Institute of Information Engineering, Automation, and Mathematics
Faculty of Chemical and Food Technology
Slovak University of Technology in Bratislava
Radlinského 9, 812 37 Bratislava, Slovak Republic

Model Predictive Control with Applications in Building Thermal Comfort Control

Ing. Ján Drgoňa

Minithesis

Specialization:
Process Control

Supervisor:
Assoc. Prof. Ing. Michal Kvasnica, PhD

In the first place, I would like to express gratitude to my supervisor Assoc. Prof. Ing. Michal Kvasnica, PhD for his professional guidance, and for patience during the last year when I was compelled to interrupt the study due to health reasons. For the second chance to finish my PhD. study I am very grateful to my family members, who were supporing me during my ilness. For the recovery itself I am utterly most grateful to all ivolved in my healing process.

Bratislava, 2015
Ján Drgoňa

Abstract

This thesis deals with class of optimal control methods, more specifically with Model Predictive Control (MPC). In the beginning, necessary mathematical background of mathematical optimization is presented, followed by taxonomy, history and general overwiev of MPC. Further as main aim of this thesis an application of MPC strategies in building thermal comfort control problems are studied.

In the first place we focused on buildings modeling and overview of available software tools for this purpose. Subsequently a appropriate formulation of control criteria were studied for ensuring maximization of thermal comfort, along with minimization of energy consumption, which are in general counteracting criteria. These control criteria can be cast in various ways, each having their pros and cons. Whereupon three different deterministic MPC strategies for controlling temperatures in buildings are proposed.

Further it is shown 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 (including outside weather and building occupancy among others). To account for random disturbances, a formulation with assumed probabilistic version of thermal comfort constraints is presented, as well as methodology how a finite-sampling approach can be used to convert probabilistic bounds into deterministic constraints.

All proposed methods are in the end compared with respect to two mentioned qualitative control criteria on simulation case studies.

Abstrakt

Táto práca sa zaoberá metódami optimálneho riadenia, so špeciálnym zameraním na prediktívne riadenie s modelom. Začiatok práce je venovaný nutným základom a prehľadom problémov matematickej optimalizácie, nasledované klasifikáciou, historickým vývojom a všeobecným prehľadom techník prediktívneho riadenia.

Hlavný cieľ tejto práce je zameranie na preskúmanie možností aplikácie metód prediktívneho riadenia v problémoch riadenia tepelného komfortu v budovách. Pozornosť je v prvom rade venovaná matematickému modelovaniu budov a softvérovým možnostiam pre tieto účely. Sú skúmané vhodné formulácie daných riadiacich kritérií, pre zabezpečenie maximalizácie tepelného komfortu, súčasne s minimalizáciou spotrebovenej energie, čo sú vo všeobecnosti protichodné kritériá. Tieto kritéria môžu byť matematicky formulované viacerými spôsobmi, každý majúc svoje výhody a nevýhody. Na základe vybraných kritérií následne navrhujeme tri rozdielne deterministické formulácie prediktívneho riadenia.

Ďalej sú prezentované techniky syntézy explicitnej reprezentácie prediktívnych zákonov riadenia so spätnou väzbou, navrhnuté na udržanie teplôt v komfortnej zóne, pričom berúc do úvahy aj evolúciu náhodného vplyvu externých porúch (zahŕ ňaj úcich vonkajšie počasie, alebo medzi inými aj prítomnost̂ l̂udí v budove). Modelovanie náhodných porúch, je predstavené ako formulácie s pravdepodobnostými verziami ohraničení špecifikujúcich komfortné teplotné zóny. Následne je predstavená metodológia popisujúca prevedenie výpočtovo náročných pravdepodobnostných ohraničení na numericky ľahko vyhodnotiteľné deterministické ohraničenia s použitím konečného počtu vzoriek.

Všetky predstavené formulácie riadenia sú ku koncu práce porovnané v simulačných štúdiách, vzhľadom na splnenie dvoch spomenutých kvalitatívnych kritérií.

Notation

Mathematical Symbols

x	real valued variable
δ	integer valued variable
x_k	state variable at k-th step
u_k	input variable at k-th step
d_k	disturbance variable at k-th step
r	reference variable
θ	random variable
ξ	vector of parameters
$\mathbb{R}^n$	Column vector of real values of length n
$\mathbb{R}^{n \times m}$	Matrix of real values of n-rows and m-columns

Abbreviations

Functions

PWA	piecewise affine function
PWQ	piecewise quadratic function

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

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

Controllers

PID	proporcional integral derivative
LQR	linear quadratic regulator
LQE	linear quadratic estimator
LQG	linear quadratic gaussian
MPC	model predictive control
RHC	receding horizon control

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
ISE	indoor temperature simulink engineering

Contents

1 Introduction **9**

2 Mathematical Background **11**
- 2.1 Terminology and Definitions on Sets and Functions . . . 11
  - 2.1.1 Sets . . . 11
  - 2.1.2 Functions . . . 13
  - 2.1.3 Polytopes . . . 15
- 2.2 Mathematical Optimization . . . 16
  - 2.2.1 Taxonomy of Optimization . . . 16
  - 2.2.2 Constrained Optimization . . . 16
  - 2.2.3 Convex Optimization . . . 20
  - 2.2.4 Non-convex Optimization . . . 22
  - 2.2.5 Multi Parametric Programming . . . 25
- 2.3 Probability Theory and Statistics . . . 28
  - 2.3.1 Classification and Differences . . . 28
  - 2.3.2 Terminology and Definitions . . . 29

3 Model Predictive Control **34**
- 3.1 Classification of MPC in Control Theory . . . 34
  - 3.1.1 Control Theory Overwiev . . . 34
  - 3.1.2 Classical vs Modern Control Theory . . . 35
  - 3.1.3 Optimal Control . . . 38
- 3.2 History and Evolution of MPC . . . 39
- 3.3 MPC Overwiev and Features . . . 44
  - 3.3.1 Receding Horizon Control . . . 45
  - 3.3.2 Standard MPC Formulation . . . 47
  - 3.3.3 Further Reading on MPC . . . 48

4 Building Control **49**
- 4.1 Building Modeling . . . 49

4.1.1 Basic concepts 49
4.1.2 Tools 50
4.1.3 Particular building model 50
4.2 HVAC Model Predictive Control 51
4.2.1 Control Objectives and Assumptions 52
4.2.2 Deterministic MPC Formulations 55
4.2.3 Stochastic MPC Formulations 58
4.3 Simulation Study 64
4.3.1 Deterministic MPC Performance 64
4.3.2 Explicit Stochastic MPC Performance 68

5 Conclusions and Aim of the Thesis **73**

Bibliography **76**

Chapter 1

Introduction

Nowadays, energy consumed for heating, cooling ventilation and air conditioning (HVAC) in residential and commercial buildings accounts for roughly 40 % of global energy use 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 principal ways can be followed to lower 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. A common 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 efficacy of HVAC control systems Levermore (2000). Various advanced control methods are nowadays available to achieve this goal, ranging from use of classical PI and state-feedback controllers Canbay et al. (2004), through methods based on artificial inteligence contepts 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 application of optimization-based schemes Ma et al. (2009), Široký et al. (2011). Advantage of the latter class is that the task of minimizing energy while respecting thermal comfort can be rigorously stated as an optimization problem, leading to a best-achievable performance.

In this thesis we therefore also follow the line of devising optimization-based controllers for control of thermal comfort in buildings. Specifically, we consider the class of Model Predictive Control (MPC) strategies Maciejowski (2002b) which utilize a mathematical model of the building to predict its future behavior. These predictions then allow optimization to select best control inputs which minimize a given cost function while maintaining physical constraints, such as limits of actuators or safety regulations. Besides control of buildings, successful applications of MPC span from chemical and petrochemical applications, up to autonomous driving, for more examples see e.g. Kvasnica (2009), Qin and Badgwell (2003).

The work is organized as follows:

- Chapter 2 is presenting a brief introduction to mathematical optimization, probabilty theory an statistics, together with rigorous definitions on sets an functions.

- Chapter 3 is devoted to more detailed introduction, history and overview of MPC techniqies and their applications.
- Chapter 4 introduces a basic concepts of building modeling and control. Here, theoretical study and design of different model predictive control strategies for building thermal comfort control are presented. Followed by the evaluation of their performance on simulation case studies for single-zone building model.
- Finally in Chapter 5 the conclusions are being made.

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 will provide basic terminology and definitions necessary for understanting following section 2.2 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 $\mathcal{S} \subseteq \mathcal{B}_r(s)$*

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

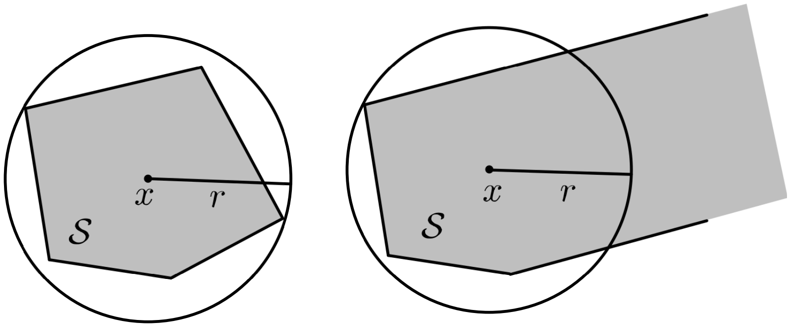
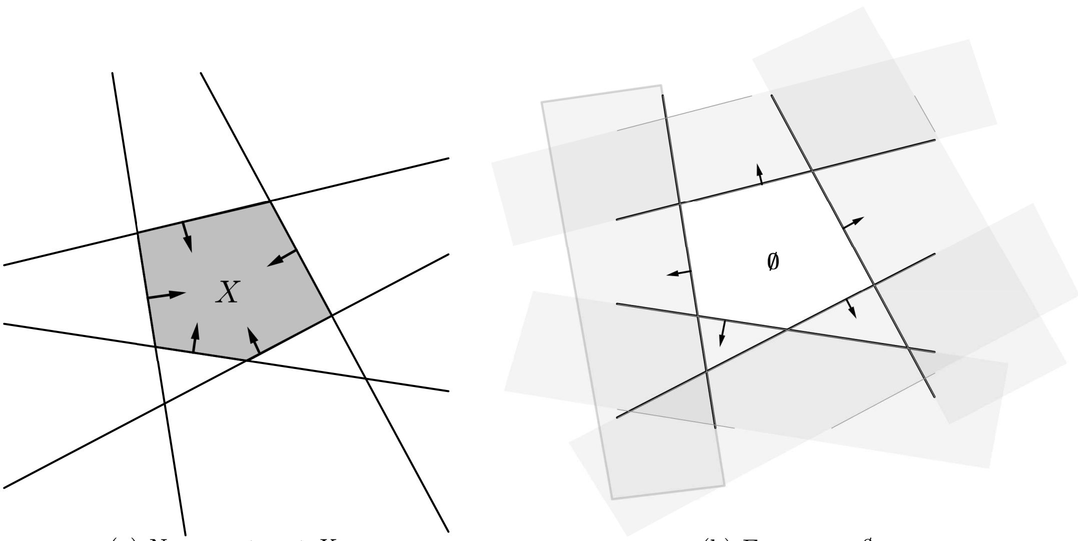


Figure 2.1: Bounded set (left) contained in a ball $\mathcal{B}_r(x)$ and unbounded set (right) uncontained in its entirety inside a ball $\mathcal{B}_r(x)$.

Definition 2.1.6 (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.



(a) Non-empty set X.

(b) Empty set $\emptyset$.

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

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 $v_i \in \mathbb{R}^n$, $\forall i = 1, \ldots, M$, is the smallest convex set containing $\mathcal{V}$ defined as*

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

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 $i = 1, \ldots, N_\mathcal{S}$ 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 $i, j \in \{1, \ldots, N_\mathcal{S}\}$*

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 compositon 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 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 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.*

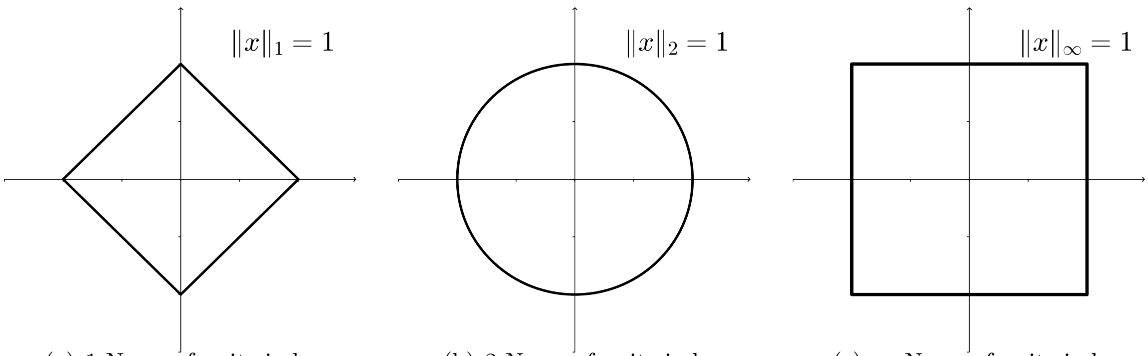
$$\|x\|_1 = \sum_i^n |x_i|. \tag{2.12}$$

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 $a_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$, $\forall i = 1, \ldots, m$

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

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

Where $a_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$, $\forall i = 1, \ldots, m$

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*

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

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 $\mathcal{V} = \{v_1, \dots, v_M\}$, for $v_i \in \mathbb{R}^n$, $\forall i = 1, \dots, M$, 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

Mathematical optimization is a important tool in making decisions and in analyzing physical systems applied in wide variety of scientific fields of study, namely economics, operations research, chemical, mechanical and finaly control engineering as a main concern of our study. More comprehensive insight into the rich topic of mathematical optimization the reader can find in references such as Boyd and Vandenberghe (2004), Society. (2014)

2.2.1 Taxonomy of Optimization

To provide a taxonomy of optimization is a very difficult task because of dense multiple connections between optimization subfields. One such comprehensive perspective focused mainly on the subfields of deterministic optimization with a single objective function can be found online NEOS (2014).

Moreover a wide collection of available solvers organized by problem type can be also found on NEOS Server web-pages NEOS-software (2014), NEOS-solvers (2014), together with valuable informations about 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 with respect to 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 nondifferentiable) NEOS (2014). For further reading see references Bertsekas (1996)

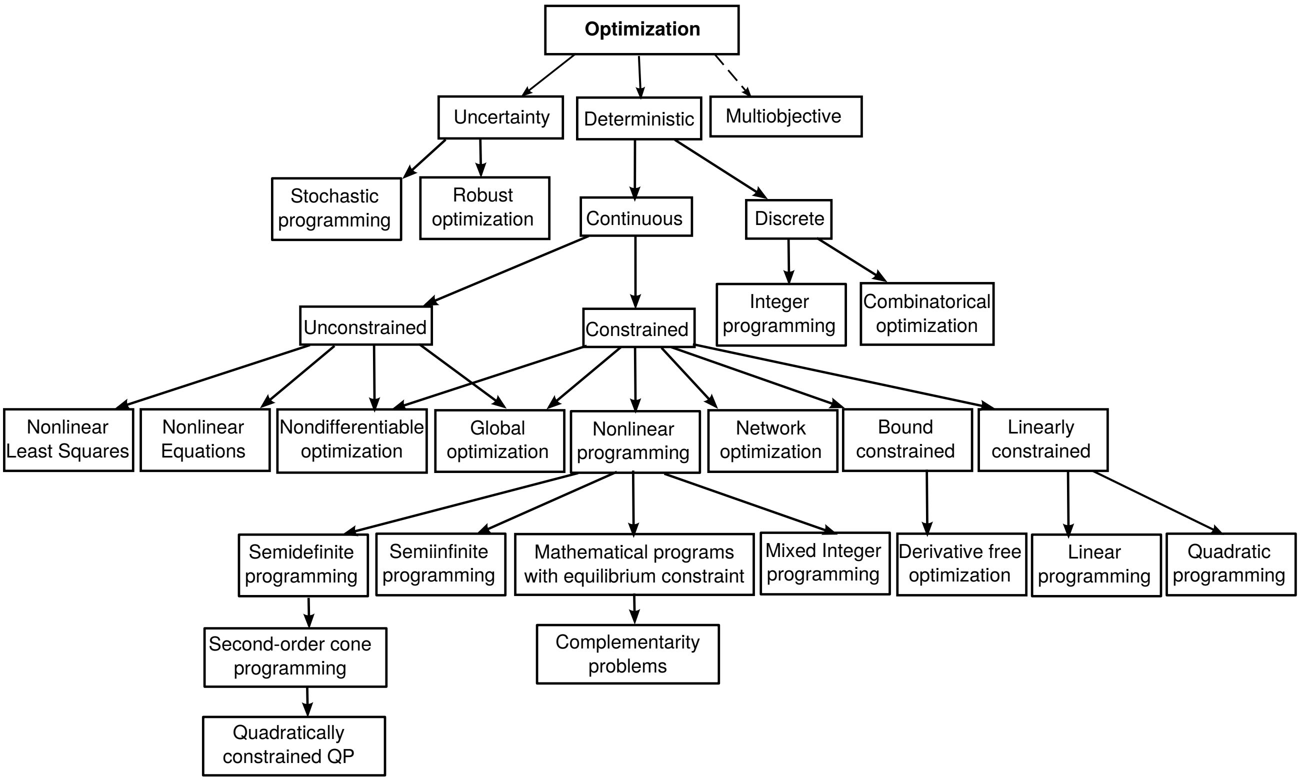


Figure 2.4: Classification of Optimization problems NEOS (2014).

Standard Optimization Problem

In mathematical optimization terminology standard optimization problem is key notion representing the problem of finding the best soultion among group of all possible and feasible solutions. Standard form of continuous constrained optimization problem is defined as follows Boyd and Vandenberghe (2004).

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

$$\text{s.t. } g_i(x) \leq 0, \qquad i = 1, \dots, m \tag{2.20b}$$

$$h_j(x) = 0, \qquad j = 1, \dots, p \tag{2.20c}$$

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 $x = (x_1, x_2, \dots, x_n)^T$ assigns concrete real value $f_0(x)$ and which overall value has to be minimized during optimization. Maximization problem can be treated by 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 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 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 together by notion of costraints set $\mathcal{S}$. More clarified classification of constraints for practical needs can be found in documentation for MATLAB $OptimizationToolbox^{TM}$ 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, these kind of constraints which must be hold during whole optimization procedure are also called *hard constraints*. However in some optimization problems can appear constraints which are prefered but not required to be satisfied, this kind of non-mandatory constraints are known as *soft constraints*, which are specific by having some additional usually called *slack variables* that are penalized in 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 an 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 optimization procedure. A *candidate solution* therefore must be unconditionally a member of feasible set of all possible solutions for a given problem.

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

$$X = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \ldots, m; \quad h_j(x) = 0, j = 1, \ldots, p\} \tag{2.21}$$

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

In general *feasible set* can be considered to be *bounded* if it is in a certain sense of a finite size, or it can be considered *unbounded* if it containts 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 dependance on the objective function specifications, thus an unique solution of the problem may not exist.

Difficulties apper also in case, if there are no intersection of the problems constraints, therefore there are no points that satisfy all the constraints simultaneously, thus the feasible region is consideret to be the null set. Which is 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 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 if often considered also a minimizer as a vector which achieves that value, usually denoted as x^* or x^{opt}, if exists.

When an objective function is not provided, the problem (2.20) is being called a *feasibility problem*. This means 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:*

$$X^* = \{x \in \mathbb{R}^n : f_0(x) = J^*; \quad g_i(x) \leq 0, i = 1, \ldots, m; \quad h_j(x) = 0, j = 1, \ldots, p\} \tag{2.22}$$

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 theoretical point of view the notion of optimal solution is crucial. However for practical reasons there was established a weaker notion of suboptimal solution of the problem, representing points which are very close to optimum. This is because practical algorithms 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*

$$X^\epsilon = \{x \in \mathbb{R}^n : f_0(x) = J^* + \epsilon; \quad g_i(x) \leq 0, i = 1, \ldots, m; \quad h_j(x) = 0, j = 1, \ldots, p\} \tag{2.24}$$

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 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*, at least some of the constraints are nonlinear functions.

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

2.2.3 Convex Optimization

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

- the objective function $f_0(x)$ is convex
- the constraint set $\mathcal{S}$ is convex;

Convex problems are very popular and prefered in comparission 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 labeled as infeasible.
- Convex problems are in contrast with non-convex problems generaly easily solved, with wide variety of suitable solvers.

However a practical problems often exhibits non-convex properties, hence convex problems are not always suitable framework for solutions real-world problems, what is the main drawback of them. But where they can be applied, they used to be extremely efficient.

In following text we will introduce two basic types of convex optimization problems with linear constraints, namely Linear programming 2.2.3 and Quadratic programming 2.2.3.

Linear Programming

(LP) 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.

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

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

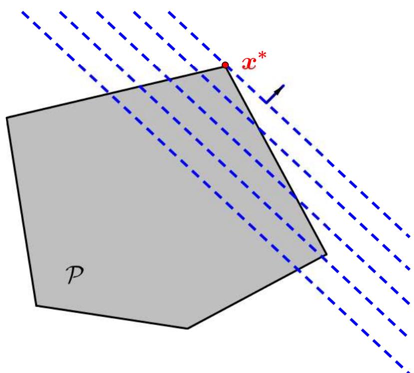
$$A_{eq}x = b_{eq}, \tag{2.25c}$$

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

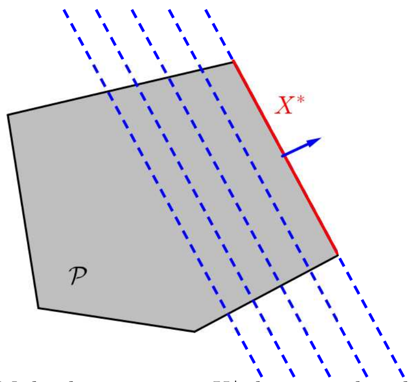
Solution Properties: LP can be geometrically interpreted as searching for a optimum x^* of a linear objective function over a given polyhedral region $\mathcal{P}$. This procedure can result in several differen scenarios.

1. *Feasible problem*, with value of 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 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 vertex of the region $\mathcal{P}$. And situation Fig. 2.5(b), when objective function is paralel to one of the constraint with resulting multiple solutions X^* of equal quality, which lies on 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 depicted as a red dot for x^*, or red line for X^* respectively.

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

Quadratic Programming

(QP) 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.

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

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

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

where $x \in \mathbb{R}^n$, $H \in \mathbb{R}^{n \times n}$, $q \in \mathbb{R}^n$, $c \in \mathbb{R}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $A_{eq} \in \mathbb{R}^{p \times n}$ and $b_{eq} \in \mathbb{R}^p$. 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 general in only two scenarios.

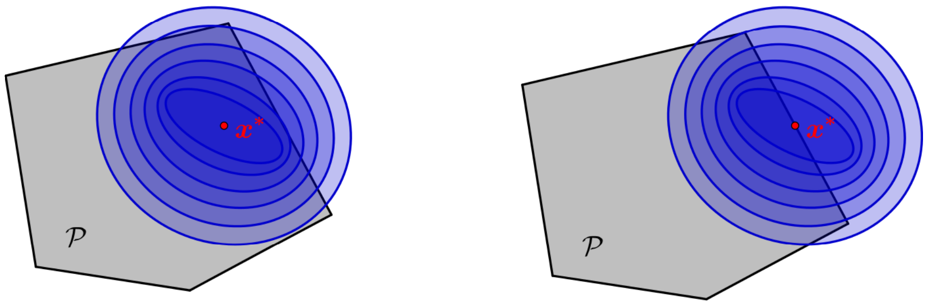
1. *Feasible problem*, with value of objective function $-\infty < J^* < \infty$, and unique optimizer x^*.
2. *Infeasible problem*, with value of objective function $J^* = \infty$, caused by empty polyhedron $\mathcal{P}$.

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

There have beed done deep research about solution and properties of QP problems, some can be found e.g. in Abrams and Ben Israel (1969), Angelis et al. (1997), Beale and Benveniste (1978), Best and Kale (2000).

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 od such problems are not convex. Because there does not exists unique approach for optimization algorithm selection, structure of general non-convex problem must be examined first, and consequently selection of appropriate methoda for particular problem class. For our purposes important class of constrained non-linear programing is called Mixed Integer Programming (MIP), containting continuous and also discrete variables. More deeper view inside a class of MIP problems can be found e.g. in Nemhauser and Wolsey. (1988), Schrijver. (1984), Wolsey. (1998)



(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 depicted by blue elipses and optimizer x^* depicted as a red dot.

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.

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

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

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

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

Solution Properties: MILP can be geometrically interpreted as searching for a optimum x^* of a linear objective function over a given polyhedral region $\mathcal{P}$, where 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 efficiency of such relaxed problems a several techniques are being used, one such method is called cutting plane method Avriel (2003), Boyd and Vandenberghe (2004) based on iterative refinement of a feasible set or objective function by means of linear inequalities, termed cuts. Cutting plane method, together with whole MILP optimization procedure is demonstraded on Fig.2.7.

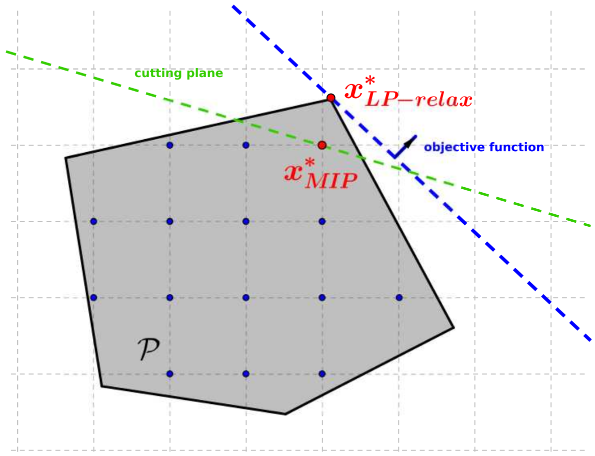


Figure 2.7: LP relaxation and cutting plane method for solution of MILP problem. Where $\mathcal{P}$ represents constraints set, objective function is represented by dashed blue line with its direction vector, additional cutting plane reshaping constraints set is represended by dashed green line, the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

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.

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

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

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

Where $x \in \mathbb{R}^n$, $\delta \in \mathbb{N}^q$, $H_1 \in \mathbb{R}^{n \times n}$, $H_2 \in \mathbb{R}^{n \times q}$, $H_3 \in \mathbb{R}^{q \times q}$, $c^T \in \mathbb{R}^n$, $d^T \in \mathbb{R}^q$, $A \in \mathbb{R}^{m \times n}$, $E \in \mathbb{R}^{m \times q}$, $b \in \mathbb{R}^m$, $A_{eq} \in \mathbb{R}^{p \times n}$, $E_{eq} \in \mathbb{R}^{p \times q}$, and $b_{eq} \in \mathbb{R}^p$. 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.

Solution and computational aspects of MIP problems

Even though from structural point of view, the difference of MILP or MIQP problems againts their convex counterparts as LP or QP problems are often very small and sometimes hidden in restriction of some variables to be integer-valued. Yet their true difference is manifested in solution of such problems and in comparisson of their computational requirements. In straightforward fashion a to obtain a solution of MILP or MIQP problem respectively is to enumerate all possible combinations of binary variables δ, and for each combination of fixed binaries as static parameters, compute optimal solution for real variables x contained in the problem as classical LP or QP problem respectively. This approach shows us main drawback of such problems in their exponetially growing complexity depending on number of included binary variables δ. Therefore a several techniques for reduction of necessary combinations of binary variables to be enumerated have been developed, namely widely used *Branch and Boud* and *Branch and Cut* methods Belotti and Mahajan. (2013), C. S. Adjiman and Floudas. (1996), Linderoth and Ralphs. (2005), Richards and How. (2005). Moreover a number of tricks and hacks are beying used, e.g. for reduction of number of binary variables to decrease a computational burden and improve performance of MIP problems in general. Commonly used commerical solvers such as CPLEX ILOG, Inc. (2003) or Gurobi Gurobi Optimization (2012), have become extremely efficient in solving MIP problems, for more available MIP solvers visit NEOS-software (2014), NEOS-solvers (2014).

2.2.5 Multi Parametric Programming

Generaly 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 exists several approaches to parameter variations and dealing with 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 and Tarantola (2008).

Second is called *interval analysis* where uncertainties are modeled by interval ranged input data and analyzing the solution of such problem.

Finally a *parametric programming* is method for obtaining and analysis of the optimal solution of an optimization problem with given a full range of parameter values, representing a feasible initial conditions of the problem. Parametric programming systematically subdivides the space of parameters into characteristic 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 F. Borrelli (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 typ 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 4.2.3. Further in this section we will define multiparametric versions of general, LP, QP, MILP and MIQP problems.

Standard Multiparametric Program

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

$$J^*(x) = \min_z \; J(z,x) \tag{2.29a}$$

$$\text{s.t. } Gz \leq w + Ex, \tag{2.29b}$$

where $z \in \mathbb{R}^s$ represents vector of optimization variables, $x \in \mathbb{R}^n$ stands for vector of parameters (state variables), $J^*(x)$ represents optimal value of the objective function $J(z,x)$, and $z^*(x)$ 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

$$J^*(x) = \min_z \; c^T z + d^T x \tag{2.30a}$$

$$\text{s.t. } Gz \leq w + Ex, \tag{2.30b}$$

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

Multiparametric Quadratic Programming

A multiparametic quadratic programming (mpQP) problem is defined as

$$\min_z \; \tfrac{1}{2} z^T H z + x^T Q z + x^T R x + d^T x \tag{2.31a}$$

$$\text{s.t. } Gz \leq w + Ex, , \tag{2.31b}$$

where $z \in \mathbb{R}^s$, $x \in \mathbb{R}^n$, $G \in \mathbb{R}^{r\times s}$, $w \in \mathbb{R}^r$, $E \in \mathbb{R}^{r\times n}$, $Q \in \mathbb{R}^{n\times s}$, $R \in \mathbb{R}^{n\times n}$, $d \in \mathbb{R}^n$, 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

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

$$\text{s.t. } Gz + S\delta \leq w + Ex, \tag{2.32b}$$

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

Solution Properties of mpP Problems

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

- *Feasibility set* $\mathcal{X}$, or domain of parameters x, as a set of parameters for which a particular problem has optimal solution, usually in a form of polytopic partition.
- *Optimal solution* or optimizer $z^*(x)$, usually in form of PWA polytopic functions defined over polytopic partition $\mathcal{X}$. It is representing a sets of parameters x, for which the optimal solution remains the same, respectively retains the same characteristics.
- *Feasibility function* $J(z,x)^*$, as a optimal value of the objective function $J(z,x)$ for the feasibility set, usually in two forms as polytopic PWA functions for mpLP or polytopic PWQ functions for mpQP problems.

Theorem 2.2.4 (Properties of mpP) *Consider a mpLP 2.30, mpQP 2.31, and mpMILP 2.32 problems then:*

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

$$\mathcal{P}_r = \{x \in \mathbb{R}^n \mid H_r x \leq K_r\}, \quad r = 1, \dots, R \tag{2.33}$$

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

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

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

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

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

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

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

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

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

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

  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 key objects here are 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 large data sets quantitative analysis, which can occur in many fields of practical or theoretical fields of study, and it also lies down mathematical foundations for Statistics, allowing modelig randomness and uncertainty of empirical data sets.

Mathematical Statistics is a branch of mathematics dealing with analysis, collection, interpretation, presentation and organization of data. Pracitical application of statistics counts for planing, summary and analysis of inaccurate or empirical observations.

The difference between Statistics and Probability theory may not seem obvious due to tight boundary between these field, but yet some fundamental differences there exists and are very briefly captured in Fig. 2.8. The probability theory is used for desription of formation, generation or evolution of stochastic data, where statistics on the other hand is used for analysing and manipulating these random-fashinoned data for modeling and identification of processes behavior by whose were these data generated.

Moreover by using probability theory tools and paying the cost of certain information loss due to partial knowledge of the states we are also able to estimate the behavior of large complex systems based on behavior of stochastic variables. Ability to comprehend and deal with large and complex systems goes far beyond the limit of classical deterministic approaches for description of dynamical systems, which is reason for large usage of statistical methods in practice.

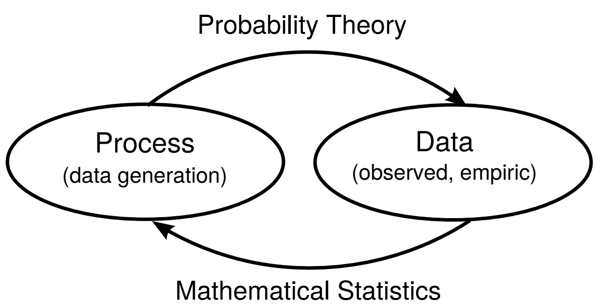


Figure 2.8: Relation between probability theory and mathematical statistics.

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

- **Confirmatory data analysis** or inferential statistics, which draws conclusions from data or where the hypothesis is formulated and subsequently confirmed or disproofed by confirmatory data analysis techniques (e.g. regression analysis, confidence intervals, etc.). Confirmatory analysis uses the traditional statistical tools of inference, significance, and confidence. It is comparable to a court trial, it is the process of evaluating evidence.

- **Exploratory data analysis** also called descriptive statistics, on the other hand describes data, i.e. summarises the data and their typical properties, or uses data sets and generating hypotheses by its techniques (e.g. cluster analysis, factor analysis, principal component analysis, etc.). If a model fits to 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, it is the process of gathering evidence.

2.3.2 Terminology and Definitions

Probability is value representing certainty respectively uncertainty of particular event. It is computed as cardinality of true or occuring events $m = |E|$, divided by 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. Basic simplest example of random event is flip of coin or dice roll. Than the probability of particular result of coin flip is 50% because flipping a coin leads to sample space composed of only two possible outcomes that are equally likely. Similarly probability of dice roll result is one occuring 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 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 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, then the probability of any subset cannot be defined either.

These assumptions can be summarised by notion of measure space, also called *Probability space* or probability triple $(\Omega, \mathcal{F}, \Pr)$ what is structured set that models a real-world process consisting of randomly occuring states, and it is constructed by three parts:

- *Sample Space* Ω, as set of all possible outcomes, with aggregated probability $\Pr(\Omega) = 1$.
- *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, what is a real valued function assigning probabilities to the events, defined on 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 scienc, usually uderstood 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 comprehensible 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 (1996), Kallenberg (1986; 2001), Papoulis (2001). Here comes the notion of *random element*, what is a generalization of the concept of *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 case 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 Varlable) *For real observation space, real-valued random variable is the function $X : \Omega \to \mathbb{R}$ if it is measurable, what means that for each set $B \in \mathcal{R}$ holds:*

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

Equivalently X is 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 $\mathbf{X} = (X_1, ..., X_n)^T$ 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 a deterministic rule, but representing the evolution of random values over time, described by probability distributions. Behavior of 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 deterministic proces which can only evolve in one way, thus stochastic process is usually understood as the probabilistic counterpart to a deterministic process Papoulis (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 $\mathbb{R}$.

Common examples of processes modeled by stochastic framework include 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 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 set of random values of model's parameters. Outputs are recorded and the process is repeated with a new set of random values. Whole procedure is repeated until a reasonable amount of data is gathered (with respect to particular case). In the end, the distribution of the outputs shows the most probable estimates as well as boundaries of expectations Dlouhy (2005).

We can roughly classify stochastic simulation approaches into following types:

- *Discrete-event simulation* representing discrete event simulation.
- *Continuous simulation* representing continuous event simulation.
- *Hybrid simulation* stands for combined simulation of discrete and continuous events.
- *Monte Carlo simulation* commonly used estimation procedure, based on averaging independently taken samples from the distribution Dlouhy (2005).
- *Random number generators* as 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 consistent theory of integration to be defined for integrals of stochastic processes with respect to 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* for data-sets with non-numerical sample space.
- *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 subclass of eliptical 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 extremely important in statistics due to central limit theorem, which says that under certain contitions normal distributions well approximates huge set of other probabilisty distributions classes (continuous or discrete) Casella G. (2001), Lyon (2014).

Definition 2.3.6 (Normal Distribution)

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

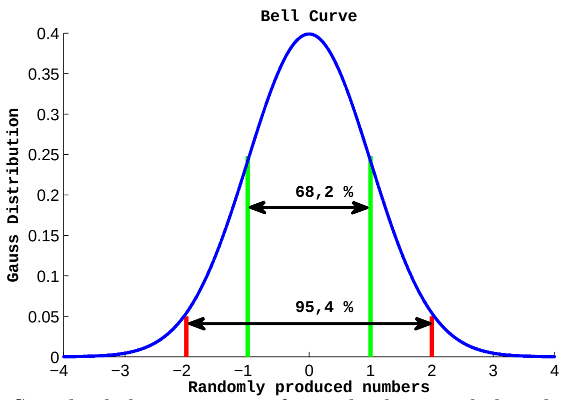
The parameter μ represents the mean and also its median and mode. The parameter σ stands for standard deviation, its variance is therefore σ^2. Thus when a random variable X is distributed normally with mean μ and variance σ^2, we write $X \sim \mathcal{N}(\mu, \sigma^2)$.

Definition 2.3.7 (Standard Normal Distribution)

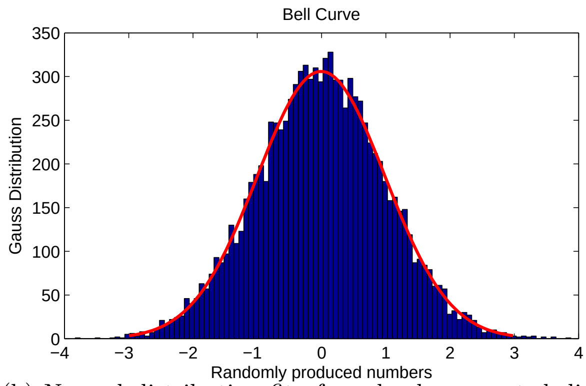
$$f(x) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{x^2}{2}} \tag{2.45}$$

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

An important property of gaussian distribution is that the values less than one standard deviation σ from the mean represent approximately 68.2% of the area under the curve, while two σ from the mean take about 95.4%, and three σ account for about 99.7% of the area, as captured in Fig. 2.9(a), moreover Fig. 2.9(b) demonstrates normal distribution approximation of randomly generated discrete data set.



(a) Standard deviations σ of standard normal distribution $\mathcal{N}(0, 1)$.



(b) Normal distribution fit of randomly generated discrete data.

Figure 2.9: Normal Distribution.

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 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 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” Bradley Anderson (2014).

3.1 Classification of MPC in Control Theory

3.1.1 Control Theory Overwiev

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).

3.1.2 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. 3.2, which are necessary for understanting the differences between classical and modern control theory methods.

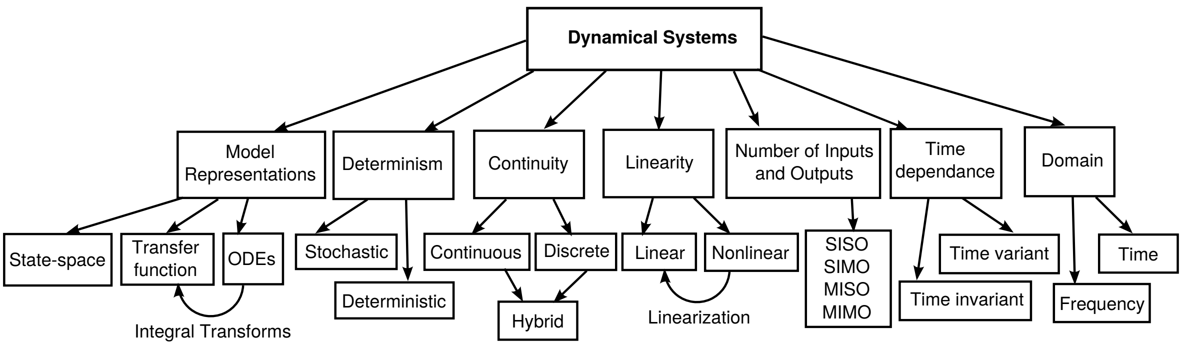


Figure 3.1: Characteristics and properties classification of Dynamical Systems.

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 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.

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 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

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.

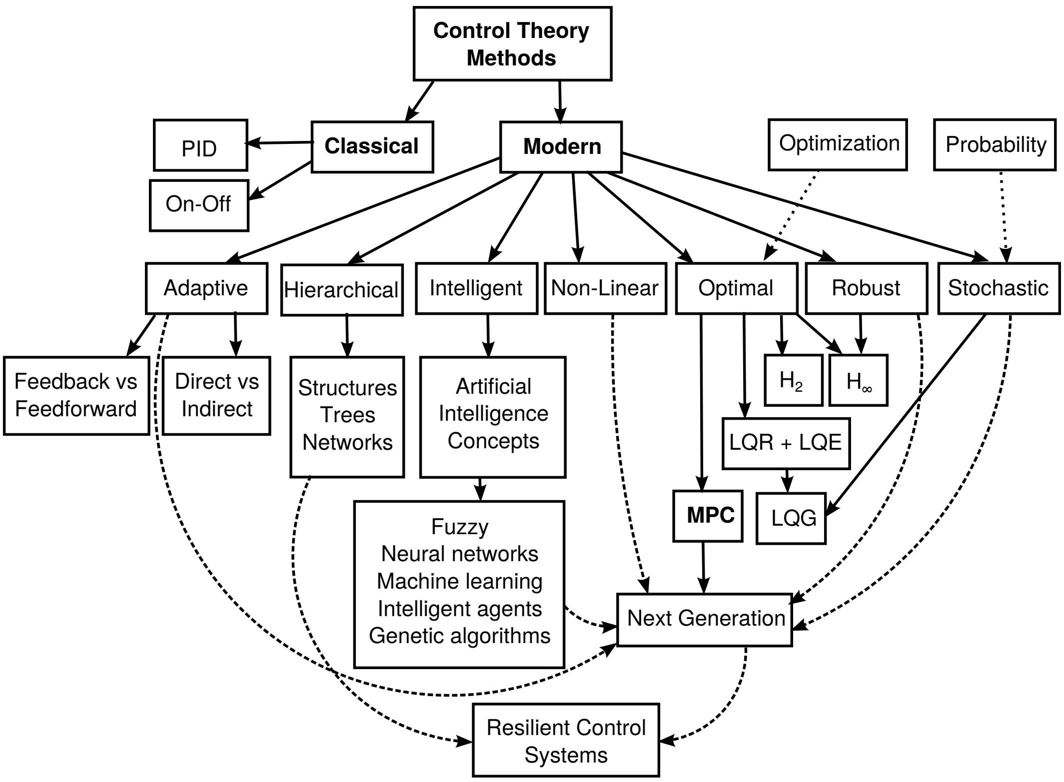


Figure 3.2: Classical vs modern control theory methods taxonomy. Where the full lines represents direct structural relations, dotted lines are depicting supporting mathematical theories, and dashed lines are outlining the evolution of separate control theory methods merging together creating a new control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

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.

3.1.3 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 Kemin Zhou (1997; 1995), Mi-Ching Tsai (2014), Skogestad S (2007).

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 controlled system to be in linear time-invariant form with quadratic objective function and missig constraints. Solution is being obtained by two Riccati equations, in form of optimal linear state feedback controller in form $u = -Kx$.

- **Linear quadratic estimator - LQE**

  In control theory literature also refered as Kalman optimal state estimator, or shortly 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 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 again linear state feedback controller similarly as for LQR. Main disadvantage of both methods, LQR and LQG are poor robust properties of resulting controllers. These drawbacks were acting as motivation combination of optimal and robust control theory methods, leading to development of $\mathcal{H}_2$ and $\mathcal{H}_\infty$ 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 2.1.17 is being used, in contrast with ∞-Norm 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.

3.2 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 article Qin and Badgwell (2003), from where the inspiration for this whole section was taken. Moreover the simplified evolution of industrial MPC algorithms is captured of Fig. 3.3, forming a structural backbone for this section.

Early Optimal Control Theory

Development of modern control concepts using optimization techniques can be traced in the early 1960's beginning with the work of Kalman Kalman (1960a;b). With first attempts for optimal control of linear systems, resulting in development of 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 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 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 Garcia et al. (1989), Richalet and Papon (1976).

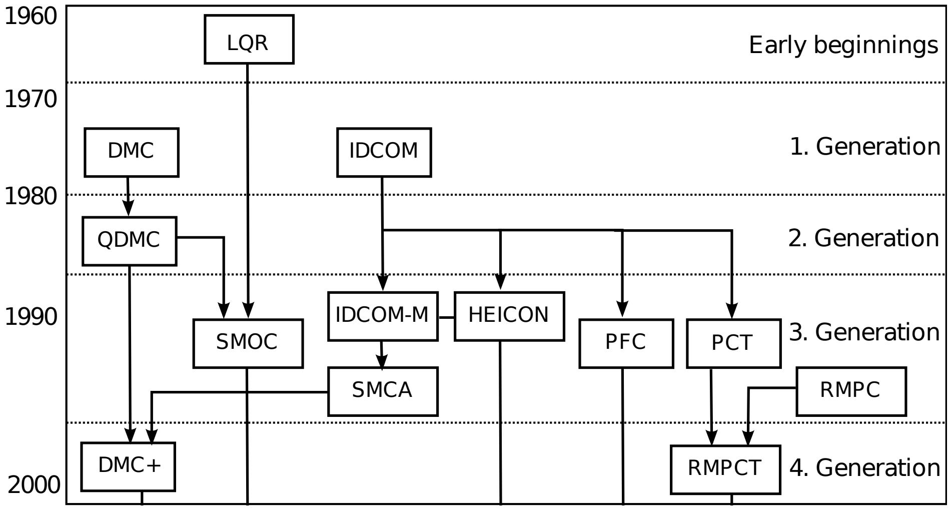


Figure 3.3: Simplified evolutionary tree of the most significant industrial MPC algorithms.

- 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 G.C. Goodwin (2001) it was estimated there may be thousands of real-world LQG applications with roughly 400 patents per year based on Kalman filter.

First Generation MPC

To handle a drawbacks of 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 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 gapbetween 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. (1978), Richalet and Papon (1976) 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

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. 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 QDMC Cutler and Haydel (1983), and Garcia and Morshedi with more comprehensive description few years later C.E. Garcia (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 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 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 D.M. Prett (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. IDCOM-M controller was a commercial trademark of Setpoint, Inc. (where M stands for multiple input/output), and was first described in a paper by Grosdidier,Froisy, and Hammann Grosdidier and Hammann (1988), and few years later by Froisy and Matsko 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 IDCOM-M called hierarchical constraint control (HIECON). IDCOM-M 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 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 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

SMOC algorithm can be than perceived as solving the LQR problem with input and output constraints, but lacking the strong stabilizing properties due to 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 J.B. Rawlings (1993), P.O.M. Scokaert (1998). Other algorithms not described but yet belonging in this section of third MPC generation was 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 MPC technology evolution is sumarised structurally in Fig. 3.3 as refered in the beginning of this section.

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 MPCs significant performance improvements, increasing safety, decreasing energy consumption or enviromental burden of plants production.

3.3 MPC Overwiev and Features

As written in the beginning of this chapter, MPC is control strategy that uses optimization to calculate 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 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.

MPC is based on iterative character of optimization process executed over finite time interval also called 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.

3.3.1 Receding Horizon Control

Standardly MPC algorithm are being implemented in the closed-loop fashion using the principle of *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. Moreover a characteristic behavior of discrete closed-loop MPC strategy is captured in Fig 4.1. Alternatively an open-loop MPC can be designed by ignoring RHC control policy and simply implementing not only the first control input but the whole control strategy computed over given prediction horizon, paying the cost of lost feedback properties of control system.

Remark 3.3.1 *Notice here, that this is one of the many algorithm possibilities for general MPC algorithm definition.*

Algorithm 1: Closed-loop MPC control algorithm (RHC).

---

1. Obtaining a model of the process by the control engineers.
2. At time t, measurements of previous process inputs and outputs are applied to the process model, to compute predictions of future process outputs over a prediction horizon N.
3. The optimal sequence of control inputs $\{u^*(t), \ldots, u^*(t + NT_s)\}$ is computed by solving the optimization problem.
4. Only the first element of the control signals sequence, i.e., $u^*(t)$ is selected and applied to the plant, to achieve feedback behavior of MPC controller.
5. The control signal is implemented over a pre-defined time interval, called sampling time T_s.
6. Time advances to the next interval, and the procedure is repeated from step 2, with new measurements at time $t + T_s$, using values of $x(t + T_s)$.

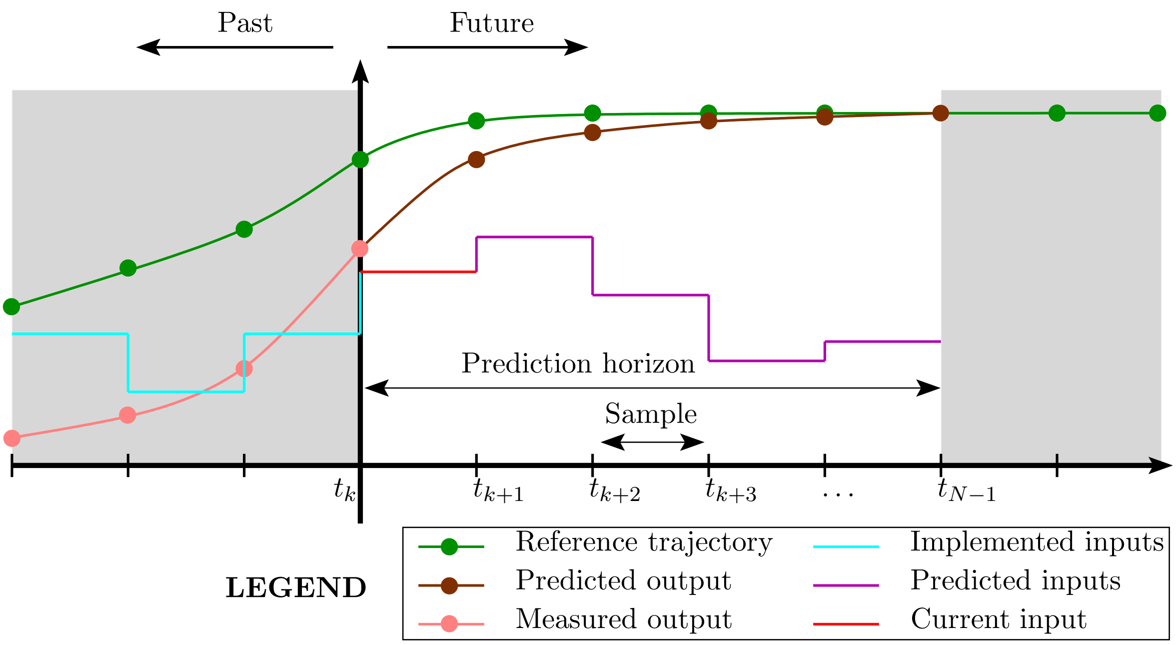


Figure 3.4: Characteristic behavior of receding horizon control policy.

3.3.2 Standard MPC Formulation

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

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

$$\text{s.t. } x_{k+1} = f(x_k, u_k, d_k), \tag{3.1b}$$

$$x_k \in \mathcal{X}, \tag{3.1c}$$

$$u_k \in \mathcal{U}, \tag{3.1d}$$

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

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 $\ell(x_k, u_k)$ in (3.1a) 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 $u_0^*, \dots, u_{N-1}^*$ 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

$$\min_{u_0,\dots,u_{N-1}} \sum_{k=0}^{N-1} (\|Q_x x_{k+1}\|_p + \|Q_u u_k\|_p) \tag{3.2a}$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k + Ed_k, \tag{3.2b}$$

$$H_x x_k \leq K_x, \tag{3.2c}$$

$$H_u u_k \leq K_u, \tag{3.2d}$$

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

Where cost function (3.2a) is represented by arbitrary p-Norm 2.1.15, over prediction horizon $N \in \mathbb{N}$ with weight matrixes $Q_x \in \mathbb{R}^{n\times n}$ and $Q_u \in \mathbb{R}^{m\times m}$, with contitions Q_x to be positive semidefinite and Q_u to be positive definite, or compactly $Q_x \succeq 0, Q_u \succ 0$. Moreover the prediction model holds form of discrete-time linear time-invariant system in a state-space representation 3.2b with incorporated disturbances d_k. With the model matrixes $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $E \in \mathbb{R}^{n\times q}$, linear constraints matrixes $H_x \in \mathbb{R}^{n_x\times n}$, $K_x \in \mathbb{R}^{n_x}$, $H_u \in \mathbb{R}^{n_u\times m}$ and $K_u \in \mathbb{R}^{n_u}$, 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 3.3.2 *Notice that, MPC oprimization problem 3.2 with cost function* (3.2a) *in form of 2-Norm 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 exhbits hybrid dynamics behavior and the resulting optimization problem becomes non-convex MIQP problem 2.28.*

3.3.3 Further Reading on MPC

There are several publications comprehensively covering theoretical and practical issues of MPC techniques. Overwiev MPC tutorial written mainly with focus on control engineers can be found in Rawlings (2000). Allgower (1999) is providing more comprehensive overview of nonlinear MPC and moving horizon estimation. Review of theoretical results on the closed-loop behavior of MPC algorithms found in Mayne (2000). Other important surveys of MPC technoloy are e.g. Garcia et al. (1989), Lee (1996), Mayne (1997), Morari (1991), Muske and Rawlings (1993), Rawlings (1994), Ricker (1991). For books dealing with MPC one can see e.g. Bordons (2004), Kouvaritakis (2001), Maciejowski (2002a), Mayne (2000), Zheng (2000).

Chapter 4

Building Control

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

Frank Gehry

4.1 Building Modeling

4.1.1 Basic concepts

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

In the literature devoted to control of buildings, various types of models were suggested. Nonlinear models provide great accuracy, but lead to difficult control synthesis. Therefore linear models are often considered. Due to their simple structure, subsequent control synthesis and verification is rather straightforward. As a downside, linear models only accurately represent the physical plant in close neighborhood of a particular linearization point. As an extension, one can also consider so-called linear hybrid systems, which are composed of a set of local linear models (each representing operation of the plant around one distinct operating point), accompanied with switching rules that select a particular local model. The upside is increased accuracy of description. However, control synthesis for such systems is difficult, since switching of local models need to be encoded as a set of logic rules.

There are several aspects which distinguish modeling (and 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 model then depends on how well we are able to estimate, or predict future evolution of these disturbances.

4.1.2 Tools

A wide range of software modeling tools for buildings is nowadays available. These include, but are not limited to, TRNSYS Beckman et al. (1994), Energy Plus Crawley et al. (2001), ESP-r A. Yahiaoui (2003). They usually consider very complex building models based on nonlinear energy and mass balances. Although such models are very accurate, they are not directly suitable for control synthesis due to high complexity. To deal this issue the middleware softwares such as BCVTB Wetter and Haves (2008), MLE+ Bernal et al. (2012) and OpenBuild Laboratory (2013) were designed for making communication bridges between Matlab and Energy Plus. More comprehensive overwiev of HVAC system modeling and simulation tools can be found in Trcka and Hensen (2010), Zhou et al. (2013), moreover a directories listing all available software tools for modeling, analysis, optimization and simulation for buildings can be found online in Energy (2014), EUROSIS (2014), Truong Xuan Nghiem (2011).

In this work, we have used the ISE (Indoor temperature Simulink Engineering) tool van Schijndel (2005), which is based on linear building models. In general, ISE is a free, MATLAB-based modeling tool for simulating the indoor temperature of a building that consists of a single zone. It uses a linear model of the zone 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. An another advantage is that ISE is standalone, i.e., it does not rely on any other external software packages. ISE being based on MATLAB/Simulink allows to easily verify performance of various control strategies just by wrapping any MATLAB-based control algorithm as a Simulink S-function.

4.1.3 Particular building model

In this work we consider a linear model of a one-zone building, obtained from ISE. 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 continuous-time domain

$$\dot{x} = Ax + Bu + Ed \tag{4.1}$$

where $x = [x_1, \ldots, x_4]^T$ is the state vector, $\dot{x}$ is the state derivative, $d = [d_1, \ldots, d_3]^T$ is the vector of disturbances, and and $A \in \mathbb{R}^{4\times 4}$, $B \in \mathbb{R}^4$, $E \in \mathbb{R}^{4\times 3}$ are the state-update matrices.

Following values of A, B, E in (4.1) were extracted from ISE:

$$A = 10^{-3} \cdot \begin{bmatrix} -0.020 & 0 & 0 & 0.020 \\ 0 & -0.020 & 0.001 & 0.020 \\ 0 & 0.001 & -0.056 & 0 \\ 1.234 & 2.987 & 0 & -4.548 \end{bmatrix}$$

$$B = 10^{-3} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.003 \end{bmatrix}, \; E = 10^{-3} \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.055 & 0 & 0 \\ 0.327 & 0.003 & 0.001 \end{bmatrix}.$$

The model is valid provided that all temperatures (i.e., states $x_1, \ldots, x_4$) are kept within

$$-30^\circ\, C \leq x_i \leq 50^\circ\, C, \; i = 1, \ldots, 4, \tag{4.2}$$

and for the following range of the heating/cooling inputs injected to the zone:

$$-500\, W \leq u \leq 10000\, W. \tag{4.3}$$

Here, positive values of u represent heating, while a negative u stands for cooling.

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), captured over the period of 30 days. For the heat generated by occupancy, the ISE model considers $d_2 = 500\, W$ during office hours, and $d_2 = 0\, W$ otherwise.

4.2 HVAC Model Predictive Control

This section is devoted to application of MPC techniques on building control problems. One of the first attempts for implementation and analysis of MPC on real building with experimental results is described in Široký et al. (2011). Standard MPC approach with weather predictions was introduced in Oldewurtel et al. (2012), or stochastic MPC formulation in Drgoňa et al. (2013), Ma et al. (2012b), Oldewurtel et al. (2010a). A method for reducing peak electricity demand in building climate control can be found in Oldewurtel et al. (2010b). Predictive control together with building parameter identification was described in Bălan et al. (2011). A distributed model-based predictive control was designed in Ma et al. (2011). Simple MPC-like feedback laws obtained by machine learning tools from closed-loop data was introduced in Klaučo et al. (2014). Explicit MPC approach based on PMV index control, combining several factors affecting the persons thermal comfort, can be found in Klaučo and Kvasnica (2014).

In this Section we propose different formulations of MPC that maximize the thermal comfort while minimizing consumed energy. This is achieved:

- first by elaborating control objectives that include maximization of thermal comfort and minimization of energy consumption, as shown in Section 4.2.1.

- secondly by employing a suitable prediction model from Section 4.1.3, based on which accurate predictions can be obtained.

These predictions then allow optimization to select best control inputs that minimize consumed energy while respecting thermal comfort criteria. In real-life situations, however, buildings are subject to external disturbances, such as exterior temperature, solar radiation, or heat generated by occupancy Oldewurtel et al. (2012). Moreover, these disturbances often evolve in a random fashion, therefore conventional MPC approaches are not directly applicable to real building control, since they are based on deterministic models and constraints Široký et al. (2011). Even though deterministic MPC formulations can be still used in simulation case studies as a valuable source of information, necessary for designing of efficient MPC controllers. Therefore in Section 4.2.2 a three different deterministic formulations of MPC are being proposed, obtained results are reviewed from paper Drgoňa and Kvasnica (2013)

Possible ways around to deal with real-world disturbances would be to consider worst-case MPC which utilizes (often conservative) bounds on possible changes of the disturbances or employing certainty equivalence MPC (CEMPC) Messina et al. (2005). Even though such strategies are able to satisfy constraints for arbitrary disturbance, the price to be paid is increased consumption of energy required for heating and cooling. Moreover based on international standards for building control (e.g. ISO 7730), zone control is being used for satysfying the thermal comfort within given range of reliability. This brings us to face the probabilistic constraints, which can not be modeled by standard deterministic procedures, but on the other hand thanks to the knowledge of the probability distributions of the past disturbances, they can be easily modeled by stochastic approaches. Therefore much attention is devoted to *stochastic* MPC which incorporate random disturbances into probabilistic constraints. However, most of stochastic MPC approaches for building control reported in the literature lead to complex optimization problems Ma et al. (2011; 2012a), Oldewurtel et al. (2010a). The downside being that complexity of such an optimization is often prohibitive for implementation on cheap process hardware, such as on Programmable Logic Controllers. The procedure for obtaining such low complexity stochastic MPC implementation will be shown in Section 4.2.3, obtained results are reviewed from paper Drgoňa et al. (2013).

4.2.1 Control Objectives and Assumptions

Building Automation System (BAS) is a control system which governs buildings such that certain comfort criteria are achieved. These include internal room temperatures, air quality, lighting etc. In addition. Instead of tracking particular reference values, BAS typically consider comfort ranges, also called zones. The task is then to manipulate the building inputs such that qualitative criteria are kept within these zones while simultaneously minimizing cost of injected energy flows.

The classical control approaches (e.g. PID) are to performing an actions based only on current timestamps, representing buildings behavior in present time, without ability to predict or estimate their future evolution. On the other hand MPC framework allows us to perform the minimization while taking prediction of disturbances (which include outside weather,

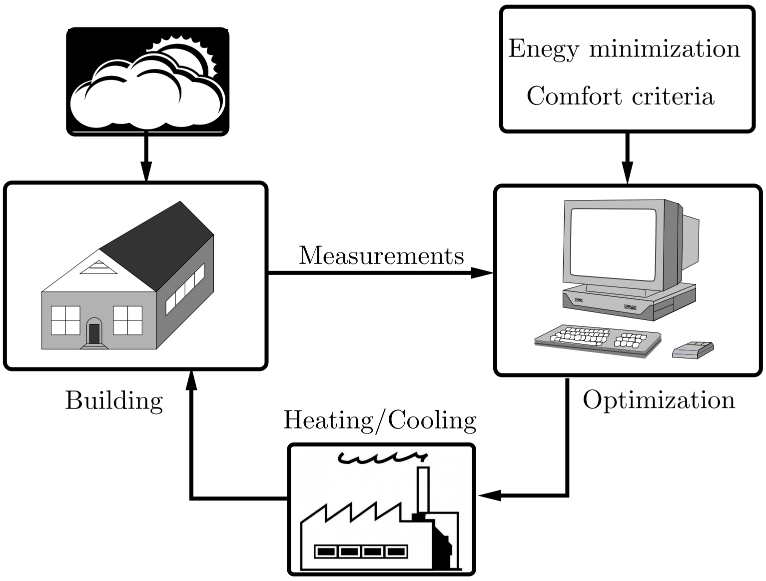


Figure 4.1: Building control scheme, consisting of a building affected by disturbances due to weather. Control inputs are selected by optimization which is initialized by current measurements. Control commands are then passed to actuators which deliver required amount of heating/cooling energy to the building.

building occupancy, solar radiation, etc.) into account. A typical setup of a HVAC control system that includes an MPC algorithm is shown in Fig. 4.1.

Maintaining of Thermal Comfort

First control objective is to maintain temperatures inside the building within a comfortable range. Two variations of this criterion are considered. In the first formulation we are interested in manipulating the injected energy flow u in (4.1) such that the internal room temperature (represented by state x_4) stays as close as possible to its setpoint r. The second formulation relaxes the first condition in the sense that we are only interested in keeping the inner temperature in a particular range, what is equivalent to satisfaction of the following constraint:

$$r - \epsilon \leq x_4 \leq r + \epsilon, \tag{4.4}$$

Here, $\epsilon > 0$ denotes a fixed width of the comfort zone.

Minimization of Energy Consumption

The second very important criterion is to manipulate inputs to the building as to minimize the energy consumption. This objective can be translated to minimizing some function of the control inputs. Typically, minimization of u^2 is employed, since the square directly represents energy of a particular flow. Other types of minimization functions can be considered as well, for instance the absolute value of u.

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/cooling capacities to become active. On the other hand, zero heating/cooling 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.

Probabilistic Disturbances Modeling

The nontrivial part of designing a suitable control strategy stems from presence of disturbance variables d in (4.1). 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 $k = 0, \dots, N$, where N is the length of the prediction window. Then we call the control strategy

$$u^*(t) = \mu(x(t), d(t), \dots, d(t + N)) \tag{4.5}$$

the **best-case scenario**.

2. If we can bound future disturbances by $\|d(t + kT_s) - d(t + (k + 1)T_s)\| \leq \theta_{\max}$ for any $0 \leq k \leq N$, then the control strategy

$$u^*(t) = \mu(x(t), d(t), \theta_{\max}) \tag{4.6}$$

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

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

$$\theta \sim \mathcal{N}(0, \sigma(t)) \tag{4.7}$$

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

$$d(t + kT_s) = d(t) + k\theta, \ \ k = 1, \dots, N, \tag{4.8}$$

where N denotes the prediction window over which the distribution in (4.7) is deemed reasonably accurate. Then the control strategy

$$u^*(t) = \mu(x(t), d(t), \sigma) \tag{4.9}$$

is called the **stochastic scenario**.

Remark 4.2.1 *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 θ for every k-th step of the prediction, we can model this behavior by simple summation of predicted disturbances as shown in formula (4.8), 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 $\theta_{\max}$, 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. As a drawback of hard constraint (4.4) here, is that it can lead to infeasibility of the control problem if a large disturbance hits the building. Because the control authority is limited by (4.3), and due to the random nature of disturbances, the hard thermal comfort constraint (4.4) needs to be relaxed in a probabilistic sense as follows:

$$\Pr(x_4 \geq r - \epsilon) \geq 1 - \alpha, \tag{4.10a}$$

$$\Pr(x_4 \leq r + \epsilon) \geq 1 - \alpha, \tag{4.10b}$$

where $1 - \alpha$ the denotes probability with which the constraints in (4.4) have to be satisfied for some $\alpha \in [0, 1]$.

Our further goal in Section 4.2.3 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.

4.2.2 Deterministic MPC Formulations

Throughout this section we assume that the prediction model $f(\cdot,\cdot,\cdot)$ in (3.1b) is a linear discrete-time model of the form

$$x(t + T_s) = \tilde{A}x(t) + \tilde{B}u(t) + \tilde{E}d(t), \tag{4.11}$$

where $\tilde{A}$, $\tilde{B}$, $\tilde{E}$ are obtained by discretizing the continuous-time model (4.1) with the sampling period equal to $T_s = 444$ seconds. Predicted states $x(t)$ and inputs $u(t)$ are subject to lower/upper limits in (3.1d) and (3.1c) with $\underline{x} = [-30,\ -30,\ -30,\ -30]^T$, $\overline{x} = [50,\ 50,\ 50,\ 50]^T$, $\underline{u} = -500$, and $\overline{u} = 10000$, cf. (4.2) and (4.3). Moreover, we assume that at each time step t we know $d(t)$, but have no information about future disturbances $d(t + 1), \ldots, d(t + NT_s)$. As a consequence we assume that $d_k = d(t)$ for all $k = 0, \ldots, N - 1$ in (3.1b).

Setpoint Temperature Tracking with Energy Minimization

First we consider the first case of Section 4.2.1, with thermal comfort objective. Here, we want to formulate the MPC optimization problem (3.1) such that:

1. the inside temperature (i.e., x_4) is kept as close as possible to its setpoint r;

2. the cost of heating and cooling is minimized.

The first requirement can be reflected by minimizing the difference between predicted values of x_4 from the setpoint r. The second objective then translates to minimizing u_k^2. Hence the cost to be minimized in (3.1a) becomes

$$\ell(x_k, u_k) = q_x(Cx_k - r)^2 + q_u u_k^2, \tag{4.12}$$

where $C = [\,0\ 0\ 0\ 1\,]$ and $q_r > 0$, $q_u > 0$ are weighting parameters. If $q_u/q_r > 1$, then minimization of energy consumption is preferred over achieving good tracking. Otherwise the priority is put towards minimizing the tracking error.

By using (4.12) in (3.1) and with (4.11) employed in (3.1b), the optimization in (3.1) becomes a strictly convex quadratic optimization problems with $u_0, \ldots, u_{N-1}$ being the decision variables. Using off-the-shelf solvers (e.g. by CPLEX ILOG (2007) or GUROBI Gurobi Optimization (2012)), such problems can be solved efficiently even for large values of the prediction horizon.

However, the choice of stage cost as in (4.12) can potentially lead to a non-zero steady-state tracking error. This downside is eliminated by minimizing the increments of u instead of their absolute values. Introduce $\Delta u_k = u_k - u_{k-1}$ and let u_k for $k = -1$ denote the optimal control action computed at the previous sampling instant, i.e., $u_{-1} = u^*(t-1)$. Then

$$\ell(x_k, u_k) = q_x(Cx_k - r)^2 + q_u \Delta u_k^2, \tag{4.13}$$

leads to an offset-free tracking of the reference r, see e.g. Kvasnica (2009). Since the stage cost (4.13) is quadratic in the decision variables x_k and Δu_k, problem (3.1) remains a quadratic program. However, the price to be paid for offset-free tracking is that we loose control over absolute values of the control signals. Hence the MPC problem can, potentially, produce control actions that are large (leading to a large energy consumption), as long as these inputs do not vary by too much in consecutive steps. Thus for above mentioned reasons in Section 4.3, the formulation with minimization of absolute value of energy 4.12 was considered as a reasonable trade-off between energy consumption and thermal comfort.

Comfort Zone Temperature Tracking

Now we consider the second case of Section 4.2.1 with thermal comfort objective, where instead of driving the indoor temperature x_4 towards the setpoint r, it suffices to keep it within the comfort zone

$$r - \epsilon \leq x_4 \leq r + \epsilon \tag{4.14}$$

some positive value of ϵ. Here, ϵ is the width of the zone and represents limits of human perception of temperature differences. Doing so allows to reduce energy consumption, since the indoor temperature is not required to reach the reference, which is in the middle of the comfort zone.

To account for the zone tracking scenario, we propose to proceed as follows. Introduce new variables $s_k \in \mathbb{R}$ and add the following constraints to (3.1) for $k = 0, \ldots, N-1$:

$$r - \epsilon - s_k \leq Cx_k \leq r + \epsilon + s_k, \tag{4.15}$$

together with constraints $s_k \geq 0$, $k = 0, \ldots, N-1$, and with $C = [\,0\;0\;0\;1\,]$. Moreover, let the stage cost be

$$\ell(s_k, u_k) = q_s s_k^2 + q_u u_k^2. \tag{4.16}$$

Then it is easy to observe that for any x_k satisfying (4.14), we have that $s_k = 0$ satisfies (4.15). Only if x_k is outside of the zone, a non-zero value of s_k needs to be employed to satisfy (4.15). In simple terms, constraint (4.15) is a soft version of (4.14) where s_k are the softening variables. Moreover, since the square s_k^2 is penalized in (4.16) via (3.1a), we get that the MPC optimization problem assigns $s_k = 0$ whenever the temperature is kept within the zone in (4.14), and $s_k > 0$ otherwise. Hence the optimization (3.1) with stage cost as in (4.16) will select control actions that keep the indoor temperature as close as possible to the comfort zone. Moreover the resulting optimization remains convex quadratic programming problem.

The stage cost in (4.16) can be furthermore modified as to minimize increments of the control action, i.e.,

$$\ell(s_k, u_k) = q_s s_k^2 + q_u \Delta u_k^2, \tag{4.17}$$

where Δu_k is defined as in Section 4.2.2. In either case, the optimization problem (3.1) is a quadratic program with decision variables x_k, u_k, and s_k for $k = 0, \ldots, N-1$.

Comfort Zone Temperature Tracking via Binary Indicators

The advantage of the comfort zone tracking approach presented in Section 4.2.2 is its simple implementation via quadratic optimization. The downside, however, is that the optimization prefers many small violations of the comfort zone over smaller number of larger violations. The reason being that any non-zero value of s_k coincides with violation of the zone in (4.14). But minimizing s_k^2 in (4.16) only minimizes magnitudes of such violations. Therefore in this section we show how to formulate the stage cost which minimizes the true number of violations, instead of their respective magnitudes.

The central idea of this approach is to devise new binary indicator variables $\delta_k \in \{0, 1\}$ for which

$$(s_k > 0) \implies (\delta_k = 1). \tag{4.18}$$

This implication means that $\delta_k = 1$ whenever the temperature violates the thermal comfort zone in (4.14) at time instant $t + k$. Minimization of the number of instances at which the zone is violated can then be achieved by using the following stage cost:

$$\ell(\delta_k, u_k) = q_\delta \delta_k + q_u u_k^2. \tag{4.19}$$

Note that the stage cost is always non-negative due to the binary nature of δ_k. The second term of (4.19) can alternatively be replaced by $q_u \Delta u_k^2$ when minimization of the increments of the control variable is desired.

Constraint (4.18) can be included into the MPC optimization problem (3.1) by applying the so-called big-M technique Williams (1993). First we rewrite (4.18) as

$$(s_k \geq \gamma) \implies (\delta_k = 1). \tag{4.20}$$

for some small positive value of γ (typically chosen as the machine precision) to convert the strict inequality in (4.18) to a non-strict one (4.20). Now let $M > 0$ be a sufficiently large constant such that $M \geq \max s_k$. Then (4.20) is equivalent to

$$s_k - \gamma \leq M\delta_k. \tag{4.21}$$

To see the equivalence, notice that for any $s_k > \gamma$ (which implies that $s_k - \gamma$ is positive and hence $s_k > 0$) the optimization needs to choose $\delta_k = 1$ in (4.21) for the constraint to hold since M is assumed to be positive. For any $s_k \leq \gamma$ (e.g. for $s_k = 0$ since $s_k \geq 0$ is assumed, cf. (4.15)), both $\delta_k = 0$ as well as $\delta_k = 1$ are feasible in (4.21). However, since δ_k are minimized in (4.19), $\delta_k = 0$ will be chosen by the optimization in such a case.

Therefore by including constraints (4.21) into the MPC optimization problem (3.1), and by using the stage cost (4.19), we can minimize the true number of violations of the comfort zone. The price to be paid is the increased complexity of the optimization problem. Specifically, (3.1) has to consider additional binary optimization variables δ_k. Such a problem can be solved as a mixed-integer problem (MIP). Its complexity is much higher compared to convex quadratic optimization problems and is, in the worst case, exponential in the number of binary decision variables. Nevertheless, modern solvers can solve MIP problems of moderate complexity in reasonable time, as will be demonstrated in Section 4.3.

4.2.3 Stochastic MPC Formulations

The aim of this Sectionis at acquire a simple implementation of stochastic MPC. This 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* Bemporad et al. (2002), Borrelli (2003). This gives rise to an explicit representation of the MPC feedback law as a Piecewise Affine (PWA) function 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 stochastic MPC problems of small size Grancharova et al. (2008).

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

Comfort Zone Temperature Tracking

An MPC optimization problem for maintaining high probability of achieving thermal comfort while minimizing energy consumption can be stated as follows:

$$\min_{u_0,\ldots,u_N} \sum_{k=0}^{N} u_k^2 \tag{4.22a}$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k + E(d_0 + k\theta), \tag{4.22b}$$

$$\Pr(Cx_k \geq r - \epsilon) \geq 1 - \alpha, \tag{4.22c}$$

$$\Pr(Cx_k \leq r + \epsilon) \geq 1 - \alpha, \tag{4.22d}$$

$$-700 \leq u_k \leq 1400, \tag{4.22e}$$

$$\theta \sim \mathcal{N}(0, \sigma(t)). \tag{4.22f}$$

Here, x_k, u_k and d_k denote, respectively, values of states, inputs and disturbances predicted at the k-th step of the prediction horizon N, initialized by current measurements of the states $x_0 = x(t)$, current disturbances $d_0 = d(t)$ and desired center of the thermal comfort zone $r = r(t)$. The predictions are obtained using a discretized version of the LTI model (4.1) with $C = [\,0\ 0\ 0\ 1\,]$. Future disturbances predicted in (4.22b) employ the random variable θ, that follows the probability distribution (4.22f) where $\sigma(t)$ is assumed to be available to the optimization. The term $d_0 + k\theta$ in (4.22b) originates directly from (4.8).

Due to the probabilistic constraints (4.22c)-(4.22d), problem (4.22) 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 θ, as captured by the following two lemmas.

Lemma 4.2.2 (Campi and Garatti (2008)) *Let $g(u, \theta) : \mathbb{R}^N \times \mathbb{R}^{n_d} \to \mathbb{R}$ be a function that is convex in u for any θ, and let θ be a random variable as in (4.7). Assume a probabilistic constraint*

$$\Pr(g(u, \theta) \leq 0) \geq 1 - \alpha \tag{4.23}$$

for some $\alpha \in [0, 1]$. Let $\theta^{(1)}, \ldots, \theta^{(M)}$ be M samples of the random variable independently extracted from (4.7). Then the probabilistic constraint in (4.23) is satisfied with confidence $1 - \beta$, i.e., $\Pr(\Pr(g(u, \theta) \leq 0) \geq 1 - \alpha) \geq 1 - \beta$, if

$$g(u, \theta^{(i)}) \leq 0, \;\; i = 1, \ldots, M, \tag{4.24}$$

holds for a sufficiently large M. ■

Lemma 4.2.3 (Alamo et al. (2010)) *The number of samples M required in Lemma 4.2.2 is bounded from below by*

$$M \geq \frac{1 + N + \ln(1/\beta) + \sqrt{2(N+1)\ln(1/\beta)}}{\alpha}. \tag{4.25}$$

■

By employing Lemma 4.2.2 we can thus replace the probabilistic constraints in (4.22c)-(4.22d) by a finite number M of deterministic constraints, each obtained for one of the realizations $\theta^{(i)}$ of the random variable. To see this, notice that for (4.22c) we can set $g(\cdot,\theta) := r - \epsilon - Cx_k$, where x_k embeds θ via (4.22b). Clearly, $g(\cdot,\cdot)$ is a single-valued linear function (hence convex) for each k, therefore Lemma 4.2.2 is applicable. Similar reasoning holds for (4.22d). In addition, Lemma 4.2.3 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., $\theta^{(i)}$, and denote by

$$y_k^{(i)} = C\left(A^k x_0 + \sum_{j=0}^{k-1} A^{k-j-1}\left(Bu_j + E\left(d_0 + (j+1)\theta^{(i)}\right)\right)\right) \tag{4.26}$$

the indoor temperature (represented by the 4-th element of the state vector), predicted at the k-th step of the prediction horizon using the disturbance $\theta^{(i)}$. Note that (4.26) follows directly by solving for $y_k = Cx_k$ from (4.22b). Then the MPC optimization problem (4.22) can be cast as

$$\min_{u_0,\ldots,u_N} \sum_{k=0}^{N} u_k^2 \tag{4.27a}$$

$$\text{s.t. } y_k^{(i)} \geq r - \epsilon,\ i = 1,\ldots,M, \tag{4.27b}$$

$$y_k^{(i)} \leq r + \epsilon,\ i = 1,\ldots,M, \tag{4.27c}$$

$$-700 \leq u_k \leq 1400, \tag{4.27d}$$

where $y_k^{(i)}$ is given per (4.26). Note that (4.26) serves as a substitution in (4.27b)-(4.27c) and is *not* considered as an equality constraint. By Lemma 4.2.2, a feasible solution to (4.27) implies that the probabilistic constraints in the original formulation (4.22) will be satisfied with a high confidence $1-\beta$. The initial conditions for problem (4.27) are the current state measurements $x_0 = x(t)$, current value of the disturbance vector $d_0 = d(t)$, and the M samples $\theta^{(1)},\ldots,\theta^{(M)}$ extracted from the probability distribution (4.7) for a current value of the standard deviation $\sigma(t)$. Most importantly, the optimization (4.27) 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 (4.10), respects limits of the control authority in (4.3), and minimizes the energy consumption, can be achieved as follows:

1. At time t, measure $x(t)$, $d(t)$, $r(t)$ and obtain $\sigma(t)$.
2. Generate M samples $\theta^{(1)},\ldots,\theta^{(M)}$ from (4.7).
3. Formulate the QP (4.27) 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 $t + T_s$.

Remark 4.2.4 *To prevent infeasibility of (4.27) during transient (for instance when the zone middle point r is changed), it is worth to soften the hard constraints in (4.27b)-(4.27c). This can be done by introducing new variables s_k, $k = 0, \ldots, N$, and by replacing (4.27b)-(4.27c) by*

$$y_k^{(i)} \geq r - \epsilon - s_k, \quad i = 1, \ldots, M, \tag{4.28a}$$

$$y_k^{(i)} \leq r + \epsilon + s_k, \quad i = 1, \ldots, M. \tag{4.28b}$$

Then the term $\sum_{k=0}^{N} q_s s_k^2$ must be added to (4.27a) to penalize violation of constraints, together with extra constraints $s_k \geq 0$, $k = 0, \ldots, N$. The value of the penalization coefficient q_s should be selected high as to discourage MPC from violating the constraints unless absolutely necessary. □

Explicit Stochastic MPC

The objective here is to employ parametric programming to pre-calculate the optimal control inputs in (4.27) 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.

Theorem 4.2.5 (Bemporad et al. (2002)) *Let*

$$\min\{U^T H U + \xi^T F U \mid GU \leq W + S\xi\} \tag{4.29}$$

be a convex quadratic program with initial conditions $\xi \in \mathbb{R}^{n_x}$ and decision variables $U \in \mathbb{R}^N$. Then the optimizer U^* of (4.29) is a piecewise affine (PWA) function of ξ:

$$U^*(\xi) = \begin{cases} F_1\xi + g_1 & \text{if } \xi \in \mathcal{R}_1, \\ & \vdots \\ F_R\xi + g_R & \text{if } \xi \in \mathcal{R}_R, \end{cases} \tag{4.30}$$

where $F_i \in \mathbb{R}^{N \times n_x}$, $g_i \in \mathbb{R}^N$, and $\mathcal{R}_i \subseteq \mathbb{R}^{n_x}$ are polyhedral regions. ■

To see the relation between Theorem 4.2.5 and the QP (4.27), notice that $U = [u_0, \ldots, u_N]$ and $\xi = [x(t), d(t), r(t), \theta^{(1)}, \ldots, \theta^{(M)}]$. Moreover, the matrices H, F, G, W, S of (4.29) can be obtained by straightforward algebraic manipulations, see e.g. Borrelli (2003). Parameters of the PWA function $U^*(\xi)$ in (4.30), i.e., gains F_i, g_i and polyhedra $\mathcal{R}_i$, can be obtained by a parametric programming solver implemented in the freely-available MPT toolbox Kvasnica et al. (2004).

Remark 4.2.6 *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*

$$u^*(t) = [\,1\ 0\ \cdots\ 0\,]\, U^*(\xi) = \tilde{F}_i\xi + \tilde{g}_i, \text{ if } \xi \in \mathcal{R}_i, \tag{4.31}$$

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

Even though the explicit representation of the MPC feedback law in (4.31) 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 (4.29). Therefore, from a practical point of view, explicit MPC solutions as in (4.30) can only be obtained for reasonably simple QP problem (4.29). Note that our QP has $2N(M+1)$ constraints, N decision variables $(u_0, \ldots, u_N)$, and $8+3M$ parameters (4 initial states $x(t)$, 3 initial disturbances $d(t)$, 1 center of the thermal comfort zone $r(t)$ and M samples $\theta^{(i)}$, each of which is a 3×1 vector). Since $M \gg N$ in practice due to (4.25), the main driving factor of complexity is thus M, the number of realizations of the random variable θ employed in (4.27).

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 4.2.2). 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 4.2, with highlighted above mentioned setup. Then we have $M = 919$ by (4.25), hence the QP in (4.27) has 18400 constraints and 927 parametric variables. Solving such a QP parametrically according to Theorem 4.2.5 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.

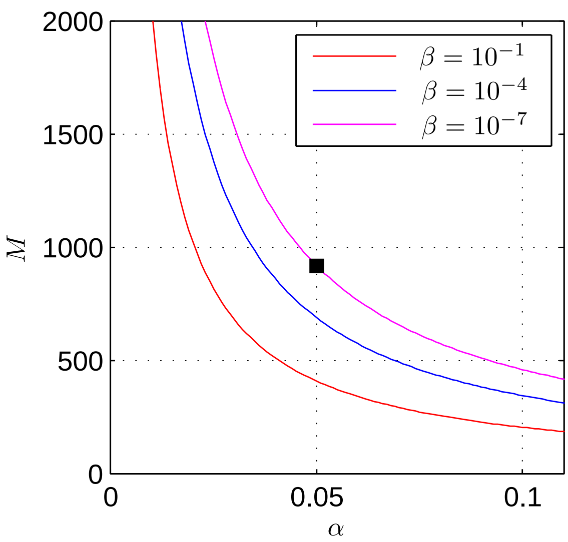


Figure 4.2: Dependance of number of M samples $\theta^{(i)}$ on parameter α, for three different settings of parameter β. Where point depicted as black square represents $M = 919$ samples for $\alpha = 0.05$, $N = 10$, and $\beta = 1 \cdot 10^{-7}$.

Fortunately, most of the constraints are redundant and can hence be discarded, allowing a tractable solution. To see this, consider the constraint in (4.27c), rewritten as

$$C(Ax_k + Bu_k + E(d_0 + k\theta^{(i)})) \leq r + \epsilon. \tag{4.32}$$

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

$$\max_i \{C(Ax_k + Bu_k + E(d_0 + k\theta^{(i)}))\} \leq r + \epsilon, \tag{4.33}$$

which is furthermore equivalent to

$$C(Ax_k + Bu_k + Ed_0) + k \max_i \{CE\theta^{(i)}\} \leq r + \epsilon. \tag{4.34}$$

Similarly, we have that (4.27b) holds if and only if

$$C(Ax_k + Bu_k + Ed_0) + k \min_i \{CE\theta^{(i)}\} \geq r - \epsilon. \tag{4.35}$$

Let

$$\overline{\theta} = \arg\max_{\theta^{(i)}} \{CE\theta^{(i)}\},\ \underline{\theta} = \arg\min_{\theta^{(i)}} \{CE\theta^{(i)}\}. \tag{4.36}$$

Please note that $\overline{\theta}$ and $\underline{\theta}$ are computed over finite set of $\theta^{(i)}$, therefore no optimization is required to determine it's values. Then for any sample $\theta^{(i)}$ with $\underline{\theta} \prec \theta^{(i)} \prec \overline{\theta}$ the constraints in (4.27b)-(4.27c) are redundant. We conclude that, instead of considering M samples $\theta^{(i)}$ in (4.27), one can equivalently state the problem using only the extremal realizations $\underline{\theta}$, $\overline{\theta}$, 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 (4.29) can be obtained rather easily.

Remark 4.2.7 *Note that the values $\underline{\theta}$ and $\overline{\theta}$ are considered as free parameters in (4.27). Since the samples $\theta^{(i)}$ 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 (4.27) is formulated using symbolic initial conditions x_0, d_0, r, $\underline{\theta}$ and $\overline{\theta}$, 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 (4.31) 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 $x(t)$, $d(t)$, $r(t)$ and obtain $\sigma(t)$.
2. Generate M samples $\theta^{(1)}, \ldots, \theta^{(M)}$ from (4.7).
3. From the generated samples pick $\underline{\theta}$ and $\overline{\theta}$ by (4.36).
4. Set $\xi = [x(t), d(t), r(t), \underline{\theta}, \overline{\theta}]$ 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 $t + T_s$.

Remark 4.2.8 *Identification of* $\underline{\theta}$ *and* $\overline{\theta}$ *in* (4.36) *does not require any optimization, as the minima/maxima are taken element-wise from a finite set.* □

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 (4.31) in the memory is linear in R.

4.3 Simulation Study

4.3.1 Deterministic MPC Performance

Purpose of this section is validate performance of various MPC formulations proposed in Section 4.2.2 and to compare them to a benchmark PI controller of the form

$$u(t) = Ke(t) + \frac{1}{T_i}\int e(t)\mathrm{d}t, \tag{4.37}$$

where $e(t) = x_4(t) - r(t)$ is the tracking error. Each controller was then validated by performing a closed-loop simulation over $2.6784 \cdot 10^6$ seconds (which corresponds to 30 days), employing the continuous-time linear building model (4.1). Historical data provided by the ISE tool were used as values of the disturbances. Visualization of the trend of $d_1(t)$, the external temperature, is shown in Fig. 4.3. The MPC controllers were implemented as a Simulink s-function where the optimization problems (3.1) were formulated in YALMIP Löfberg (2004) and solved by GUROBI Gurobi Optimization (2012). Prediction horizon was set to $N = 10$ multiples of the sampling period $T_s = 444$ seconds. At each time instant t, the MPC controllers were provided only the value of disturbances available at that time, i.e., the prediction did not explicitly account for future values of disturbances. Initial indoor temperature was set to 10°C in each simulation. For each controller we have subsequently recorded closed-loop profiles of state and input variables. Performance of each controller is then judged based on three criteria:

C1: overall consumption of heating/cooling energy, expressed in kilowatt hours;

C2: thermal comfort, expressed as percentage of sampling instants at which the indoor temperature is kept in the comfort zone (4.14);

C3: computational power required to compute control inputs, represented by the average time needed to solve the optimization in (3.1).

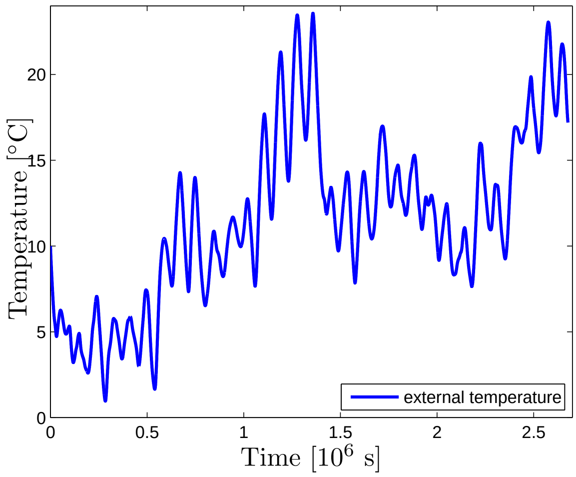
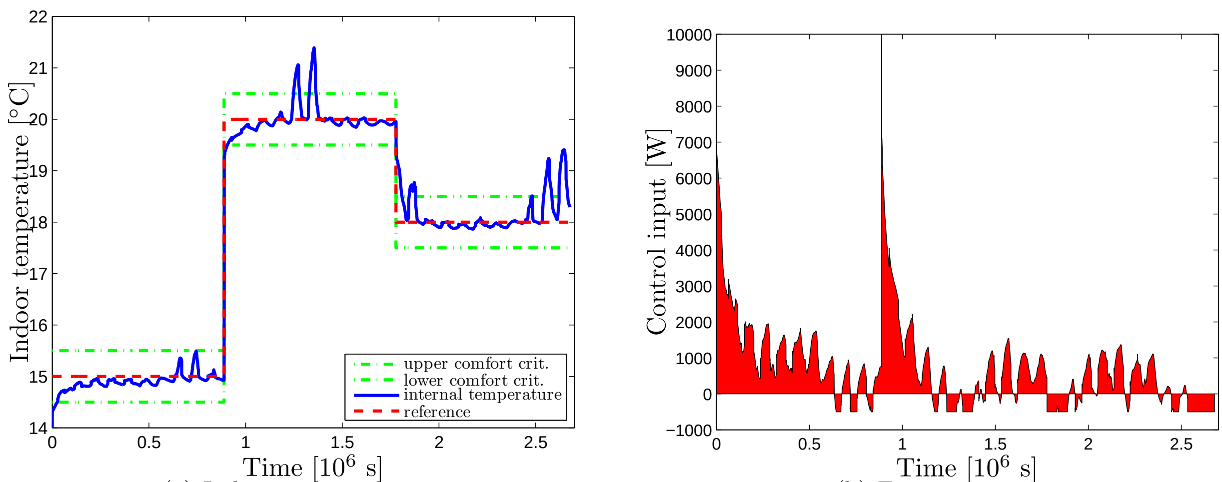


Figure 4.3: Historical trend of the external temperature over 30 days.

Standard PI control

The first investigated setup was represented by a PI controller (4.37). Closed-loop profiles recorded for this controller are shown in Fig. 4.4, where Fig. 4.4(a) shows capabilities of the controller to track a particular profile of the setpoint. It is worth pointing out that large peaks in indoor temperature around times $1.3 \cdot 10^6$ and $2.5 \cdot 10^6$ seconds are due to large variation of the external temperature, cf. Fig. 4.3. Control inputs generated by (4.37), i.e., the input energy flows, are reported in Fig. 4.4(b). Here we remark that in order to achieve satisfaction of input constraints in (4.3), the output of the PI controller was manually saturated. Performance of this control strategy, as evaluated by the three criteria mentioned above, is captured in the first line of Table 4.1.



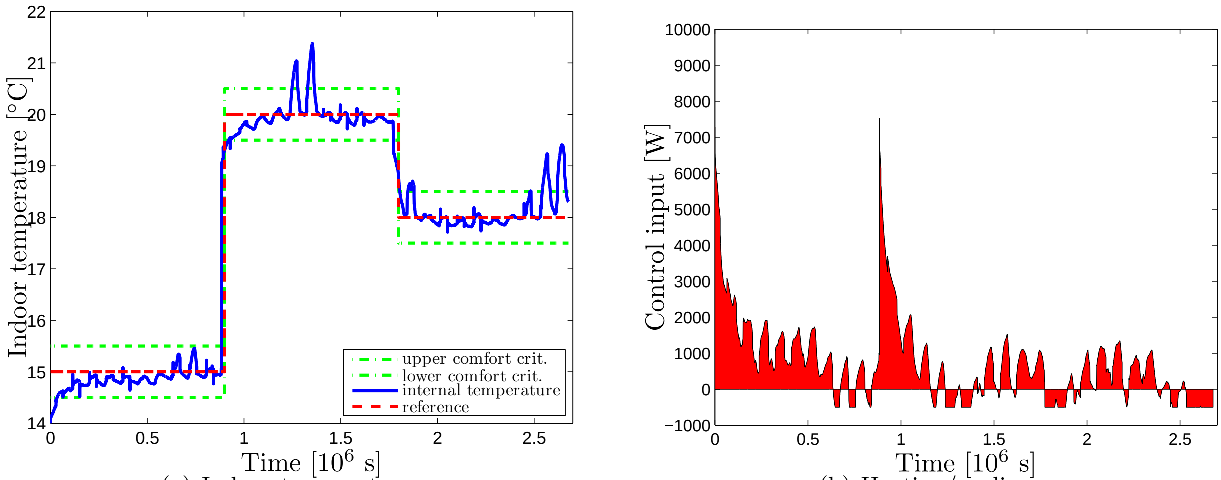
(a) Indoor temperature.

(b) Energy input.

Figure 4.4: Performance of the PI controller (4.37).

Setpoint temperature MPC

Next we have evaluated the reference tracking formulation with minimization of input energy, as proposed in Section 4.2.2. Here, the stage cost (4.12) was employed in (3.1). Performance of the scheme is shown in Fig. 4.5. Evaluating the closed-loop profiles showed that this particular MPC controlled saved 4 % of energy compared to the classical PI control. Moreover, the achieved thermal comfort was 89.2 %, an increase by 1.5 % compared to the PI case. This shows that MPC formulation achieves better reference tracking.

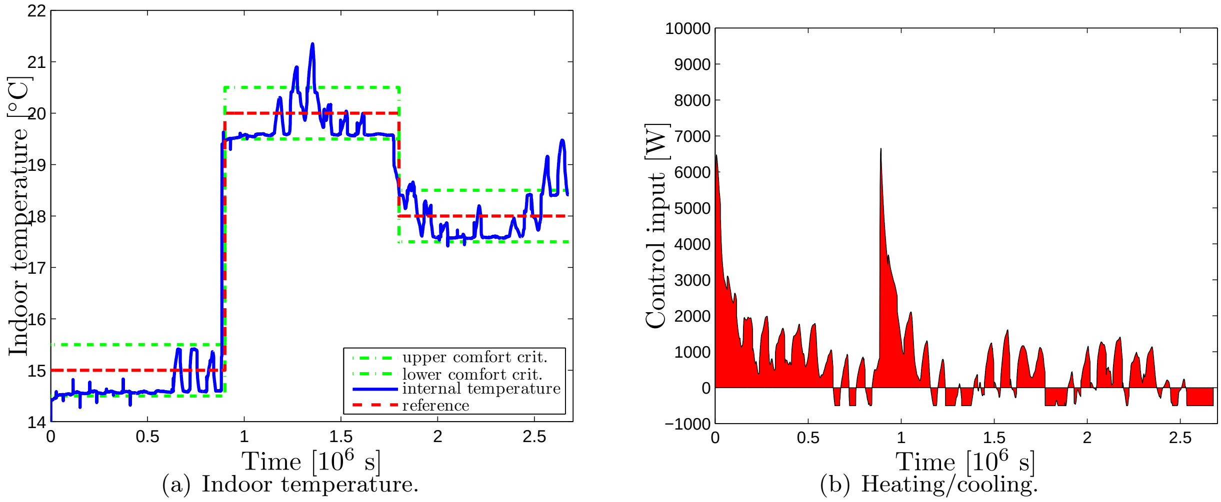


(a) Indoor temperature. (b) Heating/cooling.

Figure 4.5: Performance of the setpoint temperature MPC controller from Section 4.2.2.

Comfort zone temperature MPC

Performance of the MPC procedure from Section 4.2.2 is reported in Fig. 4.6. Here, instead of tracking the reference signal, the controller was only interested in keeping the indoor temperature within the comfort zone, whose width was set to $\epsilon = 0.5$°C. In accordance to intuition, such a zone-tracking approach allowed to further reduce amount of heating/cooling energy injected into the system. Specifically, a 9.1 % reduction of energy versus the PI benchmark was achieved, up from a 4.0 % energy reduction achieved by the previous MPC formulation, cf. the third line of Table 4.1. However, compared to the strategy of Section 4.2.2, this MPC controller leads to a lower thermal comfort (84.1 % compared to 89.2 % achieved with the strategy of Section 4.2.2). This is a consequence of the controller operating closer to the boundaries of the thermal comfort zone. Hence, when a disturbance hits the building (and because forecasts of the disturbances are not accounted for in (3.1)), there is a higher chance that the indoor temperature will be forced out of the zone. The formulation in Section 4.2.2, on the other hand, operates closer to the middle of the zone. Therefore the thermal comfort is less likely to be influenced by unmodeled disturbances.



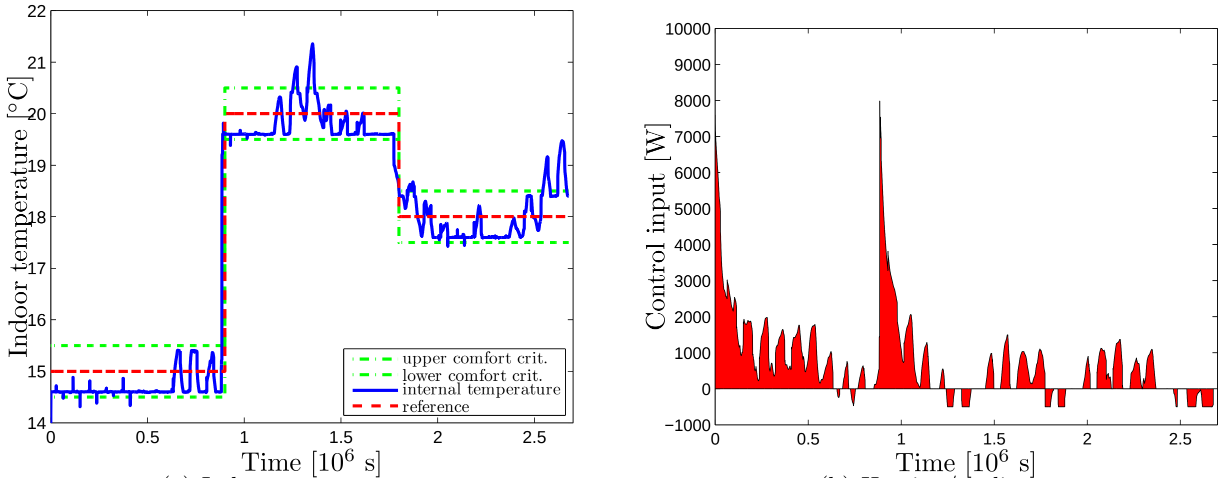
(a) Indoor temperature.

(b) Heating/cooling.

Figure 4.6: Performance of the comfort zone MPC controller from Section 4.2.2.

Minimization of Comfort Zone Violations MPC

Finally we have evaluated the MPC approach of Section 4.2.2 which uses binary variables to model violations of the comfort zone. As can be shown from the closed-loop profiles in Fig. 4.7, such an approach leads to a significant reduction of energy consumption while providing good comfort to inhabitants of the building. Specifically, compared to the PI benchmark, the procedure of Section 4.2.2 allowed to save 15 % of energy, while improving the thermal comfort criterion from 87.5 % to 88.2 %. Concrete figures are reported in final line of Table 4.1. Worth noting is a notable improvement compared to the results of comfort zone temperature MPC fro Section 4.2.2.



(a) Indoor temperature.

(b) Heating/cooling.

Figure 4.7: Performance of the MPC controller minimizing the comfort zone violations defined in Section 4.2.2.

Performance Comparisson and Computational Costs

To evaluate the computational criterion C3, for each deterministic MPC strategy we have measured the average time needed to obtain the optimal control inputs by solving the optimization problem (3.1) with stage cost and additional constraints as elaborated in Sections 4.2.2 to 4.2.2. Obtained results are summed up in Table 4.2. Confirming the conclusions of Section 4.2.2, the two MPC strategies of Sections 4.2.2 and 4.2.2, which can be formulated as quadratic optimization problems, can be solved very efficiently in a fraction of a second per one optimization. The approach of Section 4.2.2, on the other hand, requires solving mixed-integer problems. However, with the prediction horizon set to $N = 10$, even such a task can be accomplished at minor computational expenses. Note that in each case the runtime per sampling instant is well below the sampling period $T_s = 444$ seconds.

Table 4.1: Qualitative comparison of various control strategies.

Control strategy	Energy consumption [kWh]	Energy savings [%]	Thermal comfort [%]
PI controller	753.0	-	87.5
Setpoint MPC 4.2.2	722.7	4.0	89.2
Comfort zone MPC 4.2.2	684.0	9.1	84.1
Zone violations MPC 4.2.2	640.1	15.0	88.2

Table 4.2: Runtime of the optimization (3.1).

Control strategy	Average time per sample
Setpoint MPC 4.2.2	0.0117 seconds
Comfort zone MPC 4.2.2	0.0094 seconds
Zone violations MPC 4.2.2	0.0119 seconds

4.3.2 Explicit Stochastic MPC Performance

Comfort Zone Temperature Control

To validate performance of the stochastic explicit MPC strategy derived per Section 4.2.3, we have assumed 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. 4.9.

The explicit representation of the stochastic MPC feedback strategy (4.31) was obtained by formulating (4.27) in YALMIP Löfberg (2004) and solving the QP parametrically by the

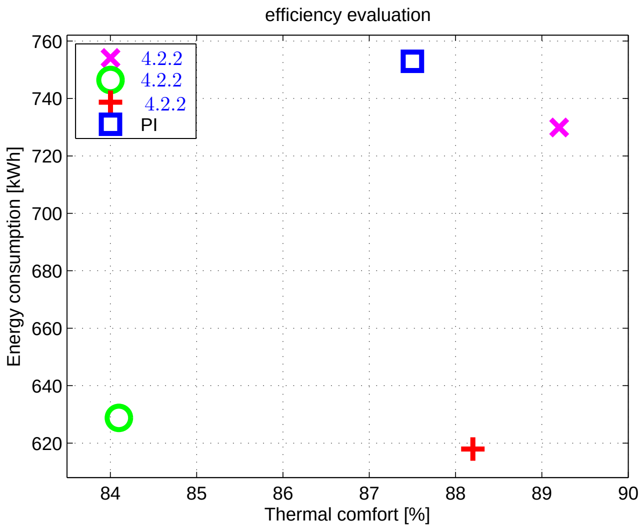


Figure 4.8: Comparison of all four investigated control approaches with respect to achievable thermal comfort (x-axis) and energy consumption (y-axis). Each point represents aggregated performance of one controller.

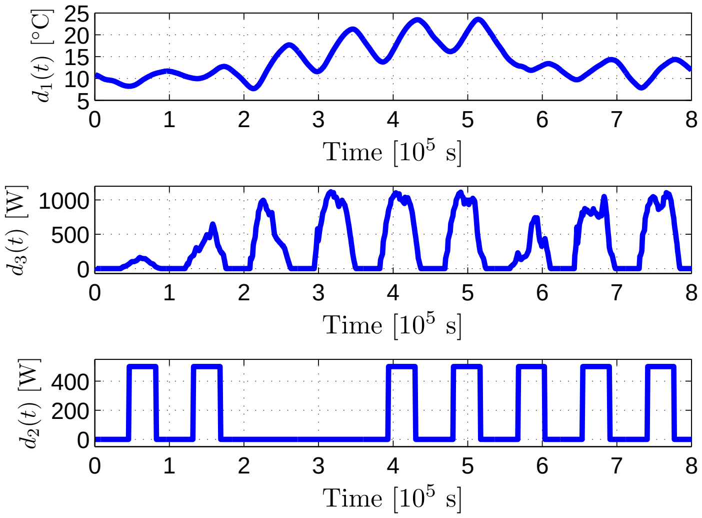


Figure 4.9: Historical trends of disturbances over 9 days. From top to bottom: external temperature d_1, solar radiation d_3, and heat generated by occupancy d_2.

MPT toolbox. The feedback law covers following ranges of initial conditions:

$$
\begin{aligned}
-10\,^\circ\mathrm{C} \leq x_i(t) &\leq 35\,^\circ\mathrm{C}, \;\; i = 1, \dots, 4, \\
15\,^\circ\mathrm{C} \leq r(t) &\leq 25\,^\circ\mathrm{C}, \\
0\,^\circ\mathrm{C} \leq d_1(t) &\leq 24\,^\circ\mathrm{C}, \\
0\,\mathrm{W} \leq d_2(t) &\leq 500\,\mathrm{W}, \\
0\,\mathrm{W} \leq d_3(t) &\leq 1200\,\mathrm{W}.
\end{aligned}
$$

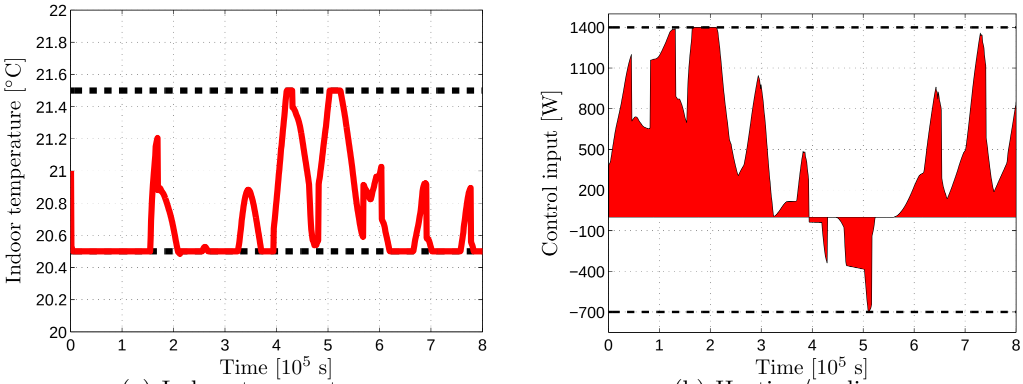
Width of the thermal comfort zone was set constantly to $\epsilon = 0.5\,^{\circ}\mathrm{C}$. With the prediction horizon $N = 10$, sampling time $T_s = 890$ seconds, $\alpha = 0.05$, and $\beta = 1 \cdot 10^{-7}$, the explicit MPC feedback (4.31) was obtained as a PWA function that consisted of 816 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 $T_s = T/15$. Simulated closed-loop profiles of the indoor temperature and consumed heating/cooling energy are shown in Fig. 4.12. 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.

Best and Worst Case Performance Comparisson

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. 4.10 and Fig. 4.11, 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 4.3 and captured in compact grapicall form in Fig. 4.13. 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.

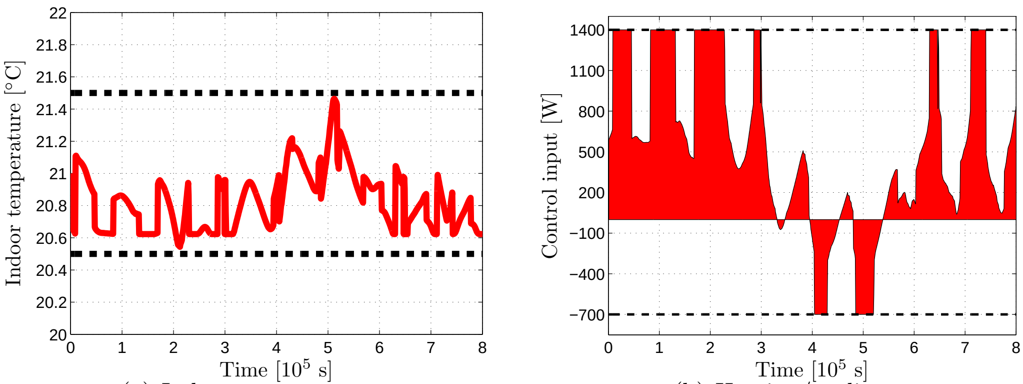
Table 4.3: Comparison of various strategies.

	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



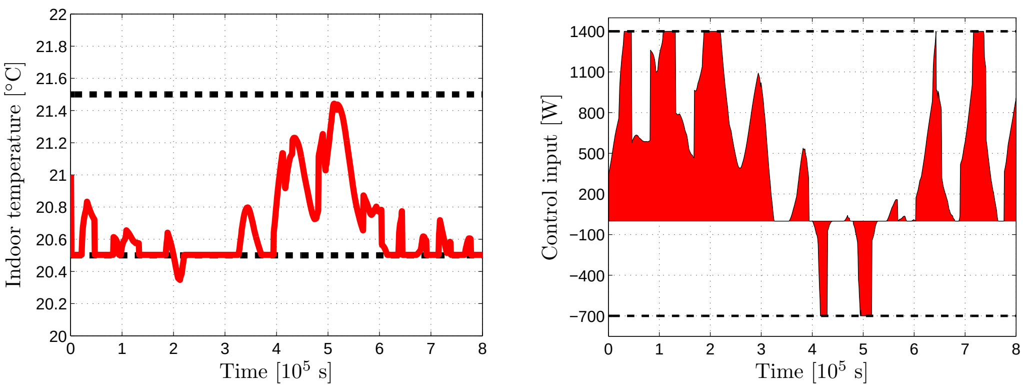
(a) Indoor temperature. (b) Heating/cooling.

Figure 4.10: Performance of the best-case MPC controller with complete knowledge of future disturbances.



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

Figure 4.11: Performance of the worst-case MPC controller which assumes conservative bounds on future disturbances.



(a) Indoor temperature. Dashed lines represent the thermal comfort zone. (b) Heating/cooling. Dashed lines denote limits of control authority in (4.3).

Figure 4.12: Performance of the stochastic MPC controller.

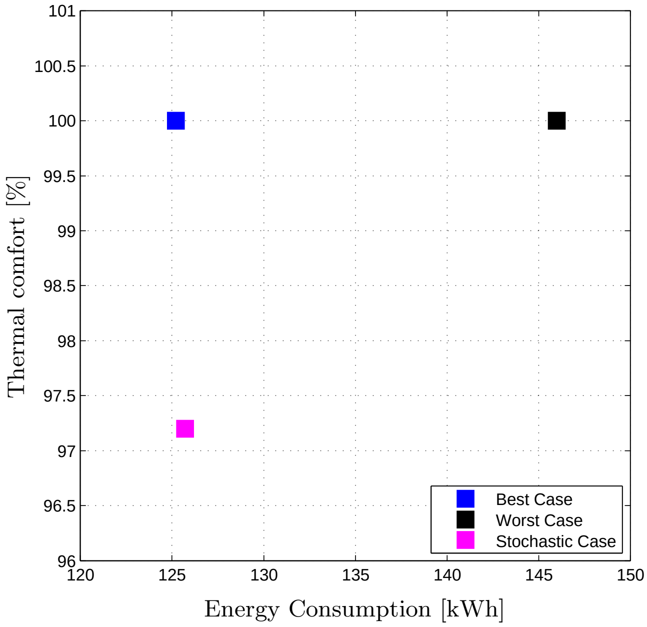


Figure 4.13: Comparison of all control approaches with respect to achievable thermal comfort (x-axis) and energy consumption (y-axis). Each point represents aggregated performance of one controller.

Chapter 5

Conclusions and Aim of the Thesis

Conclusion

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

For this purpose a supporting mathematical background was sumarised in standalone Chapter 2, presenting basic definitions on sets and functions 2.1, necessary for introduction of mathematical optimization 2.2, followed by brief overwiew of Probability and Statistics 2.3.

Chapter 3 is ment to be an introduction for MPC techniques, with its classification in control theory framework 3.1, brief historical evolution 3.2 in industrial enviroment, and basic features and standard formulations introduced in Section 3.3.

Building contol concepts are being subject of study in Chapter 4. Section 4.1 deals with building modeling as a necessary component for synthesis of MPC formulations. And finally particular MPC formulations for HVAC control problems are proposed in Section 4.2. Obtained results are supported by simulation case studies on building thermal comfort control problems presented in Section 4.3. All simulations were carried out on linear time-invariant (LTI) model of one-zone building from Section 4.1.3. For all MPC formulations proposed in this thesis a uniform control objectives are considered, and in Section 4.2.1 has been shown how to translate these criteria of comfort and low energy consumption into a corresponding mathematical form.

Three different formulations of an deterministic MPC optimization problem were investigated in Section 4.2.2. All proposed MPC strategies were then evaluated on simulations and compared to a classical approach based on a PI controller. Performance of control strategies investigated in Section 4.2.2 is compared graphically in Fig. 4.8 with respect to energy consumption and achievable level of comfort. As can be seen, all proposed MPC formulations clearly perform better than the PI alternative with respect to energy consumption. The least energy-efficient among them was the formulation from Section 4.2.2. Such a result is to be expected, since in Section 4.2.2 we try to follow the temperature setpoint as closely as possible, ignoring existence of the comfort zone. MPC from Section 4.2.2, on the other hand, only tries to keep the temperatures within the zone, which clearly allows to save energy. The most efficient formulation, however, was the approach based on zone tracking with binary indicators proposed, proposed in Section 4.2.2.

Further in Section 4.2.3 we have shown how to formulate the stochastic MPC optimization problem, and in addition how to derive an explicit representation of it. The control criteria for stochastic MPC have been kept the same as in in Section 4.2.2 for deterministic MPC formulations. The explicit stochastic approach was compared with conservative worst-case, and idealistic best-case scenarios. The comparisson of investigated control strategies is sumarised in Section 4.3.2, with compact graphicall form given in Fig. 4.13. 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 comparrison with unrealistic best-case MPC controller with given full knowledge of future disturbances over whole prediction horizon.

Aim of the Thesis

Future goals of disertation thesis from general point of view are deeper study of new digital age major topics, their structure, chaining, differences and interconnections in wider context. The outgoing result should be the examination of practical methods and technologies in industrial and practical applications, with backgrounding tehoretical knowledge, and their development historical context. The interested topcis are listed as follows:

- Methodology of science
- Mathematics
- Cybernetics
- Informatics
- Systems theory
- Control theory
- Automation and control engineering
- Chemical engineering
- Civil engineering
- Artifical inteligence
- Intelligent buildings
- Internet of things
- Technology

- Industrial applications
- Futurology and globalization

Academic goals with emphasis on practical applications can be formulated more specificly and sumarised as follows:

- Comprehensive research in field of building automation, and relevance evaluation of 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 integrated BAS.

Bibliography

L. Soethout A. Yahiaoui, J.L.M. Hensen. 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. 50

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. 22

Shmuel Agmon. The relaxation method for linear inequalities,. *Canadian Journal of Mathematics 6: 382-392, doi:10.4153/CJM-1954-037-2*, 1954. 23

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. 59

Badgwell T. A. Qin S. J. Rawlings J. B. & Wright S. J. Allgower, F. Nonlinear predictive control and moving horizon estimation-an introductory overview. *Advances in control: highlights of ECC, Berlin: Springer*, 1999. 48

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. 22

Karl J. Astrom and Richard M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers.* 2012. 35

Mordecai. Avriel. *Nonlinear Programming: Analysis and Methods.* Dover Publications. ISBN 0-486-43227-0, 2003. 23

R. Bălan, J. Cooper, K. Chao, S. Stan, and R. Donca. Parameter identification and model based predictive control of temperature inside a house. *Energy and Buildings*, 43(2):748–758, 2011. 51

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. 22

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. 50

C. Kirches S. Leyffer J. Linderoth J. Luedtke Belotti, P. and A. Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica 22:1-131.*, 2013. 25

A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, January 2002. 58, 61

Marcel Berger. *Geometry I,*. Berlin: Springer, ISBN 3-540-11658-3, 1987. 11

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. 50

D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods,*. Athena Scientific,, 1996. 16

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. 22

Camacho & Bordons. *Model predictive control.* 2004. 48

F. Borrelli. *Constrained Optimal Control of Linear and Hybrid Systems*, volume 290 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 2003. 58, 61

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 16, 17, 23

Michael Salciccioli Bradley Anderson, Aaron Bennick. University of michigan chemical engineering process dynamics and controls open textbook, 2014. URL `https://controls.engin.umich.edu/wiki/index.php/Main_Page`. 34

Maranas C. D. C. S. Adjiman, I. P. Androulakis and C. A. Floudas. A global optimisation method, alfa bb for process design. *Computers and Chemical Engineering*, 1996. 25

M. Campi and S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization*, 19(3):1211–1230, 2008. 58, 59

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. 9

Roger L. Casella G., Berger. *GStatistical Inference (2nd ed.).* Duxbury. ISBN 0-534-24312-6., 2001. 33

A.M. Morshedi C.E. Garcia. Quadratic programming solution of dynamic matrix control (qdmc). *Chemical Engineering Communications.*, pages 73–87, 1986. 42

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. 50

C. R. Cutler and B. L. Ramaker. Dynamic matrix control - a computer control algorithm. *AICHE national meeting, Houston, TX.*, 1979. 41

C. R. Cutler and B. L. Ramaker. Dynamic matrix control - a computer control algorithm. *In Proceedings of the joint automatic control conference.*, 1980. 41

Morshedi A. Cutler, C. and J. Haydel. An industrial perspective on advanced control. *In AICHE annual meeting, Washington, DC.*, 1983. 42

J.; Kuncova M. Dlouhy, M.; Fabry. *Simulace pro ekonomy.* Praha : VŠE, 2005. 32

C.E. Garcia D.M. Prett. Quadratic programming solution of dynamic matrix control (qdmc). *Fundamental process control Butterworths, Boston*, 1988. 43

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: http://dx.doi.org/10.1016/j.rser.2008.09.015. 9

J. Drgoňa and M. Kvasnica. Comparison of mpc strategies for building control. In Kvasnica M. Fikar, M., editor, *Proceedings of the 19th International Conference on Process Control*, pages 401–406, Štrbské Pleso, Slovakia, June 18-21, 2013 2013. Slovak University of Technology in Bratislava, Slovak University of Technology in Bratislava. URL `http://www.kirp.chtf.stuba.sk/publication_info.php?id_pub=1410`. 52

J. Drgoňa, M. Kvasnica, M. Klaučo, and M. Fikar. Explicit stochastic mpc approach to building temperature control. In *IEEE Conference on Decision and Control*, pages 6440–6445, Florence, Italy, 2013. 51, 52

US Energy. Building energy software tools directory, 2014. URL `http://apps1.eere.energy.gov/buildings/tools_directory/`. 50

EUROSIS. Directory of simulation software and tools, 2014. URL `http://www.eurosis.org/cms/?q=node/1318`. 50

M. Morari F. Borrelli, A. Bemporad. *Predictive Control for linear and hybrid systems.* 2014. 25

Lawrence Fristedt, Bert; Gray. *A modern approach to probability theory,.* Boston: Birkhäuser. ISBN 3-7643-3807-5., 1996. 30

J. B. Froisy and T. Matsko. Idcom-m application to the shell fundamental control problem. *AICHE annual meeting.*, 1990. 43

Alberto. Gallini. Affine function. from mathworld–a wolfram web resource, created by eric w. weisstein., 2014. URL http://mathworld.wolfram.com/AffineFunction.html. 13

Carlos E. Garcia, David M. Prett, and Manfred Morari. Model predictive control: Theory and practice - a survey. *Automatica*, 25(3):335 – 348, 1989. ISSN 0005-1098. doi: http://dx.doi.org/10.1016/0005-1098(89)90002-2. URL http://www.sciencedirect.com/science/article/pii/0005109889900022. 39, 48

M.E. Salgado G.C. Goodwin, S.F. Graebe. Control system design. *Prentice Hall, Englewood Cliffs, NJ.*, 2001. 40

A. Gelman. Exploratory data analysis for complex models,. In *Journal of Computational and Graphical Statistics, 13(4), 755âĂŞ779.*, 2004. 28

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. 58

Froisy B. Grosdidier, P. 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. 43

B. Grunbaum. *Convex Polytopes.* 2000. 11

Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2012. URL http://www.gurobi.com. 25, 56, 64

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. 9

Mosteller F. Hoaglin, D. C. and J. W. Tukey. Understanding robust and exploratory data analysis,. In *Wiley series in probability and mathematical statistics, New York*, 1983. 28

ILOG. 11.0 userâĂŹs manual. *ILOG CPLEX Division. Incline Village, NV*, 2007. 56

ILOG, Inc. *CPLEX 8.0 User Manual.* Gentilly Cedex, France, 2003. http://www.ilog.fr/products/cplex/. 25

INFORMS.org. About operations research, 2014. URL https://www.informs.org/About-INFORMS/What-is-Operations-Research. 25

K.R. Muske J.B. Rawlings. Stability of constrained receding horizon control. *IEEE Transactions on Automatic Control.*, pages 1512–1516, 1993. 44

Olav Kallenberg. *Random Measures (4th ed.),*. Berlin: Akademie Verlag. ISBN 0-12-394960-2. MR MR0854102., 1986. 30

Olav Kallenberg. *Foundations of Modern Probability (2nd ed.).* Berlin: Springer Verlag. ISBN 0-387-95313-2., 2001. 30

R. E. Kalman. Contributions to the theory of optimal control. *Bulletin de la Societe Mathematique de Mexicana, 5, 102-119.*, 1960a. 39

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of ASME, Journal of Basic Engineering, 87, 35-45.*, 1960b. 39

John C. Doyle Kemin Zhou. *Essentials of Robust Control.* 1997. 38

K. Glover Kemin Zhou, J. Doyle. *Robust and Optimal Control.* 1995. 38

M. Klaučo and M. Kvasnica. Explicit mpc approach to pmv-based thermal comfort control. In *53rd IEEE Conference on Decision and Control*, volume 53, pages 4856–4861, Los Angeles, California, USA, December 15-17, 2014 2014. 51

M. Klaučo, J. Drgoňa, M. Kvasnica, and S. Di Cairano. 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*, pages 581–586, 2014. 51

Donald E. Knuth. *he Art of Computer Programming, Volume 2: Seminumerical Algorithms.* Addison-Wesley, Boston., 1998. 32

A. N. Kolmogorov. *Grundbegriffe der Wahrscheilichkeitsrechnung,*. Springer, Berlin, 1933. 29

& Cannon M. Kouvaritakis, B. *Nonlinear predictive control, theory and practice.* 2001. 48

A. Kusiak and G. Xu. Modeling and optimization of HVAC systems using a dynamic neural network. *Energy*, 2012. 9

M. Kvasnica. *Real-Time Model Predictive Control via Multi-Parametric Programming: Theory and Tools.* VDM Verlag, Saarbruecken, January 2009. 9, 56

M. Kvasnica, P. Grieder, M. Baotic, and M. Morari. Multi-Parametric Toolbox (MPT). In *Hybrid Systems: Computation and Control*, pages 448–462, March 2004. 61

Automatic Control Laboratory. Epfl, 2013. URL `http://la.epfl.ch/openBuild`. 50

J. H. Lee. Recent advances in model predictive control and other related areas. 1996. 48

Geoff J. Levermore. *Building Energy Management Systems: Applications to low-energy HVAC and natural ventilation control.* E & FN Spon, 2 edition, 2000. 9

J. T. Linderoth and T. K. Ralphs. Noncommercial software for mixed-integer lin- ear programming. 2005. 25

Simeng Liu and Gregor 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. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-34249299364&partnerID=40&md5=ee59607C 9

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/. 64, 68

A. Lyon. Why are normal distributions normal? *The British Journal for the Philosophy of Science.*, 2014. 33

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. 51, 52

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. 52

Yudong Ma, Francesco Borrelli, Brandon Hencey, Andrew Packard, and Scott 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: http://dx.doi.org/10.1109/CDC.2009.5400677. URL http://sites.google.com/site/mpclaboratory/people/yudong-ma/files/YudongMa-CDC09_final2. 9

Yudong Ma, Sergey Vichik, and Francesco 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. 51

J. M. Maciejowski. *Predictive control with constraints.* 2002a. 48

J.M. Maciejowski. *Predictive Control with Constraints.* Prentice Hall, 2002b. 9

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. 43

Mathworks. Matlab optimization toolbox documentation: writing constraints, 2014. URL http://www.mathworks.com/help/optim/ug/writing-constraints.html. 18

D. Q. Mayne. Nonlinear model predictive control: An assessment. *Fifth international conference on chemical process control AICHE and CACHE,*, pages 217–231, 1997. 48

Rawlings J. B. Rao C. V. & Scokaert P. O. M. Mayne, D. Q. Constrained model predictive control: Stability and optimality. *Automatica.*, 36:789–814, 2000. 48

Faye C. McQuiston, Jerald D. Parker, and Jeffrey D. Spitler. *Heating, Ventilating and Air Conditioning Analysis and Design.* Wiley, 6 edition, 2005. 9

Michael J. Messina, Sezai E. Tuna, and Andrew R. Teel. Discrete-time certainty equivalence output feedback: allowing discontinuous control laws including those from model predictive control. *Automatica*, 41(4):617 – 628, 2005. ISSN 0005-1098. doi: http://dx.doi.org/10.1016/j.automatica.2004.11.015. URL http://www.sciencedirect.com/science/article/pii/S0005109804003206. 52

Da-Wei Gu Mi-Ching Tsai. *Robust and Optimal Control. A Two-port Framework Approach.* 2014. 38

& Lee-J. H. Morari, M. Model predictive control: The good, the bad, and the ugly. *In Y. Arkun, W. H. Ray (Eds.), Chemical process control - CPC IV, Fourth international conference on chemical process control*, pages 419–444, 1991. 48

Kenneth R. Muske and James B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262–287, 1993. ISSN 1547-5905. 48

G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley and Sons, New York., 1988. 22

NEOS. University of wisconsin-madison, 2014. URL http://neos-guide.org/. 16, 17

NEOS-algorithms. University of wisconsin-madison, 2014. URL http://neos-guide.org/algorithms. 16

NEOS-software. University of wisconsin-madison, 2014. URL http://neos-guide.org/solver-software. 16, 25

NEOS-solvers. University of wisconsin-madison, 2014. URL http://www.neos-server.org/neos/solvers/index.html. 16, 25

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, 2010a. 51, 52

F. Oldewurtel, A. Ulbig, A. Parisio, G. Andersson, and M. Morari. Reducing peak electricity demand in building climate control using real-time pricing and model predictive control. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 1927–1932. IEEE, 2010b. 51

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. 51, 52

Pillai S. Unnikrishna Papoulis, Athanasios. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill Science/Engineering/Math. ISBN 0-07-281725-9., 2001. 30, 31

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. 9

J.B. Rawlings P.O.M. Scokaert. Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control.*, pages 1163–1169, 1998. 44

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. 41

S.Joe Qin and Thomas A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733 – 764, 2003. ISSN 0967-0661. doi: http://dx.doi.org/10.1016/S0967-0661(02)00186-7. URL http://www.sciencedirect.com/science/article/pii/S0967066102001867. 9, 39, 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. 48

Meadows E. S. & Muske K. Rawlings, J. B. Nonlinear model predictive control: a tutorial and survey. *In Proceedings of IFAC ADCHEM. Japan*, 1994. 48

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. 41

Rault A. Testud J. L. Richalet, J. 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, 41

A. Richards and J. How. Mixed-integer programming for control. *In American Control Conference, pages 2676-2683*, 2005. 25

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. 48

Ratto M. Andres T. Campolongo F. Cariboni J. Gatelli D. Saisana M. Saltelli, A. and S. Tarantola. *Global Sensitivity Analysis.* 2008. 25

Philip K. Schneider and David H. Eberly. *Geometric Tools for Computer Graphics.* Morgan Kaufmann. ISBN 978-1-55860-594-7., 2003. 11, 13

A. Schrijver. *Linear and Integer Programming,.* John Wiley and Sons, New York., 1984. 22

Postlethwaite I Skogestad S. *Multivariable Feedback Control, Analysis and Design.* 2007. 38

INFORMS Computing Society. The nature of mathematical programming, mathematical programming glossary, 2014. URL http://glossary.computing.society.informs.org/ver2/mpgwiki/index.php?title=Main_Page. 16

P. Tøndel, T. A. Johansen, and A. Bemporad. Evaluation of Piecewise Affine Control via Binary Search Tree. *Automatica*, 39(5):945–950, May 2003. 64

M. Trcka and J. L. M. Hensen. Overview of hvac system simulation. *Automation in Construction.*, 19(2):93–99, 2010. 50

Ph.D. Truong Xuan Nghiem. Green buildings: Optimization and adaptation, 2011. URL http://www.seas.upenn.edu/~cis800/software.html. 50

J. W. Tukey. Exploratory data analysis,. In *Reading, MA: Addison-Wesley.*, 1977. 28

A. van Schijndel. Integrated heat, air and moisture modeling and simulation in hamlab. In *IEA Annex 41 working meeting, Montreal, May*, 2005. 50

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. 9, 51, 52

R. Webster. *Convexity,*. Oxford, England: Oxford University Press,, 1995. 11

Eric W. Weisstein. The webs most extensive mathematics resource., 2014. URL http://mathworld.wolfram.com/e. 11

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, https://gaia. lbl. gov/bcvtb*, 2008. 50

H.P. Williams. *Model Building in Mathematical Programming.* John Wiley & Sons, Third Edition, 1993. 57

L.A. Wolsey. *Integer Programming,*. John Wiley and Sons, New York., 1998. 22

Allgower F. & Zheng. *Nonlinear model predictive control, progress in systems and control theory.* 2000. 48

Xin Zhou, Tianzhen Hong, and Da Yan. Comparison of building energy modeling programs: Hvac systems. 08/2013 2013. 50