$\text{s.t.} \quad Ax + E\delta \leq b \tag{2.27b}$$

$$A_{\text{eq}} x + E_{\text{eq}} \delta = b_{\text{eq}}, \tag{2.27c}$$

Where $x \in \mathbb{R}^n$, $\delta \in \mathbb{N}^q$, $c^T \in \mathbb{R}^n$, $d^T \in \mathbb{R}^q$, $A \in \mathbb{R}^{m \times n}$, $E \in \mathbb{R}^{m \times q}$ $b \in \mathbb{R}^m$, $A_{\text{eq}} \in \mathbb{R}^{p \times n}$, $E_{\text{eq}} \in \mathbb{R}^{p \times q}$, and $b_{\text{eq}} \in \mathbb{R}^p$. The convexity of MILP problem is lost due to presence of integer variables δ, which is the only difference in MILP's structure comparing the problem with the classical LP problem.

Solution Properties: MILP can be geometrically interpreted as searching for an optimum x^* of a linear objective function over a given polyhedral region $\mathcal{P}$, where the optimal solution can be found only if some given variables holds integer values. This can be done by solving so-called relaxed LP problems (Agmon 1954) with fixed combination of integer variables representing classical LP problem. To enhance the efficiency of such relaxed problems several techniques are being used, one such method is called cutting plane method (Avriel 2003, Boyd and Vandenberghe 2004). This method is based on the iterative refinement of a feasible set or objective function utilizing linear inequalities, termed cuts. Cutting plane method, together with whole MILP optimization procedure is demonstrated in Fig. 2.7.

Mixed Integer Quadratic Programming

(MIQP) is an non-convex optimization problem which has a quadratic objective function (2.28a) with continuous real variables x and integer variables δ subject to linear constraints (2.28b), (2.28c), and can be in general formulated as follows.

$$J^* = \min_{x,\delta} \quad x^T H_1 x + x^T H_2 \delta + \delta^T H_3 \delta + c^T x + d^T \delta \tag{2.28a}$$

$$\text{s.t.} \quad Ax + E\delta \leq b \tag{2.28b}$$

$$A_{\text{eq}} x + E_{\text{eq}} \delta = b_{\text{eq}}, \tag{2.28c}$$

Where $x \in \mathbb{R}^n$, $\delta \in \mathbb{N}^q$, $H_1 \in \mathbb{R}^{n \times n}$, $H_2 \in \mathbb{R}^{n \times q}$, $H_3 \in \mathbb{R}^{q \times q}$, $c^T \in \mathbb{R}^n$, $d^T \in \mathbb{R}^q$, $A \in \mathbb{R}^{m \times n}$, $E \in \mathbb{R}^{m \times q}$, $b \in \mathbb{R}^m$, $A_{\text{eq}} \in \mathbb{R}^{p \times n}$, $E_{\text{eq}} \in \mathbb{R}^{p \times q}$, and $b_{\text{eq}} \in \mathbb{R}^p$. Similarly as with MILP and LP problems relation it is also with the MIQP and QP problems, where only difference in problems structure lies in presence of integer-valued variables δ.

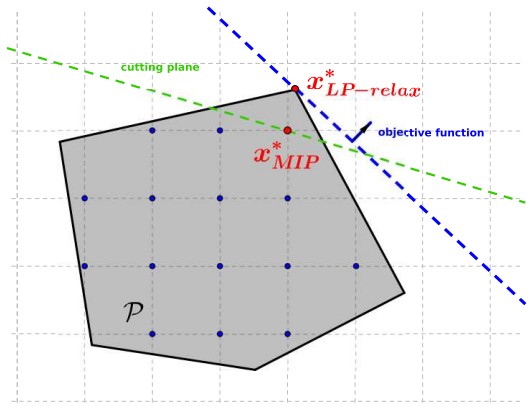


Figure 2.7: LP relaxation and cutting plane method for the solution of MILP problem. Where $\mathcal{P}$ represents constraints set, the objective function is represented by dashed blue line with its direction vector, additional cutting plane reshaping constraints set is represented by dashed green line. The only possible integer-valued solutions are depicted as a blue dots, and finally, optimizers are depicted as red dots, where $x^*_{LP-relax}$ stands for the solution of relaxed LP problem and x^*_{MIP} stands for the actual optimal solution of MILP problem.

Solution and computational aspects of MIP problems

From a structural point of view, the differences of MILP and MIQP problems against their convex counterparts (LP, QP) are often minuscule and sometimes hidden in restriction of some variables to be integer-valued. However, their actual difference manifests in the solution of such problems and comparison of their computational requirements. In straightforward fashion, to obtain a solution of MIP problems is to enumerate all possible combinations of the binary variables δ, and for each combination of the fixed binaries as static parameters, compute the optimal solution for real variables x contained in the problem as standard LP or QP problem respectively. The main drawback of MIP problems lies in their exponentially growing complexity depending on the number of included binary variables δ. Therefore several techniques have been developed for the reduction of necessary enumerated combinations of binary variables. Namely widely used *Branch and Boud* and *Branch and Cut* methods (Adjiman et al. 1996, Belotti et al. 2013, Linderoth and Ralphs 2005, Richards and How 2005). Moreover some tricks and hacks can be used, e.g. for the reduction of the number of binary variables, decreasing the computational burden and improving the performance of MIP problems in general. Commonly used state of the art solvers such as CPLEX (ILOG, Inc. 2003) or Gurobi (Gurobi Optimization 2012), have become extremely efficient in solving MIP problems. For more about available MIP solvers visit NEOS-software (2014), or NEOS-solvers (2014).

2.2.5 Multi Parametric Programming

A *parametric programming* can be classified as a subfield of *operations research*, which is a discipline that deals with the application of advanced analytical methods to obtain optimal or near-optimal solutions to complex decision-making problems (INFORMS.org 2014). In operations research, there exist several approaches to parameter variations and dealing with the uncertainties in optimization problems, for all we will name three of them.
First called *sensitivity analysis*, which studies the change of the solution as the response of the model to small perturbations of its parameters (Saltelli et al. 2008). Second is called *interval analysis* where interval-ranged input data model the uncertainties in the problem. Finally, a *parametric programming* is a method for obtaining and analysis of the optimal solution of an optimization problem with giving a full range of parameter values, representing feasible initial conditions of the problem. Parametric programming systematically subdivides the space of parameters into individual regions, which depict the feasibility and corresponding performance as a function of uncertain parameters, and subsequently provide the decision maker with a complete map of various outcomes (Borrelli et al. 2014).
Parametric problems are usually being divided into subcategories, based on number of varied parameters in the problem:

- *Parametric programming* with single parameter.
- *Multi parametric programming* with multiple parameters.

Or based on a type of the optimization problem:

- *Multiparametric convex programming*
  - *Multiparametric linear programming* (mpLP)
  - *Multiparametric quadratic programming* (mpQP)
- *Multiparametric non-convex programming*
  - *Multiparametric mixed integer linear programming* (mpMILP)

The main reason why we are dealing with multiparametric programming is to characterize and compute the state feedback solution of optimal control problems, as will be shown later in Section 3.2.3. Further, in this section, we will define multiparametric versions of standard, LP, QP, MILP and MIQP problems.

Standard Multiparametric Program

A standard multiparametric program (mpP) can be defined in general as

$$J^*(\xi) = \min_U J(U, \xi) \tag{2.29a}$$

$$\text{s.t. } GU \leq w + E\xi, \tag{2.29b}$$

where $U \in \mathbb{R}^s$ represents vector of optimization variables, $\xi \in \mathbb{R}^n$ stands for vector of parameters, $J^*(\xi)$ represents optimal value of the objective function $J(U, \xi)$, and $U^*(\xi)$ is an optimizer. With $G \in \mathbb{R}^{r \times s}$, $w \in \mathbb{R}^r$ and $E \in \mathbb{R}^{r \times n}$, where r represents number of inequalities.

Multiparametric Linear Programming

A multiparametric linear programming (mpLP) problem is defined as

$$J^*(\xi) = \min_U \; c^T U + d^T \xi \tag{2.30a}$$

$$\text{s.t. } GU \leq w + E\xi, \tag{2.30b}$$

where $U \in \mathbb{R}^s$, $\xi \in \mathbb{R}^n$, $c \in \mathbb{R}^s$, $d \in \mathbb{R}^n$, $G \in \mathbb{R}^{r \times s}$, $w \in \mathbb{R}^r$ and $E \in \mathbb{R}^{r \times n}$.

Multiparametric Quadratic Programming

A multiparametic quadratic programming (mpQP) problem is defined as

$$\min_U \; \tfrac{1}{2} U^T H U + \xi^T Q U + \xi^T R \xi + d^T \xi \tag{2.31a}$$

$$\text{s.t. } GU \leq w + E\xi, , \tag{2.31b}$$

where $U \in \mathbb{R}^s$, $\xi \in \mathbb{R}^n$, $G \in \mathbb{R}^{r \times s}$, $w \in \mathbb{R}^r$, $E \in \mathbb{R}^{r \times n}$, $Q \in \mathbb{R}^{n \times s}$, $R \in \mathbb{R}^{n \times n}$, $d \in \mathbb{R}^n$, and $H \in \mathbb{R}^{s \times s}$, where matrix $H = H^T \succ 0$ is positive semidefinite.

Multiparametric Mixed Integer Linear Programming

A multiparametric mixed-integer linear programming (mpMILP) problem is defined as

$$J^*(\xi) = \min_{U,\delta} \; b^T U + c^T \delta + d^T \xi \tag{2.32a}$$

$$\text{s.t. } GU + S\delta \leq w + E\xi, \tag{2.32b}$$

where $U \in \mathbb{R}^s$, $\delta \in \mathbb{N}^q$, $\xi \in \mathbb{R}^n$, $b \in \mathbb{R}^s$, $c \in \mathbb{R}^q$, $d \in \mathbb{R}^n$, $G \in \mathbb{R}^{r \times s}$, $S \in \mathbb{R}^{r \times q}$, $w \in \mathbb{R}^r$ and $E \in \mathbb{R}^{r \times n}$.

Solution Properties of Multiparametric Problems

The main goals of parametric programming can be described as to find and analyze the following.

- *Feasibility set* $\mathcal{X}$, or domain of the parameters ξ, as a set of parameters for which a particular problem has an optimal solution, usually in a form of the polytopic partition as in Def. 2.1.10.

- *Optimal solution* or optimizer $U^*(\xi)$, usually in the form of PWA functions defined over polytopic partition of $\mathcal{X}$. Representing a sets of parameters ξ, for which the optimal solution remains the same, respectively retains the same characteristics.
- *Feasibility function* $J^*(\xi)$ as an optimal value of the objective function $J(U, \xi)$ for the feasibility set $\mathcal{X}$, usually in two forms: PWA functions as in Def. 2.1.12 for mpLP or PWQ functions as in Def. 2.1.13 for mpQP problems.

Theorem 2.2.4 (Properties of Multiparametric Problems)

Consider a mpLP (2.30)*, mpQP* (2.31)*, and mpMILP* (2.32) *problems then:*

- *The feasibility set* $\mathcal{X}$ *of parameters* ξ *is convex for mpLP and mpQP, or possibly non-convex for mpMILP, and partitioned int o* $R \in \mathbb{N}_+$ *polyhedral regions*

$$\mathcal{P}_r = \{\xi \in \mathbb{R}^n : \ H_r\xi \leq K_r\}, \quad r \in \mathbb{N}_1^R \tag{2.33}$$

  where $H_r \in \mathbb{R}^n$ *and* $K_r \in \mathbb{R}$

- *Optimal solution* $U^*(\xi) : \mathcal{X} \mapsto \mathbb{R}^n$ *is a continuous PWA function*

$$U^*(\xi) = F_r\xi + g_r, \quad if \quad \xi \in \mathcal{P}_r \tag{2.34}$$

  where $F_r \in \mathbb{R}^{r\times n}$*, and* $g_r \in \mathbb{R}^r$

- *Feasibility function* $J^*(\xi) : \mathcal{X} \mapsto \mathbb{R}$ *is for*
  - *mpLP: continuous, convex, and piecewise affine (PWA), in form*

$$J^*(\xi) = R_r\xi + C_r, \quad if \quad \xi \in \mathcal{P}_r \tag{2.35}$$

  - *mpQP: continuous, convex, and piecewise quadratic (PWQ), in form*

$$J^*(\xi) = \xi^T Q_r\xi + R_r\xi + C_r, \quad if \quad \xi \in \mathcal{P}_r \tag{2.36}$$

  - *mpMILP: possibly discontinuous, non-convex, and piecewise affine (PWA), in form*

$$J^*(\xi) = R_r\xi + C_r, \quad if \quad \xi \in \mathcal{P}_r \tag{2.37}$$

  where $Q_r \in \mathbb{R}^{n\times n}$*,* $R_r \in \mathbb{R}^n$*, and* $C_r \in \mathbb{R}$

■

2.3 Probability Theory and Statistics

2.3.1 Classification and Differences

Probability Theory is a branch of mathematics dealing with probability, uncertainty and analysis of random phenomena in general. The principal objects here are the random variable, stochastic process and random event. These are non-deterministic mathematical abstractions which are in contrast with standard deterministic notions of such objects. Probability theory is essential for quantitative analysis of large data sets, which can occur in many practical or theoretical fields of study. Probability also lies down the mathematical foundations for statistics, allowing modeling randomness and uncertainty of empirical data sets. Moreover, by using probability theory we are also able to estimate the stochastic behaviour of the large complex systems. This ability to comprehend and deal with such complexity goes far beyond the limit of classical deterministic approaches for the description of dynamical systems.

Mathematical Statistics is a branch of mathematics dealing with the analysis, collection, interpretation, presentation and organization of data. Practical applications of statistics count for modeling, planning and analysis of inaccurate or empirical observations.

The difference between statistics and probability theory may not seem obvious due to the tight boundary between these fields. Some fundamental differences are very briefly captured in Fig. 2.8. The probability theory is used for the description of formation, generation or evolution of stochastic data, where statistics, on the other hand, is used for analysis of these random-fashioned data and modeling of the processes behaviour by whose were these data generated.

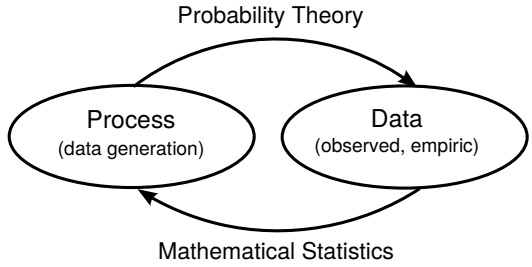


Figure 2.8: Relation between probability theory and mathematical statistics.

Statistical methods can be basicly divided into two groups, Exploratory and Confirmatory (Gelman 2004, Hoaglin et al. 1983, Tukey 1977).

- **Confirmatory data analysis** or inferential statistics, which draws conclusions from data or where the hypothesis is formulated and subsequently confirmed or disproved by confirmatory data analysis techniques (e.g. regression analysis, confidence intervals, etc.). The confirmatory analysis uses the traditional statistical tools of inference,

significance, and confidence. It is comparable to a court trial, or the process of evaluating evidence.

- **Exploratory data analysis** also called descriptive statistics, on the other hand, describes data, i.e. summarizes the data and their characteristic properties, or uses data sets to generate the hypotheses. The well known techniques are e.g. cluster analysis, factor analysis, principal component analysis, etc.. If a model fits the data, exploratory analysis finds patterns that represent deviations from the model, it isolates patterns and features of the data and reveals them forcefully for analysis. Exploratory data analysis is sometimes compared to detective work or the process of gathering evidence.

2.3.2 Terminology and Definitions

Probability is a value representing certainty of a particular event E. It is computed as the cardinality of true or occurring events $m = |E|$, divided by a number of all possible events $n = |\Omega|$, where Ω is also called a sample space. Subsets of set Ω are called random events $\mathcal{F}$ and are set of outcomes to which a probability is assigned.
This intuitive definition of probability is called classical or Laplace definition. The simplest examples of a random event are a flip of coin or dice roll. Then the likelihood of a particular result of a coin flip is 50% because flipping a coin leads to sample space composed of only two possible outcomes that are equally likely. Similarly, the probability of a dice roll result is one occurring event to six possible events of sample space.
More formalized way to define probability is by using Kolomogorov's axiomatic formulation, where sets are interpreted as events and probability itself as a measure on a class of sets.

Definition 2.3.1 (Axiomatic Probability) *Kolomogorov proposed three axioms (Kolmogorov 1933).*

1. ***Non-negativity*** *of an event probability, represented by a real number:*

$$\Pr(E) \in \mathcal{R}, \quad \Pr(E) \geq 0, \qquad \forall E \in \mathcal{F} \tag{2.38}$$

2. ***Unit Measure*** *says that probability of the certain elementary event is equal to 1, there are no elementary events outside the sample space.*

$$\Pr(\Omega) = 1. \tag{2.39}$$

3. ***σ-additivity*** *Probability of union of disjoint (mutually exclusive) events E is equal to the countable sequence of their particular probabilities.*

$$\Pr(\cup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} \Pr(E_i). \tag{2.40}$$

□

In particular, $\Pr(E)$ is always finite, in contrast with more general measure theory. Note that, if you cannot precisely define the whole sample space Ω, then the probability of any subset cannot be defined either.
Probability space $(\Omega, \mathcal{F}, \Pr)$, also called measure space, or probability triple, are notions representing summarisation of above-given Axioms 2.3.1, in the form of a structured set that models a real-world process consisting of randomly occurring states. It is constructed by three parts:

- *Sample Space* Ω, as set of all possible outcomes, with aggregated probability (2.39).
- *Set of events* or event space $\mathcal{F}$, where each event is a set containing zero or more outcomes.
- *Probability measure* Pr of event E is a real-valued function assigning probabilities to the events, defined on a set of events in probability space, that satisfies measure properties such as countable additivity (2.40), and has a form:

$$\Pr(E) = \frac{m}{n} = \frac{|E|}{|\Omega|} \tag{2.41}$$

  A measure on a set in mathematical analysis is a systematic way to assign a number to each suitable subset of that set, intuitively interpreted as its size. The difference between a probability measure and general notion of measure is that a probability measure must assign value 1 to the entire probability space.

Randomness is a broad concept in common language, philosophy and science, usually understood as a lack of pattern or predictability in events. In a sequence of some particular data types, it suggests a non-order or non-coherence, such that there is no intelligible pattern or combination. Even though a *random events* are unpredictable as individualities, the cardinalities of different outcomes over a large number of events are usually predictable. Therefore *randomness* here implies a measure of uncertainty of events and refers to situations where the certainty of the outcome is at issue.
In mathematics, there are several formal definitions of randomness. In statistics, a *random variable* also called *stochastic variable* is an assignment of a numerical value to each possible outcome of an event space, used for identification and the calculation of probabilities of the events. The axiomatic measure-theoretic definition, where continuous random variables are defined in terms of sets of numbers, along with functions that map such sets to probabilities can be found in Fristedt and Gray (1996), Kallenberg (1986; 2001), Papoulis and Pillai (2001). Here comes the notion of a *random element*, what is a generalization of the concept of a *random variable* to more complex spaces than the simple real line, defined as follows.

Definition 2.3.2 (Random Element) *Let* $(\Omega, \mathcal{F}, \Pr)$ *be a probability space and* $(E, \mathcal{E})$ *a measurable space. Than* $(E, \mathcal{E})$*-valued random variable or random element* $X \colon \Omega \to E$ *is a* $(\mathcal{F}, \mathcal{E})$*-measurable function from the set of possible outcomes* Ω *to some set* E*. The latter means that, for every subset* $B \in \mathcal{E}$*, its preimage* $X^{-1}(B) \in \mathcal{F}$ *where* $X^{-1}(B) = \{\omega :$

$X(\omega) \in B\}$. *This definition enables us to measure any subset* $B \in \mathcal{E}$ *in the target space by looking at its preimage, which by assumption is measurable.* □

When E is a topological space, then the usual choice for the σ-algebra $\mathcal{E}$ is the Borel σ-algebra $\mathcal{B}(E)$, which is the σ-algebra generated by the collection of all open sets in E. In that casem the $(E, \mathcal{E})$-valued random variable is called the E-valued random variable. Further, when space E is the real line $\mathbb{R}$, then such real-valued random variable is called just the random variable.

Definition 2.3.3 (Random Variable) *For real observation space, real-valued random variable is the function* $X \colon \Omega \to \mathbb{R}$ *if it is measurable, what means that for each set* $B \in \mathbb{R}$ *holds:*

$$\{\omega : \omega \in \Omega, X(\omega) \in B\} \in \mathcal{F} \tag{2.42}$$

Equivalently X *is a random variable if and only if for each real number* r *holds:*

$$\{\omega : \omega \in \Omega, X(\omega) \leq r\} \in \mathcal{F} \qquad \forall r \in \mathbb{R} \tag{2.43}$$

□

A *multivariate random variable* or *random vector* is a list of mathematical variables each of whose value is unknown or has random properties, either because there is inprecise knowledge of its value or because the value has not yet occurred. Normally elements of a random vector are real-valued numbers. Random vectors are often used as the underlying realizations of various types of related random variables, e.g. a random matrix, random tree, random sequence, random process, etc.

Definition 2.3.4 (Random Vector) *is a column vector* $\mathbf{X} = (\omega_1, ..., \omega_n)^T$ *with scalar-valued random variables as its components on the probability space* $(\Omega, \mathcal{F}, \Pr)$. □

A *random process* also called a *stochastic process* is a collection of random variables describing a process whose outcomes do not follow deterministic rules but representing the evolution of random values over time, described by probability distributions. The behaviour of a stochastic process is characterized by some indeterminacy: even if the initial conditions are known, there are several (often infinitely many) directions in which the process may evolve. What is in contrast with a deterministic process which can only evolve in one way. Thus the stochastic process is usually understood as the probabilistic counterpart to the deterministic process (Papoulis and Pillai 2001).

Definition 2.3.5 (Stochastic Process) *Assume to have a probability space* $(\Omega, \mathcal{F}, \Pr)$ *and a measurable space* (S, Σ) *, an* S*-valued stochastic process is a collection of* S*-valued random variables on sample space* Ω*, indexed by a totally ordered set* T *represeting time. Than a stochastic process* X *is a collection* $\{X_t : t \in T\}$ *where each* X_t *is an* S*-valued random variable from* $(\Omega, \mathcal{F}, \Pr)$*. The space* S *is called the state space of the process.* □

In case that $T = \mathbb{Z}$ or $T = \mathbb{N} + \{0\}$, we are speaking about stochastic process in discrete time. For continuous stochastic process holds that T is an interval in $\mathbb{R}$.

Typical examples of processes modeled by stochastic framework include the stock market, exchange rate fluctuations, weather phenomena evolutions, signals such as speech, audio and video, medical data such as a patient's EKG, EEG, blood pressure or temperature, and random movement such as Brownian motion.

Stochastic Simulation is a simulation that operates with random variables that can change with a certain probability. Stochastic here also means that values of particular parameters are variable or random. During stochastic simulation, a projection of stochastic model is created based on a set of random values of model's parameters. Outputs are recorded, and the process is repeated with a new set of random values, until a reasonable amount of data is gathered (w.r.t. a particular case). In the end, the distribution of the outputs shows the most probable estimates as well as boundaries of expectations (Dlouhy et al. 2005).
We can roughly classify stochastic simulation approaches into following types:

- *Discrete-event simulation.*
- *Continuous-event simulation.*
- *Hybrid simulation* representing combined simulation of discrete and continuous events.
- *Monte Carlo simulation*, which is commonly used estimation procedure, based on averaging independently taken samples from the distribution (Dlouhy et al. 2005).
- *Random number generators* are devices capable of producing a sequence of numbers which can not be "easily" identified with deterministic properties (Knuth 1998).

Frameworks for handling models of stochastic processes are *stochastic calculus of variations* allowing the computation of derivatives of random variables, and *stochastic calculus* which allows the consistent theory of integration to be defined for integrals of stochastic processes.

Probability distribution is a probability measure, which assigns a probability to each measurable subset of sample space Ω of a random experiment or some statistical data-set. There are several types of probability distribution each specific for particular data sets:

- *Categorical distribution*, when sample space is encoded by non-numerical random variables.
- *Probability mass function*, when sample space is encoded by discrete random variables.
- *Probability density function*, when sample space is encoded by continuous random variables.

It can be either univariate (probability fo single random variable) or multivariate (probability of random vector). Outcomes of more complex systems, involving stochastic processes defined in continuous time, may demand the use of more general probability measures. In

practice there are many commonly used and known distributions e.g. normal, log-normal, pareto, etc..
Normal distribution or *Gaussian distribution* is one of most important and commonly used type of univariate continuous probability distribution. It is also a subclass of elliptical distributions. The domain of the function lies between any two real limits or real numbers, as the curve approaches zero on either side. In reality, there are not many variables driven by normal distributions. However, they are still crucial in statistics due to the central limit theorem, which says that under certain conditions normal distributions well approximates a huge set of other probability distributions classes (continuous or discrete) (Casella 2001, Lyon 2014).

Definition 2.3.6 (Normal Distribution)

$$f(\omega, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\omega-\mu)^2}{2\sigma^2}} \tag{2.44}$$

The parameter μ represents the mean and also its median and mode. The parameter σ stands for standard deviation, its variance is therefore σ^2. Thus when a random variable ω is distributed normally with mean μ and variance σ^2, we write $\omega \sim \mathcal{N}(\mu, \sigma^2)$. □

Definition 2.3.7 (Standard Normal Distribution)

$$f(\omega) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{\omega^2}{2}} \tag{2.45}$$

Also called the unit normal distribution is usually denoted by $\mathcal{N}(0, 1)$, if $\mu = 0$ and $\sigma = 1$, and a random variable ω with that distribution is a standard normal deviate. □

2.4 Summary

The aim of this chapter was to present the necessary mathematical background for understanding the following content of this thesis. First, the basic notions on sets and functions are defined, second, the fundamentals of mathematical optimization are introduced, together with the basics of probability and statistics. The described optimization problem classes, particularly linear, quadratic or mixed-integer programming are directly linked to the optimal control problems such as model predictive control, which is a subject of the study in Chapter 3. The probability theory is used for handling the uncertainties in the weather predictions used in the stochastic formulations of the MPC problem, as shown in Section 5.3.

Chapter 3

Model Predictive Control

Predictive control is a discovery, not an invention, ...

IFAC Congress Munich, 1987

Model predictive control (MPC) belongs to a class of computer control algorithms, more specifically to the optimal control methods which are using mathematical model of the process to predict the future response of process on a sequence of control variable manipulations. Once the predictions are made, the control algorithm with usage of the optimization techniques computes appropriate control actions to provide desired output behavior of the process in optimal fashion.
Colloquially we can describe this method as a "look ahead" strategy, when the controller is able to foresee a future behavior of the process with usage of given knowledge about that particular process and consequently evaluate the optimal control strategy to achieve the best possible outcome, which are satysfying long term goals and criteria. This strategy stands in contrast with classical control theory techniques e.g. PID controllers, which are able to achieve only short term goals set in actual time, resulting in more costly and ofthen unsatisfactory long term performance. This phenomenon can be described as "winning the battle but losing the war" (Anderson et al. 2014).

3.1 Classification of MPC in Control Theory

Control theory can be in general described as a study of systems behavior and control, with practical emphasis on principles, design and construction of control systems. As the main objective of control theory is to affect the behavior of controlled system called plant, to achieve desired outputs properties, called reference while meeting given restrictions and real world limitations, in control theory terminology called constraints. To achieve this goal a *controller* must be designed with following capabilities executed in subsequent steps:

1. monitoring of the plants output
2. output-reference comparisson, or control error evaluation
3. evaluation of appropriate conrol actions

The above mentioned steps with evaluation of conrol actions based on control error are describing general notion of closed-loop, also called feedback controller. By measuring the difference between a actual and desired output values, feedback controller can provide a corrective action by applying this difference also called control error as feedback to the input of the system. The second paradigm in control theory is called open-loop controller, or a non-feedback controller, which computes its conrol actions as inputs into a system by using only the current state measurements and model of the system. More general definitions of feedback and conrol can be found in Astrom and Murray (2012) and are stated as follows.
Feedback is defined as the interaction of two (or more) dynamical systems that are connected together such that each system influences the other and their dynamics are thus strongly coupled. We say that a system is closed loop if the systems are interconnected in a cycle and open loop when there is no interconnection.
Control is defined as the use of algorithms and feedback in engineered systems. The basic feedback loop of measurement, computation and actuation is the central concept in control. The key issues in designing control logic are ensuring that the dynamics of the closed loop system are stable (bounded disturbances give bounded errors) and that they have the desired behavior (good disturbance rejection, fast responsiveness to changes in operating point, etc).

3.1.1 Classical vs Modern Control Theory

From historical point of view a control theory can be divided into two subfields, older methods are called classical, and younger are called modern control theory methods. The principal differences of these subfields lies in approach to dynamical systems representation and maniplation. Before going deeper let's shortly recall and sumarize basic characteristics and properties of dynamical systems captured in Fig. 3.1, which are necessary for understanting the differences between classical and modern control theory methods.

Classical Control Methods

General characteristic of classical control methods, is usage of techniques for changing the domains of dynamical systems described by ordinary differential equations (ODE) to avoid the complexities of time-domain ODE solutions. The mentioned techniques are integral transforms, changing time-domain ODE's into a regular algebraic polynomial in the transform domain, allowing easy manipulation. Namely the most used transforms here are the *Fourier transform* with frequency domain representation, more general *Laplace transform* with complex frequency domain representation also called s-domain, and its discrete-time equivalent called *Z transform*. The transformed polynomials are further formed into so

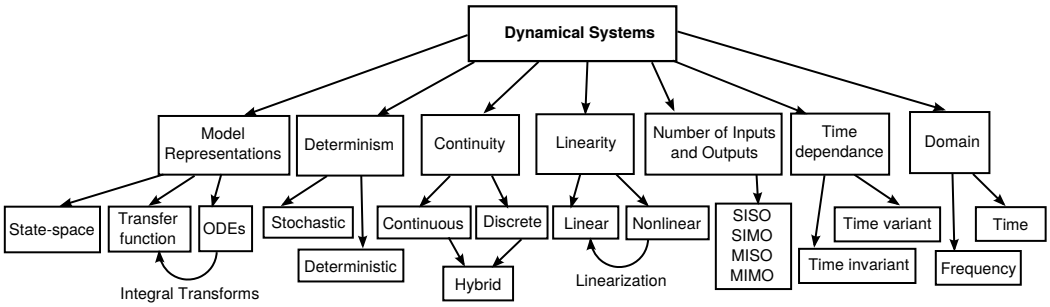


Figure 3.1: Classification of basic characteristics and properties of dynamical systems.

called *transfer function*, which is nothing less than matematical representation of input-output (I/O) system model, representing relation between an input signal and the output signal of the system. Main drawback of classical methods are, that they can be used only for control of single-input single-output (SISO) sysyems, with requirement on model of the sysyem to be linear time-invariant (LTI). Classical control methods are not able to incorporate constraints naturraly arising from industrial control problems, and has optimization lacking overall performance. Most common example of classical control methods is *proporcional integral derivative* (PID) controler, which accounts for more than 90% of the control and automation applications today, mainly thanks to its simple implementation with relative efficiency. Even though, that classical control methods are widely used in practice, and are still popular among old-fashioned control engineers, they are providing satisfactory results only in control of simple processes, but unsatisfactory results in control of more complex systems, which are forming majority of todays industrial control problems.

Modern Control Methods

Instead of changing domains to avoid the complexities of ODE solutions, modern control is using methods for conversion of high-order differential equations into a system of first-order time domain equations called state equations, which are easy to handle using well known linear algebra techniques. This model representation od dynamial systems is being called state-space repsesentation, where the inputs, outputs, and internal states of the system are described by vectors called input u, output y and state x variables respectively. Main advantage of state-space representation is perservation of the time domain character, where the response of a dynamical system is a function of various inputs, previous system states, and time, shortly $y = f(x, u, t)$. Moreover a straightforward representation and handling of multiple-input multiple-output MIMO systems is allowed using state-space model representations.

Structure and Comparisson of Control Theory Methods

The overall comparisson of basic characteristics and properties of above metioned methods can be sumarised in compact table form Tab. 3.1, with highlighted differences. Moreover

Control Theory Methods	Classical	Modern
Domain	Frequency, S-domain	Time
Model repsesentation	Transfer function	State-space
Continuity	Continuous	Continuous, Discrete, Hybrid
Linearity	Linear	Linear, Non-Linear
Time variance	Time-invariant (TI)	Time-variant (TV)
Dimensions	SISO	MIMO
Determinism	Deterministic	Deterministic, Stochastic
Optimization	NO	YES
Constraints	NO	YES
Implementation	Cheap, Easy	Expensive, Complex

Table 3.1: Basic characteristics and properties comparation of classical and modern control theory methods. Where, the red color indicates the drawback, while green color stands for advantage of the methods.

the structured classification of control theory methods is captured in Fig. 3.2. Note that structure presented here is not rigid, but contains a rich overlaps between particular control methods forming a dense network, where each node represents a method with specific properties and characteristic approaches to control problems. Further we will not investigate the comprehensive structure and describe all methods mentioned in Fig. 3.2 as it is not covered by topic of this thesis. In the following sections we will rather focus on a group of particular control methods called optimal control and more specificly on Model Predictive Control, which undergo rapid development in last few decades mainly due to rise of modern computer technology capacities.

3.1.2 Optimal Control

Optimal control is solving a problem of finding such control law for given system, that certain optimality criteria are being fulfilled. Optimal control problem can be formulated as general optimization problem defined in Section 2.2.2, consist of cost function mapping system states and control actions, states and inputs constraints, and system dynamics usually represented as a collection of differential equations with initial condition. Solution to optimal control problem can be than perceived as evaluation of such control actions paths, which are minimizing given objective function. More about optimal control theory methods can be foung e.g. in Skogestad and Postlethwaite (2007), Tsai and Gu (2014), Zhou and Doyle (1997), Zhou et al. (1995).
Based on different formulations of objective function, constraints or systmes model type, the optimal control theory medhods are branching into the following most significant representatives:

- **Linear quadratic regulator - LQR**

  This method asuumes the controlled system to be in linear time-invariant form with quadratic objective function and missig constraints. Solution is being obtained by

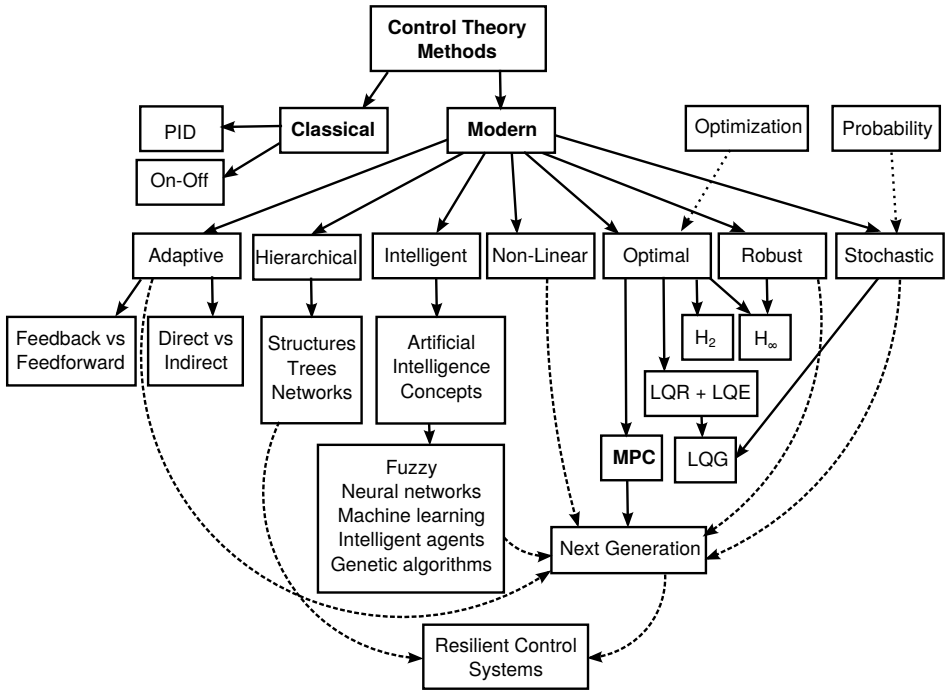


Figure 3.2: Classical vs modern control theory methods taxonomy. Where the full lines represents direct structural relations, dotted lines are depicting supporting mathematical theories, and dashed lines are outlining the evolution of separate control theory methods merging together creating a new control theory paradigms.

two Riccati equations, in form of an optimal linear state feedback controller in form $u = -Kx$.

- **Linear quadratic estimator - LQE**

  In control theory literature also refered as a Kalman optimal state estimator, or shortly a Kalman filter to honor the main contributing author of the concept. Kalman filter is processing measurements from the system, affected by disturbances during given time period and produces estimations of unmeasured and unknown system variables. The estimation of parameters is based on optimal statistical evaluation of number of measurements, which is more preciese than parameters estimation methods based on single measurement.

- **Linear quadratic gaussian regulator - LQG**

  Is an extension of a traditional LQR controller on linear systems with uncertainties in form of white gaussian noise. Structure of LQG controller is simply combination of LQR with LQE, design of both components can be done separately thanks to separation principle. The solution is an again linear state feedback controller similarly as for LQR. Main disadvantage of both methods, LQR and LQG are poor robust

properties of the resulting controllers. These drawbacks were acting as motivation, for combination of optimal and robust control theory methods, leading to development of $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control theory methods.

- **$\mathcal{H}_2$ a $\mathcal{H}_\infty$ control**

  These control methods can be equivalently formulated as an optimization problem, with only difference in usage of mathematical norms defining objective funtion. For $\mathcal{H}_2$ controller desing purposed a euclidean 2-Norm as in Def. 2.1.17 is being used, in contrast with ∞-Norm as in Def. 2.1.18 used in $\mathcal{H}_\infty$ controller design. Finding a solution for $\mathcal{H}_2$ controller isan easy problem in principle, due to uniqueness of solution given by two Riccati equations. Where in contrast finding a solution for $\mathcal{H}_\infty$ controller is very difficult problem to solve theoretically and also numerically, with usual usage of suboptimal solution with given sufficient tolerance.

- **Advanced control theory methods**

  In industrial applications under this label most commonly are mentioned model predictive control (MPC) strategies, as nowdays very popular control theory methods. Thanks to their applicability on broad range of systems, natural constraints consideration, together with their predictive capabilities, resulting in very efficient performances in most of the applications compared with concurent control strategies.

  The following sections are mentioned to provide the reader a deeper intoduction into the topic of MPC, followed by chapter with application of MPC strategies on building control problems as main interest of this thesis.

3.1.3 History and Evolution of MPC

This section will be devoted to brief history and evolution of Model Predictive Control, from early academia based concepts of optimal control theory, giving the birth to very first industrial based control applications using MPC technology. More comprehensive historical survey of industrial MPC can be found in Qin and Badgwell (2003), from where the inspiration for this whole section was taken. Moreover the simplified evolution of industrial MPC algorithms is captured in Fig. 3.3, forming a structural backbone for this section.

Early Optimal Control Theory

Development of the modern control theory concepts using optimization techniques can be traced in the early 1960's beginning with the work of Kalman (1960a;b). With first attempts for an optimal control of linear systems, resulting in development of a linear quadratic regulator (LQR), which was designed to minimize an unconstrained quadratic objective function over system states and inputs. This concept was further extended to a linear quadratic gaussian regulator (LQG), simply by adding state estimation with linear quadratic estimator (LQE), commonly called Kalman filter to honor the author. Main asset of LQR and LQG controllers are powerful stabilizing properties thanks to the infinite

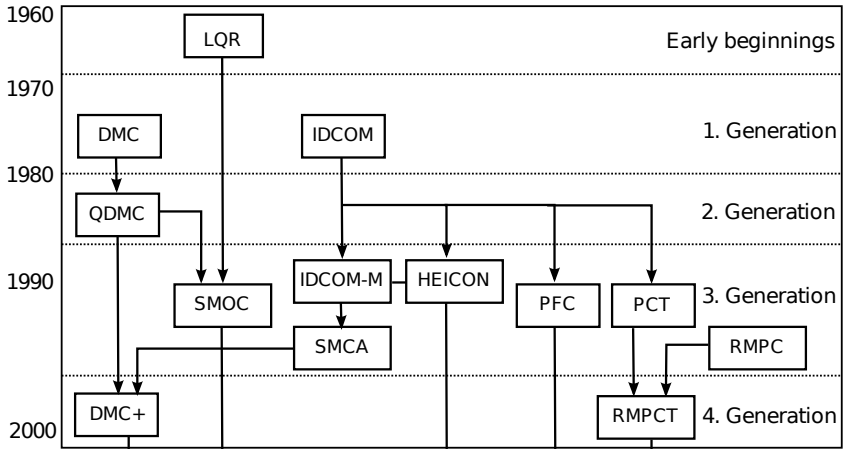


Figure 3.3: Simplified evolutionary tree of the most significant industrial MPC algorithms.

horizon. However the early practical issues handling applications were huge in quantity, the quality and impact on the industrial process control technology was strongly limited because of missing incorporation of the following listed properties in its formulation, as well as from cultural and educational reasons (García et al. 1989, Richalet et al. 1976).

- constraints
- real systems nonlinearities
- model uncertainty (robustness)
- unique performance criteria

Even though it is conceived as a firt and necessary step for developlent of following revolutionary concepts in advanced process control applications. The imediate impact of LQG control was in fields with accurate fundamental models, e.g. on aerospace industry. In Goodwin et al. (2001) it was estimated there may be thousands of real-world LQG applications with roughly 400 patents per year based on the Kalman filter.

First Generation MPC

To handle a drawbacks of a LQG approach to process control issues, a new methodology was developed in industrial enviroment with more general model based control with solution of the dynamic optimization problem on-line at each control execution over a time interval called prediction horizon. The main contribution of this approach is incorporation of the process input and output constraints directly in the problem formulation so that future constraint violations are anticipated and prevented. Moreover allowing usage of explicit multivariable mathematical models of processes. In addition an increasing flexibility was acquired by new process identification technology developed to allow quick estimation of

empirical dynamic models, significantly reducing the cost of model development. This new control paradigm for industrial process modeling and control is what we now refer to as MPC technology (Qin and Badgwell 2003).
In the beginning there was, however a wide gap between MPC theory and practice, with essential contributions from industrial engineers with their applications in process industry. First of them was developed in the late 1970s by Richalet et al. (1976; 1978) refered as Model Predictive Heuristic Control (MPHC), later called Model Algorithmic Control (MAC), with software solution referred to as IDCOM, an acronym for Identification and Command. In todays context the MPHC control algorithm would be refered as linear MPC controller. Main features of IDCOM control algorithm are:

- impulse response model
- imput and output constraints
- quadratic objective function
- finite prediction horizon
- reference trajectory
- optimal imputs computed by heuristic iterative algorithm, interpreted as the dual of identification

Another independent MPC technology was developed by engineers at Shell Oil in the early 1970s, with an initial industrial application in 1973. Subsequently Cluter and Ramaker presented unconstrained multivariable control algorithm named Dynamic Matrix Control in the 1979 (Cutler and Ramaker 1979; 1980). And Prett and Gillette, algorithm was modified to handle nonlinearities and time variant constraints (Prett and Gillette 1980). Predicted future output changes are represented as a linear combination of future input moves in compact matrix form called Dynamic Matrix. Main features of the DMC control algorithm are:

- linear step response model
- quadratic objective function
- finite prediction horizon
- output behavior specified by trying to follow the setpoint as closely as possible
- optimal imputs computed as the solution to a least squares problem

The initial IDCOM and DMC algorithms were algorithmic as well as heuristic, taking advatage of rapid development of digital computers technology. However the first MPC were not automatically stabilizing, stability was estabilished by good heuristics and well performed tuning by experienced control engineer. Moreover they were able to provide a small degree of robustness to model error. The IDCOM and DMC are classified as first generation MPC, and in contrast with LQR they had an enormous impact on industrial process control and layed the foundation the industrial MPC paradigm.

Second Generation MPC

Even though the first generation MPC algorithms provided excellent control performance of unconstrained multivariable processes, handling the process constraints was still problematic task with unsatisfactory results. The solution to this porblem came again from Shell Oil engineers at early 1980s, proposing the original DMC algorithm as a quadratic program (QP) in which input and output constraints appear explicitly. Namely Cutler et. al., came with first description of the QDMC (Cutler et al. 1983), and Garcia and Morshedi with more comprehensive description few years later (Garcia and A.M. 1986). Main features of the QDMC control algorithm are:

- linear step response model
- input and output constraints collected in a matrix of linear inequalities
- quadratic objective function
- finite prediction horizon
- output behavior specified by trying to follow the setpoint as closely as possible
- optimal imputs computed as the solution to a quadratic program

Strenght of this approach was also the fact that the resulting QP optimization problem was convex and hence easily solved by standard commercial optimization algorithms. Thanks to this qualities the QDMC algorithms refered as second generation of MPC proved to be profitable in an on-line optimization environment. As a main drawback of the QDMC approach was lack of clear way approach to handle an infeasible solution and missing recovery mode.

Third Generation MPC

From this point a popularity and usage of the MPC technology rise strongly in numbers, creating new complex problems and revealing application challenges, pointing out most important as follows.

- solving infeasibility issues
- fault tolerance control
- control requirements formulation and scaling problems

To solve infeasibility issues a new approach to constraints handling was proposed, by incorporating soft constraints which violations were penalized in objective function, and by distinguishing between high and low priority constraints. Main objective of fault tolerance as a important practical issue, was making best from control even during failure, with relaxation control specifications during this kind of situations. Third problem was difficult translation of control specifications into a consistent set of relative weights in a single

objective function for larger problems. Where these scaling problems that lead to an bad-conditioned solution, comented in Prett and Garcia (1988) as follows. The combination of multiple objectives into one objective (function) does not allow the designer to reflect the true performance requirements.
These issues were motivations for engineers of industrial comapies as Adersa, Setpoint, Inc., and Shell which were among first implementing MPC algorithms. The IDCOM-M controller was a commercial trademark of Setpoint, Inc. (where M stands for multiple input/output), and was first described in Grosdidier et al. (1988), and few years later by Froisy and Matsko (1990) implemented to a Shell fundamental control problem. Main features of the IDCOM-M control algorithm are:

- linear impulse response model
- controllability supervisor to screen out bad-conditioned plant subsets
- multi-objective quadratic function formulation, one for inputs and one for outputs
- control of coincidence points chosen from reference trajectory, as a subset of future outputs trajectories
- single move for each input
- hard or soft constraints with priority ranking

Adresa company owned nearly identical version to the IDCOM-M called hierarchical constraint control (HIECON). The IDCOM-M product was combined with setpoints identification, simulation, configuration, and control products into a single integrated system called SMCA, for Setpoint Multivariable Control Architecture.
The Shell research engineers was not far behind and in the late 1980s developed the SMOC, or Shell Multivariable Optimizing Controller, refered as a bridge between state-space and MPC algorithms (Marquis and Broustail 1998). Their approach was to combine constraint handling features of the MPC, with the richer framework for feedback by state-space methods, so that full range of linear dynamics can be represented. Main features of the SMOC control algorithm, which are now considered essential to a modern MPC formulation are listed as follows:

- state-space model
- explicit disturbance model describing the effect of unmeasured disturbances
- Kalman filter for estimation of plant states and disturbances from output measurements
- distinction between controlled variables in objective and feedback variables for estimation
- QP formulation of control problem with constraints incorporation

The SMOC algorithm can be than perceived as solving the LQR problem with input and output constraints, but lacking the strong stabilizing properties due to the finite horizon. Not long after in the 1990s a stabilizing, infinite-horizon formulation of the constrained LQR algorithm came to embrace the MPC theoretical background (Rawlings and Muske 1993, Scokaert and Rawlings 1998). Other algorithms not described but yet belonging in this section of third MPC generation was a PCT algorithm sold by Profimatics, and Honeywells RMPC algorithm.

Fourth Generation MPC

The mid and late 1990s bring significant changes in the industrial MPC landscape, mainly due to increased competition driven companies acquisitions and technologies merges. In 1995 a robust model predictive control technology RMPCT was created by merging Honeywells RMPC algorithm with Profimatics PCT controller under the label of Honeywell Hi-Spec Solutions. Second big acquisition become reality in 1996, when Aspen Technology Inc. purchased both Setpoint, Inc. and DMC Corporation, followed by by acquisition of Treiber Controls in 1998. What was resulting in subsequent merging of SMCA and DMC technologies to current Aspen Technologys DMC-plus. A simplified overwiew of the MPC technology evolution is sumarised structurally in Fig. 3.3 as refered in the beginning of this section.
The RMPCT and DMC-plus as a flagships of fourth generation of MPC technology, are being sold today with integrated high standards features of all above mentioned technologies, enhanced with following improvements.

- windows based graphical user interface
- multiple optimization levels for control objectives with different priorities
- improved identification technology based on prediction error method
- additional flexibility in the steady-state target optimization, including QP and economic objectives.
- robustness properties with direct consideration of model uncertainty

All this has been a cause to a large increase in the number and variety of practical application areas including chemicals, food processing, automotive, or aerospace applications. Mainly thanks to the MPC significant performance improvements, increasing safety, decreasing energy consumption or enviromental burden of plants production.

3.2 MPC Overwiev and Features

MPC is a control strategy that uses an optimization to calculate the optimal control inputs, with usage of mathematical model of the system and current state measurements for predicting a evolution of the system behavior, and keeping these future predictions in account during optimization. The optimization problem as proposed in Chapter 2.2.2 is composed

of two parts, objective function and constraints. In the MPC framework the cost or also called objective function evaluates fitness of a particular predicted profile of state, output and inputs with respect to qualitative criteria. Task of the optimization is then to compute the optimal profile of predicted control actions for which the cost function is minimized. The set of admissible decisions to choose from is then represented by the constraints of the optimization problem.
The MPC is based on iterative character of an optimization process executed over finite time interval also called a prediction horizon, which can be simplistically percieved as measure of how far into the future the MPC algorithm can see. At current time the plant states are being measured and a cost minimizing control strategy is computed, via a numerical algorithms over given prediction horizon.
Basic building elements forming characteristic structure of standard MPC are sumarised and listed as follows.

- Model of the system
- State measurements
- Constraints
- Objective
- Prediction horizon
- Sampling time

Note here, that multiple possibilities for each building element of MPC exist, each with specific properites which are suitable or necessary for particular control application problem.

3.2.1 Standard MPC Formulation

Standard MPC optimization problem can be formulated in a general way as follows:

$$\min_{u_0,\dots,u_{N-1}} \ell_N(x_N) + \sum_{k=0}^{N-1} \ell(x_k, u_k) \tag{3.1a}$$

$$\text{s.t. } x_{k+1} = f(x_k, u_k, d_k), \qquad k \in \mathbb{N}_0^{N-1} \tag{3.1b}$$

$$x_k \in \mathcal{X}, \qquad k \in \mathbb{N}_0^{N-1} \tag{3.1c}$$

$$u_k \in \mathcal{U}, \qquad k \in \mathbb{N}_0^{N-1} \tag{3.1d}$$

$$x_0 = x(t), \tag{3.1e}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$ and $d_k \in \mathbb{R}^q$ denote, respectively, values of states, inputs and disturbances predicted at the k-th step of the prediction horizon N. The predictions are obtained from the prediction model $f(x, u, d)$, that can be arbitrary (e.g. linear or nonlinear). Predicted states and inputs are subject to constraints sets in (3.1d) and (3.1c).

The term $\ell_N(x_N)$ in (3.1a) is called terminal penalty, while $\ell(x_k, u_k)$ is called a stage cost and its purpose is to assign a cost to a particular choice of x_k and u_k.
For a particular initial condition $x(t)$ in (3.1e), the optimization (3.1) yields the sequence $u_0^*, \ldots, u_{N-1}^*$ of control inputs that are optimal with respect to the cost (3.1a). Computational complexity of obtaining such a sequence depends on the type of the prediction model employed in (3.1b) and on the choice of the cost function (3.1).
More specifically a general MPC problem (3.1), can be given in a form

$$\min_{u_0,\ldots,u_{N-1}} \|Q_N x_N\|_p + \sum_{k=0}^{N-1} (\|Q_\mathrm{x} x_{k+1}\|_p + \|Q_\mathrm{u} u_k\|_p) \tag{3.2a}$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k + Ed_k, \qquad k \in \mathbb{N}_0^{N-1} \tag{3.2b}$$

$$H_\mathrm{x} x_k \leq K_\mathrm{x}, \qquad k \in \mathbb{N}_0^{N-1} \tag{3.2c}$$

$$H_\mathrm{u} u_k \leq K_\mathrm{u}, \qquad k \in \mathbb{N}_0^{N-1} \tag{3.2d}$$

$$x_0 = x(t). \tag{3.2e}$$

Where cost function (3.2a) is represented by an arbitrary p-Norm as in Def. 2.1.15, defined over prediction horizon $N \in \mathbb{N}$ with weight matrices $Q_\mathrm{x} \in \mathbb{R}^{n \times n}$, $Q_\mathrm{u} \in \mathbb{R}^{m \times m}$ and terminal penalty weight $Q_N \in \mathbb{R}^{n \times n}$, with conditions $Q_\mathrm{x} \succeq 0$ and $Q_N \succeq 0$ to be positive semidefinite and $Q_\mathrm{u} \succ 0$ to be positive definite. Moreover the prediction model holds the form of discrete-time linear time-invariant system in a state-space representation (3.2b) with incorporated disturbances d_k. With the model matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and $E \in \mathbb{R}^{n \times q}$, linear constraints matrices $H_\mathrm{x} \in \mathbb{R}^{n_\mathrm{x} \times n}$, $K_\mathrm{x} \in \mathbb{R}^{n_\mathrm{x}}$, $H_\mathrm{u} \in \mathbb{R}^{n_\mathrm{u} \times m}$ and $K_\mathrm{u} \in \mathbb{R}^{n_\mathrm{u}}$, where n_x, n_u stands for number of state and input inequalities. Recalling that n, m and q denotes the dimension of state, input and disturbances, respectively.

Remark 3.2.1 *Notice that, MPC optimization problem* (3.2) *with cost function* (3.2a) *in form of 2-Norm as in Def. 2.1.17 is resulting in convex QP problem* (2.26). *Additionally if the model of the system contains state variables, which can only acquire integer or binary values, than the system exhibits hybrid dynamics behavior and the resulting optimization problem becomes non-convex MIQP problem* (2.28). □

3.2.2 Receding Horizon Control

Standardly the MPC algorithms are being implemented in the closed-loop fashion using the principle of the *receding horizon control* (RHC), where the prediction horizon keeps being shifted forward, implementing only the first step of the computed control strategy and discarding the rest. The closed-loop MPC procedure can be sumarised in the following general RHC policy Algorithm 1. Moreover, a characteristic behavior of a discrete closed-loop MPC strategy is captured in Fig 5.1. Alternatively, an open-loop MPC can be designed by ignoring a RHC control policy and simply implementing, not only the first control input, but the whole control strategy computed over the given prediction horizon, paying the cost of loosing the feedback from the controlled system.

Algorithm 1 Receding horizon control.

1. At time t, measure (or estimate) the plant's state $x(t)$
2. Compute the optimal sequence of control inputs $\{u^*(t), \ldots, u^*(t + NT_s)\}$ by solving the optimization problem (5.8).
3. Select and apply only the first element of the control signals sequence, i.e., $u^*(t)$, to achieve the feedback behaviour of the MPC controller.
4. Implement the selected control signal over a pre-defined time interval, called sampling time T_s.
5. Time advances to the next interval, and the procedure is repeated from step 1, with new measurements at time $t + T_s$, using values of $x(t + T_s)$.

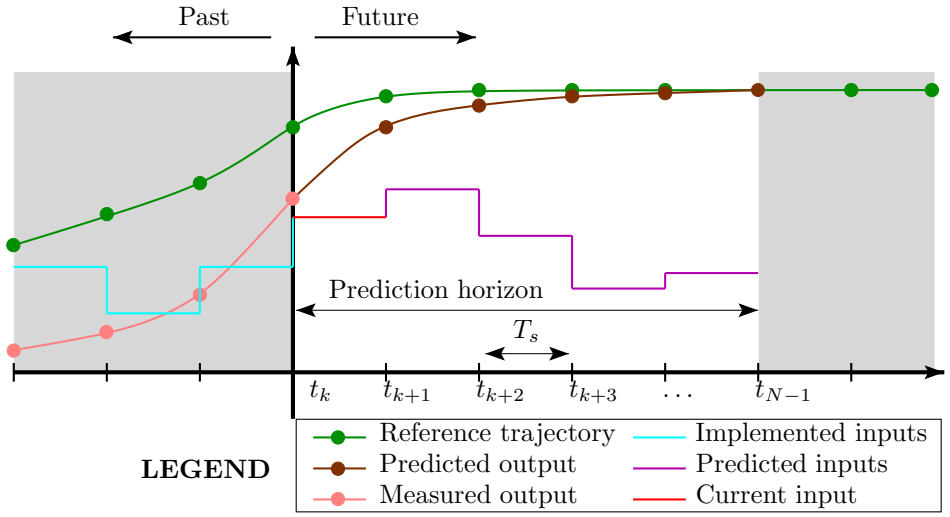


Figure 3.4: Characteristic behavior of a receding horizon control policy.

3.2.3 Explicit Solution of MPC Problem

The objective here is to employ *parametric programming* (Bemporad et al. 2002, Borrelli 2003) to pre-calculate the optimal control inputs in (3.2) for *all* admissible values of initial conditions. Hence we aim at constructing, off-line, the explicit representation of the optimizer as a function of the vector of initial conditions. Then, once we need to identify the optimal control action on-line for particular measurements, we can replace optimization by a mere function evaluation. This significantly reduces computational requirements of implementation of MPC. For further details on *parametric programming* see Section 2.2.5. Lets consider the MPC problem (3.2) to be a QP problem by using a 2-Norm in the objective function (3.2a). To see now the relation between Theorem 2.2.4 and the QP problem (3.2), notice that $U = [u_0, \ldots, u_N]$ and $\xi = [x(t), d(t)]$. Moreover, the matrices H, Q, R, d, G, w,

E of the corresponding mpQP problem (2.31) can be obtained by straightforward algebraic manipulations, see e.g. Borrelli (2003). Parameters of the PWA function $U^*(\xi)$ in (2.34), i.e., gains F_i, g_i and polyhedra $\mathcal{R}_i$, can be obtained e.g. by a parametric programming solver implemented in the freely-available MPT toolbox (Kvasnica et al. 2004).

Remark 3.2.2 *For a closed-loop implementation of MPC, only the first element of U^*, i.e., u_0^*, needs to be applied to the plant at each time instant. Therefore the explicit receding-horizon feedback law is given by*

$$u^*(t) = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} U^*(\xi) = \tilde{F}_i \xi + \tilde{g}_i, \text{ if } \xi \in \mathcal{R}_i, \tag{3.3}$$

where $\tilde{F}_i$, $\tilde{g}_i$ are obtained from F_i, g_i by retaining only the first row of a corresponding matrix. □

3.3 Summary and Further Reading

This chapter briefly presented the history and the mathematical basis behind the model predictive control. The standard formulation of the MPC problem was introduced, which is further modified for case specific building control problems in Section 5.3. From the implementation point of view, two solution methods were discussed. First, the *on-line solution* which is based on solving the optimization problem at each sampling instant, following the principle of the *receding horizon control*. And second, the *explicit solution* which exploits the *multi-parametric programming* to pre-calculate the entire solution *off-line* as a function of the initial conditions. Both approaches are further used in the context of the building climate control applications investigated in Chapter 6.

There are several publications comprehensively covering the theoretical and practical issues of MPC techniques. Tutorial overview on MPC with the focus on control engineers can be found in Rawlings (2000). Allgower et al. (1999) provides a more comprehensive overview of nonlinear MPC and moving horizon estimation (MHE). Review of theoretical results on the closed-loop behaviour of MPC algorithms can be found in Mayne et al. (2000). The paper by Mayne (2014) recalls a few past achievements in MPC, gives an overview of some current developments and suggests a few avenues for future research. On a top of that, interesting surveys on MPC technology can be found in many papers, e.g. García et al. (1989), Mayne (1997), Morari and Lee (1991), Muske and Rawlings (1993), Rawlings et al. (1994), Ricker (1991), or in books, e.g. Allgower and Zheng (2000), Camacho and Bordons (2004), Kouvaritakis and Cannon (2001), Maciejowski (2002), Mayne et al. (2000).

Part II

Contributions

Chapter 4

Building Modeling

All models are wrong, but some are useful.

George E. P. Box

The mathematical models of physical plants play a vital role in many areas, including the control synthesis, verification and simulation. They represent a mathematical abstraction that should on one hand be sufficiently accurate to capture the dynamical behavior of the plant and, on the other hand, sufficiently simple as to render the control synthesis easy.

4.1 Basic Concepts and Modeling Tools

Despite these intensive research efforts the commercialization of MPC in building sector is still in its early stages. This is partially due to the lack of direct comparison (i.e., for the same scenario) of different optimization algorithms, of different controller models and their prediction performance, of the simulation parameters such as sampling time, prediction horizon and of climate forecast, as pointed out in the review paper by Hilliard et al. (2015). The main difficulty remains, however, to obtain a good controller model of the whole building with a minimum of effort as it is the most time-consuming part (Li and Wen 2014, Prívara et al. 2011b, Sourbron et al. 2013, Široký et al. 2011). Detailed building energy simulation software (BES) allow accurate building modeling but generate models which are too complex to be used in efficient optimization algorithms (Sourbron et al. 2013, Široký et al. 2011). Low order linear models are usually preferred due to their computational tractability (Hazyuk et al. 2012). Therefore, simplified models need to be generated by means of grey-box (Bacher and Madsen 2011, Prívara et al. 2013, Reynders et al. 2014, Sourbron et al. 2013) or black-box system identification such as auto regressive (Yun et al. 2012), subspace (Ferkl and Široký 2010, Prívara et al. 2011a) and artificial neural network methods (Ruano et al. 2006) or by simplified white-box modeling (Gorecki et al. 2015, Gyalistras and Gwerder 2009, Lehmann et al. 2013, Picard et al. 2015).

While black-box identification has the advantage that no prior knowledge of the system is required and that it can deal more efficiently with large sets of data, its prediction performance for longer time horizons (e.g., more than 12 hours) is not sufficiently accurate (Prívara et al. 2013). Grey-box system identification is more suitable for a long time horizon but the method becomes very costly for large multiple-input multiple-output (MIMO) systems. As shown by (Bacher and Madsen 2011, De Coninck 2015, Reynders et al. 2014) a good choice of the structure of the grey-box model, i.e., its order, its inputs and its states, is crucial for its performance but this choice is very case specific. Therefore authors involved in the opti-control project (Gyalistras and Gwerder 2009, Lehmann et al. 2013, Sturzenegger et al. 2014), and others (Gorecki et al. 2015, Picard et al. 2015) opted for a linear white-box approach where the model is set up based on geometrical and on physical data of the building and simplified physical laws. The authors showed that this simplified approach could mimic the results (typically expressed as operative temperatures) of the more complex models obtained with BES software within an error margin of $\pm$ 0.5 to 1 K. By applying model order reduction methods, the complexity of the obtained linear model can be further reduced (Foucquier et al. 2013, Gouda et al. 2002, Kim and Braun 2012, Sturzenegger et al. 2014). Both for the grey-box and for the white-box approach, the necessary level of model complexity in order to obtain a good MPC still remains unknown and no systematic method to determine this optimal model complexity is available (Harish and Kumar 2016, Li and Wen 2014).

The building modeling plays a crucial role when applying MPC as a climate controller. However, the analysis of the optimal choice of the controller model complexity is difficult and case varying. The results presented by (Picard et al. 2016) indicate that the MPC performance is very sensitive to the prediction accuracy of the controller model. They showed that both, grey-box as well as white-box approach can lead to an efficient MPC as long as very accurate identification data sets are available. However, for the considered simulation case, the white-box MPC resulted in a better thermal comfort and used only 50% of the energy comparing to best grey-box MPC.

Some studies have investigated the influence of the model order on the model off-line prediction performance (Kramer et al. 2012). At the building component level, Gouda et al. (2002) applied a non-linear optimization technique to optimally reduced a higher building element to a second order model. Xu and Wang (2007) also reduced their model complexity to a second order model by minimizing the error between the frequency response of a higher model and their model. Fraisse et al. (2002) concludes that a wall should be represented by a fourth order model. At the multi-zone building level, Sturzenegger et al. (2014) and Kim and Braun (2012) created a linear model with a large number of states and they reduce the order by applying Model Order Reduction (MOR). Foucquier et al. (2013) also started from a high order building model but they reduced the complexity by merging different walls together.

BES programs are simulation tools that simulate the energy flows in buildings. This includes the interaction between the building envelope and its surroundings (i.e., weather, radiation heat losses, etc.), between the building envelope and its HVAC system and possibly between the HVAC system and the electrical grid. BES programs use physical equations to

describe the systems. A wide range of the software modeling tools for buildings is available nowadays. These include, but are not limited to, TRNSYS (Beckman et al. 1994), Energy Plus (Crawley et al. 2001), ESP-r (Yahiaoui et al. 2003), or Modelica (`Buildings` (Wetter et al. 2014), `IDEAS` (Baetens et al. 2015)). They usually consider very complex building models based on nonlinear energy and mass balances written in symbolic language. To deal with this issue the middleware softwares such as BCVTB (Wetter and Haves 2008), MLE+ (Bernal et al. 2012) and OpenBuild (Gorecki et al. 2015) were designed for making communication bridges between various tools. More comprehensive overwiev of HVAC system modeling and simulation tools can be found in Trcka and Hensen (2010), Zhou et al. (2013). Directories listing all available software tools for modeling, analysis, optimization and simulation for buildings can be found on-line in Energy (2014), EUROSIS (2014), Nghiem (2011).

In this work we used two distinct BES programs for obtaining two particular building models. First, Section 4.3 describes a simplified linear single-zone building model obtained from Indoor temperature Simulink Engineering (ISE) tool (van Schijndel 2005), which is used for straightforward evaluation of the developed controllers. And second, Section 4.3 introduces a complex linearized six-zone building model obtained from Integrated District Energy Assessment by Simulation (`IDEAS`) tool (Baetens et al. 2015), used for more elaborate analysis of the behaviour of the developed controllers.

4.2 Simplified Single-zone Building Model

In this Section we consider a linear model of a one-zone building, obtained from ISE. In general, ISE is a free, MATLAB-based modeling tool for simulation of the indoor temperature of a single-zone building. It uses a linear model and provides a user-friendly graphical interface to Simulink. Contrary to the complex modeling tools mentioned above, models provided by ISE are directly suitable for control synthesis. Another advantage is that ISE is standalone, i.e., it does not rely on any other external software packages and is based entirely on MATLAB/Simulink, which allows easily verify the performance of various control strategies just by wrapping any MATLAB-based control algorithm as a Simulink S-function.

The model has 4 state variables, denoted as x^1 to x^4 in the sequel. Here, x^1 is the floor temperature, x^2 represents the internal facade temperature, x^3 is the external facade temperature, and x^4 stands for the internal room temperature. All temperatures are expressed in °C. The model considers a single control input u, which represents that amount of heat injected to the zone, expressed in watts. Moreover, the model also features 3 disturbance variables $d^1, \ldots, d^3$. Here, d^1 is the external temperature (in °C), d^2 is the heat generated inside in the zone due to occupancy (in W), and d^3 is the solar radiation which heats the exterior of the building (in W).

The model can be compactly represented by a linear state-space model in the discrete-time

domain

$$x_{k+1} = Ax_k + Bu_k + Ed_k \tag{4.1a}$$
$$y_k = Cx_k \tag{4.1b}$$

where x is the state vector, d is the vector of disturbances, subindex k denotes the time period, and $A \in \mathbb{R}^{4\times4}$, $B \in \mathbb{R}^4$, $E \in \mathbb{R}^{4\times3}$ are the state-update matrices. Following values of A, B, E in (4.1) were extracted from ISE toolbox with sampling frequency $T_s = 15$ seconds:

$$A = \begin{bmatrix} 0.9997 & 0 & 0 & 0.0003 \\ 0 & 0.9997 & 0 & 0.0003 \\ 0 & 0 & 0.9992 & 0 \\ 0.0177 & 0.0428 & 0 & 0.9348 \end{bmatrix}, \; B = 10^{-4} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.4421 \end{bmatrix},$$

$$E = 10^{-3} \cdot \begin{bmatrix} 0.0007 & 0 & 0.0006 \\ 0.0007 & 0 & 0 \\ 0.8111 & 0 & 0 \\ 4.6897 & 0.0442 & 0.0088 \end{bmatrix}, \; C = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}.$$

Instead of considering an analytic model for prediction of disturbances, the ISE tool uses historical data for external temperatures (d^1) and solar radiations (d^3). For the heat generated by the occupancy, the ISE model considers $d^2 = 500\,\mathrm{W}$ during daytime hours, and $d^2 = 0\,\mathrm{W}$ otherwise.

4.3 Complex Six-zone Building Model

This section describes the modeling of an existing house (Section 4.3.1) in the BES program `IDEAS`. The Modelica library `IDEAS` is a recently developed district energy commodity flow modeling environment which enables multi-zone thermal building simulation, including building envelope, heating, ventilation and air-conditioning systems, and electric system simulation. The governing equations are discretised partial differential equations, ordinary differential equations and algebraic equations, which are solved simultaneously. The governing equations in `IDEAS` are presented in section 4.3.2, section 4.3.3 describes their linearization in order to obtain a SSM, and section 4.3.4 describes the applied model order reduction technique.

4.3.1 Building Description

The plant model is based on an existing 6-rooms terraced house in Bruges, Belgium (see Fig. 4.1) with general parameter values given by Table 4.1. The heating system is composed of one radiator per room fed by a central gas-boiler. The original building is badly insulated and it has a poor air-tightness. The column *Original* of Table 4.2 gives its overall heat transfer coefficient (U-value), its maximum volume air change per hour (ACH) and the

composition of its outer walls, floors, windows and roof. For the renovated case, the U-value is decreased by adding insulation to the outer walls (see column *Renovated* in Table 4.2). The thickness of the insulation layer varies for the different outer walls, respecting the actual renovation plans of the building. Finally, the case of a light-weight building is considered by replacing all outer walls and the roofs by an insulated wooden structure which leads to a better insulation and a lower building mass. The last row of the table indicates the number of state variables of each model.

Figure 4.1: Picture of the modeled house (Bruges, Belgium).

Table 4.1: General building parameters

Floor area	[m^2]	56
Conditioned volume	[m^3]	130.6
Total exterior surface area	[m^2]	195
Window to wall ratio	[-]	19%
Windows orientation	[-]	North-East

		Original		Renovated		Light weight	
U-value	[$W/m^2/K$]		1.28		0.65		0.36
ACH	[1/h]		8.7		4.1		4.1
Walls	[m]	concrete	0.268	concrete	0.200	wood and	0.150
	[m]	plaster	0.010	insulation	0.015-0.115	insulation	
				plaster	0.010		
Floors	[m]	reinforced concrete	0.120	reinforced concrete	0.120	reinforced concrete	0.120
	[m]	screed	0.040	insulation	0.020	insulation	0.020
	[m]	topping	0.060	screed	0.060	screed	0.060
	[m]	tiles	0.030	tiles	0.030	tiles	0.030
Windows		double glass g=0.75, U=1.4		double glass g=0.75, U=1.4		double glass g=0.75, U=1.4	
Roof	[m]	fibre-cement	0.180	fibre-cement	0.180	wood and	0.200
	[m]	insulation	0.080	insulation	0.080	insulation	
	[m]	plaster	0.010	plaster	0.010		
# States			283		286		250

Table 4.2: Parameter values and number of states in the BES model for the original, the renovated and the light weight buildings.

4.3.2 Building Thermal Model

In this study, only the building envelope, consisting of 6 thermal zones, 5 windows, 11 outer walls, 5 boundary walls with neighboring buildings, 6 roof surfaces, 3 floor surfaces on the ground, 3 floor surfaces between the ground floor and the first floor, and 6 internal walls between the zones, is considered. The heating system is idealized as a perfectly controllable, limited heating power which can directly be injected in each room. The radiators and the gas boiler are thus not modeled but they are replaced by one heat input per zone. The building envelope is modeled using the Modelica `IDEAS` library (Baetens et al. 2015). The following paragraph gives a brief description of the main equations of the building envelope

as they are developed in IDEAS. For a complete description we refer to Picard et al. (2015) and to Baetens (2015).

The thermal response of the building is governed by three main processes: (i) interaction with the surroundings by means of radiation and convection, (ii) heat transfer through the solid layers and, (iii) heat exchange between the different surfaces within the building.

The convective heat flow $\dot{Q}_{\mathrm{cv},k}(t)$ from the surroundings to an outer surface A_k is described by:

$$\begin{aligned} \dot{Q}_{\mathrm{cv},k}(t) &= h_{\mathrm{cv}}(t) A_k \left(T_{\mathrm{db}}(t) - T_{\mathrm{s},k}(t)\right) \\ h_{\mathrm{cv}}(t) &= \max\left\{5.01(v_{10}(t))^{0.85}, 5.6\right\} \mathrm{W/m^2K} \end{aligned} \tag{4.2}$$

with convective heat transfer coefficient $h_{\mathrm{cv}}(t)$, dry bulb ambient temperature $T_{\mathrm{db}}(t)$, surface temperature $T_{\mathrm{s},k}(t)$ and the undisturbed wind speed at 10 meters above the ground $v_{10}(t)$. The longwave radiation heat flow $\dot{Q}_{\mathrm{lw},k}(t)$ from the surface to the surroundings is modeled using Boltzman's law:

$$\dot{Q}_{\mathrm{lw},k}(t) = \sigma \epsilon_{\mathrm{lw},k} A_k \left(T^4_{\mathrm{s},k}(t) - T^4_{env}\right) \tag{4.3a}$$

$$T^4_{env} = F_{ce,k} T^4_{ce}(t) - (1 - F_{ce,k}) T^4_{\mathrm{db}}(t) \tag{4.3b}$$

$$F_{ce,k} = \frac{1 + \cos i_k}{2} \tag{4.3c}$$

with the Stefan-Boltzmann constant σ, surface long-wave emissivity $\epsilon_{\mathrm{lw},k}$, surface, celestial dome, dry bulb and so-called environment temperatures $T_{\mathrm{s},k}(t)$, $T_{\mathrm{ce}}(t)$, $T_{\mathrm{db}}(t)$ and T_{env}, respectively, view factor $F_{\mathrm{ce},k}$ between the surface k and the celestial dome, and inclination of the surface i_k. The short-wave solar irradiation absorbed by the exterior surface k equals:

$$\dot{Q}_{\mathrm{sw},k}(t) = \epsilon_{\mathrm{sw},k} A_k E_{\mathrm{e},k}(t) \tag{4.4}$$

with surface short-wave emissivity $\epsilon_{\mathrm{sw},k}$ and incident solar irradiation $E_{\mathrm{e},k}(t)$ on surface A_k as a function of time. Finally, the absorption and transmission through glazing is described by highly non-linear functions which depend on the spectral properties of the window, on the angle of incidence of the sun and on possible shading, see Picard et al. (2015).

The heat transfer through walls and floors is approximated as a 1-D partial differential equation which is discretized using a finite volume approach. Similarly, the temperature of zone i (T_i) is computed as:

$$C_{\mathrm{p,eq}} \frac{\partial T_i}{\partial t} = \sum_j Q_j \tag{4.5}$$

with equivalent thermal capacity $C_{\mathrm{p,eq}}$ for the air and the furniture in the zone and $\sum_j Q_j$ the sum of all convective and radiative heat flows to the air node of the zone.

Finally, interior surfaces can exchange heat with the air of the zone by means of convection:

$$\dot{Q}_{\mathrm{cv},k}(t) = h_{\mathrm{cv},k}(t) A_k \left(T_{\mathrm{db}}(t) - T_{\mathrm{s},k}(t)\right) \tag{4.6}$$

with convective heat transfer coefficient $h_{\mathrm{cv},k}(t)$ which depends on the inclination of the

surface and on the temperature difference between the surface and the air. All surfaces of a same zone can also interact by means of long-wave radiation. This interaction is modeled using the radiant star model approach:

$$\dot{Q}_{k\rightarrow \mathrm{star}}(t) = \frac{\sigma A_k}{R_k}\left(T_k^4(t) - T_{\mathrm{star}}^4(t)\right) \tag{4.7}$$

with a distribution coefficient R_k and the temperature of the fictive node T_{star} used in the radiant star model. Here $\dot{Q}_{k\rightarrow \mathrm{star}}(t)$ stands for the heat flow from surface k at temperature T_k to the fictive temperature T_{star}.

4.3.3 Linearization of the Building Thermal Model

The Modelica building envelope model, as implemented using the `IDEAS` library, is not directly usable as controller model due to its high complexity. As explained in Section 6.3.1, the Modelica reference building model is linearized around a working point in order to obtain a linear SSM. This corresponds to linearizing equations (4.2), (4.3), (4.6) and (4.7) around a working point. For typical European weather the linearization error for these equations remains typically below 1 K. However, the equations for the solar transmission and absorption through the windows are highly non-linear and they should not be linearized. The solar transmission and absorption are instead pre-computed using the `IDEAS` model and they are considered as inputs to the linearized SSM. For a complete description of the linearization process we refer to Picard et al. (2015).
The obtained SSM has the following form:

$$\frac{\partial x(t)}{\partial t} = A_{\mathrm{c}}x(t) + B_{\mathrm{c}}u(t) \tag{4.8a}$$

$$y(t) = C_{\mathrm{c}}x(t) + D_{\mathrm{c}}u(t) \tag{4.8b}$$

All states x represent temperatures. The input vector u contains the control variables, i.e., the heat flow from each radiator to the rooms (composed of 40% of radiative and 60% of convective heat flow), and the disturbances, i.e., the heat absorbed and the direct and diffuse solar radiation transmitted by each window, the direct, diffuse solar radiation and the environment temperature (i.e. a radiation temperature taking both the environment and the sky temperature into account) per orientation and inclination present in the model, the ambient temperature, and the ground temperature. Fig. 4.2 gives the temperature error between the non-linear `IDEAS` models for the three building types and the obtained SSM. The errors are computed for a full year open-loop simulation. A standard weather file of Uccle, Belgium (Meteotest 2009) is used to represent the weather condition and each zone temperature is kept within its comfort band using a PID-controller. All inputs of the non-linear and the linear models are exactly the same. As Fig. 4.2 shows, all outlier errors are below $\pm$ 1K and the median of the error is close to zero for each zone. This confirms that the obtained SSMs with pre-computed inputs are accurate approximations of the non-linear `IDEAS` models. The obtained SSMs will be further referenced as *building models*. The overall dimensions of the used building models are summarized in the Table 4.3.

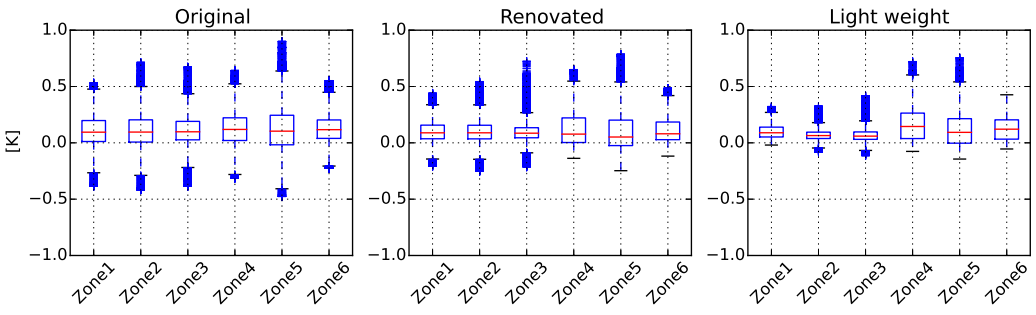


Figure 4.2: Boxplot of the temperature error between the non-linear `IDEAS` models and their linear state space models for a full year open-loop simulation. The errors are given for each building type. The centered line gives the median, the box gives the first and third quartiles, the wiskers contain 99.5% of the data, and the crosses are the outliers.

Notation	Description	Values
n_{x}	number of states	$250,283,286$
n_{u}	number of inputs	6
n_{y}	number of outputs	6
n_{d}	number of measured disturbances	44

Table 4.3: Dimensions of the three building models.

4.3.4 Model Order Reduction

Model order reduction (MOR) is an umbrella term for methods used for reducing the computational complexity of mathematical models. The method reduces the model associated state-space dimensions in order to reduce the computational cost of model evaluation. The reduced model is, however, less accurate.

Balanced Truncation

In this work, the square root balanced truncation algorithm is used to obtain the reduced order models (ROMs) of different orders. The command `reduce` of MATLAB with default settings is used. This method is based on the Hankel Singular Values (HSV) and was chosen because it guarantees an error bound and preserves most of the system characteristics in terms of stability, frequency, and time responses (Antoulas 2005). Furthermore the HSV of the building models decrease rapidly, the HSV based method gives accurate models even for very low-orders (Antoulas and Sorensen 2001). The error bound of the balanced truncation method is given by:

$$\sigma_m \leq ||M - \hat{M}||_\infty \leq 2 \sum_{i=m+1}^{n} \sigma_i. \tag{4.9}$$

with σ_i the HSVs of the original model, M and $\hat{M}$ the amplitude of the frequency response of the original and of the reduced models, $||M-\hat{M}||_\infty$ their infinity norm (i.e., the maximum difference between the two responses) and n and m the order of the original and of the reduced models (Enns 1984). Note that the HSV σ_i are sorted from large to small. For a complete description of the calculation of the HSV and of the balancing method we refer to Antoulas (2005) and Enns (1984). Based on computed error bounds, a set of ROMs with orders ranging from 4 to 100 is chosen to investigate the influence of the model complexity.

Reduced Order Model Initialization and Discretization

When applying MOR, the initial state values also need to be transformed in their reduced form. However, the MATLAB function `reduce` does not provide the transformation matrix. As the physical meaning of the initial states for the reduced models is lost by MOR, the initialization of reduced models is not straightforward. This section describes how the original SSM can be adapted to have zero initial state values without changing its input-output behavior.

We assume a LTI SSM in continuous time domain with a given initial states value $x_0 = 293.15$ K. Because x_0 is a constant the SSM (4.8) is equivalent to:

$$\frac{\partial\left(x(t)-x_0\right)}{\partial t} = A_\mathrm{c}\left(x(t)-x_0\right) + B_\mathrm{c}u(t) + A_\mathrm{c}x_0 \tag{4.10a}$$

$$y(t) = C_\mathrm{c}\left(x(t)-x_0\right) + D_\mathrm{c}u(t) + C_\mathrm{c}x_0 \tag{4.10b}$$

By substitution $\bar{x}(t) \coloneqq (x(t)-x_0)$, the model can be compactly rewritten as follows.

$$\frac{\partial\bar{x}(t)}{\partial t} = A_\mathrm{c}\bar{x}(t) + \begin{bmatrix}B_\mathrm{c} & A_\mathrm{c}x_0\end{bmatrix}\begin{bmatrix}u(t)\\1\end{bmatrix} \tag{4.11a}$$

$$y(t) = C_\mathrm{c}\bar{x}(t) + \begin{bmatrix}D_\mathrm{c} & C_\mathrm{c}x_0\end{bmatrix}\begin{bmatrix}u(t)\\1\end{bmatrix} \tag{4.11b}$$

The new SSM with state variables $\bar{x}$ has an initial states vector $\bar{x}_0 = 0$. The reduced model can now also be initialized at zero.

The discretization of the transformed continuous SSM (4.11) is necessary because the controller design and the simulations will be performed in discrete time domain. Based on the relevant dynamics and associated time constants, the unified sampling frequency $T_s = 15$ minutes was used as a motivated choice for all investigated model types. The discretized model has the following form

$$x_{k+1} = Ax_k + Bu_k + Ed_k + G, \tag{4.12a}$$

$$y_k = Cx_k + Du_k + H. \tag{4.12b}$$

Where x_k, u_k and d_k are states, inputs and disturbances at the k-th time step, respectively. The B_c matrix of the continuous SS model (4.11) contains both the control inputs u and the disturbances d. However, for control purposes it is necessary to separate them into

individual matrices. The B_c matrix is therefore split into an input matrix and a disturbance matrix which correspond, after discretization, to the matrices B and E of Eq. 4.12. The constant value matrices G and H are necessary to include the initial conditions, as explained above.

4.4 Summary

This chapter presented the basic principles of the building modeling as a crucial part of the model predictive building control. The main difficulty here is to obtain an accurate building model with a minimum of effort as it is the most time-consuming part. The trade-off between the model accuracy and simplicity is discussed. Building energy simulation software (BES) allow accurate modeling but generate models too complex to be used efficiently in optimization algorithms. Therefore, for control purposes, lower order linear models are preferred due to their computational tractability. Three main approaches to the building modeling are introduced, namely black-, grey- and white-box models, together with the discussion about their advantages and disadvantages. Two different building models are introduced. First, simplified single-zone low order linear model obtained from MATLAB's ISE toolbox (van Schijndel 2005). And second, complex six-zone higher order linearized model obtained from Modelica-based `IDEAS` library (Baetens et al. 2015). These models are further used in Chapter 6 in four case studies, as both, the controller model as well as the emulator model.

Chapter 5

Building Climate Control

Life is chaotic, dangerous, and surprising. Buildings should reflect that.

Frank Gehry

Building climate controllers are responsible for the comfort experienced in buildings. The controller controls the HVAC of the building ensuring that comfort (i.e., temperature, CO_2 concentration, etc.) remains in each room in its prescribed time-dependent band. As a similar comfort can be achieved with different sequences of control actions, the required energy to maintain the building in its comfort bound can vary significantly with the controller algorithm type. The meaning of the symbols used to describe the variables of the different controllers used in this section are listed in Table 5.1.
In this section, we focus on thermal comfort, whereby the thermal comfort and the energy use objectives are firstly defined (see Section 5.1). Section 5.2 introduces two common building controller types: a rule-based-controller (RBC) and a proportional-integral-derivative (PID) controller. Section 5.3 discusses the different formulations of a model predictive controller (MPC) considered in this work. Finally, Section 5.4 introduces the novel approach for synthesis of simple yet well-performing approximations of MPC via regression algorithms.

5.1 Control Objectives

A Building Automation and Control System (BACS) governs buildings such that certain comfort and economic criteria are fulfilled. These include internal room temperatures, air quality, lighting etc. Instead of tracking particular reference values, a BACS typically considers ranges, also called comfort bands. The task is then to manipulate the building inputs such that required comfort criteria are kept within the band while the total amount of used energy is minimized. It should be noted that the comfort and the energetic criteria are often competing as the increase of comfort typically leads to an increase of energy use.

Notation	Units	Description	Control setup
x	$[K, -]$	temperatures for the SSM, no physical meaning for the ROM	states
y	$[K]$	room temperatures	outputs
r	$[K]$	desired room temperatures	references
u	$[W]$	radiators heat flows	inputs
d	$[K, W]$	temperatures, heat flows and radiation gains	measured disturbances
p	[-]	augmented state variables	unmeasured disturbances
s	$[K]$	comfort band violations	slack variables
$\overline{y}$	$[K]$	upper comfort boundary	constraints
$\underline{y}$	$[K]$	lower comfort boundary	constraints

Table 5.1: Notation of variables used in Section 5.

In this work, thermal comfort is the most emphasized objective, treated as soft constraint to guarantee a solution.

5.1.1 Thermal Comfort

The thermal comfort objective is achieved by maintaining each room temperature y_i of the house in the comfort band as defined by the European norm ISO-7730. The lower and upper temperature bounds ($\underline{y}$, $\overline{y}$) vary between $[20, 23]°C$ and $[24, 26]°C$, respectively, as a function of the 7-days average of the ambient temperature. The comfort objective corresponds thus to the constraint:

$$\underline{y}_k - s_k \leq y_{i,k} \leq \overline{y}_k + s_k \tag{5.1}$$

with s the relaxation variable which should be minimized and the index k the sampling time.

5.1.2 Minimization of Energy Use

The second objective is to use a minimal amount of energy to achieve the comfort. In this work, the energy use is the sum of the thermal energy injected by all heating systems. The RBC and the PID controllers have been tuned such that they keep the zone temperatures as close as possible to the lower comfort bound, but without endangering the comfort. For the case of MPC, the square of the energy use instead of its absolute value is minimized in order to avoid power peaks.

5.1.3 Objective Weights

It should be pointed out that the two aforementioned qualitative criteria are counteracting against each other. To drive the internal temperature towards the comfort zone, the first

objective forces the heating systems to become active. On the other hand, zero heating is preferred by the second objective. Therefore so-called weighting parameters need to be assigned to each objective as to indicate its preference. Needless to say, achieving comfort with minimal energy must be done while satisfying all constraints of physical equipment.

5.2 Standard Building Control Strategies

Usually the controller consists of a set of rules which determine the control action as a function of inputs (e.g., a room temperature, the outside weather conditions, etc.) and a set of set points. These types of controllers are the so-called rule-based controllers (RBC). They are widely used for residential buildings because of their simple design and configuration and their low computational demands allowing cheap hardware solutions. Their main drawbacks are that they are not adaptive, not flexible, not predictive and they need to be tuned. The RBC controllers cannot track a time-varying reference or minimize the energy necessary to stay within a bound. The RBC implementation used in this work is described in Section 5.2.1. An alternative approach is presented by the PID controllers which are better suited to track a reference. However PID controllers are difficult to tune and badly tuned PID leads to oscillations, overshoots and time-lags. Moreover, one PID controller needs to be constructed for each zone individually.

5.2.1 Rule-Based Controller

The commonly used controller for residential buildings with central heat production and radiators is a hysteresis rule based controller (RBC) also called central thermostat controller. Its working principle is as follows: a temperature sensor is placed in the main room, typically the living room. Based on this temperature and a comfort band, the central heating is turned on or off. Hot water can only flow to the radiators when the central heating is on. All radiators are equipped with thermostatic valves, except those in the room of the thermostat. The valve acts as a proportional controller by controlling the water mass flow rate through the radiator and so controlling its power.
The supply temperature T_{sup} is for all radiators the same and it is calculated using a typical heat curve equation:

$$T_{\text{sup}} = r + \left(\frac{T_{\text{sup,n}} + T_{\text{ret,n}}}{2} - y_{j,\text{n}} \right) q^{1/m} + \frac{T_{\text{sup,n}} - T_{\text{ret,n}}}{2} q \tag{5.2}$$

$$q = \frac{r - (T_{\text{e},6h} + \epsilon)}{y_{j,\text{n}} - (T_{\text{e,n}} + \epsilon)} \tag{5.3}$$

where subscripts sup and ret stand for supply and return water temperatures, e stands for external air temperature, n refers to the nominal conditions ($T_{\text{sup, n}} = 70°C$, $T_{\text{ret,n}} = 50°C$, $T_{\text{e,n}} = -10°C$), and the index j refers to the room with the thermostat. The exponent m depends on the heating system (for radiator, $m = 1.3$). A correction term $\epsilon = 8$ K on

the outside temperature $T_{\mathrm{e},6h}$ (averaged over 6 hours) is added to take the solar gain into account.

The binary control action z_k of the central heating in *k-th* time step, based on the temperature measurement in *j-th* (central) room $y_{j,k}$ and given reference temperature r_k is defined by a switching rule of the relay based thermostat given by following equation

$$z_k = \begin{cases} 1 & \text{if} \quad (z_{k-1} = 1 \ \wedge \ (y_{j,k} \leq r_k + \gamma)) \vee \\ & \qquad (z_{k-1} = 0 \ \wedge \ (y_{j,k} \leq r_k - \gamma)) \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

where $\wedge$ is the logic conjunction and $\vee$ denotes for the logic disjunction. The parameter 2γ here represents the width of the hysteresis. The values of the control action represent the heating mode if $z_k = 1$ and not heating if $z_k = 0$.

Finally the actual power $u_{i,k}$ delivered by the *i-th* radiator to the *i-th* zone at the *k-th* time step is given by:

$$u_{i,k} = \begin{cases} G_i z_k (T_{\mathrm{sup},k} - y_{i,k}), & \text{if } i = j \\ \alpha_i G_i z_k (T_{\mathrm{sup},k} - y_{i,k}), & \text{otherwise} \end{cases} \tag{5.5}$$

with $\alpha_i \in [0, 1]$ the proportional gain of the thermostatic valve and G_i the total thermal conductance of the radiator. Each radiator is sized such that its maximum power is required when the outside temperature drops to $-10°\mathrm{C}$. The same power bounds are used for the other controllers.

5.3 Model Predictive Building Control

Model predictive control (MPC) is a control strategy which optimizes the control actions over a finite time-horizon by anticipating the effect of these actions, of the future disturbances and of the future constraints on the system. The ability of anticipation comes from the mathematical model of the system (i.e., the *controller model*) and the prediction of the future disturbances. Moreover MPC has the ability to directly take into account given control objectives (see Section 5.1), by penalizing them in the cost function of the optimization problem. The main drawback of this strategy is the difficulty of obtaining an accurate and computationally efficient controller model and the high computational cost (CPU) needed to solve the optimization problem.

The following sections describe the general controller setup (Section 5.3.1), the state estimator (Section 5.3.2). The deterministic MPC objective function and constraints are cast in Section 5.3.3, together with the *state condensing method* which is used to speed up the algorithm. Section 5.3.4 introduces the stochastic MPC formulation with probabilistic modeling of the disturbances, and computationally efficient explicit solution of the corresponding optimization problem.

5.3.1 Model Predictive Control Setup

Fig. 5.1 illustrates the MPC setup used in this work. The control loop consists of the building model representing the real building, the estimator, and the MPC controller which is composed of a controller model, an objective function and a set of constraints. Two cases are considered: a standard MPC (S-MPC) and an off-set free MPC (OSF-MPC). In the case of S-MPC, the estimator is used to estimate the state values $\hat{x}$ of the controller model. In the case of OSF-MPC, a set of extra states p is added to the controller model to take into account the mismatch between the controller model and the building model. In that case the estimator also estimates $\hat{p}$ (see Section 5.3.2). We assume that the building is affected by disturbances d (e.g., weather conditions), which are measured and used as perfect predictions (with zero prediction error) by MPC and the estimator. MPC optimally manipulates the control action u, which represents the heat flow injected in the building. The feedback vector y consists of temperatures.

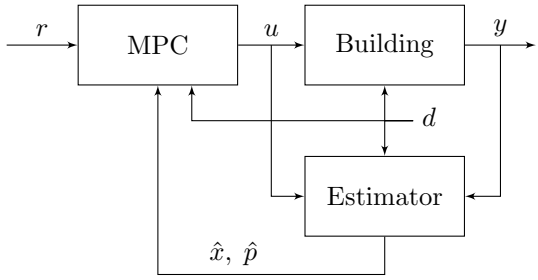


Figure 5.1: Schematic representation of the closed-loop system. Here, d are measured disturbances, y denotes the outputs, r are the output references, u are the control actions, and $\hat{x}$, $\hat{p}$ denote the estimates of the buildings states and building model mismatch, respectively.

5.3.2 State and Disturbance Estimation

A state observer is an algorithm that computes an estimate of the state values of the controller model based on the measurements of the inputs and outputs of the building model. In this work, a standard Luenberger observer is used under the following form:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L\left(y_{\mathrm{m},k} - \hat{y}_{k|k-1}\right) \tag{5.6a}$$

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_{k|k} + Ed_{k|k} \tag{5.6b}$$

$$\hat{y}_{k|k} = C\hat{x}_{k|k} + Du_{k|k} \tag{5.6c}$$

where the estimator gain L given as discrete stationary Kalman filter, can be computed e.g. by the discrete Riccati equation using the `dlqe` MATLAB function. The subscript $k|k-1$ means that the value is estimated for time k based on the observed value of time $k-1$. The vector y_{m} denotes the vector of the measured outputs and the vectors $\hat{x}_k$ and $\hat{y}_k$ stand for the estimated states and outputs of the controller model, respectively.

Remark 5.3.1 *The matrices G and H due to the initialization transformation (see Section 4.3.4) are further omitted for the clarity of the notations. However, they are still included in the calculations.* □

In the case of OSF-MPC, a set of extra fictitious states p, representing unmeasured internal disturbances, is added to the controller model to take the building model mismatch into account (Muske and Badgwell 2002). One extra state with a constant dynamic is added per each output of the controller model (Pannocchia and Rawlings 2003). This approach, also called the active disturbance rejection control, allows us to consider a simpler controller model, since the modeling error is compensated in real time. The augmented controller model is now given by:

$$\underbrace{\begin{bmatrix}\hat{x}_{k+1}\\ \hat{p}_{k+1}\end{bmatrix}}_{\tilde{x}_{k+1}} = \underbrace{\begin{bmatrix}A & \mathbf{0}\\ \mathbf{0} & I\end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix}\hat{x}_k\\ \hat{p}_k\end{bmatrix}}_{\tilde{x}_k} + \underbrace{\begin{bmatrix}B\\ \mathbf{0}\end{bmatrix}}_{\tilde{B}} u_k + \underbrace{\begin{bmatrix}E\\ \mathbf{0}\end{bmatrix}}_{\tilde{E}} d_k, \tag{5.7a}$$

$$\hat{y}_k = \underbrace{\begin{bmatrix}C & F\end{bmatrix}}_{\tilde{C}} \begin{bmatrix}\hat{x}_k\\ \hat{p}_k\end{bmatrix} + \underbrace{\begin{bmatrix}D\\ \mathbf{0}\end{bmatrix}}_{\tilde{D}} u_k. \tag{5.7b}$$

where the output disturbance matrix F was chosen as a full column rank identity matrix and all other matrices are the same as in Eq. 4.12.

Remark 5.3.2 *For the clarity of the notation, only the S-MPC equations will be used further. The equations for the case of OSF-MPC are obtained by replacing the matrices (A,B,C,D) by their augmented equivalent* $(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$*. For the observer, the gain L is also recomputed using the augmented matrices.* □

5.3.3 Deterministic MPC Formulations

The aim of this section is to devise an optimal controller policy which minimizes the energy used while maximizing the thermal comfort for the occupants. The MPC optimization problem is formulated in a quadratic way as follows

$$\min_{u_0,\dots,u_{N-1}} \sum_{k=0}^{N-1} \left(||s_k||^2_{Q_s} + ||u_k||^2_{Q_u} \right) \tag{5.8a}$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k + Ed_k, \qquad k \in \mathbb{N}_0^{N-1} \tag{5.8b}$$

$$y_k = Cx_k + Du_k, \qquad k \in \mathbb{N}_0^{N-1} \tag{5.8c}$$

$$\underline{y}_k - s_k \leq y_k \leq \overline{y}_k + s_k, \qquad k \in \mathbb{N}_0^{N-1} \tag{5.8d}$$

$$\underline{u} \leq u_k \leq \overline{u}, \qquad k \in \mathbb{N}_0^{N-1} \tag{5.8e}$$

$$x_0 = \hat{x}(t), \tag{5.8f}$$

$$d_0 = d(t). \tag{5.8g}$$

where x_k, u_k and d_k represent the values of states, the inputs and the disturbances, respectively, predicted at the k-th step of the prediction horizon N. The predictions are obtained from the LTI prediction model given by the equations (5.8b) and (5.8c). The $\underline{y}_k$ and $\overline{y}_k$ parameters represent the comfort band given by the constraints (5.8d), where the variables s_k are used as the indicators of a comfort band violation. The min/max constraints for the control input amplitude are given by (5.8e). The initial conditions of the problem (5.8f) and (5.8g) are given as the state estimates and the measurements of the disturbances. For particular initial conditions, the optimization computes the sequence $u_0^*, \ldots, u_{N-1}^*$ of control inputs that are optimal with respect to the quadratic objective function (5.8a) and the constraints. The term $\|a\|_Q^2$ in the objective function represents the weighted squared 2-norm, i.e., $a^T Q a$, with the weighting matrices Q_s and Q_u given as positive definite diagonal matrices. The first term of the quadratic cost function minimizes the square of comfort band violations, while the second term minimizes the square of the energy used.
Denote by ξ the vector which encapsulates all time-varying parameters of (5.8), i.e. the current states $x(t)$, current and future disturbances $d(t), \ldots, d(t+NT_s)$, as well as comfort boundaries signals $\underline{y}(t), \ldots, \underline{y}(t+NT_s)$ and $\overline{y}(t), \ldots, \overline{y}(t+NT_s)$. The receding horizon feedback law is then given by

$$u(\xi) = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} U_N, \tag{5.9}$$

where $\mathbf{I}$ and $\mathbf{0}$ represent, respectively, identity and zero matrices of appropriate dimensions. Note that the optimal open-loop sequence U_N in (5.9) is defined as the optimal solution of (5.8), formulated for a particular initial conditions given in (5.8f). Therefore to obtain the optimal control action for a particular value of ξ from (5.9), one needs to solve (5.8) at each sampling instant. Such a procedure, however, requires significant computational effort and must be implemented on a hardware platform that allows to run optimization algorithms. Such a requirement is in contrast to our objective of implementing the control strategy on very simple hardware with limited computational and memory storage resources. Therefore in the next section we show how to derive simple regression-based controllers that mimic the behavior of MPC policy (5.9) and can be implemented on simple hardware.

State Condensing

In the problem formulation (5.8), each input and each state is considered as an optimization variable. However, the computation cost to solve a linear-quadratic control problem is $\mathcal{O}\left(N^3(n_\mathrm{x}+n_\mathrm{u})^3\right)$, with N the control horizon, n_x the number of states and n_u the number of inputs, Frison and Jorgensen (2013). If the solver makes use of the sparsity of the problem, the complexity of the problem becomes $\mathcal{O}\left(N(n_\mathrm{x}+n_\mathrm{u})^3\right)$. Another approach is to use the so-called *state condensing* method which rewrites the large and sparse system into a smaller but denser form. In this method only the inputs are considered as optimization variable and the complexity becomes $\mathcal{O}\left(N^3 n_\mathrm{u}^3\right)$. Due to the large number of states and relatively small horizon, the condensing method is the most appropriate method for this study.

The states can be eliminated by straightforward linear algebra substitutions as follows:

$$x_1 = Ax_0 + Bu_0 + Ed_0 \tag{5.10a}$$
$$x_2 = A\left(Ax_0 + Bu_0 + Ed_0\right) + Bu_1 + Ed_1 \tag{5.10b}$$
$$\vdots$$
$$\begin{aligned} x_{k+1} = {} & A^{k+1}x_0 + \dots \\ & \left[A^k B \dots AB\ B\right]\left[u_0^T \dots u_k^T\right]^T + \dots \\ & \left[A^k E \dots AE\ E\right]\left[d_0^T \dots d_k^T\right]^T \end{aligned} \tag{5.10c}$$
$$\begin{aligned} y_k = {} & CA^k x_0 + \dots \\ & C\left[A^{k-1}B \dots AB\ B\right]\left[u_0^T \dots u_{k-1}^T\right]^T + \dots \\ & C\left[A^{k-1}E \dots AE\ E\right]\left[d_0^T \dots d_{k-1}^T\right]^T + Du_k + Fp_0 \end{aligned} \tag{5.10d}$$

The state variables from the previous time instants are substituted into the subsequent state prediction equations. Recursively adopting this procedure we obtain an explicit formula (5.10c) for calculating the state update in the $(k+1)^{\text{th}}$ time step based only on the initial state condition and predicted control actions. The output equation (5.10d) with condensed states can now replace the equations (5.8b) and (5.8c) of the controller model in the original MPC problem formulation (5.8).

5.3.4 Stochastic MPC Formulations

In this section we show how to synthesize explicit representations of MPC feedback laws that maintain temperatures in a building within of a comfortable range while taking into account random evolution of external disturbances. The upside of such an explicit MPC solution stems from the fact that optimal control input can be obtained on-line by a mere function evaluation. This task can be accomplished quickly even on cheap hardware. To account for random disturbances, our formulation assumes probabilistic version of thermal comfort constraints. A finite-sampling approach can be used to convert the probabilistic bounds into deterministic constraints. To reduce the complexity, and to allow for synthesis of explicit feedbacks in reasonable time, the set of samples is furthermore pruned, depending on activity of constraints.

The simple implementation of stochastic MPC is achieved by pre-computing, off-line, the optimal solution to a given optimal control problem for all initial conditions of interest using *parametric programming* (see Section 2.2.5). This gives rise to an explicit representation of the MPC feedback law as a Piecewise Affine (PWA) function (2.1.12) that maps initial conditions onto optimal control inputs. The upside is that the on-line implementation of such controllers reduces to a mere function evaluation. This task can be performed efficiently even on cheap hardware. However, parametric programming is only applicable to MPC problems of small size (Grancharova et al. 2008).

In our setup, however, the dimensions are large. In particular, the space of initial param-

eters for the single-zone building model from Section 4.2 is 14-dimensional, hence general-purpose explicit stochastic MPC approaches cannot be readily applied. Therefore we show how to formulate the stochastic control problem such that it can be subsequently solved, and such that the solution is not of prohibitive complexity. First, by exploiting the results of Campi and Garatti (2008) we show how to replace stochastic probability constraints by a finite number of deterministic constraints. Subsequently, exploiting a particular dynamics of the building, we show that the number of deterministic constraints can be reduced substantially as to render parametric programming useful. At the end we arrive at an explicit representation of stochastic MPC that achieves a given probability of thermal comfort while simultaneously minimizing consumption of heating/cooling energy sources. Performance of the proposed stochastic scheme is then compared versus a best-case scenario (which employs fictitious perfect knowledge of future disturbance), and against a worst-case setup that employs conservative bounds on future evolution of disturbances.

Probabilistic Disturbances Modeling

There are several aspects which distinguish control of buildings less complex that control of generic plants. Foremost, buildings can be conceived as a complex, but inherently stable systems with slow dynamics. This simplifies control synthesis to some extent. E.g., one does not need to explicitly account for closed-loop stability, and slow dynamics allows to apply control methods that are based on computational-heavy optimization. However, buildings are often affected by disturbances, which need to be considered by the controller. Some of these disturbances can be measured, some can only be estimated. These include, among others, weather conditions (external conditions, cloudiness, humidity, etc.) as well as occupancy of the building. Quality of the overall building control then depends on how well we are able to estimate, or predict future evolution of these disturbances.

The nontrivial part of designing a suitable control strategy stems from presence of disturbance variables d in (4.1). In real-life situations, at each time t one can reasonably expect that the current value of the disturbance $d(t)$ is measured. Future disturbance are, however, unknown. We assume that at each time instant t we have knowledge of building's state vector $x(t)$, as well as current values of disturbances $d(t)$. Then depending on what type of knowledge we have about future disturbances, we can aim at synthesizing one of the following three control strategies.

1. If we have a reasonably accurate model to predict weather and occupancy conditions, then the values $d(t + kT_s)$ are known for $k \in \mathbb{N}_0^N$, where N is the length of the prediction window. Then we call the control strategy

$$u^*(t) = \mu(x(t), d(t), \dots, d(t+N)) \tag{5.11}$$

   the **best-case scenario**.

2. If we can bound future disturbances by $\|d(t + kT_s) - d(t + (k+1)T_s)\| \leq \omega_{\max}$ for

any $k \in \mathbb{N}_0^N$, then the control strategy

$$u^*(t) = \mu(x(t), d(t), \omega_{\max}) \tag{5.12}$$

is referred to as the **worst-case scenario**.

3. If the future disturbances are unknown, but we know that the future disturbances follow some probability distribution (2.3.2) of rate of change of disturbances. To achieve a tractable formulation of the control problem, we therefore assume that we know the probability distribution

$$\omega \sim \mathcal{N}(0, \sigma(t)) \tag{5.13}$$

such that the future disturbances at discrete time steps $t + T_s, \ldots, t + NT_s$ are given by

$$d(t + kT_s) = d(t) + k\omega, \ k \in \mathbb{N}_1^N, \tag{5.14}$$

where N denotes the prediction window over which the distribution in (5.13) is deemed reasonably accurate. Then the control strategy

$$u^*(t) = \mu(x(t), d(t), \sigma) \tag{5.15}$$

is called the **stochastic scenario**.

Remark 5.3.3 *Please note, that the disturbances either increase or decrease linearly, what means that the confidence interval of the uncertainty is growing bigger for a longer predictions. But thanks to the knowledge of the probability distribution ω for every k-th step of the prediction, we can model this behavior by simple summation of predicted disturbances as shown in formula* (5.14)*, where range of the confidence interval for predicted disturbance in k-th step is directly dependent on range confidence interval in previous sampling step k-1.* □

It should be pointed out that the first scenario is not realistic, as weather and/or occupancy varies randomly. Due to the same reason, the worst-case scenario often requires employing conservative bounds $\omega_{\max}$, which leads to deterioration of control performance. While the third stochastic case is the most natural from the practical point of view, introduction of probabilistic functions requires a modification of the thermal comfort criterion (5.1). Because the control authority is limited and due to the random nature of disturbances, the deterministic thermal comfort constraint needs to be relaxed in a probabilistic sense as follows:

$$\Pr(y_{i,k} \geq \underline{y}_k - s_k) \geq 1 - \alpha, \tag{5.16a}$$

$$\Pr(y_{i,k} \leq \overline{y}_k + s_k) \geq 1 - \alpha, \tag{5.16b}$$

where $1 - \alpha$ the denotes probability with which the constraints in (5.1) have to be satisfied for some $\alpha \in [0, 1]$.

Stochastic Comfort Zone Temperature Tracking

Our further goal will be to synthesize the state-feedback control policy $u(t) = \mu(x(t), d(t), \sigma(t))$ that maps measurements onto optimal control inputs such that maintains thermal comfort over the prediction window N, and minimizes the total energy.
Here, the deterministic MPC problem (5.8) needs to be modified for maintaining the high probability of thermal comfort satisfaction while minimizing energy consumption. The state update equation (5.8b) is modified to:

$$x_{k+1} = Ax_k + Bu_k + E(d_0 + k\omega), \tag{5.17}$$

the comfort constraints (5.8d) are replaced by (5.16), and future disturbances predicted in (5.17) employ the random variable $\omega \sim \mathcal{N}(0, \sigma(t))$, where $\sigma(t)$ is assumed to be available to the optimization. The term $d_0 + k\omega$ originates directly from (5.14).
Due to the probabilistic constraints (5.16), resulting optimization problem is hard to solve, in general. In this thesis we propose to tackle the probabilistic constraint by employing a finite number of realizations of the random variable ω, as captured by the following two lemmas.

Lemma 5.3.4 (Campi and Garatti (2008)) *Let $g(u, \omega) : \mathbb{R}^N \times \mathbb{R}^{n_d} \to \mathbb{R}$ be a function that is convex in u for any ω, and let ω be a random variable as in (5.13). Assume a probabilistic constraint*

$$\Pr(g(u, \omega) \leq 0) \geq 1 - \alpha \tag{5.18}$$

for some $\alpha \in [0, 1]$. Let $\omega^{(1)}, \ldots, \omega^{(M)}$ be M samples of the random variable independently extracted from (5.13). Then the probabilistic constraint in (5.18) is satisfied with confidence $1 - \beta$, i.e., $\Pr(\Pr(g(u, \omega) \leq 0) \geq 1 - \alpha) \geq 1 - \beta$, if

$$g(u, \omega^{(i)}) \leq 0, \ \forall i \in \mathbb{N}_1^M, \tag{5.19}$$

holds for a sufficiently large M. ■

Lemma 5.3.5 (Alamo et al. (2010)) *The number of samples M required in Lemma 5.3.4 is bounded from below by*

$$M \geq \frac{1 + N + \ln(1/\beta) + \sqrt{2(N+1)\ln(1/\beta)}}{\alpha}. \tag{5.20}$$

■

By employing Lemma 5.3.4 we can thus replace the probabilistic constraints (5.16) by a finite number M of deterministic constraints, each obtained for one of the realizations $\omega^{(i)}$ of the random variable. In addition, Lemma 5.3.5 quantifies the lower bound on the number of such realizations, which grows only moderately with the confidence measure β.
Consider the i-th realization of the random variable, i.e., $\omega^{(i)}$, and denote by

$$y_k^{(i)} = C\left(A^k x_0 + \sum_{j=0}^{k-1} A^{k-j-1}\left(Bu_j + E\left(d_0 + (j+1)\omega^{(i)}\right)\right)\right) \tag{5.21}$$

the indoor temperature, predicted at the k-th step of the prediction horizon using the disturbance $\omega^{(i)}$. Note that (5.21) follows directly by solving for $y_k = Cx_k$ from (5.17). Then the probabilistic comfort constraints (5.16) can be replaced by finite number of deterministic constraints:

$$y_k^{(i)} \geq \underline{y}_k - s_k, \ \forall i \in \mathbb{N}_1^M, \tag{5.22a}$$

$$y_k^{(i)} \leq \overline{y}_k + s_k, \ \forall i \in \mathbb{N}_1^M, \tag{5.22b}$$

where $y_k^{(i)}$ is given per (5.21). Note that (5.21) serves as a substitution in (5.22a)-(5.22b) and is *not* considered as an equality constraint. By Lemma 5.3.4, a feasible solution to a problem employing constraints (5.22) implies that the probabilistic constraints (5.16) will be satisfied with a high confidence $1 - \beta$. The initial conditions are the current state measurements $x_0 = x(t)$, current value of the disturbance vector $d_0 = d(t)$, and the M samples $\omega^{(1)}, \ldots, \omega^{(M)}$ extracted from the probability distribution (5.13) for a current value of the standard deviation $\sigma(t)$. Most importantly, the devised optimization problem is a quadratic program in decision variables $u_0, \ldots, u_N$ since the objective function is quadratic and we have finitely many linear constraints.
Therefore a control policy that provides satisfaction of thermal comfort constraints in (5.16), respects limits of the control authority in (5.8e), and minimizes the energy consumption, can be achieved as follows:

1. At time t, measure $x(t)$, $d(t)$ and obtain $\sigma(t)$, $\underline{y}(t)$, $\overline{y}(t)$.
2. Generate M samples $\omega^{(1)}, \ldots, \omega^{(M)}$ from (5.13).
3. Formulate the corresponding QP with constraints (5.22) and solve it to obtain $u_0^*, \ldots, u_N^*$.
4. Apply $u(t) = u_0^*$ to the system and repeat from the beginning at time $t + T_s$.

Explicit Stochastic MPC

Even though the explicit representation of the MPC feedback law in (3.3) provides a simple and fast implementation of MPC on embedded hardware, it suffers from the so-called *curse of dimensionality*. Simply speaking the number of polyhedral regions $\mathcal{R}_i$ grows exponentially with the number of constraints in (2.31). Therefore, from a practical point of view, explicit MPC solutions as in (2.34) can only be obtained for reasonably simple mpQP problem (2.31). Note that our QP resulting from stochastic comfort zone temperature tracking MPC formulation has $2N(M+1)$ constraints, N decision variables $(u_0, \ldots, u_N)$, and $n_\mathrm{x} + n_{\underline{y}} + n_{\overline{y}} + (M+1)n_\mathrm{d}$ parameters. Since $M \gg N$ in practice due to (5.20), the main driving factor of complexity is thus M, the number of realizations of the random variable ω employed in (5.22).
To give the reader a flavor of complexity, consider $\alpha = 0.05$ (which corresponds to a 95 % probability of satisfying the thermal comfort criterion), $N = 10$, and $\beta = 1 \cdot 10^{-7}$ (which means a 99.9999999 % confidence in Lemma 5.3.4). The graphical representations of dependancies of number of samples M to the parameter α and three different settings of the

parameter β are shown in Fig 5.2, with highlighted above mentioned setup. Then we have $M = 919$ by (5.20), hence the resulting QP employing building model from Section 4.2 has 18400 constraints and 927 parametric variables. Solving such a QP parametrically according to Theorem 2.2.4 would lead to an explicit solution defined over billions of regions, which is not practical and defeats the purpose of cheap and fast implementation of MPC on embedded hardware.

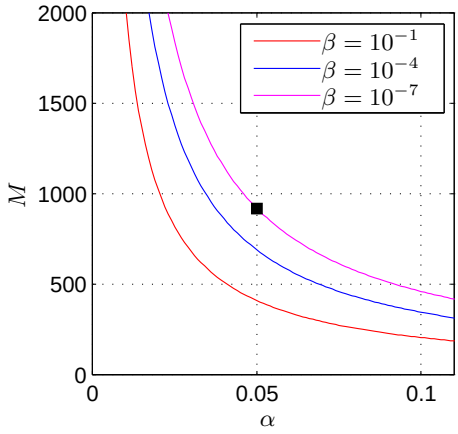


Figure 5.2: Dependance of number of M samples $\omega^{(i)}$ on parameter α, for three different settings of parameter β. Where point depicted as black square represents $M = 919$ samples for $\alpha = 0.05$, $N = 10$, and $\beta = 1 \cdot 10^{-7}$.

Fortunately, most of the constraints are redundant and can hence be discarded, allowing a tractable solution. To see this, consider the constraint in (5.22b), rewritten as

$$C(Ax_k + Bu_k + E(d_0 + k\omega^{(i)})) \leq \overline{y}_k + s_k. \tag{5.23}$$

Since the constraint is linear in all variables, it holds if and only

$$\max_i \{C(Ax_k + Bu_k + E(d_0 + k\omega^{(i)}))\} \leq \overline{y}_k + s_k, \tag{5.24}$$

which is furthermore equivalent to

$$C(Ax_k + Bu_k + Ed_0) + k \max_i \{CE\omega^{(i)}\} \leq \overline{y}_k + s_k. \tag{5.25}$$

Similarly, we have that (5.22a) holds if and only if

$$C(Ax_k + Bu_k + Ed_0) + k \min_i \{CE\omega^{(i)}\} \geq \underline{y}_k - s_k. \tag{5.26}$$

Let

$$\overline{\omega} = \arg\max_{\omega^{(i)}} \{CE\omega^{(i)}\}, \ \underline{\omega} = \arg\min_{\omega^{(i)}} \{CE\omega^{(i)}\}. \tag{5.27}$$

Remark 5.3.6 *Identification of $\underline{\omega}$ and $\overline{\omega}$ in* (5.27) *does not require any optimization, as the minima/maxima are taken element-wise from a finite set.* □

Then for any sample $\omega^{(i)}$ with $\underline{\omega} \prec \omega^{(i)} \prec \overline{\omega}$ the constraints in (5.22a)-(5.22b) are redundant. We conclude that, instead of considering M samples $\omega^{(i)}$ in (5.22), one can equivalently state the problem using only the extremal realizations $\underline{\omega}$, $\overline{\omega}$, hence $M = 2$. Using the same figures as above, this leads to a QP with only 60 constraints and 14 parameters in ξ, for which the explicit representation of the optimizer in (2.31) can be obtained rather easily.

Remark 5.3.7 *Note that the values $\underline{\omega}$ and $\overline{\omega}$ are considered as free parameters in* (5.22). *Since the samples $\omega^{(i)}$ vary in each instance of the QP, it is not possible to prune redundant constraints a-priori.* □

Implementation of stochastic explicit MPC thus requires two steps. The first one is performed completely off-line. Here, the QP is formulated using symbolic initial conditions x_0, d_0, $\underline{y}$, $\overline{y}$, $\underline{\omega}$ and $\overline{\omega}$, all concatenated into the vector ξ. Then the QP is solved parametrically for all values of ξ of interest and the explicit representation of the MPC feedback in (3.3) is obtained by the MPT toolbox. Finally, parameters of the feedback, i.e., the gains $\tilde{F}_i$, $\tilde{g}_i$, and polyhedra $\mathcal{R}_i$ are stored in the memory of the implementation hardware.
The on-line implementation of such an explicit feedback is then performed as follows:

1. At time t, measure $x(t)$, $d(t)$, and obtain $\sigma(t)$, $\underline{y}(t)$ and $\overline{y}(t)$.
2. Generate M samples $\omega^{(1)}, \ldots, \omega^{(M)}$ from (5.13).
3. From the generated samples pick $\underline{\omega}$ and $\overline{\omega}$ by (5.27).
4. Set $\xi = [x(t), d(t), \underline{y}(t), \overline{y}(t), \underline{\omega}, \overline{\omega}]$ and identify index of the polyhedron for which $\xi \in \mathcal{R}_i$. Denote the index of the "active" region by i^*.
5. Compute $u^*(t) = \tilde{F}_{i^*}\xi + \tilde{g}_{i^*}$, apply it to the system and repeat from the beginning at time $t + T_s$.

There are various ways how to identify index of the active region in Step 4. The most trivial way is to traverse through the polyhedra sequentially, stopping once $\xi \in \mathcal{R}_i$ is satisfied. Runtime complexity of such an approach is $\mathcal{O}(R)$, where R is the total number of polyhedra. More advanced approaches, such as binary search trees (Tøndel et al. 2003), can improve the runtime to $\mathcal{O}(\log_2 R)$ by pre-computing a search structure. The amount of memory required to store the PWA function (3.3) in the memory is linear in R.

5.4 Approximate Model Predictive Building Control

This section introduces a versatile framework for synthesis of simple, yet well-performing feedback strategies that mimic the behaviour of optimization-based controllers, such as those based on model predictive control (MPC). The approach is based on employing regression analysis and dimensionality reduction algorithms to derive the dependence of real-valued control inputs on measurements. The main advantage of the proposed regression-based control strategies stems from their easy implementation even on a very simple hardware.

Section 5.4.1 elaborates the methodology for the approximation of an arbitrary controller behaviour with multiple manipulated variables in the context of the building climate control. Section 5.4.2 defines the problem of interest as a multivariate regression and discusses the computational and practical aspects of the proposed approximation method. Section 5.4.3 provides the overview and differentiation of the various regression algorithms and justifies the selection of the regression trees (RT) and time delay neural networks (TDNN), which are further described in Sections 5.4.4, and 5.4.5, respectively. Section 5.4.6 introduces a systematic feature selection approach for predictive models in the scope of the building climate control applications.

5.4.1 Methodology

This work introduces the methodology and provides the detailed step by step tutorial on the synthesis of the approximated MPC strategies with low-complexity representations suitable for application in the complex building control problems. The overall methodology is represented in Fig. 5.3 and can be compactly divided into three main parts: i, modelling part represented by first three blocks, ii, the 'teacher' MPC control strategy synthesis and evaluation represented by a fourth block, and iii, machine learning controller approximation and performance evaluation part represented by last three blocks in Fig. 5.3.

The accurate building model is a fundamental precondition for the success of the model based control strategy (Prívara et al. 2013). Therefore for this work, an existing house with six zones is modelled with high accuracy using the open-source Modelica library `IDEAS` (Baetens et al. 2015): a state-of-the-art building energy simulation program (see Section 4.3). In the next step, the Modelica non-linear building model is accurately linearized, and transformed into a linear time-invariant (LTI) state space model (SSM) (Picard et al. 2015). The obtained SSM is further used for both simulations as well as controller model, such that no plant-model mismatch is considered. This case is therefore the theoretical benchmark, with the aim to investigate the performance bound of the proposed control strategy. In the next step, the MPC is formulated, tuned and implemented on-line in receding horizon fashion in MATLAB® environment. The performance of the 'teacher' MPC is evaluated on simulations by calculating the thermal comfort and the energy use. The simulation data are collected and serve as a training dataset for a machine learning approximation. Further, for the sake of the dimensionality reduction and improved approximation accuracy, only the most significant variables are selected as features in the machine

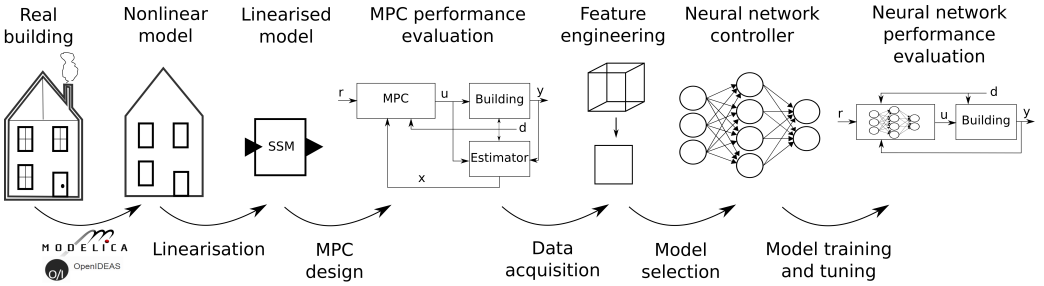


Figure 5.3: Schematic view of the methodology. From left to right: a real building is modelled using the BES Modelica library `IDEAS`. The obtained nonlinear model is then linearized and converted to a state space model (SSM) representation. The model is employed for both emulating the building dynamics in the simulations, as well as a controller model in the MPC. The MPC is being implemented and evaluated in MATLAB® environment in an on-line fashion. The simulation data are collected and reduced in dimensions by selecting only the most significant feature variables. Next, the machine learning model is selected and further trained and tuned to mimic the behaviour of the original 'teacher' MPC. Finally, the performance of the original and approximated MPC is being evaluated and compared with traditional controllers.

learning model via various feature engineering techniques. The problem to be solved in the next step is the multivariate regression problem of finding the model describing the relationship between multiple real-valued variables. For this task deep TDNNs are selected as universal function approximators (Hornik 1991). The reason behind TDNNs is that they are capable of handling complex multivariate time series regression problems which arise from the behaviour of the dynamical systems. Moreover, RT are also used for this task mainly because of their rule-based nature, making them suitable for easy implementation on today's building hardware. Next, the machine learning models are trained and tuned on the reduced data obtained from simulations of the 'teacher' control strategy, in our case a linear MPC with quadratic cost. Finally, the best performing machine learning models approximating the behaviour of the original 'teacher' MPC are evaluated on simulation scenarios. Additionally, the performances of the original and approximated MPC are compared altogether with a traditional RBC and a PID controller.

Remark 5.4.1 *In this thesis, a white box modelling approach was used in order to increase the accuracy of the case study. However, the methodology 5.4.1 is more general and suitable for any type of the modelling approach, either white-, black- or gray-box models. Similarly, the presented methodology is also not limited to a particular control strategy which serves as a 'teacher' for the machine learning model. On the contrary, any type of the advanced control strategies which require high computational resources can be approximated by this approach.* □

5.4.2 Machine Learning Problem Definition

From mathematical point of view, the parametric solution of the problem (5.8) has a form of a piecewise affine function defined over polyhedral regions, mapping the parametric space to the control inputs space, i.e. $f_{\text{MPC}} : \mathbb{R}^{n_\xi} \to \mathbb{R}^{n_\text{u}}$. However, this property holds only for the linear and quadratic objective functions, respectively, subject to the linear constraints. In the following sections, we show how to find explicitly defined approximations for the solutions of the MPC problems with an arbitrary type of the cost function and constraints by using multivariate regression.

MPC-like Regression-based Feedback Law

In a regression a set of m training data[1] $\{(\xi^{(1)}, u^{(1)}), \ldots, (\xi^{(m)}, u^{(m)})\}$ is given with $\xi^{(i)} \in \mathbb{R}^{n_\xi}$ and $u^{(i)} \in \mathbb{R}^{n_\text{u}}$. The objective is to devise a regression function $f_\Theta : \mathbb{R}^{n_\xi} \to \mathbb{R}^{n_\text{u}}$ which predicts the values of u (often called the *response* or *target variable*) that correspond to the measurements ξ (representing the *feature vector* in Machine Learning jargon) as accurately as possible.

The central idea here is to replace the implicitly defined feedback policy (5.9) by an explicit representation of the feedback law $u = f_\Theta(\xi)$, constructed by regression on a set of training data. The main advantage over the explicit MPC approach is that in regression we can control the complexity of $f_\Theta(\cdot)$ directly. In other words, we can devise the regression-based feedback strategy while considering limitations of the control hardware. Moreover, the regression approach is not limited on lower dimensional parametric space as it is in the case of the explicit MPC, which allows construction of the approximated explicit control laws also for complex problems with many parameters. The implied limitation of the approach is that the regression-based control policy is suboptimal w.r.t. the MPC cost function (5.8a). Therefore, our objective in this section is to devise the approximated explicit MPC control law, or regressor $f_\Theta(\cdot)$ that minimizes the deterioration of the control performance w.r.t. the criteria given in Section 5.1.

In general, we propose to construct the regression-based feedback policy as follows:

1. Fix the number of training data m and select the set $\{\xi^{(1)}, \ldots, \xi^{(m)}\}$ of initial values of parameters for the MPC problem (5.8).
2. For each $\xi^{(i)}$, obtain the corresponding optimal control move $u^{(i)}$ from (5.9) by solving (5.8).
3. Collect $\xi^{(i)}$ and $u^{(i)}$ into the training data set $\{(\xi^{(1)}, u^{(1)}), \ldots, (\xi^{(m)}, u^{(m)})\}$ and devise a regressor $f_\Theta(\cdot)$ that predicts the value of the control moves for an arbitrary vector of features ξ by $u = f_\Theta(\xi)$.

Data Generation

The selection of the set of initial values of the feature vector ξ as the training examples in the first step of the proposed procedure can be performed in two ways. The first option

[1] Here, $\xi^{(i)} \in \mathbb{R}^{n_\xi}$ denotes the i-th sample of a vector ξ.

is to grid the region of parameters of interest $\mathcal{P} \subseteq \mathbb{R}^{n_\xi}$ into $\xi^{(1)}, \ldots, \xi^{(m)}$, like presented in Coffey (2013). Providing representative samples $\xi^{(i)}$ for the whole range of parameters. Although, such an approach is only applicable if the dimension of the parametric space, i.e., n_ξ, is low. In particular, let n_g be the number of equidistantly-placed grid points for each element of the n_ξ-dimensional vector. Then the total number of generated points is $n_\mathrm{g}^{n_\xi}$. Please note that in the MPC problem 5.8, with the model from Section 4.3, $n_\xi = 1518$, and therefore the grid-based approach is far from practical reach in our case.
If n_ξ is large (what is usually the case in building climate control applications), an alternative way is to extract the pairs $(\xi^{(i)}, u^{(i)})$ from closed-loop profiles, as introduced in May-Ostendorp et al. (2011). Here, the building is controlled by an MPC strategy of 5.8 for a limited amount of time. While doing so, we record the values of $\xi(t)$ and the corresponding optimal control moves $u(t)$ at a fixed sampling rate T_s. The training dataset is then composed of the tuples $(\xi^{(i\Delta)}, u^{(i\Delta)})$ for $i = 0, \ldots, m$, where Δ is the collection period and m is the number of samples to be collected.

Remark 5.4.2 *In this thesis, the collection period Δ is considered to be equal to the sampling period T_s, for the sake of simplicity. In general the collection period can be an arbitrary positive integer multiple of the sampling period gives as $\Delta = \alpha T_s$, with $\alpha \in \mathbb{Z}_{>0}$. Here $\alpha > 1$ can be useful in order to reduce the computational burden in a case when big data with small sampling rate for a long time periods are available.* □

Practical Aspects

From a practical point of view to address the issue of large computational load induced by using the implicit MPC strategy, represented by (5.8), we propose to run the optimization on a remote machine. Here, a remote server takes over all computation for a limited amount of time. During this period the remote device communicates with the building over the Internet and collects the closed-loop data. An alternative approach is to artificially generate the data from simulations based on high-fidelity building models developed in BES software. After collecting enough closed-loop data, the regressor $f_\Theta(\cdot)$ is constructed on the remote computer and is subsequently uploaded to the local (simple) control hardware that resides directly in the building. From this moment, the control commands are generated by the regressor locally and the remote machine is no longer needed.
Moreover, we suggest monitoring the performance of the regressor periodically. This can be done by comparing, from time to time at a fixed or at a variable rate, the control commands generated locally by the regressor to the optimal control moves provided by solving (5.8) on a remote machine. If the mismatch exceeds some threshold, the remotely-running MPC can take over the control of the building for some period to generate new training data, followed by synthesis of a new regressor or retraining of the existing one.

Remark 5.4.3 *The proposed regression-based control policy is in no way tied to the particular MPC formulation in* (5.8)*. On the contrary, the procedure applies to training data generated by an arbitrary controller. As an example, the proposed approach can be used to extrapolate properties of controllers based on stochastic MPC formulations, such as those*

Algorithm	Function	Multivariate	Parametric	Memory	Dimensions
GLM	linear	yes	yes	no	high
NLR	nonlinear	no	yes	no	high
RT	PWC	no	no	no	high
SVM	nonlinear	no	no	no	moderate
MLP	nonlinear	yes	no	no	high
TDNN	nonlinear	yes	no	yes	high
VAR	linear	yes	yes	yes	high

Table 5.2: Comparison of the different regression algorithms.

proposed by Oldewurtel et al. (2010) and Ma et al. (2012a), or when nonlinear PMV-based thermal comfort criteria are included into (5.8) *per Cigler et al. (2012), Freire et al. (2005).* □

5.4.3 Regression Algorithm Selection

In general, the problems naturally arising in the building control applications are those with multiple controlled and manipulated variables as opposed to single input control problems. Therefore the task of finding the approximate control law $u = f_\Theta(\xi)$ belongs to the class of multivariate regression problem, for which an objective function is given by (5.28).

$$\min_{\Theta} \sum_{j=1}^{n_u} \sum_{i=1}^{m} \left(f_\Theta(\xi^{(i)})_j - u_j^{(i)} \right)^2 \tag{5.28}$$

The objective here is to minimize over the Θ the square of the difference between the pre-computed control inputs $u_j^{(i)}$ (obtained from the (5.9)) and regressor functions $f_\Theta(\xi^{(i)})_j$ w.r.t. the given features $\xi^{(i)}$ parametrized by Θ. The superscript i denotes the i-th sample of the training data, and subscript j stands for the j-th target variable, representing manipulated variable in the control context.

Table 5.2 summarizes the comparison of the seven most commonly used regression algorithms nowadays: generalized linear models (GLM), nonlinear regression (NLR), regression trees (RT), support vector machines (SVM), multilayer perceptron (MLP) neural network architecture, time delay neural networks (TDNN) and vector autoregressive model (VAR). The comparison is based on following properties. The nature of the regressor function, e.g. linear, nonlinear and piecewise constant (PWC). The ability to handle problems with more than a single target variable. The division between parametric models where the model structure needs to be selected a priori before learning and non-parametric models where the model structure is not chosen beforehand, but it is being learned instead. The memory property which says about the ability of the model to handle the time series problems. And finally the ability of the compared models to handle high dimensional learning datasets with many samples. For the MPC problem formulations resulting in linear or quadratic optimization problem, the feedback law takes a form of a piecewise affine (PWA) function

defined over a polyhedral domain of the state space. However, this property no longer holds for the MPC formulations when nonlinearity is present in the objective functions or constraints. Therefore for the sake of the versatility and flexibility of the developed methodology, we opted for the regression models with nonlinear nature of the fit. Further, we are forced to consider only the models which can handle multivariate nature of the problem (5.28). The non-parametric models were chosen instead of the parametric ones because to set up the model structure before the learning process for the problems of this complexity is a very tricky task. Moreover, the abundance of the training data supports the notion of the learning the structure of the model as well as the model parameters. The ability of the model to capture the dynamics of the data is also desired property because the parameters of the MPC problem (5.8) are naturally time-dependent. And finally, we want to choose the regression model which can also handle large high dimensional datasets, generated by the precomputed MPC solutions. After considering all these aspects, the natural choice for this type of problems is to use TDNN as the regression model. However, also the RT are chosen for investigation due to the human readable nature of the regressor and the ability to handle time series data after appropriate feature engineering procedures.

5.4.4 Regression Trees

This section is based on the previous work of the author as given in Klaučo et al. (2014). We are interested in finding an optimal regressor $f_\Theta(\cdot)$ which partitions the training data into M cells, denoted by $\mathcal{P}_1, \ldots, \mathcal{P}_M$, such that $f_\Theta(\cdot)$ is well posed in the sense of the following definition:

Definition 5.4.4 *The regressor $f_\Theta(\cdot)$ is well posed and optimal if:*

R1: $f_\Theta(\cdot)$ is a single-valued function, i.e., $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$ for all $i \neq j$;

R2: given a set $\mathcal{P}$ such that $x_j \in \mathcal{P}\ \forall j$, the domain of the regressor is equal to $\mathcal{P}$, i.e., $\mathcal{P} = \bigcup_i \mathcal{P}_i$.

R3: for all $\xi \in \mathcal{P}_i$ we have that $u = f_{reg,i}(\xi)$ where $f_{reg,i} : \mathbb{R}^{n_\xi} \to \mathbb{R}^{n_u}$ is the optimal local *regressor in the cell $\mathcal{P}_i$, i.e., $f_{reg,i}$ and $\mathcal{P}_i$ minimize the point-wise regression error*

$$\min_{f_{reg,i}, \mathcal{P}_i} \sum_{\xi_j \in \mathcal{P}_i} \|u_j - f_{reg,i}(\xi_j)\|. \tag{5.29}$$

□

Note that R1 requires that $f_\Theta(\cdot)$ provides a unique prediction of the response variable for each particular vector of parameters ξ. If $f_\Theta(\cdot)$ is a continuous function, then R1 can be relaxed to $\text{int}(\mathcal{P}_i) \cap \text{int}(\mathcal{P}_j) = \emptyset$, where $\text{int}(\cdot)$ is the interior of the corresponding set. Then, R2 forces the regressor to be defined for all ξ from a region of interest $\mathcal{P}$. Finally, R3 provides optimal regression of the training data. Note that optimal selection of local regressors $f_{\text{reg},i}$ depends on the selection of the corresponding cell $\mathcal{P}_i$ and vice versa.

Once the optimal cells $\mathcal{P}_i$ and the associated local regression functions $f_{\text{reg},i}(\cdot)$ are constructed, the overall regression policy is given by

$$u = f_\Theta(\xi) = \begin{cases} f_{\text{reg},1}(\xi) & \text{if } \xi \in \mathcal{P}_1, \\ \vdots & \\ f_{\text{reg},M}(\xi) & \text{if } \xi \in \mathcal{P}_M. \end{cases} \tag{5.30}$$

Construction of Regression Trees

A popular approach for construction of the regressor functions $f_\Theta(\cdot)$ is by employing binary regression trees (Breiman 1993). Such trees consist of a finite number of nodes, each of which may contain pointers to two child nodes. Nodes without any children are called *leaf nodes*. Each leaf node contains a local expression of the regressor. All non-leaf nodes, on the other hand, contain an expression of a splitting function $\sigma : \mathbb{R}^{n_\xi} \to \mathbb{R}$ and pointers to a maximum of two child nodes. The *left* child node is visited if $\sigma(\xi) \leq 0$, while the *right* node is explored if $\sigma(\xi) > 0$.

The regression tree can be constructed by a recursive procedure, summarized as Algorithm 2. In particular, in Step 2 of Alg. 2 we need to determine the optimal splitting function $\sigma : \mathbb{R}^{n_\xi} \to \mathbb{R}$, along with optimal local regressors $f_{\text{L}} : \mathbb{R}^{n_\xi} \to \mathbb{R}^{n_{\text{u}}}$ and $f_{\text{R}} : \mathbb{R}^{n_\xi} \to \mathbb{R}^{n_{\text{u}}}$ that solve the following optimization problem:

$$\min_{\sigma, f_{\text{L}}, f_{\text{R}}} \left(\sum_{\xi^{(i)} \in \mathcal{P}_{\text{L}}} \|u^{(i)} - f_{\text{L}}(\xi^{(i)})\| + \sum_{\xi^{(j)} \in \mathcal{P}_{\text{R}}} \|u^{(j)} - f_{\text{R}}(\xi^{(j)})\| \right), \tag{5.31}$$

where

$$\mathcal{P}_{\text{L}} = \{\xi : \ \sigma(\xi) \leq 0\}, \ \mathcal{P}_{\text{R}} = \{\xi : \ \sigma(\xi) > 0\}, \tag{5.32}$$

are the cells generated by the split $\sigma(\cdot)$. Note that, since $\mathcal{P}_{\text{L}}$ and $\mathcal{P}_{\text{R}}$ depend on σ, problem (5.31) is nonlinear in the decision variables.

Once the optimal split and the optimal local regressors are computed in Step 2, we need to determine whether the currently explored node needs to be subdivided. This decision is based on two criteria: the number of points in each of the split cells, and the regression error in each cell. The former is computed in Step 3 and the latter is evaluated in Step 5. If the number of points (card$(\cdot)$) in the left cell (cf. (5.32)) drops below a pre-defined threshold $m_{\min}$, or when the local regression error is smaller than $e_{\min}$, exploration of the left cell is terminated and a leaf node containing the corresponding local regressor $f_{\text{L}}(\cdot)$ is returned in Step 7. Otherwise, the left cell is explored recursively in Step 9. The right cell is treated similarly in Steps 11−15. Finally, the node, which consists of the split $\sigma(\cdot)$ and the pointers to child nodes $\mathcal{N}_{\text{L}}$, $\mathcal{N}_{\text{R}}$, is returned in Step 16.

Remark 5.4.5 *Alternatively, the stopping criteria in Steps 6 and 11 can be modified to stop the recursion when a maximal tree depth D_{max} is reached. Such a criterion gives the designer a direct control over size of the tree, and hence over complexity of the implementation in the control hardware.* □

Algorithm 2 Construction of the binary regression tree.

function TREENODE($\{\xi^{(1)}, \dots, \xi^{(m)}\}$, $\{u^{(1)}, \dots, u^{(m)}\}$)
  Compute optimal splitting function $\sigma(\cdot)$ and optimal local regressors $f_{\mathrm{L}}(\cdot)$ and $f_{\mathrm{R}}(\cdot)$ from (5.31).
  Split the set $\{1, \dots, m\}$ into subsets $\mathcal{L} = \{i : \sigma(\xi^{(i)}) \leq 0\}$ and $\mathcal{R} = \{i : \sigma(\xi^{(i)}) > 0\}$.
  Denote $(\xi_{\mathrm{L}}, u_{\mathrm{L}}) = \{(\xi^{(i)}, u^{(i)}) : i \in \mathcal{L}\}$ and $(\xi_{\mathrm{R}}, u_{\mathrm{R}}) = \{(\xi^{(i)}, u^{(i)}) : i \in \mathcal{R}\}$.
  Evaluate the regression errors

$$e_{\mathrm{L}} = \sum_{i \in \mathcal{L}} \|u^{(i)} - f_{\mathrm{L}}(\xi^{(i)})\|, \; e_{\mathrm{R}} = \sum_{j \in \mathcal{R}} \|u^{(j)} - f_{\mathrm{R}}(\xi^{(j)})\|. \tag{5.33}$$

  if $\mathrm{card}(\xi_{\mathrm{L}}) < m_{\min}$ **or** $e_{\mathrm{L}} < e_{\min}$ **then** ▷ left node stopping criterion
    $\mathcal{N}_{\mathrm{L}} = \text{LEAFNODE}(f_{\mathrm{L}})$.
  else
    $\mathcal{N}_{\mathrm{L}} = \text{TREENODE}(\xi_{\mathrm{L}}, u_{\mathrm{L}})$.
  end if
  if $\mathrm{card}(\xi_{\mathrm{R}}) < m_{\min}$ **or** $e_{\mathrm{R}} < e_{\min}$ **then** ▷ right node stopping criterion
    $\mathcal{N}_{\mathrm{R}} = \text{LEAFNODE}(f_{\mathrm{R}})$.
  else
    $\mathcal{N}_{\mathrm{R}} = \text{TREENODE}(\xi_{\mathrm{R}}, u_{\mathrm{R}})$.
  end if
return $\mathcal{N} = \{\sigma, \mathcal{N}_{\mathrm{L}}, \mathcal{N}_{\mathrm{R}}\}$. ▷ obtain the regression tree
end function

The difficulty of constructing an optimal regression tree stems from the nonlinearity of the optimization problem in (5.31)−(5.32) for general types of the split function $\sigma(\cdot)$ and of the local regressors $f_{\mathrm{L}}(\cdot)$, $f_{\mathrm{R}}(\cdot)$. To simplify the computation, standard regression tree approaches restrict splits to orthogonal hyperplanes of the form $\sigma = \alpha^T \xi + \beta$, where the optimal vector α is selected from the finite set $\{[1, 0, \dots, 0]^T, [0, 1, 0, \dots, 0]^T, \dots, [0, \dots, 0, 1]^T\}$. Moreover, the local regressors are typically assumed to be constant functions, i.e., $f_{\mathrm{L}}(\xi) = g_{\mathrm{L}}$ and $f_{\mathrm{R}}(\xi) = g_{\mathrm{R}}$. Then Algorithm 2 generates a binary tree that encodes a piecewise constant (PWC) regressor $f_\Theta(\cdot)$ where each cell $\mathcal{P}_i$ is an axis-aligned hyperbox. However, such simplifications set back the quality of the regression. As a consequence, a high number of nodes is typically required to achieve the desired regression error. Moreover, the RTs are only single variate regression models, and due to this fact, a standalone RT needs to be constructed for each zone separately, what increases the complexity of the final controller and also causes some loss of the performance by not taking into account the coupling between the controlled outputs.

Optimal Node Splitting with Affine Splits and Affine Regressors

In this section, theoretical contributions towards regression trees are introduced. The standard regression trees presented in Section 5.4.4 are limited only to the splitting functions that are orthogonal hyperplanes, and the local regressors inside each cell are assumed to

be constant. This section shows how to split the nodes optimally by general hyperplanes. Simultaneously, the local constant regressors are replaced by affine expressions that provide a better approximation of the training data. As a consequence, such a regression-based feedback policy can be described as a PWA function defined over a polyhedral domain, akin to explicit MPC solutions (see Section 3.2.3). The added benefit is that the regression function is directly encoded as a binary tree, which offers a fast implementation even on simple hardware.
First we show how to obtain optimal expressions of affine functions

$$\sigma(\xi) := \alpha^T \xi - \beta, \tag{5.34a}$$
$$f_{\mathrm{L}}(\xi) := F_{\mathrm{L}}\xi + g_{\mathrm{L}}, \tag{5.34b}$$
$$f_{\mathrm{R}}(\xi) := F_{\mathrm{R}}\xi + g_{\mathrm{R}} \tag{5.34c}$$

from (5.31)−(5.32) by solving a mixed-integer quadratic program (MIQP). Here, $\alpha \in \mathbb{R}^{n_\xi}$, $\beta \in \mathbb{R}$, $F_{\mathrm{L}} \in \mathbb{R}^{n_{\mathrm{u}}} \times n_\xi$, $g_{\mathrm{L}} \in \mathbb{R}^{n_{\mathrm{u}}}$, $F_{\mathrm{R}} \in \mathbb{R}^{n_{\mathrm{u}} \times n_\xi}$, $g_{\mathrm{R}} \in \mathbb{R}^{n_{\mathrm{u}}}$.

Theorem 5.4.6 *Given are m datapoints $\{(\xi_1, u_1), \ldots, (\xi_m, u_m)\}$. The optimal split $\sigma(\cdot)$ and optimal local regressors $f_L(\cdot)$, $f_R(\cdot)$ as in (5.34) that solve (5.31)−(5.32) are given as the optimal solution to the mixed-integer quadratic program (MIQP)*

$$\min \sum_{i=1}^{m} (u_i - z_i)^T (u_i - z_i) \tag{5.35a}$$
$$s.t. \; -M(1-\delta_i) \le z_i - (F_L\xi_i + g_L) \le M(1-\delta_i), \tag{5.35b}$$
$$-M\delta_i \le z_i - (F_R\xi_i + g_R) \le M\delta_i, \tag{5.35c}$$
$$\alpha^T\xi_i \le \beta + M(1-\delta_i), \tag{5.35d}$$
$$\alpha^T\xi \ge \beta + \epsilon - M\delta_i, \tag{5.35e}$$
$$\|\alpha\|_\infty = 1, \tag{5.35f}$$

where the minimization is performed over continuous decision variables α, β, F_L, g_L, F_R, g_R, $z_i \in \mathbb{R}^{n_u}$, and the binary variables $\delta_i \in \{0,1\}$, $i \in \mathbb{N}_1^m$. Note that constraints in (5.35b)−(5.35e) are enforced for $i \in \mathbb{N}_1^m$ with a small positive numerical tolerance ϵ and a sufficiently large constant M. ■

Proof. Consider a fixed index i and assume that $\delta_i = 1$ is the optimal decision. First we show that such a decision implies that $z_i = F_{\mathrm{L}}\xi_i + g_{\mathrm{L}}$ and that α, β satisfy $\alpha^T\xi_i \le \beta_i$, i.e., ξ_i is associated to the "left" (i.e., non-positive) side of the split. The latter follows directly from (5.35d) which, for $\delta_i = 1$, reduces to $\alpha^T\xi_i \le \beta$. Similarly, $z_i = F_{\mathrm{L}}\xi_i + g_{\mathrm{L}}$ follows directly from (5.35b) for $\delta_i = 1$. Moreover, notice that (5.35c) and (5.35e) are inactive for $\delta_i = 1$ since M is assumed to be sufficiently large. Next, assume $\delta_i = 0$ is the optimal solution in (5.35). Then $z_i = F_{\mathrm{R}}\xi_i + g_{\mathrm{R}}$ by (5.35c) and $\alpha^T\xi_i > \beta_i$ from (5.35e) since $\epsilon > 0$ is assumed. Note that (5.35b) and (5.35d) are both inactive when $\delta_i = 0$. Finally, optimality of the split and of the associated fits follows from minimizing the fitting error in (5.35a),

which is equivalent to (5.31) since $z_i = f_{\mathrm{L}}(\xi_i)$ if $\delta_i = 1$ and $z_i = f_{\mathrm{R}}(\xi_i)$ if $\delta_i = 0$. Note that (5.35a) yields the same optimizer as minimization of $\sum \|u_i - z_i\|$.
Although MIQPs are still nonlinear due to presence of binary variables δ_i, they can be solved efficiently with state-of-the-art solvers, such as with GUROBI (Gurobi Optimization 2012).

Remark 5.4.7 *The MIQP* (5.34) *is related to construction of so-called* hinging hyperplane *(HH) regressors that are frequently used to identify PWARX models of dynamical systems. Compared to the HH formulation of Roll et al. (2004), problem* (5.34) *has fewer binary optimization variables and is thus easier to solve as the number of datapoints increases. An another advantage is that construction of discontinuous regressors does not require additional binary/continuous variables as in the case of HH formulations.* □

Remark 5.4.8 *The reason for constraint* (5.35f) *is to rule out the trivial solution* $\alpha = \beta = 0$ *when all points are allocated only to one side of the split. Note that* (5.35f) *normalizes the largest value in* α *to* ± 1. □

Remark 5.4.9 *If* M *is chosen sufficiently large, problem* (5.35) *is always feasible. Rules of selecting a suitable constant* M *are discussed e.g. in Bemporad and Morari (1999). Note that finding a suitable* M *requires the decision variables of* (5.35) *to be bounded.* □

Employing (5.35) in Step (5.31) of Algorithm 2 generates a node $\mathcal{N}$ that consists of an affine split $\sigma(\cdot)$. Such a split divides the space of independent variables ξ into polyhedra $\mathcal{P}_{\mathrm{L}}$, $\mathcal{P}_{\mathrm{R}}$ per (5.32). As our next result we provide a formal proof that the tree generated by recursive application of Algorithm 2 to the set of training data $\{(\xi_1, u_1), \ldots, (\xi_m, u_m)\}$ encodes a piecewise affine regressor $f_\Theta(\cdot)$ as in (5.30) which is defined over polyhedra $\mathcal{P}_1, \ldots, \mathcal{P}_M$ that satisfy R1 and R2 of Definition 5.4.4 (note that R3 is achieved by (5.35a)). To do so, we first prove an intermediate statement.

Theorem 5.4.10 *Let the polyhedron* $\mathcal{P} \subseteq \mathbb{R}^{n_\xi}$ *and the split* $\sigma(\cdot)$ *of* (5.34a) *be given. Denote by*

$$\mathcal{P}_L = \mathcal{P} \cap \{\xi : \ \alpha^T \xi \leq \beta\}, \ \mathcal{P}_R = \mathcal{P} \cap \{\xi : \ \alpha^T \xi > \beta\}, \tag{5.36}$$

the sets that originate from splitting $\mathcal{P}$ *by* $\sigma(\cdot)$. *Then:*

1. $\mathcal{P}_L$ *and* $\mathcal{P}_R$ *are polyhedra,*
2. $\mathcal{P}_L \cap \mathcal{P}_R = \emptyset$,
3. $\mathcal{P}_L \cup \mathcal{P}_R = \mathcal{P}$.

■

Proof. Notice that the set $\mathcal{S}_{\leq} = \{\xi : \ \alpha^T \xi \leq \beta\}$ is a closed polyhedron and the set $\mathcal{S}_{>} = \{\xi : \ \alpha^T \xi > \beta\}$ is an open polyhedron. Then the first statement follows directly from the fact that intersection of two polyhedra yields a polyhedron, see, e.g. Ziegler (1994). The second statement follows from the fact that intersection is an associative operation. Therefore $\mathcal{P}_{\mathrm{L}} \cap \mathcal{P}_{\mathrm{R}} = (\mathcal{P} \cap \mathcal{S}_{\leq}) \cap (\mathcal{P} \cap \mathcal{S}_{>}) = (\mathcal{P} \cap \mathcal{P}) \cap (\mathcal{S}_{\leq} \cap \mathcal{S}_{>})$. Clearly, $\mathcal{P} \cap \mathcal{P} = \mathcal{P}$

and $\mathcal{S}_{\leq} \cap \mathcal{S}_{>} = \emptyset$. Since $\mathcal{P} \cap \emptyset = \emptyset$, the result follows. For the third statement, notice that union is also an associative operation. Hence $\mathcal{P}_{\mathrm{L}} \cup \mathcal{P}_{\mathrm{R}} = (\mathcal{P} \cap \mathcal{S}_{\leq}) \cup (\mathcal{P} \cap \mathcal{S}_{>}) = (\mathcal{P} \cap \mathcal{P}) \cup (\mathcal{S}_{\leq} \cap \mathcal{S}_{>}) = \mathcal{P} \cup \emptyset = \mathcal{P}$.

Theorem 5.4.11 *Let $\mathcal{T}$ be a tree generated by Algorithm 2. Denote by T the set of leaf nodes of $\mathcal{T}$, and let $\mathcal{P}_{T_i}$, $i = 1, \ldots, \mathrm{card}(T)$ be the region of validity of the i-th leaf node. Assume that the polyhedron $\mathcal{P}$ is given such that $\xi_i \in \mathcal{P}$ for all ξ_i from the training data. Then*

$$\bigcup_i \mathcal{P}_{T_i} = \mathcal{P}, \tag{5.37a}$$

$$\mathcal{P}_{T_i} \cap \mathcal{P}_{T_j} = \emptyset, \ \forall i, j \in T, \ i \neq j. \tag{5.37b}$$

■

Proof. Consider the root node of $\mathcal{T}$ $\sigma(\cdot)$, which splits $\mathcal{P}$ into $\mathcal{P}_{\mathrm{L}}$, $\mathcal{P}_{\mathrm{R}}$ per (5.36). By Theorem 5.4.10, $\mathcal{P}_{\mathrm{L}} \cup \mathcal{P}_{\mathrm{R}} = \mathcal{P}$ and $\mathcal{P}_{\mathrm{L}} \cap \mathcal{P}_{\mathrm{R}} = \emptyset$. At the second level of the tree, $\mathcal{P}_{\mathrm{L}}$ is split by $\sigma_{\mathrm{L}}(\cdot)$ into $\mathcal{P}_{\mathrm{LL}}$ and $\mathcal{P}_{\mathrm{LR}}$. Similarly, $\mathcal{P}_{\mathrm{R}}$ from the first level is split into $\mathcal{P}_{\mathrm{RL}}$ and $\mathcal{P}_{\mathrm{RR}}$ by $\sigma_{\mathrm{R}}(\cdot)$. Again, by Theorem 5.4.10, we have that $\mathcal{P}_{\mathrm{LL}} \cup \mathcal{P}_{\mathrm{LR}} = \mathcal{P}_{\mathrm{L}}$, $\mathcal{P}_{\mathrm{RL}} \cup \mathcal{P}_{\mathrm{RR}} = \mathcal{P}_{\mathrm{R}}$, and $\mathcal{P}_{\mathrm{LL}} \cap \mathcal{P}_{\mathrm{LR}} = \emptyset$, $\mathcal{P}_{\mathrm{RL}} \cap \mathcal{P}_{\mathrm{RR}} = \emptyset$. Therefore $\mathcal{P}_{\mathrm{LL}} \cup \mathcal{P}_{\mathrm{LR}} \cup \mathcal{P}_{\mathrm{RL}} \cup \mathcal{P}_{\mathrm{RR}} = \mathcal{P}$ and $\mathcal{P}_{\mathrm{LL}} \cap \mathcal{P}_{\mathrm{LR}} \cap \mathcal{P}_{\mathrm{RL}} \cap \mathcal{P}_{\mathrm{RR}} = \emptyset$ at the second level of the tree. Then (5.37) follows by induction.

Algorithm 2 can be easily adapted to return the corresponding regions of validity $\mathcal{P}_i$ of the i-th terminal node and thus to obtain an explicit representation of the regressor $f_\Theta(\cdot)$ as in (5.30). Then $f_\Theta(\xi)$ can be evaluated by searching sequentially through $\mathcal{P}_1, \ldots, \mathcal{P}_M$, stopping once $\xi \in \mathcal{P}_i$. However, a more efficient way to evaluate $f_\Theta(\cdot)$ is to directly traverse the binary tree starting from its root node. Here, the associate splitting function is evaluated and, based on its value,either the left or the right branch is explored. Such a procedure is repeated at each subsequent child node until a leaf node is encountered, whereupon the evaluation is stopped and the predicted value $u = f_{\mathrm{reg},i}(\xi)$ is returned. It is well known that the computational effort needed to evaluate $f_\Theta(\xi)$ via a binary tree is $\mathcal{O}(\log_2 M)$, where M is the number of leaf nodes of the tree. The total memory storage is $\mathcal{O}(M)$. The data that need to be stored are the splits associated to each non-leaf node, pointers to child nodes, and local regressors in each leaf node.

5.4.5 Deep Time Delay Neural Networks

It has been shown that a neural network (NN) with a sufficient number of hidden units can approximate arbitrary continuous function defined on a closed and bounded set (Cybenko 1989, Hornik 1991). Because of this, the NNs are popularly used as general nonlinear regression models. NNs are mathematical models inspired by the human brain, defined by the interconnections (synapses) of the individual neurons in the successive layers. This can be visualized as a directed graph. When no cycles are present in the graph the network is called feedforward; otherwise, it is called recurrent.

Mathematically, NN is a function $f(\cdot)$ composed of weighted sums (synapses) of bounded monotone functions $g(\cdot)$, also called neuron activation functions, which are again composite of the activation functions from the previous layers of the net, all the way down to the input layer, which consist of the features ξ. The compact representation of the general multilayer feedforward NN, also called multilayer perceptron (MLP) is given by the set of equations (5.38).

$$f(\xi)_i = g_{\mathrm{M},i}\left(z_{\mathrm{M}}\right) \tag{5.38a}$$

$$z_{\mathrm{M}} = B_{\mathrm{M}} + \sum_{j=1}^{n_{\mathrm{M}}} W_{\mathrm{M},j} g_{\mathrm{M}-1,j}\left(z_{\text{M-1}}\right) \tag{5.38b}$$

$$z_{\mathrm{M}-1} = B_{\mathrm{M}-1} + \sum_{k=1}^{n_{\mathrm{M}-1}} W_{\mathrm{M}-1,k} g_{\mathrm{M}-2,k}\left(z_{\text{M-2}}\right) \tag{5.38c}$$

$$\vdots$$

$$z_1 = B_1 + \sum_{l=1}^{n_\xi} W_{1,l}\xi_l \tag{5.38d}$$

Here, the function $f(\cdot)_i$ represents the i-th output of the net, $g_{\mathrm{M},i}(\cdot)$ is the activation function of the i-th neuron, while z_{M} is the total weighted sum of the inputs in the M-th layer (i.e. the outputs from the (M$-$1)-th layer). ξ_l stands for the l-th feature. The number of the units (neurons) in the the M-th layer is given by n_M. $W_{\mathrm{M},j}$ denotes the weight of the output of the j-th neuron in the M-th layer and B_{M} is the bias term for the M-th layer. Te weights and biases are grouped into a single vector Θ, regarded as a vector of optimized variables in the optimization problem, which needs to be solved in order to train the NN.

The individual types of the neural networks are being differentiated by the different setup of the three parameters: first, the structure of the neural network imposed by the interconnections of the neurons, second, the activation function type which determines the output of the individual neurons and third, the weights of the interconnections. The first two parameters are to be chosen prior the learning process, while the weights are updated during the learning. In our case, we opted for the deep time delay neural network (TDNN) with sigmoid activation functions. The reason behind the TDNNs is that they are capable of handling the time series modelling problems which are characterized by the dynamically dependent data points in time, as it is in the case of the data generated by the control problems. The only difference between TDNN and MLP lies in the input layer, where the dynamic is captured by the set of delayed time signals $\{\xi^{(t)}, \xi^{(t-1)}, \ldots, \xi^{(t-N)}\}$ derived from the original features ξ. The reason for using a deep architecture is a substantially better performance in comparison with the shallow nets, as showed by Hinton et al. (2006) and Busseti et al. (2012). The compact schematic representation of the TDNN with deep architecture, consistent with the equations (5.38) is shown in Fig. 5.4.

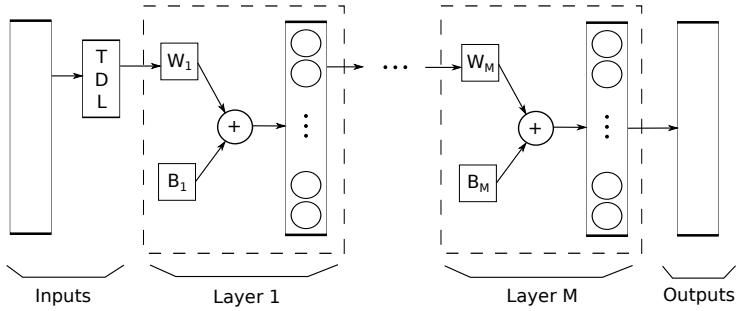


Figure 5.4: Schematic representation of the time delay neural network with deep architecture. Here, TDL stands for the time delay operator, W_{M} and B_{M} are weights and biases for the M-th layer, respectively.

5.4.6 Feature Engineering

Feature Engineering (FE) is a process of selection, creation, or learning of the features used in the machine learning model. Due to the strong influence of the features on the performance of the predictive model, it is considered to be a fundamental part of almost any practical machine learning application (Heaton 2017). The gains from using only the most relevant features are threefold: first, improved performance, second, reduced complexity, and third, improved interpretability of the developed models. This is, however, typically where most of the effort in a machine learning project goes, learning is often the quickest part (Domingos 2012). Engineering features properly is primarily a manual, difficult and time-consuming task, mostly because it is domain-specific, while learners can be largely general-purpose. Additionally, each type of the model will respond differently to different types of engineered features (Heaton 2017). Therefore, one of the holy grails of machine learning is to automate more of the feature engineering process (Domingos 2012). For these reasons a substantial research interest has been given in the recent years into the development of the feature learning algorithms (Hinton and Salakhutdinov 2006, Le et al. 2012), or advanced semi-automated feature engineering systems (Anderson et al. 2013).
Even though the feature creation, transformation and learning are powerful tools for obtaining new, well-performing features, the main drawback here, is that the physical meaning of the original features is lost during this process. Therefore, these methods are not suitable for the problem of interest as given in 5.4.2. We are primarily interested only in the selection of the most relevant features $\tilde{\xi}$ as a subset of the original feature vector ξ, i.e. $\mathbb{R}^{n_{\tilde{\xi}}} \subset \mathbb{R}^{n_\xi}$, without loss of their physical meaning. Several feature selection (FS) methods suitable for building energy models are available in the literature nowadays. In Magoules and Zhao (2016) authors presented a heuristic approach tailored for support vector machine (SVM) models, while in Dodier and Henze (2004) a statistical Wald's test has been used for identification and of irrelevant features for neural network (NN) models. In overall there are three categories of FS algorithms: filter, wrapper and embedded methods. However, the finding of all relevant features is, in general, a NP-hard problem (Guyon and Elisseeff 2003).

In following, we present a simple and systematic approach for efficient FS for predictive models in the scope of the building climate control applications. The method presented here is versatile and can be used for identification and selection of the most relevant variables in the dynamical model of the building, either for controller approximation, reducing the complexity of the model, or for reduction of the cost of the sensory equipment in the real building. The presented method is composed of three independent steps, first the manual selection and elimination of linearly dependent features, second, a principal component analysis (PCA) based feature selection, and third, selection of the disturbance features based on the model dynamics.

Manual Selection and Elimination of Linearly Dependent Features

The most straightforward is the manual FS, which is done by exploiting the engineering knowledge of the system. In contrast with the standard MPC scheme from Fig. 5.1, the machine learning controller from the closed-loop system representation as given in Fig. 5.5 is approximating and replacing not only behaviour of the MPC but also the state observer. In our case, this setup reduces the parametric space ξ by discarding the 286 state variables

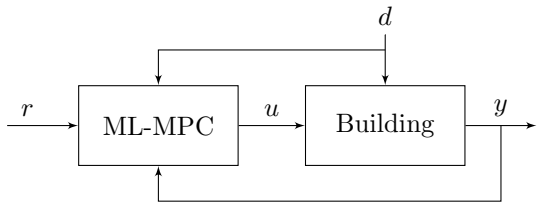


Figure 5.5: Schematic representation of the closed-loop system with machine learning controller mimicking the behavior of the MPC and state observer. Here, d are measured disturbances, y denotes the outputs, u are the control actions, r is are reference signals covering lower and upper comfort bounds, respectively.

x and manually replacing them by the six output variables y as a part of the reduced feature vector $\tilde{\xi}$. Further, all linearly dependent features can be discarded, because they do not carry the additional information for the machine learning model (Das 1971). In our case, we consider the same comfort boundaries for all zones, and therefore the 12 original variables are reduced to a single variable. Similarly, by the elimination of the linearly dependent variables from the vector d, we cut 44 original disturbances only to 26 features.

Principal Component Analysis Feature Selection

Principal component analysis (PCA) is a well-known multivariate statistical technique used mainly for the dimensionality reduction of the high-dimensional datasets. It is an eigenvector-based method, which reveals the internal structure of the data by extracting the relevant information by representing it as a set of the new uncorrelated orthogonal variables, also called principal components. The principal components are in fact a linear combination of the original variables constructed in a way to have the largest possible variance. The first

principal component has the largest variance, and each following component is computed under the constraint to be orthogonal to the previous component and to have the largest possible variance. Mathematically, PCA depends upon the singular value decomposition (SVD) of a data matrix (rectangular matrices), or upon the eigenvalue decomposition of a covariance matrix (positive semi-definite matrices) (Abdi and Williams 2010).
Because FS is essentially a dimensionality reduction problem several techniques based on PCA have been developed, see, e.g. Jolliffe (1986), Krzanowski (1987), Lu et al. (2007). In this work, we choose a simple and computationally efficient adaptation of the method from Song et al. (2010). The procedure is defined by the following steps:

1. Compute the covariance matrix of the feature vector ξ, given as: $\Sigma = \frac{1}{m}\xi^T\xi$.

2. Perform SVD of Σ, to obtain: $\Sigma = USV^T$, where U are the principal component coefficients, and S are the principal component variances, i.e. eigenvalues of the Σ.

3. Compute the percentage of the total variance captured by i-th principal component, given as: $v_i = \frac{S_{i,i}}{tr(S)}$.[2]

4. Define the precision threshold η value for the retained variability of the data and select only the q most significant principal components which total accumulative variance is within the given threshold: $\max q$, s.t. $\sum_{i=1}^{q} v_i \leq \eta$.

5. Compute the normalized contribution ν_j of the j-th feature ξ_j on selected principal components from step 4. by summation of the absolute values of the coefficients of the first q columns of the matrix U: $\nu_j = \frac{\sum |U_{j;1,\dots,q}|}{\max\limits_{1\leq k\leq n_\xi} (\sum_{i=1}^{q} |U_{k;i}|)}$.

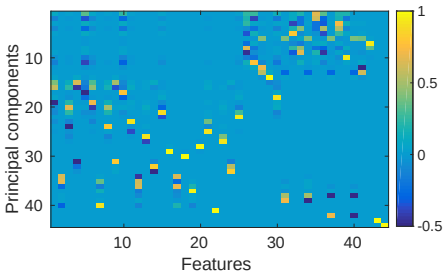
6. Define the threshold ψ value for the minimal contribution of the features on the principal coefficients and select the p most important features by satisfying: $\nu_j \geq \psi,\ \forall j \in \mathbb{N}_0^{n_\xi}$.

The method is demonstrated on the disturbance vector d with two months data derived from six-zone building model from Section 4.3. The results for the parameter values $\eta = 99.9\%$, and $\psi = 99.0\%$ are shown in Fig. 5.6, where Fig. 5.6(a) illustrates the PCA coefficient matrix U (Step 2.), Fig. 5.6(b) shows the total variance percentage of the individual principal components (Step 3.), Fig. 5.6(c) illustrates only the 9 most significant PCA coefficients of the U matrix (Step 4.), and Fig. 5.6(d) shows the normalized contribution of the individual features ξ_j on selected most significant PCA coefficients (Step 5.). By this method, 22 out of 44 disturbances are selected.

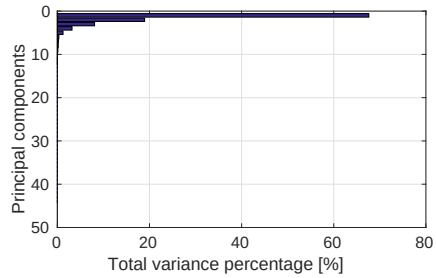
Feature Selection Based on Model Disturbances Dynamics

The idea behind the feature selection via the analysis of the disturbances dynamics is simple, based on the investigation of the disturbance matrix E from the LTI model 4.12. The

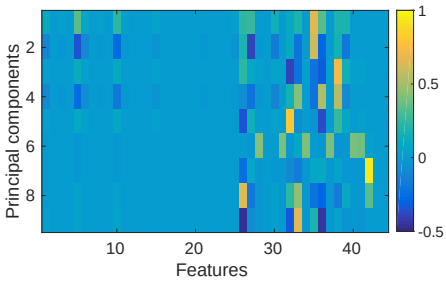
[2]Here, $tr(S)$ denotes the trace of the matrix S.



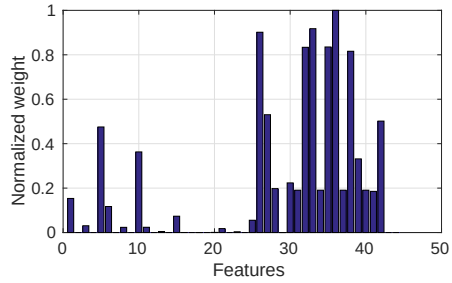
(a) All PCA coefficients (U matrix).



(b) Variance of the principal components.



(c) Most significant PCA coefficients.



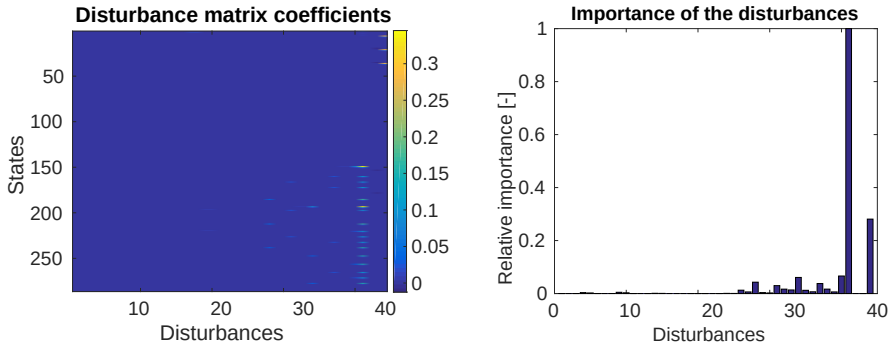
(d) Feature weights on most significant components.

Figure 5.6: Feature selection via principal component analysis.

method is based on the importance metric Id for each disturbance given by the following equation:

$$Id_j = \max(d^j) \sum_{i=1}^{n_x} |E_{i,j}|. \tag{5.39}$$

The sum of the absolute values of the coefficients for j-th column of the matrix E represents the aggregated dynamical effect of the j-th disturbance on the state variables. This effect is scaled with the maximal values of the disturbances d from the dataset. Than the relative importance rId_j for the j-th disturbance is computed by dividing the metric Id_j by the overall maximal value of metric $\max(Id_j), \forall j \in \mathbb{N}_0^{n_d}$. The Fig. 5.7 shows the visual representation of this metric for the disturbances of the model from Section 4.3. We can see from Fig. 5.7(a), that the matrix E is very sparse. Therefore it is not surprising that the relative importance of the disturbances is uneven as shown in Fig. 5.7(b). The feature selection is now based on the selection of the threshold for the relative importance and elimination all the disturbances not meeting this criterion. In our case, we chose this threshold to be equal to 0.001, which in combination with previous approaches as given in Sections 5.4.6 and 5.4.6, results in the selection of 12 out of 44 original disturbance variables.



(a) Disturbance matrix coefficients. (b) Relative importance of the disturbances.

Figure 5.7: Feature selection via disturbance dynamics analysis.

Feature Engineering Overview

The dimensionality of the original parametric space ξ for the MPC problem 5.8, with prediction horizon $N = 22$, and with the model from Section 4.3, is given by the formula: $n_{\mathrm{x}} + Nn_{\mathrm{r}} + Nn_{\mathrm{d}} = 1518$ (see Table 4.3). By applying the FS methods presented in Sections 5.4.6, 5.4.6 and 5.4.6, the dimensionality of the reduced parametric space $\tilde{\xi}$ is now given by the formula $n_{\mathrm{y}} + Nn_{\tilde{\mathrm{r}}} + Nn_{\tilde{\mathrm{d}}}$ and contains only 204 selected features. Here $n_{\tilde{\mathrm{r}}} = 1$ and $n_{\tilde{\mathrm{d}}} = 8$ represent the selected reference and disturbance signals dimensions, respectively. Table 5.3 summarizes the FS procedure applied on the disturbances, where the 8 most important signals are selected by the intersection of the results from each FS method (see the last three columns of Table 5.3).

The time delayed features are used in general, to improve the performance of the ML models. The value of the prediction horizon $N = 22$ in the MPC setup is used as a time delay operator in the context of the TDNN, generating 22 shifted signals which correspond to the predicted values of the comfort boundary $\underline{y}$ and disturbance d signals, respectively. Further, due to the effect of the building mass, also the past values of the output variables y are used, which allows preserving more of the process memory and hence improve the performance. The dimensionality of the reduced parametric space $\tilde{\xi}$ for the TDNN is now given as $22(n_{\mathrm{y}} + n_{\tilde{\mathrm{r}}} + n_{\tilde{\mathrm{d}}}) = 330$. The feature engineering for the RT differs however due to the different nature of the model. Here we use only two shifted signals for the predicted values of the comfort boundary and disturbance signals, respectively. The delayed signals of the output variables are not used in this case. Instead, the linear time is transformed into the three sinusoidal signals ($n_{\mathrm{t}} = 3$) with different frequencies corresponding to days, weeks and months. The dimensionality of the reduced parametric space $\tilde{\xi}$ for th RT is now given as $n_{\mathrm{y}} + 2(n_{\tilde{\mathrm{r}}} + n_{\tilde{\mathrm{d}}}) + n_{\mathrm{t}} = 27$. The overview of the control and machine learning setup with overall dimensions of the selected and transformed features is given in Table 5.4.

5.5 Summary

This chapter presented the main theoretical contributions towards the thermal comfort control in buildings. The first section of this chapter defined the control objectives, namely thermal comfort and minimization of energy use, representing the key performance measures for case studies in Chapter 6. In the sequel, the different control strategies have been introduced. The current practice rule-based (RBC) and PID controllers are confronted with advanced model predictive control (MPC) based algorithms. Section 5.3 presented the general MPC setup used in this thesis with the state and disturbance estimation. Two different MPC formulations were proposed. First, the ideal case deterministic MPC formulation with the perfect knowledge about the future disturbances. Here, the computational efficiency of the MPC algorithm was increased by using so-called *state condensing method*. Second, the more realistic stochastic MPC formulation with the probabilistic modeling of the disturbances. In this case, the computational efficiency of the implementation was achieved by the explicit solution of the corresponding optimization problem.
Section 5.4 introduced the flagship of this thesis, the versatile methodology for the synthesis of simple, yet well-performing approximated MPC feedback strategies suitable for handling the complex building control problems with hundreds or even thousands of parameters. This approach is based on the *teacher* MPC which feeds the multivariate regression algorithms with closed-loop data to derive the dependence of the real-valued control inputs on measurements. Two regression models were studied in detail, the regression trees (RT) and the deep time delay neural networks (TDNN). Here, the significant theoretical contribution was made towards the RT. The performance of the RT was increased by optimally splitting the nodes of the tree by the general hyperplanes instead of the orthogonal hyperplanes and by replacing the local constant regressors by affine expressions. As a consequence, such a regression-based feedback policy can be described as a PWA function defined over a polyhedral domain, in contrast with the traditional RT described as a PWC function defined over an axis-aligned hyperbox domain.
The performance, the complexity and the implementation cost of the proposed regression-based algorithms are improved by employing the dimensionality reduction algorithms for selecting only the most significant MPC parameters as features for machine learning models. The main advantage of the proposed regression-based control strategies stems from their easy implementation even on a very simple hardware. The performance and computational efficiency of all devised MPC-based control strategies is a subject of the four simulation case studies investigated in Chapter 6.

Notation	Description	Units	Manual	PCA	Dynamics
d_1	Absorbed heat in layer 1 of window 1	$[W]$	yes	yes	no
d_2	Absorbed heat in layer 2 of window 1	$[W]$	no	no	no
d_3	Absorbed heat in layer 3 of window 1	$[W]$	yes	no	no
d_4	Direct solar radiation through window 1	$[W]$	yes	no	no
d_5	Diffuse solar radiation through window 1	$[W]$	yes	yes	no
d_6	Absorbed heat in layer 1 of window 2	$[W]$	yes	yes	no
d_7	Absorbed heat in layer 2 of window 2	$[W]$	no	no	no
d_8	Absorbed heat in layer 3 of window 2	$[W]$	yes	no	no
d_9	Direct solar radiation through window 2	$[W]$	no	no	no
d_{10}	Diffuse solar radiation through window 2	$[W]$	yes	yes	no
d_{11}	Absorbed heat in layer 1 of window 3	$[W]$	yes	no	no
d_{12}	Absorbed heat in layer 2 of window 3	$[W]$	no	no	no
d_{13}	Absorbed heat in layer 3 of window 3	$[W]$	no	no	no
d_{14}	Direct solar radiation through window 3	$[W]$	no	no	no
d_{15}	Diffuse solar radiation through window 3	$[W]$	yes	yes	no
d_{16}	Absorbed heat in layer 1 of window 4	$[W]$	no	no	no
d_{17}	Absorbed heat in layer 2 of window 4	$[W]$	no	no	no
d_{18}	Absorbed heat in layer 3 of window 4	$[W]$	no	no	no
d_{19}	Direct solar radiation through window 4	$[W]$	no	no	no
d_{20}	Diffuse solar radiation through window 4	$[W]$	no	no	no
d_{21}	Absorbed heat in layer 1 of window 5	$[W]$	yes	no	no
d_{22}	Absorbed heat in layer 2 of window 5	$[W]$	no	no	no
d_{23}	Absorbed heat in layer 3 of window 5	$[W]$	no	no	no
d_{24}	Direct solar radiation through window 5	$[W]$	no	no	no
d_{25}	Diffuse solar radiation through window 5	$[W]$	yes	yes	no
d_{26}	Direct sun radiation on horizontal surface	$[W]$	yes	yes	yes
d_{27}	Diffuse sun radiation on horizontal surface	$[W]$	yes	yes	no
d_{28}	Weighted sun radiation temperature between ground and sky temperature 1	$[K]$	yes	yes	yes
d_{29}	Direct sun radiation on vertical surface with orientation 1	$[W/m^2]$	yes	no	no
d_{30}	Diffuse sun radiation on vertical surface with orientation 1	$[W/m^2]$	yes	yes	no
d_{31}	Weighted sun radiation temperature between ground and sky temperature 2	$[K]$	yes	yes	yes
d_{32}	Direct sun radiation on vertical surface with orientation 2	$[W/m^2]$	yes	yes	yes
d_{33}	Diffuse sun radiation on vertical surface with orientation 2	$[W/m^2]$	yes	yes	yes
d_{34}	Weighted sun radiation temperature between ground and sky temperature 3	$[K]$	no	yes	yes
d_{35}	Direct sun radiation on vertical surface with orientation 3	$[W/m^2]$	yes	yes	yes
d_{36}	Diffuse sun radiation on vertical surface with orientation 3	$[W/m^2]$	yes	yes	no
d_{37}	Weighted sun radiation temperature between ground and sky temperature 4	$[K]$	no	yes	yes
d_{38}	Direct sun radiation on vertical surface with orientation 4	$[W/m^2]$	yes	yes	yes
d_{39}	Diffuse sun radiation on vertical surface with orientation 4	$[W/m^2]$	yes	yes	no
d_{40}	Weighted sun radiation temperature between ground and sky temperature 5	$[K]$	no	yes	yes
d_{41}	Ambient temperature	$[K]$	yes	yes	yes
d_{42}	Convective heat coefficient	$[W/m^2]$	yes	yes	no
d_{43}	Dummy input for constant value	$[-]$	no	no	no
d_{44}	Ground temperature	$[K]$	yes	no	yes

Table 5.3: Selection of the disturbances as features based on the methods from Section 5.4.6.

Notation	Control setup	ML setup	RT dim.	TDNN dim.
ξ	MPC parameters	original features	1518	1518
$\tilde{\xi}$	subset of MPC parameters	selected/transformed features	27	330
y	outputs	selected features	6	6
$\underline{y}$	lower comfort bounds	selected features	1	1
d	disturbances	selected features	8	8
t	time	transformed features	3	-
N	prediction horizon	number of time delays	2	22
u	control inputs	targets	1	6

Table 5.4: Machine learning setup and feature selection overview, with corresponding dimensions of the RT and TDNN models.

Chapter 6

Case Studies

If you wait until there is another case study in your industry, you will be too late!

Seth Godin

This chapter introduces application of several advanced MPC strategies formulated in Section 5 on building climate control problems employing the mathematical models from Section 4.

6.1 Explicit Stochastic Model Predictive Control

In this section we compare the performance of the stochastic explicit MPC controller against best-case and worst-case scenarios, formulated in Section 5.3.4. The approach is demonstrated on a case study that assumes control of the room temperature in a single-zone building model from Section 4.2. It will be shown that the stochastic controller attains almost the same level of performance as the unrealistic best-case controller, while taking into account the uncertainties about the future evolution of the disturbances.

6.1.1 Simulation Setup

To validate performance of the stochastic explicit MPC strategy derived per Section 5.3.4, we have assumed a building model from Section 4.2 and a simulation scenario that covered 9 days of historical data. Historical evolution of disturbances (outdoor temperature, heat generated by occupancy, and solar heat) is shown in Fig. 6.1.

The explicit representation (3.3) of the stochastic MPC feedback strategy was obtained by formulating (5.8) with probabilistic constraints (5.22) in YALMIP (Löfberg 2004) and solving the QP parametrically by the MPT toolbox (Herceg et al. 2013). The feedback law

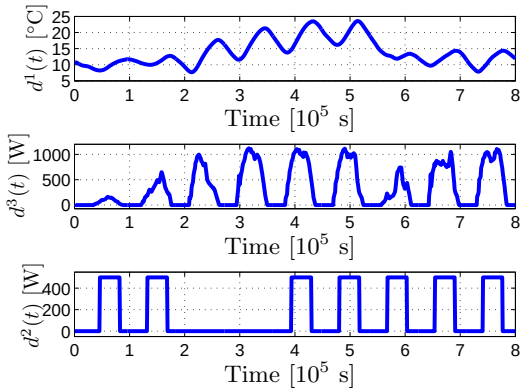


Figure 6.1: Historical trends of disturbances over 9 days. From top to bottom: external temperature d^1, solar radiation d^3, and heat generated by occupancy d^2.

covers following ranges of initial conditions:

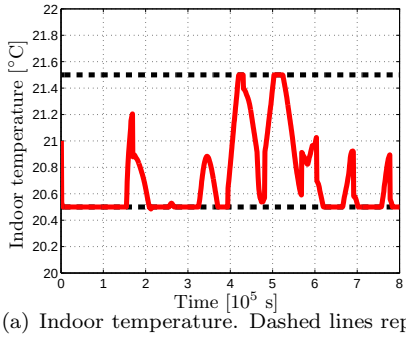
$$\begin{aligned} -10\,^\circ\mathrm{C} \leq x_k^i \leq 35\,^\circ\mathrm{C},\ i \in \mathbb{N}_1^4, \\ 15\,^\circ\mathrm{C} \leq r_k \leq 25\,^\circ\mathrm{C}, \\ 0\,^\circ\mathrm{C} \leq d_k^1 \leq 24\,^\circ\mathrm{C}, \\ 0\,\mathrm{W} \leq d_k^2 \leq 500\,\mathrm{W}, \\ 0\,\mathrm{W} \leq d_k^3 \leq 1200\,\mathrm{W}. \end{aligned}$$

Here the thermal comfort zone was defined by the reference signal r and its width, which was set to $\pm 0.5\,^\circ$C. With the prediction horizon $N = 10$, sampling time $T_s = 890$ seconds, $\alpha = 0.05$, and $\beta = 1{\cdot}10^{-7}$, the explicit MPC feedback (3.3) was obtained as a PWA function that consisted of 816 polyhedra in the 14-dimensional space of initial conditions. Please note that the sampling time was determined analytically based on known time constant of the building model by formula $T_s = T/15$.

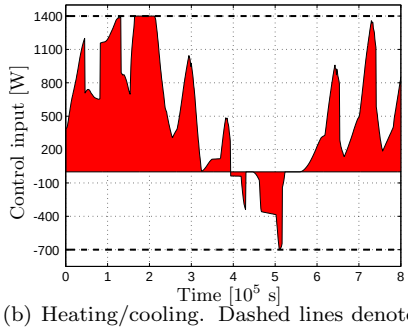
6.1.2 Simulation Results

Simulated closed-loop profiles of the indoor temperature and consumed heating/cooling energy governed by the stochastic MPC are shown in Fig. 6.4. Here the dashed lines represent the thermal comfort zone and limits of control authority, respectively. As can be seen, the stochastic controller allows for seldom violations of the thermal comfort zone while maintaining hard limits of the control authority. Overall, the stochastic controller maintains the indoor temperature within the comfort zone for 97.2 % of samples.
Performance of the explicit stochastic MPC scheme was then compared to two alternatives. One is represented by a best-case MPC controller, which assumes perfect knowledge of future disturbances over a given prediction horizon. The other alternative is a worst-case scenario which employs conservative bounds on the rate of change of future disturbances. Hence it guarantees satisfaction of constraints in robust fashion, while only minimizing

the energy consumption with respect to the worst possible disturbance. Simulated profiles under the best-case and worst-case policies are shown in Fig. 6.2 and Fig. 6.3, respectively. Both controllers always managed to keep the indoor temperature within the thermal comfort zone. Moreover, the best-case scenario provides least energy consumption. The worst-case approach, on the other hand, maintains the temperature further away from the boundary of the comfort zone, which leads to an increased consumption of heating/cooling energy. This is a consequence of using conservative bounds on the rate of change of disturbances in the future. Aggregated results are reported in Table 6.1 and captured in compact grapicall form in Fig. 6.5. It should be pointed out that the best-case scenario, although it performs best, is only of fictitious nature since in practice future disturbances are not know precisely. The stochastic scenario, on the other hand, can be easily employed in practice. Moreover, it provides performance comparable to the best-case approach.

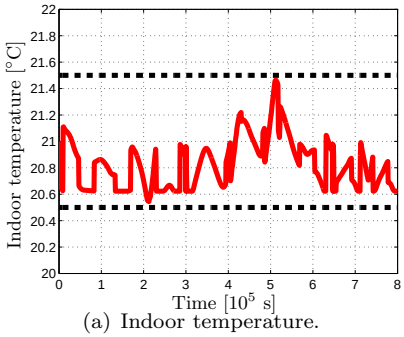


(a) Indoor temperature. Dashed lines represent the thermal comfort zone.

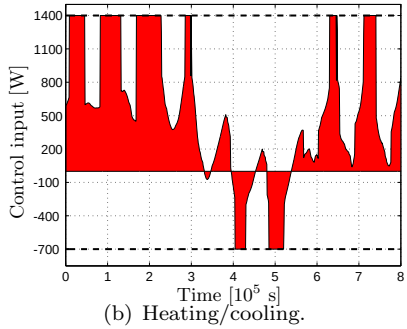


(b) Heating/cooling. Dashed lines denote limits of control authority.

Figure 6.2: Performance of the best-case MPC controller with complete knowledge of future disturbances.

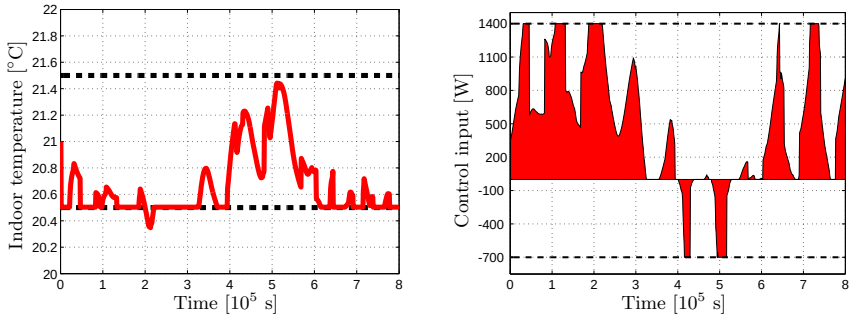


(a) Indoor temperature.



(b) Heating/cooling.

Figure 6.3: Performance of the worst-case MPC controller which assumes conservative bounds on future disturbances.



(a) Indoor temperature. (b) Heating/cooling.

Figure 6.4: Performance of the stochastic MPC controller.

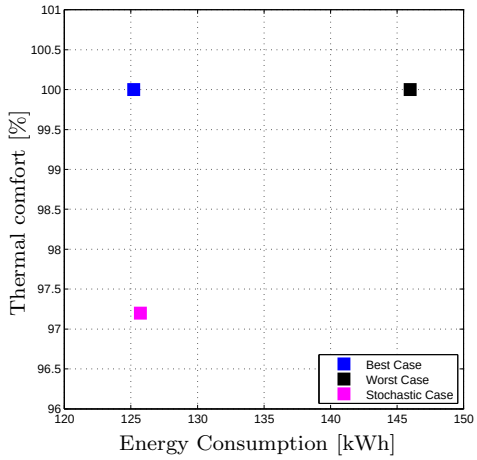


Figure 6.5: Comparison of all control approaches with respect to achievable thermal comfort (x-axis) and energy consumption (y-axis). Each point represents aggregated performance of one controller.

6.1.3 Summary

In this Section a performance of the stochastic MPC was compared against the best-case and worst-case scenarios. It was shown that the stochastic controller outperforms the worst-case controller and attained almost the same level of performance as the unrealistic best-case controller, while taking into account the uncertainties about the future evolution of the disturbances. Moreover, the computationally efficient explicit feedback law (3.3) was obtained as a PWA function, allowing for easy employment in practice even on a simple hardware.

	Thermal comfort	Consumed energy
Stochastic MPC	97.2 %	125.7 kWh
Best-case MPC	100.0 %	125.2 kWh
Worst-case MPC	100.0 %	146.0 kWh

Table 6.1: Comparison of various MPC strategies.

6.2 Approximate MPC via Enhanced Regression Trees

This section shows the performance of simple, yet well-performing feedback strategies that mimic the behavior of model predictive control (MPC) as proposed in Section 5.4. The approach is based on employing the enhanced regression trees, given in Section 5.4.4, to derive the dependence of real-valued control inputs on measurements. This section furthermore illustrates how to refine the local regressors such that the overal feedback strategy guarantees the satisfaction of the input constraints. The approach is demonstrated on a case study that assumes control of the room temperature in a single-zone building model from Section 4.2. It will be shown that the simple feedback law attains almost the same level of performance as the complex MPC controller.

6.2.1 Refinement of Local Regressors

Although the regression-based control policy (5.30) synthesized per Section 5.4.4 provides an optimal regression of the training data, it does not posses guarantees that $u = f_\Theta(\xi)$ satisfies input constraints $u \in \mathcal{U}$. Therefore in this section we show how to replace $f_\Theta(\cdot)$ in (5.30) by a different function

$$\tilde{f}_{\text{reg}}(\xi) = \begin{cases} \tilde{f}_{\text{reg},1}(\xi) & \text{if } \xi \in \mathcal{P}_1, \\ \vdots & \\ \tilde{f}_{\text{reg},M}(\xi) & \text{if } \xi \in \mathcal{P}_M, \end{cases} \tag{6.1}$$

defined over the same cells $\mathcal{P}_1, \ldots, \mathcal{P}_M$, but with the local regressors $\tilde{f}_{\text{reg},i}(\cdot)$ being such that $u = \tilde{f}_{\text{reg}}(\xi) \in \mathcal{U}$ for all $\xi \in \mathcal{P}$.

The case in which satisfaction of input constraints is easily obtained is when $\mathcal{P}$ is a polytope and $\mathcal{U} \subseteq \mathbb{R}^{n_\text{u}}$ is a polyhedron. Consider the i-th terminal node of the binary tree, which is composed of the local regressor $f_{\text{reg},i}(\xi) = F_i\xi + g_i$ and the polytopic region of validity $\mathcal{P}_i$. Then, the optimal refinement $\tilde{f}_{\text{reg},i}(\xi) = \tilde{F}_i\xi + \tilde{g}$ of $f_{\text{reg},i}(\cdot)$ such that $\tilde{f}_{\text{reg},i}(\xi) \in \mathcal{U}$ for all $\xi \in \mathcal{P}_i$ is computed by solving the following quadratic program:

$$\min_{\tilde{F}_i, \tilde{g}_i} \sum_{v \in \mathcal{V}_i} (f_{\text{reg},i}(v) - \tilde{f}_{\text{reg},i}(v))^T (f_{\text{reg},i}(v) - \tilde{f}_{\text{reg},i}(v)) \tag{6.2a}$$

$$\text{s.t. } \tilde{f}_{\text{reg},i}(v) \in \mathcal{U}, \ \forall v \in \mathcal{V}_i, \tag{6.2b}$$

where $\mathcal{V}_i = \{v_{i,1}, \ldots, v_{i,n_v}\}$ are the vertices of $\mathcal{P}_i$. Here, (6.2a) optimizes parameters of $\tilde{f}_{\mathrm{reg},i}(\cdot)$ such that they are as close as possible to $f_{\mathrm{reg},i}(\cdot)$. Since both functions are assumed to be affine, minimizing the point-wise mismatch at the vertices of $\mathcal{P}_i$ is equivalent to minimizing the integrated squared error $\int \|f_{\mathrm{reg},i}(\xi) - \tilde{f}_{\mathrm{reg},i}(\xi)\| \, \mathrm{d}\xi$, evaluated over $\mathcal{P}_i$. Moreover, it is trivial to prove that if $\mathcal{U}$ is a convex set, $\mathcal{P}_i$ is a polytope, and $\tilde{f}_{\mathrm{reg},i}(\cdot)$ is an affine function, then $\tilde{f}_{\mathrm{reg},i}(v) \in \mathcal{U}$ for each vertex of $\mathcal{P}_i$ is necessary and sufficient for $\tilde{f}_{\mathrm{reg},i}(\xi) \in \mathcal{U}$ for all points $\xi \in \mathcal{P}_i$. Finally, note that with $\mathcal{U}$ a polyhedron, constraints in (6.2b) are linear in the decision variables $\tilde{F}_i$, $\tilde{g}_i$ that constitute $\tilde{f}_{\mathrm{reg},i}(\xi) = \tilde{F}_i\xi + \tilde{g}_i$ since the vertices are known. Moreover, the objective function (6.2a) is quadratic in $\tilde{F}_i$ and $\tilde{g}_i$ since $f_{\mathrm{reg},i}(\xi) = F_i\xi + g_i$ is known in each terminal node. Notice that (6.2) is feasible in each terminal node for an arbitrary non-empty polyhedron $\mathcal{U}$.

Therefore the refined feedback policy $u = \tilde{f}_{\mathrm{reg}}(\xi)$ which provides $u \in \mathcal{U}$ for all $\xi \in \mathcal{P}$ can be easily obtained by solving (6.2) in each terminal node of $f_\Theta(\cdot)$ in (5.30). Notice that only parameters of the local regressors are modified, while the splits stay the same. Clearly, there is no guarantee that the refined local regressor $\tilde{f}_{\mathrm{reg},i}(\xi) = \tilde{F}_i\xi + \tilde{g}$ is optimal w.r.t. (5.31). Unfortunately, since the refinement in (6.2) depends on vertices of $\mathcal{P}_i$, which in turn depend on the split $\sigma(\cdot)$, it is not possible to include (6.2b) directly into (5.35) and still be able to solve the regression problem as a mixed-integer quadratic program. Also note that enforcing $f_\Theta(\xi_j) \in \mathcal{U}$ for all the training datapoints $\xi_1, \ldots, \xi_m$ is merely necessary, but not sufficient, to guarantee that $f_\Theta(\xi) \in \mathcal{U}$ for an arbitrary $\xi \in \mathcal{P}$.

6.2.2 Simulation Results

In this section we demonstrate performance of the proposed regression-based control policy on simulation scenario over the period of 31 days that involves control of a single-zone building from Section 4.2, with the corresponding evolution of the disturbances is shown in Fig. 6.6.

We assume that the state vector $x(t)$ and the vector of disturbances $d(t)$ can be measured at each time instant t, but future evolution of disturbances is unknown. Therefore we assume a constant dynamics for disturbances in the prediction model. The quadratic objective function (5.8a) of the MPC problem (5.8) is modified to the linear form:

$$\min_{u_0,\ldots,u_{N-1}} \sum_{k=0}^{N-1} (Q_{\mathrm{s}} s_k + |u_k|) \tag{6.3}$$

where $s_k \geq 0$ are the slack variables that soften the thermal comfort constraints in (5.8d). To limit the magnitude of violations of the optimal thermal zone, the non-negative slacks are penalized by a large penalty Q_{s} in (6.3). Moreover, the prediction model in (5.8b) is defined by the state-update matrices of (4.1) discretized with sampling period $T_s = 900$ seconds. The input constraints (5.8e) are $\underline{u} = -1000$ W, $\overline{u} = 2000$ W. The lower and upper comfort boundaries are defined by the reference signal T_{ref} and by the the width of the thermal comfort zone $\zeta = 0.5°\mathrm{C}$ as: $lb_k = T_{\mathrm{ref}} - \zeta$ and $ub_k = T_{\mathrm{ref}} + \zeta$, respectively. Note that the problem (5.8) with the objective (6.3) is a linear program with parameters

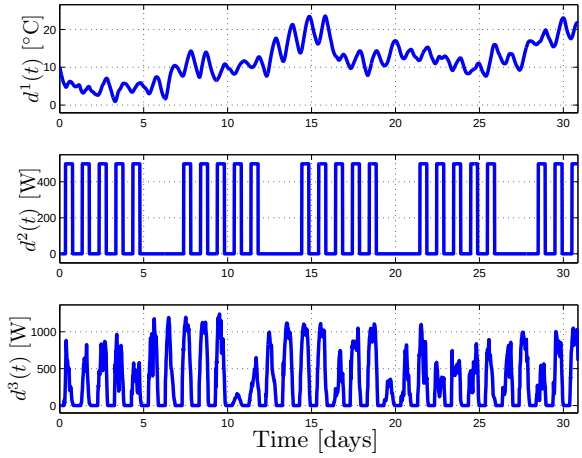


Figure 6.6: Historical trends of the disturbance variables over 31 days: d^1 is the external temperature, d^2 stands for heat due to occupancy, and d^3 represents the solar radiation.

$\xi = [x(t)^T, d(t)^T, T_{\text{ref}}] \in \mathbb{R}^8$.

To construct the regression-based control policy $u = f_\Theta(\xi)$ as in (5.30), we have first performed a closed-loop simulation over the period of 5 days and collected the closed-loop training data $\{(\xi_1, u_1), \ldots, (\xi_m, u_m)\}$ with $m = 486$ (5 days sampled at 900 seconds). The loop was closed by the receding horizon feedback (5.9) where the optimal open-loop sequence was generated at each simulation step by solving the corresponding MPC problem with prediction horizon $N = 48$ steps (12 hours sampled at 900 seconds), the reference temperature set constantly to $T_{\text{ref}} = 20°\text{C}$, and penalty on the slacks $Q_{\text{s}} = 1 \cdot 10^6$. The same historical profiles of disturbances as in Fig. 6.6 were employed in the simulation. The initial state for the simulation was set to $x(0) = [20, 20, 20, 20]^T$. The training data generated over 5 days by the MPC controller are shown in the top plot in Fig. 6.7(a). The same figure also depicts the performance of the MPC controller on the remaining 24 days of simulation. The accumulated heating/cooling cost under the optimal MPC feedback policy was 592 kWh. Note that the violations of the lower temperature range in the first 7 days are due to the control action being saturated at $\overline{u}$.

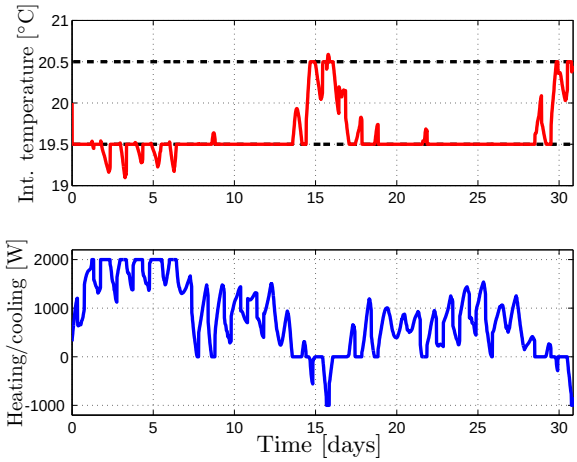
Then we have computed the regression-based feedback policy $u = f_\Theta(\xi)$ as in (5.30) by applying Algorithm 2 to the training data. In Step 2 of the algorithm, optimal splits and local regressors were calculated by solving the MIQP problem (5.35). The final tree consisted of 5 non-terminal, and 6 terminal nodes, along with 6 local affine regressors. Subsequently, the local regressors were refined per the procedure of Section 6.2.1 to guarantee that the regression-based feedback always provides satisfaction of input constraints.

To validate performance of the proposed regression-based feedback strategy and to evaluate decrease in performance with respect to MPC, we have performed a closed-loop simulation under the same conditions of the MPC scenario described above. The closed-loop profiles

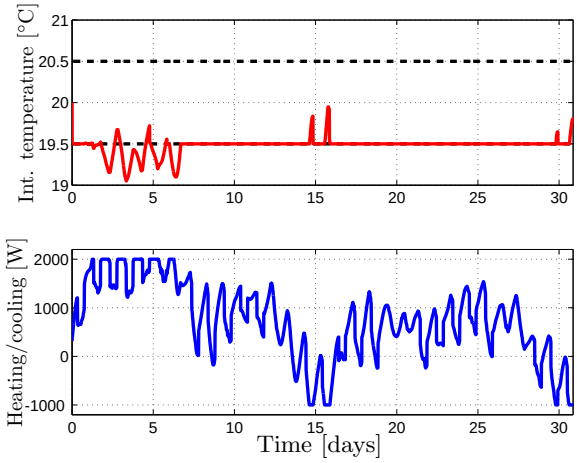
of the internal building temperature and of the control actions provided by $f_\Theta(\cdot)$ are shown in Fig. 6.7(b). As can be clearly seen, performance of $f_\Theta(\cdot)$ is close to the behavior of the MPC policy in Fig. 6.7(a). The only notable differences are between days 13 to 17, which correspond to hot days (cf. top part of Fig. 6.6). Here, the MPC policy is able to exploit the thermal zone to slightly reduce consumption of cooling energy by allowing the internal temperature to hit the upper limit of the thermal comfort zone. A similar phenomenon also occurs in the final 3 days of the simulation.
Besides these difference, the regression-based feedback matches the MPC controller appropriately. Although the regressor was only trained on the first 5 days of MPC profiles, the local affine regressors are able to reasonably extrapolate the control actions also in situations that were not present in the training data. The accumulated energy consumption under $f_\Theta(\cdot)$ was 611 kWh, an increase of mere 3% over the MPC strategy. The biggest difference between $f_\Theta(\cdot)$ and the MPC feedback, though, is in the implementation cost. While the MPC policy requires solving a corresponding optimization problem at each sampling instant, the regression-based controller only needs to evaluate the function in (5.30) via the binary search tree. The tree consisted of 5 affine splits $\sigma(\cdot)$ as in (5.34a), which require storing 45 floating point numbers (the vectors $\alpha \in \mathbb{R}^8$ and the scalar β for each split). The 6 local affine regressors contributed by another 54 floating point numbers. To evaluate $f_\Theta(\cdot)$ for a particular ξ, one then needs to traverse through the 4 levels of the tree, evaluating $\sigma(\xi)$ at each level. In total, the traversal requires less than 100 floating point operations at each sampling instant. The low memory storage, combined with low implementation effort, make the proposed regression-based feedback strategy easily applicable on typical building automation hardware, such as on programmable logic controllers.

6.2.3 Summary

This case study had shown the performance of simple, yet well-performing feedback strategies that mimic the behavior of MPC. The regression-based strategies were synthesized by employing the enhanced regression trees (RT) introduced in Section 5.4.4, to derive the dependence of real-valued control inputs on measurements. It was furthermore illustrated how to refine the local regressors such that the overal feedback strategy guarantees the satisfaction of the input constraints. Although the regressor was only trained on the first 5 days of the MPC profiles, it was able attain almost the same level of performance as the complex MPC controller, and it was able to reasonably extrapolate the control actions also in situations that were not present in the training data. The constructed simple regression-based feedback law consisted only of 5 non-terminal, and 6 terminal nodes with 6 local affine regressors, therefore providing a cheap implementation cost.



(a) MPC feedback.



(b) Regression-based feedback policy in (6.1).

Figure 6.7: Closed-loop profiles. The first 5 days were used as training data. The top figures show the internal building temperature (solid red line), along with $\pm\zeta$ range of the reference temperature (black dashed lines). The bottom figures depict the associated optimal control actions.

6.3 Impact of the Controller Model Complexity on MPC Performance

MPC for buildings requires accurate controller models of the building envelope and its HVAC systems. Controller models are typically obtained by means of black- or grey-box system identification or using a white-box modelling approach. However, the necessary level of the model complexity used by each method in order to obtain a good MPC performance remains a priori unknown and no systematic method or examples showing the optimal complexity is available. This case study systematically investigates the influence of the controller model accuracy on the evaluation of the building climate controller performance and the minimum number of states necessary to obtain optimal control performance.

6.3.1 Methodology

The methodology is graphically represented in Fig. 6.8. For this study, a 6-rooms house

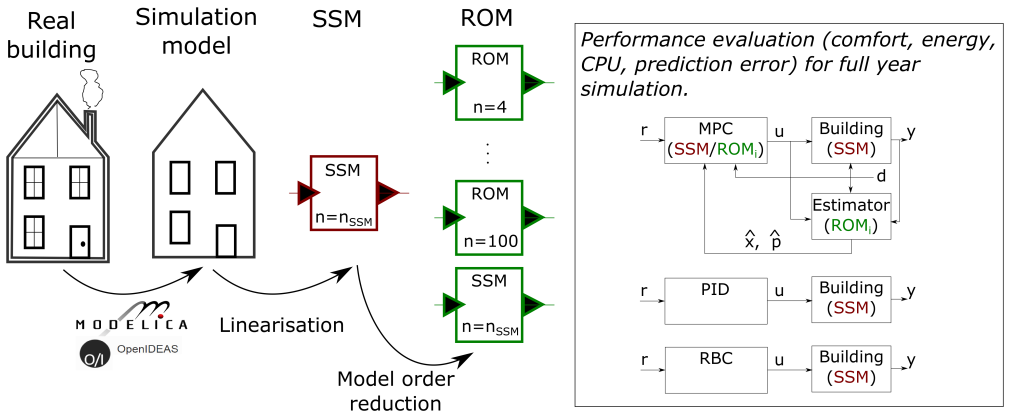


Figure 6.8: Schematic view of the methodology. From left to right : a 6-rooms house is modeled using the BES Modelica library `IDEAS`. The obtained model is then linearized and converted to a time-invariant SSM. Balanced truncation MOR technique is used to obtain ROMs of different orders. Finally, the upper bound of the controller performance is computed by using the SSM model both as controller and plant model (theoretical benchmark). The performances of the MPC using the different ROMs as controller model are compared with the upper bound and with a RBC and a PID controller.

described in the Section 4.3 is used. In the first modelling phase the existing building is used to ensure reasonable parameter values. While the Modelica reference building model is accurate in the physics it describes, its mathematical formulation combines non-linear partial differential equations, ordinary differential equations and algebraic equations. However, the non-linear model can be accurately linearized around a given working point

(see Section 4.3.3), and as such it can be transformed into a form of LTI SSM (Picard et al. 2015). The obtained SSM is then used for simulations and control purposes in MATLAB ® environment. The MPC problem (5.8) is solved using the SSM both as controller and as plant model, i.e. with no plant model mismatch. This case cast as the theoretical performance bound (PB). The accuracy of the SSM (containing more than 250 states) is further artificially decreased by reducing its number of states to different orders ranging from 4 to 100 using model order reduction technique from Section 4.3.4. The ROMs are used to mimic the best possible low order controller models which can be obtained for these orders by means of system identification, for example. Finally, the influence of the controller model complexity is investigated by evaluating the thermal comfort, the energy use, the computational effort (CPU) and the prediction error of the different MPCs, each using a controller model with a different complexity while the building model is kept unchanged. Additionally, the MPC performance is compared with a traditional thermostat RBC and a PID controller.

In order to generalize the results, the same methodology is repeated for to three different scenarios (see Section 4.3.1): 1) the original building, 2) the same building but with an improved insulation level (Renovated building), and 3) the same building but with light-weight wooden walls instead of concrete walls (Lightweight building).

6.3.2 Simulation Setup

This section discusses the performance criteria, setup of the simulation parameters and tuning of the individual controllers. All parameters are chosen based on the explicit analyzes.

Performance Criteria

The controller performances are evaluated using four performance keys: energy use, thermal comfort level, 1-step ahead prediction error and CPU time. The energy use corresponds to the heat delivered by the radiators and is expressed in kWh. The thermal comfort κ is defined as the number of sampling instants in which one of the zone temperatures y_i falls inside the relaxed comfort bounds $[\underline{y}_k - \tau, \overline{y}_k + \tau]$, divided by the simulation length N_{sim} and the number of zones N_{rooms}:

$$\kappa = \frac{\sum_i^{N_{\text{rooms}}} \sum_k^{N_{\text{sim}}} \epsilon_{i,k}}{N_{\text{rooms}} N_{\text{sim}}} \times 100\% \tag{6.4a}$$

$$\epsilon_{i,k} = \begin{cases} 1 & \text{if} \quad \underline{y}_k - \tau \leq y_{i,k} \leq \overline{y}_k + \tau \\ 0 & \text{otherwise} \end{cases} \tag{6.4b}$$

with a violation tolerance $\tau = 0.1$K. The 1-step ahead prediction error is the error between the prediction of the zone temperatures made by the Luenberger observer and the outputs of the building model at the next time step. Finally, the CPU time corresponds to the overall simulation time.

Simulation Parameters

Based on the dynamic response of the building models, the sampling period was chosen equal to $T_s = 900$ seconds. The maximum heating power of the radiators for each building, representing the upper bound on the control action, is shown in Table 6.2. The values are chosen such that the maximum powers correspond to the loads needed to achieve thermal comfort on the coldest days. The state values $x(0)$ are initialized to $20°C$, based on the

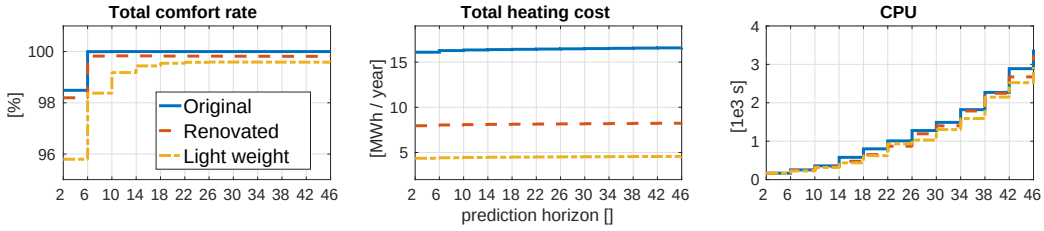
Model Type	Maximum Radiator Gains $\overline{u}$ $[W]$
Original	$[2940\ 960\ 300\ 1400\ 460\ 253]^T$
Renovated	$[1680\ 685\ 154\ 1000\ 320\ 232]^T$
Light Weight	$[840\ \ 343\ \ 77\ \ 500\ \ 160\ 116]^T$

Table 6.2: Maximum heating power of the radiators (per zone).

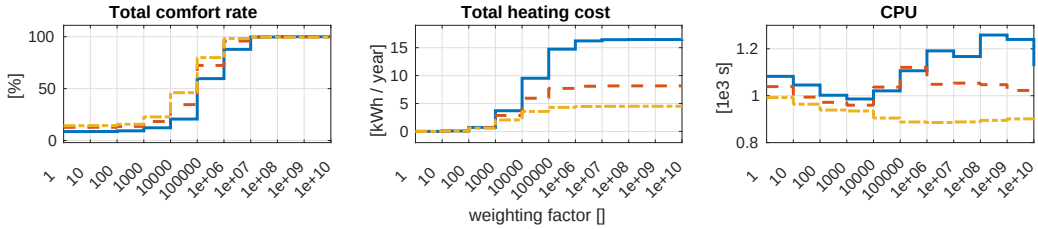
procedure described in Section 4.3.4. The disturbance vector d was generated from a typical year in Uccle, Belgium (Meteotest 2009). The overall simulation period was chosen to be a single year.

Controllers Tuning

To improve thermal comfort satisfaction of the benchmark controllers (RBC and PID) and as such ensuring a fair comparison with MPC and neural network controllers, the reference temperatures r_k are shifted slightly above the lower boundary of the comfort band lb_k. For the RBC controller: $r_k = \underline{y}_k + 2.5°C$, while the width of the switching zone was equal to $0.5°C$. For the PID controller: $r_k = \underline{y}_k + 1°C$. On a top of that, the positive change in the reference signal was shifted by one hour (4 sampling instants) before the positive change of the upper comfort boundary $\underline{y}$ occurs. This was done to increase the comfort satisfaction for the RBC and PID controllers. For the purpose of PID design, a decoupling of the MIMO SSM into the set of six SISO models is done a priori. The derivation of the PID parameters is performed using MATLAB command `pidtune` with a posteriori manual tuning for enhancing the performance. Both RBC and PID are well tuned with respect to given performance criteria to provide a fair comparison with the MPC controllers.
In case of MPC, the values of the prediction horizon N and the weighting factor $\frac{Q_s}{Q_u}$ are chosen based on the dependence of the MPC performance on the parameter values, as shown in Fig. 6.9. With emphasis on thermal comfort satisfaction the choice of the prediction horizon is set to $N = 40$ steps (i.e., 10 hours), and weighting factor $\frac{Q_s}{Q_u} = 10^8$. We assume here that MPC has full disturbances preview, hence the perfect weather predictions are provided. The comfort band is given by two time-varying parameters, lb_k and ub_k, representing the lower and upper bound, respectively as defined in Section 5.1 based on ISO-7730.
The MPC is constructed in the MATLAB environment, using the modeling and optimization toolbox YALMIP (Löfberg 2004). The closed-loop simulation was performed by applying the optimal control inputs $u^\star(t)$, computed at each sampling instant T_s by MPC to the



(a) Effect of the prediction horizon N.



(b) Effect of the weighting factor $\frac{Q_s}{Q_u}$.

Figure 6.9: Analysis of the MPC performance based on the change of the parameters N and $\frac{Q_s}{Q_u}$, while fixing the rest of the parameters.

building model. The objective function (Eq. (5.8a)) is quadratic, and all constraints are linear; therefore the problem (5.8) can be solved as a convex quadratic program (QP). In this study, we used the state of the art optimization solver GUROBI (Gurobi Optimization 2012).

6.3.3 Simulation Results

Fig. 6.10 presents the performance key values for the full year simulations and for the three building types using different controllers. The bars represent the PID, the RBC and the MPCs with different ROMs as controller models. The stars represent the results for the equivalent OSF-MPCs. Fig. 6.10a shows that the comfort of MPC using the SSM as controller model is very close to 100% for all buildings. The minimal comfort violations here are caused by the small overheating of the well insulated buildings during the hot days. This confirms that the radiators are sized properly. The benchmark controllers are also well tuned as they show a good comfort satisfaction, spanning from 95.6% to 99.6% for the PID and from 93.9% to 95.2% for the RBC. The high comfort satisfaction of the benchmark controllers, however, is coupled to a high energy use (see Fig. 6.10b). In comparison to the MPC with a controller model of highest order, the PID uses 6.6%, 8.9% and 8.0% more energy in the original, the renovated, and the light weight buildings respectively. Due to its switching behavior, the RBC is even less efficient using 11.2%, 12.8% and 8.0% more energy in the original, the renovated, and the light weight buildings, respectively.
Fig. 6.10c shows a decrease of the one-step ahead prediction error with an increase of the

controller model complexity. Here the ROMs with $n_x \geq 30$ have negligible prediction error for all three building types. The prediction error for equivalent controller complexities also decreases with decrease of the building mass and with increase of the insulation. The former is due to the fact that heavy structures require more states to compute the thermal diffusion accurately and the latter is due to a smaller influence of disturbances (ambient temperature). From Fig. 6.10 it appears that MPCs using a ROM of order lower than 30 score significantly worse than MPCs using a higher order ROM. This is due to the prediction error made by the observer, as shown in Fig. 6.10c. Even a very small error difference of 0.2-0.3K between ROM 20 and ROM 30 results in a significant difference in thermal comfort of about 15% without affecting the energy use. Good controller models are thus effectively crucial for multi-zone building control. Note that obtaining an accurate 30 states controller model for a 6-zone building using system identification is a challenging task (Prívara et al. 2013).

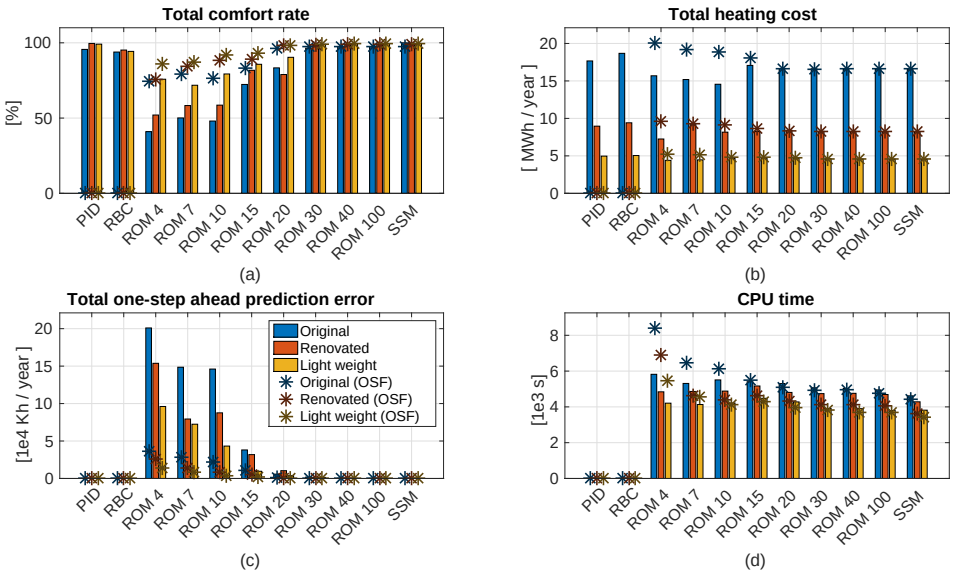


Figure 6.10: Comparison of the performance keys evaluated for the PID, RBC, S-MPC and the OSF-MPC approach for different controller model orders.

Fig. 6.10a shows that OSF-MPCs using low order ROM achieve a significantly better comfort than S-MPC with the same model complexity. This comfort improvement, however, comes with an increase in the energy use (Fig. 6.10b) for the OSF-MPCs using very low order ROMs ($n_x \leq 15$). For ROMs with $n_x > 15$, the comfort improvement comes with a small or negligible increase in energy use. This can be explained by the prediction errors shown in Fig. 6.10c. The OSF approach adds one constant dynamic variable per output to the controller model, compensating the initialization error at each sampling instant, rather than improving the dynamical behavior of the ROM on the whole prediction horizon. Therefore when the model mismatch between controller model and building model is too

large, the OSF method will not guarantee a good performance. Overall, in the case of a sufficiently small model mismatch, the OSF method will improve the MPC results.
A reason to limit the controller model complexity is the computational effort required to solve the optimization problem. Fig. 6.11 shows, however, that when applying the dense approach, the CPU time becomes independent of the number of states. The CPU times for full year simulation scenarios and all building types using the dense approach have an average of 75.9 minutes with the maximum equal to 98.0 minutes and the minimum equal to 54.9 minutes with all computations performed on a 2.8 GHz machine with 2 CPU units each with 6 cores, under the GNU/Linux 64-bit Debian 3.16.7 operating system. As shown by Fig. 6.11 the sparse approach leads to intractable CPU times for a large number of states.

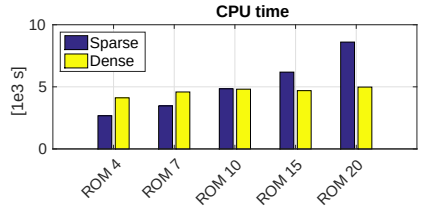


Figure 6.11: Comparison of the computational demands of the sparse and dense formulation of the control problem.

6.3.4 Summary

This case study systematically investigates the influence of the controller model complexity on the performance evaluation of MPC used in building climate control. The number of necessary states decreases with the level of insulation of the building and it increases with its mass content. This study shows that a prediction error of more than 0.5 K within the MPC prediction horizon can significantly lower the MPC performance. In the case of the investigated 6-rooms house, a minimum of 30 states was necessary to obtain the optimal control performance. This number, however, is significantly higher than the typical orders used by black- or grey-box system identification techniques. The minimum number of states might be chosen lower when offset-free MPC (OSF-MPC) is used instead of conventional MPC. However, OSF-MPC might significantly increase the energy use when a significant plant model mismatch is present. Finally, the case study shows that the computational effort required to solve the optimization problem becomes independent on the number of states of the controller model when a dense approach is used. The controller model can thus be as complex as necessary to generate accurate predictions without increasing the solving time.

6.4 Approximate MPC via Time Delay Neural Networks

This section studies the behavior of the approximated MPC controllers (see Section 5.4) on a large scale simulation case study employing the six-zone building model from Section 4.3. The methodology from Section 5.4.1 is followed. Two machine learning (ML) algorithms are trained and tuned on closed-loop simulation data from original MPC. Than the performance of the ML controllers, namely basic regression trees (RT) (see Section 5.4.4) and the time delay neural networks (TDNN) with deep architecture (see Section 5.4.5) is compared with the performance bound (PB) MPC (see Section 5.3.3) and benchmark control strategies: RBC and PID (see Section 5.2). Section 6.4.1 sets up the simulation study and Section 6.4.2 presents and discusses the results. It will be shown that for complex building control problems employing more than a thousand of parameters the performance of the classical RT is somewhat unsatisfactory. On the other hand the deep TDNN are able to cope with the increased complexity and generate low-complexity and well-performing representations of the MPC laws.

6.4.1 Simulation Setup

The overall simulation setup, and MPC used here is based on the case study 6.3, with the performance criteria and tuning of the RBC, PID and MPC conttrollers given in 6.3.2. Here, only the renovated case of the 6-zone building model from Section 4.3 is used.
The overall simulation time was 90 days (8640 data points) which correspond to three months (January to March) in the winter period. The control profiles generated by MPC (see Section 5.3.3) in the first 60 days (5760 data points) were used as a training set for the machine learning models (see Section 5.4), while the last 30 days (2880 data points) acted as a test set for evaluation and comparison of the designed controller. All simulations were performed on a 2.8 GHz machine with two CPU units each with six cores, under a GNU/Linux 64-bit Debian 3.16.7 operating system.

Machine Learning Models Training and Tuning

In the case of the TDNN, all features and targets are normalised to fall in the range $[-1, 1]$. The scaled conjugated gradient was chosen as the fastest training algorithm based on the comparison of the algorithms on similar dataset type provided by MathWorks[1]. To further accelerate the training of the NN, we used the hyperbolic tangent sigmoid activation function for all hidden neurons. Linear functions define the output layer. The initial training performance measure of for the TDNN was a mean squared error (MSE), however, from the control point of view, the actual performance measure was given by the criteria from Section 6.3.2. Because the maximum number of iterations was 4000, the evaluation of the closed-loop simulation performance of the TDNN in each iteration would

[1] https://www.mathworks.com/help/nnet/ug/choose-a-multilayer-neural-network-training-function.html

dramatically slow down the learning process. The simulation performance was therefore evaluated only each 100 iterations as a stopping criterion for the training. This modification in the learning process was done mainly because the MSE of the trained models is not strongly correlated with their control performance. In fact, the NN with lower MSE may have lower control performance due to overfitting. To reduce the overfitting and increase the generalisation of the NN, we retrained each net several times with randomly initialized weights Θ. Another simple method for improving the generalisation of the NN, inspired by Occam's razor, is to use only the NN that is just large enough to provide an adequate fit (Hagan et al. 1996). By training and evaluation of the different architectures multiple times, we choose the simplest best performing network structure with two hidden layers with 24 and 12 neurons, respectively. For the recall, our TDNN had 330 features in the input layer and six-dimensional output layer. This accounts for the total number of 8280 weight parameters Θ, which are represented as a floating point numbers. Assuming that 4 bytes are needed to store a single-precision floating point number, the memory footprint of the TDNN parameters used in this study is 33.1 kilobytes.

The advantage of the RT is that no pre-processing of the data is necessary. A single RT was constructed for each building zone via MATLAB's `fitrtree` function. The MSE was used as a training performance measure as well as a splitting criterion 5 for the RT. The stopping criteria for the Algorithm 2 were given as follows: the minimum number of branch node observations $m_{\min} = 10$, the quadratic error tolerance per node $e_{\min} = 10^{-6}$, the maximal number of decision splits (maximal tree depth) $D_{\max} = m - 1$. These values were chosen in order to grow deep trees, trading the better regression performance for increased complexity. The final set of the six trees, consisted of 1567, 1723, 1385, 1397, 1391 and 1139 nodes, respectively. Out of which, the number of terminal nodes for each tree was 784, 862, 693, 699, 696 and 570, respectively. In total, the combined number of all nodes for six RTs is 8602, out of which 4298 are splitting nodes and 4304 are terminal nodes. For binary regression tree, employing 27 features, the single splitting node requires storing 28 floating point numbers (the vectors $\alpha \in \mathbb{R}^{27}$ and the scalar β for each split), while storing one terminal node requires just a single floating point number. Assuming that 4 bytes are needed to store a single-precision floating point number, the total memory footprint of the six RTs used in this study is hence 481.4 kilobytes, which is roughly fifteen times more than in the case of TDNN.

Both, TDNN and RT, were designed and trained by using built-in functions from MATLAB's Statistics and Machine Learning Toolbox$^{\text{TM}}$. Finally, the post-processing saturation of the control laws was performed to provide the satisfaction of the input constraints.

Heuristic Rules for Post Processing of TDNN Control Laws

To enhance the performance of the TDNN, we introduce simple heuristic rules defined by Algorithm 3. These rules can be used for post-processing of the generated control actions u for each zone, adjusting the room temperatures y to be closer to the lower comfort boundary $\underline{y}$. This behaviour is achieved by slightly decreasing the control action when the temperature is too high, and slightly increasing the control action when the temperature is

about to violate the comfort boundary $\underline{y}$. The tuning parameters of the algorithm, together with their values for our particular case are as follows: lower threshold $thr_{\text{low}} = 0.1$, upper threshold $thr_{\text{up}} = 0.4$ and corresponding lower $G_{\text{low}} = 0.1$ and upper $G_{\text{up}} = 0.05$ correction gain, respectively. Because the rules are intuitive and easily tunable in the field, they increase the adaptivity of the proposed approach also for various cases with some degree of uncertainty.

Algorithm 3 Enhancement of the time delay neural network control laws.

```
function CtrlEnhance(u)                          ▷ given TDNN control action u
    if day then                                  ▷ apply this procedure only during the daytime
        if y < y + thr_low then                  ▷ room temperature too close to lower comfort boundary
            u ← u + G_low ū                      ▷ increase control action
            return u                             ▷ obtain enhanced control action
        end if
        if y > y + thr_up then                   ▷ room temperature too far from lower comfort boundary
            u ← u − G_up ū                       ▷ decrease control action
            return u                             ▷ obtain enhanced control action
        end if
    end if
end function
```

6.4.2 Simulation Results

This section presents the results of the simulation case study for a heating control of a six-zone house. To validate the performance of the proposed approximated MPC strategies we have performed closed-loop simulations under the same conditions for all investigated controllers. The dynamical behaviour of the investigated controllers is shown and discussed, together with the comparison of the control performances.

Control Profiles

To demonstrate the dynamical behaviour of the building controlled via different controllers and to verify the tuning, we provide the closed-loop profiles over a representative time window of ten days chosen from the test set. For the clarity, we decide to show the temperature profiles only for the second zone of the building. Fig. 6.12 gives the corresponding disturbance profiles. The RBC profiles with typical switching behaviour of the control action and oscillatory indoor temperature profiles are shown in Fig. 6.13(a). Conservative reference setting in the middle of the comfort bounds (see Section) achieved the highest comfort satisfaction. The control profiles of the well-tuned PID controller with reference tracking behaviour are shown in Fig. 6.13(b). Again we opted here for the reference setting slightly above the lower comfort boundary in order to achieve the highest possible comfort satisfaction. The control profiles of the performance bound MPC are presented in Fig. 6.14(a).

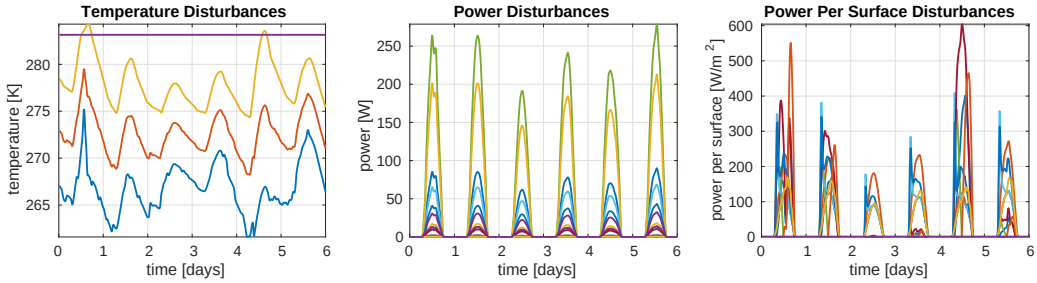
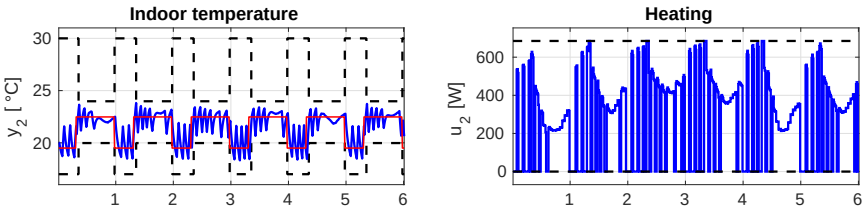


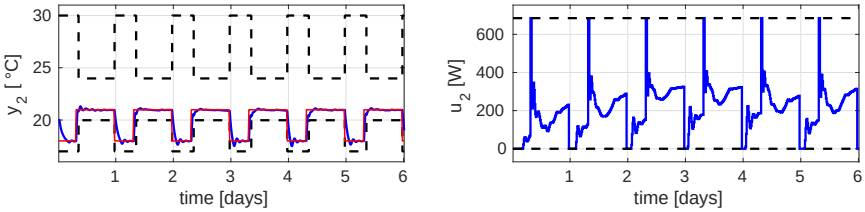
Figure 6.12: 6-days test set disturbance profiles. Left: temperature disturbances, ambient temperature (yellow), ground temperature (purple) and radiation temperatures (red and blue). Middle: solar radiation through and absorbed by each window. Right: solar radiation per surface orientation.

Here we can observe the optimal performance of MPC minimizing the consumed energy by keeping the zone temperature as close as possible to the lower comfort boundary while taking the full disturbances preview into account.

The control profiles of the supervised machine learning controllers trained on two months data approximating MPC actions acting as a 'teacher' are shown in Figs. 6.14(b), 6.14(c) and 6.14(d). First, RT behavior is given in Fig. 6.14(b). The RT are roughly mimicking MPC and can maintain high comfort standard with some energy saving potential in comparison with the RBC. However, in overall, they fail to generalise the learned MPC behaviour well on the new test set data and they perform worse than the group of the PID controllers. The main limitation of the RT is the piecewise constant nature of their approximation, which for the problems of this complexity generates overly deep trees with too many branches and nodes, which are hard to tune. The second limiting factor of the RT is their single variate nature. Because of that, a single RT controller needs to be constructed for each zone separately, not allowing to capture the coupling between the individual zones. Not taking into account the heat gains from the adjacent zones causes the increase of the temperature in some zones, which is the reason for the additional increase in the energy consumption by the RT controllers. On the other hand, the deep TDNN can mimic the MPC actions for all controlled zones with limited complexity and good generalisation with new data. The control profiles of the TDNN are shown in Fig. 6.14(c). However, because the TDNN tries to mimic the optimal MPC behaviour by keeping the temperatures of the zones closer to the lower boundary, the smaller violations of the comfort bounds occur due to the approximation errors. In order to improve the performance of the TDNN, simple heuristic rules from Section 6.4.1 were implemented, slightly adjusting the control actions generated by the TDNN. As shown in Fig. 6.14(d), the enhanced TDNN with heuristic are keeping the room temperatures closer to the lower boundaries with only occasional violations, which results in the improved performance for both, the comfort as well as the energy consumption.

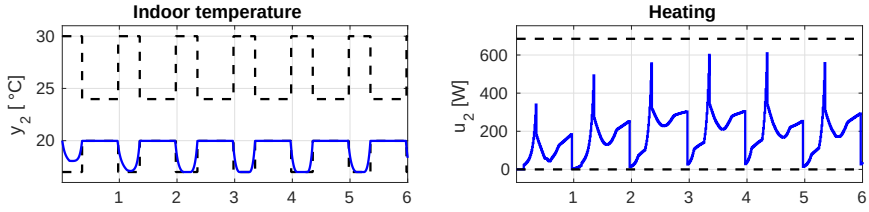


(a) Control profiles of the RBC.

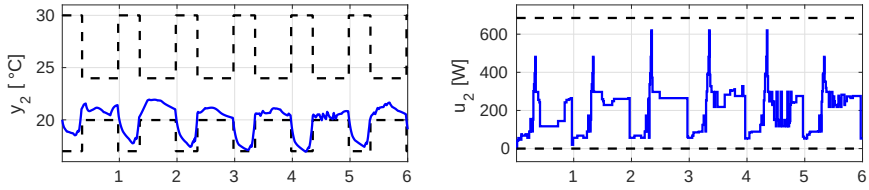


(b) Control profiles of the PID.

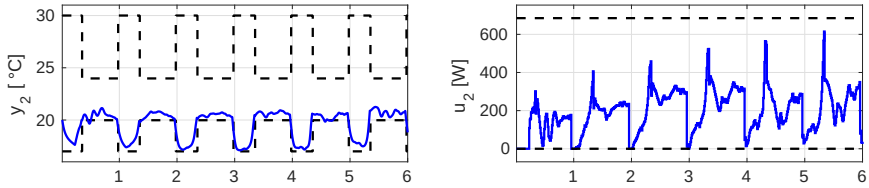
Figure 6.13: Single zone 6-days test set control profiles of the traditional controllers. Right column: closed-loop response of the indoor temperature (blue) in the second building zone w.r.t. the reference (red) and the comfort constraints (black). Left column: corresponding profile of the control action (blue) w.r.t. the control boundaries (black).



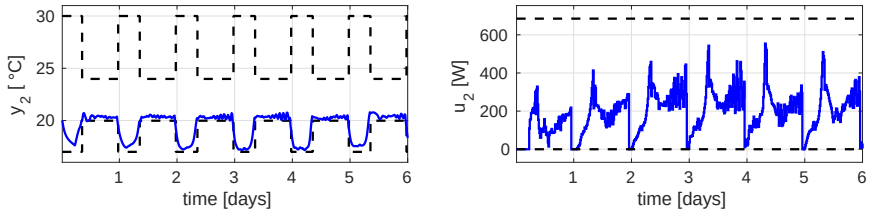
(a) Control profiles of the MPC.



(b) Control profiles of the RT.



(c) Control profiles of the TDNN.



(d) Control profiles of the TDNN enhanced with heuristics.

Figure 6.14: Single zone 6-days test set control profiles of the MPC-based controllers. Right column: closed-loop response of the indoor temperature (blue) in the second building zone w.r.t. the reference (red) and the comfort constraints (black). Left column: corresponding profile of the control action (blue) w.r.t. the control boundaries (black).

Performance Comparison

The overall simulation performances of the investigated controllers on 30-days test set based on the criteria given in Section 6.3.2 are compared in Table 6.3. The comparison w.r.t. the energy consumption and comfort satisfaction is also compactly represented in Fig. 6.15, here the best performance is in lower right corner. All designed controllers can

Method	**Comfort [%]**	**Energy savings [%]**	**CPU time [ms]**
RBC 5.2.1	98.15	-	0.13
PID 5.2	100.00	7.62	0.14
MPC (Gurobi) 5.3	100.00	15.89	15.19
RT 5.4.4	98.23	4.10	3.70
TDNN 5.4.5	95.14	11.64	2.01
TDNN 5.4.5 and Heuristics 6.4.1	97.54	12.82	2.13

Table 6.3: 30-days test set performance comparison, evaluated w.r.t. pefrormance criteria 6.3.2.

achieve high comfort satisfaction performance; this comes however with different costs on the energy consumption. With regards to the energy savings, the RBC is taken here as a benchmark controller, then the PID energy savings potential is equal to 7.62%, while the original MPC acting as a performance bound (PB) scores close to 16%. Even though, the RT-based control policy is able to improve the energy consumption of the RBC by 4%, it scores worse than well-tuned PID, which makes RT less attractive in practice due to higher construction time and tuning effort in comparison with the PID. Moreover, the evaluation time of the RT is almost twice as big as for the TDNN. The TDNN is outperforming both RBC and PIDs in the energy consumption; w.r.t RBC by 11.6% and 12.8%, w.r.t PID by 4.0% and 5.2%, before and after the enhancement of the control laws, respectively. In comparison with the PB MPC, the energy savings of the TDNN drop only by 3%, this accounts roughly for 80.7% restoration of the optimal MPC performance, with just a small increase in the discomfort equal to 2.5%. The biggest difference between machine learning controllers and the MPC feedback, though, is in the implementation cost. While the MPC policy requires solving (5.8) at each sampling instant, the TDNN-based controller only needs to evaluate the function in (5.38). Hence, the implementation of TDNN is reducing the computational effort imposed by the PB MPC roughly by the factor of 7. Moreover, the advanced software libraries are no longer required, and the representation of the TDNN can be easily implemented also on low-level hardware with low memory footprint[2], which makes this approach easily applicable on typical building automation hardware, such as on programmable logic controllers.

[2]The actual size of the TDNN used in this study, implemented as MATLAB functions spanned from 100 KB to 1 MB.

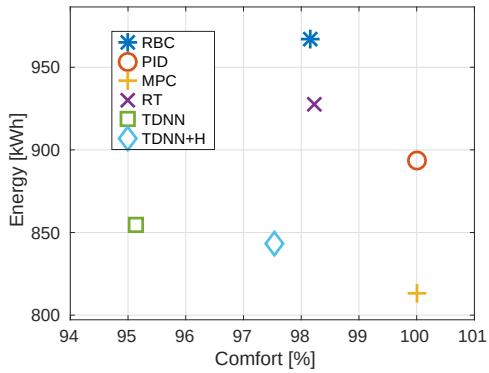


Figure 6.15: 30-days Test Set Performance Comparison.

6.4.3 Summary

This case study showed how to design and evaluate the simple well-performing approximations of the optimization-based controllers by using advanced machine learning algorithms. The focus was given on the complex multi-zone building control problems, employing multiple continuous control inputs. It is important to note, that even though the problems of multivariate time series approximation are particularly challenging, there exist powerful tools which are frequently used in different domains, e.g. pattern recognition, weather and trajectory forecasting, etc.. Despite this fact, these methods received so far only little interest in the context of the building control applications.

The method presented in this work is based on the multivariate regression algorithms, namely deep time delay neural networks (TDNN) and set of regression trees (RT). The regression is used to mimic the complex behaviour, in our case the model predictive controller (MPC), based on the simulation data. A simple tunable feature selection (FS) method introduced in Section 5.4.6 was further used in order to decrease the dimensionality of the parametric space, hence reducing the complexity of the devised controller and reducing the implementation cost of the sensory equipment. This selection was based particularly on the prior engineering knowledge of the system, the principal component analysis (PCA) algorithm and the dynamical analysis of the building's model.

The approach was demonstrated on a temperature control of a six-zone building (see Section 4.3), described by the linear model with 286 states and 42 disturbances. The results from a three months simulation case study showed that the TDNN approximations were able to maintain high comfort and energy savings, with just a small loss of the performance compared to the original MPC, while drastically reducing the complexity of the solution. Moreover, the TDNN controllers showed a good generalization capabilities with respect to new data. The only requirement for a good generalization is that the new measurements have the same probability distribution as the training data. This fact provides a valuable clue for the selection of the representative training data set, which can be chosen based on the historical weather profiles for particular location and season. Please note that the

selection of the most relevant training data significantly reduces the training time of the TDNN. The complete coverage of all operating conditions would be practically infeasible due to the high dimensionality of the parametric space. To further enhance the TDNN controllers, a simple heuristic rules were introduces slightly improving the performance in both terms, comfort as well as energy consumption. The set of RT, however, scored worse, even though they outperformed the traditional RBC, their performance was worse than a well-tuned set of PID controllers. The reason behind this is twofold. First, the piece-wise constant nature of the RT does not allow them to mimic more complex control laws with high precision, and second, their single-variate nature does not take into account the dynamical coupling between the zones. Moreover, because the deep branching of the RT, which was used in this work, the memory footprint of the RTs was roughly fifteen times higher in comparison with the TDNN.

Chapter 7

Conclusions

No one can be a great thinker who does not recognize that as a thinker it is his first duty to follow his intellect to whatever conclusions it may lead.

John Stuart Mill

Main aim of this thesis was investigation of the different mathematical formulations of the model predictive control (MPC), and their applications in building automation system (BAS), in particular for control of heating, ventilation and air conditioning (HVAC) systems.

For this purpose a supporting mathematical background was summarized in Chapter 2. Section 2.1 defined several notions on sets and functions, necessary for the introduction of mathematical optimization in Section 2.2, followed by brief overview of the probability and statistics in Section 2.3. Chapter 3 acts as an introduction for the general MPC technology. Chapter 4 deals with the building modeling as a necessary component for synthesis of MPC formulations. Overall building control concepts are subject of study in Chapter 5. For all control strategies proposed in this thesis a uniform control objectives are considered in Section 5.1. The different MPC formulations for HVAC control problems are proposed in Section 5.3.

All results from the simulation case studies dealing with the building thermal comfort control problems are demonstrated in Chapter 6. Here, Section 6.1 investigated the performance of the explicit stochastic MPC in comparison with the conservative worst-case, and idealistic best-case scenarios. We can conclude that proposed explicit stochastic MPC controller was able to maintain almost similar energy consumption demands, with lost nor even 3% of thermal comfort in comparison with unrealistic best-case MPC controller with given full knowledge of future disturbances over the whole prediction horizon. The fundamental limitation of explicit MPC solution, however, is that the complexity of the computed control law grows exponentially with the dimensionality of the parametric space imposed by prediction horizon and number of variables. Therefore it can be applied only

on the hardware with storage capacity large enough to accommodate the solution . However, this is usually not a realistic assumption, since the size of the explicit MPC solutions can easily exceed several megabytes even for the systems with low complexity, making it infeasible for complex building control problems with several thousand parameters.

Section 6.2.2 evaluated the performance of the enhanced regression trees, introduced in the Section 5.4.4, used for the approximation of the MPC laws on the building control problem with modest complexity. The low memory storage, combined with the low implementation effort, overcomes the drawbacks of the explicit MPC solutions and make the proposed regression-based feedback strategy easily applicable also on a typical building automation hardware, such as on programmable logic controllers. However the cost to be paid are the sub-optimality of the devised solution and loosing the guarantees on closed-loop behaviour, such as stability and output constraints satisfaction. However, as pointed out in Section 5.3.4 the buildings can be conceived as a complex, but inherently stable systems with slow dynamics, which means that one does not need to explicitly account for the closed-loop stability.

Section 6.3 systematically investigated the required controller model complexity necessary to obtained the optimal control performance for a given building. The number of necessary states decreased with the level of insulation of the building and it increased with its mass content. In the case of the investigated six-rooms house, a minimum of 30 states was necessary to obtain the optimal control performance which is significantly higher than the typical orders used by black- or grey-box system identification techniques. The minimum number of states might be chosen lower when offset-free MPC (OSF-MPC) is used instead of the conventional MPC. However, OSF-MPC might significantly increase the energy use when poor controller models (high plant-model mismatch) are used.

Section 6.4 demonstrated the practical aspects of the automatic procedure for the synthesis of the near-optimal MPC-like controllers with low-complexity also on the complex building control problems with high number of the parameters. The advantage of this approach is that the devised controllers are readily applicable to low-level hardware to control the building in real time, without the need for the advanced software libraries. The methodology 5.4.1 is versatile and applicable to any building controlled by an arbitrary optimization-based control algorithm. In comparison with the performance bound MPC, the energy savings of the TDNN drop only by 3%, this accounts roughly for 80.7% restoration of the optimal MPC performance, with just a small increase in the discomfort equal to 2.5%. Moreover, TDNN-based controllers trained on a limited dataset of two months showed a good generalization on new data, which makes the use of these type of control strategies even more promising for the practical applications.

This kind of studies act like the intermediate steps towards more advanced data-driven low-complexity well-performing controllers in the future. Particularly, the vision of the self-constructing and self-tuning near-optimal controllers with easy plug and play implementation is very appealing. The possible direction in minimizing the commissioning effort and cost could be combining the black- or grey-box system identification techniques with automatic MPC design and tuning, together with the automated synthesis of simplified controllers. The challenge here emanates from a complex task of hyper-parameters setting

and iterative nature of the training and tuning process for the machine learning models. However, with extensive research effort nowadays in the field of hyper-parameter optimization and machine learning in general, the efficient tools for handling these challenges can soon be expected and implemented also in the building sector. Another disadvantage to being tackled in the future is the lack of the analytical closed-loop performance guarantees concerning stability and constraints satisfaction. Although, this appears to be only the minor issue for non safety-critical applications like building control.

Bibliography

H. Abdi and L. J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010. ISSN 1939-0068. doi:10.1002/wics.101. URL http://dx.doi.org/10.1002/wics.101. 89

R. A. Abrams and A. Ben Israel. A duality theorem for complex quadratic programming. *Journal of Optimization Theory and Applications*, 4(4):245–252, 1969. 21

C. S. Adjiman, I. P. Androulakis, C. D. Maranas, and C. A. Floudas. A global optimisation method, alfa bb for process design. *Computers and Chemical Engineering*, 1996. 23

C. Aghemo, J. Virgone, G. V. Fracastoro, A. Pellegrino, L. Blaso, J. Savoyat, and K. Johannes. Management and monitoring of public buildings through ICT based systems: Control rules for energy saving with lighting and HVAC services. *Frontiers of Architectural Research*, 2(2):147 – 161, 2013. ISSN 2095-2635. 2

Shmuel Agmon. The relaxation method for linear inequalities. *Canadian Journal of Mathematics 6: 382-392, doi:10.4153/CJM-1954-037-2*, 1954. 22

T. Alamo, R. Tempo, and A. Luque. On the sample complexity of randomized approaches to the analysis and design under uncertainty. In *American Control Conference (ACC), 2010*, pages 4671–4676. IEEE, 2010. 71

A. Alessio and A. Bemporad. *A Survey on Explicit Model Predictive Control, in Nonlinear Model Predictive Control: Towards New Challenging Applications*, pages 345–369. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-01094-1. 3

F. Allgower and A. Zheng. *Nonlinear model predictive control, progress in systems and control theory.* Vol. 26. Basel, Boston, Berlin: Birkhauser Verlag, 2000. 47

F. Allgower, T. A. Badgwell, S. J. Qin, J. B. Rawlings, and S. J. Wright. Nonlinear predictive control and moving horizon estimation-an introductory overview. *Advances in control: highlights of ECC, Berlin: Springer*, 1999. 47

B. Anderson, A. Bennick, and M. Salciccioli. University of michigan chemical engineering process dynamics and controls open textbook, 2014. 33

M. R. Anderson, D. Antenucci, V. Bittorf, M. Burgess, M. J. Cafarella, A. Kumar, F. Niu, Y. Park, Ch. Ré, and Ce Zhang. Brainwash: A data system for feature engineering. In *CIDR*, 2013. 87

A. Antoulas. *Approximation of Large-Scale Dynamical Systems*. Society for Industrial and Applied Mathematics, 2005. doi:10.1137/1.9780898718713. 58, 59

A. Antoulas and D. C. Sorensen. Approximation of large-scale dynamical systems: An overview. *Applied Mathematics and Computer Science*, 11(5):1093–1122, 2001. 58

K. J. Astrom and R. M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers.* Princeton University Press, 2012. 34

M. Avriel. *Nonlinear Programming: Analysis and Methods.* Dover Publications. ISBN 0-486-43227-0, 2003. 22

P. Bacher and H. Madsen. Identifying suitable models for the heat dynamics of buildings. *Energy and Buildings*, 43(7):1511–1522, 2011. 51, 52

R. Baetens. *On externalities of heat pump-based low-energy dwellings at the low-voltage distribution grid*. PhD thesis, KU Leuven, Belgium, 2015. 56

R. Baetens, R. De Coninck, F. Jorissen, D. Picard, L. Helsen, and D. Saelens. OpenIDEAS - An open framework for integrated district energy simulations. In *IPSBA Building simulation 2015*, Hyderabad, 2015. 53, 55, 60, 75

S. Baldi, I. Michailidis, Ch. Ravanis, and E. B. Kosmatopoulos. Model-based and model-free "plug-and-play" building energy efficient control. *Applied Energy*, 154:829 – 841, 2015. ISSN 0306-2619. 3

E. M. L. Beale and R. Benveniste. Quadratic programming. In *Design and Implementation of Optimization Software*, pages 249–258. Sijthoff and Noordhoff, Alphen aan den Rijn, Netherlands, 1978. 21

W. Beckman, L. Broman, A. Fiksel, S. Klein, E. Lindberg, M. Schuler, and J. Thornton. TRNSYS: The most complete solar energy system modeling and simulation software. *Renewable energy*, 5(1):486–488, 1994. 53

P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and Mahajan A. Mixed-integer nonlinear optimization. *Acta Numerica 22:1-131.*, 2013. 23

A. Bemporad and C. Filippi. Suboptimal explicit mpc via approximate multiparametric quadratic programming. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*, volume 5, pages 4851–4856 vol.5, 2001. doi:10.1109/.2001.980975. 3

A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999. 84

A. Bemporad, Morari. M., V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3 – 20, 2002. ISSN 0005-1098. doi:10.1016/S0005-1098(01)00174-1. 3, 46

A. Bemporad, A. Oliveri, T. Poggi, and M. Storace. Synthesis of stabilizing model predictive controllers via canonical piecewise affine approximations. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5296–5301, Dec 2010. doi:10.1109/CDC.2010.5717868. 3

M. Berger. *Geometry I*. Berlin: Springer, ISBN 3-540-11658-3, 1987. 9

W. Bernal, M. Behl, T. Nghiem, and R. Mangharam. MLE+: a tool for integrated design and deployment of energy efficient building controls. In *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 123–130. ACM, 2012. 53

D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, 1996. ISBN 1-886529–04-3. 15

M. J. Best and J. Kale. Quadratic programming for large-scale portfolio optimization. In J. Keyes, editor, *Financial Services Information Systems*, pages 513–529. CRC Press, 2000. 21

F. Borrelli. *Constrained Optimal Control of Linear and Hybrid Systems*, volume 290. Springer-Verlag, 2003. 3, 46, 47

F. Borrelli, A. Bemporad, and M. Morari. *Predictive Control for linear and hybrid systems*. Model Predictive Control Lab, UC-Berkeley, 2014. 24

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 14, 16, 22

L. Breiman. *Classification and regression trees*. CRC press, 1993. 81

E. Busseti, I. Osband, and S. Wong. Deep learning for time series modeling. Technical report, 2012. 86

E. F. Camacho and A. C. Bordons. *Model predictive control*. Springer Verlag, 2004. ISBN 978-0-85729-398-5. 47

M. Campi and S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization*, 19(3):1211–1230, 2008. 69, 71

C. Canbay, A. Hepbasli, and G. Gokcen. Evaluating performance indices of a shopping centre and implementing HVAC control principles to minimize energy usage. *Energy and Buildings*, 36(6):587–598, 2004. 1

R. L. Casella, G.and Berger. *Statistical Inference (2nd ed.)*. Duxbury, 2001. ISBN 0-534-24312-6. 32

M. del Mar Castilla, J. D. Álvarez, F. de A. Rodriguez, and M. Berenguel. *Comfort Control in Buildings*. Springer-Verlag London, 2014. doi:10.1007/978-1-4471-6347-3. 1

J. Cigler, S. Prívara, Z. Váňa, E. Žáčeková, and L. Ferkl. Optimization of predicted mean vote index within model predictive control framework: Computationally tractable solution. *Energy and Buildings*, 52, 2012. 79

J. Cigler, D. Gyalistras, J. Široký, V. Tiet, and L. Ferkl. Beyond Theory: the Challenge of Implementing Model Predictive Control in Buildings. In *Proceedings of 11th Rehva World Congress, Clima*, 2013. 2

B. Coffey. Approximating model predictive control with existing building simulation tools and offline optimization. *Journal of Building Performance Simulation*, 6(3):220–235, 2013. doi:10.1080/19401493.2012.737834. 3, 78

D. Crawley, L. Lawrie, F. Winkelmann, W. Buhl, Y. Huang, C. Pedersen, R. Strand, R. Liesen, D. Fisher, and M. Witte. EnergyPlus: creating a new-generation building energy simulation program. *Energy and Buildings*, 33(4):319–331, 2001. 53

C. R. Cutler and B. L. Ramaker. Dynamic matrix control—a computer control algorithm. *AICHE national meeting, Houston, TX.*, 1979. 40

C. R. Cutler and B. L. Ramaker. Dynamic matrix control—a computer control algorithm. *In Proceedings of the joint automatic control conference.*, 1980. 40

C. R. Cutler, A. Morshedi, and J. Haydel. An industrial perspective on advanced control. *In AICHE annual meeting, Washington, DC.*, 1983. 41

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989. ISSN 1435-568X. doi:10.1007/BF02551274. 85

S. K. Das. Feature selection with a linear dependence measure. *IEEE Transactions on Computers*, 20(9):1106–1109, September 1971. ISSN 0018-9340. doi:10.1109/T-C.1971.223412. URL http://dx.doi.org/10.1109/T-C.1971.223412. 88

P. L. De Angelis, P. M. Pardalos, and G. Toraldo. Quadratic programming with box constraints. In I. M. Bomze, editor, *Developments in Global optimization*, pages 73–93, Dordrecht, The Netherlands, 1997. Kluwer Academic Publishers. 21

Roel De Coninck. *Grey-Box Based Optimal Control for Thermal Systems in Buildings-Unlocking Energy Efficiency and Flexibility.* PhD thesis, KU Leuven, Belgium, 2015. 52

M. Dlouhy, J. Fabry, and M. Kuncova. *Simulace pro ekonomy.* Praha: VŠE, 2005. 31

Robert H. Dodier and Gregor P. Henze. Statistical analysis of neural networks as applied to building energy prediction. *Journal of Solar Energy Engineering*, 126(1):592–600, 2004. doi:doi: 10.1115/1.1637640. 87

A. Domahidi, M. N. Zeilinger, M. Morari, and C. N. Jones. Learning a feasible and stabilizing explicit model predictive control law by robust optimization. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 513–519, Dec 2011. doi:10.1109/CDC.2011.6161258. 3

A. Domahidi, F. Ullmann, M. Morari, and Jones C. N. Learning decision rules for energy efficient building control. *Journal of Process Control*, 24(6):763 – 772, 2014. ISSN 0959-1524. doi:10.1016/j.jprocont.2014.01.006. Energy Efficient Buildings Special Issue. 2, 3

P. Domingos. A few useful things to know about machine learning. *Commun. ACM*, 55 (10):78–87, October 2012. ISSN 0001-0782. doi:10.1145/2347736.2347755. 87

A.I. Dounis and C. Caraiscos. Advanced control systems engineering for energy and comfort management in a building environment-a review. *Renewable and Sustainable Energy Reviews*, 13(6-7):1246–1261, 2009. doi:10.1016/j.rser.2008.09.015. 1

US Energy. Building energy software tools directory, 2014. URL http://apps1.eere.energy.gov/buildings/tools_directory/. 53

D. F. Enns. Model reduction with balanced realizations: An error bound and a frequency weighted generalization. In *Decision and Control, 1984. The 23rd IEEE Conference on*, pages 127–132, Dec 1984. doi:10.1109/CDC.1984.272286. 59

EUROSIS. Directory of simulation software and tools, 2014. URL http://www.eurosis.org/cms/?q=node/1318. 53

L. Ferkl and J. Široký. Ceiling radiant cooling: Comparison of armax and subspace identification modelling methods. *Building and Environment*, 45(1):205 – 212, 2010. ISSN 0360-1323. doi:https://doi.org/10.1016/j.buildenv.2009.06.004. 51

A. Foucquier, A. Brun, G. A. Faggianelli, and F. Suard. Effect of wall merging on a simplified building energy model: accuracy vs number of equations. In *Proceedings of 13th Conference of International Building Performance Simulation Association*, Chambéry, France, August 2013. 52

G. Fraisse, Ch. Viardot, O. Lafabrie, and G. Achard. Development of a simplified and accurate building model based on electrical analogy. *Energy and Buildings*, 34(10):1017–1031, 2002. 52

R. Freire, G. Oliveira, and N. Mendes. Thermal comfort based predictive controllers for building heating systems. In *Proc. of the 16th IFAC World Congress*, 2005. 79

G. Frison and J.B. Jorgensen. A fast condensing method for solution of linear-quadratic control problems. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 7715–7720, Dec 2013. doi:10.1109/CDC.2013.6761114. 67

B. Fristedt and L. Gray. *A modern approach to probability theory.* Boston: Birkhäuser. ISBN 3-7643-3807-5., 1996. 29

J. B. Froisy and T. Matsko. Idcom-m application to the shell fundamental control problem. *AICHE annual meeting*, 1990. 42

A. Gallini. Affine function, 2014. URL http://mathworld.wolfram.com/AffineFunction.html. 12

C. E. García, D. M. Prett, and M. Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335 – 348, 1989. ISSN 0005-1098. doi:10.1016/0005-1098(89)90002-2. 39, 47

C.E. Garcia and Morshedi A.M. Quadratic programming solution of dynamic matrix control (qdmc). *Chemical Engineering Communications.*, pages 73–87, 1986. 41

A. Gelman. Exploratory data analysis for complex models. In *Journal of Computational and Graphical Statistics, 13(4), 755–779.*, 2004. 27

G.C. Goodwin, S.F. Graebe, and M.E. Salgado. Control system design. *Prentice Hall, Englewood Cliffs, NJ.*, 2001. 39

T. Gorecki, F. Qureshi, and C. N. Jones. Openbuild: An integrated simulation environment for building control. In *2015 IEEE Conference on Control Applications (CCA)*, pages 1522–1527. IEEE, 2015. 51, 52, 53

M. M. Gouda, S. Danaher, and C. P. Underwood. Building thermal model reduction using nonlinear constrained optimization. *Building and Environment*, 37(12):1255–1265, 2002. 52

A. Grancharova, J. Kocijan, and T.A. Johansen. Explicit stochastic predictive control of combustion plants based on gaussian process models. *Automatica*, 44(6):1621–1631, 2008. 68

P. Grieder and M. Morari. Complexity reduction of receding horizon control. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, volume 3, pages 3179–3190 Vol.3, Dec 2003. doi:10.1109/CDC.2003.1273112. 3

P. Grosdidier, B. Froisy, and M. Hammann. The idcom-m controller. *Proceedings of the 1988 IFAC workshop on model based process control. Oxford: Pergamon Press.*, pages 31–36, 1988. 42

B. Grunbaum. *Convex Polytopes.* Springer-Verlag, second edition,, 2000. 9

Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2012. URL http://www.gurobi.com. 23, 84, 107

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, March 2003. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=944919.944968. 87

D. Gyalistras, M. Gwerder, F. Schildbach, C.N. Jones, M. Morari, B. Lehmann, K. Wirth, and V. Stauch. Analysis of Energy Savings Potentials for Integrated Room Automation. In *Clima - RHEVA World Congress*, Antalya, Turkey, May 2010. 1

Dimitrios Gyalistras and Markus Gwerder. Use of weather and occupancy forecasts for optimal building climate control (opticontrol), two years progress report. Technical report, Terrestrial Systems Ecology ETH Zurich, 2009. 51, 52

Martin T. Hagan, Howard B. Demuth, and Mark Beale. *Neural Network Design*. PWS Publishing Co., Boston, MA, USA, 1996. ISBN 0-534-94332-2. 111

M. Hamdi and G. Lachiver. A fuzzy control system based on the human sensation of thermal comfort. In *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, volume 1, pages 487–492. IEEE, 1998. 1

V.S.K.V. Harish and A. Kumar. Reduced order modeling and parameter identification of a building energy system model through an optimization routine. *Applied Energy*, 162: 1010–1023, 2016. 52

I. Hazyuk, Ch. Ghiaus, and D. Penhouet. Optimal temperature control of intermittently heated buildings using model predictive control: Part i – building modeling. *Building and Environment*, 51(0):379 – 387, 2012. ISSN 0360-1323. doi:10.1016/j.buildenv.2011.11.009. 51

Jeff Heaton. An empirical analysis of feature engineering for predictive modeling. *CoRR*, abs/1701.07852, 2017. URL http://arxiv.org/abs/1701.07852. 87

M. Herceg, M. Kvasnica, C. Jones, and M. Morari. Multi-parametric toolbox 3.0. In *2013 European Control Conference*, pages 502–510, 2013. 95

Trent Hilliard, Miroslava Kavgic, and Lukas Swan. Model predictive control for commercial buildings: trends and opportunities. *Advances in Building Energy Research*, pages 1–19, 2015. 51

G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. ISSN 0036-8075. doi:10.1126/science.1127647. 87

G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006. ISSN 0899-7667.

doi:10.1162/neco.2006.18.7.1527. URL http://dx.doi.org/10.1162/neco.2006.18.7.1527. 86

D. C. Hoaglin, F. Mosteller, and J. W. Tukey. Understanding robust and exploratory data analysis. In *Wiley series in probability and mathematical statistics, New York*, 1983. 27

K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251 – 257, 1991. ISSN 0893-6080. doi:10.1016/0893-6080(91)90009-T. 76, 85

B. Huyck, H. Ferreau, M. Diehl, J. De Brabanter, J. Van Impe, B. De Moor, and F. Logist. Towards online model predictive control on a programmable logic controller: practical considerations. *Mathematical Problems in Engineering*, 2012, 2012. 2

ILOG, Inc. *CPLEX 8.0 User Manual*. Gentilly Cedex, France, 2003. URL http://www.ilog.fr/products/cplex/. 23

INFORMS.org. About operations research, 2014. 24

I.T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986. 89

Olav Kallenberg. *Random Measures (4th ed.)*. Berlin: Akademie Verlag. ISBN 0-12-394960-2. MR MR0854102., 1986. 29

Olav Kallenberg. *Foundations of Modern Probability (2nd ed.)*. Berlin: Springer Verlag. ISBN 0-387-95313-2., 2001. 29

R. E. Kalman. Contributions to the theory of optimal control. *Bulletin de la Societe Mathematique de Mexicana, 5, 102-119.*, 1960a. 38

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of ASME, Journal of Basic Engineering, 87, 35-45.*, 1960b. 38

Donghun Kim and James E Braun. Reduced-order building modeling for application to model-based predictive control. In *Proceedings of the 5th National Conference of IBPSA-USA*, volume 5, pages 554–561, Madison, USA, August 2012. 52

M. Klaučo, J. Drgoňa, M. Kvasnica, and S. Di Cairano. Building temperature control by simple mpc-like feedback laws learned from closed-loop data. *IFAC Proceedings Volumes*, 47(3):581 – 586, 2014. ISSN 1474-6670. doi:10.3182/20140824-6-ZA-1003.01633. 19th IFAC World Congress. 80

D. E. Knuth. *he Art of Computer Programming, Volume 2: Seminumerical Algorithms.* Addison-Wesley, Boston., 1998. 31

A. N. Kolmogorov. *Grundbegriffe der Wahrscheilichkeitsrechnung*. Springer, Berlin, 1933. 28

B. Kouvaritakis and M. Cannon. *Nonlinear predictive control, theory and practice*. IET Digital Library, 2001. ISBN 978-0-85296-984-7. 47

Rick Kramer, Jos Van Schijndel, and Henk Schellen. Simplified thermal and hygric building models: a literature review. *Frontiers of Architectural Research*, 1(4):318–325, 2012. 52

W. J. Krzanowski. Selection of variables to preserve multivariate data structure, using principal components. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 36(1):22–33, 1987. ISSN 00359254, 14679876. URL http://www.jstor.org/stable/2347842. 89

A. Kusiak and G. Xu. Modeling and optimization of HVAC systems using a dynamic neural network. *Energy*, 2012. 1

M. Kvasnica, P. Grieder, M. Baotic, and M. Morari. Multi-Parametric Toolbox (MPT). In *Hybrid Systems: Computation and Control*, pages 448–462, March 2004. 47

M. Kvasnica, J. Löfberg, M. Herceg, L. Čirka, and M. Fikar. Low-complexity polynomial approximation of explicit mpc via linear programming. In *Proceedings of the 2010 American Control Conference*, pages 4713–4718, June 2010a. doi:10.1109/ACC.2010.5531092. 3

M. Kvasnica, I. Rauová, and M. Fikar. Automatic code generation for real-time implementation of model predictive control. In *Proceedings of the 2010 IEEE International Symposium on Computer-Aided Control System Design*, pages 993–998, 2010b. 3

Khang Le, Romain Bourdais, and Hervé Guéguen. From hybrid model predictive control to logical control for shading system: A support vector machine approach. *Energy and Buildings*, 84:352 – 359, 2014. ISSN 0378-7788. doi:10.1016/j.enbuild.2014.07.084. 3

Quoc Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg Corrado, Jeff Dean, and Andrew Ng. Building high-level features using large scale unsupervised learning. In *International Conference in Machine Learning*, 2012. 87

B. Lehmann, D. Gyalistras, M. Gwerder, K. Wirth, and S. Carl. Intermediate complexity model for model predictive control of integrated room automation. *Energy and Buildings*, 58(0):250 – 262, 2013. 51, 52

Geoff J. Levermore. *Building Energy Management Systems: Applications to low-energy HVAC and natural ventilation control*. E & FN Spon, 2 edition, 2000. 1

Xiwang Li and Jin Wen. Review of building energy modeling for control and operation. *Renewable and Sustainable Energy Reviews*, 37:517–537, 2014. 2, 51, 52

J. T. Linderoth and T. K. Ralphs. Noncommercial software for mixed-integer linear programming. *RC Press Operations Research Series*, 2005. 23

S. Liu and G. P. Henze. Evaluation of reinforcement learning for optimal control of building active and passive thermal storage inventory. *Journal of Solar Energy Engineering*, 129 (2):215–225, 2007. 1

J. Löfberg. YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In *Proc. of the CACSD Conference*, Taipei, Taiwan, 2004. Available from http://users.isy.liu.se/johanl/yalmip/. 95, 106

Y. Lu, I. Cohen, X. S. Zhou, and Qi Tian. Feature selection using principal feature analysis. In *Proceedings of the 15th ACM International Conference on Multimedia*, MM '07, pages 301–304, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-702-5. doi:10.1145/1291233.1291297. 89

A. Lyon. Why are normal distributions normal? *The British Journal for the Philosophy of Science.*, 2014. 32

Y. Ma, F. Borrelli, B. Hencey, A. Packard, and S. Bortoff. Model predictive control of thermal energy storage in building cooling systems. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 392 –397, 2009. doi:10.1109/CDC.2009.5400677. 1

Y. Ma, G. Anderson, and F. Borrelli. A distributed predictive control approach to building temperature regulation. In *American Control Conference (ACC), 2011*, pages 2089–2094. IEEE, 2011. 2

Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves. Model predictive control for the operation of building cooling systems. *Control Systems Technology, IEEE Transactions on*, 20(3):796–803, 2012a. 2, 79

Y. Ma, S. Vichik, and F. Borrelli. Fast stochastic MPC with optimal risk allocation applied to building control systems. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 7559–7564. IEEE, 2012b. 2

J. M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2002. ISBN 0-201-39823-0. 2, 47

Door Frédéric Magoules and Hai-Xiang Zhao. *Data Mining and Machine Learning in Building Energy Analysis*. John Wiley and Sons, 2016. ISBN 978-1-84821-422-4. 87

P. Marquis and J. P. Broustail. Smoc, a bridge between state space and model predictive controllers: Application to the automation of a hydrotreating unit. *Proceedings of the 1988 IFAC workshop on model based process control, Oxford: Pergamon Press.*, pages 37–43, 1998. 42

Mathworks. Matlab optimization toolbox documentation: writing constraints, 2014. URL http://www.mathworks.com/help/optim/ug/writing-constraints.html. 16

P. May-Ostendorp, G. P. Henze, Ch. D. Corbin, B. Rajagopalan, and C. Felsmann. Model-predictive control of mixed-mode buildings with rule extraction. *Building and Environment*, 46(2):428 – 437, 2011. ISSN 0360-1323. doi:10.1016/j.buildenv.2010.08.004. 3, 78

D. Q. Mayne. Nonlinear model predictive control: An assessment. *Fifth international conference on chemical process control AICHE and CACHE,*, pages 217–231, 1997. 47

D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967 – 2986, 2014. ISSN 0005-1098. doi:10.1016/j.automatica.2014.10.128. 47

D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica.*, 36:789–814, 2000. 47

F. C. McQuiston, J. D. Parker, and J. D. Spitler. *Heating, Ventilating and Air Conditioning Analysis and Design*. Wiley, 6 edition, 2005. 1

Houcem Eddine Mechri, Alfonso Capozzoli, and Vincenzo Corrado. Use of the ANOVA approach for sensitive building energy design . *Applied Energy*, 87(10):3073 – 3083, 2010. ISSN 0306-2619. doi:10.1016/j.apenergy.2010.04.001. 2

Meteotest. *METEONORM Version 6.1 - Edition 2009*, 2009. 57, 106

M. Morari and J. H. Lee. Model predictive control: The good, the bad, and the ugly. *Chemical process control - CPC IV, Fourth international conference on chemical process control*, pages 419–444, 1991. 47

K. R. Muske and T. A. Badgwell. Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, 12(5):617 – 632, 2002. ISSN 0959-1524. doi:10.1016/S0959-1524(01)00051-8. 66

K. R. Muske and J. B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262–287, 1993. ISSN 1547-5905. 47

G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley and Sons, New York., 1988. 21

NEOS. University of wisconsin-madison, 2014. URL http://neos-guide.org/. xi, 15

NEOS-algorithms. University of wisconsin-madison, 2014. URL http://neos-guide.org/algorithms. 15

NEOS-software. University of wisconsin-madison, 2014. URL http://neos-guide.org/solver-software. 15, 23

NEOS-solvers. University of wisconsin-madison, 2014. URL http://www.neos-server.org/neos/solvers/index.html. 15, 23

T. X. Nghiem. Green buildings: Optimization and adaptation, 2011. URL http://www.seas.upenn.edu/~cis800/software.html. 53

F. Oldewurtel, A. Parisio, C.N. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth. Energy efficient building climate control using stochastic model predictive control and weather predictions. In *American Control Conference (ACC), 2010*, pages 5100–5105. IEEE, 2010. 2, 79

F. Oldewurtel, A. Parisio, C.N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari. Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy and Buildings*, 45:15–27, 2012. 2

G. Pannocchia and J. B. Rawlings. Disturbance Models for Offset-Free Model-Predictive Control. *AIChE Journal*, 49(2):426–437, 2003. 66

G. Pannocchia, J. B. Rawlings, and S. J. Wright. Fast, large-scale model predictive control by partial enumeration. *Automatica*, 43(5):852 – 860, 2007. ISSN 0005-1098. doi:10.1016/j.automatica.2006.10.019. 3

A. Papoulis and S. U. Pillai. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill Science/Engineering/Math. ISBN 0-07-281725-9., 2001. 29, 30

T. Parisini and R. Zoppoli. A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica*, 31(10):1443 – 1451, 1995. ISSN 0005-1098. doi:10.1016/0005-1098(95)00044-W. 3

M. Parry, O. Canziani, J. Palutikof, P. van der Linden, and C. Hanson. *Climate change 2007: impacts, adaptation and vulnerability.* Intergovernmental Panel on Climate Change, 2007. 1

D. Picard, F. Jorissen, and L. Helsen. Methodology for Obtaining Linear State Space Building Energy Simulation Models. In *11th International Modelica Conference*, pages 51–58, Paris, 2015. 51, 52, 56, 57, 75, 105

D. Picard, M. Sourbron, F. Jorissen, Z. Váňa, J Cigler, L. Ferkl, and L. Helsen. Comparison of model predictive control performance using grey-box and white-box controller models of a multi-zone office building. In *4th International High Performance Buildings Conference*, Purdue, 2016. 52

D. M. Prett and C. E. Garcia. Quadratic programming solution of dynamic matrix control (qdmc). *Fundamental process control Butterworths, Boston*, 1988. 42

D. M. Prett and R. D. Gillette. Optimization and constrained multivariable control of a catalytic cracking unit. *In Proceedings of the joint automatic control conference.*, 1980. 40

S. Prívara, Z. Váňa, D. Gyalistras, J. Cigler, C. Sagerschnig, M. Morari, and L. Ferkl. Modeling and identification of a large multi-zone office building. In *2011 IEEE International Conference on Control Applications (CCA)*, pages 55–60, Sept 2011a. doi:10.1109/CCA.2011.6044402. 51

S. Prívara, J. Široký, L. Ferkl, and Jiří Cigler. Model predictive control of a building heating system: The first experience. *Energy and Buildings*, 43(2):564–572, 2011b. 51

S. Prívara, J. Cigler, Z. Váňa, F. Oldewurtel, C. Sagerschnig, and E. Žáčeková. Building modeling as a crucial part for building predictive control. *Energy and Buildings*, 56(0): 8 – 22, 2013. ISSN 0378-7788. doi:10.1016/j.enbuild.2012.10.024. 51, 52, 75, 108

S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733 – 764, 2003. ISSN 0967-0661. doi:10.1016/S0967-0661(02)00186-7. 38, 40

J. B. Rawlings. Tutorial overview of model predictive control. *Control Systems, IEEE*, 20 (3):38–52, Jun 2000. ISSN 1066-033X. doi:10.1109/37.845037. 47

J. B. Rawlings and K. R. Muske. Stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, pages 1512–1516, 1993. 43

J. B. Rawlings, E. S. Meadows, and K. Muske. Nonlinear model predictive control: a tutorial and survey. *In Proceedings of IFAC ADCHEM. Japan*, 1994. 47

Glenn Reynders, Jan Diriken, and Dirk Saelens. Quality of grey-box models and identified parameters as function of the accuracy of input and observation signals. *Energy and Buildings*, 82:263–274, 2014. 51, 52

J. Richalet, A. Rault, J. L. Testud, and J. Papon. Algorithmic control of industrial processes. *In Proceedings of the 4th IFAC symposium on identification and system parameter estimation.*, pages 1119–1167, 1976. 39, 40

J. Richalet, A. Rault, J. L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5):413 – 428, 1978. ISSN 0005-1098. doi:10.1016/0005-1098(78)90001-8. 40

A. Richards and J. How. Mixed-integer programming for control. *In American Control Conference, pages 2676-2683*, 2005. 23

N. L. Ricker. Model predictive control: State of the art. *In Y. Arkun, W. H. Ray (Eds.), Chemical process control - CPC IV, Fourth international conference on chemical process control*, pages 271–296, 1991. 47

J. Roll, A. Bemporad, and L. Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40:37–50, 2004. 84

K. W. Roth, D. Westphalen, J. Dieckmann, S. D. Hamilton, and W. Goetzler. Energy consumption characteristics of commercial building hvac systems - volume iii: Energy savings potential. Technical report, 2002. 2

A.E. Ruano, E.M. Crispim, E.Z.E. Conceicao, and M.M.J.R. Lucio. Prediction of building's temperature using neural networks models. *Energy and Buildings*, 38(6):682–694, 2006. 51

A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis.* The Primer, John Wiley and Sons., 2008. 24

Philip K. Schneider and David H. Eberly. *Geometric Tools for Computer Graphics.* Morgan Kaufmann. ISBN 978-1-55860-594-7., 2003. 9, 11

A. Schrijver. *Linear and Integer Programming,.* John Wiley and Sons, New York., 1984. 21

P.O.M. Scokaert and J.B. Rawlings. Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control.*, pages 1163–1169, 1998. 43

S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control, Analysis and Design.* Willey, 2nd edition, 2007. ISBN 978-0-470-01168-3. 36

F. Song, Z. Guo, and D. Mei. Feature selection using principal component analysis. In *2010 International Conference on System Science, Engineering Design and Manufacturing Informatization*, volume 1, pages 27–30, Nov 2010. doi:10.1109/ICSEM.2010.14. 89

M. Sourbron, C. Verhelst, and L. Helsen. Building models for model predictive control of office buildings with concrete core activation. *Journal of building performance simulation*, 6(3):175–198, 2013. 51

D. Sturzenegger, D. Gyalistras, V. Semeraro, M. Morari, and R. Smith. BRCM Matlab Toolbox: Model Generation for Model Predictive Building Control. In *American Control Conference*, pages 1063–1069, Portland, June 2014. 52

S. Summers, C. N. Jones, J. Lygeros, and M. Morari. A multiscale approximation scheme for explicit model predictive control with stability, feasibility, and performance guarantees. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 6327–6332, Dec 2009. doi:10.1109/CDC.2009.5400583. 3

P. Tøndel, T. A. Johansen, and A. Bemporad. Evaluation of Piecewise Affine Control via Binary Search Tree. *Automatica*, 39(5):945–950, May 2003. 74

M. Trcka and J. L. M. Hensen. Overview of hvac system simulation. *Automation in Construction*, 19(2):93–99, 2010. 53

Mi Ching Tsai and Da Wei Gu. *Robust and Optimal Control. A Two-port Framework Approach.* Springer London, 2014. ISBN 978-1-4471-6256-8. 36

J. W. Tukey. Exploratory data analysis. In *Reading, MA: Addison-Wesley.*, 1977. 27

A. van Schijndel. Integrated heat, air and moisture modeling and simulation in hamlab. In *IEA Annex 41 working meeting, Montreal, May*, 2005. 53, 60

A. Van Schijndel, H. Schellen, J. Wijffelaars, and K. van Zundert. Application of an integrated indoor climate, HVAC and showcase model for the indoor climate performance of a museum. *Energy and Buildings*, 40(4):647–653, 2008. 2

Z. Váňa, J. Cigler, J. Široký, E. Žáčeková, and L. Ferkl. Model-based energy efficient control applied to an office building. *Journal of Process Control*, 24(6):790 – 797, 2014. ISSN 0959-1524. doi:https://doi.org/10.1016/j.jprocont.2014.01.016. Energy Efficient Buildings Special Issue. 2

R. Webster. *Convexity*. Oxford, England: Oxford University Press, 1995. 9

E. W. Weisstein. The web's most extensive mathematics resource, 2014. URL http://mathworld.wolfram.com/. 9

M. Wetter and P. Haves. A modular building controls virtual test bed for the integration of heterogeneous systems. In *Third National Conference of IBPSA-USA, Berkeley/California*, 2008. URL https://gaia.lbl.gov/bcvtb. 53

M. Wetter, W. Zuo, T. S. Nouidui, and X. Pang. Modelica buildings library. *Journal of Building Performance Simulation*, 7(4):253–270, 2014. 53

L.A. Wolsey. *Integer Programming*. John Wiley and Sons, New York., 1998. 21

Xinhua Xu and Shengwei Wang. Optimal simplified thermal models of building envelope based on frequency domain regression using genetic algorithm. *Energy and Buildings*, 39 (5):525–536, 2007. 52

A. Yahiaoui, J.L.M. Hensen, and L. Soethout. Integration of control and building performance simulation software by run-time coupling,. *Proceedings of 8th International IBPSA Conference, International Building Performance Simulation Association, Eindhoven, The Netherlands*, pages 1435–1442, 2003. 53

K. Yun, R. Luck, P. J. Mago, and H. Cho. Building hourly thermal load prediction using an indexed arx model. *Energy and Buildings*, 54:225–233, 2012. 51

K. Zhou and J. C. Doyle. *Essentials of Robust Control*. Prentice Hall, 1997. ISBN 0-13-525833-2. 36

K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1995. ISBN 978-0134565675. 36

Xin Zhou, Tianzhen Hong, and Da Yan. Comparison of building energy modeling programs: Hvac systems. *Building Simulation*, 08/2013 2013. 53

G. M. Ziegler. *Lectures on Polytopes*. Springer, 1994. 84

J. Široký, F. Oldewurtel, J. Cigler, and S. Prívara. Experimental analysis of model predictive control for an energy efficient building heating system. *Applied Energy*, 88(9):3079 – 3087, 2011. ISSN 0306-2619. doi:10.1016/j.apenergy.2011.03.009. 1, 2, 51

Appendix A

Author's Publications

This is a list[1] of scientific publications of the author. The list is structured based on the categorization of Slovak Accreditation Committee[2]. Total number of citations: 7.

Accreditation Category A; ADC - Articles in journals:

1. **Drgoňa, J.**, Klaučo, M., Janeček, F., Kvasnica, M.: Optimal control of a laboratory binary distillation column via regionless explicit MPC. *Computers & Chemical Engineering*, ISSN: 0098-1354, vol. 96, pg. 139–148, 2017. (IF: 2.581)

2. *Picard, D.,* ***Drgoňa, J.****, Helsen, L., Kvasnica, M.: Impact of the Controller Model Complexity on MPC Performance Evaluation for Buildings. Energy and Buildings, 2017. (after 1st review, IF:* 2.973*)*

3. *Holaza, J., Klaučo, M.,* ***Drgoňa, J.****, Oravec, J., Kvasnica, M. and Fikar M.: MPC-Based Reference Governor Control of a Continuous Stirred-Tank Reactor. Computers & Chemical Engineering, 2017. (after 1st review, IF:* 2.581*)*

4. ***Drgoňa, J.****, Picard, D., Helsen, L., Kvasnica, M.: Approximated Model Predictive Control for Complex Building Control Problems via Machine Learning, Applied Energy, 2017. (Submitted, IF:* 5.746*)*

Accreditation Category A; AFC - IFAC World Congress:

5. Klaučo, M., **Drgoňa, J.**, Kvasnica, M., Di Cairano, S.: Building Temperature Control by Simple MPC-like Feedback Laws Learned from Closed-Loop Data. In Preprints of the 19th IFAC World Congress Cape Town (South Africa) August 24 - August 29, 2014, pg. 581–586, 2014.

[1] Items in italics are not yet published, they are either accepted or submitted

[2] A minimum requirement for obtaining the PhD degree is to co-author 1 publication of category A, and 2 publications of category B

Accreditation Category B; AFC - Conference Proceedings:

6. **Drgoňa, J.**, Janeček, F., Klaučo, M., Kvasnica, M.: Regionless Explicit MPC of a Distillation Column. In IEEE 2016 European Control Conference (ECC), Aalborg, Denmark, pp. 1568-1573, 2016. *(personally delivered talk).*

7. **Drgoňa, J.**, Klaučo, M., Kvasnica, M.: MPC-Based Reference Governors for Thermostatically Controlled Residential Buildings. In IEEE Conference on Decision and Control (CDC), Osaka, Japan, vol. 54, pp. 1334-1339, 2015. *(personally delivered talk).*

8. **Drgoňa, J.**, Kvasnica, M., Klaučo, M., Fikar, M.: Explicit Stochastic MPC Approach to Building Temperature Control. In IEEE Conference on Decision and Control (CDC), Florence, Italy, pp. 6440-6445, 2013. citations: 4.

9. Kvasnica, M., Szűcs, A., Fikar, M., **Drgoňa, J.**: Explicit MPC of LPV Systems in the Controllable Canonical Form. In IEEE 2013 European Control Conference (ECC), Zürich, Switzerland, pp. 1035-1040, 2013. citations: 1.

Accreditation Category B; AFD - Conference Proceedings:

10. ***Drgoňa, J.**, Takáč, Z., Horňák, M., Valo, R., Kvasnica, M.: Fuzzy Control of a Laboratory Binary Distillation Column, Accepted to the 21st International Conference on Process Control, Slovakia, 2017.*

11. *Ingole, D., **Drgoňa, J.**, Kalúz, M., Klaučo, M., Bakošová, M., Kvasnica, M.: Model Predictive Control of a Combined Electrolyzer-Fuel Cell Educational Pilot Plant, Accepted to the 21st International Conference on Process Control, Slovakia, 2017.*

12. *Ingole, D., **Drgoňa, J.**, Kvasnica, M.: Offset-Free Hybrid Model Predictive Control of Bispectral Index in Anesthesia, Accepted to the 21st International Conference on Process Control, Slovakia, 2017.*

13. Sharma, A., **Drgoňa, J.**, Ingole, D., Holaza, J., Valo, R., Koniar, S., Kvasnica M.: Teaching Classical and Advanced Control of Binary Distillation Column. In 11th IFAC Symposium on Advances in Control Education, Bratislava, Slovakia, vol. 11, pp. 348-353, 2016.

14. **Drgoňa, J.**, Klaučo, M., Valo, R., Bendžala, J., Fikar, M.: Model Identification and Predictive Control of a Laboratory Binary Distillation Column. In Proceedings of the 20th International Conference on Process Control, Slovak Chemical Library, Štrbské Pleso, Slovakia, 2015. *(personally delivered talk).*

15. **Drgoňa, J.**, Kvasnica, M.: Comparison of MPC Strategies for Building Control. In Proceedings of the 19th International Conference on Process Control, Slovak University of Technology in Bratislava, Štrbské Pleso, Slovakia, pp. 401-406, 2013. *(personally delivered talk)*, citations: 2.

Accreditation Category B; AFG - Abstracts on Conference Proceedings:

16. Ingole, D., **Drgoňa, J.**, Kalúz, M., Klaučo, M., Bakošová, M., Kvasnica, M.: Explicit Model Predictive Control of a Fuel Cell. In The European Conference on Computational Optimization, Leuven, Belgium, vol. 4, 2016. *(personally delivered talk).*

17. Picard, D., **Drgoňa, J.**, Helsen, L.,Kvasnica, M.: Impact of the controller model complexity on MPC performance evaluation for building climate control. In The European Conference on Computational Optimization, Leuven, Belgium, vol. 4, 2016. *(personally delivered talk).*

Appendix B

Curriculum Vitae

Ján Drgoňa

Date of Birth: April 29, 1989
Citizenship: Slovakia
Email: drgona.jan@gmail.com
Homepage: http://www.kirp.chtf.stuba.sk/∼drgona

Professional Experience

- **Position:** PhD Student
  Organization: Institute of Information Engineering, Automation, and Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Slovakia
  Topic: Model Predictive Control with Applications in Building Thermal Comfort Control
  Supervisor: prof. Michal Kvasnica
  Skills covered: Model predictive control, HVAC control, modeling and control of hybrid systems, fuzzy control, state estimation
  Dates: September 2012 - present (*expected graduation in August 2017*)

- **Position:** Visiting PhD Student
  Organization: Department of Mechanical Engineering, Division of Applied Mechanics and Energy Conversion, KU Leuven, Belgium
  Topic: Approximated MPC Laws for Complex Building Climate Control Problems via Machine Learning Algorithms
  Supervisor: prof. Lieve Helsen
  Skills covered: Machine learning, building physics
  Dates: September 2016 - June 2017

Education

- **Degree:** Msc.
  Organization: Institute of Information Engineering, Automation, and Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Slovakia
  Topic: Efficient Modeling of Hybrid Systems
  Supervisor: prof. Michal Kvasnica
  Skills covered: Control theory, optimal control, intelligent control, system modeling and identification, object oriented programming, information engineering and industrial information systems
  Dates: 2010 - 2012

- **Advanced Study:** Visiting Msc. Student
  Organization: Automatic Control, Department of Electrical Engineering (ISY), Institute of Technology Linköping University, Linköping, Sweden
  Topic: Efficient Modeling of Hybrid Systems
  Supervisor: prof. Johan Löfberg
  Skills covered: Mathematical modeling, mixed integer programming, YALMIP
  Dates: February 2012 - April 2012

- **Degree:** Bsc. **Organization:** Institute of Information Engineering, Automation, and Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Slovakia
  Topic: Convex Optimization and Model Predictive Control (in Slovak)
  Supervisor: prof. Michal Kvasnica
  Skills covered: Process control, embedded system control, chemical and energetic engineering, linear algebra, statistics, optimization, thermodynamics
  Dates: 2007 - 2010

Research Interests

Overall research interests deal with the design of Model Predictive Controllers (MPC) suitable for application on embedded hardware (via explicit or approximate solutions), modeling and control of hybrid systems, and use of advanced control strategies for energy demanding systems. Latest research interests are focused on the approximation of the MPC laws via machine learning algorithms and application of the developed algorithms in the context of the building control.

Technical Skills

- **Advanced:** MATLAB/Simulink, MATLAB toolboxes (YALMIP, MPT, Polynomial, System Identification, Machine Learning), HYSDEL, Optimisation solvers (Gurobi,

CPLEX, etc.), Version control systems (Git, Mercurial), LaTeX, Linux, MS Windows, MS Office.

- **Basic:** Python, C/C++, HTML, Programming of industrial control systems (Siemens Simatic, Foxboro, B&R).

Certificates

- **Subject:** Machine Learning
  Speaker(s): Andrew Ng
  Organization: Standford University via Coursera
  Course license: KPJ3KJPFNR4U
  Date: 2016

- **Subject:** TEMPO Spring School on Theory and Numerics for Nonlinear Model Predictive Control
  Speaker(s): Moritz Diehl, James Rawlings
  Organization: IMTEK, University of Freiburg, Germany
  Date: 2015

Obtained Scholarships

- Erasmus+ Scholarship (September 2016 - January 2017)
- National Scholarship Program of the Slovak Republic (February 2017 - June 2017)

Participation on Research Projects

- Internal STU research funding - 2017: Approximated Model Predictive Control of Energy Systems, principal investigator
- APVV SK-CN-2015-0016 – CN-SK cooperation: Robust Model Predictive Control Meets Robotics
- APVV-15-0007 - Optimal Control for Process Industries
- APVV 0551-11 - Advanced and Effective Methods of Optimal Process Control
- VEGA 1/0403/15 - Verifiably Safe Optimal Control
- VEGA 1/0112/16 - Control of Energy Intensive Processes with Uncertainties in Chemical Technologies and Biotechnologies

Appendix C

Resumé

Predkladaná dizertačná práca pojednáva o syntéze a implementácii prediktívneho riadenia na problémy energeticky efektívneho riadenia vykurovacích systémov budov za účelom zabezpečenia vysokej tepelnej pohody vnútorného prostredia. V zahraničnej literatúre je tento typ riadenia známy pod pojmom *model predictive control (MPC)*. O pokročilé regulátory, ktoré riadia teplotu v budovách, je v praxi veľký záujem, pretože výrazne znižujú náklady na vykurovanie a klimatizovanie budov a zároveň zvyšujú komfort obyvateľov a produktivitu pracovníkov.
V úvode tejto práce (kapitola 1) uvádzame prehľad súčasnej literatúry a identifikujeme hlavné výzvy a problémy spojené s aplikáciou pokročilých metód prediktívneho riadenia v praxi. Prediktívne regulátory patria do kategórie optimálneho riadenia. Z toho dôvodu sú v kapitole 2 položené teoretické základy, ako sú matematické definície o množinách, funkciách a polygónoch, ako aj úvody do matematickej optimalizácie, pravdepodobnosti a štatistiky, ktoré sú nevyhnutné na plné porozumenie tomuto textu. O histórii, základných súčastiach a vlastnostiach MPC následne pojednáva kapitola 3.
Teoretické prínosy tejto dizertačnej práce sú detailne opísané v jej druhej časti, ktorá je rozdelená do troch kapitol. Kapitola 4 poskytuje základné informácie o matematickom modelovaní budov ako aj prehľad dostupných simulačných programov. Predstavené sú dva konkrétne matematické modely rezidenčných budov vo forme lineárneho časovo nemenného systému v stavovom opise, viď rovnice (4.1) a (4.12). Oba modely sú použité v simulačných štúdiách v kapitole 6, ktoré slúžia na overenie energetickej a výpočtovej efektivity pokročilých prediktívnych regulátorov navrhnutých v kapitole 5.
Hlavné výhody prediktívneho riadenia budov v porovnaní s klasickými regulátormi sú:

- schopnosť predpovedať správanie sa budovy na základe jej matematického modelu, predpovediach počasia a iných porúch,
- schopnosť systematického prístupu k znižovaniu energetických nákladov,
- schopnosť dodržiavať požiadavky na kvalitu vnútorného prostredia, akou je napríklad tepelný komfort,

- schopnosť rešpektovať technologické a fyzikálne ohraničenia budov.

Nevýhodou prediktívneho riadenia je však jeho náročná aplikácia v praxi, ktorá je aj napriek intenzívnemu vedeckému záujmu ešte stále v počiatočnom štádiu. A to hlavne z dvoch dôvovov. Prvým je fakt, že MPC vyžaduje presný matematický model budovy a jej vykurovacích, ventilačných a klimatizačných (anglicky *heating, ventilation and air conditioning (HVAC)*) systémov. Ako bolo spomenuté v kapitole 4, získanie presného matematického modelu budovy je však časovo aj technicky náročná úloha. Navyše potrebná úroveň zložitosti modelu na dosiahnutie dobrého výkonu MPC zostáva a priori neznáma a na toto určenie nie je k dispozícii žiadna systematická metóda.
Prvým cieľom tejto práce je preto systematická štúdia zložitosti predikčného modelu potrebného na dosiahnutie optimálneho správania sa regulátora pre konkrétnu budovu. Použitá metodológia zobrazená v obr. 6.8 uvažuje tri varianty zložitého nelineárneho model existujúcej šesť-zónovej budovy zobrazenej na obr. 4.1, ktoré sú následne linearizované na modely s vyšším počtom stavov, opísané v podkapitole 4.3. S využitím metódy redukcie zložitosti modelu (anglicky *model order reduction (MOR)*) je vygenerovaná množina modelov s rôznym počtom stavov, t.j. s rôznou úrovňou zložitosti. Výkonnosť MPC s použitím redukovaných modelov (anglicky *reduced order models (ROM)*) je systematicky porovnávaná s výkonnosťou MPC využívajúceho najzložitejší model budovy, slúžiac ako horná medza výkonnosti. Výsledky simulačnej štúdie vzhľadom na kritériá definované v podkapitole 6.3.2 sú kompaktne zobrazené v obr. 6.10. Štúdia ukázala, že minimálny počet stavov modelu sa znižuje s úrovňou izolácie a zvyšuje sa s nárastom hmotnosného obsahu budovy. Bolo demonštrované že predikčná chyba viac ako 0.5 K v rámci predikčného horizontu MPC výrazne znižuje kvalitu riadenia. V prípade skúmanej šesť-zónovej budovy, minimálny počet stavov modelu bol rovný 30. Toto číslo je však výrazne vyššie ako mnohé modely získané pomocou identifikačných techník. Bolo tiež ukázané, že matematická formulácia MPC s modelovaním porúch (anglicky *offset-free MPC (OSF-MPC)*) mierne znižuje nároky na zložitosť modelu a zlepšuje kvalitu riadenia aj pre modely s rozumne malými predikčnými chybami. Obr. 6.11 porovnáva výpočtové nároky on-line implementácie MPC s využitím takzvanej hustej formulácie (anglicky *dense approach*) oproti často využívanej riedkej formulácie (anglicky *sparse approach*). Obr. 6.11 dokazuje že zvýšené nároky na zložitosť lineárneho modelu nemajú pri využití hustej formulácie vplyv na výpočtové zaťaženie riešenia príslušného optimalizačného problému.
Druhou azda závažnejšou prekážkou aplikácie MPC v praxi sú zvýšené požiadavky na hardvér a softvér. Z matematického pohľadu totiž ide pri implementácii MPC o riešenie zložitých optimalizačných úloh 3.1, ktoré vyžadujú veľké výpočtové kapacity a tým pádom sa nedajú použiť na ľahko dostupných a lacných zariadeniach. Pokročilé algoritmy riadenia ako MPC navyše vyžadujú špeciálne vyškolených pracovníkov schopných ladiť a odstraňovať poruchy pri zavádzaní tejto technológie do praxe.
Druhým cieľom tejto práce bolo preto navrhnúť efektívne a vysoko kvalitné MPC regulátory, ktoré je možné použiť a implementovať v dostupných nízkonákladových zariadeniach. Toto riešenie sa v tejto práci dosahuje dvomi principiálne odlišnými spôsobmi.
Prvým prístupom je získanie takzvaného explicitného riešenia príslušného optimalizačného problému aj pre formulácie MPC ktoré zohľadňujú neistoty v predpovedi počasia. Ide o

takzvané stochastické prediktívne riadenie, ktoré je teoreticky detailne predstavené v podkapitole 5.3.4. Toto riešenie využíva takzvané viac-parametrické programovanie (anglicky multiparametric programming), viď. podkapitola 2.2.5. Takéto riešenia sú najprv predpočítané v režime off-line a v on-line režime potom umožňujú jednoduchú a výpočtovo efektívnu implementáciu MPC aj na zariadeniach s obmedzeným výpočtovým výkonom. Výsledky tejto simulačnej štúdie sú predstavené v podkapitole 6.1. Výkonnosť stochastického regulátora predstaveného na obr. 6.4 je porovnaná s najlepším možným scenárom z obr. 6.2, ktorý uvažuje dokonalé poznanie predpovede počasia a s najhorším možným scenárom z obr. 6.3, ktorý uvažuje konzervatívne ohraničenia na tepelnú pohodu. Výsledky potvrdzujú že navrhnutý stochastický regulátor dosahuje takmer rovnakú efektivitu spotreby energie ako nerealistický najlepší možný scenár. Výpočtová efektivita je navyše zabezpečená vďaka explicitnému riešeniu s nízkymi pamäťovými nárokmi. Nevýhodou tohto prístupu sú však striktné obmedzenia na zložitosť riešeného problému.

Nevýhoda obmedzenia zložitosti problému je v druhom prístupe prekonaná zostrojením regulátorov ktoré napodobňujú správanie prediktívneho regulátora cez využitie algoritmov strojového učenia (viď. podkapitola 5.4). Navrhnutá metodológia je kompaktne zobrazená v obr. 5.3. Jej prvá časť spočíva vo využití presného matematického modelu budovy na návrh prediktívneho regulátora, ktorého správanie v uzavretej slučke je simulované v programovom prostredí MATLAB. Druhá časť metodológie je založená práve na využití strojového učenia na aproximáciu správania vzorového prediktívneho regulátora. Konkrétne ide o využitie algoritmov mnohorozmerovej regresie, akými sú regresné stromy (anglicky *regression trees (RT)*) predstavené podkapitole 5.4.4) a časovo oneskorené neurónové siete s hlbokou architektúrov (anglicky *deep time delay neural networks (TDNN)*) predstavené v podkapitole 5.4.5. Podkapitola 5.4.4 navyše predstavuje významné teoretické prínosy k regresným stromom. Výkonnosť RT je zvýžená vďaka optimálnemu rozdeľovaniu uzlov stromu pomocou všeobecných nadrovín 2.1.19 oproti klasickým pravouhlým nadrovinám a vďaka výmene konštantných lokálnych regresorov za afínne výrazy 2.1.11. V tejto časti práce ide v neposlednom rade aj o využitie techník redukcie zložitosti na výber najvýznamnejších parametrov, bližšie opísaných v podkapitole 5.4.6. Konkrétne sa využíva analýza hlavných komponentov (anglicky *principal component analysis (PCA)*), alebo analýza vplyvu porúch v matematickom modeli budovy. Redukciou použitých parametrov sme schopní znížiť výpočtovú náročnosť zostrojovaných regulátorov ako aj znížiť náklady potrebné na senzorifikáciu budov. Hlavná výhoda navrhovaných metód spočíva v ich ľahkej implementácii aj na lacných zariadeniach bez potreby pokročilých softvérových knižníc. Takéto jednoduché regulátory sú navyše zostrojiteľné aj pre zložité problémy s viacerými vstupmi a výstupmi (MIMO) a s veľkým množstvom parametrov, ktoré sú bežné pri problémoch riadenia budov. Všetky tieto vlastnosti sú následne demonštrované na dvoch simulačných štúdiách.

Prvá simulačná štúdia zaoberajúca sa aproximovanými zákonmi prediktívneho riadenia s využitím rozšírených regresných stromov je predstavená v podkapitole 6.2.2. Ukázali sme (viď podkapitola 6.2.1), ako upraviť príslušné RT na dodržiavanie obmedzení na akčné zásaky cez riešenie kvadratického optimalizačného problému (6.2). V tejto štúdii je použitý jednoduchý matematický model jedno-zónovej budovy, opísaný v podkapitole 4.2.

Aj napriek relatívne malej trénovacej množine dát (päť dní), dokázali RT, ktorých priebeh je zobrazený na obr. 6.7(b), dosiahnuť takmer rovnakú výkonnosť ako omnoho zložitejší prediktívny regulátor ktorého profil je zobrazený na obr. 6.7(a). Navyše boli RT schopné uspokojivo extrapolovať akčné zásahy aj na situácie ktoré sa v trénovacej množine nenachádzali.
Druhá simulačná štúdia zaoberajúca sa aproximovanými zákonmi prediktívneho riadenia s využitím neurónových sietí je predstavená v podkapitole 6.4. Simulačné nastavenie ako aj model budovy je prevziaty z podkapitoly 6.3. Bolo ukázané že výkonnosť regulátorov založených na TDNN môže byť vylepšená jednoduchými heuristickými pravidlami 3. Priebehy riadenia pomocou natrénovaných neurónových sietí zobrazené v obr. 6.14(c) a obr. 6.14(d) sú následne porovnané s klasickými regresnými stromami v obr. 6.14(b), zložitým prediktívnym regulátorom v obr. 6.14(a) ako aj s tradičnými prístupmi k riadeniu ako sú regulátory založené na pravidlách (anglicky *rule-based control (RBC)*) v obr. 6.13(a) a skupina PID regulátorov v obr. 6.13(b). Výsledky tejto štúdie zhrnuté v tabuľke 6.3 a v obr. 6.15 demonštrujú schopnosť zostrojiť jednoduché a výkonné aproximácie MPC s využitím neurónových sietí, aj pre veľmi zložité problémy riadenia budov s viacerými vstupmi a výstupmi, ktoré obsahujú aj viac ako tisíc parametrov.
Na záver je možno zhrnúť, že predkladaná práca sa zaoberá návrhom výpočtovo efektívnych algoritmov prediktívneho riadenia a ich využitím na problémy energeticko efektívneho riadenia tepelnej pohody v budovách. Teoretické ako aj aplikačné prínosy tejto práce slúžia ako nevyhnutné medzikroky vedúce k výkonným a nízkonákladovým metódam pokročilého riadenia budov s vysokým praktickým potenciálom. Výzvou zároveň však zostávajú analytické záruky dodržania ohraničení, ako aj záruky kvality riadenia počas celého operačného cyklu budovy. Ako budúce rozšírenia tejto práce vnímam víziu využitia dostupných dát na automatizovú syntézu a ladenie pokročilých, zároveň však jednoduchých prediktívnych regulátorov, ako aj prepojenie s automatizovanými metódami identifikácie presných matematických modelov budov do jedného funkčného rámca.