Hierarchical distributed optimization and predictive control of a smart grid

Von der Universität Bayreuth
zur Erlangung des akademischen Grades eines

Doktor der Naturwissenschaften (Dr. rer. nat.)

genehmigte Abhandlung

vorgelegt von

Philipp Braun

aus Kaiserslautern

1. Gutachter: Prof. Dr. Lars Grüne
2. Gutachter: Prof. Dr. Matthias Gerdts

Tag der Einreichung: 19.05.2016
Tag des Kolloquiums: 12.09.2016

Abstract

English Version: The energy transition, from a centralized to a decentralized and sustainable power supply using small scale power plants, presents new challenges to the distribution grid provider who is responsible for maintaining the stability of the electricity network. Furthermore, the rapid uptake of power generation from residential photovoltaic panels and wind turbines, together with decreasing prices for residential storage devices, is likely to lead to a reorganization of the energy market. Thus, new procedures to ensure the overall network stability need to be developed, which are flexible with respect to the underlying network and scalable, to be able to handle the amount of data of a fast growing network of renewable energy producers.

To this end, we consider (distributed) model predictive control (MPC) and hierarchical distributed optimization algorithms. We examine a network of residential energy systems (RESs) where every resident is equipped with solar photovoltaic panels and local storage devices, i.e., each RES is consuming, generating, and storing power. The RESs are connected through a grid provider responsible for the stability of the overall network.

We propose three different hierarchical distributed optimization algorithms. The flexibility of the algorithms allows for a plug and play manner of implementation. Scalability is obtained by solving the optimization problems on the level of the RESs and not on the level of the grid provider. Furthermore, with respect to a specific centralized optimization problem, convergence of the distributed optimization algorithms to the central optimum can be proven. In addition, we show how distributed optimization can be used to obtain a real-time pricing scheme depending on the power supply and the power demand, in contrast to the static pricing schemes in current widespread use.

It is verified numerically that the properties of the open-loop solutions carry over to the closed-loop by embedding the distributed optimization algorithms in receding horizon schemes. The results are illustrated using a dataset on power generation and power consumption of residential customers of the company Ausgrid.

Deutsche Fassung: Die Energiewende, die von der Energieerzeugung aus zentralen Großkraftwerken zur Energiegewinnung aus dezentralen Kleinkraftwerken führt, stellt die Netzbetreiber vor neue Aufgaben bei der Sicherung der Netzstabilität. Der Anstieg der Stromerzeugung aus privaten Solarzellen und Windkraftanlagen zusammen mit dem immer größer werdenden Angebot an privaten Energiespeichern führt zu einer Umstrukturierung des Energiemarktes. Dies macht neue Methoden bei der Gewährleistung einer stabilen Stromversorgung erforderlich, die flexibel bezüglich des Netzwerkes einsetzbar sind und gleichzeitig gut skalierbar sein müssen, um die Datenmenge eines schnell wachsenden Netzwerkes aus erneuerbaren Energieerzeugern handhaben zu können.
Diese Arbeit setzt sich in diesem Kontext mit der modellprädiktiven Regelung (MPC), beziehungsweise mit der verteilten modellprädiktiven Regelung, und mit hierarchischer verteilter Optimierung auseinander. Hierfür betrachten wir ein Netzwerk aus Konsumenten, zusätzlich ausgestattet mit privaten Stromerzeugern und Stromspeichern, und verbunden über einen Netzbetreiber, der für die Stabilität des Netzwerkes verantwortlich ist.

Wir untersuchen drei verteilte Optimierungsalgorithmen, die Flexibilität und Skalierbarkeit gewährleisten, indem die Optimierung auf der Ebene der einzelnen Konsumenten und nicht bei dem Netzbetreiber stattfindet. Zudem weisen wir Konvergenz der verteilten Optimierung bezüglich einem zugehörigen zentralen Optimierungsproblem nach. Dabei zeigen wir auch, wie die verteilte Optimierung bei der Echtzeitpreisbildung verwendet werden kann und somit der Strompreis im Gegensatz zu den heute gängigen Festpreisen von Angebot und Nachfrage abhängt.

Dass sich die Ergebnisse des offenen Regelkreises der verteilten Optimierung auch auf den geschlossenen Regelkreis übertragen lassen, weisen wir numerisch mit Hilfe der modellprädiktiven Regelung nach. Für die Simulationen werden reale Daten über den Stromverbrauch und die Stromerzeugung einzelner Haushalte verwendet, die von der Firma Ausgrid veröffentlicht wurden.

Contents

Abstract **v**

1 Introduction **1**
1.1 Motivation and focus of this thesis 1
1.2 Outline and contributions of this thesis 2

2 Basic definitions on predictive control and convex optimization **7**
2.1 Basic definitions and discrete-time systems 7
2.2 Feasibility, admissibility and optimal control 9
2.3 MPC and the receding horizon principle 10
2.3.1 Centralized, distributed, and decentralized MPC 12
2.3.2 From stabilizing MPC to economic MPC 12
2.4 Convex optimization 13
2.4.1 Convex sets and convex functions 14
2.4.2 Properties of convex functions 16
2.4.3 Convex optimization 18

3 A network of residential energy systems **21**
3.1 The energy transition 21
3.2 The Ausgrid dataset 24
3.3 Network stability: the energy provider perspective 26
3.4 Modeling of a residential energy system 27
3.4.1 Residential energy systems using storage devices 27
3.4.2 Additional approaches in smart grid applications 33
3.5 Control architectures of the electricity grid 36

4 Predictive control in the context of smart grids **39**
4.1 Centralized and decentralized predictive control 39
4.1.1 The centralized control setting 39
4.1.2 The decentralized control setting 40
4.2 Performance metrics and the cost functional 41
4.2.1 Performance metrics of predictive control schemes 42
4.2.2 Extensions of the objective function 43

4.3 Comparison of centralized and decentralized control . . . 44
4.4 From the open-loop to the closed-loop solution . . . 45
4.4.1 Non-uniqueness of the optimizer $\mathbf{z}^\star$ . . . 45
4.4.2 Warm-start . . . 47
4.4.3 Robustness of model predictive control schemes . . . 48
4.5 Prediction of power consumption and power generation . . . 50
4.6 Numerical simulations . . . 50
4.6.1 Comparison of decentralized and centralized control . . . 50
4.6.2 The impact of the prediction horizon . . . 53
4.6.3 Controllable loads . . . 54
4.6.4 Robustness verification via Monte-Carlo simulations . . . 55
4.6.5 The computational complexity . . . 56

5 A cooperative distributed optimization algorithm 59
5.1 Assumptions and notations . . . 60
5.2 The hierarchical distributed optimization algorithm . . . 61
5.2.1 The distributed optimization algorithm . . . 61
5.2.2 Convergence of the distributed optimization algorithm . . . 63
5.3 Application to residential energy systems . . . 68
5.3.1 The communication structure of the distributed optimization algorithm 69
5.3.2 Numerical complexity of the distributed optimization algorithm . . . 71
5.4 Extension to non-convex optimization . . . 73
5.5 Numerical simulations . . . 76
5.5.1 Distributed MPC using hierarchical distributed optimization . . . 76
5.5.2 Distributed MPC for different model dynamics . . . 81

6 Relaxed distributed optimization using the dual gradient method 85
6.1 The Lagrangian function and duality . . . 87
6.2 The dual ascent method . . . 88
6.2.1 Definition and convergence of the dual ascent method . . . 88
6.2.2 The dual ascent algorithm . . . 89
6.2.3 The distributed dual ascent algorithm . . . 92
6.3 The dual ascent algorithm for a network of RESs . . . 94
6.3.1 Cooperative application of the dual ascent algorithm . . . 95
6.3.2 Price-based non-cooperative dual ascent application . . . 99
6.3.3 General properties of the (non-)cooperative control setting . . . 102
6.4 Numerical simulations . . . 106
6.4.1 The impact of the relaxation parameter δ . . . 106
6.4.2 Price-based MPC simulations . . . 107

7 The alternating direction method of multipliers 117
7.1 ADMM: problem formulation and convergence results . . . 118
7.1.1 Problem formulation . . . 118

7.1.2 Definitions and notations . . . 120
7.1.3 Convergence of the alternating direction of multipliers method . . . 121
7.1.4 Simplification of the ADMM formulation . . . 128
7.2 ADMM for a network of residential energy systems . . . 131
7.2.1 The hierarchical distributed optimization algorithm . . . 131
7.2.2 ADMM in the receding horizon context . . . 131
7.3 Situation-based control of a network of RESs . . . 131
7.3.1 Vertical fluctuations . . . 133
7.3.2 Temporary islanded operation of a network of RESs . . . 134
7.3.3 Peak-detection . . . 139
7.3.4 Power balance . . . 139
7.3.5 Time-varying tube constraints . . . 140
7.4 Numerical simulations . . . 141
7.4.1 Islanded operation of a microgrid . . . 141
7.4.2 Time-varying tube constraints . . . 144

8 Conclusions **147**
8.1 Comparison of the distributed optimization algorithms . . . 147
8.2 Future work . . . 148

Acronyms **150**

Glossary **151**

Bibliography **152**

Chapter 1

Introduction

1.1 Motivation and focus of this thesis

The energy transition, from centralized fuel-based power generation to decentralized power generation based on renewable energy sources, leading to a reorganization of the national energy markets, presents many challenges for electricity distribution networks designed for one-way power flow.

For example, small-scale rooftop solar photovoltaic (PV) distributed generation has seen dramatic growth in the last years. Over 70% of the 70 GW installed PV capacity in the European Union (EU) as of 2012, for example, was rooftop-mounted (both residential and commercial/industrial) [37].

As PV penetration levels increase, integrating solar PV into the grid creates potential problems for utilities and customers alike. Reverse power flow in the low-voltage network during daytime periods of peak PV generation coupled with low residential load leads to well-recognized increases in distribution feeder voltages (the so-called voltage rise problem), with the potential for adverse impacts on the operation of voltage control devices and the safety of customer-owned devices [54], [56].

One approach to mitigate the fluctuations in power demand and power generation in an electricity distribution network based on renewable energies is to consider decentralized storage devices, such as batteries or fuel cells, storing energy at times with low demand and providing additional energy at times with high demand. With recent advances in battery technology, widespread deployment of battery storage at the residential level, particularly as a complement to rooftop solar photovoltaics, is expected to occur over the next decade. Nevertheless, if the charging and the discharging is poorly scheduled the batteries can increase the technical challenges encountered by current electricity networks. This has led to a significant research effort in the area of battery scheduling; see [52, 54, 74, 79, 83] and the references therein. Model predictive control (MPC) or equivalently receding horizon control provides a natural tool in the context of battery scheduling where the future power demand and power generation can only be estimated up to some time into the future.

However, the size of electricity networks, the rapid change of individual components in the electricity network, and participants with possibly conflicting objectives and sensitive

data make the design of appropriate MPC algorithms challenging. The MPC schemes can range from a centralized approach which requires full communication of all system variables, to a distributed approach with limited communication of relevant system variables, to a decentralized approach requiring no communication at all. Unsurprisingly, centralized approaches achieve the best performance when compared with distributed and decentralized approaches, but in general, suffer from an inability to scale to large networks.
In this thesis, we investigate distributed optimization algorithms and distributed model predictive control (DiMPC) schemes with the goal of recovering the performance of the corresponding centralized model predictive control (CMPC) schemes whilst remaining scalable. In other words, we focus on the solution of single, finite time horizon, optimization problems implemented in a distributed fashion. The receding horizon properties of the distributed optimization algorithms are not investigated here but numerical simulations are provided for the closed-loop solutions.
At least in the control literature, the field of distributed optimization traces its roots to the thesis of Tsitsiklis [99] (see also [15]). Much of the recent work in this field has involved multi-agent systems trying to optimize a global objective function under different conditions, see for example [35, 55, 76, 77, 114] and the references therein. We investigate three distributed optimization algorithms: an algorithm based on the primal optimization problem, a dual decomposition algorithm, and an algorithm based on the alternating direction method of multipliers (ADMM). In the case of the dual decomposition algorithm, we show how the distributed optimization algorithm can be used to generate a real-time price signal for energy prices. A detailed description of the structure of the thesis is given in the following.

1.2 Outline and contributions of this thesis

Chapter 2 – Basic definitions on predictive control and convex optimization In this chapter, we introduce the necessary notions used throughout the thesis. In particular, we define time-varying discrete time systems, we introduce the MPC concept, and discuss convexity in the context of finite dimensional optimal control problems (OCPs).

Chapter 3 – A network of residential energy systems In this chapter, we provide several models of electricity distribution grids consisting of residential energy systems (RESs) coupled through their energy demand. The RESs, characterized as time-varying discrete-time systems, incorporate controllable storage devices and local power generation in the dynamics of the power demand profile of the distribution grid. The proposed models are compared to similar works; for example papers on demand-side management and papers about vehicle-to-grid installations, where the batteries of electric vehicles are used as residential storage devices.
The algorithms developed in this thesis are tested using data from a dataset containing the power consumption and the power generation of 300 residents recorded for the time span of one year. The dataset, which is used throughout the thesis, is introduced in this

chapter.
The chapter concludes by presenting different control architectures for distribution grids. We go from decoupled systems in a decentralized architecture via distributed architectures with communication to a completely coupled centralized setting controlled by a central entity.

Chapter 4 – Predictive control in the context of smart grids In this chapter, centralized model predictive control (CMPC) and decentralized model predictive control (DeMPC) algorithms for the system dynamics in Chapter 3 are presented. To be able to compare the CMPC and the DeMPC scheme, we define appropriate performance metrics and discuss possible extensions of the cost functional.
In the subsequent section, closed-loop properties of MPC schemes are discussed. In particular, we provide an example which shows that in our setting different optimal solutions of the open-loop OCP may lead to different closed-loop solutions with different performance properties. As a second property, we motivate the concept of warm-starts commonly used in MPC to reduce the computational complexity of the OCPs. In the last part of this section, we discuss robustness of MPC. We provide a framework to verify robustness of the MPC closed-loop solution with respect to disturbances in the model dynamics.
The chapter concludes with numerical simulations comparing the performances of CMPC and DeMPC based on the dataset provided in Chapter 3.

Chapter 5 – A cooperative distributed optimization algorithm In this chapter, a hierarchical iterative cooperative distributed optimization algorithm is proposed to solve a single convex OCP at a fixed time instant. The algorithm calls for communication between a coordinating central entity (CE) and the individual systems responsible for the optimization. A particular convex OCP with coupled cost functional is split into smaller subproblems of fixed size. The subproblems are solved by the individual systems in parallel and the CE communicates the information between the systems. We show that the solution recovers the solution of the original OCP if the algorithm is iterated until convergence.
Embedded in the MPC context, we obtain a distributed model predictive control (DiMPC) algorithm without the curse of dimensionality of CMPC but with CMPC performance properties. Applied to the network of RESs, the number of variables communicated by the CE is independent of the number of RESs, and moreover, no private data of the RESs is exchanged between the RESs via the CE.
The chapter concludes with numerical simulations showing the performance of the distributed optimization algorithm and the performance of DiMPC. Additionally, we illustrate the performance of a possible extension of the distributed optimization algorithm for non-convex optimization problems.

Chapter 6 – Relaxed distributed optimization using the dual gradient method In this chapter we propose a hierarchical distributed dual ascent algorithm to solve a relaxed version of the OCP considered in Chapter 5. The distributed dual ascent algorithm has the same communication structure as the cooperative algorithm proposed in Chapter 5,

but can be implemented in a non-cooperative way where every RES optimizes over its own costs.
The distributed dual ascent algorithm is interpreted as a price-negotiation mechanism between the CE and the RESs. While the CE sets electricity prices such that the fluctuations in the aggregated power demand are mitigated, the RESs use the flexibility of their storage devices to reduce their electricity costs. The results of the non-cooperative real-time pricing scheme are compared to the results of the cooperative setting used in Chapter 5.
The chapter concludes with numerical simulations in Section 6.4, illustrating the impact of the relaxed OCP replacing the original OCP. Moreover, the performance of the proposed scheme from the point of view of the CE and from the point of view of the RESs is investigated in the open-loop and in the closed-loop context.

Chapter 7 – The alternating direction method of multipliers In this chapter, we consider a third hierarchical distributed optimization algorithm, the alternating direction method of multipliers, for the computation of an optimal solution of a convex OCP. The proposed scheme is a cooperative algorithm similar to the one considered in Chapter 5, but in contrast to the algorithm considered in Chapter 5, ADMM is able to consider coupling constraints between the individual RESs.
Here, we examine different possible couplings between the RESs. In particular, we propose an OCP where the solution of the OCP yields the maximal time interval a network of RESs can operate in an islanded mode, i.e., without external supply from the main grid. Furthermore, we show how time-varying tube constraints on the aggregated power demand can be included in the OCP formulation.
In the MPC context, the ADMM scheme offers the possibility of changing the objective of the minimization problem at every time instant. This can be done by changing the minimization problem of the CE without changing the minimization problems of the RESs. Thus, we show how the islanded operation of the network of RESs and the minimization of the vertical deviations in the aggregated power demand can be embedded in a receding horizon scheme to optimize specific goals at a certain time instant.
Numerical simulations, illustrating the flexibility of the ADMM scheme, conclude the chapter. Here, closed-loop solutions of the islanded operation of the network of RESs and closed-loop solutions, including time-varying tube constraints in the OCP formulation, are provided.

Chapter 8 – Conclusions The thesis concludes with a comparison of the presented distributed optimization algorithms and with future research directions which are beyond the scope of this work but are a natural extension of the results presented here.

Publications The results of this thesis are partially based on the papers [108, 19, 21, 20, 18]. Contributions to the model of RESs, extended in Chapter 3, were made in [108] and [18]. The inclusion of controllable loads was discussed in [22]. Additionally, in the paper [108], centralized and decentralized MPC, and the robustness verification presented

in Chapter 4 are discussed. The distributed optimization algorithm given in Chapter 5 extends the algorithm presented in [19, 21]. Moreover, the warm-start properties (see Section 4.4.2) and the non-uniqueness of optimal open-loop trajectories (see Section 4.4.1) in the context of MPC are discussed in [21]. The ideas of the Market Maker mechanism for the price-based control approach in Chapter 6 are taken from [108], [20]. Chapter 7 contains results previously published in [18].
Additionally, the papers [24], [23], and [107, 110] originated from the work on this thesis. The results of these papers are not discussed in this thesis.

Chapter 2

Basic definitions on predictive control and convex optimization

This thesis focuses on distributed predictive control and distributed optimization in the context of smart grids represented by time-varying discrete-time systems. In this chapter, we provide the necessary background for these topics. We define discrete-time dynamical systems, introduce model predictive control (MPC) and discuss general properties of convex optimization.

2.1 Basic definitions and discrete-time systems

We consider $i \in \mathbb{N}_{\mathcal{I}} := \{1, \dots, \mathcal{I}\}$ (nonlinear) time-varying discrete-time control systems

$$x_i(k+1) = f_i(x_i(k), u_i(k)), \tag{2.1a}$$

$$z_i(k) = h_i(x_i(k), u_i(k); s_i(k)), \tag{2.1b}$$

where $x_i(k) \in \mathbb{R}^{n_i}$ is the state, $u_i(k) \in \mathbb{R}^{m_i}$ is the input, $s_i(k) \in \mathbb{R}^{d_i}$ is a time-varying parameter, $z_i(k) \in \mathbb{R}^p$ is the communication variable and $n_i, m_i, d_i, p \in \mathbb{N}$ are the dimensions for all $i \in \mathbb{N}_{\mathcal{I}}$ and for all $k \in \mathbb{N}$. The functions $f_i : \mathbb{R}^{n_i \times m_i} \to \mathbb{R}^{n_i}$ are time invariant while the functions $h_i : \mathbb{R}^{n_i \times m_i} \to \mathbb{R}^p$ are time-varying due to the sequences $\mathbf{s}_i(\cdot) := (s_i(k))_{k \in \mathbb{N}} \subset \mathbb{R}^{d_i}$ for all $i \in \mathbb{N}_{\mathcal{I}}$. Observe that we make the assumption that the dimensions of the state x_i, the input u_i and the parameters s_i depend on the individual system i, whereas the dimension of the communication variable is independent of the index $i \in \mathbb{N}_{\mathcal{I}}$. The time index is denoted by $k \in \mathbb{N}$ throughout this thesis. The variables z_i are called communication variables to express that they are used to exchange information between the systems $i \in \mathbb{N}_{\mathcal{I}}$. In contrast to z_i, the variables x_i and u_i, the sequences $\mathbf{s}(\cdot)$ and the functions f_i and h_i are in general only known to system $i \in \mathbb{N}_{\mathcal{I}}$.
The variables $y_i \in \mathbb{R}^{q_i}$, $q_i \in \mathbb{N}$, $i \in \mathbb{N}_{\mathcal{I}}$, are used to describe arbitrary variables throughout this thesis. For example, y_i can represent x_i, u_i or z_i ($i \in \mathbb{N}_{\mathcal{I}}$). To denote the trajectory of a time-dependent variable, we define

$$\mathbf{y}_i(\cdot) = (y_i(k))_{k \in \mathbb{N}} \subset \mathbb{R}^{q_i},$$

for $q_i \in \mathbb{N}$. If the overall system is considered then we define

$$y(k) := \begin{pmatrix} y_1(k) \\ \vdots \\ y_{\mathcal{I}}(k) \end{pmatrix}.$$

As a result, the overall dynamical system reads

$$x(k+1) = f(x(k), u(k)), \tag{2.2a}$$
$$z(k) = h(x(k), u(k); s(k)), \tag{2.2b}$$

where the dimensions of the variables x, u, z and s and the definitions of the function $f : \mathbb{R}^{n_1 \cdot \ldots \cdot n_{\mathcal{I}}} \times \mathbb{R}^{m_1 \cdot \ldots \cdot m_{\mathcal{I}}} \to \mathbb{R}^{n_1 \cdot \ldots \cdot n_{\mathcal{I}}}$ and $h : \mathbb{R}^{n_1 \cdot \ldots \cdot n_{\mathcal{I}}} \times \mathbb{R}^{m_1 \cdot \ldots \cdot m_{\mathcal{I}}} \to \mathbb{R}^{\mathcal{I}p}$ follow from the definitions of the individual systems. To simplify the notation, we set $n := n_1 \cdot \ldots \cdot n_{\mathcal{I}}$ and $m := m_1 \cdot \ldots \cdot m_{\mathcal{I}}$. In the context of MPC, we use the notation

$$\mathbf{y}_i(k; N) = (y_i(k), \ldots, y_i(k+N-1))$$

for finite trajectories at a fixed time $k \in \mathbb{N}$ and a given prediction horizon $N \in \mathbb{N} \cup \{\infty\}$. If the time index k and the prediction horizon N are clear from the context, we use the short notation $\mathbf{y}_i := \mathbf{y}_i(k; N)$. We make an exception in this notation for the state variables x_i, $i \in \mathbb{N}_{\mathcal{I}}$. For the state variables x_i, we define

$$\mathbf{x}_i := \mathbf{x}_i(k; N) = (x_i(k+1), \ldots, x_i(k+N)).$$

Specifically for the communication variables $z_i(k)$, $i = 1, \ldots, \mathcal{I}$, the average will be an important quantity. The average over the number of systems $\mathcal{I}$ is denoted by

$$\hat{y}(k) = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} y_i(k) \tag{2.3}$$

for all $k \in \mathbb{N}$. The definitions given for a single time instant k or for a single system i are also used for the general cases, i.e., $\mathbf{y}$, $\mathbf{y}(\cdot)$, $\hat{\mathbf{y}}$, $\hat{\mathbf{y}}(\cdot)$ are defined analogously.

Remark 2.1.1. *Note that the variable z_i should not be confused with the output y_i in standard definitions of dynamical systems.*[1] *Throughout this thesis, we assume that the state x_i is known or observable.*[2] *We distinguish the variables z_i from x_i to emphasize that x_i is only known to the system $i \in \mathbb{N}_{\mathcal{I}}$ whereas z_i might also be known to the other systems $\mathbb{N}_{\mathcal{I}} \backslash \{i\}$.*

[1] See, for example, [96].
[2] See, for example, [96, Chapter. 6] for observability.

2.2 Feasibility, admissibility and optimal control

In this section, we assume that the sequence $\mathbf{s}(\cdot)$ is fixed, i.e., given by exogenous data. The states and the inputs are in general constrained, i.e., $x_i \in \mathbb{X}_i$ and $u_i \in \mathbb{U}_i$ for given sets $\mathbb{X}_i \subset \mathbb{R}^{n_i}$ and $\mathbb{U}_i \subset \mathbb{R}^{m_i}$ for all $i \in \mathbb{N}_{\mathcal{I}}$. For a given control sequence $\mathbf{u}_i(k,N) \in \mathbb{U}_i^N$ and an initial state $x_{i,0} = x_i(k) \in \mathbb{X}_i$, we denote the solution of (2.1) by $\mathbf{x}_i^{u_i}(k;N)_{x_{i,0}}$ and $\mathbf{z}_i^{u_i}(k;N)_{x_{i,0}}$. The trajectory $\mathbf{x}_i^{u_i}(k;N)_{x_{i,0}}$ is called admissible if $\mathbf{x}_i^{u_i}(k;N)_{x_{i,0}} \in \mathbb{X}_i^N$ holds. For a given initial state $x_{i,0} \in \mathbb{X}_i$, a given time index $k \in \mathbb{N}$ and a given $N \in \mathbb{N} \cup \{\infty\}$, we define the set of all admissible control sequences

$$\mathbb{U}_i^{k,N}(x_{i,0}) = \left\{ \mathbf{u}_i(k;N) \subset \mathbb{U}_i^N \;\middle|\; \begin{array}{l} x_i(k) = x_{i,0}, \\ x_i(j+1) = f_i(x_i(j), u_i(j)) \in \mathbb{X}_i, \\ u_i(j) \in \mathbb{U}_i, \\ \quad \forall\, j = k, \ldots, k+N-1 \end{array} \right\},$$

i.e., $\mathbf{x}_i^u(k;N)_{x_{i,0}} \in \mathbb{X}_i^N$ for all $\mathbf{u}_i(k;N) \in \mathbb{U}_i^{k,N}(x_{i,0})$. (In the case $N = \infty$ we define $k + \infty - 1 := \infty$.) We assume that $\mathbb{U}_i^{0,\infty}(x_i) \neq \emptyset$ for all $x_i \in \mathbb{X}_i$ for all $i \in \mathbb{N}_{\mathcal{I}}$. Since the dynamics (2.1a) is time invariant, $\mathbb{U}_i^{k,N}(x_{i,0}) = \mathbb{U}_i^{0,N-k}(x_{i,0})$ holds for all $k \in \mathbb{N}$ and for all $N \in \mathbb{N} \cup \{\infty\}$. For the overall system we define the sets $\mathbb{X}^N = \mathbb{X}_1^N \times \ldots \times \mathbb{X}_{\mathcal{I}}^N$ and $\mathbb{U}^{k,N}(x_0) = \mathbb{U}_1^{k,N}(x_{1,0}) \times \ldots \times \mathbb{U}_{\mathcal{I}}^{k,N}(x_{\mathcal{I},0})$ for all $N \in \mathbb{N} \subset \{\infty\}$ analogously. Similarly, we can define the admissible sets in the variables $\mathbf{z}_i$ by

$$\mathbb{D}_i^{k,N}(x_{i,0}) = \left\{ \mathbf{z}_i(k;N) \in \mathbb{R}^{p \times N} \;\middle|\; \begin{array}{l} x_i(k) = x_{i,0} \\ x_i(j+1) = f_i(x_i(j), u_i(j)) \\ z_i(j) = h_i(x_i(j), u_i(j), s(j)) \\ u_i(j) \in \mathbb{U}_i, \quad x_i(j+1) \in \mathbb{X}_i \\ \qquad \forall j = k, \ldots, k+N-1 \end{array} \right\} \tag{2.4}$$

for all $i \in \mathbb{N}_{\mathcal{I}}$ and $\mathbb{D}^{k,N}(x_0) = \mathbb{D}_1^{k,N}(x_{1,0}) \times \ldots \times \mathbb{D}_{\mathcal{I}}^{k,N}(x_{\mathcal{I},0})$. Optimization problems can either be written in the unknowns $\mathbf{u}(k;N)$ or in the unknowns $\mathbf{z}(k;N)$. Depending on whether the representation $\mathbb{U}^{k,N}(x_0)$ or $\mathbb{D}^{k,N}(x_0)$ is favorable.

We define the feedback $\mu_i : \mathbb{R}^{n_i} \to \mathbb{R}^{m_i}$ as a mapping from the state space to the input variables for $i \in \mathbb{N}_{\mathcal{I}}$. The solutions of the closed-loop system

$$x_i^{\mu_i}(k+1) = f_i(x_i^{\mu_i}(k), \mu_i(x(k)) \tag{2.5}$$

are denoted as $\mathbf{x}_i^{\mu_i}(k;N)_{x_i(k)}$. The feedback is called admissible, if $\mathbf{x}_i^{\mu_i}(k;N)_{x_i} \in \mathbb{X}_i^N$ holds for $i \in \mathbb{N}_{\mathcal{I}}$. The same definitions are used for the overall system dynamics (2.2).

For a given running cost $\ell : \mathbb{R}^{\mathcal{I}p} \to \mathbb{R}$, and[3] a given $N \in \mathbb{N}$, we define the average cost functionals

$$J_N(x(k), \mathbf{u}(k;N)) := \frac{1}{N} \sum_{j=k}^{k+N-1} \ell(z^{\mathbf{u}}(j)_{x(k)}), \tag{2.6}$$

$$J_\infty(x(k), \mathbf{u}(\cdot)) := \limsup_{N \to \infty} J_N(x(k), \mathbf{u}(k;N)). \tag{2.7}$$

[3]The running costs can be extended to a mapping $\ell : \mathbb{R}^{\mathcal{I}p} \times \mathbb{R}^{n_1 \cdot \ldots \cdot n_{\mathcal{I}}} \times \mathbb{R}^{m_1 \cdot \ldots \cdot m_{\mathcal{I}}} \to \mathbb{R}$ which includes the states x and the input u, whenever this is suitable.

Observe that the index k in the input and in the initial state is necessary to implicitly define $\mathbf{s}(k;N)$. With the cost functional, we can define the optimal value function

$$V_N(x(k)) := \inf_{\mathbf{u}(k;N)\in\mathbb{U}^{k,N}(x(k))} J_N(x(k),\mathbf{u}(k;N)) \tag{2.8}$$

representing the minimal costs for a given initial state over a given horizon N.
For the cost functional (2.8), we assume that a minimum exists, i.e., we assume that a $\mathbf{u}^\star(k;N)\in\mathbb{U}^{k,N}(x(k))$ exists such that

$$V_N(x(k)) = J_N(x(k),\mathbf{u}^\star(k;N)) \tag{2.9}$$

holds. If it is possible to solve the infinite-horizon optimal control problem (OCP), i.e., to find

$$\mathbf{u}^\star(\cdot)\in \operatorname{argmin}_{\mathbf{u}(\cdot)\in\mathbb{U}^{k,\infty}(x(k))} J_\infty(x(k),\mathbf{u}(\cdot)), \tag{2.10}$$

then the solution $\mathbf{u}^\star(\cdot)$ is said to be optimal.[4] In addition, due to the dynamic programming principle,[5] $\mathbf{u}^\star(\cdot)$ defines a feedback in the case of an ideal model without disturbances.
However, it is in general not possible to solve the OCP (2.10) since it is infinite dimensional in the unknowns $\mathbf{u}^\star(\cdot)$. In addition, the OCP (2.10) requires full knowledge of $\mathbf{s}(\cdot)$ at any time step k, but in the application, which will be introduced in Chapter 3, we only have an approximation of the sequence $\mathbf{s}(\cdot)$ and the approximation is only available up to $\mathbf{s}(k;N)$. For this reason, we use model predictive control (MPC) instead of optimal control to iteratively compute a feedback by solving a finite OCP at every time instant k.

2.3 MPC and the receding horizon principle

Typically in the literature, model predictive control (MPC) is introduced via the receding horizon principle visualized in Figure 2.1. Instead of directly solving the infinite-dimensional OCP (2.10), we approximate its solution by using MPC, which constructs a feedback law by iteratively solving finite OCPs

$$\mathbf{u}^\star(k;N)\in \operatorname{argmin}_{\mathbf{u}(k;N)\in\mathbb{U}^{k,N}(x(k))} J_N(x(k),\mathbf{u}(k;N)) \tag{2.11}$$

at every time instant $k\in\mathbb{N}$ on a finite horizon $N\in\mathbb{N}$.
From the optimal solution $\mathbf{u}^\star(k;N)$, usually the first element[6] is used to define the feedback $\mu(k) := u^\star(k)$ before the horizon is shifted and the OCP of the time step $k+1$, i.e., to find

$$\mathbf{u}^\star(k+1;N+1)\in \operatorname{argmin}_{\mathbf{u}(k+1;N+1)\in\mathbb{U}^{k,N}(x^\mu(k+1))} J_N(x^\mu(k+1),\mathbf{u}(k+1;N+1)),$$

is solved. The corresponding control algorithm, which can be found in several monographs on MPC (see for example [66, 25, 85, 46]) is given in Algorithm 1.

[4] Optimal control problems are commonly defined as minimization problems and are not given in the form (2.10). In this thesis, an OCP can be a minimization problem if we are interested in the minimum value, or it can be given in the form (2.10) if we are interested in a minimizer of the corresponding minimization problem.

[5] See, for example, [46, Chapter 3.4].

[6] Different approaches using multiple elements of the solution $\mathbf{u}^\star(k;N)$ can be found in the literature on m-step MPC. See for example [80] and [24].

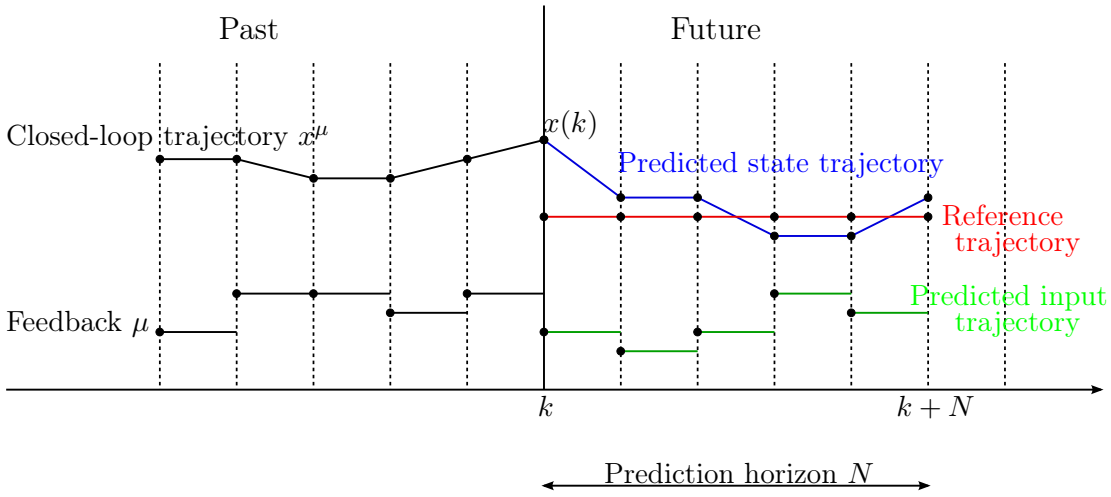

Figure 2.1: *Visualization of the receding horizon principle. At every time instant k a finite OCP over the prediction horizon N penalizing the deviation from a given reference trajectory is solved. The first part of the optimal trajectory is used as a feedback, then the procedure is repeated at time $k+1$.*

Algorithm 1 Model predictive control algorithm

1. Measure the current state $x(k) \in \mathbb{X}$.
2. Solve the OCP (2.11)

$$\mathbf{u}^\star(k;N) \in \operatorname{argmin}_{\mathbf{u}(k;N)\in\mathbb{U}^{k,N}(x(k))} J_N(x(k),\mathbf{u}(k;N))$$

   for a given prediction horizon $N \in \mathbb{N}$.
3. Define the feedback $\mu(x(k)) := \mathbf{u}^\star(k)$ and compute $x^\mu(k+1)$ and $z^\mu(k)$.

Shift the horizon by setting $k = k+1$ and go to step 1.

In many applications, the crucial step in the MPC Algorithm 1 is the efficient computation of a solution of the OCP (2.11). Depending on the definition of the system dynamics and the definition of the constraints, the OCP can be classified as either a continuous or a discrete optimization problem and, as either a convex or a non-convex optimization problem. In this thesis we focus on distributed optimization algorithms for convex optimization problems, i.e., we develop algorithms in Chapters 5 to 7, which exploit the decoupled structure of the system dynamics (2.1). In the context where the system dynamics (2.2) are decomposable into local system dynamics (2.1), one can distinguish among centralized model predictive control (CMPC), distributed model predictive control (DiMPC), and decentralized model predictive control (DeMPC) schemes.

2.3.1 Centralized, distributed, and decentralized MPC

The definition of the local system dynamics (2.1) for $i = 1, \dots, \mathcal{I}$ and the global system dynamics (2.2) indicate the possibility of different control schemes.
In CMPC an algorithm based on the overall dynamics (2.2) and an overall cost functional (2.6) is used. In contrast, in DeMPC the $\mathcal{I} \in \mathbb{N}$ decoupled systems compute a feedback solely on their local dynamics (2.1) and their local average cost functionals

$$J_{N,i}(x_i(k), \mathbf{u}_i(k;N)) := \frac{1}{N} \sum_{j=k}^{k+N-1} \ell_i(z_i^{\mathbf{u}_i}(j)_{x_i(k)}), \tag{2.12}$$

$$J_{\infty,i}(x_i(k), \mathbf{u}_i(\cdot)) := \limsup_{N\to\infty} J_{N,i}(x_i(k), \mathbf{u}_i(k;N)), \tag{2.13}$$

where $\ell_i : \mathbb{R}^p \to \mathbb{R}$, $i = 1, \dots, \mathcal{I}$, are independent of the behavior of the other systems.
Finally, DiMPC lies between DeMPC and CMPC. While the individual systems optimize on their own based on their local dynamics, the systems are coupled through the average cost functional[7] and hence the optimal costs $V_N(x(k))$ depend on the interaction of the solutions of the individual optimization problems $\mathbf{u}_i^\star(k;N)$. To obtain the optimal solution of the coupled cost functional, we assume that in DiMPC the systems can exchange information by means of the communication variables z_i between each other.
In Chapter 4, we investigate CMPC and DeMPC schemes in the context of smart grids. Distributed control will be the focus of the Chapters 5, 6 and 7. In particular, how the beneficial properties from CMPC and DeMPC carry over to DiMPC, will be of interest.

2.3.2 From stabilizing MPC to economic MPC

In the classical MPC literature, stabilizing MPC for system dynamics

$$x(k+1) = f(x(k), u(k))$$

with or without terminal costs and constraints are considered. In this case, one chooses a desired steady-state $(x_s, u_s) \in \mathbb{X} \times \mathbb{U}$, such that $x_s = f(x_s, u_s)$, and derives conditions on the cost functional (2.6), and consequently on the running costs ℓ, that ensure that the feedback obtained by Algorithm 1 drives an initial state $x(0)$ to the steady-state x_s. Moreover, the usual focus has been on performance estimates based on the prediction horizon N and on controllability assumptions, comparing the costs of the infinite horizon OCP

$$V_\infty(x(0)) = \inf_{\mathbf{u}(0;\infty)\in\mathbb{U}^{0,\infty}(x(0))} \sum_{j=0}^{\infty} \ell(x^{\mathbf{u}}(j), u(j))$$

[7] Distributed MPC is also considered in the context of coupled system dynamics and coupled/decoupled cost functionals. In this thesis, only the case of decoupled system dynamics is considered. See for example [85, Chapter 6] for a general DiMPC setting.

to the costs resulting from the MPC application

$$V_\infty^\mu(x(0)) = \sum_{j=0}^{\infty} \ell(x^\mu(j), \mu(x^\mu(j))) \tag{2.14}$$

for given positive semi-definite running costs $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ penalizing the deviation from the optimal pair (x_s, u_s).[8]

Depending on the application however, the running costs may be chosen according to other criteria rather than a design parameter to penalize the deviation from a desired optimal pair (x_s, u_s). This has led to the recent development of so-called economic MPC [32, 6, 7], where the existence of a (possibly unknown) optimal steady-state (x_s, u_s) for given running costs is assumed. Under suitable dissipativity[9] conditions of the system dynamics f, convergence results and performance properties of the economic MPC closed-loop solution towards the optimal steady-state (x_s, u_s) can still be guaranteed. The results in [32, 6, 7] using terminal costs and terminal constraints were extended in [43, 47, 45] to the so-called unconstrained MPC case. Recent advances on economic MPC and in particular on distributed economic MPC are summarized in the thesis [72].

In this thesis, we design MPC schemes based on the dynamics (2.2) and running costs $\ell : \mathbb{R}^{\mathcal{I}p} \to \mathbb{R}$ defined in the communication variables z. Hence, we are not seeking for a stabilizing controller but rather to optimize the performance with respect to a given criterion depending on z. Consequently, the proposed algorithms can be classified as economic MPC. Since we assume that the sequence $\mathbf{s}(\cdot)$, which influences the variable z, is only known up to some time into the future, the cost function is designed solely on the available knowledge and as a consequence, the running costs might even be time-dependent, i.e., for all $k \in \mathbb{N}$, $\ell_k : \mathbb{R}^{\mathcal{I}p} \to \mathbb{R}$ has to be defined based on $\mathbf{s}(k; N)$.

For time-varying (distributed) control systems, with time-varying running costs, performance estimates of economic MPC are out of the scope of this thesis. However, we use the values

$$V_\mathcal{N}^\mu(x(0)) = \sum_{j=0}^{\mathcal{N}} \ell(x^\mu(j), \mu(x^\mu(j))) \tag{2.15}$$

to compare the performance of centralized, distributed, and decentralized MPC schemes with different cost functions on a finite simulation of length $\mathcal{N} \in \mathbb{N}$ to obtain some insight into the performance of the proposed algorithms.

2.4 Convex optimization

As already discussed, for MPC algorithms at every time instant k, an OCP (2.11) has to be solved. Depending on the running costs ℓ, the system dynamics (2.2) defined through f and h, and the state and input constraint sets $\mathbb{X}$ and $\mathbb{U}$, respectively, the optimization problem can be classified as a convex or a non-convex optimization problem.

[8] See for example the monographs [85] and [46] for a comprehensive study of this topic.

[9] Dissipation of dynamical systems refers to the term characterized by Willems [105, 106].

Regardless of the classification of the OCP, by redefining the variables and the involved functions, the problem can be written in the form

$$\min_{\mathbf{y}\in D} \phi(\mathbf{y}) \tag{2.16}$$

for a function $\phi : \mathbb{R}^n \to \mathbb{R}$, a set $D \subset \mathbb{R}^n$ and the vector of unknowns $\mathbf{y}$.
Whereas in the context of non-convex optimization the convergence of an algorithm towards a global optimum can in general not be guaranteed, convex optimization problems possess several properties which are favorable for optimization algorithms. In this thesis we concentrate on distributed optimization algorithms for convex optimization problems. To this end, the notion of convexity is introduced and the properties of convex functions are highlighted in the following sections. Convexity in the context of optimization can be found for example in the monographs [14], [15] and [17].

2.4.1 Convex sets and convex functions

First, we define convex sets and convex functions.

Definition 2.4.1 (Convex set)**.** *A set $D \subset \mathbb{R}^n$ is called convex if for all $\mathbf{y}_1, \mathbf{y}_2 \in D$ and for all $\lambda \in (0, 1)$*

$$\lambda \mathbf{y}_1 + (1 - \lambda)\mathbf{y}_2 \in D$$

holds.

Geometrically, a set D is convex if and only if for two arbitrary points $\mathbf{y}_1, \mathbf{y}_2 \in D$, the line segment connecting the two points lies in D. For functions, we introduce three notions of convexity.

Definition 2.4.2 (Convex functions)**.** *Let $\phi : D \to \mathbb{R}$ be a function defined on a convex set $D \subset \mathbb{R}^n$. The function ϕ is called*

(i) convex if

$$\phi(\lambda \mathbf{y}_1 + (1 - \lambda)\mathbf{y}_2) \leq \lambda\phi(\mathbf{y}_1) + (1 - \lambda)\phi(\mathbf{y}_2), \tag{2.17}$$

(ii) strictly convex if

$$\phi(\lambda \mathbf{y}_1 + (1 - \lambda)\mathbf{y}_2) < \lambda\phi(\mathbf{y}_1) + (1 - \lambda)\phi(\mathbf{y}_2), \tag{2.18}$$

(iii) strongly convex if there exists an $\alpha > 0$ such that

$$\phi(\lambda \mathbf{y}_1 + (1 - \lambda)\mathbf{y}_2) \leq \lambda\phi(\mathbf{y}_1) + (1 - \lambda)\phi(\mathbf{y}_2) - \frac{1}{2}\alpha\lambda(1 - \lambda)\left\|\mathbf{y}_1 - \mathbf{y}_2\right\|^2 \tag{2.19}$$

holds for all $\mathbf{y}_1, \mathbf{y}_2 \in D$, $\mathbf{y}_1 \neq \mathbf{y}_2$, and for all $\lambda \in (0, 1)$.

From the definition of convex functions one obtains the following implications.

Remark 2.4.3. *Given a convex set $D \subset \mathbb{R}$ and a convex function $\phi : D \to \mathbb{R}$, $D \subset \mathbb{R}^n$, we obtain the implications*

$$\phi \text{ strongly convex} \quad \Longrightarrow \quad \phi \text{ strictly convex} \quad \Longrightarrow \quad \phi \text{ convex.}$$

Along with the definition of convex functions, we additionally define concave functions.

Definition 2.4.4 (Concave functions)**.** *The function $\phi : D \to \mathbb{R}$, defined on a convex set $D \subset \mathbb{R}^n$, is called concave, if $-\phi$ is convex.*

Strict and strong concavity are defined analogously. Observe that there exist functions which are neither convex nor concave.[10]
For differentiable functions, there exist equivalent convexity characterizations based on the first and the second derivative.

Lemma 2.4.5. *Let $D \subset \mathbb{R}^n$ be convex. Let $\phi : D \to \mathbb{R}$ be continuously differentiable and let $\nabla\phi$ denote the gradient of ϕ.*

(i) ϕ is convex, if and only if

$$f(\mathbf{y}_2) \geq f(\mathbf{y}_1) + (\mathbf{y}_2 - \mathbf{y}_1)^T \nabla f(\mathbf{y}_1)$$

for all $\mathbf{y}_1, \mathbf{y}_2 \in D$,

(ii) If

$$f(\mathbf{y}_2) > f(\mathbf{y}_1) + (\mathbf{y}_2 - \mathbf{y}_1)^T \nabla f(\mathbf{y}_1)$$

for all $\mathbf{y}_1, \mathbf{y}_2 \in D$ with $\mathbf{y}_1 \neq \mathbf{y}_2$, then ϕ is strictly convex,

(iii) ϕ is strongly convex with parameter α, if and only if

$$(\nabla f(\mathbf{y}_2) - \nabla f(\mathbf{y}_1))^T (\mathbf{y}_2 - \mathbf{y}_1) \geq \alpha \|\mathbf{y}_2 - \mathbf{y}_1\|^2 \tag{2.20}$$

for all $\mathbf{y}_1, \mathbf{y}_2 \in D$.

Lemma 2.4.6. *Let $D \subset \mathbb{R}^n$ be convex. Let $\phi : D \to \mathbb{R}$ be twice continuously differentiable and let $\nabla^2\phi$ denote the Hessian of ϕ.*

(i) ϕ is convex, if and only if $\nabla^2\phi(\mathbf{y}) \succeq 0$ for all $\mathbf{y} \in D$, i.e., the matrix $\nabla^2\phi(\mathbf{y})$ is positive semidefinite for all $\mathbf{y} \in D$.

(ii) If $\nabla^2\phi(\mathbf{y}) \succ 0$ for all $\mathbf{y} \in D$, then ϕ strictly convex, i.e., $\nabla^2\phi(\mathbf{y})$ is positive definite for all $\mathbf{y} \in D$.

(iii) ϕ is strongly convex with parameter $\alpha > 0$, if and only if $\nabla^2\phi(\mathbf{y}) - \alpha I \succeq 0$ for all $\mathbf{y} \in D$, where I denotes the identity matrix.

[10] Consider for example the function $\phi : \mathbb{R} \to \mathbb{R}$ defined as $\phi(x) = \sin x$.

A proof of Lemma 2.4.5 and Lemma 2.4.6 is given in [15, Proposition A.39 – A.40] and in [14, Proposition B.5].

Example 2.4.7. *The function $\phi_1 : \mathbb{R} \to \mathbb{R}$, $\phi_1(\mathbf{y}) = \mathbf{y}$ is convex according to Definition (2.4.2) (i) but Inequality (2.18) is satisfied with equality for all $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^n$ and thus, ϕ_1 is not strictly convex. The function $\phi_2 : \mathbb{R} \to \mathbb{R}$, $\phi_2(\mathbf{y}) = e^{\mathbf{y}}$ is strictly convex, since $\nabla^2\phi_2(\mathbf{y}) = e^{\mathbf{y}} \succ 0$ for all $\mathbf{y} \in \mathbb{R}$. However, $e^{\mathbf{y}} \to 0$ for $\mathbf{y} \to -\infty$ and hence, there exists no parameter $\alpha > 0$ such that $e^{\mathbf{y}} - \alpha \succ 0$ for all $\mathbf{y} \in \mathbb{R}$. The function $\phi_3 : \mathbb{R} \to \mathbb{R}$, $\phi_3(x) = \mathbf{y}^2$ is strongly convex, since $\nabla^2\phi_3(\mathbf{y}) = 2$ and $\nabla^2\phi_3(\mathbf{y}) - \alpha \succeq 0$ holds for all $\mathbf{y} \in \mathbb{R}$ and $\alpha = 2$.*

2.4.2 Properties of convex functions

Convex functions defined on convex sets guarantee favorable conditions in the context of optimization problems with respect to the existence and uniqueness of local and global minima.

Definition 2.4.8. *Let $D \subset \mathbb{R}^n$ and $\phi : D \to \mathbb{R}$. Then $\phi(\mathbf{y}^\star) \in \mathbb{R}$ is called local minimum of ϕ, if there exists an $\varepsilon > 0$ such that*

$$\phi(\mathbf{y}^\star) \leq \phi(\mathbf{y}) \tag{2.21}$$

holds for all $\mathbf{y} \in D \cap B_\varepsilon(\mathbf{y}^\star)$.[11] If Equation (2.21) is satisfied for all $\mathbf{y} \in D$, then $\phi(\mathbf{y}^\star) \in \mathbb{R}$ is called global minimum of ϕ. For a local (global) minimum $\phi(\mathbf{y}^\star)$, $\mathbf{y}^\star \in D$ is called local (global) minimizer of the function ϕ. The minimum of a function is denoted by $\phi^\star = \phi(\mathbf{y}^\star)$.

Theorem 2.4.9. *Let $D \subset \mathbb{R}^n$ be closed and convex and let $\phi : D \to \mathbb{R}$ be twice continuously differentiable.*

(i) If ϕ is convex, then every local minimum of ϕ is also a global minimum.

(ii) If ϕ is strictly convex, then ϕ has at most one global minimum.

(iii) If ϕ is strongly convex, then the unique global minimum of ϕ is attained.

Proof. (i) Let $\mathbf{y}^\star \in D$ be a local minimizer. Assume that $\mathbf{y}^\star$ is not a global minimizer, i.e., there exists a $\mathbf{y} \in D$ such that $\phi(\mathbf{y}) < \phi(\mathbf{y}^\star)$. Along with the convexity of ϕ, this implies

$$\phi(\lambda\mathbf{y} + (1-\lambda)\mathbf{y}^\star) \leq \lambda\phi(\mathbf{y}) + (1-\lambda)\phi(\mathbf{y}^\star) < \lambda\phi(\mathbf{y}^\star) + (1-\lambda)\phi(\mathbf{y}^\star) = \phi(\mathbf{y}^\star)$$

for all $\lambda \in (0, 1]$. This contradicts the assumption that $\mathbf{y}^\star$ is a local minimum.

(ii) Assume there exist two global minima, $\phi(\mathbf{y}^\star) = \phi(\mathbf{y}^\sharp)$ with $\mathbf{y}^\star, \mathbf{y}^\sharp \in D$, $\mathbf{y}^\star \neq \mathbf{y}^\sharp$. Due to the convexity of D and the strict convexity of ϕ, it holds that $\frac{1}{2}(\mathbf{y}^\star + \mathbf{y}^\sharp) \in D$ and

$$\phi\left(\frac{1}{2}(\mathbf{y}^\star + \mathbf{y}^\sharp)\right) < \frac{1}{2}\phi(\mathbf{y}^\star) + \frac{1}{2}\phi\left(\mathbf{y}^\sharp\right) = \phi(\mathbf{y}^\star)$$

which contradicts the assumption that $\mathbf{y}^\star$ is a global minimizer.

[11] The set $B_\varepsilon(\mathbf{y}^\star)$ defines a ball with radius $\varepsilon > 0$ centered at $\mathbf{y}^\star$, i.e., $B_\varepsilon(\mathbf{y}^\star) = \{\overline{\mathbf{y}} \in \mathbb{R}^n | \|\overline{\mathbf{y}} - \mathbf{y}^\star\| < \varepsilon\}$.

(iii) If D is compact, then the global minimum of ϕ is attained. By (ii), the global minimum is unique. Hence, we assume that D is unbounded.

If $\nabla\phi(\mathbf{y}^\star) = 0$ then $\nabla^2\phi(\mathbf{y}^\star) \succeq \alpha I$ implies that $\mathbf{y}^\star$ is a local minimizer. Moreover, by (ii) $\mathbf{y}^\star$ is the unique global minimizer. Assume that ϕ does not attain its global minimum. Let $\mathbf{y} \in D$ be arbitrary and $\mathbf{y}_f \in D$ be arbitrary and fixed. Using the Taylor expansion we obtain the expression

$$\phi(\mathbf{y}) = \phi(\mathbf{y}_f) + \nabla\phi(\mathbf{y}_f)(\mathbf{y} - \mathbf{y}_f)^T + \frac{1}{2}(\mathbf{y} - \mathbf{y}_f)\nabla^2\phi(\xi)(\mathbf{y} - \mathbf{y}_f)^T$$

for a $\xi \in D$. With the characterization of strongly convex functions from Lemma 2.4.6 (iii), this provides the estimate

$$\phi(\mathbf{y}) \geq \phi(\mathbf{y}_f) + \nabla\phi(\mathbf{y}_f)(\mathbf{y} - \mathbf{y}_f)^T + \frac{\alpha}{2}\|\mathbf{y} - \mathbf{y}_f\|^2. \tag{2.22}$$

The function $\varphi : \mathbb{R}^n \to \mathbb{R}$,

$$\varphi(\mathbf{y}) = \phi(\mathbf{y}_f) + \nabla\phi(\mathbf{y}_f)(\mathbf{y} - \mathbf{y}_f)^T + \frac{\alpha}{2}\|\mathbf{y} - \mathbf{y}_f\|^2$$

attains its global minimum for $\mathbf{y}$ satisfying the condition

$$0 = \nabla\varphi(\mathbf{y}) = \nabla\phi(\mathbf{y}_f) + \alpha(\mathbf{y} - \mathbf{y}_f).$$

Using the solution $\mathbf{y}^\sharp = -\frac{1}{\alpha}\nabla\phi(\mathbf{y}_f) + \mathbf{y}_f$ in Inequality (2.22) leads to the estimate

$$\begin{aligned}
\phi(\mathbf{y}) &\geq \phi(\mathbf{y}_f) + \nabla\phi(\mathbf{y}_f)^T(\mathbf{y}^\sharp - \mathbf{y}_f) + \frac{\alpha}{2}\|\mathbf{y}^\sharp - \mathbf{y}_f\|^2 \\
&= \phi(\mathbf{y}_f) + \nabla\phi(\mathbf{y}_f)^T\left(-\frac{1}{\alpha}\nabla\phi(\mathbf{y}_f)\right) + \frac{\alpha}{2}\left\|-\frac{1}{\alpha}\nabla\phi(\mathbf{y}_f)\right\|^2 \\
&= \phi(\mathbf{y}_f) - \frac{1}{2\alpha}\left\|\nabla\phi(\mathbf{y}_f)\right\|^2
\end{aligned}$$

which implies that ϕ is lower bounded, i.e., $\phi(\mathbf{y}) > m > -\infty$ for all $\mathbf{y} \in D$ and $m \in \mathbb{R}$ chosen appropriately.

Let $\mathbf{y} \in D$ be arbitrary and $\mathbf{y}_f \in D$ be arbitrary and fixed. Then from the definition of strong convexity (Equation (2.19) for $\lambda = 1/2$) we obtain

$$\frac{1}{4}\alpha\left\|\mathbf{y} - \mathbf{y}_f\right\|^2 + 2m - \phi(\mathbf{y}_f) \leq \frac{1}{4}\alpha\left\|\mathbf{y} - \mathbf{y}_f\right\|^2 + 2\phi\left(\frac{1}{2}\mathbf{y} + \frac{1}{2}\mathbf{y}_f\right) - \phi(\mathbf{y}_f) \leq \phi(\mathbf{y})$$

which implies that $\phi(\mathbf{y}) \to \infty$ for $\|\mathbf{y}\| \to \infty$ since $2m - \phi(\mathbf{y}_f)$ is fixed. For all $r \in \mathbb{R}_{>0}$ such that $D \cap B_r(0) \neq \emptyset$, the set $D \cap B_r(0)$ is compact and hence, $\phi : D \cap B_r(0) \to \mathbb{R}$ attains its minimum. Due to the condition $\phi(\mathbf{y}) \to \infty$, $\phi : D \to \mathbb{R}$ attains its minimum.

□

Theorem 2.4.9 (iii) guarantees that the minimum of a strongly convex function defined on a closed and convex set is attained. Similarly, for a strict convex function defined on a closed and compact set, we know that at most one global minimum exists. If the set is additionally compact, then the continuity of the function and the strict convexity ensure that the minimum is attained. Hence, in the following chapters, we either consider strictly convex functions defined on convex and compact sets, or strongly convex functions defined on convex and closed sets to make sure that the unique global minimum is attained.

2.4.3 Convex optimization

As pointed out in the last section, convex functions defined on convex sets have desirable properties with respect to optimization. The optimization problem

$$\begin{aligned} \min_{\mathbf{y}\in\mathbb{R}^n} \quad & \phi(\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{y} \in D \end{aligned}$$

is said to be convex if the function $\phi : \mathbb{R}^n \to \mathbb{R}$ and the set $D \subset \mathbb{R}^n$ are convex. By introducing the functions $\chi : \mathbb{R}^n \to \mathbb{R}^{m_1}$ and $c : \mathbb{R}^n \to \mathbb{R}^{m_2}$, $m_1, m_2 \in \mathbb{N}$, optimization problems can equivalently be represented using equality and inequality constraints

$$\begin{aligned} \min_{\mathbf{y}\in\mathbb{R}^n} \quad & \phi(\mathbf{y}) \\ \text{s.t.} \quad & \chi(\mathbf{y}) = 0 \\ & c(\mathbf{y}) \leq 0 \end{aligned}$$

and the functions h and c are defined such that

$$D = \{\mathbf{y} \in \mathbb{R}^n | \chi(\mathbf{y}) = 0 \ \wedge \ c(\mathbf{y}) \leq 0\}$$

is satisfied. A special form of inequality constraints are polyhedral constraints. In this case, c is defined as $c(\mathbf{y}) = A\mathbf{y} - b$ for a matrix $A \in \mathbb{R}^{n\times m_1}$ and a vector $b \in \mathbb{R}^{m_1}$. Equality constraints can always be written as inequality constraints by using the definition

$$c(\mathbf{y}) = \begin{pmatrix} \chi(\mathbf{y}) \\ -\chi(\mathbf{y}) \end{pmatrix}. \tag{2.23}$$

From the representation (2.23) we obtain that the set D can only be convex if the c is affine.

Example 2.4.10 (Affine function)**.** *A function $c : \mathbb{R}^n \to \mathbb{R}$,*

$$h(\mathbf{y}) := A\mathbf{y} + b \tag{2.24}$$

for $A \in \mathbb{R}^{n\times m_1}$ and $b \in \mathbb{R}^{m_1}$ is called affine. Since

$$c(\lambda\mathbf{y}_1 + (1-\lambda)\mathbf{y}_2) = \lambda h(\mathbf{y}_1) + (1-\lambda)h(\mathbf{y}_2)$$

holds for all $\mathbf{y}_1, \mathbf{y}_2$ and for all $\lambda \in (0,1)$, affine functions are convex but not strictly convex. Moreover, $-h$ is affine and hence convex.

In Chapters 5 to 7 we concentrate on distributed optimization algorithms for convex optimization problems. Leveraging the properties introduced in the last section, efficient algorithms have been developed, such as interior point or sequential quadratic programming algorithms (see for example [17] or [78]) to compute the solution of convex optimization problems. In this thesis, the underlying optimization algorithms to solve a single optimization problem will be considered as a black box algorithm. The focus of this thesis is to decompose a single optimization problem into several smaller problems and to develop algorithms with guaranteed performance if the smaller problems are solved instead of the original one.

Chapter 3

A network of residential energy systems

With economically viable residential storage on the horizon, researchers have in recent years moved from the analysis of relatively rudimentary and largely uncoordinated battery energy storage systems [74] to systems of increasing scale and sophistication [52], [54], [98], [48], [82], [79]. In this chapter, a dynamical model of a small-scale residential energy system is introduced. The dynamical system is the basis for the remainder of this thesis. For the numerical evaluation of our approach, we use the dataset [1] provided by the Australian electricity company Ausgrid.[1] The dataset is described in Section 3.2. The chapter is closed by comparing different control structures for electricity networks.

3.1 The energy transition

Electricity distribution networks, originally designed for one-way power flow from large generators to residential customers via transmission and distribution networks, have to cope with many challenges that come along with the energy transition.
An electricity grid classically consists of several central, mainly fossil fuel-based power plants, such as coal-fired or natural gas plants, which are coupled through the transmission grid to the energy providers, which again are connected to residential and industrial consumers via a distribution network and are responsible to cover their energy demand (see Figure 3.1). The average power consumption $\hat{w}$ in kilowatt ([kW]) of residential customers is more or less periodic over a single day. In Figure 3.2 (top), the power consumption w_i of $i = 1, \dots, 100$ residential customers from the Ausgrid dataset (see Section 3.2) and the average consumption $\hat{w} = \sum_{i=1}^{\mathcal{I}} w_i$ during a single day and during a week is visualized. One can observe characteristic maximal peaks in the power demand in the morning and in the early evening as well as valleys in the demand around noon and during the night. The periodic behavior and the centralized power generation make it relatively easy for the energy providers to balance the power generation and the power demand and hence,

[1] Visit http://www.ausgrid.com.au for more information on Ausgrid.

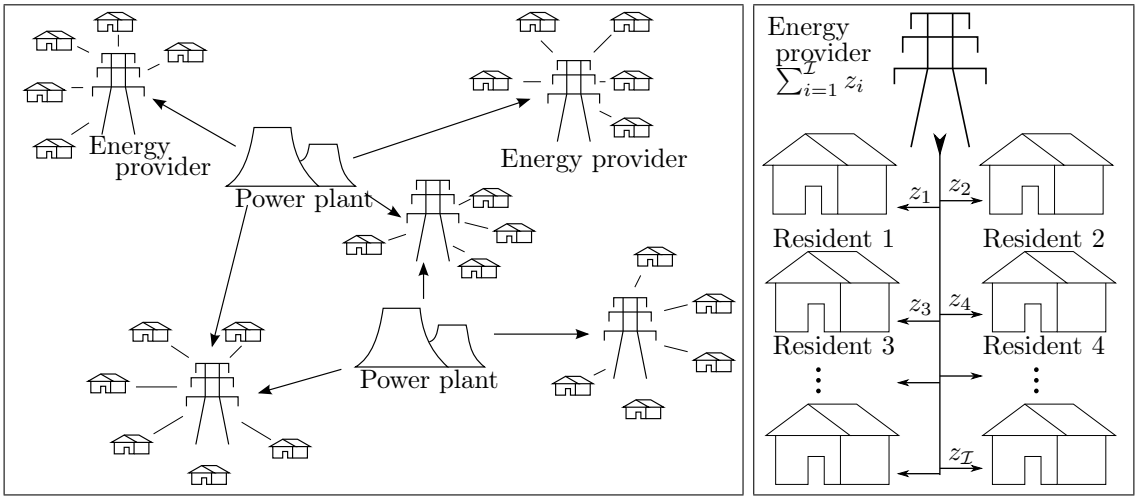

Figure 3.1: *Electricity distribution network for a one-way power flow. Residents obtain energy from their energy provider which cover their demand from centralized power plants. On the left, the overall network, and on the right, a single energy provider responsible for the connected residents.*

maintain voltage stability and prevent power outages and blackouts in the electricity grid. The energy transition has led to a rapid change in the classical distribution grid over the

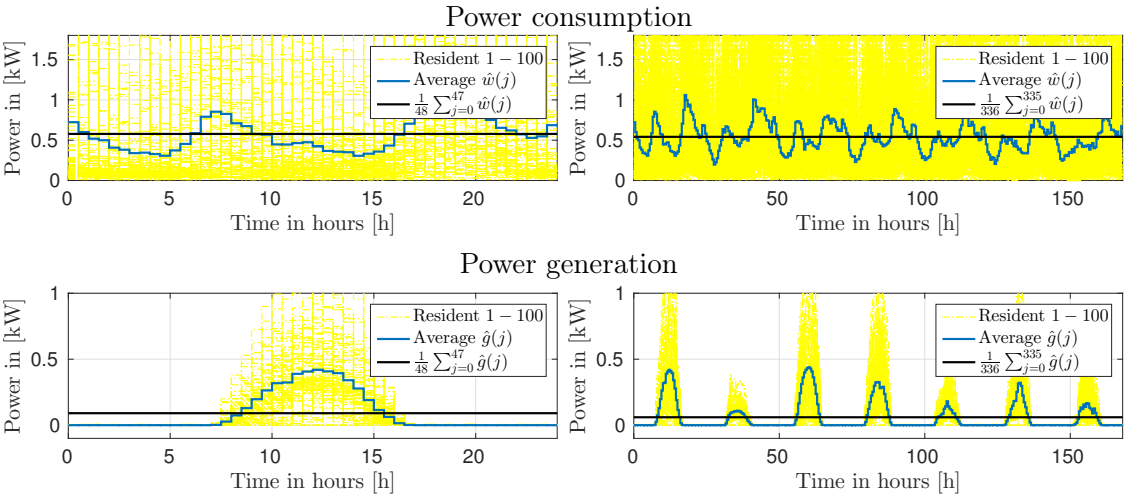

Figure 3.2: *Power and average power consumption and generation of* 100 *residents during a single day (left) and during a week (right). The power is generated using solar photovoltaic panels. One can observe a periodic behavior in the average consumption as well as in the average generation profile.*

past years. The growing impact of decentralized renewable energy technologies like photovoltaic panels and wind turbines shifts power generation from centralized power plants to decentralized sustainable power generation, necessitating a flexible grid. For example in 2015, the power generation from renewable energies covered 32.6% of the gross electricity

consumption in Germany compared to 17% in 2010 [4].
In Figure 3.2 (bottom), the power generation of 100 residents with rooftop solar photovoltaic installations is visualized. Not surprisingly, the power generation follows a periodic behavior with a peak at the middle of the day. Additionally, the deviation in the generated power during a day is rather significant, which makes the long term prediction of power generation from photovoltaic panels (and also wind turbines) difficult.

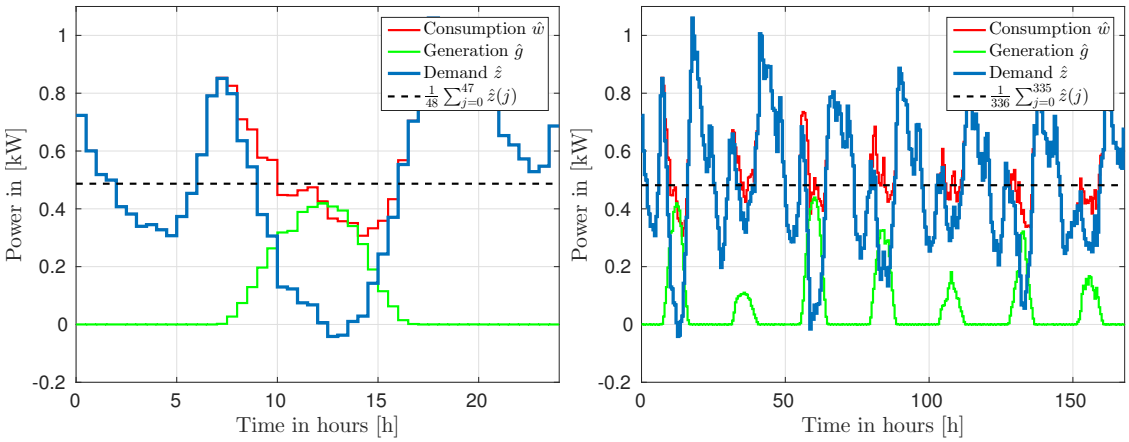

Figure 3.3: *Average power demand of* 100 *residents during a single day (left) and during a week (right). One can observe a periodic behavior and characteristic peaks in the morning and in the afternoon.*

To maintain the network stability under these changing conditions, the energy provider has to cope with several new conditions in the network. The shift from centralized to decentralized generation makes the coordination of a rapidly changing network more challenging. Additionally, in case of an increased power demand, the power generation of photovoltaic panels and wind turbines cannot be controlled to increase the power generation, since the generation depends solely on the intensity of the sun and the wind, respectively. Moreover, the reaction time of central coal-fired and nuclear power plants in general is too slow to balance these unpredictable shortages while staying profitable. If one compares the peaks of power consumption with the peaks of power generation in Figure 3.2 one also observes that the peak times do not coincide. Hence, the energy from solar photovoltaic panels is mainly generated at times where the demand is low and, consequently the variation between minimal and maximal demand is increased (see Figure 3.3). In our dataset, there are even days, where the power demand is below zero, i.e., power is pushed back into the grid.
Several publications tackle these problems from different point of views, see for example the recent survey paper [31] and [100] and the references therein. Here, we concentrate on the incorporation of additional distributed storage devices in the electricity network and we concentrate on a single energy provider and its customers as visualized in Figure 3.4 on the left. We assume that every customer connected to the energy provider has energy consuming devices, photovoltaic panels to generate power and a storage device to store

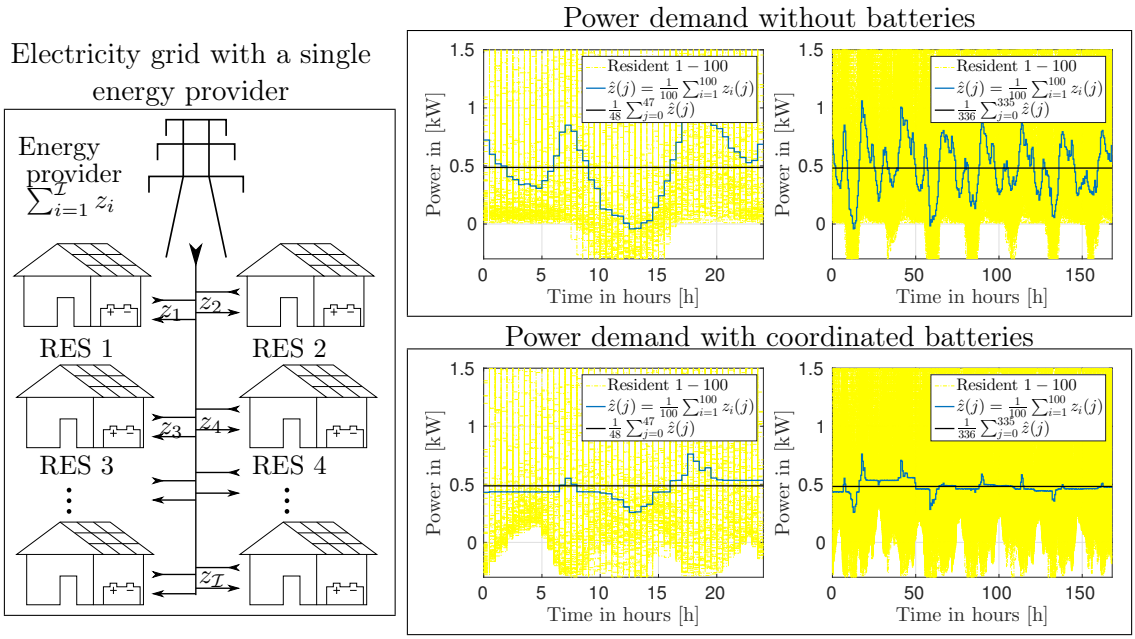

Figure 3.4: *Electricity grid with a single energy provider connected to a number of RESs. The RESs not only consume power, but are also able to generate and store power. If they are used in the correct way, the batteries help to shave the peaks in the average power demand.*

energy. A customer equipped with these devices will be denoted as a residential energy system (RES) throughout this work. In this case, an RES is a consumer as well as a producer, and energy can be exchanged between the individual RESs. The local grid is not disconnected from the main grid and the task of the energy provider remains maintenance of network stability and to provide enough energy for the users. If the batteries are used in a coordinated way, they can be used to flatten the aggregated power demand and to help the energy provider accomplish his task. This means coordination of the battery usage of the RESs is necessary for peak-shaving. In the case where all RESs charge their batteries to increase the overall energy demand, the energy surplus turns into an energy shortage. A similar effect can happen if all batteries are discharged to prevent a shortage. On the contrary, if the batteries are used in a certain coordinated way, the peaks in the energy demand profile can be reduced significantly. Hierarchical distributed algorithms for the coordination of distributed RESs will be the main contribution in the following chapters. Before these algorithms are introduced, the dataset which will be used to visualize our findings, is introduced in the following.

3.2 The Ausgrid dataset

To simulate the dynamical behavior of the electricity grid visualized in Figure 3.4, we use a dataset provided by the Australian electricity company Ausgrid. Among others, the publicly available dataset [1] contains the power consumption and power generation

of 300 customers in New South Wales, Australia, recorded from 1 July 2010 to 30 June 2011. Every customer is equipped with a smart meter measuring the energy consumption and generation on a 30 minutes time scale. Hence, the data models the electricity grid in Figure 3.4 without batteries. The data shown in the Figures 3.2, 3.3 and 3.4 are also based on the dataset. A comprehensive study of the dataset which contains more information than just the consumption and the generation profiles can be found in [84], giving a closer insight to the data and also explaining how the dataset can be used for further research. In our analysis, we only consider the sequence[2] of power consumption[3]

$$\mathbf{w}(\cdot) \in \mathbb{R}^{300\times 17520}$$

and the sequence of power generation

$$\mathbf{g}(\cdot) \in \mathbb{R}^{300\times 17520}$$

of the 300 customers. The 300 customers are referenced using the index $i \in \mathbb{N}_{300}$. To specify a certain time step, the index $k \in \mathbb{N}$ is used. We make the assumption that the power consumption and power generation, measured in kilowatt ([kW]), are constant between two consecutive time steps k and $k+1$.
In Figure 3.5, the aggregated power consumption, generation and demand from 1 July 2010 to 30 June 2011 is shown. For the consumption $\sum_{i=1}^{300} w_i(k)$, we observe a maximal peak of 419.669[kW] at 6:30pm of May 8 and a minimal peak of 0[kW] at 2:30am of March 4. Even though extreme peaks are rare, a significant difference between the summer and winter months can be observed. The average consumption during the whole year is given by 119.512[kW]. The aggregated generated power $\sum_{i=1}^{300} g_i(k)$ varies significantly between 0[kW] and 208[kW]. The average power generation is given by 36.277[kW]. The energy provider has to compensate the fluctuations in the aggregated power demand $\sum_{i=1}^{300} (w_i(k) - g_i(k))$. The maximal peaks of consumed power are only reduced slightly but the aggregated energy demand is less than zero at 2818 out of 17520 time steps which means at around 16% of the time, the energy provider has to handle a reverse power flow. In the next section, we equip the residences with a storage device turning the residences into residential energy systems (RESs). Hence, if the proper incentives are provided by the energy provider to the RESs the storage devices can be used to flatten the deviations.

Remark 3.2.1. *In [84], the dataset is analyzed to generate a clean dataset by removing data corresponding to residents with anomalous measurements. This means, the authors remove customers without consumption during certain time intervals and customers with power generation below a certain threshold. Removing the anomalous measurements is not necessary in our application since isolated storage devices without a consumer or residents without photovoltaic panels can also still be part of the electricity grid. The third group of customers which is removed in the reference are customers generating energy during times*

[2] To really obtain sequences $\mathbf{w}(\cdot) = (\mathbf{w}(k))_{k\in\mathbb{N}}$ and $\mathbf{g}(\cdot) = (\mathbf{g}(k))_{k\in\mathbb{N}}$ the finite dataset is periodically extended.

[3] The sequence of power consumption is the sum of the static load and the elastic load of given in the dataset.

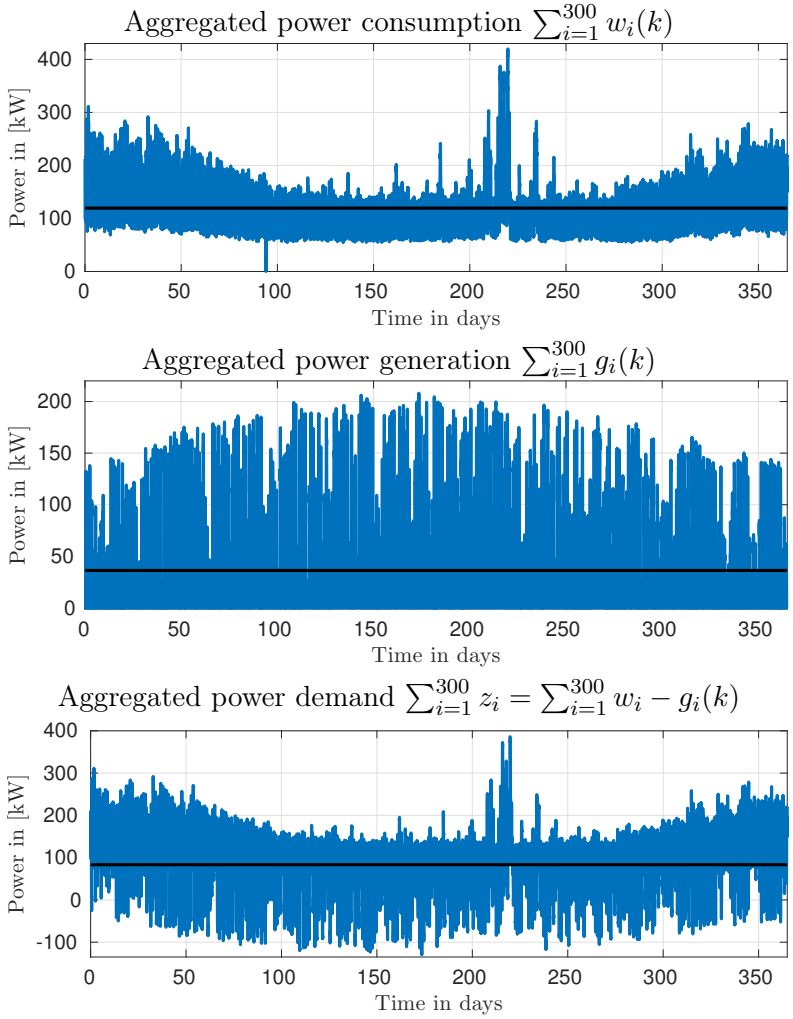

Figure 3.5: *Aggregated power consumption, generation and demand of* 300 *residences. The aggregated profiles are visualized with a sampling rate of* 30 *minutes. The averages over the whole year are indicated with a black line.*

without sun which should not be possible using photovoltaic panels. In our application, the photovoltaic panels could be replaced for example by wind turbines.

3.3 Network stability: the energy provider perspective

With the proposed dataset, we can now summarize the problem which will be considered in the next chapters. As already mentioned, the focus is on the interaction between a single energy provider and its customers which either act in a cooperative way to achieve a common goal, or act selfishly and non-cooperatively, optimizing only their own well-being. Regardless of the behavior of the customers, we assume that the energy provider bears the responsibility of a reliable energy supply. Hence, we investigate the stability of the

electricity grid from the energy provider point of view and not from the view of a single customer. This implies that we are not interested in the power demand of an individual system, but in the average power demand

$$\hat{\mathbf{z}}(\cdot) = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i(\cdot)$$

of the overall network. Stability of the network in this context means that the vertical deviations in the average power demand have to be as small as possible. The energy provider is interested in minimizing the maximal and minimal peaks in the average power demand to keep the variations on a minimal level. In this context, we are looking for a control in the scale of minutes or hours fitting to the available dataset. This should not be confused with frequency control in the scale of seconds or milliseconds, considered, for example, in [52], [42] or [71].
To achieve the goal of a flattened energy demand, we introduce an additional degree of freedom in the electricity network in the form of energy storage devices.

3.4 Modeling of a residential energy system

The power demand $z_i(k)$ in [kW] of a resident $i \in \mathbb{N}_{\mathcal{I}} = \{1, \dots, \mathcal{I}\}$ at time $k \in \mathbb{N}$ is defined through its power consumption and its power generation, i.e.,

$$z_i(k) = w_i(k) - g_i(k).$$

We treat the resident as a residential energy system (RES) by introducing an input variable u_i which can be used to manipulate the power demand $z_i(k)$ via

$$z_i(k) = w_i(k) - g_i(k) + u_i(k).$$

The input u_i can be interpreted in different ways. u_i can represent local generators which influence the power demand by increasing or decreasing the power generation depending on the overall demand (e.g. [60]). The input u_i can also represent a controllable load which can be shifted between different time steps k and is discussed in several papers on demand response or demand-side management (e.g. [69]). Moreover, u_i can belong to a storage device which is used to store a surplus of energy when necessary and provides energy in times of shortages. In the following subsection u_i refers to storage devices.

3.4.1 Residential energy systems using storage devices

In recent years we have seen an increasing development and variety of residential energy storages such as batteries (see for example *pv magazine* [2] for a list of batteries available on the market). Although we concentrate on rechargeable batteries for the model dynamics, we keep the dynamics flexible and general to be able to capture other storage devices as well, i.e., batteries can be replaced, for example, by gravitational potential energy storages or fuel cells depending on the grid under consideration. Consistent with the dataset, we

consider discrete dynamics in the time scale of minutes and hours instead of continuous dynamics covering every detail. Nevertheless, the general formulation makes our approach very flexible with respect to the underlying storage device. Similar models are used in the work [59], [102], and in the works [8, 9, 10].

The simplified residential energy system

A basic simplified model of RES $i \in \mathbb{N}_{\mathcal{I}}$ is defined through the dynamics

$$x_i(k+1) = x_i(k) + Tu_i(k), \tag{3.1a}$$

$$z_i(k) = w_i(k) - g_i(k) + u_i(k). \tag{3.1b}$$

The model is introduced in [109] and [108] and adds a second Equation (3.1a) to the power demand (3.1b), representing the storage device. Here, x_i denotes the state of charge (SOC) of the battery in kilowatt hours ([kWh]) and the input u_i is the battery charge/discharge rate in [kW]. With the input u_i, the already introduced power demand z_i depending on the fixed load and generation sequences $(w_i(k))_{k\in\mathbb{N}}$ and $(g_i(k))_{k\in\mathbb{N}}$ is flexible and can be manipulated. Here, T represents the length of the sampling interval in hours ([h]). The individual systems are coupled through their demand to the other systems and connected with a single energy provider at the point of common coupling visualized in Figure 3.4. Additionally, the constraints on the battery capacity

$$0 \leq x_i(k) \leq C_i \tag{3.2}$$

and on the maximal charge/discharge rate

$$\underline{u}_i \leq u_i(k) \leq \overline{u}_i \tag{3.3}$$

for all $k \in \mathbb{N}$ were introduced. The model dynamics gives a very simplified view of an electricity grid. However, even though it does not capture any characteristics of a specific storage device, the dynamics is descriptive and can capture the interplay between the individual RESs and the energy provider and hence, turn out to be useful to visualize and interpret distributed control concepts.

The generalized residential energy system

The conversion of energy either from the battery to the grid or from the grid to the battery is always connected to loss depending on the energy conversion efficiency. Additionally, the electric charge has to be converted from alternating current to direct current if the battery is charged from the grid but not if the battery is charged from the solar panels. Hence, a different efficiency can be expected in these cases. To capture these effects, we split the input variable into three variables

$$u_i = \begin{pmatrix} u_{i_1} \\ u_{i_2} \\ u_{i_3} \end{pmatrix} = \begin{pmatrix} u_i^{+_w} \\ u_i^{+_g} \\ u_i^{-} \end{pmatrix} \in \mathbb{R}^3$$

and introduce the energy efficiencies

$$B_i = T \cdot \begin{pmatrix} c_{E_{i_1}} & c_{E_{i_2}} & 1 \end{pmatrix} = T \cdot \begin{pmatrix} c_{E_{+_w}} & c_{E_{+_g}} & 1 \end{pmatrix} \in \mathbb{R}^{1\times 3}_{\geq 0}.$$

The variable $u_i^{+w} \in \mathbb{R}_{\geq 0}$ represents charging from the grid, $u_i^{+g} \in \mathbb{R}_{\geq 0}$ represents charging from the solar panels and $u_i^- \in \mathbb{R}_{\geq 0}$ represents discharging of the battery. (We use the subscripts $\cdot^w$ and $\cdot^g$ to indicate that charging the storage device from the grid increases the consumption w, and charging the storage device from the energy generated in the solar panels decreases the energy which is fed into the grid.) The constants $c_{E_{+_w}}, c_{E_{+_g}} \in (0, 1]$ define the energy efficiency of the storage device. The loss of energy when the battery is discharged is captured in the equation of the communication variables[4]

$$z_i(k) = D_i u_i(k) + w_i(k) - g_i(k)$$

with

$$D_i = \begin{pmatrix} 1 & 1 & c_{E_{i_3}} \end{pmatrix} = \begin{pmatrix} 1 & 1 & c_{E_{i\_}} \end{pmatrix} \in \mathbb{R}^{1\times 3}$$

and $c_{E_{i\_}} \in (0, 1]$.
Even if the battery is not used, i.e., $u_i \equiv 0$, the storage device loses energy. We assume that the loss of energy depends on the state of charge of the battery and the temperature of the battery [97, Ch. 7]. If the battery is not used, we assume that the temperature is constant, and hence, does not influence the state of charge of the battery. The uncontrolled dynamics, therefore, only depends on the state of charge of the battery, i.e.,

$$x_i(k+1) = A_i x_i(k)$$

for $A_i = c_{E_{i_x}} \in (0, 1]$. In total, the time-varying discrete-time system in form (2.1) is given by

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k), \tag{3.4a}$$
$$z_i(k) = D_i u_i(k) + s_i(k) \tag{3.4b}$$

where $s_i(k) = w_i(k) - g_i(k)$.

Remark 3.4.1. *In [27], the authors provide characteristics for several storage devices. For example the self-discharge of a Li-ion battery is specified with* 0.1%–0.3% *during a single day. Using a discretization of* $T = 0.5$*[h] leads to a constant* $c_{E_{i_x}} > 0.999$ *and thus, the loss of energy for Li-ion batteries is negligible. Moreover, the cycle efficiency of Li-ion batteries is higher than* 90%*, i.e., the losses in the (dis-)charging process are less than* 10% *[27].*

Rewriting the constraints (3.3) in the extended input variables u_i^{+w}, u_i^{+g} and u_i^- leads to

$$\begin{array}{rcccl} \underline{u}_i & \leq & u_i^-(k) & \leq & 0, \\ 0 & \leq & u_i^{+w}(k) + u_i^{+g}(k) & \leq & \overline{u}_i, \\ 0 & \leq & u_i^{+w}(k), & & \\ 0 & \leq & u_i^{+g}(k), & & \end{array}$$

[4]Note that the loss of energy can equivalently be captured in the dynamics of the storage device. In this case one has to define $c_{E_{i\_}} \in [1, \infty)$.

for all $k \in \mathbb{N}$. Since $u_i^{+_g}$ depends on the amount of energy generated by the solar panels, additionally the constraint

$$u_i^{+_g}(k) \leq g(k)$$

needs to be introduced.
A constraint which is not modeled so far is the fact that the battery cannot be charged and discharged at the same time. This condition is captured by the nonlinear equality constraints

$$\left(u_i^{+_w}(k) + u_i^{+_g}(k)\right) \cdot u_i^{-}(k) = 0. \tag{3.5}$$

for all $k \in \mathbb{N}$.
Since we are using a coarse sampling time in the magnitude of 30 minutes, it is reasonable to assume that the battery switches from charging to discharging, and vice versa, within the given time interval. Thus, we introduce a different condition than (3.5). When the battery switches between charging and discharging, only a fraction of the time interval is used to charge/discharge the battery. Let $\lambda, (1-\lambda) \in [0,1]$ represent the time of the interval used for charging/discharging. Then, the upper and lower bounds for charging and discharging have to be adapted, i.e.,

$$\begin{aligned} u_i^{-}(k) &\geq (1-\lambda)\underline{u}_i \\ u_i^{+_w}(k) + u_i^{+_g}(k) &\leq \lambda \overline{u}_i \end{aligned}$$

which is equivalent to

$$\begin{aligned} \frac{u_i^{-}(k)}{\underline{u}_i} &\leq (1-\lambda) \\ \frac{u_i^{+_w}(k) + u_i^{+_g}(k)}{\overline{u}_i} &\leq \lambda \end{aligned}$$

if $\underline{u}_i < 0$ and $\overline{u}_i > 0$ holds. To avoid the variable λ, we combine the constraints and obtain the upper and lower bounds

$$0 \leq \frac{u_i^{-}(k)}{\underline{u}_i} + \frac{u_i^{+_w}(k) + u_i^{+_g}(k)}{\overline{u}_i} \leq 1.$$

for the charging and discharging rate.
In summary, the generalized dynamics of RES i are given by

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k), \tag{3.6a}$$

$$z_i(k) = D_i u_i(k) + s_i(k) \tag{3.6b}$$

with

$$A_i = c_{E_{i_x}}, \qquad B_i = T \cdot \left(c_{E_{+_w}}\ c_{E_{+_g}}\ 1\right), \qquad \text{and} \qquad D_i = \left(1\ 1\ c_{E_{i_-}}\right),$$

subject to the constraints

$$\begin{array}{rcccl} 0 & \leq & x_i(k) & \leq & C_i, \\ \underline{u}_i & \leq & u_i^-(k) & \leq & 0, \\ 0 & \leq & u_i^{+w}(k) + u_i^{+g}(k) & \leq & \overline{u}_i, \\ 0 & \leq & u_i^{+w}(k) & \leq & \overline{u}_i \\ 0 & \leq & u_i^{+g}(k) & \leq & g_i(k), \\ 0 & \leq & \frac{u_i^-(k)}{\underline{u}_i} + \frac{u_i^{+w}(k)+u_i^{+g}(k)}{\overline{u}_i} & \leq & 1. \end{array} \tag{3.7}$$

Remark 3.4.2. *Note that the system dynamics defines a time-varying dynamical system subject to linear inequality constraints. The inequalities* (3.7) *contain redundant information and can be written in a more compact way if necessary.*

The nonlinear residential energy system

In the algorithms introduced in the following chapters, we concentrate on the linear dynamics (3.6) together with the linear constraints (3.7). Nevertheless, in Chapter 5 we propose a distributed optimization algorithm which is able to handle nonlinear dynamics and nonlinear constraints (see Algorithm 7). To this end, we give a possible extension for the dynamics of the RESs.
In general any dynamical system of the form (2.1) can be used to model RESs connected only through their energy demand $\sum_{i=1}^{\mathcal{I}} z_i$. In principle, partial differential equations or delay differential equations can also be used to model the dynamics of the batteries. Here, we restrict ourselves to an extension of the dynamics (3.6a) by additional nonlinear terms. If batteries are used extensively, they can heat up quite quickly. Hence, we assume that the temperature of the battery depends on the charging/discharging rates. A strong usage leads to high temperatures and corresponding losses which we assume to be quadratic and which modify (3.6a) to be

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) - T c_{H_i} \left(\left(u_i^{+g}(k) + u_i^{+w}(k) \right)^2 + u_i^-(k)^2 \right)$$

depending on the constant $c_{H_i} \in \mathbb{R}_{\geq 0}$. Additionally, batteries are usually only used within a certain interval of their capacity. If the SOC of the battery is above or below that value, batteries are in general less efficient. To incorporate this effect, we add another term to the dynamics

$$\begin{aligned} x_i(k+1) =& A_i x_i(k) + B_i u_i(k) \\ & - T c_{H_i} \left(\left(u_i^{+g}(k) + u_i^{+w}(k) \right)^2 + u_i^-(k)^2 \right) \\ & + T \left(\frac{c_{C_{i_-}}}{x_i(k)} u_i^-(k) - \frac{C_i c_{C_{i_+}}}{C_i - x_i(k)} \left(u_i^{+g}(k) + u_i^{+w}(k) \right) \right) \end{aligned} \tag{3.8}$$

with constants $c_{C_{i_-}}$, $c_{C_{i_+}} \in \mathbb{R}_{\geq 0}$. If the battery reaches its capacity C_i or is almost empty, it becomes ineffective and energy in the charging/discharging process is lost. In Figure 3.6

the effects of the constants $c_{C_{i_-}}$ and $c_{C_{i_+}}$ are visualized for the charging and discharging process of a battery. For $c_{C_{i_+}} > 0$ the maximal capacity C_i is not reached anymore. If the battery is discharged, the losses increase with decreasing SOC of the battery.

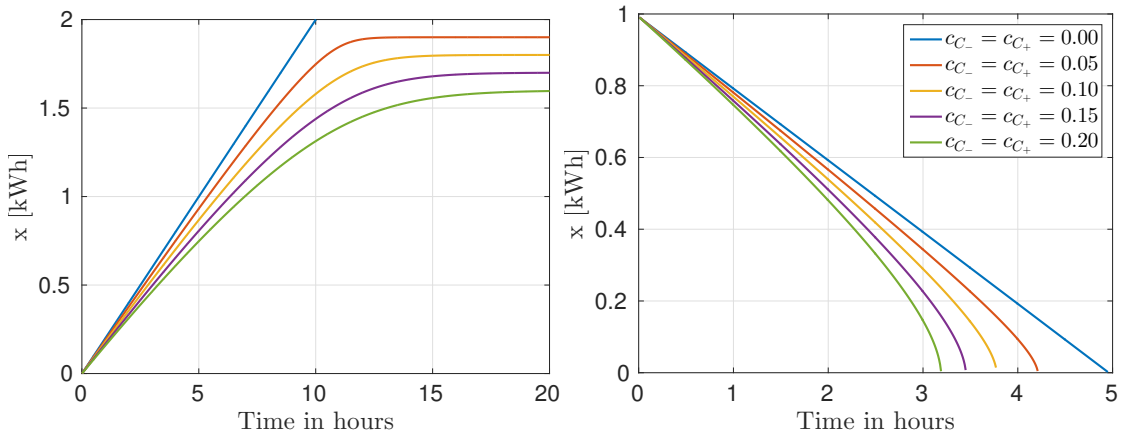

Figure 3.6: *Charging (left) and discharging (right) curves of the battery model* (3.8) *with capacity* $C = 2$*[kWh]. The battery is charged (left) with* $u^{+w} + u^{+g} \equiv 0.2$ *and discharged (right) with* $u^- \equiv -0.2$ *depending on the constants* c_{C_-} *and* c_{C_+}*. The other constants are set to* 1 *except for* $c_H = 0$*.*

For the constraints, we have to make sure that

$$u_i^{+w}(k) + u_i^{+g}(k) \geq 0$$

charges the battery, i.e., the energy loss has to be smaller than 100%. This can be modeled by the constraints

$$\left(c_{E_{+w}}\ c_{E_{+g}}\ 0\right) u_i(k) - c_{H_i}\left(u_i^{+w}(k) + u_i^{+g}(k)\right)^2 - \frac{C_i c_{C_{i_+}}}{C_i - x_i(k)}(u_i^{+w}(k) + u_i^{+g}(k)) \geq 0.$$

Likewise, when the battery is discharged, the losses have to be taken into account for the lower bound of the discharging rate

$$u_i^-(k) - c_H u_i^-(k)^2 + \frac{c_{C_-}}{x_i(k)} u_i^-(k) \geq \underline{u}_i.$$

In [97, Section 7.1.13], it is argued that the maximal charging/discharging rate depends on the SOC of the battery. The maximal charging rate decreases when the battery reaches its maximum capacity and the maximal discharging rate decreases with the SOC of the battery. These effects can be incorporated using the state dependent constraints

$$\begin{array}{rcccl} \left(\frac{x_i(k)}{C_i}\right)^\alpha \underline{u}_i & \leq & u_i^m(k) & \leq & 0, \\ 0 & \leq & u_i^{+w} + u_i^{+g}(k) & \leq & \left(1 - \frac{x_i(k)}{C_i}\right)^\alpha \overline{u}_i \end{array} \tag{3.9}$$

with $\alpha \in [1, \infty)$. The factor $(x_i(k)/C_i) \leq 1$ and $(1 - (x_i(k)/C_i)) \leq 1$, respectively, ensure that the maximal charging and discharging rate can only be used when the battery is charged completely or when the battery is completely empty. Depending on α, the effects increase when the battery is almost empty or close to its maximal capacity. In these cases, for large values of α the maximal rates drop significantly for $x_i(k)/C_i$ and $1 - (x_i(k)/C_i)$ approaching 0. The parameter α for the charging and discharging rate can of course be chosen differently.

3.4.2 Additional approaches in smart grid applications

As already discussed, the input u used for peak-shaving can be interpreted in different ways. In the remainder of this section we further discuss approaches from the literature and the connection to our model.
In the papers [8, 9, 10] a similar model to our network of RESs is proposed and a similar optimization problem is discussed. The authors propose a linear model to minimize the electricity costs of customers which are coupled through the aggregated energy demand. Some customers are able to store energy, and some customers have dispatchable energy generation units. While the focus of the paper [8] is on the demand-side, i.e., the electricity cost reduction of single customers in a day-ahead market, our focus is on the supply side aiming to obtain a smooth load profile, and hence the role of a central entity (CE) or a grid operator is much stronger in our setting. The day-ahead pricing scheme considered in [8] is also applicable to our setting. In this thesis, electricity costs are only discussed in Chapter 6 in the form of a real-time pricing scheme. The authors of [8] motivate their problem from a game theoretic point of view. To find a Nash equilibrium of their corresponding minimization problem, a proximal point algorithm is used (see [92] or [81], for example) with guaranteed performance properties. In [9] the authors extend their work by considering a non-cooperative setting and by using an asynchronous communication structure between the customers and the CE. In [10] additional coupling constraints between the consumers, similar to the setting in Chapter 7, are considered. Moreover, a reoptimization algorithm on a shrinking horizon is proposed. The simulations in [8, 9, 10] are not considered in a MPC context with a moving horizon. Only a fixed time interval of a single day is taken into consideration.

Controllable loads

Controllable or elastic load refers to a power demand which is not fixed to a certain time schedule. A recent study [12] suggests that up to 60% of the consumption of a household, in the form of appliances such as air conditioners and refrigerators, is elastic or schedulable. Therefore, an alternate, but complementary approach to the use of energy storage devices to reduce the grid variations involves energy consumption scheduling [101], [68, 69], [90]. Similar to the storage devices, the degree of freedom in the load w can be used to manipulate the demand z.
In [101] the authors propose an algorithm to coordinate demand of a consumer cooperative via an iterative hierarchical distributed optimization scheme. The total demand of

a single consumer during the time interval under consideration is known and constant, but the demand of individual consumers can be shifted from one time step to another as long as individual convex constraints are satisfied. The consumers optimize their power demand based on virtual prices sent by a CE. The virtual prices and the power demands are updated iteratively until convergence is obtained. The proposed distributed optimization algorithm converges to the solution of a corresponding centralized optimization problem and minimizes the total costs of the cooperative. Different from the distributed optimization algorithms discussed in Chapter 5 to 7 the CE has to communicate with every consumer individually. The proposed algorithm only works under the assumption that the total power demand over the prediction horizon is constant. Thus, storage devices including losses cannot be considered. In contrast to our work, the authors only consider a single optimization problem at a fixed time step. The setting is not embedded in a receding horizon scheme.

In [68, 69], the authors propose a model, where each appliance of a residence is considered separately. In this case, an energy consumption scheduler is proposed to optimally schedule the load of the different appliances to minimize electricity costs. Likewise, in [70] the authors propose a model by considering every appliance individually and solve a mixed-integer problem to compute an energy schedule.

Since our focus is on the dynamical behavior of the overall electricity grid and not on the behavior of a single residence and a single optimization problem, we use a formulation which is easier to handle in the dynamical behavior by aggregating the elastic load in a single appliance. Nevertheless, the concepts of elastic loads are still captured, and the proposed algorithms can be easily extended to more complex systems. We assume, that the load of a RES consists of the consumption or static load $\mathbf{w}_i(\cdot)$, already introduced at the beginning of this chapter, plus additional controllable load denoted by $\mathbf{w}_{c_i}(\cdot)$ in [kWh] for all $i \in \mathbb{N}_{\mathcal{I}}$. As in the static case, we assume $\mathbf{w}_{c_i}(\cdot)$ to be given sequences for all $i \in \mathbb{N}_{\mathcal{I}}$. As already motivated, controllable load is attached to a certain time interval. More precisely, we assume that $w_{c_i}(k)$ has to be scheduled during the time interval from $k-\overline{N}+1$ to k for a given $\overline{N} \in \mathbb{N}$, i.e., $w_{c_i}(k)$ has its deadline at time k but can already be scheduled in the $\overline{N}$ time steps before the deadline. Introducing the input $u_{c_i} \in \mathbb{R}$ in the power demand

$$z_i(k) = D_i u_i(k) + s_i(k) + u_{c_i}(k) \tag{3.10}$$

leads to time-dependent constraints

$$\sum_{j=0}^{k} w_{c_i}(j) - \sum_{j=0}^{k-1} u_{c_i}(j) \leq u_{c_i}(k) \leq \sum_{j=0}^{k+\overline{N}-1} w_{c_i}(j) - \sum_{j=0}^{k-1} u_{c_i}(j) \tag{3.11}$$

for all $k \in \mathbb{N}$ and for each RES $i \in \mathbb{N}_{\mathcal{I}}$. Observe that at time k, $u_{c_i}(j)$ is fixed for all $j < k$, rather than a control variable, since it is a control action that has been applied.

To model that only a certain amount of the controllable load can be scheduled at a fixed time step, we introduce the lower and upper bounds

$$\underline{w}_{c_i} \leq u_{c_i}(k) \leq \overline{w}_{c_i} \tag{3.12}$$

for all $k \in \mathbb{N}$ and given bounds $\underline{w}_{c_i}$, $\overline{w}_{c_i} \in \mathbb{R}$ for all $i \in \mathbb{N}_{\mathcal{I}}$. We assume that the $\underline{w}_{c_i}$, $\overline{w}_{c_i}$ are chosen such that conditions (3.11) and (3.12) can be simultaneously satisfied.
For $q \in \{0, \dots, N-1\}$ and $i \in \mathbb{N}_{\mathcal{I}}$ the constraints (3.11) simplify to the expressions

$$\lambda_i^q(k) := \sum_{j=0}^{k+q} w_{c_i}(j) - \sum_{j=0}^{k-1} u_{c_i}(j) \leq \sum_{j=k}^{k+q} u_{c_i}(j), \tag{3.13a}$$

$$\Lambda_i^q(k) := \sum_{j=0}^{k+\min\{q+\overline{N},N\}-1} w_{c_i}(j) - \sum_{j=0}^{k-1} u_{c_i}(j) \geq \sum_{j=k}^{k+q} u_{c_i}(j). \tag{3.13b}$$

The term $\min\{q+\overline{N}, N\}$ reflects that at a given time instant k only the controllable load of the future $N \in \mathbb{N}$ time steps is known and hence, only controllable load with a deadline in the next N time steps are covered by the dynamical bounds. Observe that the bounds can be easily updated by

$$\lambda_i^q(k+1) = \lambda_i^q(k) + w_{c_i}(k+q+1) - u_{c_i}(k) \qquad \text{and} \tag{3.14}$$

$$\Lambda_i^q(k+1) = \Lambda_i^q(k) + w_{c_i}(k+\min\{q+\overline{N}, N\}) - u_{c_i}(k). \tag{3.15}$$

Using λ_i^q and Λ_i^q avoids the possible confusion whether $u_{c_i}(k)$ is a control variable which still needs to be set, or if $u_{c_i}(k)$ is a fixed value.
Observe that the model of controllable load can be easily coupled with the battery model since the input u_{c_i} as well as the bounds are decoupled from the inputs in the battery model. Simulations of a network of RESs with storage devices and controllable loads are provided in Section 4.6.3.

Remark 3.4.3. *The presented approach makes a couple of assumptions which are necessary to keep the model simple and to be able to compute solutions in a reasonable time. The model does not capture, for example, that once a dishwasher starts it has to run for a certain amount of time and cannot be stopped and started throughout the given scheduling window. Additionally, the dishwasher has to run for a fixed amount of time and the running time cannot be stretched on the whole scheduling window only using a little amount of power the whole time. The simplification avoids that every appliance has to be considered individually. Extending the model in this direction makes a general analysis of the results impossible in the context of this thesis. In practice, appliances like air conditioning systems and refrigerators are very flexible and thus, can compensate for the inflexibility of other machines.*
Another assumption, which is implied by the constraints (3.13)*, is that controllable load with a closer deadline is handled first. Once again, neglecting this assumption, would lead to a model where every appliance has to be considered individually.*

Vehicle to grid installations and local generators

A large number of papers have investigated the impact of electric vehicles in future smart grids (see e.g. [57], [111], [38], [65], [93]). The batteries in the electric vehicles can either

be used as a controllable load or as a storage device which is only temporarily available for the RES. In the context of controllable loads, the fleet of electric vehicles is often used in the so-called valley-filling, i.e., the batteries of the cars are charged at times of low demand and high generation of renewable energy [64, 65]. In this setting, the cars have to be charged in a certain time window which can be realized by our model of controllable load.
Using the electric vehicles as a temporary storage device as in [38] can be implemented in our network of RESs by using time-varying constraints on the upper and lower capacity of the battery, i.e.,

$$\underline{C}_i(k) \quad \leq \quad x_i(k) \quad \leq \quad \overline{C}_i(k),$$

where now $x_i(k)$ represents the state of charge of the battery of the electric vehicle and $(\underline{C}_i(k))_{k\in\mathbb{N}}, (\overline{C}_i(k))_{k\in\mathbb{N}} \subset \mathbb{R}$ represent time-varying bounds of RES $i \in \mathbb{N}_{\mathcal{I}}$. At times $k \in \mathbb{N}$ where the vehicle is not connected to the grid, we set $\underline{C}_i(k) = \overline{C}_i(k) = 0$. If the car is idle, we set $\underline{C}_i(k) = 0$ and $\overline{C}_i(k) = C_i$ according to the model in Section 3.4.1. At times where the car is needed, one can set $\underline{C}_i(k) = \overline{C}_i(k) = C_i$ to ensure that the battery is fully charged at time k.
In the same way as electric vehicles can be used in valley-filling to increase the power demand, local generators such as diesel generators, can be implemented in the network to reduce high peak demands by providing additional power.
Generators can be included by introducing an additional input $u_i^{gen} \in [0, G_{\max}]$ in the dynamics of the power demand z_i for all $i \in \mathbb{N}_{\mathcal{I}}$, with a maximal power generation $G_{\max} \in \mathbb{R}_{>0}$. To capture the slow dynamics of a generator, the constraints

$$|u_i^{gen}(k+1) - u_i^{gen}(k)| \leq \overline{u}_i^{gen},$$

$\overline{u}_i^{gen} \in \mathbb{R}_{>0}$ for all $i \in \mathbb{N}_{\mathcal{I}}$ and for all $k \in \mathbb{N}$ are, for example, used [59] to bound the difference in the power generation in two consecutive time steps.
To summarize, the model introduced in Section 3.4.1 is quite general and covers many models considered in the smart grid literature. Moreover, the battery model can be easily extended on a local level of a single RES without changing the structure of the overall grid, only connected through the aggregated power demand $\sum_{i=1}^{\mathcal{I}} z_i$.

3.5 Control architectures of the electricity grid

For the control of electricity grids, many information exchange structures are possible [91]. Most of them can be described by one of the structures given in Figure 3.7 and in Figure 3.8. In the centralized setting, Figure 3.7 (left), the energy provider takes all the decisions for the RESs. In this case, a central controller is considered to compute the optimal solution of the overall problem based on the data provided from the RESs. The individual RESs play no role in the decision making process of how to charge or discharge their battery and only have to execute the decision taken by the grid operator or the CE. This setting is very inflexible with respect to changes in the network. The central controller needs to

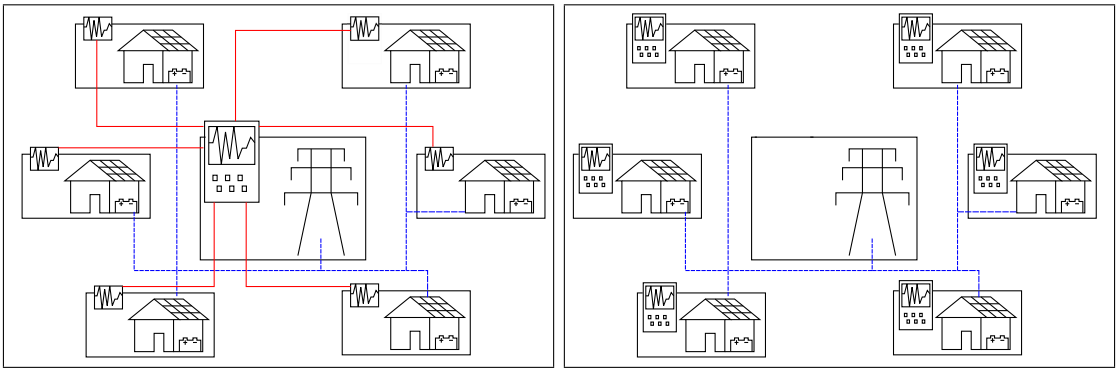

Figure 3.7: *Centralized (left) and decentralized (right) control structure of the electricity grid.*

be adapted every time a local RES changes its settings. A centralized MPC scheme is, for example, used in [50] for the optimal operation of an isolated microgrid. In this thesis, a centralized MPC algorithm is discussed in the next chapter in Section 4.1.1.
In the decentralized control framework (Figure 3.7 right) on the other hand, every RES acts on its own. Either selfishly, with respect to its own goals, or cooperatively to achieve a common goal. Regardless of the goals, the main difference compared to the centralized setting is that there is no possibility to exchange information between the RESs and the grid operator. The grid operator cannot influence the decisions made by the individual participants of the grid. On the other hand, these limitations in the control from the grid operator point of view makes the network very flexible with respect to RESs settings. Decentralized control schemes are usually discussed in papers where the well-being of a single RES is considered and not the performance of the overall grid. For example in [49], an economic MPC scheme is proposed to minimize the electricity costs of a heat pump in a building. We discuss a decentralized control setting in Section 4.1.2.

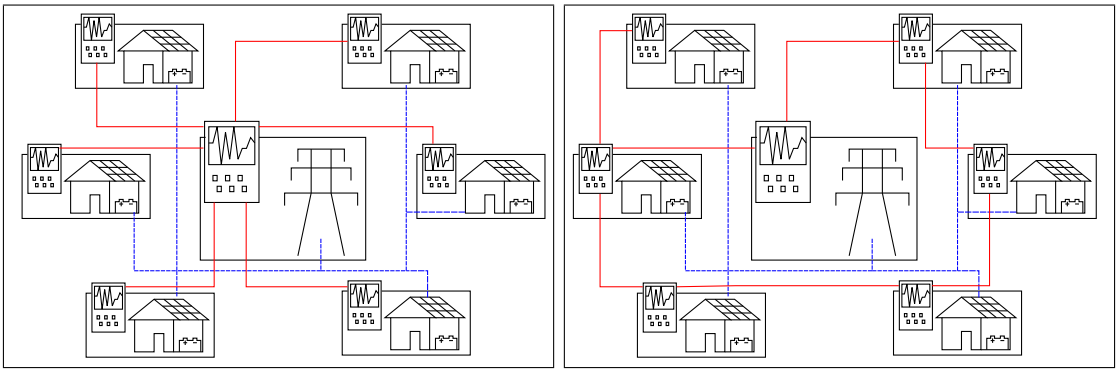

Figure 3.8: *Hierarchical distributed (left) and distributed (right) control structure of the electricity grid. The red lines indicate communication and show a communication between the RESs and the CE (left) and a neighbor-to-neighbor communication (right).*

In the distributed control framework in Figure 3.8, usually two settings are distinguished. In the hierarchical structure, Figure 3.8 (left), every RES is connected to the CE like a grid

operator similar to the centralized setting. But in contrast to the centralized framework, the RESs have their own controller and can individually make their decisions. The CE ensures that important information is communicated between the RESs. This allows a flexibility of decentralized settings with additional structure due the information exchange. In Chapters 5 to 7, hierarchical distributed optimization algorithms are proposed and analyzed for this framework.
In the distributed control framework visualized in Figure 3.8 (right), the CE may or may not exist. In this setting, every RES has its own controller and can communicate with its neighbors. In this context, algorithms and assumptions are usually derived based on the underlying graph connecting the RESs. Control architectures with neighbor-to-neighbor communication are, for example, considered [60, 61], [26] and in [59]. In this thesis, we only analyze the hierarchical distributed framework and not the distributed framework since we put the focus on the interaction between energy provider and RESs and not on the interaction between RESs.

Remark 3.5.1. *Often, a proposed control scheme cannot be characterized exactly into centralized, decentralized, or distributed settings. Moreover, the expressions centralized, distributed and decentralized are used differently in different fields and different publications depending on the application. This section has introduced these notions as they are used in the upcoming chapters.*

It is worth mentioning that the modeling of the RESs in Section 3.4 together with the problem formulation in Section 3.3 allows for all three different control concepts. The RESs are physically decoupled, i.e., only the unconstrained power demand z_i is influenced by all the RESs. Hence, decentralized control schemes are possible but one would expect that the result with respect to the problem formulation in Section 3.3 improves with the amount of information exchanged between the RESs. For our setting, this can be observed in the numerical simulations in the Sections 4.6 and 5.5.

Chapter 4

Predictive control in the context of smart grids

In [109], it is argued that a simple rule-based controller is not able to exploit the degrees of freedom in terms of storage devices in the electricity grid introduced in Section 3.4. To have a significant impact on the overall power demand $\sum_{i=1}^{\mathcal{I}} \mathbf{z}_i(\cdot)$, sophisticated control schemes are necessary. In this chapter, we formalize the concept of MPC introduced in Section 2.3, in the context of small scale electricity grids. In particular we formalize economic MPC characteristics of the underlying control problem by considering appropriate cost functions to satisfy the goals of the energy provider. In this context, the mathematical formulation of the goal of the energy provider of peak-shaving is defined.

4.1 Centralized and decentralized predictive control

In (2.6) and (2.7), the notion of the average cost functionals were introduced. In this section, we give a specific choice for the cost functionals starting with the centralized control setting and then continuing with the decentralized setting. Moreover, we compare the two approaches and emphasize their advantages and disadvantages.
Throughout this chapter we assume that the communication variable is one-dimensional, i.e., $z_i \in \mathbb{R}$ for all $i \in \mathbb{N}_{\mathcal{I}}$, which conforms with the system dynamics introduced in Section 3.4. An extension to the case $z_i \in \mathbb{R}^p$, $p \in \mathbb{N}$, for all $i \in \mathbb{N}_{\mathcal{I}}$, is straightforward.

4.1.1 The centralized control setting

Assume that the sequence of partial sums $(S_N)_{N\in\mathbb{N}}$,

$$S_N := \frac{1}{N} \sum_{j=0}^{N-1} \hat{w}(j) - \hat{g}(j) \tag{4.1}$$

is convergent, i.e., the overall average demand exists and is given by $\overline{\overline{\zeta}} := \lim_{N\to\infty} S_N$. Then a natural choice for the running costs for peak-shaving is to penalize the deviation

from the average at every time instant, i.e.,

$$\ell(z) = \left(\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} z_i - \overline{\overline{\zeta}} \right)^2 = \left(\hat{z} - \overline{\overline{\zeta}} \right)^2. \tag{4.2}$$

As already pointed out, $\mathbf{w}(\cdot)$ and $\mathbf{g}(\cdot)$ correspond to the power consumption and the power generation, respectively, and can thus only be estimated up to some hours into the future. Hence, the average demand at time k has to be computed based on $\mathbf{w}(k;N)$ and $\mathbf{g}(k;N)$ for a given prediction horizon $N \in \mathbb{N}$. We define the reference value for the centralized running costs at time k as

$$\hat{\zeta}(k) = \frac{1}{N\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \sum_{j=k}^{k+N-1} w_i(j) - g_i(j) = \frac{1}{N} \left(\hat{w}(k;N) - \hat{g}(k;N) \right) \mathbb{1}^T \tag{4.3}$$

and the running costs $\ell^k : \mathbb{R}^{\mathcal{I}} \to \mathbb{R}$ is given by

$$\ell^k(z) = \left(\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} z_i - \hat{\zeta}(k) \right)^2 = \left(\hat{z} - \hat{\zeta}(k) \right)^2.$$

With this definition, the cost functional[1] for the centralized control structure at time k is defined as

$$J_N(x(k), \mathbf{u}(k;N)) = \sum_{j=k}^{k+N-1} (\hat{z}(j) - \hat{\zeta}(k))^2 = \left\| \hat{\mathbf{z}}(k;N) - \hat{\zeta}(k)\mathbb{1} \right\|^2. \tag{4.4}$$

Observe that the cost functional is time-varying due to the reference value $\hat{\zeta}(k)$. The CMPC algorithm is summarized in Algorithm 2. The CMPC algorithm is applicable to any system dynamics introduced in Section 3.4. The system dynamics only indirectly appears in the definition of the admissible set $\mathbb{U}^{k,N}(x_0)$. The performance of Algorithm 2 is analyzed in Section 4.6.

4.1.2 The decentralized control setting

The difference between the decentralized and the centralized control setting lies in the communication and in the place where the decision is taken. In decentralized control, no information is exchanged between the individual RESs and hence every RES is completely independent from a CE and the other participants in the network. This implies that RES i, $i \in \mathbb{N}_{\mathcal{I}}$, is only able to optimize its power demand z_i with respect to its own predicted average demand

$$\zeta_i(k) = \frac{1}{N} \sum_{j=k}^{k+N-1} (w_i(j) - g_i(j)) = \frac{1}{N} (w_i(k;N) - g_i(k;N)) \mathbb{1}^T, \tag{4.6}$$

[1] Here the cost functional instead of the average cost functional is used to simplify the notation. Observe that the cost functional and the corresponding average cost functional have the same minimizers $\mathbf{u}^\star(k;N)$.

Algorithm 2 Centralized model predictive control algorithm

RES i, $i \in N_{\mathcal{I}}$:

1. Measure the current state $x_i(k) \in \mathbb{X}_i$, set $x_{i,0} = x_i(k)$ and predict the power demand and the power generation $\mathbf{w}_i(k;N)$, $\mathbf{g}_i(k;N)$.

2. Send $x_{i,0}$, $\mathbf{w}_i(k;N)$ and $\mathbf{g}_i(k;N)$ to the CE.

Central Entity:

3. Compute the average power demand

$$\hat{\zeta}(k) = \frac{1}{N}\left(\hat{\mathbf{w}}(k;N) - \hat{\mathbf{g}}(k;N)\right)\mathbb{1}^T$$

   to define the running costs ℓ^k and the objective function J_N.

4. Solve the OCP

$$\mathbf{u}^\star(k;N) \in \operatorname{argmin}_{\mathbf{u}(k;N)\in\mathbb{U}^{k,N}(x_0)} J_N(x_0, \mathbf{u}(k;N)) \tag{4.5}$$

   to obtain $\mathbf{u}_i^\star(k;N)$ for all $i \in N_{\mathcal{I}}$.

5. Define the feedback $\mu_i(x(k)) := u_i^\star(k)$ and send $\mu_i(x(k))$ to the individual RESs.

Shift the horizon by setting $k = k+1$ and go to step 1.

which gives rise to the local running costs $\ell_i^k : \mathbb{R} \to \mathbb{R}$ defined by

$$\ell_i^k(z_i) = (z_i - \zeta_i(k))^2 \tag{4.7}$$

and the local objective functions

$$J_{i,N}(x_i(k), \mathbf{u}_i(k;N)) = \sum_{j=k}^{k+N-1} \ell_i^k(z_i(j)) = \|\mathbf{z}_i(k;N) - \zeta_i(k)\mathbb{1}\|^2 . \tag{4.8}$$

The corresponding procedure is given in Algorithm 3. Instead of one optimization problem in centralized control, $\mathcal{I}$ optimization problems have to be solved in every time step of the decentralized algorithm. Additionally, since no communication between the systems is necessary, the CE vanishes. To be able to compare Algorithm 2 and Algorithm 3 on a quantitative level, we have to introduce appropriate performance metrics.

4.2 Performance metrics and the cost functional

In this section, several performance metrics for the MPC algorithms are motivated. Additionally, alternative cost functionals are discussed to shift the focus to different performance metrics.

Algorithm 3 Decentralized model predictive control algorithm

RES i, $i \in N_{\mathcal{I}}$:

1. Measure the current state $x_i(k) \in \mathbb{X}_i$, set $x_{i,0} = x_i(k)$ and predict the power demand and the power generation $\mathbf{w}_i(k;N)$, $\mathbf{g}_i(k;N)$.

2. Compute the average power demand

$$\zeta_i(k) = \frac{1}{N}\left(\mathbf{w}_i(k;N) - \mathbf{g}_i(k;N)\right)\mathbb{1}^T$$

   to define the running costs ℓ_i^k and the objective function $J_{i,N}$

3. Solve the OCP

$$\mathbf{u}_i^\star(k;N) \in \operatorname{argmin}_{\mathbf{u}_i(k;N)\in\mathbb{U}_i^{k,N}(x_{i,0})} J_{i,N}(x_{i,0}, \mathbf{u}_i(k;N)).$$

   to obtain $\mathbf{u}_i^\star(k;N)$.

5. Define the feedback $\mu_i(x_i(k)) := u_i^\star(k)$.

Shift the horizon by setting $k = k+1$ and go to step 1.

4.2.1 Performance metrics of predictive control schemes

As emphasized in Chapter 3, the main objective is on peak-shaving from the energy provider point of view. Moreover, since it is not possible to solve the average functional (2.7) on an infinite horizon, we use closed-loop solutions $\mathbf{z}^\mu(0;\mathcal{N})$ of a simulation of length $\mathcal{N} \in \mathbb{N}$ or open-loop solutions $\mathbf{z}^\star(k;N)$ over the prediction horizon N to measure the performance.

Remark 4.2.1. *Performance estimates for $\mathbf{z}^\mu(\cdot)$ based on the open-loop solutions $\mathbf{z}^\star(k;N)$ and based on the length of the prediction horizon $N \in \mathbb{N}$ are beyond the scope of this thesis. Thus, we use performance metrics on the finite closed-loop solution $\mathbf{z}^\mu(0;\mathcal{N})$, where $\mathcal{N} \in \mathbb{N}$ denotes the number of considered time steps. To be able to compare the results of the closed-loop solution and the open-loop solution $\mathbf{z}^\star(k;N)$ and to show that the properties from the open loop carry over to the closed loop we additionally consider performance metrics for the open-loop solutions.*

The peak-to-peak (PTP) variation is defined as the difference between the maximal and the minimal power demands

$$\max_{k=0,\dots,\mathcal{N}-1} \hat{z}^\mu(k) - \min_{k=0,\dots,\mathcal{N}-1} \hat{z}^\mu(k) \tag{PTP}$$

$$\max_{j=k,\dots,k+N-1} \hat{z}^\star(j) - \min_{j=k,\dots,k+N-1} \hat{z}^\star(j) \tag{PTP$_k$}$$

either for the closed-loop or for the open-loop solution. The mean-quadratic-deviations

(MQDs)

$$\frac{1}{\mathcal{N}} \left\| \hat{\mathbf{z}}^{\mu}(0;\mathcal{N}) - \frac{\hat{\mathbf{z}}^{\mu}(0;\mathcal{N})\mathbb{1}^T}{\mathcal{N}}\mathbb{1} \right\|^2, \tag{MQD}$$

$$\frac{1}{N} \left\| \hat{\mathbf{z}}^{\star}(k;N) - \frac{\hat{\mathbf{z}}^{\star}(k;N)\mathbb{1}^T}{N}\mathbb{1} \right\|^2, \tag{MQD$_k$}$$

in the closed-loop and in the open-loop formulation, measure the quadratic deviation from the average power demands $\frac{1}{\mathcal{N}}\hat{\mathbf{z}}^{\mu}(0;\mathcal{N})\mathbb{1}^T$ and $\frac{1}{N}\hat{\mathbf{z}}^{\star}(k;N)\mathbb{1}^T$, respectively. The MQD resembles the cost function J_N except for the computation of the average demand. Observe that in the case with losses in the battery model, the power demand z is not constant over the time interval of consideration and hence cannot be computed by

$$\frac{1}{\mathcal{N}} \left(\hat{\mathbf{w}}(0;\mathcal{N}) - \hat{\mathbf{g}}(0;\mathcal{N}) \right) \mathbb{1}^T \qquad \text{and} \qquad \frac{1}{N} \left(\hat{\mathbf{w}}(k;N) - \hat{\mathbf{g}}(k;N) \right) \mathbb{1}^T,$$

respectively. The average loss of energy (LOE) can be measured by

$$\frac{1}{\mathcal{N}} \left(\left(\hat{\mathbf{z}}^{\mu}(0;\mathcal{N}) \right) \mathbb{1}^T - \left(\hat{\mathbf{w}}(0;\mathcal{N}) - \hat{\mathbf{g}}(0;\mathcal{N}) \right) \mathbb{1}^T + \frac{1}{T} \left(\hat{x}(0) - \hat{x}^{\mu}(\mathcal{N}) \right) \right), \tag{LOE}$$

$$\frac{1}{N} \left(\left(\hat{\mathbf{z}}^{\star}(k,N) \right) \mathbb{1}^T - \left(\hat{\mathbf{w}}(k;N) - \hat{\mathbf{g}}(k;N) \right) \mathbb{1}^T + \frac{1}{T} \left(\hat{x}(k) - \hat{x}^{\star}(k+N) \right) \right). \tag{LOE$_k$}$$

Since the storage devices should not be used to waste energy, the loss of energy should be as small as possible, even if it is not covered in the cost functional.
Peak-shaving of the aggregated power demand can also be interpreted as the minimization of the deviation in consecutive time steps k and $k+1$, for this purpose we additionally define the averaged smoothing factors (ASFs)

$$\frac{1}{\mathcal{N}-1} \sum_{k=0}^{\mathcal{N}-2} \left(\hat{z}^{\mu}(k+1) - \hat{z}^{\mu}(k) \right)^2, \tag{ASF}$$

$$\frac{1}{N-1} \sum_{j=k}^{N-2} \left(\hat{z}^{\star}(j+1) - \hat{z}^{\star}(j) \right)^2. \tag{ASF$_k$}$$

4.2.2 Extensions of the objective function

The performance metrics motivate the definition of different objective functions J_N and $J_{i,N}$, for $i = 1, \ldots, \mathcal{I}$, respectively. The LOE can be taken into consideration by additionally penalizing the usage of the batteries. In this case, for $\eta > 0,\ \nu > 0$,

$$J_N(x(k), \mathbf{u}(k,N)) = \eta \left\| \hat{\mathbf{z}}(k;N) - \hat{\zeta}(k)\mathbb{1} \right\|^2 + \nu \sum_{i=1}^{\mathcal{I}} \|\mathbf{u}_i(k;N)\|^2 \tag{4.9}$$

penalizes the charging and discharging behavior of the RESs. The parameters η and ν can be used to shift the focus from optimizing with respect to PTP and MQD to LOE. To

obtain good performance with respect to ASF, the cost functional

$$J_N(x(k), \mathbf{u}(k;N)) = \sum_{j=k}^{N-2} \left(\hat{\mathbf{z}}(j+1) - \hat{\mathbf{z}}(j)\right)^2 \tag{4.10}$$

is defined.[2] More generally, the cost functional

$$J_N(x(k), \mathbf{u}(k;N)) = (\hat{\mathbf{z}} - \zeta(k))Q(\hat{\mathbf{z}} - \zeta(k))^T \tag{4.11}$$

for a positive semidefinite matrix $Q \in \mathbb{R}^{N\times N}$ and a reference vector $\zeta(k) \in \mathbb{R}^N$ can be used. For example, (4.10) is realized through $\zeta(k) = 0$ and

$$Q = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 2 & -1 \\ & & & -1 & 1 \end{pmatrix}.$$

The local objective functions in the decentralized setting in Section 4.1.2 can be extended in a similar way. Most of our numerical results will use (4.4). However, it should be noted that the cost functional can be replaced and the results in the following will still hold.

4.3 Comparison of centralized and decentralized control

In the centralized MPC Algorithm 2, the CE takes all decisions for the overall system. The OCP (4.5) is based on the knowledge of the admissible sets $\mathbb{U}_i^{k,N}(x_{i,0})$ for all $i \in \mathbb{N}_{\mathcal{I}}$, and the average $\hat{\zeta}(k)$ defines the average of all systems. Hence, the CE has all the information defining the dynamical behavior of the electricity grid, which implies that a good result with respect to the performance metrics can be expected. Due to the coupling in the objective function, RESs with a high/low demand can compensate for other RESs.
In the decentralized MPC Algorithm 3, in contrast, the RESs are decoupled and they take their decisions independent of the other RESs in the network. Furthermore, only the deviation from the individual average demand $\zeta_i(k)$ is penalized which prevents that one RES compensates for other RESs. Hence, regarding the performance, the centralized control setting is expected to lead to better results than the decentralized setting. This gap in the performance is also visualized in Section 4.6 where the two algorithms are compared. However, the knowledge of all components by the CE also contains disadvantages. Every RES has to report its system dynamics and its constraints to the CE, which might be undesirable for the customers for reasons of privacy of data. If a single RES changes its system dynamics or constraints, for example, by adding an additional storage device or power generation unit, these changes have to be reported to the CE and the whole

[2]Note that the cost functional (4.9) cannot be written in terms of running costs ℓ^k. Regardless, the cost functional can be used to define an OCP solved at every time step in a receding horizon scheme.

controller has to be modified. The same situation occurs if the network is enlarged by a new RES, making the whole approach very inflexible for electricity grids with rapidly changing components.
Moreover, the dimension of the optimization problem grows linearly with the number of RESs connected to the grid. This implies a cubic increase of the complexity of the OCP due to the size of the underlying linear system solved in the optimization algorithm.[3] [4] Therefore, the CMPC algorithm is not suitable for real-time applications with a high number of RESs but can be used as a benchmark to assess the quality of different control approaches.
In the decentralized MPC algorithm no information on the system dynamics and the constraints is shared, i.e., privacy of data is not an issue. Also, changes in the system dynamics only have impact on the local level and the local controller which makes the control approach very flexible. Furthermore, the dimension of the OCPs is independent of the number of RESs. Every RES has its own controller performing the optimization steps in parallel. Hence, the size of the grid has no impact on the real-time applicability in this control setting.

4.4 From the open-loop to the closed-loop solution

In MPC, a feedback controller is computed by iteratively solving OCPs, i.e., open-loop solutions are used to define the feedback law μ. In this section, we point out three properties of MPC which arise when we consider the closed-loop solution instead of the open-loop solution. In Section 4.4.1, we discuss non-unique solutions in the OCP. In Section 4.4.2, we introduce the concept of warm-start which is commonly used in the MPC context to reduce the computational complexity and in Section 4.4.3 we investigate the robustness of MPC and in particular robustness with respect to our application in smart grids.

4.4.1 Non-uniqueness of the optimizer $\mathbf{z}^\star$

The centralized cost functional (4.4), $J_N : \mathbb{R}^{\mathcal{I}\times N} \to \mathbb{R}$,

$$J_N(\mathbf{z}(k;N)) = \left\| \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i(k;N) - \hat{\zeta}(k) \right\|^2,$$

defined in the communication variables $\mathbf{z}(k;N)$ instead of $x(k)$ and $\mathbf{u}(k;N)$, is convex but not strictly convex. Hence, even though a unique minimum

$$\begin{array}{rcll} J_N^{\star k} & = & \min_{\mathbf{z}(k;N)} & J_N(\mathbf{z}(k;N)) \\ & & \text{s.t.} & \mathbf{z}(k;N) \in \mathbb{D}^{k,N}(x(k)) \end{array}$$

[3] The cubic increase refers to an implementation without exploiting any structure. Depending on the optimization problem it might be possible to decrease the complexity by exploiting sparsity for example.
[4] Possible algorithms are for example sequential quadratic programming (SQP) or interior point methods [78].

exists under suitable convexity and compactness conditions on the set

$$\mathbf{u}(k;N) \in \mathbb{U}^{k,N}(x(k))$$

and suitable conditions on the dynamics (2.2), the minimizer $\mathbf{z}^\star(k;N)$ such that $J_N^\star(k) = J_N(\mathbf{z}^\star(k;N))$ holds, does not have to be unique. The choice of the optimizer $\mathbf{z}^\star(k;N)$ is not important for the solution of the OCP, but it may have an impact on the closed-loop solution, since the admissible set $\mathbb{U}^{k+1,N}(x(k+1))$ at the next time step depends on the choice of the optimizer $\mathbf{z}^\star(k;N)$. We illustrate this scenario in the following example.

Example 4.4.1. *Consider the simplified system dynamics*

$$\begin{aligned} x_i(k+1) &= x_i(k) + u_i(k) \\ z_i(k) &= u_i(k) + s_i(k) \end{aligned}$$

for $i = 1, 2$ and fix the horizon $N = 2$. The initial states are set to $x_i(0) = 1$ and the first elements of the time-dependent sequences are given by $\mathbf{s}_i(0;3) = (1 \;\; 1 \;\; -1)$ for $i = 1, 2$. The dynamics and the inputs are subject to the constraints

$$x_i(j) \in [0, 2], \qquad u_i(j) \in [-1, 1]$$

for $i = 1, 2$ and $j \in \mathbb{N}$.
For the time instant $k = 0$, we get

$$\hat{\zeta}(0) = \frac{1}{2 \cdot 2} \sum_{i=1}^{2} \sum_{j=0}^{1} s_i(j) = 1.$$

Hence, the OCP

$$\begin{array}{rcll} J_2^{\star 0} & = & \min\limits_{\mathbf{z}(0;2)} & J_2(\mathbf{z}(0;2)) \\ & & s.t. & x(0) = (1 \; 1)^T \\ & & & x(j+1) = x(j) + u(j) \quad \textit{for } j = 0, 1 \\ & & & z(j) = u(j) + s(j) \quad \textit{for } j = 0, 1 \\ & & & x(j) \in [0, 2]^2 \quad \textit{for } j = 0, 1 \\ & & & u(j) \in [-1, 1]^2 \quad \textit{for } j = 0, 1 \end{array}$$

leads to the optimal value $J_N^{\star 0} = 0$ which is, for example, obtained for

$$J_2^{\star 0} = J_2(\mathbf{z}^\star(0;2)) = J_2(\mathbf{z}^{\star\star}(0;2))$$

defined through the trajectories

$$\mathbf{u}^\star(0;2) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \qquad \mathbf{z}^\star(0;2) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix},$$

and

$$\mathbf{u}^{\star\star}(0;2) = \begin{pmatrix} 1 & 0 \\ -1 & 0 \end{pmatrix}, \qquad \mathbf{z}^{\star\star}(0;2) = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}.$$

At time instant $k = 1$, the average power demand is given by $\hat{\zeta}(1) = 0$. If we define the feedback $\mu^\star(x(0)) = \mathbf{u}^\star(0)$, the initial state $\mathbf{x}(1) = (1\ 1)^T$ is obtained. The corresponding OCP at time $k = 1$ is defined as

$$\begin{array}{lll} \min\limits_{\mathbf{z}(1;2)} & J_2(\mathbf{z}(0;2)) & \\ s.t. & x(1) = (1\ 1)^T & \\ & x(j+1) = x(j) + u(j) & \text{for } j = 1,2 \\ & z(j) = u(j) + s(j) & \text{for } j = 1,2 \\ & x(j) \in [0,2]^2 & \text{for } j = 1,2 \\ & u(j) \in [-1,1]^2 & \text{for } j = 1,2 \end{array}$$

with optimal solution $J_2^\star(1) = J_2(\mathbf{z}^\star(1;2)) = 0$ and

$$\mathbf{u}^\star(1;2) = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix}, \qquad \mathbf{z}^\star(1;2) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

In contrast, if we define the feedback $\mu^{\star\star}(x(0)) = \mathbf{u}^{\star\star}(0)$, the initial state is given by $\mathbf{x}(1) = (2\ 0)^T$ leading to the OCP

$$\begin{array}{lll} \min\limits_{\mathbf{z}(1;2)} & J_2(\mathbf{z}(1;2)) & \\ s.t. & x(1) = (2\ 0)^T & \\ & x(j+1) = x(j) + u(j) & \text{for } j = 1,2 \\ & z(j) = u(j) + s(j) & \text{for } j = 1,2 \\ & x(j) \in [0,2]^2 & \text{for } j = 1,2 \\ & u(j) \in [-1,1]^2 & \text{for } j = 1,2. \end{array}$$

Since the battery of RES 2 is empty, an optimal solution is given by

$$\mathbf{u}^{\star\star}(1;2) = \begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix}, \qquad \mathbf{z}^{\star\star}(1;2) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix},$$

i.e., $J_2^{\star\star_1} = J_2(\mathbf{z}^{\star\star}(1;2)) = \frac{1}{4} > J_2^{\star_1}$. This implies that different optimal open-loop solutions can lead to different closed-loop solutions and different performances.

Remark 4.4.2. *This result motivates the investigation of the performance of the MPC closed-loop solution depending on the optimization algorithm. For example, active set methods, such as sequential quadratic programming, or interior point methods (see [78] and [17]) may lead to different solutions.*

4.4.2 Warm-start

The real-time applicability of MPC algorithms strongly depends on the complexity of the OCP solved at every time instant. Since algorithms like interior point methods or sequential quadratic programming iteratively compute an optimal solution, the number of iterations to satisfy a given stopping criteria has a significant impact on the overall computation

time. The number of iterations can be reduced by a good initial guess $\mathbf{z}^0(k;N)$ which is close to an optimal solution, i.e., $\left\|\mathbf{z}^0(k;N) - \mathbf{z}^\star(k;N)\right\|$ is small and $\mathbf{z}^\star(k;N)$ denotes an optimal solution of the OCP at time k.
Here, MPC provides a natural strategy for the initialization of $\mathbf{z}^0(k+1;N)$ based on the optimal solution $\mathbf{z}^\star(k;N)$ from the previous time instant. This method, called warm-start, can be used in our context as well. Let $\mathbf{z}_i^\star(k;N)$ and $\mathbf{z}_i^\star(k+1;N)$ denote two consecutive optimal solutions for $i \in \mathbb{N}_{\mathcal{I}}$. Since the underlying optimization problems only differ in the initial value $x_i(k)$ and $x_i(k+1)$, respectively, and in the shifted sequences $\mathbf{w}_i(\cdot)$ and $\mathbf{g}_i(\cdot)$, we expect that the reference value and the optimal communication variables only change slightly, i.e., $\hat{\zeta}(k) \approx \hat{\zeta}(k+1)$ and $z^\star(j)_{x(k)} \approx z^\star(j)_{x(k+1)}$[5] for all $j \in \{k+1,\dots,k+N-1\}$. Hence, we define the initialization

$$\begin{aligned} z^0(j)_{x(k+1)} &:= z^\star(j)_{x(k)} \\ u^0(j)_{x_i(k+1)} &:= u^\star(j+1)_{x(k)} \end{aligned}$$

for all $j \in \{k+1,\dots,k+N-1\}$. The values $u^0(k+N)_{x(k+1)}$ and $z^0(k+N)_{x(k+1)}$ have to be chosen such that $\mathbf{u}^0(k+1,N) \in \mathbb{U}^{k+1,N}(x(k+1))$ and $\mathbf{z}^0(k+1,N) \in \mathbb{D}^{k+1,N}(x(k+1))$, depending on the dynamics of the model. For details on warm-start in MPC, see [46, Subsection 10.5].

4.4.3 Robustness of model predictive control schemes

The model dynamics (2.1) used to predict the dynamical behavior of a RES cannot capture all the details of a real storage device in general. Moreover, the prediction of the power demand relies on a perfect knowledge of the future power generation and power consumption. Since this assumption, is not realistic, we have to assume that the system dynamics is only modeled up to a certain accuracy and is subject to additional unknown disturbances which have to be compensated by the MPC feedback.
MPC computes a feedback iteratively by solving an OCP at every time step, adjusting the errors made through the unknown deviations in the dynamics and hence possesses certain inherent robustness properties even if the dynamics are only approximated. In this section, we investigate how the robustness properties of MPC in the context of the proposed electricity grid can be examined. We concentrate on uncertainties in the predicted sequences $\mathbf{w}(\cdot)$ and $\mathbf{g}(\cdot)$ where potentially the biggest inaccuracies occur due to errors in the weather forecast and due to the attitude of the consumers who might spontaneously change their consumption patterns during the day. To this end, we introduce the disturbance sequences $(r_i(k))_{k\in\mathbb{N}} \subset \mathbb{R}$ and $(d_i(k))_{k\in\mathbb{N}} \subset \mathbb{R}$ and the disturbances $(\Delta\mathbf{r}_{k,i})_{k\in\mathbb{N}} \subset \mathbb{R}^N$ and $(\Delta\mathbf{d}_{k,i})_{k\in\mathbb{N}} \subset \mathbb{R}^N$ for all $i \in \mathbb{N}_{\mathcal{I}}$. We assume that the disturbance sequences are of a special form relating $\mathbf{r}_i(\cdot)$ and $(\Delta\mathbf{r}_{k,i})_{k\in\mathbb{N}}$, and $\mathbf{d}_i(\cdot)$ and $(\Delta\mathbf{d}_{k,i})_{k\in\mathbb{N}}$, respectively, for all $i \in \mathbb{N}_{\mathcal{I}}$.

Assumption 4.4.3. *Let $(r_i(k))_{k\in\mathbb{N}}$ and $(d_i(k))_{k\in\mathbb{N}}$ be sequences of independently and identically normally distributed random variables with zero mean and standard deviation*

[5] We use the notation $z(j)_{x(k)}$ and $z(j)_{x(k+1)}$ to indicate, that $z(j)_{x(k)}$ is an element of the vector $\mathbf{z}(k;N)$ and $z(j)_{x(k+1)}$ is an element of the vector $\mathbf{z}(k+1;N)$. The same holds for the input u.

$\kappa \in \mathbb{R}_{\geq 0}$ for all $i \in \mathbb{N}_{\mathcal{I}}$. Then the disturbances $\Delta\mathbf{r}_i(k;N)$ and $\Delta\mathbf{d}_i(k;N)$ at time $k \in \mathbb{N}$ are defined as

$$\Delta\mathbf{r}_{k,i} := (\Delta\mathbf{r}_{k,i}(1), \ldots, \Delta\mathbf{r}_{k,i}(N))$$
$$\Delta\mathbf{d}_{k,i} := (\Delta\mathbf{d}_{k,i}(1), \ldots, \Delta\mathbf{d}_{k,i}(N))$$

with

$$\Delta\mathbf{r}_{k,i}(j+1) := \Delta\mathbf{r}_{k,i}(j) + r_i(k+j), \quad \Delta\mathbf{r}_{k,i}(1) := 0, \tag{4.12a}$$
$$\Delta\mathbf{d}_{k,i}(j+1) := \Delta\mathbf{d}_{k,i}(j) + d_i(k+j), \quad \Delta\mathbf{d}_{k,i}(1) := 0, \tag{4.12b}$$

for $j = 1, \ldots, N-1$.

Hence at a certain time step k instead of the real power consumption $\mathbf{w}(k,N)$ and the real power generation, only the disturbed vectors

$$\tilde{\mathbf{w}}_i(k;N) := \mathbf{w}_i(k;N) + \Delta\mathbf{r}_i(k;N), \tag{4.13a}$$
$$\tilde{\mathbf{g}}_i(k;N) := \mathbf{g}_i(k,N) + \Delta\mathbf{d}_i(k;N), \tag{4.13b}$$

are known. The disturbed MPC algorithm using the perturbed datasets (4.13) is given in Algorithm 4. The disturbance sequences constructed in this way capture two properties

Algorithm 4 Disturbed model predictive control

1. Measure the current state $x_i(k) \in \mathbb{X}_i$, set $x_{i,0} = x_i(k)$ and predict $\tilde{\mathbf{w}}_i(k;N)$, $\tilde{\mathbf{g}}_i(k;N)$ for all $i \in \mathbb{N}_{\mathcal{I}}$.

2. Compute the average power demand $\hat{\zeta}(k)$ and solve the OCP based on the knowledge of $\tilde{\mathbf{w}}_i(k;N)$, $\tilde{\mathbf{g}}_i(k;N)$.

3. Define the feedback $\mu_i(k) := \mathbf{u}_i^\star(k)$ and implement $\mu_i(k)$ in the system dynamics without disturbances $\mathbf{w}_i(k;N)$, $\mathbf{g}_i(k;N)$.

Shift the horizon by setting $k = k+1$ and go to step 1.

which are appropriate to the application of our smart grid model. The standard deviation increases with the time index j representing that the weather forecast and the consumption patterns are more reliable for time steps in the near future, and due to the construction, the probability that the weather changes dramatically between two consecutive time steps is small. The impact of perturbed data in centralized and decentralized MPC is numerically analyzed in Section 4.6.4 using Monte-Carlo simulations.

Remark 4.4.4. *Note that the disturbance sequences* (4.12) *satisfy the properties of a Wiener process [44, Chapter 4.1 and Chapter 7.1].*

4.5 Prediction of power consumption and power generation

The prediction of reliable forecasts of the power consumption $\mathbf{w}(\cdot)$ and the power generation $\mathbf{g}(\cdot)$ is a topic on its own. There is a broad literature addressing the topic of forecasts in the context of smart grids. In [67], five consumption forecasting techniques are presented and compared to each other. More recent works on consumption forecasting models are presented in [86], [5] and [104]. In [51], an overview of different methods to forecast photovoltaic power generation is provided. Methods are split in short-term forecasts, ranging from 30 minutes to 6 hours, and long-term forecasts over several days. For short-term predictions, cloud covering is considered whereas in the long term predictions numerical weather predictions based on associated models are used. The photovoltaic power generation is predicted based on weather forecasts [63] using methods like machine learning [94] and neural networks [113]. The paper [50] incorporates predictions of power generation and load forecasts into an MPC context of a microgrid. Throughout this thesis, we assume that the sequences $\mathbf{w}(\cdot)$ and $\mathbf{g}(\cdot)$ or at least the approximations defined in Equation (4.13) are known. Nevertheless, in the case where the sequences are not known, the algorithms proposed in the cited papers can still be used in the MPC formulations to predict $\mathbf{w}(k;N)$ and $\mathbf{g}(k;N)$ at every time step k.

4.6 Numerical simulations

In this section, the numerical performance of the decentralized and the centralized control scheme is analyzed. To simplify the analysis, we focus on the linear model dynamics (3.1),

$$\begin{aligned} x_i(k+1) &= x_i(k) + Tu_i(k) \\ z_i(k) &= w_i(k) - g_i(k) + u_i(k) \end{aligned}$$

subject to the constraints (3.2) and (3.3) for all $i \in \mathbb{N}_{\mathcal{I}}$. For the charging and discharging rates the constants $-\underline{u}_i = \overline{u}_i = 0.3$ are used for all $i \in \mathbb{N}_{\mathcal{I}}$. Moreover, the battery capacities are set to $C_i = 2$, the initial state of charge of the batteries is set to $x_i(0) = 0.5$ for all $i \in \mathbb{N}_{\mathcal{I}}$, the discretization parameter is set to $T = 0.5$ and the cost functionals (4.4) and (4.8)

$$J_N(x(k), \mathbf{u}(k;N)) = \left\| \hat{\mathbf{z}}(k;N) - \hat{\zeta}(k)\mathbb{1} \right\|^2, \quad J_{i,N}(x_{i,0}, \mathbf{u}_i(k;N)) = \|\mathbf{z}_i(k;N) - \zeta_i(k)\mathbb{1}\|^2$$

are used for all $i \in \mathbb{N}_{\mathcal{I}}$. The CMPC Algorithm 2 and the DeMPC Algorithm 3 are implemented in MATLAB[6].[7]

4.6.1 Comparison of decentralized and centralized control

In Section 4.3, we argued that the CMPC Algorithm 2 outperforms the DeMPC Algorithm 3. In Figure 4.1 and Figure 4.2, the open-loop solutions of a single OCP cor-

[6] See *http://mathworks.com*

[7] The MATLAB functions `quadprog` and `fmincon` using an interior point algorithm were applied to the minimization problems to compute optimal solutions.

responding to step 2 of Algorithm 2 and Algorithm 3 for a setting of 50 RESs and a prediction horizon of $N = 48$ are compared to the uncontrolled setting. We observe in Figure 4.1, that both approaches reduce the minimal and maximal peaks in the average demand profile $\hat{\mathbf{z}}$ significantly. Nevertheless, due to the additional information available in the centralized OCP, as already expected, the centralized open-loop solution outperforms the decentralized open-loop solution. Since in the decentralized setting every RES

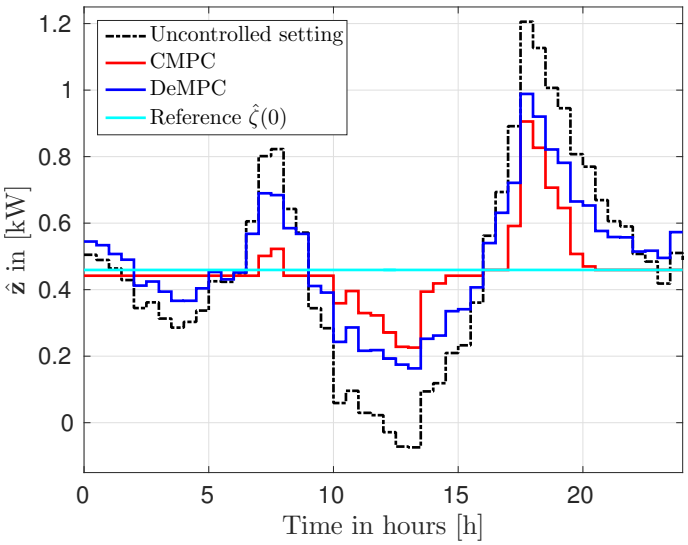

Figure 4.1: *Optimal solution of a single optimization problem using a centralized and a decentralized algorithm. As a reference the uncontrolled demand and the reference value are visualized.*

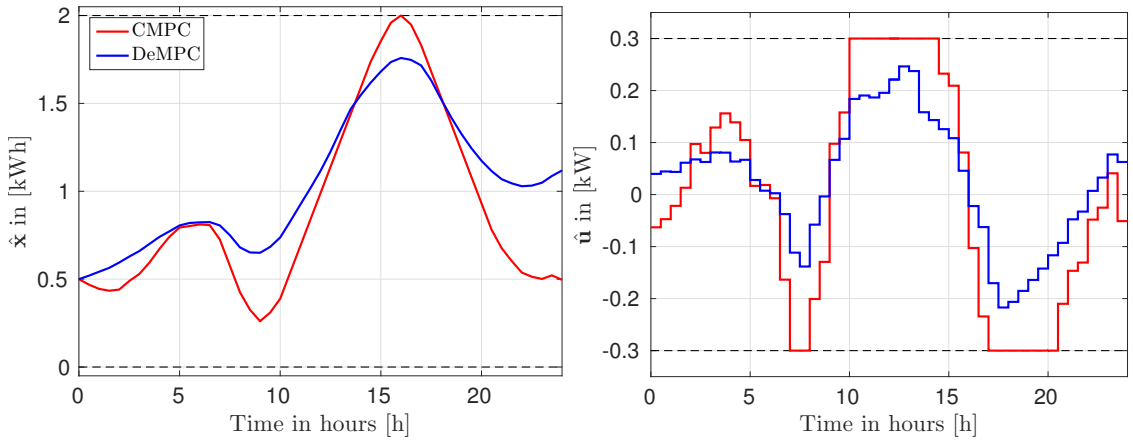

Figure 4.2: *The open-loop trajectories of the batteries (left) and the optimal charging/discharging behaviors (right) corresponding to the power demand in Figure 4.1.*

penalizes the difference to the local reference ζ_i instead of the global average $\hat{\zeta}$ the lack of information results in an open-loop solution, where on average, the bounds on the charging

and discharging rates are not active. This leads to the fact that the capacities of the batteries are not fully used. In contrast, in the centralized OCP one can observe that either the battery constraints or the constraints on the charging/discharging rate are active at time steps where the open-loop solution in Figure 4.1 has a deviation from the reference value. Thus, in the centralized setting, the potential of the batteries is fully used, whereas in the decentralized setting possible performance improvements are lost due to the lack of information.

The same conclusions can be drawn from the closed-loop solutions of Algorithm 2 and 3 visualized in the Figures 4.3–4.5 for a simulation of 1 month (30 days). On the left the full simulation and on the right the first three days of the simulation are visualized. Again, both approaches reduce the fluctuations in the average power demand significantly compared to the uncontrolled setting and CMPC outperforms DeMPC. The results with respect to the

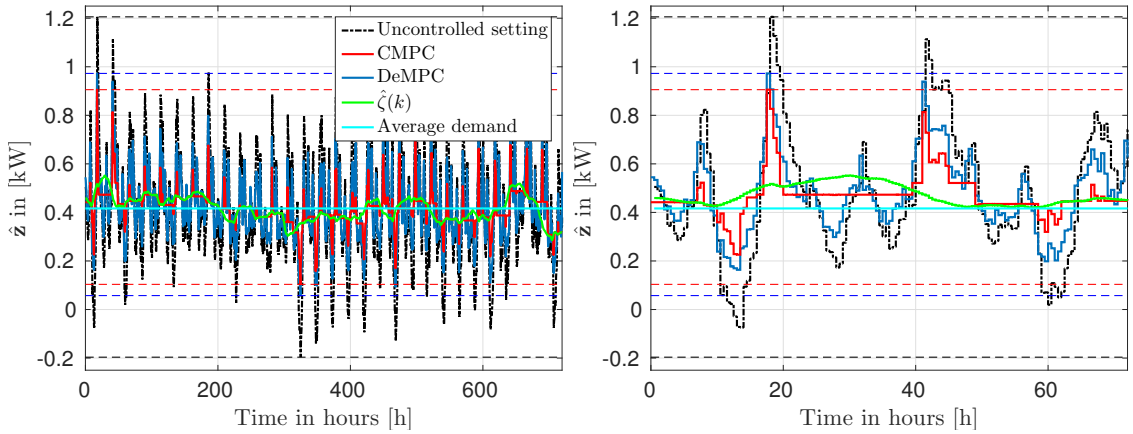

Figure 4.3: *Average power demand of* 50 *RESs simulated over* 30 *days (left) and the first* 3 *days (right).*

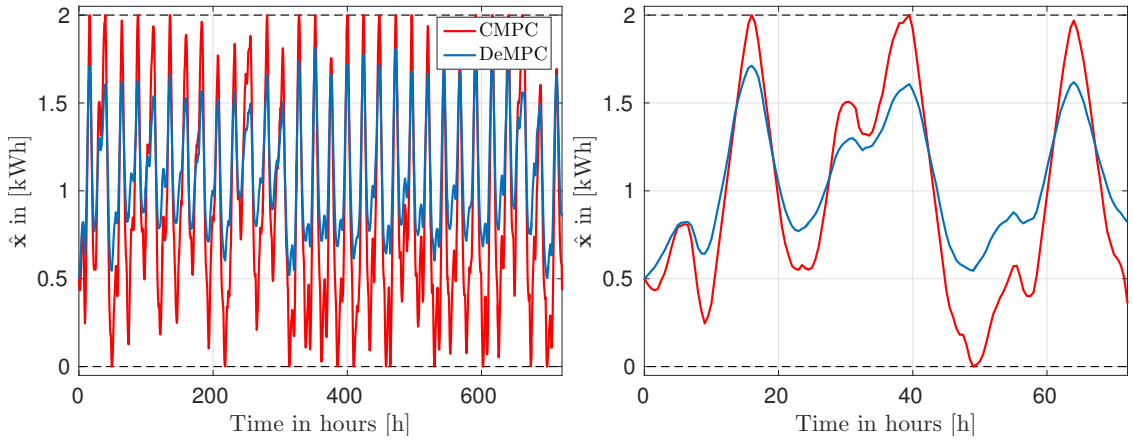

Figure 4.4: *Average state of charge of* 50 *RESs simulated over* 30 *days (left) and the first* 3 *days (right).*

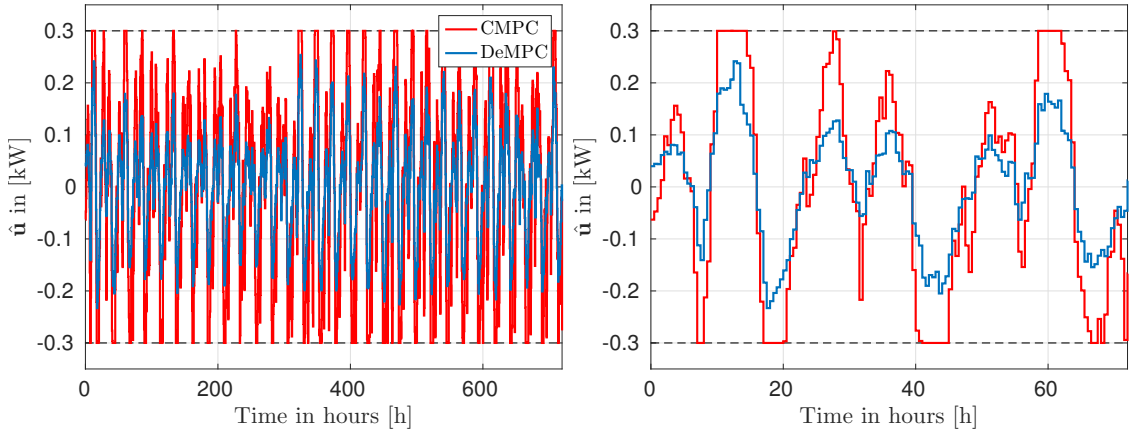

Figure 4.5: *Average charging/discharging profiles of* 50 *RESs simulated over* 30 *days (left) and the first* 3 *days (right).*

performance metrics are summarized in Table 4.1. In the MPC closed-loop simulation, the PTP value is reduced from 1.4020 to 0.8020 which is the maximal reduction possible with the constraints $-\underline{u}_i = \overline{u}_i = 0.3$ for all $i \in \mathbb{N}_{\mathcal{I}}$. To further reduce the vertical fluctuations, the bounds have to be increased. Observe that even though the performance with respect

	open loop $N = 48$			closed loop $\mathcal{N} = 1440$		
	Uncontrolled	DeMPC	CMPC	Uncontrolled	DeMPC	CMPC
PTP	1.2798	0.8253	0.6789	1.4020	0.9152	0.8020
MQD	0.0922	0.0356	0.0138	0.0529	0.0193	0.0052
ASF	0.0121	0.0058	0.0046	0.0074	0.0039	0.0011

Table 4.1: *Performance metrics of the centralized and the decentralized MPC algorithm.*

to the measures MQD and ASF are not covered explicitly by the cost functional, the performances are improved significantly compared to the uncontrolled setting.

4.6.2 The impact of the prediction horizon

In the first simulation we fixed the prediction horizon $N = 48$. In Figure 4.6 the performance of the centralized and the decentralized MPC algorithm are visualized with respect to the prediction horizon N varying from $N = 6$ (i.e., 3 hours) to $N = 60$ (i.e., 30 hours). Again, the performance is measured for 50 RESs and a simulation length of $\mathcal{N} = 1440$ times steps (i.e., 30 days). For the PTP performance, we observe that a prediction horizon of $N = 18$ is enough to obtain the optimal value of the centralized setting. In the decentralized control setting, also the biggest improvements with respect to the PTP values can be observed in a prediction horizon between 6 and 18. Nevertheless, the results still slightly improve for bigger horizons.

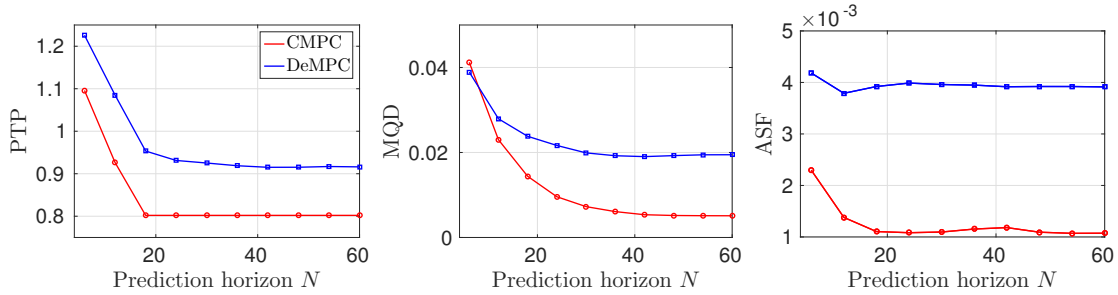

Figure 4.6: *Performance metrics for centralized and decentralized MPC depending on the prediction horizon N.*

4.6.3 Controllable loads

In Section 3.4.2, we argued that the model can be extended to handle controllable loads by adding an additional degree of freedom. In Figure 4.7, the closed-loop solution of the CMPC algorithm is visualized using the system dynamics (3.10)

$$\begin{aligned} x_i(k+1) &= x_i(k) + u_i(k) \\ z_i(k) &= w_i(k) - g_i(k) + u_i(k) + u_{c_i}(k) \end{aligned}$$

for $i = 1, \ldots, 20$. Here, u_{c_i} represents the controllable loads which have to satisfy the constraints (3.12) for $\underline{w}_{c_i} = 0$ and $\overline{w}_{c_i} = 1$ and the time-varying constraints (3.11) for all $i \in \mathbb{N}_{20}$. For the CMPC simulation in Figure 4.7, the scheduling window $\overline{N}$ is set to $\overline{N} = 6$. We assume that 30% of the load is controllable and is hidden in the additional input u_{c_i}. Thus, we define $\mathbf{w}(\cdot) := 0.7 \cdot \mathbf{w}(\cdot)$ and the remaining $0.3 \cdot \mathbf{w}(\cdot)$ are covered by $\mathbf{u}_c$.

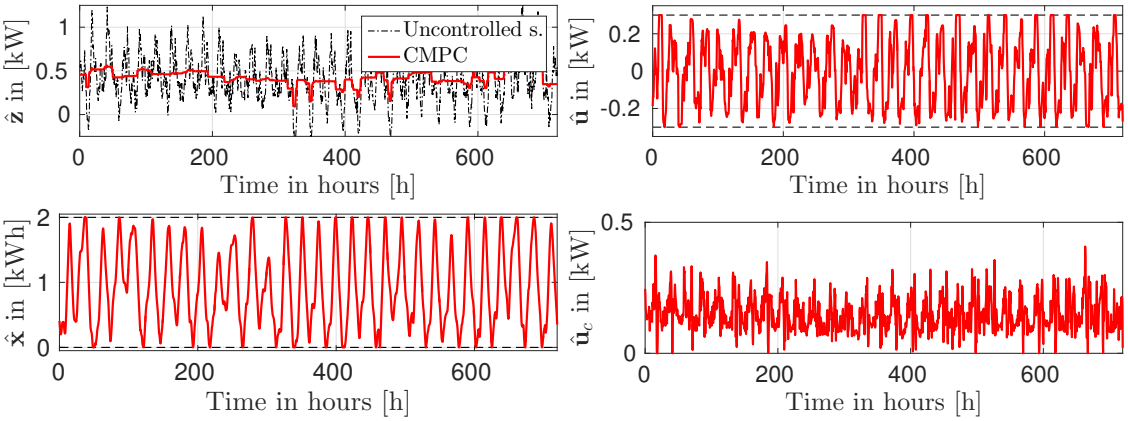

Figure 4.7: *Performance of CMPC including controllable loads.*

As expected, the additional degree of freedom improves the performance of the centralized and the decentralized control scheme. In Figure 4.8, the performance metrics in terms of the scheduling window $\overline{N}$ are visualized. The controllable loads have very little impact on the ASF since the objective function is not chosen according to this criteria. Moreover,

we observe that for $\overline{N} > 18$ in the centralized setting and for $\overline{N} > 30$ in the decentralized setting almost no improvement can be observed and hence, the controllable load does not have to be predicted more than 9 or more than 15 hours in advance. In the centralized setting, the PTP value is not reduced any further since the reference signal $\hat{\zeta}(\cdot)$ changes over time as well.

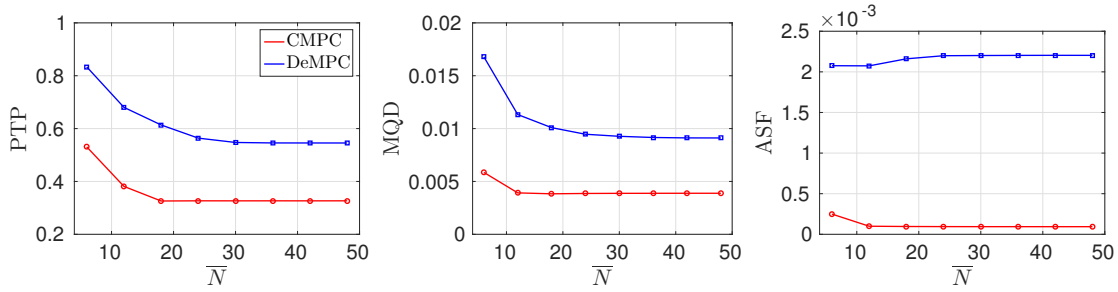

Figure 4.8: *Performance metrics for centralized and decentralized MPC including controllable loads.*

4.6.4 Robustness verification via Monte-Carlo simulations

To verify the robustness of the CMPC scheme (Algorithm 2) and the DeMPC scheme (Algorithm 3) a Monte-Carlo simulation using 1000 realizations of perturbed sequences $\tilde{\mathbf{w}}(\cdot)$ and $\tilde{\mathbf{g}}(\cdot)$ is used in Algorithm 4 to simulate the system dynamics of 20 RESs over a week (i.e., 336 MPC iterations). The perturbed sequences are constructed according to Assumption 4.4.3 with standard deviations $\kappa = 0.1, 0.2, \ldots, 0.5$. The simulations with inaccurate forecast are compared to the nominal case, i.e., $\kappa = 0$. Moreover, in Table 4.2

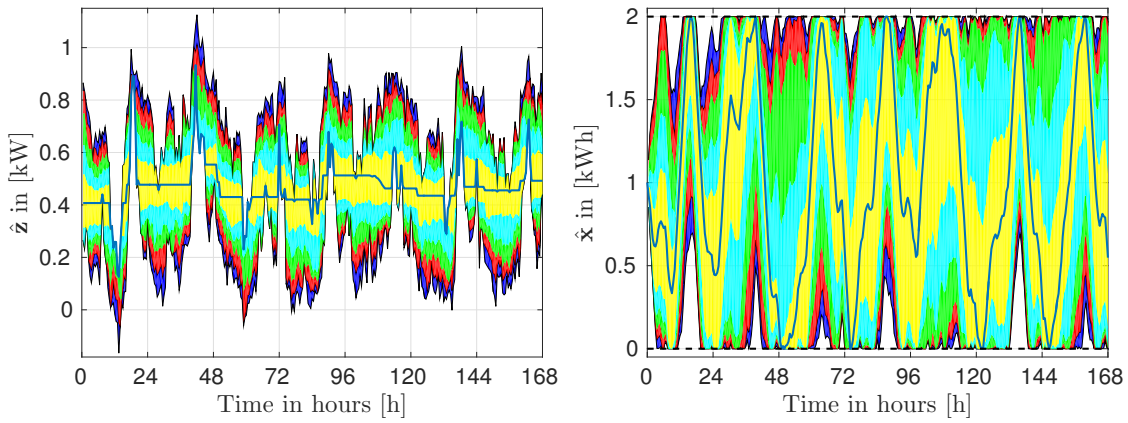

Figure 4.9: *Impact of inaccurate forecasts on the average power demand* $\hat{\mathbf{z}}$ *(left) and the average battery profile* $\hat{\mathbf{x}}$ *(right) for* $\kappa \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ *compared to the nominal trajectory using the CMPC Algorithm 2 combined with Algorithm 4. The areas show the maximal and minimal values of the Monte-Carlo simulation.*

	standard deviation κ	performance metrics (max./av./min) PTP	MQD	ASF
CMPC	0	0.81/0.81/0.81	0.008/0.008/0.008	2.9/2.9/2.9 $\times 10^{-3}$
	0.1	0.81/0.81/0.81	0.010/0.009/0.008	3.2/2.9/2.4 $\times 10^{-3}$
	0.2	0.81/0.81/0.81	0.016/0.011/0.009	3.7/3.1/2.4 $\times 10^{-3}$
	0.3	0.89/0.81/0.81	0.021/0.014/0.010	4.4/3.5/2.7 $\times 10^{-3}$
	0.4	1.00/0.81/0.81	0.026/0.017/0.012	5.2/4.2/3.3 $\times 10^{-3}$
	0.5	1.10/0.83/0.81	0.031/0.020/0.013	6.2/5.0/3.9 $\times 10^{-3}$
DeMPC	0	0.97/0.97/0.97	0.028/0.028/0.028	8.1/8.1/8.1 $\times 10^{-3}$
	0.1	1.09/1.02/0.95	0.034/0.031/0.029	8.6/8.2/7.7 $\times 10^{-3}$
	0.2	1.18/1.06/0.97	0.039/0.035/0.032	9.1/8.3/7.8 $\times 10^{-3}$
	0.3	1.25/1.10/0.99	0.043/0.038/0.034	9.5/8.6/7.9 $\times 10^{-3}$
	0.4	1.30/1.12/1.00	0.046/0.041/0.036	10.0/9.0/8.1 $\times 10^{-3}$
	0.5	1.35/1.15/1.02	0.049/0.043/0.037	10.6/9.3/8.3 $\times 10^{-3}$

Table 4.2: *Performance metrics of the Monte-Carlo simulations using different standard deviations κ in the inaccurate forecast.*

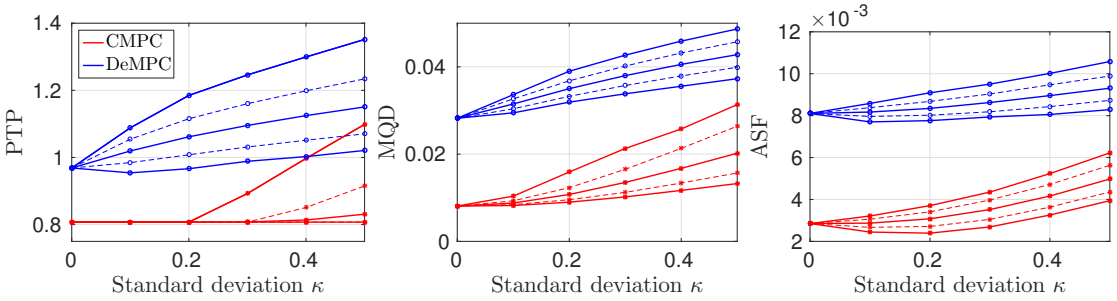

Figure 4.10: *Maximum, average and minimum performance metrics using a Monte-Carlo simulation with* 1000 *perturbed sequences. Between the dashed lines are* 90% *of the solutions.*

and in Figure 4.10 the performance results are summarized. Even for large disturbances the MPC algorithms provide good results.

4.6.5 The computational complexity

Whereas CMPC outperforms DeMPC in performance, DeMPC clearly outperforms CMPC in terms of the computational complexity. In Figure 4.11, the computational complexity of centralized and decentralized MPC depending on the number of RESs is visualized. The figure shows the average computational time in seconds of 100 minimization problems. The dashed lines show the maximal and the minimal computation times. The computational complexity of CMPC grows nonlinearly which makes the method intractable if the number of RESs becomes too big. On the other hand the computational complexity of the decent-

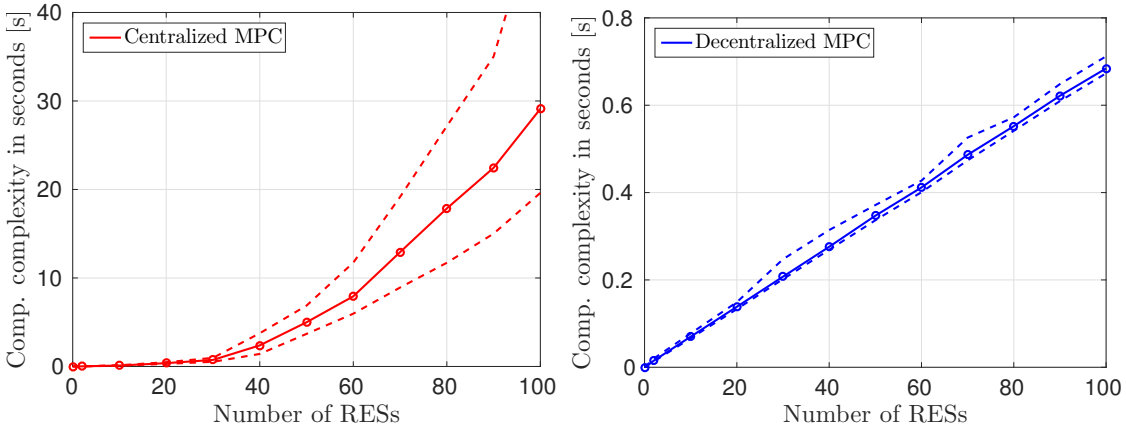

Figure 4.11: *Computational complexity of centralized and decentralized MPC. The lines show the average computation time of* 100 *OCPs and the dashed lines show the maximal and the minimal computation time.*

ralized MPC algorithm grows linearly with a moderate slope even though the optimization problems corresponding to the individual RESs were solved sequentially. If the minimization problems are solved in parallel by every RES individually, then the computational complexity is constant and hence the applicability of DeMPC is independent of the size of the network.

Chapter 5

A cooperative distributed optimization algorithm

In Chapter 4, we introduced a centralized and a decentralized MPC approach. CMPC outperforms DeMPC but is no longer suitable if the number of RESs is large, due to the computational complexity of the OCP

$$\mathbf{u}^\star(k;N) \in \operatorname{argmin}_{\mathbf{u}(k;N)\in\mathbb{U}^{k,N}(x_0)} J_N(x_0,\mathbf{u}(k;N)). \tag{5.1}$$

In this chapter, we fix the time instant $k \in \mathbb{N}$ and concentrate on the solution of a single OCP. We propose a hierarchical distributed optimization algorithm, which iteratively computes an optimal solution of the problem (5.1) while keeping the flexibility of the decentralized setting. Before we give the algorithm and a proof of convergence, we introduce necessary concepts for the optimization problem.

The well-known curse of dimensionality encountered when trying to solve large-scale optimization problems has led to several workarounds including various distributed optimization algorithms. In the control literature, this approach was initiated in the thesis of Tsitsiklis [99] (see also [15]), wherein decentralized optimization schemes with or without communication between agents or processors were studied. Recent works in the field of multi-agent systems, particularly in the field of consensus, have seen a resurgence of interest in the area of distributed optimization, see for example [35, 55, 58, 75, 76, 77, 114] and the references therein. Much of this work assumes the existence of a global cost function decomposable into the sum of cost functions for each agent. This allows individual agents to solve optimization problems locally and, under various communication schemes and topologies, to arrive at a globally optimal solution without the need to solve a (potentially large) optimization problem centrally.

A notable recent exception is [26], where the global cost functional is not separable, similar to the cost functionals considered in this thesis. In [26], it is assumed that each agent knows the global cost function, but only has access to its local decision variables and local constraint set. Furthermore, [26] assumes a global coupled inequality constraint where each agent knows its (functional) contribution to the coupling constraint. In this setting, [26] presents a distributed optimization algorithm based on a neighbor-to-neighbor

communication. More importantly, it is shown that this distributed optimization algorithm converges to the solution of the global optimization problem. In contrast to [26], the algorithm presented here relies on the presence of a CE all agents communicate with.
This chapter is structured as follows. In Section 5.1, we introduce basic notations and definitions necessary for the remainder of the chapter. In Section 5.2, a hierarchical distributed optimization algorithm is presented and convergence properties of the algorithm are shown. In Section 5.3, the application of the distributed optimization algorithm based on the linear model of the RESs introduced in Chapter 3 is discussed. In Section 5.4, the algorithm is extended to nonlinear model dynamics. The chapter is concluded with numerical simulations in Section 5.5 analyzing the performance of the proposed optimization algorithm in the context of MPC.

5.1 Assumptions and notations

To solve the OCP (5.1) in a distributed way, we exploit the convexity properties given in Section 2.4.1. Since we concentrate on a single OCP at a fixed time instant $k \in \mathbb{N}$ and a fixed prediction horizon $N \in \mathbb{N}$, we omit the index k and N in the notations whenever it does not lead to ambiguity. The OCP (5.1) is written in the unknowns $\mathbf{u} = \mathbf{u}(k;N)$ but the performance is measured in the variables $\mathbf{z}(\cdot)$. Now, let us rewrite the optimization problem in the variables $\mathbf{z} = \mathbf{z}(k;N)$ using the definition of the admissible set $\mathbb{D}^{k,N}(x_0)$ (cf. (2.4)). Similar to the notation used for the variables, we omit the initial value x_0 in the notation of the set $\mathbb{D}^{k,N}(x_0)$ in this chapter, i.e., we write $\mathbb{D} = \mathbb{D}^{k,N}(x_0)$ and $\mathbb{D}_i = \mathbb{D}_i^{k,N}(x_{i,0})$. Accordingly, the optimal control problem (5.1) is rewritten as

$$\begin{aligned} \min \quad & G(\mathbf{z}) \\ \text{s.t.} \quad & \mathbf{z} \in \mathbb{D} \end{aligned} \tag{5.2}$$

for a function $G : \mathbb{R}^{p\mathcal{I}\times N} \to \mathbb{R}$. For $i = 1, \dots, \mathcal{I}$, we define the mappings $\mathbf{z} \mapsto \mathbf{p}_i$ such that[1]

$$\mathbf{p}_i = \begin{pmatrix} \mathbf{z}_1^T & \dots & \mathbf{z}_{i-1}^T & \mathbf{z}_{i+1}^T & \dots & \mathbf{z}_{\mathcal{I}}^T \end{pmatrix}^T \in \mathbb{R}^{p(\mathcal{I}-1)\times N}. \tag{5.3}$$

We then define the local objective functions $g_i(\cdot;\mathbf{p}_i) : \mathbb{R}^{p\times N} \to \mathbb{R}$ as

$$g_i(\mathbf{z}_i;\mathbf{p}_i) := G(\mathbf{z}) \tag{5.4}$$

for all $i \in \mathbb{N}_{\mathcal{I}}$. In order to obtain a convex optimization problem (5.2), the function G has to be convex and the set $\mathbb{D}$ has to be convex. To ensure convexity of $\mathbb{D}$, we impose the following assumptions on the system dynamics.

Assumption 5.1.1. *We assume that the local systems dynamics* (2.1) *satisfy the following properties for all* $i \in \mathbb{N}_{\mathcal{I}}$*:*

- *The sets* $\mathbb{U}_i \in \mathbb{R}^{m_i}$ *are compact and convex and the sets* $\mathbb{X}_i \in \mathbb{R}^{n_i}$ *are convex for all* $i = 1, \dots, \mathcal{I}$.

[1] Alternatively the notation $\mathbf{z}_{-i} = \mathbf{p}_i$ is used in the literature for the definition in Equation (5.3) for all $i \in \mathbb{N}_{\mathcal{I}}$

- *The functions* $f_i : \mathbb{X}_i \times \mathbb{U}_i \to \mathbb{R}^{n_i}$ *are affine in* $x_i \in \mathbb{X}_i$ *and* $u_i \in \mathbb{U}_i$ *for all* $i = 1, \ldots, \mathcal{I}$, *i.e.,*

$$f_i(x_i, u_i) = A_i x_i + B_i u_i$$

for given matrices $A_i \in \mathbb{R}^{n_i \times n_i}$ *and* $B_i \in \mathbb{R}^{n_i \times m_i}$.

- *The functions* $h_i : \mathbb{X}_i \times \mathbb{U}_i \times \mathbb{R}^p \to \mathbb{R}^p$ *are affine in* $x_i \in \mathbb{X}_i$ *and* $u_i \in \mathbb{U}_i$ *for fixed* $s_i \in \mathbb{R}^p$ *for all* $i = 1, \ldots, \mathcal{I}$, *i.e.,*

$$h_i(x_i, u_i; s_i) = C_i x_i + D_i u_i + c_i(s_i)$$

for given matrices $C_i \in \mathbb{R}^{p \times n_i}$, $D_i \in \mathbb{R}^{p \times m_i}$ *and given functions* $c_i : \mathbb{R}^p \to \mathbb{R}^p$.

- *Additionally, we assume that the local systems satisfy a feasibility assumption. For all* $i \in \mathbb{N}_{\mathcal{I}}$, *for all prediction horizons* $N \in \mathbb{N}$, *for all time steps* $k \in \mathbb{N}$ *and for all initial values* $x_i(k) \in \mathbb{X}_i$, *we assume that there exists* $\mathbf{u}_i(k; N) \in \mathbb{U}_i^{k;N}(x_i(k))$ *such that* $\mathbf{x}_i(k; N) \in \mathbb{X}_i^N$ *holds.*

Lemma 5.1.2. *If Assumption 5.1.1 holds, then the sets* $\mathbb{D}_i$ *are convex and compact for all* $i \in \mathbb{N}_{\mathcal{I}}$.

Proof. For a fixed initial value $x_{i,0} = x_i(k) \in \mathbb{X}_i$, $i \in \mathbb{N}_{\mathcal{I}}$, the sets

$$\overline{\mathbb{X}}_{i_{x_0}}^N = \left\{ \mathbf{x}_i \in \mathbb{X}_i^N \;\middle|\; \begin{array}{l} x_i(j+1) = f_i(x_i(j), u_i(j)) \\ x_i(j+1) \in \mathbb{X}_i, \quad u_i(j) \in \mathbb{U}_i(x_i(j)) \\ \quad \forall\, j = k, \ldots, k+N-1 \end{array} \right\} \tag{5.5}$$

are convex and compact since f_i is affine (and continuous), $\mathbb{U}_i$ is convex and compact and $\mathbb{X}_i$ is convex. This implies that $\mathbb{D}_i$ is compact due to the continuity of h_i. The convexity follows immediately from the convexity and compactness of $\mathbb{U}_i$ and $\overline{\mathbb{X}}_{i_{x_0}}^N$ and the linearity of f_i and h_i for all $i \in \mathbb{N}_{\mathcal{I}}$. □

5.2 The hierarchical distributed optimization algorithm

5.2.1 The distributed optimization algorithm

In this section, we propose a hierarchical distributed optimization algorithm (see Algorithm 5) to solve optimization problems of the form (5.2). Instead of solving one minimization problem, several iterations are performed at every time step k in which every system $i \in \mathbb{N}_{\mathcal{I}}$ minimizes the local objective function $g_i(\cdot; \mathbf{p}_i)$ for changing parameters $\mathbf{p}_i$. The CE broadcasts the communication variables and computes the variable stepsize θ in every iteration. The hierarchical distributed optimization algorithm splits the optimization problem into local optimization problems, where the number of unknowns is independent of the number of systems in the network, and an optimization problem for the CE. The optimization problem of the CE has only one unknown, namely the stepsize. Observe that only the communication variables z_i are broadcast among the systems and only subsystem $i \in \mathbb{N}_{\mathcal{I}}$ makes use of the set $\mathbb{D}_i$. In other words, no knowledge of the system dynamics of the other subsystems is necessary to compute a solution.

Algorithm 5 Hierarchical cooperative distributed optimization algorithm

Input:

- **Subsystem** i, $i \in \mathbb{N}_{\mathcal{I}}$: Define the admissible set $\mathbb{D}_i$ based on the initial state $x_i(k) \in \mathbb{X}_i$ and the time-dependent quantity $\mathbf{s}_i(k; N)$.
- **CE**: Define the number of subsystems $\mathcal{I}$, the prediction horizon N, the maximal iteration number $\ell_{\max} \in \mathbb{N} \cup \{\infty\}$ and a desired precision $\varepsilon \in \mathbb{R}_{\geq 0}$.

Initialization:

- **Subsystem** i, $i \in \mathbb{N}_{\mathcal{I}}$: Define and transmit $\mathbf{z}_i^{\star_1}, \mathbf{z}_i^1 \in \mathbb{D}_i$.
- **CE**: Set the iteration counter $\ell = 1$ and $G^1 = \infty$, receive $\mathbf{z}_i^1$, $i \in \mathbb{N}_{\mathcal{I}}$.

Main loop:

Phase 1 (CE): Receive, $\mathbf{z}_i^{\star_\ell}$ $i = 1, 2, \ldots, \mathcal{I}$.

- Compute the stepsize $\theta^\ell = \underset{\theta \in [0,1]}{\operatorname{argmin}} \; G(\theta \mathbf{z}^{\star_\ell} + (1-\theta)\mathbf{z}^\ell)$.
- Compute $\mathbf{z}^{\ell+1} := \theta^\ell \mathbf{z}^{\star_\ell} + (1-\theta^\ell)\mathbf{z}^\ell$ and evaluate the performance index

$$G^{\ell+1} := G(\mathbf{z}^{\ell+1}). \tag{5.6}$$

- If $|G^{\ell+1} - G^\ell| < \varepsilon$ or $\ell \geq \ell_{\max}$ holds, terminate the main loop. Otherwise, transmit $\mathbf{z}^{\ell+1}$ to the subsystems.

Phase 2 (Subsystem i, $i \in \mathbb{N}_{\mathcal{I}}$**)**: Receive $\mathbf{z}^{\ell+1}$.

- Define $\mathbf{p}_i^{\ell+1}$ (see Equation (5.3))
- Solve the local minimization problem

$$\mathbf{z}_i^{\star_{\ell+1}} = \underset{\mathbf{z}_i \in \mathbb{D}_i}{\operatorname{argmin}} \; g_i(\mathbf{z}_i; \mathbf{p}_i^{\ell+1}). \tag{5.7}$$

- Transmit $\mathbf{z}_i^{\star_{\ell+1}}$.

Increment the iteration counter $\ell = \ell + 1$ and repeat the loop.

Remark 5.2.1. *In Algorithm 5 the communication of the variable $\mathbf{z} \in \mathbb{R}^{p\mathcal{I} \times N}$ from the CE to all the subsystems $i \in \mathbb{N}_{\mathcal{I}}$ is necessary. For special cases of the algorithm we show how the amount of data communicated by the CE can be reduced significantly (see Algorithm 6).*

5.2.2 Convergence of the distributed optimization algorithm

In this section, we prove convergence of Algorithm 5 to the optimal solution of the minimization problem (5.2). To this end, we first formalize the crucial steps and the involved functions of the algorithm. Throughout this section, we assume that $\mathbb{D}_i \subset \mathbb{R}^p$ is convex, compact and non-empty for all $i \in \mathbb{N}_{\mathcal{I}}$. The set $\mathbb{D} \in \mathbb{R}^{p\mathcal{I}\times N}$ is defined as $\mathbb{D} = \mathbb{D}_1 \times \ldots \times \mathbb{D}_{\mathcal{I}}$ and thus, is also convex, compact and non-empty.
Let the global objective function $G : \mathbb{D} \to \mathbb{R}$ be continuous and convex on $\mathbb{D}$ and let the local functions $g_i(\cdot; \mathbf{p}_i) : \mathbb{D}_i \to \mathbb{R}$ be strictly convex for fixed parameters $\mathbf{p}_i$ for all $i \in \mathbb{N}_{\mathcal{I}}$. Then, the local minimizer

$$\mathbf{z}_i^\star := \underset{\mathbf{z}_i \in \mathbb{D}_i}{\operatorname{argmin}}\ g_i(\mathbf{z}_i; \mathbf{p}_i)$$

is unique according to Theorem 2.4.9. Due to the continuity of G, g_i is continuous in $\mathbf{p}_i$. Since $g_i(\cdot, \mathbf{p}_i)$ is additionally strictly convex the uniqueness of the minimizer $\mathbf{z}_i^\star$ implies that the mapping

$$\mathbf{p}_i \mapsto \mathbf{z}_i^\star \tag{5.8}$$

from the parameters to the minimizers is continuous for all $i \in \mathbb{N}_{\mathcal{I}}$. (A corresponding result in a more general setting can be found in [11, Ch. 1], for example.) As a first step of proving convergence, we show that the sequence $(G^\ell)_{\ell\in\mathbb{N}}$ computed in (5.6) is non-increasing.

Lemma 5.2.2. *Let $G : \mathbb{D} \to \mathbb{R}$ be continuous and convex and let the local functions $g_i(\cdot; \mathbf{p}_i) : \mathbb{D}_i \to \mathbb{R}$ be strictly convex for all parameters $\mathbf{p}_i$ and for all $i \in \mathbb{N}_{\mathcal{I}}$. Then the sequence $(G^\ell)_{\ell\in\mathbb{N}}$ generated by Algorithm 5 is non-increasing, i.e., $G^{\ell+1} \leq G^\ell$ holds for all $\ell \in \mathbb{N}$. If, additionally, $\mathbf{z}^{\star\ell} \neq \mathbf{z}^\ell$, then $G^{\ell+1} < G^\ell$ holds. Hence, the sequence $(G^\ell)_{\ell\in\mathbb{N}}$ is monotonically decreasing until Algorithm 5 stops.*

Proof. Since $\mathbf{z}_i^{\star\ell}$ is the global minimum of $g_i(\cdot; \mathbf{p}_i^\ell)$, and since $g_i(\cdot; \mathbf{p}_i^\ell)$ is strictly convex, it holds that

$$\begin{aligned} g_i(\mathbf{z}_i^{\star\ell}; \mathbf{p}_i^\ell) &\leq g_i\left(\frac{1}{\mathcal{I}}\mathbf{z}_i^{\star\ell} + \left(1 - \frac{1}{\mathcal{I}}\right)\mathbf{z}_i^\ell; \mathbf{p}_i^\ell\right) \\ &\leq g_i(\mathbf{z}_i^\ell; \mathbf{p}_i^\ell) \\ &= G(\mathbf{z}^\ell) \end{aligned}$$

for all $i \in \mathbb{N}_{\mathcal{I}}$. For $i \in \mathbb{N}_{\mathcal{I}}$, we define the vectors

$$\Delta\mathbf{z}_i = \begin{pmatrix} 0^T & \ldots & 0^T & \left(\mathbf{z}_i^{\star\ell} - \mathbf{z}_i^\ell\right)^T & 0^T & \ldots & 0^T \end{pmatrix}^T. \tag{5.9}$$

With this definition, $G^{\ell+1}$ can be written in the form

$$\begin{aligned} G(\mathbf{z}^{\ell+1}) &= G\left(\mathbf{z}^\ell + \theta^\ell\left(\mathbf{z}^{\star\ell} - \mathbf{z}^\ell\right)\right) \\ &= G\left(\mathbf{z}^\ell + \theta^\ell \sum_{i=1}^{\mathcal{I}} \Delta\mathbf{z}_i\right). \end{aligned}$$

Since $\theta^\ell \in [0,1]$ is chosen optimal with respect to $\mathbf{z}^{\star\ell}$ and $\mathbf{z}^\ell$, we obtain the estimate

$$G\left(\mathbf{z}^\ell + \theta^\ell \sum_{i=1}^{\mathcal{I}} \Delta\mathbf{z}_i\right) \leq G\left(\mathbf{z}^\ell + \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \Delta\mathbf{z}_i\right) = G\left(\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \left(\mathbf{z}^\ell + \Delta\mathbf{z}_i\right)\right). \tag{5.10}$$

Applying Jensens inequality to the convex function G permits to extract the sum out of the function yielding the estimate

$$G\left(\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \left(\mathbf{z}^\ell + \Delta\mathbf{z}_i\right)\right) \leq \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} G\left(\mathbf{z}^\ell + \Delta\mathbf{z}_i\right) \tag{5.11}$$

and allowing to identify the local objective functions

$$G\left(\mathbf{z}^\ell + \Delta\mathbf{z}_i\right) = g_i\left(\mathbf{z}_i^{\star\ell}; \mathbf{p}_i^\ell\right)$$

for all $i \in \mathbb{N}_{\mathcal{I}}$. Since $\mathbf{z}^{\star\ell}$ is the solution of the local optimization, the inequality

$$g_i\left(\mathbf{z}_i^{\star\ell}; \mathbf{p}_i^\ell\right) \leq g_i\left(\mathbf{z}_i^\ell; \mathbf{p}_i^\ell\right)$$

holds for all $i \in \mathbb{N}_{\mathcal{I}}$. This leads to the result

$$\begin{aligned}\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} G\left(\mathbf{z}^\ell + \Delta\mathbf{z}_i\right) &= \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} g_i\left(\mathbf{z}_i^{\star\ell}; \mathbf{p}_i^\ell\right) \\ &\leq \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} g_i\left(\mathbf{z}_i^\ell; \mathbf{p}_i^\ell\right) = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} G\left(\mathbf{z}^\ell\right).\end{aligned}$$

which implies the assertion

$$G^{\ell+1} = G(\mathbf{z}^{\ell+1}) \leq G(\mathbf{z}^\ell) = G^\ell.$$

In the case $\mathbf{z}_i^{\star\ell} \neq \mathbf{z}_i^\ell$ for at least one index $i \in \mathbb{N}_{\mathcal{I}}$ we obtain

$$g_i\left(\mathbf{z}_i^{\star\ell}; \mathbf{p}_i^\ell\right) < g_i\left(\mathbf{z}_i^\ell; \mathbf{p}_i^\ell\right)$$

due to the strict convexity of $g_i\left(\cdot; \mathbf{p}_i^\ell\right)$ which implies $G^{\ell+1} < G^\ell$. □

The proof of Lemma 5.2.2 shows that the constant stepsize $\theta^\ell = \frac{1}{\mathcal{I}}$ also leads to a non-increasing sequence providing an alternative way to update $\mathbf{z}^\ell$. This immediately follows from the estimate (5.10) and from Inequality (5.11). The result is briefly summarized in the following corollary.

Corollary 5.2.3. *Let $G : \mathbb{D} \to \mathbb{R}$ be continuous and convex and let the local functions $g_i(\cdot; \mathbf{p}_i) : \mathbb{D}_i \to \mathbb{R}$ be strictly convex for all parameters $\mathbf{p}_i$ and for all $i \in \mathbb{N}_{\mathcal{I}}$. Then the sequence $(G^\ell)_{\ell\in\mathbb{N}}$ generated by Algorithm 5 using the constant stepsize $\theta^\ell = \frac{1}{\mathcal{I}}$ is non-increasing. If, additionally, $\mathbf{z}^{\star\ell} \neq \mathbf{z}^\ell$, then $G^{\ell+1} < G^\ell$ holds. Hence, the sequence $(G^\ell)_{\ell\in\mathbb{N}}$ is monotonically decreasing until Algorithm 5 stops.*

Since the sequence $(G^\ell)_{\ell\in\mathbb{N}} \subset \mathbb{R}$ is non-increasing, it is straightforward to show that the sequence is convergent.

Corollary 5.2.4. *Let $G : \mathbb{D} \to \mathbb{R}$ be continuous and convex and let the local functions $g_i(\cdot;\mathbf{p}_i) : \mathbb{D}_i \to \mathbb{R}$ be strict convex for all parameters $\mathbf{p}_i$ and for all $i \in \mathbb{N}_\mathcal{I}$. Then, the sequence $(G^\ell)_{\ell\in\mathbb{N}} \subset \mathbb{R}$ of Algorithm 5 converges as $\ell \to \infty$, i.e., there exists an $G^\sharp \in \mathbb{R}$ such that $\lim_{\ell\to\infty} G^\ell = G^\sharp$ holds.*

Proof. The function G is continuous, convex and defined on a compact set. This implies that the minimum $G^\star = \min_{\mathbf{z}\in\mathbb{D}} G(\mathbf{z})$ is attained (Theorem 2.4.9). Since $(G^\ell)_{\ell\in\mathbb{N}}$ is monotonically decreasing by Lemma 5.2.2 and bounded from below by $G^\star$, $(G^\ell)_{\ell\in\mathbb{N}}$ converges, i.e., $\lim_{\ell\to\infty} G^\ell = G^\sharp$. □

In Lemma 5.2.2 and Corollary 5.2.4 we have shown that the sequence $(G^\ell)_{\ell\in\mathbb{N}}$ is converging. Our remaining task, which is the main result of this section, is to demonstrate that the limit of the sequence $(G^\ell)_{\ell\in\mathbb{N}}$ is equal to the solution of (5.2) if we additionally assume that G is differentiable.

Theorem 5.2.5. *Let $G : \mathbb{D} \to \mathbb{R}$ be continuously differentiable and convex and let the local functions $g_i(\cdot;\mathbf{p}_i) : \mathbb{D}_i \to \mathbb{R}$ be strictly convex for all parameters $\mathbf{p}_i$ and for all $i \in \mathbb{N}_\mathcal{I}$. The limit $G^\sharp$ of the sequence $(G^\ell)_{\ell\in\mathbb{N}}$ generated by Algorithm 5 coincides with the unique solution $G^\star = \min_{\mathbf{z}\in\mathbb{D}} G(\mathbf{z})$ of the OCP (5.2).*

Proof. Let $\mathbf{z}^\star$ denote a solution of (5.2), i.e., $G^\star = G(\mathbf{z}^\star)$. For any $\tilde{\mathbf{z}}$ with

$$G(\tilde{\mathbf{z}}) > G^\star \tag{5.12}$$

one step of Algorithm 5 with $\mathbf{z}_i^\ell = \tilde{\mathbf{z}}_i$ for $i = 1, \ldots, \mathcal{I}$ yields

$$G(\mathbf{z}^{\ell+1}) < G(\tilde{\mathbf{z}}) \tag{5.13}$$

or

$$G(\mathbf{z}^{\ell+1}) = G(\tilde{\mathbf{z}}) \tag{5.14}$$

due to Lemma 5.2.2. If Equation (5.14) holds, we additionally obtain from Lemma 5.2.2 that $\mathbf{z}_i^{\ell+1} = \tilde{\mathbf{z}}_i$ for all $i \in \mathbb{N}_\mathcal{I}$, i.e., Algorithm 5 is stationary. To show that this case cannot happen while (5.12) simultaneously holds, we define the function $\phi : [0,1]^\mathcal{I} \to \mathbb{R}$,

$$\phi(\eta) := G\left(\tilde{\mathbf{z}} + \sum_{i=1}^{\mathcal{I}} \eta_i \Delta\mathbf{z}_i\right),$$

using the definition

$$\Delta\mathbf{z}_i = \begin{pmatrix} 0^T & \ldots & 0^T & (\mathbf{z}_i^\star - \tilde{\mathbf{z}}_i)^T & 0^T & \ldots & 0^T \end{pmatrix}^T$$

for all $i \in \mathbb{N}_{\mathcal{I}}$, similar to Equation (5.9). With the notation $\mathbb{1} = (1, \ldots, 1) \in \mathbb{R}^{\mathcal{I}}$, it holds that

$$\phi(\mathbb{1}) = G^\star < G(\tilde{\mathbf{z}}) = \phi(0). \tag{5.15}$$

Since G is convex, ϕ is convex and the directional derivative of ϕ in $0 \in \mathbb{R}^{\mathcal{I}}$ with respect to $\eta = \mathbb{1}$ is less than zero, i.e.,

$$0 > \langle \operatorname{grad} \phi(0), \mathbb{1} \rangle = \sum_{i=1}^{\mathcal{I}} \frac{\partial \phi}{\partial \eta_i}(0). \tag{5.16}$$

Inequality (5.15) implies the existence of an index $i \in \mathbb{N}_{\mathcal{I}}$ such that $\mathbf{z}_i^\star \neq \tilde{\mathbf{z}}_i$ and, thus, $0 > \frac{\partial \phi}{\partial \eta_i}(0)$ holds. However, the i-th system updates in this case $\tilde{\mathbf{z}}_i$, which contradicts the assumption $\mathbf{z}_i^{\ell+1} = \tilde{\mathbf{z}}_i$. Hence, Inequality (5.13) holds for all $\tilde{\mathbf{z}}_i \in \mathbb{D}_i$, $i = 1, \ldots, \mathcal{I}$, satisfying Inequality (5.12).
The function G is continuous and defined on a compact set. Therefore, there exists an (admissible) accumulation point $\mathbf{z}^\sharp$ of the sequence $(\mathbf{z}^\ell)_{\ell \in \mathbb{N}}$ satisfying the equality

$$G(\mathbf{z}^\sharp) = G^\sharp.$$

Now, it is clear that $G^\sharp \geq G^\star$. To show that $G^\sharp = G^\star$, assume to the contrary that $G^\sharp > G^\star$. Since the solutions of the local optimization problems depend continuously on the parameters $\mathbf{p}_i$, as per (5.8), and the function G is continuous, the descent property (5.13) at the accumulation point $\mathbf{z}^\sharp$ implies the existence of an $\varepsilon > 0$ such that the inequality

$$G(\mathbf{z}^{\ell+1}) < G^\sharp \tag{5.17}$$

is satisfied whenever $\mathbf{z}^\ell \in B_\varepsilon(\mathbf{z}^\sharp)$ holds.[2] Since $\mathbf{z}^\sharp$ is an accumulation point, there exists an index $\ell \in \mathbb{N}$ such that $\mathbf{z}^\ell \in B_\varepsilon(\mathbf{z}^\sharp)$ holds and, thus, Inequality (5.17) holds. However, due to the monotonicity of the sequence $(G^\ell)_{\ell \in \mathbb{N}}$ (Lemma 5.2.2), this contradicts the definition of $G^\sharp$. Therefore, the assertion $G^\sharp = G^\star$ holds. □

Theorem 5.2.5 is the main result of this section. Since $G^\sharp$ coincides with $G^\star$, the performance of CMPC and cooperative distributed model predictive control (DiMPC) are equal if Algorithm 5 is used in the distributed control setting.
In the remainder of this section, we consider a specific form of the objective function. We assume that there exists a function $\overline{G} : \widehat{\mathbb{D}} \to \mathbb{R}$ such that

$$G(\mathbf{z}) = \overline{G}\left(\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i \right)$$

for all $\mathbf{z} \in \mathbb{D}$ and $\widehat{\mathbb{D}} \subset \mathbb{R}^{p \times N}$ is defined as

$$\widehat{\mathbb{D}} := \left\{ \hat{\mathbf{z}} \in \mathbb{R}^{p \times N} \,\middle|\, \hat{\mathbf{z}} = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i,\ \mathbf{z}_i \in \mathbb{D}_i \ \forall\ i \in \mathbb{N}_{\mathcal{I}} \right\}.$$

[2] $B_\varepsilon(\mathbf{z}^\sharp)$ represents the open ball of radius $\varepsilon > 0$ centered at $\mathbf{z}^\sharp$.

Since $\mathbb{D}_i$ is convex and compact for all $i \in \mathbb{N}_{\mathcal{I}}$ the set $\widehat{\mathbb{D}}$ is also convex and compact. If the function G is convex and the local functions $g_i(\cdot; \mathbf{p}_i)$ are strictly convex for all $i \in \mathbb{N}_{\mathcal{I}}$, then also $\overline{G}$ is strictly convex in $\hat{\mathbf{z}}$.

Example 5.2.6. *The centralized cost functional* (4.4) *penalizing the deviation from a given reference* $\hat{\zeta} \in \mathbb{R}$ *can be written in the form*

$$G(\mathbf{z}) = \left\| \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z} - \mathbb{1}\hat{\zeta} \right\|^2 \tag{5.18}$$

or in the average variables

$$\overline{G}(\hat{\mathbf{z}}) = \left\| \hat{\mathbf{z}} - \mathbb{1}\hat{\zeta} \right\|^2. \tag{5.19}$$

The function G is convex but not strictly convex for $\mathcal{I} > 1$ whereas $\overline{G}$ is strictly convex in $\hat{\mathbf{z}}$.

Due to the strict convexity of $\overline{G}$ and the convexity and compactness of the set $\widehat{\mathbb{D}}$, the global minimizer $\hat{\mathbf{z}}^\star \in \mathbb{D}$ and the global minimum $\overline{G}^\star$ are unique (see Theorem 2.4.9). This result is summarized in the following corollary.

Corollary 5.2.7. *Let $\overline{G} : \widehat{\mathbb{D}} \to \mathbb{R}$ be continuously differentiable and strictly convex and let the local functions $g_i(\cdot; \mathbf{p}_i) : \mathbb{D}_i \to \mathbb{R}$ be strictly convex for all parameters $\mathbf{p}_i$ and for all $i \in \mathbb{N}_{\mathcal{I}}$. Then Algorithm 5 applied to the objective function $\overline{G}$ satisfies the following properties.*

- *The minimum*

  $$G^\star = \overline{G}^\star = \min_{\hat{\mathbf{z}} \in \widehat{\mathbb{D}}} \overline{G}\left(\hat{\mathbf{z}}\right)$$

  and the minimizer

  $$\hat{\mathbf{z}}^\star = \underset{\hat{\mathbf{z}} \in \widehat{\mathbb{D}}}{argmin}\ \overline{G}\left(\hat{\mathbf{z}}\right)$$

  are unique.

- *For the sequences $(\mathbf{z}_i^\ell)_{\ell \in \mathbb{N}}$, $i = 1, \ldots, \mathcal{I}$, the convergence*

  $$\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i^\ell = \hat{\mathbf{z}}^\ell \to \hat{\mathbf{z}}^\star$$

  for $\ell \to \infty$ holds.

So far, we have shown that $(G^\ell)_{\ell \in \mathbb{N}}$, $(\overline{G}^\ell)_{\ell \in \mathbb{N}}$ and $(\hat{\mathbf{z}}^\ell)_{\ell \in \mathbb{N}}$ converge. As a final step, it would be desirable if it could additionally be shown that $(\mathbf{z}_i^\ell)_{\ell \in \mathbb{N}}$ converges for all $i \in \mathbb{N}_{\mathcal{I}}$. Unfortunately, we can only show that the difference between two consecutive solutions, $\mathbf{z}_i^\ell$ and $\mathbf{z}_i^{\ell-1}$, converges to zero for $\ell \to \infty$.

Theorem 5.2.8. *Let $\overline{G} : \widehat{\mathbb{D}} \to \mathbb{R}$ be continuously differentiable and strictly convex and let the local functions $g_i(\cdot; \mathbf{p}_i) : \mathbb{D}_i \to \mathbb{R}$ be strictly convex for all parameters $\mathbf{p}_i$ and for all $i \in \mathbb{N}_{\mathcal{I}}$. Let $(\mathbf{z}_i^\ell)_{\ell \in \mathbb{N}}$, $i \in \mathbb{N}_{\mathcal{I}}$, be the sequence generated by Algorithm 5. Then, the stepsize $\|\Delta \mathbf{z}_i^\ell\|$ with $\Delta \mathbf{z}_i^\ell := \mathbf{z}_i^\ell - \mathbf{z}_i^{\ell-1}$ converges to zero for $\ell \to \infty$.*

Proof. Let $\varepsilon > 0$. We define the set

$$N_\varepsilon = \left\{ \hat{\mathbf{z}} \in \widehat{\mathbb{D}} \,\middle|\, \overline{G}(\hat{\mathbf{z}}) \leq \overline{G}(\hat{\mathbf{z}}^\star) + \varepsilon \right\}.$$

Let $\delta > 0$. Then there exists an $\varepsilon > 0$ such that

$$N_\varepsilon \subset B_\delta(\hat{\mathbf{z}}^\star).$$

due to the strict convexity of $\overline{G}$ and the compactness of $\mathbb{D}_i$, $i \in \mathbb{N}_{\mathcal{I}}$. From Corollary 5.2.7, we obtain the convergence $\hat{\mathbf{z}}^\ell \to \hat{\mathbf{z}}^\star$ for $\ell \to \infty$. Let $\delta > 0$ be arbitrary. Then there exists an $\ell \in \mathbb{N}$ and an $\varepsilon > 0$ such that $\hat{\mathbf{z}}^\ell \in B_\delta(\hat{\mathbf{z}}^\star)$, or equivalently $\hat{\mathbf{z}}^\ell - \hat{\mathbf{z}}^\star \in B_\delta(0)$, and $\hat{\mathbf{z}}^\ell \in N_\varepsilon$. For the local objective function, we obtain the estimate

$$g_i(\mathbf{z}_i^{\star\ell}; \mathbf{p}_i^\ell) \leq g_i(\mathbf{z}_i^\ell; \mathbf{p}_i^\ell) = \overline{G}(\hat{\mathbf{z}}^\ell) \leq \overline{G}(\hat{\mathbf{z}}^\star) + \varepsilon$$

which implies

$$\hat{\mathbf{z}}^\ell - \frac{1}{\mathcal{I}} \mathbf{z}_i^\ell + \frac{1}{\mathcal{I}} \mathbf{z}_i^{\star\ell} \in N_\varepsilon \subset B_\delta(\hat{\mathbf{z}}^\star).$$

Therefore, we can conclude that for all sufficiently large ℓ

$$\left(\hat{\mathbf{z}}^\star - \hat{\mathbf{z}}^\ell + \frac{1}{\mathcal{I}} \mathbf{z}_i^\ell - \frac{1}{\mathcal{I}} \mathbf{z}_i^{\star\ell} \right) - \left(\hat{\mathbf{z}}^\star - \hat{\mathbf{z}}^\ell \right) = \frac{1}{\mathcal{I}} \left(\mathbf{z}_i^\ell - \mathbf{z}_i^{\star\ell} \right) \in B_{2\delta}(0),$$

i.e., $\|\mathbf{z}_i^\ell - \mathbf{z}_i^{\star\ell}\| \to 0$ for $\ell \to \infty$. Finally, we obtain $\|\mathbf{z}_i^{\ell+1} - \mathbf{z}_i^\ell\| \to 0$ for $\ell \to \infty$ using the definition of $\mathbf{z}_i^{\ell+1}$

$$\mathbf{z}_i^{\ell+1} - \mathbf{z}_i^\ell = (\theta^\ell \mathbf{z}_i^{\star\ell} + (1 - \theta^\ell) \mathbf{z}_i^\ell) - \mathbf{z}_i^\ell = \theta^\ell (\mathbf{z}_i^{\star\ell} - \mathbf{z}_i^\ell)$$

for $\theta^\ell \in [0, 1]$. □

5.3 Application to residential energy systems

In Section 4.3, the advantages and disadvantages of a centralized control approach in the context of smart grids were discussed. In the previous section, we presented a hierarchical distributed optimization algorithm which splits the optimization problem into local tasks performed by subsystems and a global task performed by the CE. In the limit, Algorithm 5 recovers the optimal solution of the centralized control approach. In this section, we investigate advantages of the distributed control algorithm over the centralized approach in the context of the electricity grid introduced in Chapter 3.

5.3.1 The communication structure of the distributed optimization algorithm

Algorithm 5 consists of three main steps which we investigate in the following. The computation of an optimal solution of the local problems (5.7) by subsystems or the RESs, a solution of the optimization problem of the CE to obtain the stepsize θ^ℓ and the communication between the RESs and the CE.

General properties using the objective function G

The optimization problems of the RESs only depend on the parameters $\mathbf{p}_i$ and the local system dynamics (2.1) of the individual RESs, i.e., the system dynamics of RES i, which define the set $\mathbb{D}_i$, is private and does not need to be known by the other RESs and the CE. For this reason, similar to the decentralized setting, the local system dynamics can be changed without changing any component other than the local controller of the corresponding RES.
The global optimization problem in Phase 1 is an optimization problem in one single variable $\theta \in [0, 1]$ and hence, can be solved efficiently, sometimes even explicitly, independent of the size of the overall network. Moreover, the CE only requires the variables z to compute the next iterate. The variables u_i and x_i remain private for all $i \in \mathbb{N}_\mathcal{I}$.
The number of variables that have to be transmitted grows linearly with the number of RESs or the dimension of $\mathbf{z}$. Moreover, the communication variables do not remain private between the RESs, since every system needs to know $\mathbf{z}^\ell$ to define $\mathbf{p}_i^\ell$. This might prevent customers to join a network using Algorithm 5. Through the objective function $\overline{G}$, these problems can be circumvented.

Properties using the objective functions $\overline{G}$

For objective functions of the form $G(\mathbf{z}) = \overline{G}(\hat{\mathbf{z}})$, the number of transmitted variables can be made independent of the number of RESs in the network. More precisely, it is sufficient that every RES sends pN values to the CE and the CE publishes $Np+1$ values. Moreover, the CE does not have to send the $Np+1$ values to the RESs individually but only has to make sure that $Np+1$ values are publicly available and can be accessed by every RES.
The reduction to $Np+1$ values works as follows: In the case of the function $\overline{G}$, instead of the parameters $\mathbf{p}_i \in \mathbb{R}^{(\mathcal{I}-1)p\times N}$, $i = 1, \dots, \mathcal{I}$, we can define the average parameters

$$\hat{\mathbf{p}}_i = \frac{1}{\mathcal{I}} \sum_{l=1}^{\mathcal{I}} \mathbf{z}_l - \frac{1}{\mathcal{I}} \mathbf{z}_i = \hat{\mathbf{z}} - \frac{1}{\mathcal{I}} \mathbf{z}_i \tag{5.20}$$

for $i = 1, \dots, \mathcal{I}$, and write the local objective functions with parameters $\hat{\mathbf{p}}_i$, i.e., $g_i(\cdot; \hat{\mathbf{p}}_i)$. As a consequence, the dimension of the parameters $\hat{\mathbf{p}}_i \in \mathbb{R}^{p\times N}$ is independent of $\mathcal{I}$. To avoid the communication with every RES in iteration ℓ of Algorithm 5, observe that $\hat{\mathbf{p}}_i^{\ell+1} = \hat{\mathbf{z}}^{\ell+1} - \frac{1}{\mathcal{I}} \mathbf{z}_i^{\ell+1}$, i.e., the individual parameter $\mathbf{p}_i^{\ell+1}$ is obtained from the general information $\hat{\mathbf{z}}^{\ell+1}$ plus a local information. Even though only $\mathbf{z}_i^{\star\ell}$ is known to RES i, it holds that $\mathbf{z}_i^{\ell+1} := \theta^\ell \mathbf{z}_i^{\star\ell} + (1-\theta^\ell)\mathbf{z}_i^\ell$ and hence, $\mathbf{z}_i^{\ell+1}$ can be computed easily if the stepsize θ^ℓ is

known in every iteration $\ell \in \mathbb{N}$. By publishing $\hat{\mathbf{z}}^{\ell+1}$ and θ^ℓ in every iteration ℓ, every RES is able to compute $\hat{\mathbf{p}}_i^{\ell+1}$ independent of the size of the network. Algorithm 5 is rewritten for the special case of $G(\mathbf{z}) = \hat{G}(\hat{\mathbf{z}})$ in Algorithm 6. In this case, the variables $\mathbf{z}_i^\ell$ are only

Algorithm 6 Hierarchical cooperative distributed optimization for a network of RESs

Input:

- **RES** i, $i \in \mathbb{N}_\mathcal{I}$: Define the admissible set $\mathbb{D}_i$ based on the initial state $x_i(k) \in \mathbb{X}_i$ and the time-dependent quantity $\mathbf{s}_i(k; N)$.
- **CE**: A continuously differentiable and strictly convex function $\overline{G}$, number of RESs $\mathcal{I}$, prediction horizon N, maximal iteration number $\ell_{\max} \in \mathbb{N} \cup \{\infty\}$, desired precision $\varepsilon \in \mathbb{R}_{\geq 0}$.

Initialization:

- **RES** i, $i \in \mathbb{N}_\mathcal{I}$: Define and transmit $\mathbf{z}_i^{\star 1}, \mathbf{z}_i^1 \in \mathbb{D}_i$.
- **CE**: Set the iteration counter $\ell = 1$ and $G^1 = \infty$, receive $\mathbf{z}_i^1$, $i \in \mathbb{N}_\mathcal{I}$ and compute $\hat{\mathbf{z}}^1 = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i^1$.

Main loop:
Phase 1 (CE): Receive $\mathbf{z}_i^{\star\ell}$ for $i \in \mathbb{N}_\mathcal{I}$.

- Compute $\hat{\mathbf{z}}^{\star\ell} = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i^{\star\ell}$.
- Compute the stepsize $\theta^\ell = \underset{\theta \in [0,1]}{\operatorname{argmin}} \overline{G}(\theta \hat{\mathbf{z}}^{\star\ell} + (1-\theta)\hat{\mathbf{z}}^\ell)$.
- Compute $\hat{\mathbf{z}}^{\ell+1} := \theta^\ell \hat{\mathbf{z}}^{\star\ell} + (1-\theta^\ell)\hat{\mathbf{z}}^\ell$ and evaluate the performance index

$$G^{\ell+1} := \overline{G}(\hat{\mathbf{z}}^{\ell+1}). \tag{5.21}$$

- If $|G^{\ell+1} - G^\ell| < \varepsilon$ or $\ell \geq \ell_{\max}$ holds, terminate the algorithm. Otherwise, transmit $\hat{\mathbf{z}}^{\ell+1}$ and θ^ℓ to the RESs.

Phase 2 (RES i, $i \in \mathbb{N}_\mathcal{I}$**)**: Receive $\hat{\mathbf{z}}^{\ell+1}$ and θ^ℓ.

- Compute $\mathbf{z}_i^{\ell+1} = \theta^\ell \mathbf{z}_i^{\star\ell} + (1-\theta^\ell)\mathbf{z}_i^\ell$
- Solve the local minimization problem

$$\mathbf{z}_i^{\star\ell+1} = \underset{\mathbf{z}_i \in \mathbb{D}_i}{\operatorname{argmin}} \; g_i\left(\mathbf{z}_i; \hat{\mathbf{z}}^{\ell+1} - \frac{1}{\mathcal{I}}\mathbf{z}_i^{\ell+1}\right).$$

- Transmit $\mathbf{z}_i^{\star\ell+1}$.

Increment the iteration counter $\ell = \ell + 1$ and repeat the loop.

known to RES i and the CE, but not to the other RESs. Privacy is maintained since only a single RES has access to the average demand $\hat{\mathbf{z}}^{\ell+1}$ from which no individual information of other RESs can be recovered. The communication structure of Algorithm 6 is visualized in Figure 5.1.

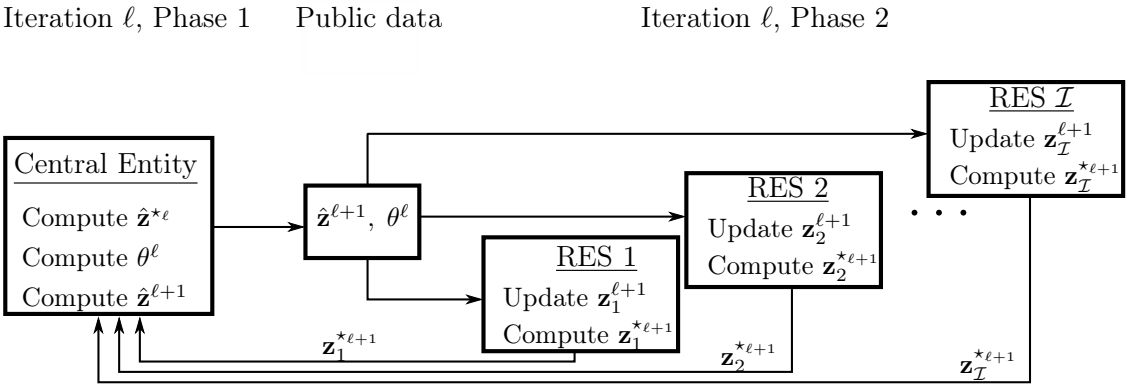

Figure 5.1: *Communication structure of Algorithm 6.*

Remark 5.3.1. *In Algorithm 6 the optimal states* $\mathbf{x}_i^{\star}$ *and the optimal input* $\mathbf{u}_i^{\star}$ *for* $i \in \mathbb{N}_{\mathcal{I}}$ *can either be recovered from the system dynamics and the optimal power demand* $\mathbf{z}_i^{\star}$ *or by computing* $\mathbf{x}_i^{\ell+1} = \theta^{\ell}\mathbf{x}_i^{\star\ell} + (1-\theta^{\ell})\mathbf{x}_i^{\ell}$ *and* $\mathbf{u}_i^{\ell+1} = \theta^{\ell}\mathbf{u}_i^{\star\ell} + (1-\theta^{\ell})\mathbf{u}_i^{\ell}$ *in every iteration* ℓ *similar to the update of* $\mathbf{z}^{\ell}$*.*

5.3.2 Numerical complexity of the distributed optimization algorithm

The numerical complexity for the central entity

In Algorithm 6, the CE has to solve the optimization problem

$$\underset{\theta\in[0,1]}{\operatorname{argmin}}\ \overline{G}(\theta\hat{\mathbf{z}}^{\star\ell} + (1-\theta)\hat{\mathbf{z}}^{\ell}) \tag{5.22}$$

in the unknown θ subject to box constraints independent of the number of systems $\mathcal{I}$. This implies that the numerical complexity is independent of the number of RESs if we neglect the effort to compute the average $\hat{\mathbf{z}}^{\ell+1}$. For many convex objective functions, the minimization problem with respect to θ can be solved explicitly, e.g. for the function

$$\overline{G}(\hat{\mathbf{z}}) = \left\| \hat{\mathbf{z}} - \hat{\zeta}\mathbb{1} \right\|^2 \tag{5.23}$$

with $\hat{\zeta} \in \mathbb{R}$, an explicit solution can be computed.

Example 5.3.2. *The optimal stepsize* θ *in iteration* ℓ *of Algorithm 6 for the cost function* (5.23) *can be computed by projecting the expression*

$$\tilde{\theta} = \frac{\left(\hat{\mathbf{z}}^{\star\ell} - \hat{\mathbf{z}}^{\ell}\right)\left(\hat{\mathbf{z}}^{\ell} - \hat{\zeta}\mathbb{1}\right)^T}{\left(\hat{\mathbf{z}}^{\star\ell} - \hat{\mathbf{z}}^{\ell}\right)\left(\hat{\mathbf{z}}^{\star\ell} - \hat{\mathbf{z}}^{\ell}\right)^T}$$

to the interval $[0,1]$*, i.e.,* $\theta = \max\{0, \min\{\tilde{\theta}, 1\}\}$*. In order to show this, define the function*

$$\begin{aligned}\phi(\theta) &= \overline{G}(\theta\hat{\mathbf{z}}^{\star\ell} + (1-\theta)\hat{\mathbf{z}}^{\ell}) \\ &= \left(\hat{\zeta}\mathbb{1} - \hat{\mathbf{z}}^{\ell} - \theta\left(\hat{\mathbf{z}}^{\star\ell} - \hat{\mathbf{z}}^{\ell}\right)\right)\left(\hat{\zeta}\mathbb{1} - \hat{\mathbf{z}}^{\ell} - \theta\left(\hat{\mathbf{z}}^{\star\ell} - \hat{\mathbf{z}}^{\ell}\right)\right)^{T}.\end{aligned}$$

Since ϕ *is strictly convex, the assertion follows by setting* $\phi'(\theta) = 0$ *and projecting the resulting* θ *on the interval* $[0,1]$*. The derivative is given by*

$$\phi'(\theta) = -2 \cdot \left(\hat{\mathbf{z}}^{\star\ell} - \hat{\mathbf{z}}^{\ell}\right)\left(\hat{\zeta}\mathbb{1} - \hat{\mathbf{z}}^{\ell} - \theta\left(\hat{\mathbf{z}}^{\star\ell} - \hat{\mathbf{z}}^{\ell}\right)\right)^{T} \tag{5.24}$$

$$= 2\theta \cdot \left(\hat{\mathbf{z}}^{\star\ell} - \hat{\mathbf{z}}^{\ell}\right)\left(\hat{\mathbf{z}}^{\star\ell} - \hat{\mathbf{z}}^{\ell}\right)^{T} - 2 \cdot \left(\hat{\mathbf{z}}^{\star\ell} - \hat{\mathbf{z}}^{\ell}\right)\left(\hat{\zeta}\mathbb{1} - \hat{\mathbf{z}}^{\ell}\right)^{T} \tag{5.25}$$

from which the assertion follows. In the case where the explicit expression for θ^{ℓ} *is not defined, i.e.,*

$$\left(\hat{\mathbf{z}}^{\star\ell} - \hat{\mathbf{z}}^{\ell}\right)\left(\hat{\mathbf{z}}^{\star\ell} - \hat{\mathbf{z}}^{\ell}\right)^{T} = 0,$$

we have $\hat{\mathbf{z}}^{\star\ell} = \hat{\mathbf{z}}^{\ell}$ *and according to Lemma 5.2.2,* $z_i^{\star\ell} = z_i^{\ell}$ *for all* $i \in \mathbb{N}_{\mathcal{I}}$ *which implies that the algorithm already found the minimum.*

The goal of Algorithm 5 and 6 is not to reduce the computational complexity of the CE to a minimum but to render the complexity independent of the number of RESs. Instead of one unknown θ one can also consider the case that every system $i \in \mathbb{N}_{\mathcal{I}}$ belongs to a cluster $m \in \mathbb{N}_M$, for a fixed $M \in \mathbb{N}$ which will be denoted by $\mathbf{z}_{m,i}$. Every cluster has its own variable $\theta_m \in [0,1]$ and $\theta^T = (\theta_1 \ldots, \theta_M)$. Without loss of generality, we assume that systems are ordered and we define

$$\mathbf{Z}_m^T = \left(\mathbf{z}_{m,m_1}^T \ \ldots \ \mathbf{z}_{m,m_{\mathcal{I}}}^T\right)$$

for the clusters $m = 1, \ldots, M$ and

$$\mathbf{Z}^T = \left(\mathbf{Z}_1^T \ \ldots \ \mathbf{Z}_M^T\right).$$

For $\theta \in [0,1]^M$, a clustering and given communication variables $\mathbf{z}_i^{\star\ell}$ and $\mathbf{z}_i^{\ell}$ for $i = 1, \ldots, \mathcal{I}$, we define

$$\mathbf{Z}(\theta) = \begin{pmatrix} \theta_1\mathbf{Z}_1^{\star\ell} \\ \vdots \\ \theta_M\mathbf{Z}_M^{\star\ell} \end{pmatrix} + \begin{pmatrix} (1-\theta_1)\mathbf{Z}_1^{\ell} \\ \vdots \\ (1-\theta_M)\mathbf{Z}_M^{\ell} \end{pmatrix}.$$

The minimization problem

$$\underset{\theta\in[0,1]}{\operatorname{argmin}}\ G(\theta\mathbf{Z}^{\star\ell} + (1-\theta)\mathbf{Z}^{\ell})$$

of the CE in Algorithm 5 can be replaced by

$$\underset{\theta\in[0,1]^M}{\operatorname{argmin}}\ G(\mathbf{Z}(\theta)).$$

In this case the CE has to solve a convex optimization problem in M variables subject to box constraints. The computational complexity is still independent of the number of systems and only depends on the number of clusters. The clusters can be used to increase the speed of convergence and is a generalization of the case $M = 1$ described in Algorithm 5. The generalization can also be applied to the function $\overline{G}$ and the variables $\hat{z}$.

The numerical complexity for the residential energy systems

The optimization problem of the RESs in every iteration $\ell \in \mathbb{N}$ is of the same complexity as the optimization problem of the decentralized control scheme considered in Chapter 4. Whereas in the decentralized control setting at every time instant only one optimization problem has to be solved, in Algorithm 6 multiple optimization problems have to be solved until the stopping criteria are reached. However, our numerical simulations indicate that a few iterations ℓ are sufficient to obtain good results if the warm-start technique introduced in Section 4.4.2 is used in the MPC context (see Section 5.5.1 for the numerical results). Moreover, the numerical results in Section 5.5.1 show that the number of iterations ℓ only grows moderately with the number of RESs.

5.4 Extension to non-convex optimization

Currently, Algorithm 5 (or Algorithm 6, respectively) is not applicable to nonlinear dynamics, i.e., throughout this chapter we made the assumption that f_i is linear for all $i \in \mathbb{N}_{\mathcal{I}}$ (cf. Assumption 5.1.1). Imposing linear dynamics leads to a convex optimization problem introduced in Section 2.4. For nonlinear dynamics, it is not guaranteed that local optima are also global optima which makes finding a global optimal solution even more difficult in the centralized case. Additionally, Algorithm 5 uses convex combinations of the form

$$\mathbf{x}_i^{\ell+1} = \theta^\ell \mathbf{x}_i^{\star\ell} + (1-\theta)\mathbf{x}_i^\ell, \qquad \mathbf{u}_i^{\ell+1} = \theta^\ell \mathbf{u}_i^{\star\ell} + (1-\theta)\mathbf{u}_i^\ell, \qquad \mathbf{z}_i^{\ell+1} = \theta^\ell \mathbf{z}_i^{\star\ell} + (1-\theta)\mathbf{z}_i^\ell,$$

for all $i \in \mathbb{N}_{\mathcal{I}}$ and $\theta \in [0,1]$, which can only be used due to the linearity of the functions f_i, h_i and the convexity of the sets $\mathbb{X}_i$ and $\mathbb{U}_i$ for all $i \in \mathbb{N}_{\mathcal{I}}$. If these assumptions are dropped, it cannot be assured that $\mathbf{x}_i^\ell$ and $\mathbf{u}_i^\ell$ are feasible for all $\ell \in \mathbb{N}$ and for all $i \in \mathbb{N}_{\mathcal{I}}$. Nevertheless, the idea of Algorithm 5 is still applicable to find at least a suboptimal solution of the underlying non-convex optimization problem. We generalize the Assumptions 5.1.1 to obtain an algorithm which is applicable to the nonlinear dynamics (3.8).

Assumption 5.4.1. *In the remainder of this section, we assume that the local system dynamics satisfy the following properties for all $i \in \mathbb{N}_{\mathcal{I}}$.*

- *The sets $\mathbb{U}_i \in \mathbb{R}^{m_i}$ are compact and convex and the sets $\mathbb{X}_i \in \mathbb{R}^{n_i}$ are non-empty for all $i = 1, \ldots, \mathcal{I}$.*

- *The functions* $f_i : \mathbb{X}_i \times \mathbb{U}_i \to \mathbb{R}^{n_i}$ *are continuous in* $x_i \in \mathbb{X}_i$ *and* $u_i \in \mathbb{U}_i$ *for all* $i = 1, \ldots, \mathcal{I}$
- *The functions* $h_i : \mathbb{X}_i \times \mathbb{U}_i \times \mathbb{R}^p \to \mathbb{R}^p$ *are affine in* $x_i \in \mathbb{X}_i$ *and* $u_i \in \mathbb{U}_i$ *for fixed* $s_i \in \mathbb{R}^p$ *for all* $i = 1, \ldots, \mathcal{I}$*, i.e.,*

  $$h_i(u_i; s_i) = D_i u_i + c_i(s_i)$$

  for given matrices, $D_i \in \mathbb{R}^{p \times m_i}$ *and given functions* $c_i : \mathbb{R}^p \to \mathbb{R}^p$*.*
- *Additionally, we assume that the local systems satisfy a feasibility assumption. For all* $i \in \mathbb{N}_{\mathcal{I}}$*, for all prediction horizons* $N \in \mathbb{N}$*, for all time steps* $k \in \mathbb{N}$ *and for all initial values* $x_i(k) \in \mathbb{X}_i$*, we assume that there exists* $\mathbf{u}_i(k; N) \in \mathbb{U}_i^{k;N}(x_i(k))$ *such that* $\mathbf{x}_i(k; N) \in \mathbb{X}_i^N$ *holds.*

With these assumptions, we modify Algorithm 5. In particular, the stopping criterion of the CE is changed and a new feasibility variable $\nu_i^\ell \in \{0, 1\}$ is introduced for each RES $i \in \mathbb{N}_{\mathcal{I}}$. The updated formulas are given in Algorithm 7. Since we still assume that $\mathbb{U}_i$ is convex, it holds that

$$\mathbf{u}_i^{\ell+1} = \theta^\ell \mathbf{u}_i^{\star\ell} + (1 - \theta)\mathbf{u}_i^\ell \tag{5.26}$$

is feasible for all $i \in \mathbb{N}_{\mathcal{I}}$ and for all $\ell \in \mathbb{N}$. The same holds for the variable $\mathbf{z}_i$ due to the definition of the function $h_i(u_i; s_i) = D_i u_i + c_i(s_i)$ for all $i \in \mathbb{N}_{\mathcal{I}}$. Since f_i is possibly nonlinear, we have

$$\theta^\ell x_i^{\star\ell}(j+1) + (1-\theta)x_i^\ell(j+1) \neq f_i(\theta^\ell x_i^{\star\ell}(j) + (1-\theta)x_i^\ell(j), u_i^{\ell+1}(j)), \tag{5.27}$$

in general. For this reason, we define

$$x_i^{\ell+1}(j+1) = f_i(x_i^{\ell+1}(j), u_i^{\ell+1}(j)) \tag{5.28}$$

for $j = k, \ldots, k+N-1$, where $\mathbf{x}_i^{\ell+1}$ does not need to satisfy the constraints $\mathbf{x}_i^{\ell+1} \in \mathbb{X}_i^N$ for all $i \in \mathbb{N}_{\mathcal{I}}$. If we disregard the feasibility of $\mathbf{x}_i^{\ell+1}$, we still obtain a monotonically decreasing sequence $(G^\ell)_{\ell \in \mathbb{N}}$ due to the computation of $\mathbf{z}^\ell$ even though it might not correspond to a feasible state $\mathbf{x}^\ell$. Even if $\mathbf{x}^\ell$ in iteration ℓ is infeasible, i.e., $\mathbf{x}^\ell \notin \mathbb{X}^N$ the algorithm is continued as if it were feasible. The feasibility can be easily verified by the RESs. To this end, we define the boolean variable $\nu_i \in \{0, 1\}$, which is sent to the CE in every iteration ℓ, and is set to $\nu_i^\ell := 0$ if $\mathbf{x}_i^\ell \notin \mathbb{X}_i^N$ and is set to $\nu_i^\ell := 1$ if $\mathbf{x}_i^\ell \in \mathbb{X}_i^N$ for $i = 1, \ldots, \mathcal{I}$.
In the case that $\nu_i^\ell = 0$ for at least one RES $i \in \mathbb{N}_{\mathcal{I}}$, a feasible solution in iteration ℓ can be obtained through the definition

$$x_i^{\star\ell}(j+1) = f_i(x_i^{\star\ell}(j), u_i^{\star\ell}(j)) \tag{5.29}$$

for $j = k, \ldots, k+N-1$ and for $i = 1, \ldots, \mathcal{I}$. Thus, in every iteration $\mathbf{u}^{\star\ell}$ and $\mathbf{x}^{\star\ell}$ defined through Equation (5.29) are feasible even if $\mathbf{x}^{\ell+1}$ according to Equation (5.28) is infeasible.

Algorithm 7 Hierarchical distributed optimization for nonlinear dynamics

Input: Define the admissible sets $\mathbb{D}_i$ based on the initial states $x_i(k) \in \mathbb{X}_i$ and the time-dependent quantities $\mathbf{s}_i(k;N)$. Set the number of RESs $\mathcal{I}$, the prediction horizon N, the maximal iteration number $\ell_{\max} \in \mathbb{N} \cup \{\infty\}$ and the desired precisions $\varepsilon_1, \varepsilon_2, \delta > 0$.

Main loop:

Phase 1 (CE): Receive $\mathbf{z}_i^{\star\ell}$ and ν_i^ℓ, $i = 1, \dots, \mathcal{I}$.

- Compute the stepsize $\theta^\ell = \underset{\theta\in[0,1]}{\operatorname{argmin}}\, G(\theta \mathbf{z}^{\star\ell} + (1-\theta)\mathbf{z}^\ell)$.
- Compute $\mathbf{z}^{\ell+1} = \theta^\ell \mathbf{z}^{\star\ell} + (1-\theta^\ell)\mathbf{z}^\ell$ and evaluate the performance indices $G^{\star\ell} = G(\mathbf{z}^{\star\ell})$ and $G^{\ell+1} = G(\mathbf{z}^{\ell+1})$.
- If $|G^{\ell+1} - G^\ell| < \varepsilon_1$,
  - If $\nu_i^\ell = 1$ for all $i \in \mathbb{N}_\mathcal{I}$, transmit θ^ℓ and terminate the algorithm.
  - If there exits an $i \in \mathbb{N}_i$ such that $\nu_i^\ell = 0$ and $|G^{\star\ell} - G^\ell| < \varepsilon_2$, transmit $\theta^\ell := 1$ and terminate the algorithm.
  - Otherwise transmit θ^ℓ and $\mathbf{z}^{\ell+1}$.
- If $\ell = \ell_{\max}$ or $\theta^\ell < \delta$
  - If $\nu_i^\ell = 1$ for all $i \in \mathbb{N}_\mathcal{I}$, transmit θ^ℓ and terminate the algorithm.
  - If there exists an $i \in \mathbb{N}_\mathcal{I}$ with $\nu_i^\ell = 0$, transmit $\theta^\ell := 1$ and terminate the algorithm.
- Otherwise transmit θ^ℓ and $\mathbf{z}^{\ell+1}$.

Phase 2 (RES i, $i \in \mathbb{N}_\mathcal{I}$): Receive θ^ℓ and $\mathbf{z}^{\ell+1}$

- Update the input and the communication variables i.e.,

$$\mathbf{u}_i^{\ell+1} = \theta^\ell \mathbf{u}_i^{\star\ell} + (1-\theta^\ell)\mathbf{u}_i^\ell \quad \text{and} \quad \mathbf{z}_i^{\ell+1} = \theta^\ell \mathbf{z}_i^{\star\ell} + (1-\theta^\ell)\mathbf{z}_i^\ell.$$

- Compute $\mathbf{x}_i^{\ell+1}$ based on $\mathbf{u}_i^{\ell+1}$, i.e.,

$$x_i^{\ell+1}(j+1) = f_i(x_i^{\ell+1}(j), u_i^{\ell+1}(j)), \qquad j = k, \dots, k+N-1.$$

- If $\mathbf{x}_i^{\ell+1} \in \mathbb{X}_i^N$ (i.e., $\mathbf{x}_i^{\ell+1}$ is feasible), set $\nu_i^{\ell+1} = 1$. Otherwise, set $\nu_i^{\ell+1} = 0$.
- Define $\mathbf{p}_i^{\ell+1}$ as in Equation (5.3).
- Solve the minimization problem $\mathbf{z}_i^{\star\ell+1} = \underset{\mathbf{z}_i\in\mathbb{D}_i}{\operatorname{argmin}}\, g_i(\mathbf{z}_i; \mathbf{p}_i^{\ell+1})$ to obtain a minimizer.
- Transmit $\mathbf{z}_i^{\star\ell+1}$ and $\nu_i^{\ell+1}$.

Increment the iteration counter $\ell = \ell + 1$ and repeat the loop.

Hence, if we encounter that $\mathbf{z}^\ell$ is a local minimum, i.e., $\theta^\ell = 0$, or numerically $\theta^\ell \leq \delta$ for a given precision $\delta > 0$, Algorithm 7 is terminated by the CE with $\mathbf{z}_i^{\ell+1}$, $\mathbf{u}_i^{\ell+1}$ and $\mathbf{x}_i^{\ell+1}$ if $\mathbf{x}_i^{\ell+1} \in \mathbb{X}^N$ or with $\mathbf{z}_i^{\star\ell+1}$, $\mathbf{u}_i^{\star\ell+1}$ and $\mathbf{x}_i^{\star\ell+1}$ (corresponding to $\theta^\ell = 1$) if $\mathbf{x}_i^{\ell+1} \notin \mathbb{X}^N$.
In the case that the algorithm does not run into infeasibility in iteration ℓ, i.e., $\mathbf{x}^{\ell+1} \in \mathbf{X}^N$, and additionally $|G^{\ell+1} - G^\ell| < \varepsilon_1$ holds for a given $\varepsilon_1 > 0$, then Algorithm 7 is stopped similar to Algorithm 5. Moreover, if $\mathbf{x}^{\ell+1} \notin \mathbf{X}^N$ but $|G^{\star\ell} - G^\ell| < \varepsilon_2$ for a given $\varepsilon_2 > 0$, Algorithm 7 is also stopped and $\mathbf{z}^{\star\ell}$, $\mathbf{u}^{\star\ell}$ and $\mathbf{x}^{\star\ell}$ are returned.
Even though we will not investigate the convergence properties of Algorithm 7 for nonlinear dynamics here, the numerical results in Section 5.5.2 show its applicability to the nonlinear model of RESs introduced in Chapter 3.

5.5 Numerical simulations

In this section, we analyze the performance and computational complexity of DiMPC using the hierarchical distributed optimization algorithms introduced in the preceding sections. In the first part of the section, we focus on the performance compared to CMPC, and in the second part we concentrate on the performance for different system dynamics of the RESs.
We emphasize that all conclusions presented in the following sections are solely based on the obtained numerical results with the dataset presented in Chapter 3.

5.5.1 Distributed MPC using hierarchical distributed optimization

To demonstrate the performance of the hierarchical distributed optimization Algorithm 6 we use the linear model dynamics (3.1),

$$\begin{aligned} x_i(k+1) &= x_i(k) + Tu_i(k) \\ z_i(k) &= w_i(k) - g_i(k) + u_i(k) \end{aligned}$$

subject to the constraints (3.2) and (3.3) for all $i \in \mathbb{N}_\mathcal{I}$. For the charging and discharging rates the constants $-\underline{u}_i = \overline{u}_i = 0.3$ are used for all $i \in \mathbb{N}_\mathcal{I}$. Moreover, the battery capacities are set to $C_i = 2$, the initial state of charge of the batteries is set to $x_i(0) = 0.5$ for all $i \in \mathbb{N}_\mathcal{I}$ and the discretization parameter is set to $T = 0.5$.[3] For the definition of the cost functional we use Equation (4.4), i.e.,

$$J_N(x(k), \mathbf{u}(k; N)) = \left\| \hat{\mathbf{z}}(k; N) - \hat{\zeta}(k)\mathbb{1} \right\|^2$$

or, equivalently, in the notation of Algorithm 6

$$\overline{G}(\hat{\mathbf{z}}) = \left\| \hat{\mathbf{z}} - \hat{\zeta}\mathbb{1} \right\|^2.$$

[3] We point out that individual parameters $\underline{u}_i$, $\overline{u}_i$ and C_i, $i \in \mathbb{N}_\mathcal{I}$, do not change the performance of the algorithms. The results presented in this section are also valid for non-homogeneous grids with individual parameters. Individual parameters were tested by adding random numbers to the values used in this section.

Unless specified otherwise, all minimization problems involved in the numerical experiments are solved using the Interior Point Optimizer (IPOPT) [103] and the HSL mathematical software library [3] to solve the underlying linear systems of equations.

Benefits of warm-start and variable θ

In Theorem 5.2.5, we showed that the optimal value $G^\star$ obtained by the centralized OCP coincides with the value $\lim_{\ell\to\infty} G^\ell$ obtained by the distributed optimization Algorithm 6. In Figure 5.2, we visualize the number of iterations of this algorithm for the first two days of a simulation length of $\mathcal{N} = 144$, and 20 RESs, which are necessary to ensure the accuracy $|G^\ell(k) - G^\star(k)| \leq 10^{-i}$ for $i = 1, \ldots, 5$. Additionally, Figure 5.2 shows the importance of the variable stepsize θ. If the fixed value $\theta = 1/\mathcal{I}$ is used instead of a variable θ according to the minimization problem (5.22), approximately twice as many iterations are necessary to obtain a certain accuracy. The variable stepsize θ in combination with warm-start reduces the number of iterations even further. Table 5.1 gives associated numerical values. Note that for all accuracies there are time instances at which the warm-start initialization already satisfied the termination criterion of the algorithm. Hence, when using warm-start the minimal number of iterations is always 0.

	without warm-start		with warm-start	
	$\theta = 1/\mathcal{I}$	variable θ	$\theta = 1/\mathcal{I}$	variable θ
average number of iterations				
$\varepsilon = 10^{-1}$	8.61	3.81	1.59	0.27
$\varepsilon = 10^{-2}$	23.90	15.05	13.57	1.44
$\varepsilon = 10^{-3}$	59.33	33.04	28.20	3.86
$\varepsilon = 10^{-4}$	99.85	51.44	42.02	7.67
$\varepsilon = 10^{-5}$	142.69	65.89	55.56	11.57
maximal number of iterations				
$\varepsilon = 10^{-1}$	12	6	20	3
$\varepsilon = 10^{-2}$	42	24	63	16
$\varepsilon = 10^{-3}$	86	46	107	38
$\varepsilon = 10^{-4}$	131	67	152	58
$\varepsilon = 10^{-5}$	176	89	197	69
minimal number of iterations				
$\varepsilon = 10^{-1}$	6	3	0	0
$\varepsilon = 10^{-2}$	10	6	0	0
$\varepsilon = 10^{-3}$	13	11	0	0
$\varepsilon = 10^{-4}$	14	16	0	0
$\varepsilon = 10^{-5}$	16	19	0	0

Table 5.1: *Number of iterations needed to obtain a certain accuracy for variable and fixed θ, with and without warm-start.*

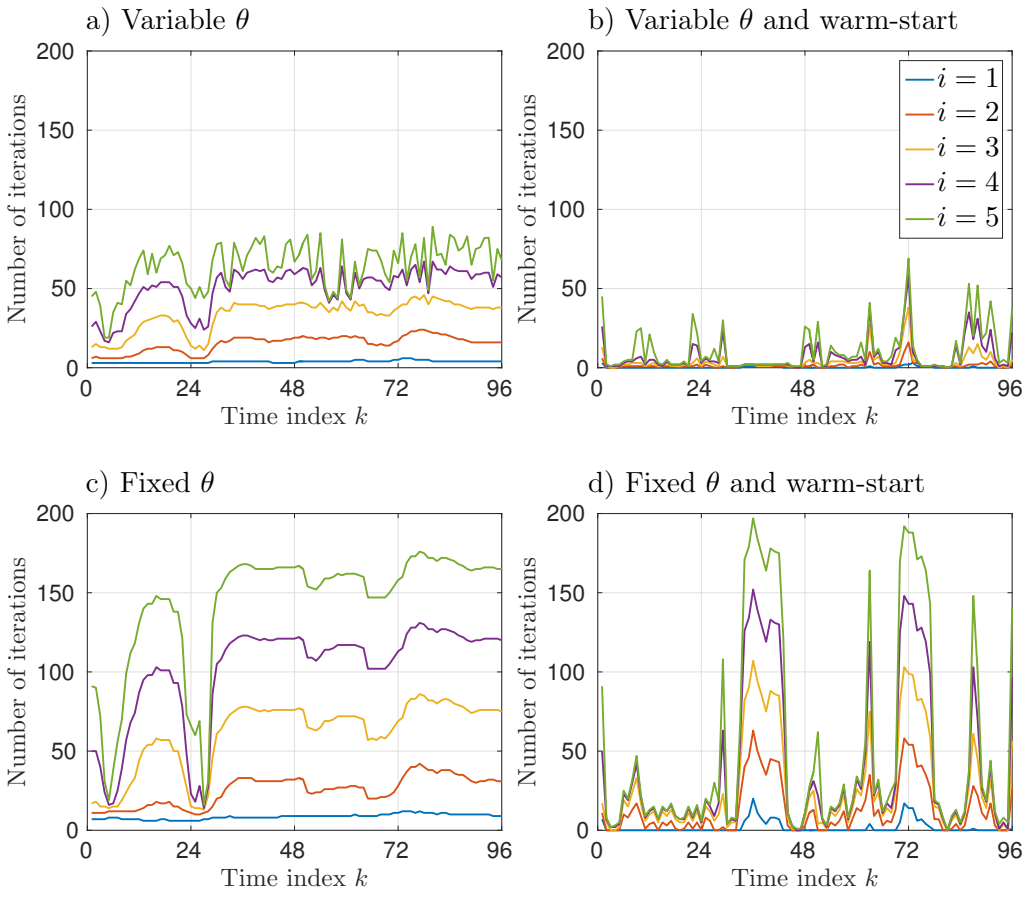

Figure 5.2: *Number of iterations needed to ensure* $|G^\ell(k) - G^\star(k)| \leq 10^{-i}$ *for* $i \in \{1, 2, \ldots, 5\}$ *at time* k *with variable stepsize* θ *according to Remark 5.3.2 (top) and with fixed* $\theta = \mathcal{I}^{-1}$ *(bottom) as well as with (right) and without warm-start (left).* $G^\star$ *denotes the solution of the CMPC problem.*

In Figure 5.3, we visualize

$$\frac{1}{\mathcal{N}} \sum_{k=0}^{\mathcal{N}-1} |G^\ell(k) - G^\star(k)|,$$

i.e., the average deviation from the optimal solution computed by the CMPC algorithm in iteration ℓ. The average is taken with respect to the simulation length $\mathcal{N} = 144$. The figure illustrates the rate of convergence of the hierarchical distributed optimization algorithm using fixed $\theta = 1/\mathcal{I}$ and variable θ with and without warm-start. We obtain linear convergence in all cases and variable θ clearly outperforms the method with fixed θ. After the accuracy of the optimizer is reached, the results do not improve anymore.

Impact of the Number of RESs

In Figure 5.4 the influence of the number of RESs instead of the accuracy is analyzed. We vary the number of RESs from 10 to 300 in steps of 10 and count the number of iterations

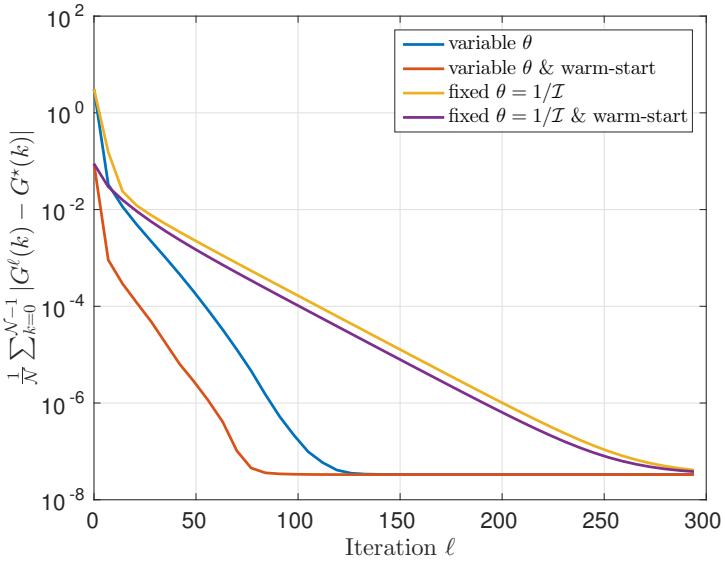

Figure 5.3: *Average speed of convergence of the hierarchical distributed optimization algorithm for different settings.*

until the accuracy $|G^\ell - G^\star| \leq 10^{-2}$ is obtained with and without warm-start. On average the optimization algorithms using warm-start clearly outperform the algorithm without warm-start independent of the number of RESs. However, we also observe that in the worst case, the algorithm with warm-start requires more iterations than the one without. The number of iterations is not independent of the number of systems. The number of iterations seems to increase sublinearly with respect to the number of RESs (for more than 50 RESs).

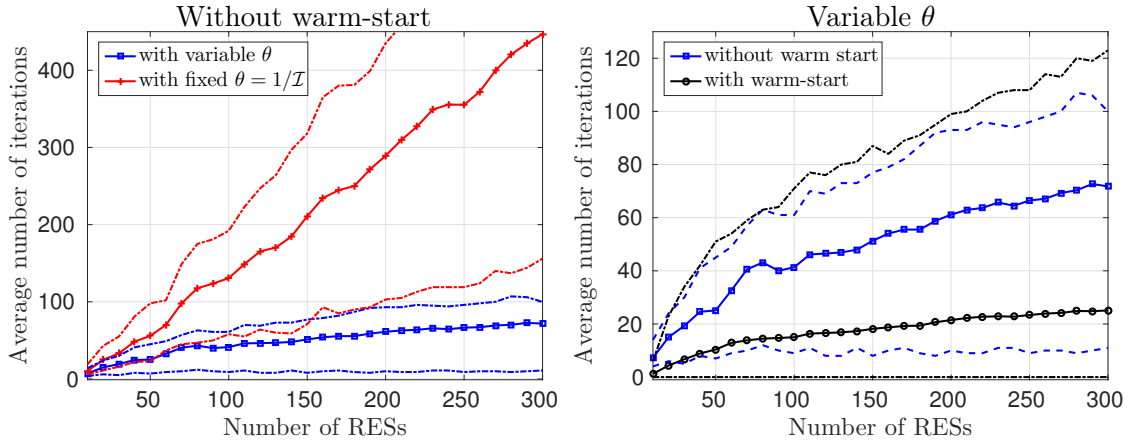

Figure 5.4: *Average number of iterations needed to obtain the accuracy $|G^\ell - G^\star| \leq 10^{-2}$ for a different number of RESs with and without warm-start for fixed and for variable θ. The dashed lines show the maximal and minimal number of iterations to obtain the accuracy $|G^\ell - G^\star| \leq 10^{-2}$.*

Imperfect Optimization

In Figure 5.4 we see that the algorithm needs about 15 iterations on average to obtain an accuracy of 10^{-2} in the setting of 100 RESs, variable θ, and warm-start. However, if we do not iterate up to a certain accuracy and, instead, always solve a fixed number of OCPs at every time step, we conclude that 2 optimization steps are already sufficient to obtain a closed-loop performance which is close to CMPC (cf. Figure 5.5 and Table 5.2). To obtain a comparable result without warm-start, about 10 optimization steps are necessary. Note that Figure 5.5 only shows the first 3 days of a simulations of one week (i.e., $\mathcal{N} = 336$). The missing time steps show a similar behavior and are not visualized to keep the illustration simple. The results for the DiMPC simulations were obtained in MATLAB using a setting of 100 RESs.

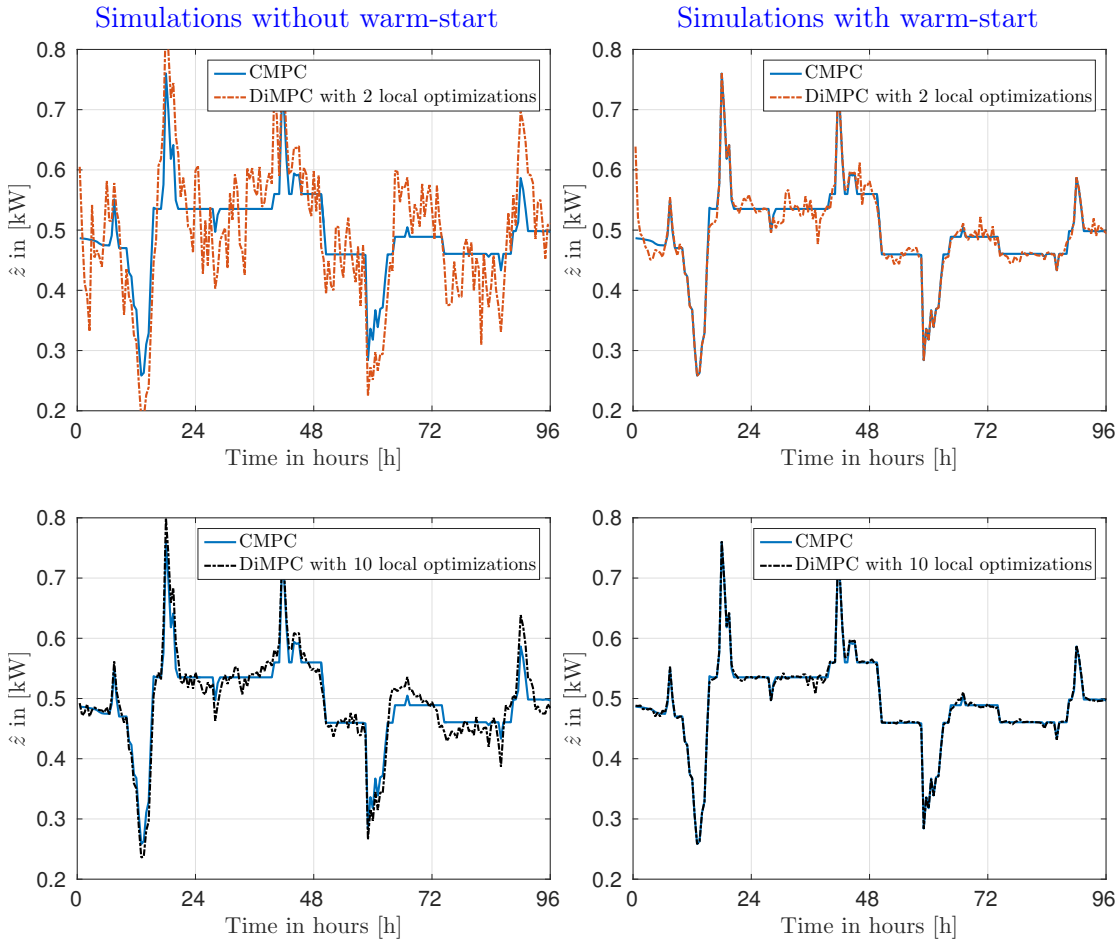

Figure 5.5: *Performance of DiMPC for imperfect optimization without warm-start (left) and with warm-start (right) compared to CMPC.*

Remark 5.5.1. *For the considered dataset in this section, i.e., the* 144 *samples and a*

	PTP	MQD	ASF
Uncontrolled setting	1.1016	0.0505	0.0057
DeMPC	0.7011	0.0187	0.0029
CMPC	0.5016	0.0032	0.0007
DiMPC with warm-start			
2 local optimizations	0.5026	0.0035	0.0008
5 local optimizations	0.5023	0.0033	0.0007
10 local optimizations	0.5023	0.0033	0.0006
DiMPC without warm-start			
2 local optimizations	0.6712	0.0103	0.0036
5 local optimizations	0.5999	0.0060	0.0013
10 local optimizations	0.5638	0.0046	0.0009

Table 5.2: *Performance of DiMPC with imperfect optimization (with and without warm-start) compared to CMPC and DeMPC. For the DiMPC algorithm the RESs solve a fixed amount of optimization problems at every time step k.*

variable number of RESs, the values $G^\star$ are in the interval $[0.054, 1.850]$. A large (small) $G^\star$ corresponds to a large (small) deviation from the average $\hat{\zeta}$. Therefore, we use the absolute error

$$\left|G^\ell - G^\star\right| \leq \varepsilon$$

instead of the relative error

$$\left|G^\ell - G^\star\right| \leq \varepsilon \cdot G^\star$$

as a measure of the quality of the results. If $G^\star$ is small the performance with respect to our metrics is good even if the relative error might still be large. The choice $\varepsilon = 10^{-2}$ seems to be reasonable for our application for most of the numerical simulations, but can be replaced by any other value.

5.5.2 Distributed MPC for different model dynamics

In this section we investigate the performance of DiMPC using the hierarchical distributed optimization Algorithm 6 and 7, respectively, for different system dynamics. In particular we demonstrate that not only the simple model dynamics can be handled by the proposed algorithm. For nonlinear dynamics, the DiMPC approach also clearly outperforms the DeMPC approach. In this section we use a setting of 300 RESs and a simulation length of one month ($\mathcal{N} = 1440$ and $T = 0.5$) with initial conditions $x_i(0) = 1$ for all $i \in \mathbb{N}_{300}$. The simulations are performed in MATLAB. We consider three different system dynamics defined in Section 3.4.1.

Linear system dynamics without losses

As a reference, we use the simplified system dynamics without losses

$$\begin{aligned} x_i(k+1) &= x_i(k) + Tu_i(k), \\ z_i(k) &= w_i(k) - g_i(k) + u_i(k) \end{aligned}$$

subject to the box constraints

$$\begin{array}{rcccl} 0 & \leq & x_i(k) & \leq & 2 \\ -0.3 & \leq & u_i(k) & \leq & 0.3 \end{array}$$

for all $i \in \mathbb{N}_{300}$.

Linear system dynamics with losses

The linear system with losses is defined as

$$\begin{aligned} x_i(k+1) &= x_i(k) + 0.95u_i^+(k) + u_i^-(k), \\ z_i(k) &= w_i(k) - g_i(k) + u_i^+(k) + 0.95u_i^-(k) \end{aligned}$$

for all $i \in \mathbb{N}_{300}$. Here, we consider that 5% of the power is lost in the charging and in the discharging process representing a cycle efficiency of the storage device of circa 90%. The constraints are defined as

$$\begin{array}{rcccl} 0 & \leq & x_i(k) & \leq & 2, \\ -0.3 & \leq & u_i^-(k) & \leq & 0, \\ 0 & \leq & u_i^+(k) & \leq & 0.3, \\ 0 & \leq & -u_i^-(k) + u_i^+(k) & \leq & 0.3 \end{array}$$

Nonlinear system dynamics

The nonlinear system dynamics are defined by Equation (3.8)

$$\begin{aligned} x_i(k+1) = x_i(k) &+ 0.95u_i^+(k) + u_i^-(k) - 0.1 \cdot T\left(u_i^+(k)^2 + u_i^-(k)^2\right) \\ &+ T\left(\frac{0.1u_i^-(k)}{\varepsilon + x_i(k)} - \frac{0.2u_i^+(k)}{2 - 2\varepsilon - x_i(k)}\right) \\ z_i(k) = w_i(k) &- g_i(k) + u_i^+(k) + 0.95u_i^-(k) \end{aligned}$$

for all $i \in \mathbb{N}_{300}$. The parameter $\varepsilon = 10^{-2}$ is included for numerical reasons to prevent the division by zero and to allow the storage device to be fully charged or fully discharged. In addition to the linear constraints

$$\begin{array}{rcccl} 0 & \leq & x_i(k) & \leq & 2, \\ -0.3 & \leq & u_i^-(k) & \leq & 0, \\ 0 & \leq & u_i^+(k) & \leq & 0.3, \\ 0 & \leq & -u_i^-(k) + u_i^+(k) & \leq & 0.3. \end{array}$$

we introduce the nonlinear constraints

$$0.95u_i^+(k) - 0.1u_i^+(k)^2 - \frac{0.2}{2 - 2\varepsilon - x_i(k)}u_i^+(k) \geq 0$$
$$u_i^-(k) - 0.1u_i^-(k)^2 + \frac{0.1}{\varepsilon + x_i(k)}u_i^-(k) \geq -0.3.$$

motivated in Section 3.4.1.

Numerical results

The numerical results comparing the performance of DeMPC and DiMPC for different system dynamics are summarized in Table 5.3 and visualized in Figure 5.6. The results for the DiMPC setting are obtained using the hierarchical distributed optimization Algorithm 6 and 7, respectively. In Figure 5.6, only the first three days of the simulation are shown. The remaining days show a similar behavior.

	PTP	MQD	ASF	LOE
Uncontrolled	1.2031	0.0560	0.0043	0
DeMPC				
Linear model (without losses)	0.8022	0.0198	0.0018	0
Linear model (with losses)	0.8116	0.0203	0.0019	0.0067
Nonlinear model	0.8544	0.0297	0.0019	0.0399
DiMPC				
Linear model (without losses)	0.6067	0.0049	4.62×10^{-4}	0
Linear model (with losses)	0.6214	0.0054	4.98×10^{-4}	0.0066
Nonlinear model	0.6463	0.0089	5.93×10^{-4}	0.0244

Table 5.3: *Performance of distributed and decentralized MPC for different model dynamics and a simulation of one month ($\mathcal{N} = 1440$, $T = 0.5$).*

As expected, the linear model without losses leads to the best results. Nevertheless, the results of the linear model with losses and the nonlinear model show only a slight deterioration. In all the cases, the DiMPC approach outperforms the DeMPC approach without communication (see Table 5.3). Even if up to 10% of the power is lost in the linear model with losses, the difference in the battery profile $\hat{x}$ with respect to the model without losses is negligible (see Figure 5.6). As already seen in Figure 3.6, in the nonlinear dynamics, the battery capacities C_i, $i \in \mathbb{N}_{300}$, reduce, i.e., due to the losses, the maximal SOC can not be reached. It is worth pointing out that in the case with losses, it is not unusual to have times k where $\hat{u}^+(k) \neq 0$ and $\hat{u}^-(k) \neq 0$. This does not only appear in the average variables, but is also observed in the individual variables of the RESs, i.e., we have $u_i^+(k) \neq 0$ and $u_i^-(k) \neq 0$ for RESs $i \in \mathbb{N}_{300}$ and time steps $k \in \mathbb{N}_{\mathcal{N}}$. This implies that wasting energy can lead to an improved performance with respect to the proposed cost functional.

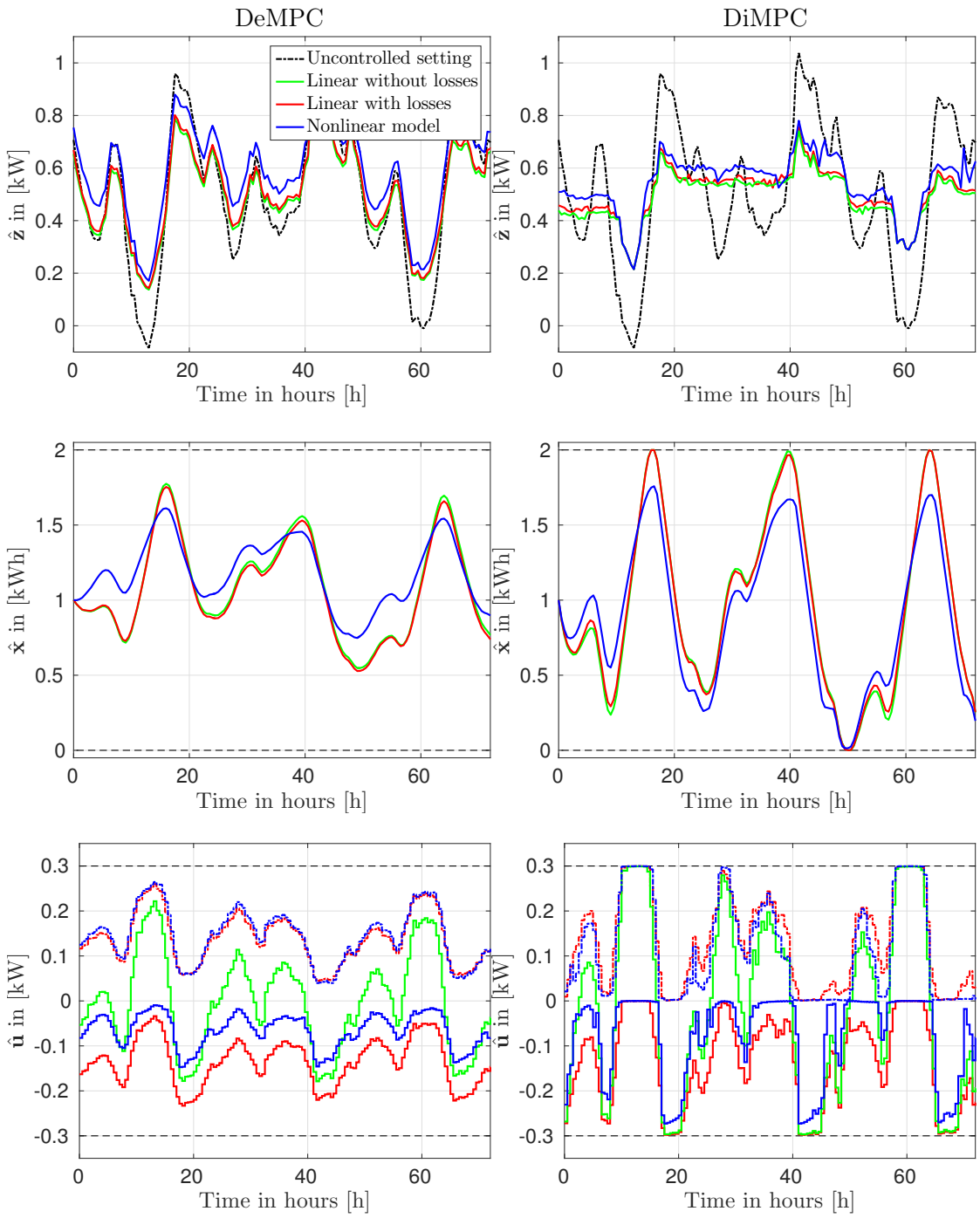

Figure 5.6: *Performance of DeMPC and DiMPC for different model dynamics. For model dynamics considering losses, the input is split in u^+ and u^-.*

Chapter 6

Relaxed distributed optimization using the dual gradient method

In Chapter 5, we proposed a distributed optimization algorithm to solve an optimization problem of the form

$$\begin{array}{ll} \min\limits_{\mathbf{z}_i} & \phi\left(\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\mathbf{z}_i\right) \\ \text{s.t.} & \mathbf{z}_i \in \mathbb{D}_i \quad \forall i = 1,\ldots,\mathcal{I} \end{array} \tag{6.1}$$

for a convex function $\phi : \mathbb{R}^N \to \mathbb{R}$ and convex sets $\mathbb{D}_i \subset \mathbb{R}^N$ for all $i \in \mathbb{N}_{\mathcal{I}}$. The proposed algorithm assumes that subsystems cooperate to find a global minimizer of the optimization problem (6.1), via communication with a CE. In the context of a network of RESs, the approach discussed in Chapter 5 assumes that the RESs are willing to help the CE or the grid provider in peak reduction, without optimizing with respect to their own objectives. In this chapter, we introduce a distributed dual ascent algorithm, based on the dual gradient method [14], to solve a relaxation of the optimization problem (6.1). The dual ascent algorithm offers a non-cooperative interpretation of the solution process.

Similar to the algorithm in the previous chapter, the dual ascent algorithm computes the optimal solution by exchanging the information between the RESs and the CE. In contrast to the cooperative setting, we give a real-time-pricing interpretation of the algorithm where on one hand, the CE stabilizes the grid by minimizing the vertical deviations in the power demand, and on the other hand, the RESs minimize their electricity costs.

To this end, instead of a flat pricing scheme, where the electricity price is independent of the time of the day, we investigate a negotiation based real-time-pricing scheme, where the price of energy is variable and is based on the aggregated power demand $\hat{\mathbf{z}}$ over a certain time interval. In this pricing scheme, RESs can for example react on variable prices by changing their consumption patterns by shifting flexible or controllable loads to times with lower prices or, as will be considered here, by using storage devices like batteries to store energy at times where energy is cheap and use it at times where energy is expensive. In this case, the CE is still responsible for stabilizing the grid but if the correct price signals are sent to the RESs, they indirectly help in the stabilization process and, additionally, can benefit from reduced electricity prices.

The price-based control scheme presented in this chapter is an alternative of the earlier work in [108, 109] and [20]. In these papers, the authors propose an algorithm based on a so-called Market Maker, see [39], [13], [95]. The Market Maker implements a simple iterative strategy to set prices for buying and selling electricity within a network of RESs. Whereas in [108, 109], the convergence of the Market Maker algorithm cannot be shown in general, in this thesis, we establish such convergence for the price-based approach using dual decomposition.

In [68], the authors propose an approach for real-time electricity prices with a similar motivation, i.e., peak reduction and cost minimization. In this paper, the prices are set by the energy provider without a negotiation process and the consumer, or the RES in our notation, minimizes its electricity costs by anticipating future electricity prices. In [89], a similar dual decomposition algorithm to create time-varying adaptive prices is used. However, a different smart grid model is used without local storage devices. Instead, it is assumed that at every time instant $k \in \mathbb{N}$ customers have a lower and an upper bound on their power demand $z_i(k)$, $i \in \mathbb{N}_{\mathcal{I}}$. Then customers maximize their welfare by maximizing their power demand z_i (based on a strictly increasing utility function) minus the costs for electricity. In this case, there is no coupling between two consecutive time steps k and $k+1$ and every time instant can be considered independently. Thus, local storage devices cannot be handled with this approach.

In [102], a game theoretic approach for real-time pricing is presented. The authors consider micro-storage devices, similar to the batteries used in this thesis, for peak-shaving and cost reduction. In contrast to our approach, customers optimize their battery usage based on predicted prices without a negotiation process with the energy provider. The authors in [112] optimize the schedule for controllable loads instead of the optimal usage of batteries. In this reference, a different distributed optimization algorithm with the same communication structure as the one proposed in our work is used. The approach is not embedded in a receding horizon scheme. In [38], a similar distributed dual ascent algorithm in the context for the optimal charging of electric vehicles is considered. The algorithm, however, is not used to model real-time electricity prices and the algorithm is not used in an MPC context. In the context of MPC, dual decomposition algorithms for quadratic separable cost functions are considered in [33, 34, 40, 41], for example.

The remainder of this chapter is structured as follows. In Section 6.1, the Lagrangian function and duality are introduced. In Section 6.2, the dual ascent algorithm is defined and convergence properties of the algorithm are given. Furthermore, it is shown how the algorithm can be implemented in a distributed way. Section 6.3 explains the application of the distributed dual ascent algorithm for a network of RESs. In particular, a cooperative and a non-cooperative implementation of the algorithm are discussed. Here, the non-cooperative scheme introduces a real-time pricing mechanism for electricity prices. The chapter is concluded in Section 6.4 with numerical simulations demonstrating the open-loop and the closed-loop performance of the dual ascent algorithm using the pricing scheme.

6.1 The Lagrangian function and duality

Similar to Chapter 5, we concentrate on the solution of a single OCP at a fixed time instant k. To introduce the notion of duality and the Lagrangian, we define the primal optimization problem

$$\begin{aligned} \min_{\mathbf{y}\in D} \quad & \phi(\mathbf{y}) \\ \text{s.t.} \quad & 0 = \chi(\mathbf{y}) := A\mathbf{y}^T - b. \end{aligned} \tag{P}$$

with a convex function $\phi : \mathbb{R}^n \to \mathbb{R}$, a convex set $D \subset \mathbb{R}^n$ and a matrix $A \in \mathbb{R}^{m\times n}$ and a vector $b \in \mathbb{R}^m$ defining linear equality constraints $\chi : \mathbb{R}^n \to \mathbb{R}^m$. The Lagrangian $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ of the primal problem (P) is defined as

$$\mathcal{L}(\mathbf{y}, \lambda) = \phi(\mathbf{y}) + \lambda(A\mathbf{y}^T - b). \tag{6.2}$$

The variables $\lambda \in \mathbb{R}^m$ are called Lagrange multipliers. Given the primal problem (P) and the Lagrangian (6.2), we can define the dual function $\psi : \mathbb{R}^m \to \mathbb{R} \cup \{-\infty\}$,

$$\psi(\lambda) = \inf_{\mathbf{y}\in D} \mathcal{L}(\mathbf{y}, \lambda), \tag{6.3}$$

and the dual problem

$$\max_{\lambda\in\mathbb{R}^m} \psi(\lambda). \tag{D}$$

Observe that ψ is defined as an extended real-valued function since the infimum does not need to be finite for all $\mathbf{y} \in D$. The primal problem (P), the Lagrangian (6.2) and the dual problem (D) are related through the following results which allow a characterization of the optimal solution of the primal problem (P) based on the solution of the dual (D), and vice-versa. The results and corresponding proofs can, for example, be found in [15, Appendix C].

Theorem 6.1.1 (Duality theorem)**.** *If the primal problem* (P) *has an optimal solution* $\mathbf{y}^\star$*, then the dual problem* (D) *has an optimal solution* $\lambda^\star$ *and the optimal values are equal, i.e.,*

$$\phi(\mathbf{y}^\star) = \phi^\star = \psi^\star = \psi(\lambda^\star).$$

Theorem 6.1.2 (Saddle point theorem)**.** *For the primal problem* (P) *and the dual problem* (D)*, the following holds:* $\mathbf{y}^\star$ *is optimal for the primal problem* (P) *and* $\lambda^\star$ *is optimal for the dual problem* (D) *if and only if the saddle point condition*

$$\mathcal{L}(\mathbf{y}^\star, \lambda) \le \mathcal{L}(\mathbf{y}^\star, \lambda^\star) \le \mathcal{L}(\mathbf{y}, \lambda^\star) \tag{6.4}$$

holds for all $\mathbf{y} \in D$ *and for all* $\lambda \in \mathbb{R}^m$*.*

Hence, with these two theorems, if an optimal solution of the dual problem is known, a solution of the primal problem can be computed based on the Lagrangian by solving the minimization problem

$$\mathbf{y}^\star \in \operatorname*{argmin}_{\mathbf{y}\in D} \mathcal{L}(\mathbf{y}, \lambda^\star). \tag{6.5}$$

The advantage of the minimization problem (6.5) over the original primal problem (P) is that the constraints $\chi(\mathbf{y}) = 0$ do not have to be taken into account. Likewise, if $\mathbf{y}^\star$ is known an optimal solution of the dual problem is obtained by solving the unconstrained problem

$$\lambda^\star \in \operatorname*{argmax}_{\lambda\in\mathbb{R}^m} \mathcal{L}(\mathbf{y}^\star, \lambda). \tag{6.6}$$

Moreover, if the function ϕ is strictly convex and D is convex and compact, the primal problem has a unique optimal solution $\mathbf{y}^\star$ (by Theorem 2.4.9) and thus, the dual problem also has an optimal solution $\lambda^\star$. We exploit these properties in the dual ascent algorithm introduced in the next section.

6.2 The dual ascent method

6.2.1 Definition and convergence of the dual ascent method

Instead of solving the primal optimization problem (P) directly, we take a look at an optimal solution of the dual problem (D) from which an optimal solution of the primal problem is obtained. If the dual function ψ is known, the dual problem (D) can be solved by a gradient ascent method. The key idea is to find a sequence $(\lambda^\ell)_{\ell\in\mathbb{N}}$ such that $\psi(\lambda^{\ell+1}) > \psi(\lambda^\ell)$ holds for all $\ell \in \mathbb{N}$. Typically, directions $(d^\ell)_{\ell\in N}$ satisfying the ascending condition

$$d^\ell \nabla\psi(\lambda^\ell)^T > 0 \tag{6.7}$$

are computed to fulfill this task, e.g. $d^\ell = \nabla\psi(\lambda^\ell)$ is a possible choice. Then, the gradient method is defined by

$$\lambda^{\ell+1} = \lambda^\ell + c^\ell d^\ell \tag{6.8}$$

using a sequence $(c^\ell)_{\ell\in N}$ of suitable stepsizes. Here, we distinguish between the following two cases:

- **Constant stepsize** $c > 0$: $c^\ell = c$ for all $\ell \in \mathbb{N}$.
- **Diminishing stepsize**: a sequence $(c^\ell)_{\ell\in\mathbb{N}}$ satisfying

$$c^\ell \overset{\ell\to\infty}{\longrightarrow} 0 \qquad \text{and} \qquad \sum\nolimits_{\ell=0}^{\infty} c^\ell = \infty, \tag{6.9}$$

  e.g. the harmonic numbers $c^\ell = 1/(\ell+1)$.

Convergence of the gradient method can be proven under Lipschitz continuity of the gradient $\nabla\psi$ [14, Proposition 1.2.3 and Proposition 1.2.4].

Theorem 6.2.1 (Convergence of the dual ascent method)**.** *Assume the existence of a Lipschitz constant $L > 0$ such that*

$$\|\nabla\psi(\lambda_1) - \nabla\psi(\lambda_2)\| \leq L\|\lambda_1 - \lambda_2\|$$

holds for all $\lambda_1, \lambda_2 \in \mathbb{R}^m$. Let $(\lambda^\ell)_{\ell\in\mathbb{N}}$ be generated according to (6.8) with ascent directions $(d^\ell)_{\ell\in\mathbb{N}}$ satisfying (6.7). In addition, let one of the following two conditions be satisfied.

- ***Constant stepsize**: There exists $\varepsilon > 0$ such that*

$$\varepsilon \leq c^\ell \leq (2-\varepsilon)\cdot\frac{|d^\ell\nabla\psi(\lambda^\ell)^T|}{L\|d^\ell\|^2}$$

  and $d^\ell \neq 0$ holds for all $\ell \in \mathbb{N}$.

- ***Diminishing stepsize**: There exist positive scalars c_1, c_2 such that the stepsize $(c^\ell)_{\ell\in\mathbb{N}}$ satisfies (6.9) and*

$$c_1\|\nabla\psi(\lambda^\ell)\|^2 \leq d^\ell\nabla\psi(\lambda^\ell)^T, \quad \|d^\ell\|^2 \leq c_2\|\nabla\psi(\lambda^\ell)\|^2.$$

Then, either $\psi(\lambda^\ell) \to -\infty$ for $\ell \to \infty$ or $(\psi(\lambda^\ell))_{\ell\in\mathbb{N}}$ converges to a finite value and $\nabla\psi(\lambda^\ell) \to 0$ for $\ell \to \infty$. Furthermore, every accumulation point of $(\lambda^\ell)_{\ell\in\mathbb{N}}$ is a stationary point of ψ.

6.2.2 The dual ascent algorithm

Before we present the dual ascent algorithm, we show first how an ascent direction can be computed. Since a closed expression of the dual function ψ and its derivative $\nabla\psi$ is not known, we use the Lagrangian to compute $\nabla\psi(\lambda^\ell)$ for a given Lagrange multiplier $\lambda^\ell \in \mathbb{R}^m$. The corresponding results read as follows [14, Prop. 6.1.1].

Theorem 6.2.2. *Let $\phi : \mathbb{R}^n \to \mathbb{R}$ be strongly convex and suppose that $\mathbf{y}_\lambda \in D$ is the unique minimizer of the Lagrangian $\mathcal{L}(\cdot, \lambda)$ for a given $\lambda \in \mathbb{R}^m$ over the convex and closed set D. Then, the dual function $\psi : \mathbb{R}^m \to \mathbb{R} \cup \{\infty\}$ is continuously differentiable and satisfies*

$$\nabla\psi(\lambda) = A\mathbf{y}_\lambda^T - b.$$

To show the Lipschitz-continuity of $\nabla\psi$ necessary for the convergence statement in Theorem 6.2.1, a few definitions and auxiliary results are needed.

Definition 6.2.3. *For the dual function $\psi : \mathbb{R}^m \to \mathbb{R}\cup\{-\infty\}$, we define the finite domain of ψ as*

$$D_\psi := \{\lambda \in \mathbb{R}^m | \psi(\lambda) > -\infty\}.$$

Based on Definition 6.2.3, concavity of the dual function can be shown, cf. [14, Proposition 5.1.2].

Proposition 6.2.4. *The finite domain D_ψ of the dual function $\psi : \mathbb{R}^m \to \mathbb{R} \cup \{-\infty\}$ is convex and ψ is concave.*

Proof. Let $\mathbf{y} \in D$, $\lambda_1, \lambda_2 \in D_\psi$, and $\beta \in [0,1]$ be given. Then, the linearity of the Lagrangian with respect to the Lagrange multiplier λ implies that

$$\mathcal{L}(\mathbf{y}, \beta\lambda_1 + (1-\beta)\lambda_2) = \beta\mathcal{L}(\mathbf{y}, \lambda_1) + (1-\beta)\mathcal{L}(\mathbf{y}, \lambda_2).$$

Taking the minimum on both sides leads to the estimate

$$\min_{\mathbf{y}\in D} \mathcal{L}(\mathbf{y}, \beta\lambda_1 + (1-\beta)\lambda_2) \geq \min_{\mathbf{y}\in D} \beta\mathcal{L}(\mathbf{y}, \lambda_1) + \min_{\mathbf{y}\in D} (1-\beta)\mathcal{L}(\mathbf{y}, \lambda_2)$$

which by definition is equivalent to

$$\psi(\beta\lambda_1 + (1-\beta)\lambda_2) \geq \beta\psi(\lambda_1) + (1-\beta)\psi(\lambda_2),$$

i.e., ψ is concave. Furthermore, for $\lambda_1, \lambda_2 \in D_\psi$, the last inequality implies $\psi(\beta\lambda_1 + (1-\beta)\lambda_2) > -\infty$, and thus, $\beta\lambda_1 + (1-\beta)\lambda_2 \in D_\psi$ for all $\beta \in [0,1]$, which shows convexity of D_ψ and completes the proof. □

With the help of Theorem 6.2.2, an ascent direction can be computed by first minimizing the Lagrangian for a fixed λ and then evaluating the function χ. Theorem 2.4.9 provides characteristics to ensure that the assumption on the uniqueness of the minimizer of the Lagrangian $\mathcal{L}(\cdot, \lambda)$ are satisfied for a fixed λ. Hence, we either require that ϕ is strongly convex on a closed convex domain or we demand that ϕ is strictly convex on a compact and convex domain. It remains to show that $\nabla\psi$ is Lipschitz continuous. To this end, the following general property of projections is required [14, Proposition 2.1.3].

Theorem 6.2.5 (Projection Theorem)**.** *Let the set $D \subset \mathbb{R}^n$ be non-empty, closed, and convex.*

(a) For $\mathbf{y} \in \mathbb{R}^n$, there uniquely exists $[\mathbf{y}]^+ \in D$ satisfying

$$[\mathbf{y}]^+ = \operatorname*{argmin}_{\mathbf{v}\in D} \|\mathbf{v} - \mathbf{y}\|.$$

This vector $[\mathbf{y}]^+$ is called the projection of $\mathbf{y}$ on D.

(b) The mapping $\eta : \mathbb{R}^n \to D$ defined by $\eta(\mathbf{y}) = [\mathbf{y}]^+$ is continuous and nonexpansive, i.e.,

$$\left\|[\mathbf{y}_1]^+ - [\mathbf{y}_2]^+\right\| \leq \|\mathbf{y}_1 - \mathbf{y}_2\|$$

for all $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^n$.

Theorem 6.2.6. *Consider the primal problem* (P) *with closed and convex domain D. Assume that ϕ is continuously differentiable and strongly convex with parameter α according to* (2.19)*. Then, the dual ψ is concave and $\nabla\psi$ is Lipschitz continuous with Lipschitz constant $L := \|A\|^2/\alpha$.*

Proof. The following proof is based on ideas presented in [28]. Let $\mathbf{y}_1, \mathbf{y}_2 \in D$, $\mathbf{y}_1 \neq \mathbf{y}_2$. Using Inequality (2.20) and the Cauchy-Schwartz inequality yields

$$\alpha\|\mathbf{y}_1 - \mathbf{y}_2\|^2 \leq (\nabla\phi(\mathbf{y}_1) - \nabla\phi(\mathbf{y}_2))(\mathbf{y}_1 - \mathbf{y}_2)^T \leq \|\nabla\phi(\mathbf{y}_1) - \nabla\phi(\mathbf{y}_2)\| \cdot \|\mathbf{y}_1 - \mathbf{y}_2\|,$$

or equivalently

$$\|\mathbf{y}_1 - \mathbf{y}_2\| \leq \frac{1}{\alpha}\|\nabla\phi(\mathbf{y}_1) - \nabla\phi(\mathbf{y}_2)\|. \tag{6.10}$$

For fixed λ, the minimizer $\tilde{\mathbf{y}}_\lambda$ of the Lagrangian $\mathcal{L}(\cdot, \lambda)$ on $\mathbb{R}^n$, i.e.,

$$\tilde{\mathbf{y}}_\lambda := \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^n} \mathcal{L}(\mathbf{y}, \lambda),$$

(see also Theorem 6.2.2) is given by the solution of

$$\nabla\phi(\tilde{\mathbf{y}}_\lambda) + A^T\lambda^T = 0. \tag{6.11}$$

Since ϕ is strongly convex, the Lagrangian $\mathcal{L}(\cdot, \lambda)$ is also strongly convex and $\tilde{\mathbf{y}}_\lambda$ is unique. Using (6.11) to substitute the gradient $\nabla\phi(\cdot)$ in (6.10) yields

$$\alpha\|\mathbf{y}_{\lambda_1} - \mathbf{y}_{\lambda_2}\| \leq \left\|-A^T\lambda_1^T + A^T\lambda_2^T\right\| \leq \|A\|\,\|\lambda_1 - \lambda_2\|.$$

With Theorem 6.2.5, we obtain the estimate

$$\begin{aligned}\|\mathbf{y}_{\lambda_1} - \mathbf{y}_{\lambda_2}\| &= \left\|[\tilde{\mathbf{y}}_{\lambda_1}]^+ - [\tilde{\mathbf{y}}_{\lambda_2}]^+\right\| \\ &\leq \|\tilde{\mathbf{y}}_{\lambda_1} - \tilde{\mathbf{y}}_{\lambda_2}\| \\ &\leq \frac{\|A\|}{\alpha}\|\lambda_1 - \lambda_2\|,\end{aligned} \tag{6.12}$$

where $\mathbf{y}_{\lambda_1}$ and $\mathbf{y}_{\lambda_2}$ denote the unique minimizer of the Lagrangian for fixed Lagrange multiplier subject to the constraint set D. Since the gradient of the dual function can be evaluated using the expression $\nabla\psi(\lambda) = A\mathbf{y}_\lambda^T - b$, one obtains the estimate

$$\begin{aligned}\|\nabla\psi(\lambda_1) - \nabla\psi(\lambda_2)\| &= \|(A\mathbf{y}_{\lambda_1}^T - b) - (A\mathbf{y}_{\lambda_2}^T - b)\| \\ &\leq \|A\|\|\mathbf{y}_{\lambda_1} - \mathbf{y}_{\lambda_2}\|.\end{aligned} \tag{6.13}$$

Combining Equation (6.12) and (6.13) provides the Lipschitz continuity estimate

$$\|\nabla\psi(\lambda_1) - \nabla\psi(\lambda_2)\| \leq \frac{\|A\|^2}{\alpha}\|\lambda_1 - \lambda_2\|$$

which shows the assertion. □

With the results presented in this section the dual ascent algorithm can be stated and is summarized in Algorithm 8.

Algorithm 8 Dual ascent algorithm

Input:

- Strongly convex function $\phi : \mathbb{R}^n \to \mathbb{R}$,
- Non-empty, closed and convex set $D \subset \mathbb{R}^n$, and
- Linear coupling $\chi : \mathbb{R}^n \to \mathbb{R}^m$, $\chi(\mathbf{y}) = A\mathbf{y}^T - b$ with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.

Initialization: Set $\ell = 0$ and $\lambda^0 \in \mathbb{R}^m$.
Main loop:

- Solve the minimization problem

$$\mathbf{y}^\ell := \underset{\mathbf{y} \in D}{\operatorname{argmin}} \left(\phi(\mathbf{y}) + \lambda^\ell A \mathbf{y}^T \right). \tag{6.14}$$

- Compute an ascent direction d^ℓ, e.g. $d^\ell = \chi(\mathbf{y}^\ell)$.
- Choose a stepsize c^ℓ according to Theorem 6.2.1 and update the Lagrange multiplier

$$\lambda^{\ell+1} = \lambda^\ell + c^\ell \cdot d^\ell.$$

Increment the iteration counter $\ell = \ell + 1$ and repeat the loop.

6.2.3 The distributed dual ascent algorithm

The dual ascent Algorithm 8 is an efficient method, if the solution of the minimization problem (6.14) can be obtained easily or if the objective function is ϕ is separable, and hence, the minimization problem (6.14) can be solved in a distributed way. Here, we will introduce a distributed dual ascent algorithm after we give a proper definition of a separable functions.

Definition 6.2.7 (Separable functions)**.** *Let $\phi : \mathbb{R}^n \to \mathbb{R}$ be a function in the variables*

$$\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_\mathcal{I})$$

with $\mathbf{y}_i \in \mathbb{R}^{n_i}$, $i \in \mathbb{N}_\mathcal{I}$ and $n = \sum_{i=1}^{\mathcal{I}} n_i$. Then, ϕ is called separable if there exist functions $\phi_i : \mathbb{R}^{n_i} \to \mathbb{R}$, $i \in \mathbb{N}_\mathcal{I}$, such that

$$\phi(\mathbf{y}) = \sum_{i=1}^{\mathcal{I}} \phi_i(\mathbf{y}_i)$$

holds for all $\mathbf{y} \in \mathbb{R}^n$.

We assume that the objective function $\phi : \mathbb{R}^n \to \mathbb{R}$ can be written in the form

$$\phi(\mathbf{y}) = \overline{\phi} \left(\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{y}_i \right) + \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \phi_i(\mathbf{y}_i)$$

for strongly convex functions $\overline{\phi} : \mathbb{R}^N \to \mathbb{R}$, and $\phi_i : \mathbb{R}^N \to \mathbb{R}$, $i \in \mathbb{N}_{\mathcal{I}}$, and $N \in \mathbb{N}$. Since the function $\overline{\phi}$ and the function ϕ_i depend on the variable $\mathbf{y}_i$ for $i \in \mathbb{N}_{\mathcal{I}}$, the function ϕ is not separable. Nevertheless, by introducing the variables $\hat{\mathbf{a}} \in \mathbb{R}^N$ and the linear constraints $\chi : \mathbb{R}^{\mathcal{I} \times N} \times \mathbb{R}^N \to \mathbb{R}^N$,

$$\chi(\mathbf{y}, \hat{\mathbf{a}}) = -\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{y}_i^T + \hat{\mathbf{a}}^T,$$

we observe that the minimization problem

$$\min_{\mathbf{y} \in D} \phi(\mathbf{y})$$

is equivalent to the minimization problem

$$\begin{aligned} \min_{\hat{\mathbf{a}} \in \mathbb{R}^N,\, \mathbf{y}_i \in D_i} \quad & \overline{\phi}\,(\hat{\mathbf{a}}) + \tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \phi_i(\mathbf{y}_i) \\ \text{s.t.} \quad & -\tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \mathbf{y}_i^T + \hat{\mathbf{a}}^T = 0 \end{aligned} \tag{6.15}$$

for $D_i \subset \mathbb{R}^N$, $i \in \mathbb{N}_{\mathcal{I}}$, defined such that $D = D_1 \times \ldots \times D_{\mathcal{I}}$. The Lagrangian of the minimization problem (6.15) is given by

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{a}}, \lambda) = \overline{\phi}\,(\hat{\mathbf{a}}) + \lambda \hat{\mathbf{a}}^T + \frac{1}{\mathcal{I}} \left(\sum_{i=1}^{\mathcal{I}} \phi_i(\mathbf{y}_i) - \lambda \mathbf{y}_i^T \right), \tag{6.16}$$

and thus, the Lagrangian $\mathcal{L}(\cdot, \cdot, \lambda)$ is separable for fixed Langrange multipliers $\lambda \in \mathbb{R}^N$. This implies that a solution of the optimization problem (6.14) can be obtained by solving the minimization problems

$$\hat{\mathbf{a}}^{\star_\lambda} = \operatorname*{argmin}_{\hat{\mathbf{a}} \in \mathbb{R}^N} \left(\overline{\phi}(\hat{\mathbf{a}}) + \hat{\mathbf{a}} \lambda^T \right)$$

and

$$\mathbf{y}_i^{\star_\lambda} = \operatorname*{argmin}_{\mathbf{y}_i \in D_i} \left(\phi_i(\mathbf{y}_i) - \mathbf{y}_i \lambda^T \right)$$

for $i \in \mathbb{N}_{\mathcal{I}}$ in parallel. Since $\overline{\phi}$ and ϕ_i, $i = 1, \ldots, \mathcal{I}$, are strongly convex by assumption, also the functions $\overline{\phi}(\hat{\mathbf{a}}) - \hat{\mathbf{a}} \lambda^T$ and $\phi_i(\mathbf{y}_i) + \mathbf{y}_i \lambda^T$, $i = 1, \ldots, \mathcal{I}$ are strongly convex for fixed $\lambda \in \mathbb{R}^N$. The corresponding distributed dual ascent algorithm is summarized in Algorithm 9.

Remark 6.2.8. *The distributed dual ascent algorithm can be applied to the more general problem*

$$\min_{\mathbf{y}_i \in D_i} \quad \sum_{i=1}^{\mathcal{I}+1} \phi_i(\mathbf{y}_i) \qquad \textit{s.t.} \qquad \sum_{i=1}^{\mathcal{I}+1} A_i \mathbf{y}_i^T - b = 0$$

where $\phi_i : D_i \to \mathbb{R}$, $D_i \subset \mathbb{R}^{n_i}$, $n_i \in \mathbb{N}$, *for* $i = 1, \ldots, \mathcal{I}+1$, *and* $A_i \in \mathbb{R}^{m \times n_i}$, $i = 1, \ldots, \mathcal{I}+1$ *and* $b \in \mathbb{R}^m$. *Since we consider optimization problems in the unknowns* $\mathbf{z}_i$ *and* $\hat{\mathbf{z}} = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i$, $i \in \mathbb{N}_{\mathcal{I}}$, *we use the notation used in Algorithm 9 for simplicity of exposition.*

Algorithm 9 Distributed dual ascent algorithm

Input:

- Strongly convex functions $\overline{\phi} : \mathbb{R}^N \to \mathbb{R}$ and $\phi_i : \mathbb{R}^N \to \mathbb{R}$, $i = 1, \ldots, \mathcal{I}$,
- Non-empty, closed and convex sets $D_i \subset \mathbb{R}^N$, $i = 1, \ldots, \mathcal{I}$, and
- Linear coupling $\chi : \mathbb{R}^{N(\mathcal{I}+1)} \to \mathbb{R}^N$, $\chi(\mathbf{y}, \hat{\mathbf{a}}) = -\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{y}_i^T + \hat{\mathbf{a}}^T$.

Initialization: Set $\ell = 0$ and $\lambda^0 \in \mathbb{R}^m$.
Main Loop:
Phase 1 (Subsystem i, $i \in \mathbb{N}_{\mathcal{I}}$): Receive λ^ℓ

- Solve the minimization problem

$$\mathbf{y}_i^\ell := \underset{\mathbf{y}_i \in D_i}{\operatorname{argmin}} \left(\phi_i(\mathbf{y}_i) - \lambda^\ell \mathbf{y}_i^T \right) \tag{6.17}$$

  and send $\mathbf{y}_i^\ell$ to the CE.

Phase 2 (CE): Receive $\mathbf{y}_i^\ell$ for all $i = 1, \ldots, \mathcal{I}$

- Solve the minimization problem

$$\hat{\mathbf{a}}_i^\ell := \underset{\hat{\mathbf{a}} \in \mathbb{R}^N}{\operatorname{argmin}} \left(\overline{\phi}(\hat{\mathbf{a}}) + \lambda^\ell \hat{\mathbf{a}}^T \right). \tag{6.18}$$

- Compute an ascent direction d^ℓ, e.g. $d^\ell := \chi(\mathbf{y}^\ell, \hat{\mathbf{a}}^\ell)$.
- Choose a stepsize c^ℓ according to Theorem 6.2.1 and update the Lagrange multiplier

$$\lambda^{\ell+1} := \lambda^\ell + c^\ell \cdot d^\ell$$

  and broadcast $\lambda^{\ell+1}$.

Increment the iteration counter $\ell = \ell + 1$ and repeat the loop.

6.3 The dual ascent algorithm for a network of RESs

In this section we show how the distributed dual ascent Algorithm 9, can be applied to the network of RESs introduced in Chapter 3. In particular, we consider the linear model dynamics

$$x_i(k+1) = \alpha_i x_i(k) + T \left(\beta_i u_i^+(k) + u_i^-(k) \right) \tag{6.19a}$$
$$z_i(k) = s_i(k) + u_i^+(k) + \gamma_i u_i^-(k) \tag{6.19b}$$

subject to the constraints

$$\begin{aligned} 0 &\leq & x_i(k) & \leq C_i & \text{(6.20a)}\\ \underline{u}_i &\leq & u_i^-(k) & \leq 0 & \text{(6.20b)}\\ 0 &\leq & u_i^+(k) & \leq \overline{u}_i & \text{(6.20c)}\\ 0 &\leq & \frac{u_i^-(k)}{\underline{u}_i} + \frac{u_i^+(k)}{\overline{u}_i} & \leq 1 & \text{(6.20d)} \end{aligned}$$

introduced in Section 3.4. Together with the system dynamics and the constraints, we recall the sets $\mathbb{D}_i$, $i \in \mathbb{N}_{\mathcal{I}}$, and $\mathbb{D} = \mathbb{D}_1 \times \ldots \times \mathbb{D}_{\mathcal{I}}$ defined by Equation (2.4) at a fixed time instant $k \in \mathbb{N}$. Since the system dynamics (6.19) and the constraints (6.20) are linear, the sets $\mathbb{D}_i$, $i \in \mathbb{N}_{\mathcal{I}}$, and the set $\mathbb{D}$ are convex and compact (cf. Lemma 5.1.2).[1] In the following section, we show how the distributed dual ascent Algorithm 9 can be used to find an approximation of the optimal solution of the minimization problem

$$\underset{\mathbf{z}_i \in \mathbb{D}_i}{\operatorname{argmin}} \left\| \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i - \mathbb{1}\hat{\zeta} \right\|^2 \tag{6.21}$$

considered in the previous chapter.

6.3.1 Cooperative application of the dual ascent algorithm

To be able to apply Algorithm 9, we rewrite the optimization problem (6.21) and introduce the variables

$$\hat{\mathbf{a}} = \hat{\mathbf{z}} = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i.$$

Thus, the optimization problem (6.21) can be equivalently written as

$$\begin{aligned} &\underset{\mathbf{z} \in \mathbb{D},\, \hat{\mathbf{a}} \in \mathbb{R}^N}{\operatorname{argmin}} && \overline{G}(\hat{\mathbf{a}}) \\ &\text{s.t.} && \chi(\mathbf{z}, \hat{\mathbf{a}}) = 0 \end{aligned} \tag{6.22}$$

with $\overline{G} : \mathbb{R}^N \to \mathbb{R}$, $\overline{G}(\hat{\mathbf{a}}) = \frac{\eta}{2} \left\| \hat{\mathbf{a}} - \mathbb{1}\hat{\zeta} \right\|^2$ for fixed $\eta > 0$ and $\chi : \mathbb{R}^{\mathcal{I} \times N} \times \mathbb{R}^N \to \mathbb{R}^N$, $\chi(\mathbf{z}, \hat{\mathbf{a}}) = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i^T - \hat{\mathbf{a}}^T$. Note that the scaling $\eta > 0$ only changes the minimum $\overline{G}^\star$ but not the (possibly non-unique) minimizers $(\mathbf{z}^\star, \hat{\mathbf{a}}^\star)$.
Since the function $\overline{G}$ does not depend on the variables $\mathbf{z}$, the objective function is not strongly convex and thus, the assumptions of the dual ascent algorithm are not satisfied.

[1] The results presented in the following also hold under the more general setting of Assumption 5.1.1, We use the particular system dynamics (6.19) subject to the constraints (6.20), in particular, for illustrations in Section 6.4 showing numerical results. For non-convex sets $\mathbb{D}_i$, $i \in \mathbb{N}_{\mathcal{I}}$, convergence of the distributed dual algorithm cannot be guaranteed.

Therefore, we define the functions $F_i : \mathbb{R}^N \to \mathbb{R}$, $F_i(\mathbf{z}_i) = \frac{1}{\mathcal{I}}\frac{\delta}{2}\|\mathbf{z}_i\|^2$ for $i = 1, \dots, \mathcal{I}$ and $\delta > 0$, and define the relaxed optimization problem

$$\begin{array}{ll} \underset{\mathbf{z}\in\mathbb{D},\hat{\mathbf{a}}\in\mathbb{R}^N}{\operatorname{argmin}} & \overline{G}(\hat{\mathbf{a}}) + \sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i) \\ \text{s.t.} & \chi(\mathbf{z}, \hat{\mathbf{a}}) = 0. \end{array} \tag{6.23}$$

The problem (6.23) is only an approximation of the original problem (6.22) and only in the case $\eta > 0$ and $\delta = 0$ the minimizers of the original problem are recovered. However, for the relaxed problem (6.23) convergence of Algorithm 9 can be shown if the stepsize is chosen appropriately. Before the corresponding result is given we introduce the notation $K : \mathbb{R}^{\mathcal{I}\times N} \times \mathbb{R}^N \to \mathbb{R}$,

$$K(\mathbf{z}, \hat{\mathbf{a}}) = \overline{G}(\hat{\mathbf{a}}) + \sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i)$$

for the overall objective function.

Theorem 6.3.1. *Let the functions $\overline{G} : \mathbb{R}^N \to \mathbb{R}$, $F_i : \mathbb{R}^N \to \mathbb{R}$, $i \in \mathbb{N}_{\mathcal{I}}$ and $\chi : \mathbb{R}^{\mathcal{I}\times N} \times \mathbb{R}^N \to \mathbb{R}^N$ be defined as*

$$\overline{G}(\hat{\mathbf{a}}) = \frac{\eta}{2}\left\|\hat{\mathbf{a}} - \mathbb{1}\hat{\zeta}\right\|^2, \qquad F_i(\mathbf{z}_i) = \frac{1}{\mathcal{I}}\frac{\delta}{2}\|\mathbf{z}_i\|^2 \quad \textit{and} \quad \chi(\mathbf{z}, \hat{\mathbf{a}}) = -\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} \mathbf{z}_i^T + \hat{\mathbf{a}}^T \tag{6.24}$$

for a given reference $\hat{\zeta} \in \mathbb{R}^N$ and given parameters $\eta > 0$ and $\delta > 0$. Furthermore, let the ascent direction d^ℓ in Algorithm 9 be defined as $d^\ell = \chi(\mathbf{z}^\ell, \hat{\mathbf{a}}^\ell)$. Additionally, assume that the sequence $(c^\ell)_{\ell\in\mathbb{N}}$ is defined such that

$$\varepsilon \leq c^\ell \leq (2 - \varepsilon)\frac{\min\left\{\frac{\delta}{\mathcal{I}}, \eta\right\}}{\frac{1}{\mathcal{I}} + 1}. \tag{6.25}$$

for a fixed sufficiently small $\varepsilon > 0$. Then the following holds:

(i) The sequence $(\mathbf{z}^\ell, \hat{\mathbf{a}}^\ell)_{\ell\in\mathbb{N}}$ converges to the unique optimal solution of the minimization problem (6.23), i.e., $(\mathbf{z}^\ell, \hat{\mathbf{a}}^\ell) \to (\mathbf{z}^\star, \hat{\mathbf{a}}^\star)$ for $\ell \to \infty$.

(ii) The sequence $(\mathbf{z}^\ell, \hat{\mathbf{a}}^\ell)_{\ell\in\mathbb{N}}$ approaches primal feasibility, i.e., $\chi(\mathbf{z}^\ell, \hat{\mathbf{a}}^\ell) \to 0$ for $\ell \to \infty$ and every accumulation point $\lambda^\star$ of $(\lambda^\ell)_{\ell\in\mathbb{N}}$ is an optimal solution of the dual problem.

Proof. We show that the overall objective function K is strongly convex and we compute the Lipschitz constant of the gradient of the dual function according to Theorem 6.2.6 from which the results follow. The Hessian of the function K is given by

$$\nabla^2 K(\mathbf{z}, \hat{\mathbf{a}}) = \begin{bmatrix} \frac{\delta}{\mathcal{I}} I & & & \\ & \ddots & & \\ & & \frac{\delta}{\mathcal{I}} I & \\ & & & \eta I \end{bmatrix}$$

and hence, K is strongly convex with parameter

$$\alpha = \min\left\{\frac{\delta}{\mathcal{I}}, \eta\right\} \tag{6.26}$$

with respect to Lemma 2.4.6. The strong convexity implies uniqueness of the primal solution $(\mathbf{z}^\star, \hat{\mathbf{a}}^\star)$ (cf. Theorem 2.4.9) and the existence of an optimal dual solution $\lambda^\star$ (see Theorem 6.1.1). For the constraints χ, we use the notation

$$A := \begin{bmatrix} -\frac{1}{\mathcal{I}} I & \cdots & -\frac{1}{\mathcal{I}} I & I \end{bmatrix}$$

to obtain

$$\chi(\mathbf{z}, \hat{\mathbf{a}}) = A \begin{pmatrix} \mathbf{z}_1^T \\ \vdots \\ \mathbf{z}_{\mathcal{I}}^T \\ \hat{\mathbf{a}}^T \end{pmatrix}.$$

Since $A \cdot A^T = \left(\frac{1}{\mathcal{I}} + 1\right) I$ holds, we obtain the norm $\|A\| = \sqrt{1 + \frac{1}{\mathcal{I}}}$ which provides the Lipschitz constant

$$L = \frac{\frac{1}{\mathcal{I}} + 1}{\min\left\{\frac{\delta}{\mathcal{I}}, \eta\right\}} \tag{6.27}$$

of the dual function ψ according to Theorem 6.2.6. The gradient of the dual function is given by $\nabla\psi(\lambda) = \chi(\mathbf{z}_\lambda, \hat{\mathbf{a}}_\lambda)$ where $(\mathbf{z}_\lambda, \hat{\mathbf{a}}_\lambda) \in \mathbb{D} \times \mathbb{R}^N$ denotes the unique minimizer of the optimization problem

$$(\mathbf{z}_\lambda, \hat{\mathbf{a}}_\lambda) = \operatorname*{argmin}_{(\mathbf{z}, \hat{\mathbf{a}}) \in \mathbb{D} \times \mathbb{R}^N} \mathcal{L}(\mathbf{z}, \hat{\mathbf{a}}, \lambda)$$

(see Theorem 6.2.2). The concavity of the dual function implies that every stationary point $\lambda^\star$ (i.e., $\psi(\lambda^\star) = \chi(\mathbf{z}_{\lambda^\star}, \hat{\mathbf{a}}_{\lambda^\star}) = 0$) of the sequence $(\lambda^\ell)_{\ell \in \mathbb{N}}$ is a maximum of the dual problem due to Lemma 2.4.5 (i). Furthermore, Theorem 2.4.9 (i) implies that $\lambda^\star$ corresponds to a global maximum. Finally, the saddle point Theorem 6.1.2 implies the convergence $(\mathbf{z}^\ell, \hat{\mathbf{a}}^\ell) \to (\mathbf{z}^\star, \hat{\mathbf{a}}^\star)$ for $\ell \to \infty$. This concludes the proof. □

Remark 6.3.2. *Note that the functions F_i, $i \in \mathbb{N}_{\mathcal{I}}$, can be defined in a different way. For example the functions*

$$F_i(\mathbf{z}_i) = \frac{1}{\mathcal{I}} \frac{\delta}{2} \|\mathbf{z}_i - \zeta_i \mathbb{1}\|^2,$$

with $\zeta_i = \frac{1}{N} \mathbf{s}_i \mathbb{1}^T$ similar to the decentralized control setting (cf. Equation (4.6)), lead to the same stepsize condition (6.25). If other strongly convex functions are used, the condition on the stepsize might change.

Theorem 6.3.1 guarantees convergence for every $\eta > 0, \delta > 0$ of the distributed dual ascent Algorithm 9 if the stepsize c^ℓ is chosen according to (6.25). Nevertheless, only for fixed $\eta > 0$ and $\delta \to 0$ (or equivalently $\eta \to \infty$ and $\delta > 0$ fixed) a solution of the original problem (6.21) is recovered. In Section 6.4.1, the implication of $\delta \to 0$ is investigated numerically.
To obtain a good approximation of the original problem, it is necessary to choose a small δ (if we additionally assume that $\eta = 1$ is fixed). Unfortunately, a small δ leads to a small Lipschitz constant L, and hence a small stepsize and slow convergence. This effect even increases with the number of RESs in the network since the term $\delta/\mathcal{I}$ in the definition of the Lipschitz constant (6.27) goes to zero. Even though the theoretical bound on the stepsize becomes quite small for large networks, numerically, larger stepsizes can often be used to obtain good results. If we use the residual

$$\mathbf{r}^\ell := -\hat{\mathbf{z}}^\ell + \hat{\mathbf{a}}^\ell \tag{6.28}$$

in the ℓ-th iteration of Algorithm 9, we can define an alternative stepsize rule which guarantees convergence.

Corollary 6.3.3. *If the stepsize $(c^\ell)_{\ell\in\mathbb{N}}$ in Algorithm 9 is defined as*

$$c^{\ell+1} = \begin{cases} c^\ell & \text{if } \left\|\mathbf{r}^{\ell+1}\right\| < \left\|\mathbf{r}^\ell\right\| \\ \max\left\{\frac{c^\ell}{2}, \frac{\min\left\{\frac{\delta}{\mathcal{I}},\eta\right\}}{\frac{1}{\mathcal{I}}+1}\right\} & \text{if } \left\|\mathbf{r}^{\ell+1}\right\| \geq \left\|\mathbf{r}^\ell\right\| \end{cases} \tag{6.29}$$

and $c^0 > 0$, then the convergence properties of Theorem 6.3.1 hold.[2]

Proof. The result follows immediately from Theorem 6.3.1 since the stepsize decreases until the condition (6.25) is satisfied for all following iterations ℓ. □

The stepsize rule of Corollary 6.3.3 is used in the numerical simulations in Section 6.4. In our setting, the stopping criterion is met in most of the cases with a stepsize

$$c^\ell \geq 2\frac{\min\left\{\frac{\delta}{\mathcal{I}},\eta\right\}}{\frac{1}{\mathcal{I}}+1},$$

i.e., the condition (6.25) is not satisfied, and less iterations are necessary to obtain a solution of the optimization problem (6.23).

Remark 6.3.4. *For the function $\overline{G}(\hat{\mathbf{a}}) = \frac{\eta}{2}\|\hat{\mathbf{a}} - \hat{\zeta}\mathbb{1}\|^2$, the update $\hat{\mathbf{a}}^{\ell+1}$ of the CE in Algorithm 9 can be computed explicitly as $\hat{\mathbf{a}}^{\ell+1} = \hat{\zeta}\mathbb{1} - \frac{1}{\eta}\lambda^\ell$.*

Remark 6.3.5. *The results in this section focus on the function $\overline{G}(\hat{\mathbf{a}}) = \|\hat{\mathbf{a}} - \mathbb{1}\hat{\zeta}\|^2$. However, we point out that similar results can be derived for arbitrary strongly convex functions $\overline{G}$ defined on convex and closed sets.*

[2] Although the stepsize rule is only heuristic, the stepsize rule of Corollary 6.3.3 significantly reduces the average number of iterations in our numerical simulations.

6.3.2 Price-based non-cooperative dual ascent application

Algorithms 5 and 6 in Chapter 5 implicitly include the assumption that the RESs and the CE cooperate to achieve a common goal. The distributed dual ascent algorithm 9 offers a different interpretation of the objective functions. The local optimization problem (6.17) of RES i

$$\mathbf{z}_i^{\ell+1} := \underset{\mathbf{z}_i \in \mathbb{D}_i}{\operatorname{argmin}} \left(F_i(\mathbf{z}_i) - \lambda^\ell \mathbf{z}_i^T \right)$$

solely depends on the variables $\mathbf{z}_i$ and the dual variables λ defined by the CE. The minimization problem does not depend on the variables of the other RESs. Thus, the dual ascent algorithm can be interpreted as a non-cooperative algorithm, where every RESs optimizes its own costs and the CE can influence the decisions of the RESs by means of the dual variable λ.
In this section, we give a non-cooperative interpretation of the dual ascent algorithm in the form of a negotiation process between the CE or the energy provider and the RESs. We show how the energy provider can reduce the fluctuations in the average power demand with real-time electricity price signals. Thus, the dual ascent algorithm corresponds to a negotiation of electricity prices until a price signal is found, which is accepted by the CE and the RESs.

Assumptions on the objective function

To be able to give an interpretation of electricity costs in Algorithm 9, we have to start with assumptions on the objective function. We describe the energy price at a certain time by a function $p : \mathbb{R} \to \mathbb{R}$ depending on the amount of energy used in the corresponding time interval. Furthermore, we assume that the following holds:

- The function p is increasing, i.e., a higher energy demand leads to higher costs, and
- if the power demand is 0 then also the costs have to be 0.

Linear costs

The simplest case is to consider linear costs, i.e., at a certain time the energy price can be described by the function

$$p(z; c) = Tcz$$

depending on the sampling time T and a constant $c \in \mathbb{R}_{\geq 0}$ chosen by the energy provider based on the average power demand $\hat{z}$. With respect to Algorithm 9 these costs are realized rewriting the original minimization problem (6.21) into the form

$$\begin{aligned} \min_{\mathbf{z} \in \mathbb{D},\, \hat{\mathbf{a}} \in \mathbb{R}^N} \quad & \left\| \hat{\mathbf{a}} - \mathbb{1}\hat{\zeta} \right\|^2 \\ \text{s.t.} \quad & \tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \mathbf{z}_i^T - \hat{\mathbf{a}}^T = 0. \end{aligned}$$

Indeed, this minimization problem implies local updates of the form

$$\begin{aligned}\hat{\mathbf{a}}^{\star\lambda} &= \operatorname*{argmin}_{\hat{\mathbf{a}}\in\mathbb{R}^N} \left(\left\| \hat{\mathbf{a}} - \mathbb{1}\hat{\zeta} \right\|^2 - \hat{\mathbf{a}}^T\lambda \right) \\ \mathbf{z}_i^{\star\lambda} &= \operatorname*{argmin}_{\mathbf{z}_i\in\mathbb{D}_i} \mathbf{z}_i\lambda^T\end{aligned}$$

(cf. Equations (6.17) and (6.18) in Algorithm 9) and we can identify $c(k) = \lambda(k)/T$ as the parameter of the CE defining the price for electricity at a fixed time instant k. As already argued for the minimization problem (6.23), in this case, the local functions F_i satisfy $F_i \equiv 0$ for all $i \in \mathbb{N}_{\mathcal{I}}$, which are convex but not strongly convex. Thus, the assumptions of Algorithm 9 are not satisfied so that convergence cannot be guaranteed. In fact, even if a diminishing stepsize is used in Algorithm 9, and the sequence $(\lambda^\ell)_{\ell\in\mathbb{N}}$ seems to converge, the sequence $(\hat{\mathbf{z}}^\ell)_{\ell\in\mathbb{N}}$ does not converge since already small changes in the vector λ imply that it is beneficial to use all the capacity of the battery at a time step $\lambda(j)$ instead of $\lambda(m)$ if $\lambda(j) > \lambda(m)$.
This behavior can also be observed in the Market Maker approach presented in [108], where the authors propose a price-based negotiation algorithm with linear cost functionals. Even though the algorithm improves the closed-loop behavior compared to a decentralized control scheme, convergence of the proposed approach could not be shown. As a remedy, we propose an approach where the electricity prices contain an additional penalty term.

Linear costs with quadratic penalty term

To circumvent the problems of linear costs, we propose linear costs with an additional quadratic penalty term

$$p(z;c) := T \cdot a \left(z + b\,(z-c)^2 - bc^2 \right), \tag{6.30}$$

for $a, b, c \in \mathbb{R}_{>0}$. The price still contains the linear term az, but in addition, the demand deviating from a given reference c is penalized. Either a power demand above the reference is penalized by additional costs, or the earning is reduced if too much energy is sold to the grid operator. In particular, $p(0;c) = 0$, i.e., zero demand does not incur any energy costs. Since, $p(\cdot;c)$ is not monotonically increasing on $\mathbb{R}$ the constants a and b have to be fixed by the CE such that $p(\cdot;c)$ is monotonically increasing on the domain of interest and higher demand leads to higher costs. This is always possible since both the net consumption profiles $(\mathbf{s}_i^\ell)_{\ell\in\mathbb{N}}$ and the battery capacities C_i , $i \in \mathbb{N}_{\mathcal{I}}$, are bounded.
Minimizing the electricity costs with respect to the function $p(\cdot;c)$ over the prediction horizon $N \in \mathbb{N}$ can be achieved by the minimization problem

$$\begin{aligned}\min_{\mathbf{z}\in\mathbb{D},\hat{\mathbf{a}}\in\mathbb{R}^N} \quad & \tfrac{\eta}{2} \left\| \hat{\mathbf{a}} - \mathbb{1}\hat{\zeta} \right\|^2 + \tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \left(\rho \mathbb{1} \mathbf{z}_i^T + \tfrac{\delta}{2} \left\| \mathbf{z}_i \right\|^2 \right) \\ \text{s.t.} \quad & -\tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \mathbf{z}_i^T + \hat{\mathbf{a}}^T = 0\end{aligned} \tag{6.31}$$

for $\eta, \rho, \delta \in \mathbb{R}_{>0}$. In the context of Algorithm 9, the optimization problems of the CE and of the RESs read

$$\hat{\mathbf{a}}^{\star_\lambda} = \underset{\hat{\mathbf{a}} \in \mathbb{R}^N}{\operatorname{argmin}} \left(\frac{\eta}{2} \left\| \hat{\mathbf{a}} - \mathbb{1}\hat{\zeta} \right\|^2 + \hat{\mathbf{a}}\lambda^T \right),$$
$$\mathbf{z}_i^{\star_\lambda} = \underset{\mathbf{z}_i \in \mathbb{D}_i}{\operatorname{argmin}} \left(\rho \mathbb{1}\mathbf{z}_i^T + \frac{\delta}{2} \|\mathbf{z}_i\|^2 - \mathbf{z}_i\lambda^T \right).$$

The functions $F_i(\mathbf{z}_i) = \frac{1}{\mathcal{I}} \left(\rho \mathbb{1}\mathbf{z}_i^T + \frac{\delta}{2} \|\mathbf{z}_i\|^2 \right)$, $i \in \mathbb{N}_\mathcal{I}$, are strongly convex due to the quadratic term[3] and hence, Algorithm 9 is applicable. More importantly, the objective function of RES $i \in \mathbb{N}_\mathcal{I}$ can be written in the form

$$\rho \mathbb{1}\mathbf{z}_i^T + \frac{\delta}{2} \|\mathbf{z}_i\|^2 - \mathbf{z}_i\lambda^T = \rho \mathbb{1}\mathbf{z}_i^T + \frac{\delta}{2} \left\| \mathbf{z}_i - \frac{1}{\delta}\lambda \right\|^2 - \frac{1}{2\delta}\lambda\lambda^T$$

providing the price interpretation $p(\cdot; c)$ by identifying the parameters

$$a = \frac{\rho}{T}, \quad b = \frac{1}{2} \cdot \frac{\delta}{T}, \quad c(k) = \frac{1}{\delta}\lambda(k). \tag{6.32}$$

The constant term $\frac{1}{2\delta}\lambda\lambda^T$ does not have any impact on the minimizer $\mathbf{z}_i^{\star_\lambda}$ and thus, it does not have to be considered in the minimization problem. In summary, minimization of the electricity costs over the prediction horizon N is equivalent to solving the minimization problem (6.17). The price contains the linear component Taz, but is also constructed such that deviations from a given reference value c are penalized. The latter means that a power demand above this reference incurs extra costs and that feeding in too much energy reduces the earning per unit. In Figure 6.1, the price for electricity is visualized for fixed parameters a and b.[4] If $\lambda(k)$ and consequently also $c(k)$ is small, energy is expensive and vice-versa. Figure 6.1 additionally shows that it is possibly to obtain prices, which are almost linear, but in contrast to linear prices, convergence of the dual ascent algorithm is guaranteed. Note that the parameters ρ and δ can be set individually for every RES, e.g. specific contracts depending on the respective demand profiles. In Section 6.4.2, numerical simulations investigating the pricing scheme are provided. In particular, the impact of batteries on the price signals c are investigated and the benefits on the electricity costs are discussed.

Remark 6.3.6. *Note that an additional linear term $\rho \mathbb{1}\mathbf{z}_i^T$ does not change the convexity properties of a function. Consequently, Theorem 6.3.1 also holds for $F_i(\mathbf{z}_i) = \rho \mathbb{1}^T\mathbf{z}_i + \frac{\delta}{2} \|\mathbf{z}_i\|^2$, $i \in \mathbb{N}_\mathcal{I}$.*

Remark 6.3.7. *The objective function $\overline{G}(\hat{\mathbf{a}}) = \|\hat{\mathbf{a}} - \mathbb{1}\hat{\zeta}\|^2$ can be replaced by any other strongly convex function. Also constraints on $\hat{\mathbf{a}} \in \mathbb{A}$ for a closed and convex set $\mathbb{A} \subset \mathbb{R}^N$ can be included as long as feasibility of the optimization problem is guaranteed.*

[3] In particular the functions are strongly convex with parameter $\alpha = \delta/\mathcal{I}$ according to the Lemma 2.4.6.

[4] The parameters chosen here are just for illustration. The value used for a can be scaled arbitrarily to obtain realistic energy prices.

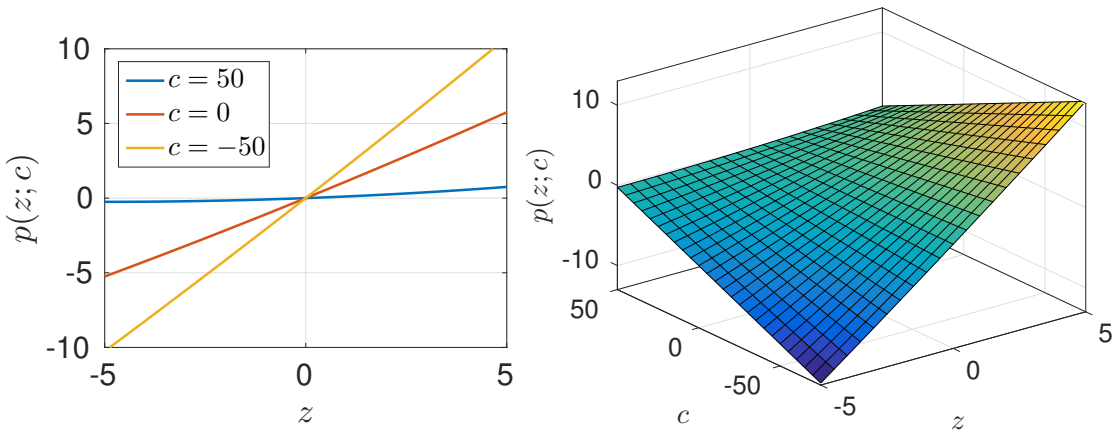

Figure 6.1: *Visualization of the electricity cost function $p(\cdot;c)$ (left) and $p(\cdot;\cdot)$ (right) from* (6.30) *for $a = 2.2$ and $b = 0.0091$ (corresponding to $\rho = 1.1$, $\delta = 0.02$ and $T = 0.5$). In the area of interest, the price is almost linear with respect to c and the demand z. The nonlinearity of the cost function can still be seen in $p(z;50)$ (left).*

6.3.3 General properties of the (non-)cooperative control setting

In this section, we highlight properties of the cooperative control setting, and the non-cooperative control setting using real-time prices. We discuss similarities and differences of both schemes and investigate how the performance (with respect to the performance metrics) deteriorates if a non-cooperative scheme is used instead of a cooperative scheme. The cooperative optimization Algorithm 6 introduced in Chapter 5 returns an optimal solution of the original minimization problem

$$\min_{\mathbf{z}_i \in \mathbb{D}_i} \quad \left\| \tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \mathbf{z}_i - \mathbb{1}\hat{\zeta} \right\|^2. \tag{6.33}$$

With the distributed dual ascent Algorithm 9, we obtain an approximation of the solution of (6.33) by solving the minimization problem

$$\begin{aligned} \min_{\mathbf{z}_i \in \mathbb{D}_i, \hat{\mathbf{a}} \in \mathbb{R}^N} \quad & \tfrac{\eta}{2} \left\| \hat{\mathbf{a}} - \mathbb{1}\hat{\zeta} \right\|^2 + \tfrac{1}{\mathcal{I}} \tfrac{\delta}{2} \textstyle\sum_{i=1}^{\mathcal{I}} \|\mathbf{z}_i\|^2 \\ \text{s.t.} \quad & -\tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \mathbf{z}_i^T + \hat{\mathbf{a}}^T = 0 \end{aligned} \tag{6.34}$$

for fixed $\eta, \delta \in \mathbb{R}_{>0}$ in a cooperative way. As already pointed out, for $\eta = 2$ and $\delta \to 0$, an optimal solution of the original problem is recovered. To obtain a price interpretation, we introduced the minimization problem

$$\begin{aligned} \min_{\mathbf{z}_i \in \mathbb{D}_i, \hat{\mathbf{a}} \in \mathbb{R}^N} \quad & \tfrac{\eta}{2} \left\| \hat{\mathbf{a}} - \mathbb{1}\hat{\zeta} \right\|^2 + \tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \left(\rho \mathbb{1}^T \mathbf{z}_i + \tfrac{\delta}{2} \|\mathbf{z}_i\|^2 \right) \\ \text{s.t.} \quad & -\tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \mathbf{z}_i^T + \hat{\mathbf{a}}^T = 0 \end{aligned} \tag{6.35}$$

with $\eta, \delta, \rho \in \mathbb{R}_{>0}$. This problem can be solved using a non-cooperative interpretation of Algorithm 9.

In this subsection, we concentrate on characteristic properties of the cooperative optimization problems (6.33) and (6.34), and the non-cooperative optimization problem (6.35). In particular, we concentrate on properties, which have to be considered in an MPC closed-loop implementation.

Implications of the monotonicity of electricity prices

According to our assumption, the price function $p(\cdot;c)$ is monotonically increasing, i.e., feeding in energy results in a profit in the non-cooperative setting. This implies that the battery of each RES is empty at the end of the prediction horizon,[5] i.e., $x_i(k+N)=0$ for all $i \in \mathbb{N}_{\mathcal{I}}$, whereas in the cooperative setting the state of charge of the batteries $x_i(k+N)$ for all $i \in \mathbb{N}_{\mathcal{I}}$ is determined by the optimization criteria. This is one characteristic, which distinguishes the optimal cooperative solution from the optimal non-cooperative solution. In the cooperative control setting, the additional constraints $x_i(k+N)=0$ can be included by using the constraints

$$\mathbb{D}_i^0 = \left\{ \mathbf{z}_i \in \mathbb{R}^N \;\middle|\; \begin{array}{l} x_i(k) = x_i^0,\ x_i(k+N)=0 \\ \text{System dynamics} \quad (6.19) \\ \text{Constraints} \quad (6.20) \\ \qquad \forall j = k, \ldots, k+N-1 \end{array} \right\}$$

instead of $\mathbb{D}_i$ for all $i \in \mathbb{N}_{\mathcal{I}}$ to ensure that the battery is empty at the end of the prediction horizon. If the state of charge of the battery at the end of the prediction horizon is known, then the energy used in the prediction window is known up to losses in the battery model and it can be used to simplify the objective function. To illustrate this property, we consider the simplified dynamics (6.19) without losses, i.e., $\alpha_i = \beta_i = \gamma_i = 1$ for all $i \in \mathbb{N}_{\mathcal{I}}$. Here, the empty battery at the end of the prediction horizon implies that the energy demand is constant for all feasible demand profiles $\mathbf{z}_i \in \mathbb{D}_i$, i.e.,

$$\mathbb{1}\mathbf{z}_i^T = \frac{x_i(k)}{T} + \sum_{j=k}^{k+N-1} s_i(j).$$

This allows for the following equivalent characterization of the set of minima in the cooperative setting:

$$\operatorname*{argmin}_{\mathbf{z}_i \in \mathbb{D}_i^0} \left\| \hat{\mathbf{z}} - \hat{\zeta}\mathbb{1} \right\|^2 = \operatorname*{argmin}_{\mathbf{z}_i \in \mathbb{D}_i^0} \|\hat{\mathbf{z}}\|^2 - 2\hat{\zeta}\mathbb{1}^T\hat{\mathbf{z}} + N\hat{\zeta}^2 = \operatorname*{argmin}_{\mathbf{z}_i \in \mathbb{D}_i^0} \|\hat{\mathbf{z}}\|^2 .$$

[5] We assume that the prediction horizon N and the maximal discharging rate $\underline{u}_i$, $i \in \mathbb{N}_{\mathcal{I}}$, are chosen such that the battery can always be discharged within the prediction horizon.

Analogously, for the non-cooperative problem one obtains

$$\begin{aligned}\min_{\mathbf{z}_i\in\mathbb{D}_i} \frac{\eta}{2}\left\|\hat{\mathbf{z}}-\mathbb{1}\hat{\zeta}\right\|^2+\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\left(\rho\mathbb{1}\mathbf{z}_i^T+\frac{\delta}{2}\left\|\mathbf{z}_i\right\|^2\right)\\ =\underset{\mathbf{z}_i\in\mathbb{D}_i^0}{\operatorname{argmin}}\quad \frac{\eta}{2}\left\|\hat{\mathbf{z}}\right\|^2+\eta\hat{\zeta}\mathbb{1}\hat{\mathbf{z}}^T+\frac{\eta}{2}N\hat{\zeta}^2+\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\left(\rho\mathbb{1}\mathbf{z}_i^T\right)+\frac{\delta}{2\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\left\|\mathbf{z}_i\right\|^2\\ =\underset{\mathbf{z}_i\in\mathbb{D}_i^0}{\operatorname{argmin}}\quad \frac{\eta}{2}\left\|\hat{\mathbf{z}}\right\|^2+\frac{\delta}{2\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\left\|\mathbf{z}_i\right\|^2.\end{aligned}$$

This implies that the set of optimal solutions is independent of the reference value $\hat{\zeta}$. In particular, no a priori knowledge of the average demand is needed in the case without losses in the battery model. Furthermore, observe that the constant ρ disappears in the objective function since it only appears in the linear term and does not have an impact on the minimizer. Nevertheless, ρ is necessary to obtain the price-interpretation and monotone electricity costs. We summarize these results in the following corollary.

Corollary 6.3.8. *Consider the system dynamics* (6.19) *subject to the constraints* (6.20) *and let* $\alpha_i=\beta_i=\gamma_i=1$ *for all* $i\in\mathbb{N}_{\mathcal{I}}$. *Additionally, let* $\eta,\rho,\gamma\in\mathbb{R}_{>0}$ *be fixed. Then the following equivalences hold.*

(i) The original cooperative minimization problem:

$$\underset{\mathbf{z}_i\in\mathbb{D}_i^0}{\textit{argmin}}\quad \left\|\tfrac{1}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}}\mathbf{z}_i-\mathbb{1}\hat{\zeta}\right\|^2 \qquad\Longleftrightarrow\qquad \underset{\mathbf{z}_i\in\mathbb{D}_i^0}{\textit{argmin}}\quad \left\|\tfrac{1}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}}\mathbf{z}_i\right\|^2.$$

(ii) The relaxed cooperative optimization problem:

$$\begin{aligned}&\underset{\mathbf{z}_i\in\mathbb{D}_i^0,\hat{\mathbf{a}}\in\mathbb{R}^N}{\textit{argmin}} && \tfrac{\eta}{2}\left\|\hat{\mathbf{a}}-\mathbb{1}\hat{\zeta}\right\|^2+\tfrac{1}{\mathcal{I}}\tfrac{\delta}{2}\textstyle\sum_{i=1}^{\mathcal{I}}\left\|\mathbf{z}_i\right\|^2\\ &\textit{s.t.} && -\tfrac{1}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}}\mathbf{z}_i^T+\hat{\mathbf{a}}^T=0\\ \Longleftrightarrow\quad &\underset{\mathbf{z}_i\in\mathbb{D}_i^0,\hat{\mathbf{a}}\in\mathbb{R}^N}{\textit{argmin}} && \tfrac{\eta}{2}\left\|\tfrac{1}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}}\mathbf{z}_i\right\|^2+\tfrac{1}{\mathcal{I}}\tfrac{\delta}{2}\textstyle\sum_{i=1}^{\mathcal{I}}\left\|\mathbf{z}_i\right\|^2\\ &\textit{s.t.} && -\tfrac{1}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}}\mathbf{z}_i^T+\hat{\mathbf{a}}^T=0.\end{aligned}$$

(iii) The non-cooperative optimization problem:

$$\begin{aligned}&\underset{\mathbf{z}_i\in\mathbb{D}_i,\hat{\mathbf{a}}\in\mathbb{R}^N}{\textit{argmin}} && \tfrac{\eta}{2}\left\|\hat{\mathbf{a}}-\mathbb{1}\hat{\zeta}\right\|^2+\tfrac{1}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}}\left(\rho\mathbb{1}^T\mathbf{z}_i+\tfrac{\delta}{2}\left\|\mathbf{z}_i\right\|^2\right)\\ &\textit{s.t.} && -\tfrac{1}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}}\mathbf{z}_i^T+\hat{\mathbf{a}}^T=0\\ \Longleftrightarrow\quad &\underset{\mathbf{z}_i\in\mathbb{D}_i^0,\hat{\mathbf{a}}\in\mathbb{R}^N}{\textit{argmin}} && \tfrac{\eta}{2}\left\|\tfrac{1}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}}\mathbf{z}_i\right\|^2+\tfrac{1}{\mathcal{I}}\tfrac{\delta}{2}\textstyle\sum_{i=1}^{\mathcal{I}}\left\|\mathbf{z}_i\right\|^2\\ &\textit{s.t.} && -\tfrac{1}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}}\mathbf{z}_i^T+\hat{\mathbf{a}}^T=0.\end{aligned}$$

The derivations show that the minimizer of the relaxed problem (6.34) only differs from the minimizer of the non-cooperative problem (6.35) because of the constraint $x_i(k+N)=0$ for all $i \in \mathbb{N}_{\mathcal{I}}$ independent of the parameter ρ (if a model without losses is considered). Moreover, if δ is chosen small, we can expect that a good approximation of a minimizer of the original problem (6.33) is obtained.
If losses are considered, the term $\mathbb{1}\mathbf{z}_i^T$ is not constant for all $\mathbf{z}_i \in \mathbb{D}_i^0$ anymore. Hence, for smaller constants β_i and γ_i, the impact of the linear term in the non-cooperative optimization problem increases (see Section 6.4.2).

The reference value $\hat{\zeta}$

The reference value $\hat{\zeta}(k) = \frac{1}{N}\hat{\mathbf{s}}(k;N)\mathbb{1}^T$ (see Equation (4.3)) at time instant k is computed by the CE based on the predicted power consumption $\mathbf{w}_i(k;N)$ and the predicted power generation $\mathbf{g}_i(k;N)$ of the RESs $i \in \mathbb{N}_{\mathcal{I}}$. In the cooperative setting we assume that the RESs report the correct parameters $\mathbf{s}_i = \mathbf{w}_i - \mathbf{g}_i$, $i \in \mathbb{N}_{\mathcal{I}}$ to the CE. In contrast to this, in the non-cooperative setting we have to assume that the RESs may send the wrong information to the CE to benefit from lower electricity prices. Thus, in the non-cooperative setting, $\hat{\zeta}$ has to be computed based on $\mathbf{z}$ instead of $\mathbf{s}$.
One way to do this, is to replace the reference value $\hat{\zeta}$ by $\frac{1}{N}\hat{\mathbf{z}}\mathbb{1}^T$ in the function $\overline{G}$ (and use that $\hat{\mathbf{z}} = \hat{\mathbf{a}}$).[6] In this case the function $\overline{G}$ reads

$$\begin{aligned}
\overline{G}(\hat{\mathbf{a}}) &= \frac{\eta}{2}\left\|\hat{\mathbf{a}} - \frac{1}{N}\left(\hat{\mathbf{a}}\mathbb{1}^T\right)\mathbb{1}\right\|^2 \\
&= \frac{\eta}{2}\left(\hat{\mathbf{a}}\hat{\mathbf{a}}^T - \frac{2}{N}\left(\hat{\mathbf{a}}\mathbb{1}^T\right)\hat{\mathbf{a}}\mathbb{1}^T + \frac{1}{N^2}\left(\hat{\mathbf{a}}\mathbb{1}^T\right)^2\mathbb{1}\mathbb{1}^T\right) \\
&= \frac{\eta}{2}\left(\hat{\mathbf{a}}\hat{\mathbf{a}}^T - \frac{2}{N}\left(\hat{\mathbf{a}}\mathbb{1}^T\right)\left(\mathbb{1}\hat{\mathbf{a}}^T\right) + \frac{N}{N^2}\left(\hat{\mathbf{a}}\mathbb{1}^T\right)\left(\mathbb{1}\hat{\mathbf{a}}^T\right)\right) \\
&= \frac{\eta}{2}\hat{\mathbf{a}}\left(I - \frac{1}{N}\left(\mathbb{1}^T\mathbb{1}\right)\right)\hat{\mathbf{a}}^T. \qquad (6.36)
\end{aligned}$$

The matrix $\left(\mathbb{1}^T\mathbb{1}\right)/N$ has the eigenvalues 1 with multiplicity one and 0 with multiplicity $N-1$. This implies that $I - \left(\mathbb{1}^T\mathbb{1}\right)/N$ has the eigenvalues 1 with multiplicity $N-1$ and 0 with multiplicity one, i.e., the Hessian $\nabla^2\overline{G}$ is not positive definite, $\overline{G}$ is not strongly convex and Algorithm 9 cannot be applied.
Since the function (6.36) cannot be used in the non-cooperative setting of Algorithm 9, we define the reference value $\hat{\zeta}$ depending on $\mathbf{z}^\ell$ in a different way. In the price-based non-cooperative control setting of Algorithm 9 the CE computes the reference value

$$\hat{\zeta}^\ell = \frac{1}{N}\hat{\mathbf{z}}^\ell\mathbb{1}^T$$

and solves the minimization problem

$$\hat{\mathbf{a}}^{\star\lambda} = \underset{\hat{\mathbf{a}}\in\mathbb{R}^N}{\operatorname{argmin}} \left(\frac{\eta}{2}\left\|\hat{\mathbf{a}} - \mathbb{1}\hat{\zeta}^\ell\right\|^2 + \hat{\mathbf{a}}^T\lambda\right)$$

[6] Note that in general $\hat{\mathbf{s}}\mathbb{1}^T \neq \hat{z}\mathbb{1}^T$ if losses in the battery model are considered.

in iteration $\ell \in \mathbb{N}$. The advantage of a changing average is that the RESs cannot send wrong information in the first iteration on their power demand $\mathbf{z}^0$ to manipulate the reference value $\hat{\zeta}$.

Warm-start and the stepsize selection

Similar to the algorithm introduced in Chapters 4 and 5, one can use warm-start (see Section 4.4.2) in the receding horizon context of Algorithm 9 to reduce the number of iterations at every time instant. Here, we use the optimal Lagrange variables at time instant k

$$\lambda_k^\star(k;N) = (\lambda_k^\star(k), \dots, \lambda_k^\star(k+N-2), \lambda_k^\star(k+N-1))$$

to initialize the Lagrange multipliers at time $k+1$ as

$$\lambda_{k+1}^0(k+1;N) := (\lambda_k^\star(k+1), \dots, \lambda_k^\star(k+N-1), \lambda_k^\star(k+N-1))\,.$$

With the choice $\lambda_{k+1}^0(k+N) = \lambda_k^\star(k+N-1)$ it is safeguarded that the price function $p(\cdot;\lambda_{k+1}^0(k+N)/\delta)$ is monotonically increasing on the domain of interest.
The stepsize at a fixed time instant k is selected according to Corollary 6.3.3. If we proceed from time instant k to $k+1$ we initialize c_{k+1}^0 by $c_{k+1}^0 = 2 \cdot c_k^{\text{end}}$ where c_k^{end} denotes the stepsize of Algorithm 9 at the iteration where the stopping criteria is satisfied. This makes sure that the stepsize does not only decrease but also increases again at the next time instant.

6.4 Numerical simulations

In this section, we examine the performance of the distributed dual ascent Algorithm 9. Throughout this section we consider a setting of $\mathcal{I} = 100$ RESs defined through the system dynamics (6.19) and the constraints (6.20). In the simulations using the minimization problem (6.31) (corresponding to the real-time prices (6.30)), we vary the number of RESs with a battery. For RESs with a battery, we choose the constraints $C_i = 4$, $\overline{u}_i = -\underline{u}_i = 1$ and the initial conditions $x_i(0) = 0$. (For the RESs without the battery we set $C_i = 0$.) Additionally, we vary the loss of energy from $\beta_i = \gamma_i = 1$ to $\beta_i = \gamma_i = 0.95$ to $\beta_i = \gamma_i = 0.9$ for all $i = 1, \dots, 100$, and investigate the impact of losses.

6.4.1 The impact of the relaxation parameter δ

The original optimization problem

$$\min_{\mathbf{z}_i \in \mathbb{D}_i} \quad \left\| \tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \mathbf{z}_i - \mathbb{1}\hat{\zeta} \right\|^2$$

(see the minimization problem (6.33)) considered in the preceding chapter by Algorithm 6, cannot be solved using the distributed dual ascent Algorithm 9 since the assumptions on

the objective function are not satisfied. Instead the relaxed problem

$$\begin{aligned} \min_{\mathbf{z}_i \in \mathbb{D}_i, \hat{\mathbf{a}} \in \mathbb{R}^N} \quad & \frac{\eta}{2} \left\| \hat{\mathbf{a}} - \mathbb{1}\hat{\zeta} \right\|^2 + \frac{1}{\mathcal{I}} \frac{\delta}{2} \sum_{i=1}^{\mathcal{I}} \|\mathbf{z}_i\|^2 \\ \text{s.t.} \quad & -\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i^T + \hat{\mathbf{a}}^T = 0 \end{aligned}$$

(see the optimization problem (6.34)) is solved by Algorithm 9 for fixed $\eta, \delta \in \mathbb{R}_{>0}$. In Figure 6.2, the open-loop solution for $\eta = 1$ and varying δ compared to the solution of the original optimization problem (6.33) is shown. For this simulation, every RESs is equipped with a battery without losses (i.e., $\beta_i = \gamma_i = 1$ for all $i \in \mathbb{N}_{\mathcal{I}}$). For a decreasing

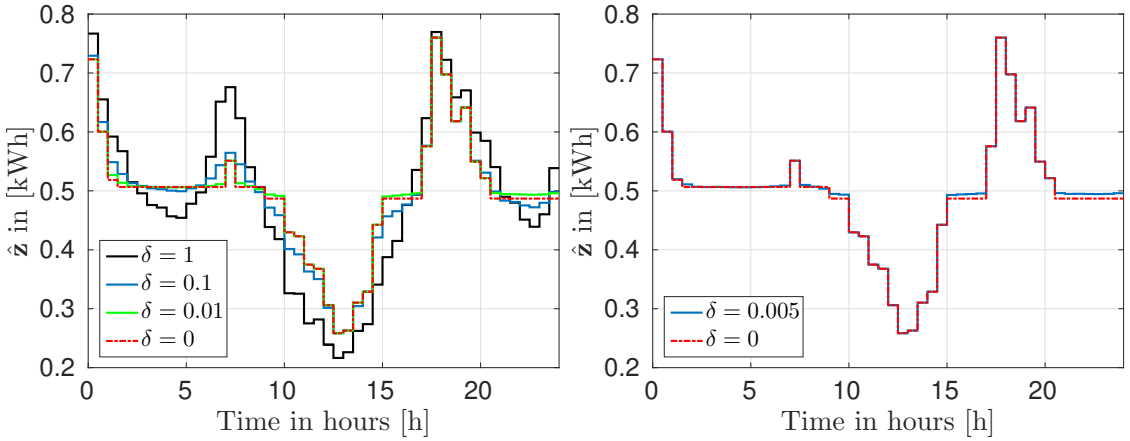

Figure 6.2: Comparison of the solution of the centralized optimization problem and the relaxed optimization problem for different relaxation parameters δ.

parameter δ the difference between the solution $\hat{\mathbf{z}}$ of the original optimization problem and the relaxed optimization problem is shrinking. For $\delta = 0.005$ the difference in the solutions is negligible (see Figure 6.2 (right)). The corresponding performance metrics are summarized in Table 6.1. Here, we observe that the parameter $\delta = 0.1$ is already small enough to obtain results such that the difference to the original problem formulation are insignificant. The same conclusions can be drawn for the setting with losses, i.e., $\beta_i \neq 1$ and $\gamma_i \neq 1$ for $i \in \mathbb{N}_{\mathcal{I}}$.

6.4.2 Price-based MPC simulations

In Figure 6.2 and in Table 6.1 we have seen that in the cooperative application of the distributed dual ascent Algorithm 9, the performance of the original problem using Algorithm 6 is recovered if the parameter δ is chosen small enough.
In this section we concentrate on the difference between the cooperative optimization Algorithm 6 and the non-cooperative application of Algorithm 9, i.e., we investigate the difference in the solution of the original optimization problem

$$\min_{\mathbf{z}_i \in \mathbb{D}_i} \quad \left\| \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i - \mathbb{1}\hat{\zeta} \right\|^2$$

	PTP	MQD	ASF
Uncontrolled	1.1016	0.0879	0.0093
$\delta = 0$	0.5016	0.0100	0.0025
Distributed dual ascent Algorithm 9			
$\delta = 1$	0.5531	0.0198	0.0028
$\delta = 0.1$	0.5016	0.0110	0.0022
$\delta = 0.01$	0.5016	0.0100	0.0024
$\delta = 0.005$	0.5016	0.0100	0.0025

Table 6.1: *Performance of the open-loop solution of the original problem ($\delta = 0$) compared to the solution using the dual ascent Algorithm 9 for different relaxations δ.*

(see the minimization problem (6.33)) and the price-based optimization problem

$$\begin{aligned} \min_{\mathbf{z}_i \in \mathbb{D}_i, \hat{\mathbf{a}} \in \mathbb{R}^N} \quad & \tfrac{\eta}{2} \left\| \hat{\mathbf{a}} - \mathbb{1}\hat{\zeta} \right\|^2 + \tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \left(\rho \mathbb{1}^T \mathbf{z}_i + \tfrac{\delta}{2} \left\| \mathbf{z}_i \right\|^2 \right) \\ \text{s.t.} \quad & -\tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \mathbf{z}_i + \hat{\mathbf{a}} = 0 \end{aligned}$$

(see the optimization problem (6.35)) with $\eta = 1$, $\rho = 1.1$ and $\delta = 0.02$. In particular, we examine the open-loop and the closed-loop solutions depending on the number of RESs with a battery. From the point of view of the energy provider or the CE, we consider the performance metrics, and from the point of view of an individual RES, we consider the energy prices and the savings.

Open-loop simulations

In Figure 6.3, we compare the solution of the original cooperative optimization problem with the solution of the non-cooperative problem for a different number of batteries in the overall system and different constants β_i and γ_i, $i \in \mathbb{N}_{100}$. We vary the number of batteries from 0 to 100 in steps of 10. The red line corresponds to the setting where every RESs has a battery.

We observe that for the case without losses, the solution of cooperative control and non-cooperative control only varies slightly due to the penalty term $\frac{\delta}{2}\|\mathbf{z}_i\|^2$ and due to the empty battery at the end of the prediction horizon in the non-cooperative setting. As argued in Section 6.3.3, the additional linear term in the cost functional does not influence the optimal power demand profile. Since $x_i(0) = 0$, the deviation from the average in the first 2 hours cannot be compensated.

While in the cooperative setting the performance does not deteriorate if losses are considered and if enough storage devices are in the network, in the non-cooperative setting the performance worsens with the loss of energy. If the losses are too big and the gradient in the corresponding electricity costs is too small, it does not pay off to store energy at one time instant to use it at a different time instant. Nevertheless, even if $\beta_i = \gamma_i = 0.9$ and hence 19% of the energy is lost, the deviation of the aggregated power demand from the average can be reduced significantly.

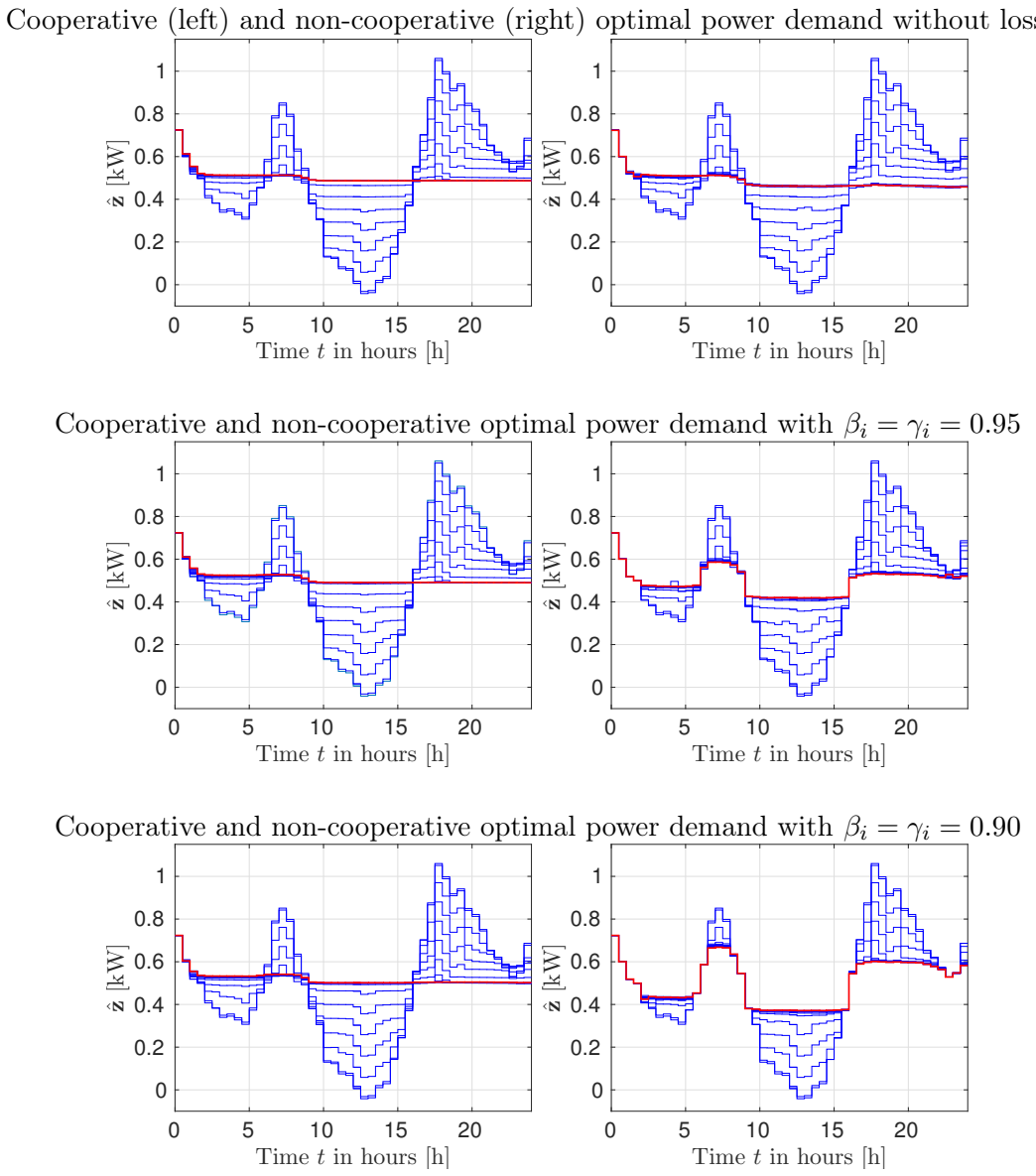

Figure 6.3: *Comparison of the cooperative and the non-cooperative solution of a single optimization problem with increasing number of batteries in the network (blue lines) and increasing losses (top to bottom). The red line indicates the setting where every RES owns a battery.*

Remark 6.4.1. *The deviation can be reduced even further if a bigger slope in the price function p with respect to the parameter c is considered (see Figure 6.1).*

The optimal dual variables $\lambda^\star$ corresponding to the non-cooperative settings in Figure 6.3 (right) are shown in Figure 6.4.
In Figure 6.5, the performance metrics of the non-cooperative open-loop solutions are illustrated. The usage of storage devices significantly improve the results — even if losses

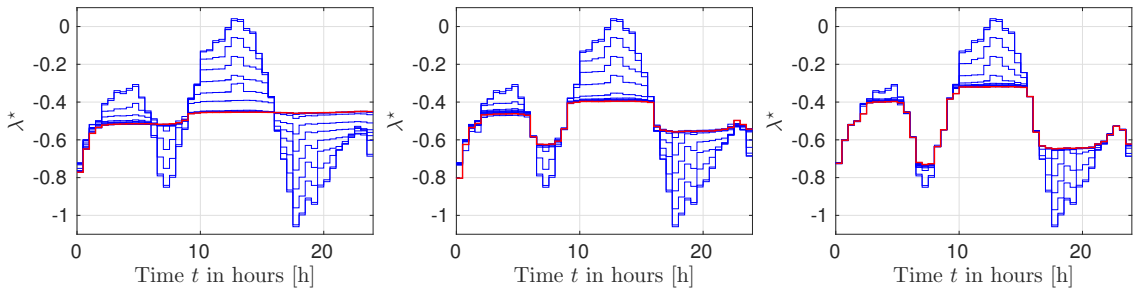

Figure 6.4: *Optimal dual variables $\lambda^\star$ of a single optimization problem corresponding to the case without losses (left), $\beta_i = \gamma_i = 0.95$ (middle) and $\beta_i = \gamma_i = 0.9$.*

are taken into account. If 60% of the RESs own a battery the results do not improve anymore. In the case were only a small number of RESs have a battery, the losses do not have a big impact on the performance.

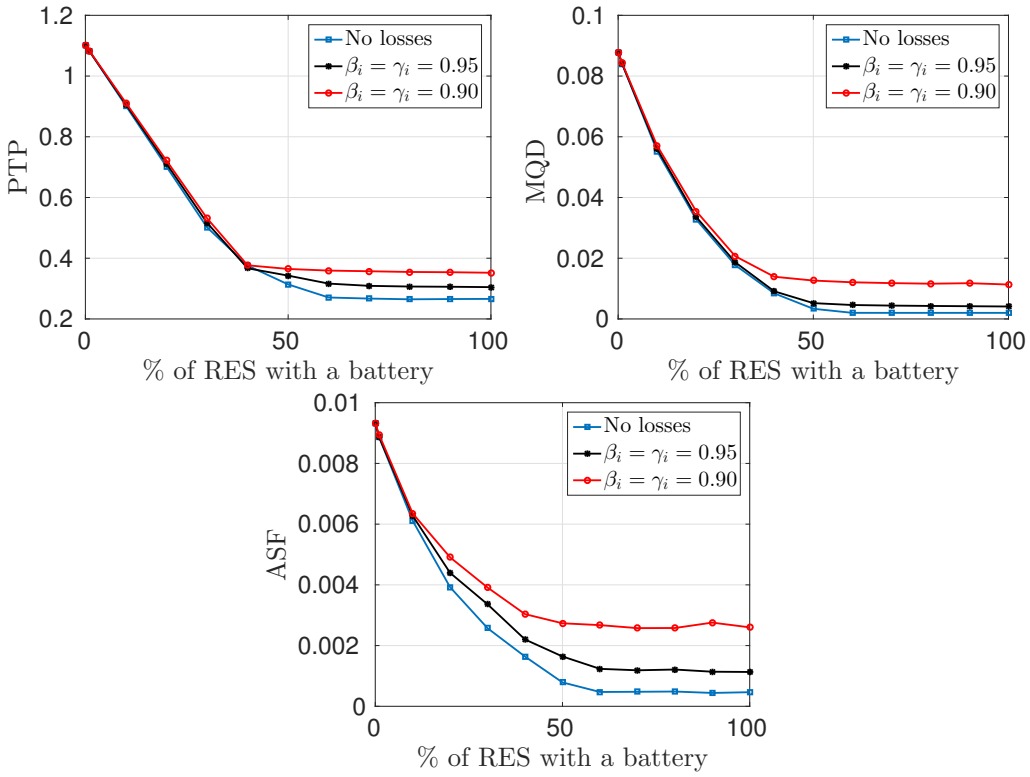

Figure 6.5: *Performance metrics of the non-cooperative open-loop solution. If a certain amount of RESs is equipped with a battery, the performance does not improve anymore.*

Not only the energy provider benefits from the smoothened aggregated power demand, the individual RESs can benefit as well from lower energy prices, if the prices which result

from the case without batteries are taken as a reference. In Figure 6.6 (left), the average electricity costs[7] for the RESs are shown depending on the number of RESs with a battery. Figure 6.6 (right) shows that, on average, the costs can be reduced up to 14% in the case without losses and up to 8% if $\gamma_i = \beta_i = 0.9$.

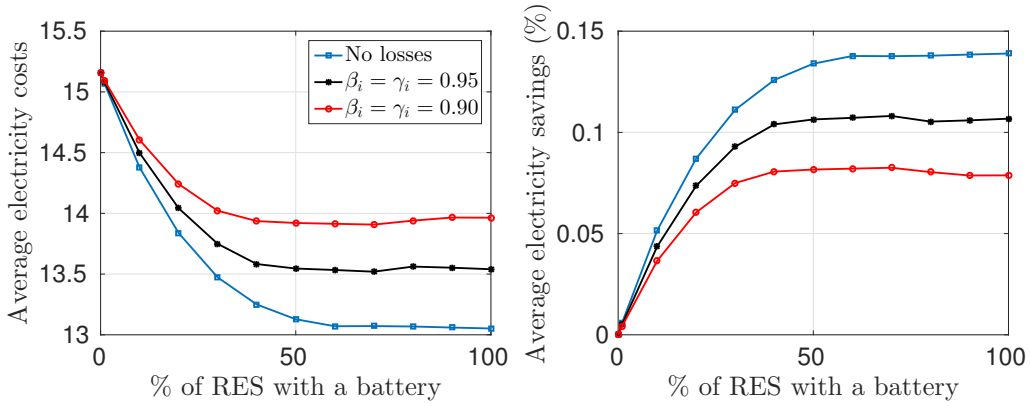

Figure 6.6: *Average open-loop electricity costs for every RES and savings with respect to the number of batteries in the network.*

Figure 6.7 shows the savings of the individual systems if 1%, 20%, 50% and 100% of the RESs own a battery. Not only the RESs with a battery reduce their costs, but also most of the RESs without a battery benefit if a significant amount of batteries is installed. If significantly many participants own a battery, the effect of single systems is reduced. This can be observed, for example, looking at RES 1. Moreover, we point out, that even though most of the RESs can reduce their costs, there exist single RESs with increased costs. Figure 6.7 corresponds to the setting without losses. The cases with loss of energy show a similar behavior.

Closed-loop simulations

Although we cannot prove that the open-loop results carry over to the closed loop, we use a simulation of $\mathcal{N} = 387$ time steps[8] to verify that the closed loop performs as expected. In Figure 6.8, the closed-loop solution for 100 RESs is visualized. In this setting only 50 RESs are equipped with batteries. The average over the state $\hat{\mathbf{x}}(0;\mathcal{N})$ and the input $\hat{\mathbf{u}}(0;\mathcal{N}) = \hat{\mathbf{u}}^+(0;\mathcal{N}) + \hat{\mathbf{u}}^-(0;\mathcal{N})$ is only with respect to the RESs with a battery. As one already expects from the open-loop control problem, the peaks in the average demand $\hat{\mathbf{z}}(0;\mathcal{N})$ are decreased significantly. If losses are considered in the battery model, then the battery capacities are not fully used since it does not pay off to buy and sell energy if the gradient in the cost functional is small.

[7] To obtain realistic energy prices, the numbers can be scaled arbitrarily in Figure 6.6 (left).

[8] The number of time steps is chosen such that $x_i(0) = x_i(\mathcal{N}) = 0$ for all $i \in \mathbb{N}_{\mathcal{I}}$. This is important to be able to calculate the overall savings of the individual systems. (To be able to pick $\mathcal{N}$ such that $x_i(\mathcal{N}) = 0$ holds for all $i \in \mathbb{N}_{\mathcal{I}}$, the closed-loop was simulated for a bigger number of time instants than $\mathcal{N} = 387$.)

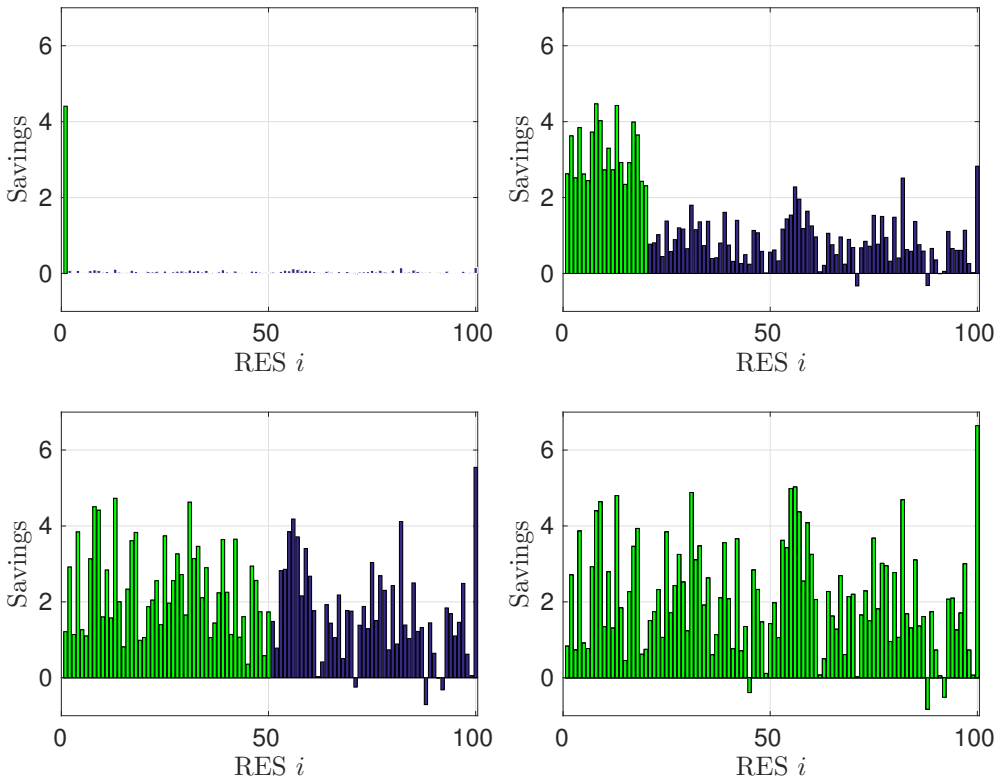

Figure 6.7: *Savings of the individual RESs. Green bars indicate the RESs with batteries and blue bars indicate the RESs without batteries. Observe that not all RESs benefit from batteries. (The results correspond to the setting without losses.)*

In Figures 6.9 and 6.10, the closed-loop counterparts of Figures 6.5 and 6.6 are presented. Again, we obtain the same qualitative results as in the open-loop setting. Nevertheless, in Figure 6.10 we observe that the average savings are significantly lower compared to the open-loop problem. We like to point out, that this occurs due to the high peaks on the first day of the simulation and is not a reason of the closed-loop simulation compared to the open-loop simulation in general.
The closed-loop counterpart of Figure 6.7 is given in Figure 6.11. We obtain the same qualitative behavior as in the open-loop case. The results from the point of a view of a single system, RES 1, are additionally summarized in Table 6.2. If RES 1 is the only RES with a storage device, the expected savings are high. The effect decreases with the increase of RESs equipped with storage devices.

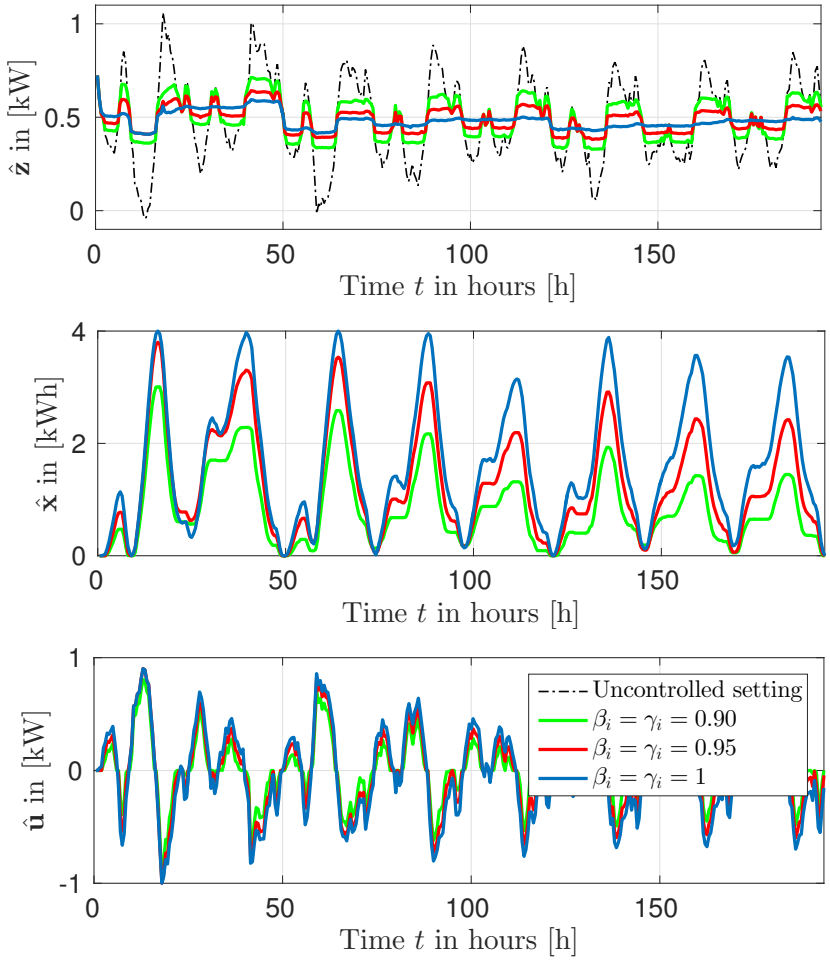

Figure 6.8: *Closed-loop trajectories of the non-cooperative control problem for different parameters β_i and γ_i for the case that 50% of the RESs are equipped with a battery.*

Average savings (left) and average savings in percent (right)

RES	1%	20%	50%	100%
1	23.69	11.69	4.74	3.49
$2-20$	0.10	13.02	7.27	6.26
$21-50$	0.12	2.77	6.35	5.25
$50-100$	0.08	2.97	4.91	5.89

RES	1%	20%	50%	100%
1	15.93%	7.87%	3.19%	2.34%
$2-20$	0.09%	11.67%	6.52%	5.61%
$21-50$	0.11%	2.59%	5.92%	4.90%
$50-100$	0.07%	2.52%	4.18%	5.01%

Table 6.2: *Average closed-loop savings of the groups of RESs with respect to the percentage of RESs with batteries in the system. (The results correspond to the setting without losses.)*

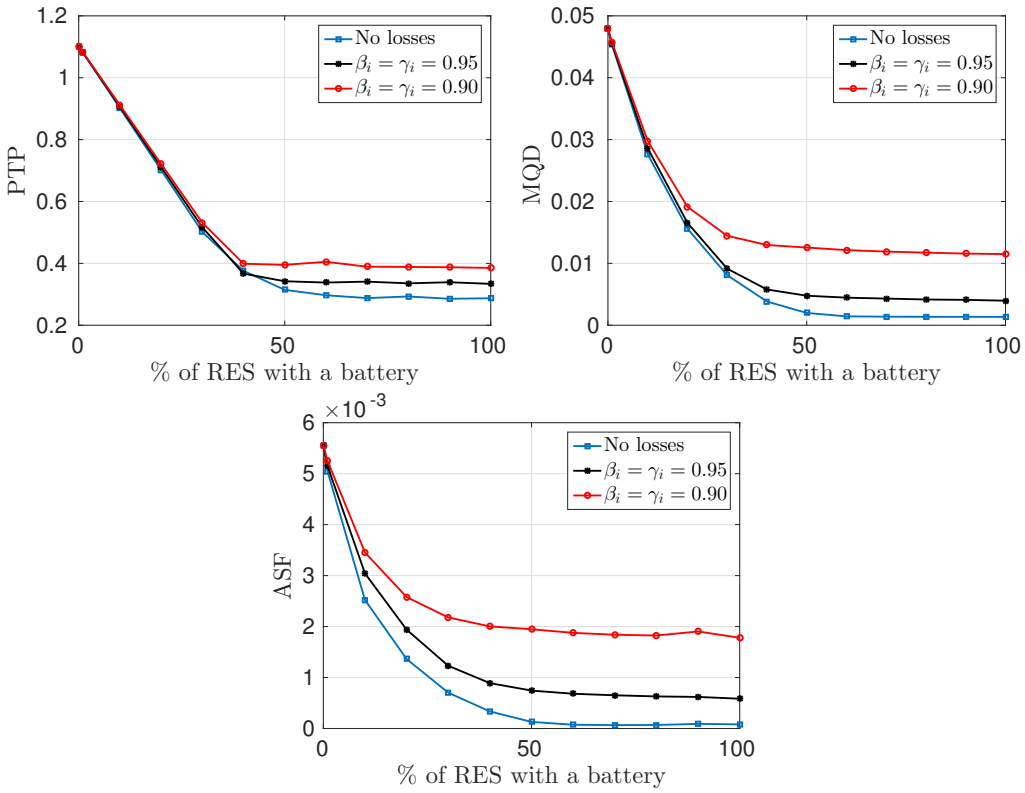

Figure 6.9: *Performance measures of the closed-loop control problem for a simulation of length* $\mathcal{N} = 387$ *with and without losses in the battery model.*

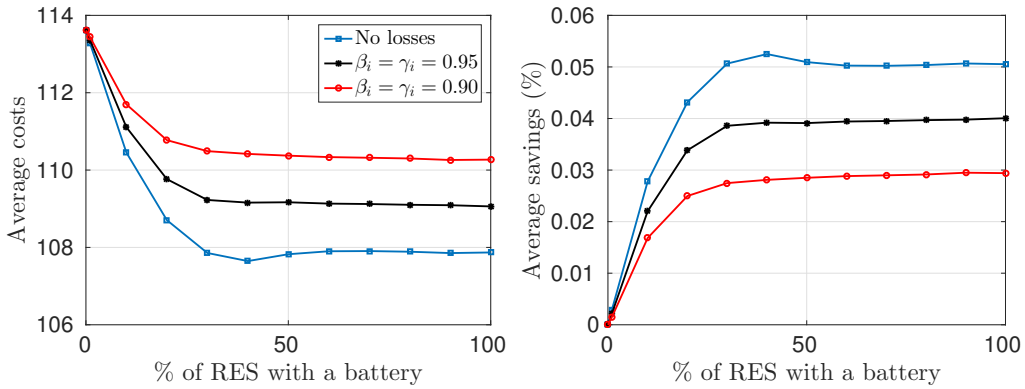

Figure 6.10: *Average closed-loop electricity costs for every RES and savings with respect to the number of installed batteries.*

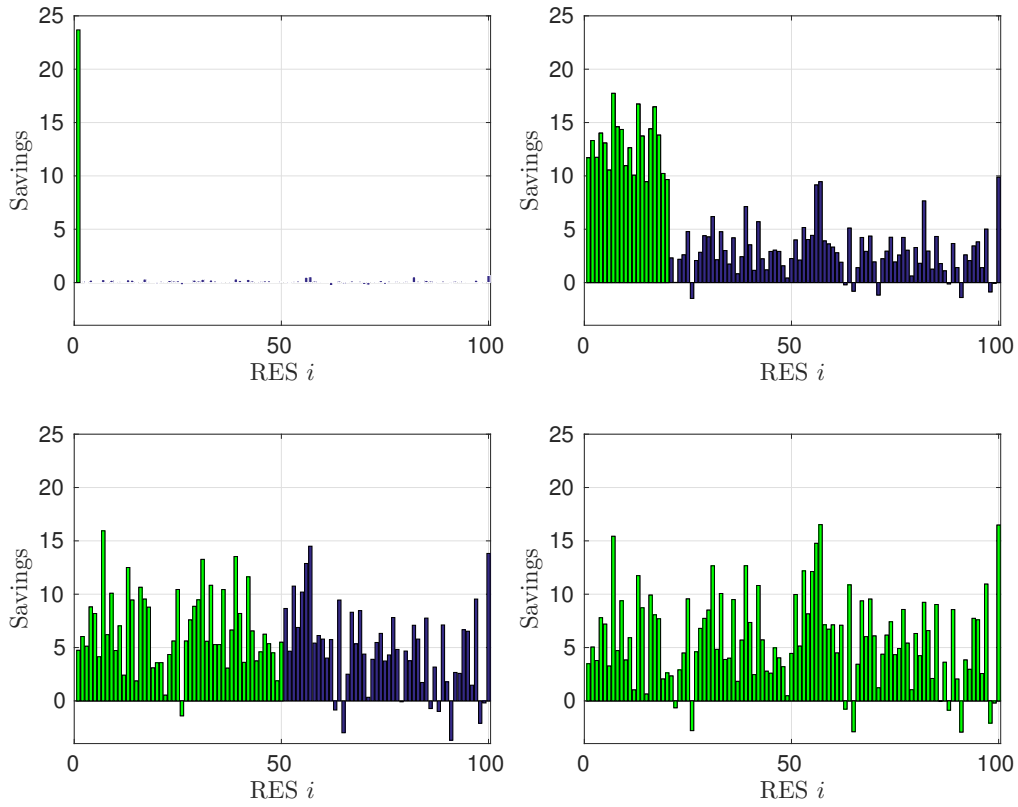

Figure 6.11: *Closed-loop savings of the individual RESs. Green bars indicate the RESs with batteries and blue bars indicate the RESs without batteries. Observe that not all RESs benefit from batteries. (The results correspond to the setting without losses.)*

Chapter 7

The alternating direction method of multipliers

In Chapter 5, we introduced a cooperative hierarchical distributed optimization algorithm which we used in a receding horizon scheme to obtain a DiMPC algorithm for a network of RESs. In Chapter 6, we explained how a relaxed version of the optimization problem considered in Chapter 5 can be solved using a distributed dual ascent algorithm. Furthermore, we gave a non-cooperative interpretation of the dual ascent algorithm using a real-time pricing scheme. However, with the dual ascent algorithm, only an approximation of the original optimization problem can be solved. Moreover, we have seen that the maximal stepsize of the dual ascent algorithm goes to zero if the number of RESs and/or the relaxation parameter δ goes to zero.

In this chapter, we introduce a different algorithm based on dual decomposition, the alternating direction method of multipliers (ADMM), which is applicable to the original minimization problem without the need of a strictly convex objective function. Here, we consider again a cooperative algorithm. But in contrast to Chapter 5, ADMM can handle additional coupling constraints among the RESs in the optimization problem, which offers additional possibilities in the problem formulation. The presentation of the ADMM approach is based on [16], but can largely also be found in [15].

Applications of ADMM in the context of electricity networks are discussed in [87] for example. In [87], ADMM is used to coordinate the charging of a fleet of electric vehicles connected to the grid. The authors introduce optimization problems for valley-filling, peak-shaving and an optimization problem to minimize the electricity costs for charging the fleet of electric vehicles. The authors only consider a single time instant $k \in \mathbb{N}$ and ADMM is not embedded in the receding horizon context. In this chapter, we investigate the closed-loop performance of DiMPC using ADMM and in particular demonstrate the flexibility in the objective function of the CE.

This chapter is structured as follows. In Section 7.1, the optimization problem under consideration is formulated and ADMM is introduced. Furthermore, a proof of convergence of the scheme is provided. In Section 7.2, we show how the algorithm can be applied to the network of RESs before we give explicit examples of applications in Section 7.3. The

chapter concludes with numerical simulations in Section 7.4.

7.1 ADMM: problem formulation and convergence results

In this section, we define the optimization problem under consideration throughout this chapter. Afterwards, necessary concepts for describing ADMM are introduced before a proof of convergence to an optimal solution is given. In the last part of the section, the original ADMM formulation is rewritten to obtain a simplification with respect to the communication structure of the algorithm.

7.1.1 Problem formulation

We consider a minimization problem of the form

$$\begin{aligned}(\mathbf{z}^\star, \sigma^\star) \quad \in \quad & \underset{(\mathbf{z},\sigma)}{\operatorname{argmin}} \quad \textstyle\sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i) + \overline{G}\left(\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} \mathbf{z}_i\right) + H(\sigma) \\ & \text{s.t.} \quad\quad A\tfrac{1}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}} \mathbf{z}_i^T + B\sigma^T - b = 0 \\ & \quad\quad\;\; \mathbf{z}_i \in \mathbb{D}_i \qquad \forall\, i = 1, \dots, \mathcal{I} \\ & \quad\quad\;\; \sigma \;\in \mathbb{S}\end{aligned} \tag{7.1}$$

where $F_i : \mathbb{R}^N \to \mathbb{R}$ denote objective functions of the individual RESs $i = 1, \dots, \mathcal{I}$, $\overline{G} : \mathbb{R}^N \to \mathbb{R}$ is the objective function coupling the individual RESs and $H : \mathbb{R}^M \to \mathbb{R}$ is an objective function of the CE in the artificial variable $\sigma \in \mathbb{R}^M$. We assume that a (possibility non-unique) minimizer $(\mathbf{z}^\star, \sigma^\star)$ of the minimization problem (7.1) exists. The sets $\mathbb{D}_i \subset \mathbb{R}^N$ for $i = 1, \dots, \mathcal{I}$ and $\mathbb{S} \subset \mathbb{R}^M$ are convex, closed and non-empty. The RESs are not only coupled through the objective function $\overline{G}$, but additionally through linear equality constraints defined by the matrices $A \in \mathbb{R}^{N\times q}$, $B \in \mathbb{R}^{M\times q}$ and the vector $b \in \mathbb{R}^q$.[1] As a first step to decouple the objective function, we introduce copies of the variables $\mathbf{a}_i = \mathbf{z}_i$, for $i = 1, \dots, \mathcal{I}$, and rewrite the minimization problem (7.1) as

$$\begin{aligned}(\mathbf{z}^\star, \mathbf{a}^\star, \sigma^\star) \quad \in \quad & \underset{(\mathbf{z}_i,\mathbf{a}_i,\sigma)}{\operatorname{argmin}} \quad \textstyle\sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i) + \overline{G}\left(\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} \mathbf{a}_i\right) + H(\sigma) \\ & \text{s.t.} \quad\quad A\tfrac{1}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}} \mathbf{a}_i^T + B\sigma^T - b^T = 0 \\ & \quad\quad\;\; \mathbf{z}_i \in \mathbb{D}_i \qquad \forall\, i = 1, \dots, \mathcal{I} \\ & \quad\quad\;\; \sigma \;\in \mathbb{S} \\ & \quad\quad\;\; \mathbf{z}_i - \mathbf{a}_i = 0 \quad \forall\, i = 1, \dots, \mathcal{I}.\end{aligned} \tag{7.2}$$

For the overall objective function, we use the notation

$$K(\mathbf{z}, \mathbf{a}, \sigma) = \sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i) + \overline{G}\left(\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} \mathbf{a}_i\right) + H(\sigma), \tag{7.3}$$

[1] Throughout this chapter we assume that z_i is one-dimensional for all $i \in \mathbb{N}_{\mathcal{I}}$ to simplify the notation. The extension of the results to the case $z_i \in \mathbb{R}^p$, $p \in \mathbb{N}$, is straightforward.

where $K : R^{\mathcal{I}\times N} \times R^{\mathcal{I}\times N} \times \mathbb{R}^M \to \mathbb{R}$, and we define $K^\star = K(\mathbf{z}^\star, \mathbf{a}^\star, \sigma^\star)$ as the optimal solution of problem (7.2). The variables $\mathbf{a} \in \mathbb{R}^{\mathcal{I}\times N}$ and $\sigma \in \mathbb{R}^M$ are assigned to the CE. We define the set

$$\mathbb{P} = \left\{ (\mathbf{a}, \sigma) \in \mathbb{R}^{\mathcal{I}\times N} \times \mathbb{S} \;\middle|\; A\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{a}_i^T + B\sigma^T - b^T = 0 \right\}$$

and again rewrite the minimization problem

$$\begin{aligned} (\mathbf{z}^\star, \mathbf{a}^\star, \sigma^\star) \quad \in \quad & \underset{(\mathbf{z}_i, \mathbf{a}_i, \sigma)}{\operatorname{argmin}} \quad \textstyle\sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i) + \overline{G}\left(\hat{\mathbf{a}}\right) + H(\sigma) \\ & \text{s.t.} \qquad (\mathbf{a}, \sigma) \in \mathbb{P} \\ & \qquad\quad \mathbf{z}_i \in \mathbb{D}_i \\ & \qquad\quad \mathbf{z}_i - \mathbf{a}_i = 0 \quad \forall i = 1, \ldots, \mathcal{I}, \end{aligned}$$

where the variables $(\mathbf{a}, \sigma)$ and $\mathbf{z}_i$ are only coupled through the constraints $\mathbf{z}_i - \mathbf{a}_i = 0$ for all $i \in N_{\mathcal{I}}$. The augmented Lagrangian $\mathcal{L}_\rho : \mathbb{D} \times \mathbb{P} \times \mathbb{R}^{\mathcal{I}\times N} \to \mathbb{R}$ of the minimization problem is defined as

$$\mathcal{L}_\rho(\mathbf{z}, \mathbf{a}, \sigma, \lambda) = \sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i) + \overline{G}\left(\hat{\mathbf{a}}\right) + H(\sigma) + \sum_{i=1}^{\mathcal{I}} \lambda_i (\mathbf{z}_i - \mathbf{a}_i)^T + \frac{\rho}{2} \sum_{i=1}^{\mathcal{I}} \|\mathbf{z}_i - \mathbf{a}_i\|^2$$

for $\rho > 0$. For $\rho = 0$, the usual definition of the Lagrangian from Equation (6.2) is recovered. Even if the functions F_i, $i \in \mathbb{N}_{\mathcal{I}}$ and $\overline{G}$ are not strongly convex, the term $\frac{\rho}{2} \sum_{i=1}^{\mathcal{I}} \|\mathbf{z}_i - \mathbf{a}_i\|^2$ ensures strong convexity in $\mathbf{z}$ and $\mathbf{a}$ of the augmented Lagrangian.

Remark 7.1.1. *The functions $\mathcal{L}_\rho(\cdot, \mathbf{a}, \sigma, \lambda)$ and $\mathcal{L}_\rho(\mathbf{z}, \cdot, \sigma, \lambda)$ are strongly convex for $\rho > 0$ if the functions F_i, $i \in \mathbb{N}_{\mathcal{I}}$, the function $\overline{G}$ and the function H are convex.*

Equivalently, the augmented Lagrangian can be written in the variables $\nu := \lambda/\rho$ as

$$\mathcal{L}_\rho(\mathbf{z}, \mathbf{a}, \sigma, \nu) = \sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i) + \overline{G}\left(\hat{\mathbf{a}}\right) + H(\sigma) + \frac{\rho}{2} \sum_{i=1}^{\mathcal{I}} \|\mathbf{z}_i - \mathbf{a}_i + \nu_i\|^2 - \frac{\rho}{2} \sum_{i=1}^{\mathcal{I}} \nu_i \nu_i^T.$$

With these definitions one iteration of the ADMM algorithm consists of the three steps

$$\mathbf{z}_i^{\ell+1} \in \underset{\mathbf{z}_i \in \mathbb{D}_i}{\operatorname{argmin}} \; \mathcal{L}_\rho(\mathbf{z}, \mathbf{a}^\ell, \sigma^\ell, \nu^\ell) \quad \text{for all } i = 1, \ldots, \mathcal{I}, \tag{7.4a}$$

$$(\mathbf{a}^{\ell+1}, \sigma^{\ell+1}) \in \underset{(\mathbf{a}, \sigma) \in \mathbb{P}}{\operatorname{argmin}} \; \mathcal{L}_\rho(\mathbf{z}^{\ell+1}, \mathbf{a}, \sigma, \nu^\ell), \tag{7.4b}$$

$$\nu_i^{\ell+1} = \nu_i^\ell + \left(\mathbf{z}_i^{\ell+1} - \mathbf{a}_i^{\ell+1}\right) \qquad \text{for all } i = 1, \ldots, \mathcal{I}, \tag{7.4c}$$

which are repeated iteratively for $\ell \in \mathbb{N}$ similar to the dual ascent Algorithm 8. The first step, minimization problem (7.4a), is solved by the RESs in parallel. The minimization

problem (7.4b) and the update (7.4c) are executed by the CE. Using the definition of the Lagrangian and omitting the constant terms, the updates read

$$\mathbf{z}_i^{\ell+1} \in \underset{\mathbf{z}_i \in \mathbb{D}_i}{\operatorname{argmin}} \; F_i(\mathbf{z}_i) + \frac{\rho}{2} \left\| \mathbf{z}_i - \mathbf{a}_i^{\ell} + \nu_i^{\ell} \right\|^2 \qquad \text{for all } i = 1, \ldots, \mathcal{I}, \tag{7.5a}$$

$$(\mathbf{a}^{\ell+1}, \sigma^{\ell+1}) \in \underset{(\mathbf{a},\sigma) \in \mathbb{P}}{\operatorname{argmin}} \; \overline{G}\,(\hat{\mathbf{a}}) + H(\sigma) + \frac{\rho}{2} \sum_{i=1}^{\mathcal{I}} \left\| \mathbf{z}_i^{\ell+1} - \mathbf{a}_i + \nu_i^{\ell} \right\|^2, \tag{7.5b}$$

$$\nu_i^{\ell+1} = \nu_i^{\ell} + \left(\mathbf{z}_i^{\ell+1} - \mathbf{a}_i^{\ell+1} \right) \qquad \text{for all } i = 1, \ldots, \mathcal{I}. \tag{7.5c}$$

The ADMM formulation (7.5) simplifies the original problem,[2] but the complexity of the optimization problem (7.5b) of the CE increases with the number of RESs in the network. In other words, the complexity of the centralized control setting has only been shifted to the CE. Before we simplify the ADMM formulation by reducing the number of variables, we prove convergence of ADMM towards an optimal solution of the original problem (7.1) in the general form (7.5).

7.1.2 Definitions and notations

Before we provide a convergence result of ADMM, we give several definitions which are necessary for the proof.

Definition 7.1.2 ([36], Definition 5)**.** *A function $\phi : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is called proper if it is not everywhere ∞ that is, there exists an $\mathbf{y} \in \mathbb{R}^n$ such that $\phi(\mathbf{y}) \in \mathbb{R}$. Such a function is called closed if the epigraph*

$$\boldsymbol{epi}(\phi) := \{(\mathbf{y}, t) \in \mathbb{R}^n \times \mathbb{R} | \phi(\mathbf{y}) \leq t\}$$

is closed.

For the problems considered in this thesis, we can use the following connection between the epigraph of a function ϕ and convexity of ϕ.

Remark 7.1.3 ([17])**.** *A function is convex if and only if its epigraph is a convex set.*

Additionally, we need the definition of a subderivative.

Definition 7.1.4. *Let $\phi : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ be an extended real-valued function. A vector $\overline{\mathbf{y}} \in \mathbb{R}^n$ is called a subderivative of ϕ at $\mathbf{y}^\star \in \mathbb{R}^n$, if*

$$\overline{\mathbf{y}}(\mathbf{y} - \mathbf{y}^\star)^T \leq \phi(\mathbf{y}) - \phi(\mathbf{y}^\star)$$

for all $\mathbf{y} \in \mathbb{R}^n$. The set of subdifferentials of ϕ at $\mathbf{y}^\star$ is denoted by

$$\partial\phi(\mathbf{y}^\star) := \{\overline{\mathbf{y}} \in \mathbb{R}^n | \overline{\mathbf{y}}(\mathbf{y} - \mathbf{y}^\star)^T \leq \phi(\mathbf{y}) - \phi(\mathbf{y}^\star),\ \forall \mathbf{y} \in \mathbb{R}^n\}.$$

[2] Observe that (7.5b) is an unconstrained minimization problem in the variables $\mathbf{a} \in \mathbb{R}^{\mathcal{I} \times N}$, whereas the original problem is a constrained optimization problem in the variables $\mathbf{z} \in \mathbb{R}^{\mathcal{I} \times N}$.

Remark 7.1.5. *For a convex differentiable function $\phi : D \to \mathbb{R}$ ($D \subset \mathbb{R}^n$ open) the subdifferential and the derivative coincide, i.e.,*

$$\{D\phi(\mathbf{y})\} = \partial\phi(\mathbf{y}).$$

The subdifferentials give a simple interpretation of a global minimizer $\mathbf{y}^\star$ of a function ϕ through the inequality

$$0 = 0(\mathbf{y} - \mathbf{y}^\star)^T \leq \phi(\mathbf{y}) - \phi(\mathbf{y}^\star)$$

for all $\mathbf{y} \in \mathbb{R}^n$. Hence, we can state the following remark.

Remark 7.1.6. *$\mathbf{y}^\star$ is a minimizer of the convex function ϕ if and only if $0 \in \partial\phi(\mathbf{y}^\star)$.*

The last result we need is a linearity result on subdifferentials for which we have to define the domain of a function and the relative interior of convex sets.

Definition 7.1.7. *Let $\phi : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ be an extended real-valued function. We define the domain of the function ϕ as*

$$dom(\phi) := \{\mathbf{y} \in \mathbb{R}^n | \phi(\mathbf{y}) < \infty\}.$$

Definition 7.1.8 (Relative interior of convex sets)**.** *For a convex set $D \subset \mathbb{R}^n$, we define the relative interior of D*

$$ri(D) := \{\mathbf{y} \in D | \forall \overline{\mathbf{y}} \in D \; \exists \lambda > 1 : \lambda\mathbf{y} + (1-\lambda)\overline{\mathbf{y}} \in D\}.$$

Theorem 7.1.9 ([88], Theorem 23.8)**.** *Let $\phi_i : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, $i = 1, \ldots, \mathcal{I}$, be proper convex functions and let*

$$\phi = \phi_1 + \ldots + \phi_{\mathcal{I}}.$$

If $\cap_{i=1}^m ri(dom(\phi_i)) \neq \emptyset$, then

$$\partial\phi(\mathbf{y}) = \partial\phi_1(\mathbf{y}) + \ldots + \partial\phi_{\mathcal{I}}(\mathbf{y})$$

for all $\mathbf{y} \in \mathbb{R}^n$.

7.1.3 Convergence of the alternating direction of multipliers method

With the definitions given in the preceding section, we show convergence of ADMM using the updates (7.5) under suitable assumptions on the functions F_i, $i \in \mathbb{N}_{\mathcal{I}}$, $\overline{G}$ and H and on the sets $\mathbb{P}$ and $\mathbb{D}_i$, $i \in \mathbb{N}_{\mathcal{I}}$.

Assumption 7.1.10 ([16], Assumption 1 and 2)**.** *Suppose the following conditions hold:*

(i) The functions $F_i : \mathbb{R}^N \to \mathbb{R} \cup \{\infty\}$, $i = 1, \ldots, \mathcal{I}$, $\overline{G} : \mathbb{R}^N \to \mathbb{R} \cup \{\infty\}$ and $H : \mathbb{R}^m \to \mathbb{R} \cup \{\infty\}$ are closed, proper, and convex.

(ii) The unaugmented Lagrangian $\mathcal{L}_0$ has a saddle point, i.e., there exist $(\mathbf{z}^\star, \mathbf{a}^\star, \sigma^\star, \lambda^\star)$, not necessarily unique, for which

$$\mathcal{L}_0(\mathbf{z}^\star, \mathbf{a}^\star, \sigma^\star, \lambda) \leq \mathcal{L}_0(\mathbf{z}^\star, \mathbf{a}^\star, \sigma^\star, \lambda^\star) \leq \mathcal{L}_0(\mathbf{z}, \mathbf{a}, \sigma, \lambda^\star) \tag{7.6}$$

holds for all $\mathbf{z}, \mathbf{a} \in \mathbb{R}^{\mathcal{I}\times N}$, $\sigma \in \mathbb{R}^m$ and $\lambda \in \mathbb{R}^{\mathcal{I}\times N}$.

Remark 7.1.11. *Convex functions subject to non-empty, convex and compact constraints satisfy Assumption 7.1.10, for example. If the functions F_i, $i \in \mathbb{N}_{\mathcal{I}}$, $\overline{G}$ and H are convex, then the function K, defined in Equation (7.3), is convex as well. Since convex functions attain their minimum on compact sets, there exists a (possibly non-unique) primal optimal solution $(\mathbf{z}^\star, \mathbf{a}^\star, \sigma^\star)$. This implies the existence of an optimal dual solution $\lambda^\star$ satisfying the saddle point condition (ii) (see Theorem 6.1.2 and Theorem 6.1.1).*

Theorem 7.1.12 ([16], Chapter 3)**.** *Under the Assumptions 7.1.10, the ADMM iterates satisfy the following:*

- *The residual $\mathbf{r}^\ell := \mathbf{z}^\ell - \mathbf{a}^\ell$ converges to zero for $\ell \to \infty$, i.e., $\mathbf{r}^\ell \to 0$ for $\ell \to \infty$ and the iterates $\mathbf{z}^\ell$ and $\mathbf{a}^\ell$ approach feasibility.*
- *The sequence $(K(\mathbf{z}^\ell, \mathbf{a}^\ell, \sigma^\ell))_{\ell\in\mathbb{N}}$ converges to the optimal value $K^\star$ of problem (7.1) for $\ell \to \infty$.*
- *The dual variables λ^ℓ converge to the optimal dual point $\lambda^\star$ for $\ell \to \infty$.*

Proof. We give the proof presented in [16] adapted to our notation and setting.
We define the function $L : \mathbb{R}^{\mathcal{I}\times N} \times \mathbb{R}^{\mathcal{I}\times N} \to \mathbb{R}$,

$$L(\mathbf{a}, \lambda) = \rho \sum_{i=1}^{\mathcal{I}} \|\mathbf{a}_i - \mathbf{a}_i^\star\|^2 + \frac{1}{\rho} \sum_{i=1}^{\mathcal{I}} \|\lambda_i - \lambda_i^\star\|^2,$$

penalizing the deviation from the unknown optimal primal and dual variables $\mathbf{a}^\star$ and $\lambda^\star$. We show, by using Lyapunov-like arguments, that L satisfies

$$L^{\ell+1} = L(\mathbf{a}^{\ell+1}, \lambda^{\ell+1}) \leq L(\mathbf{a}^\ell, \lambda^\ell) = L^\ell$$

and thus, convergence of the sequences $(\mathbf{a}^\ell)_{\ell\in\mathbb{N}}$ and $(\lambda^\ell)_{\ell\in\mathbb{N}}$ can be obtained. Assume that the following inequalities hold:

$$L^{\ell+1} - L^\ell \leq -\frac{1}{\rho} \sum_{i=1}^{\mathcal{I}} \|\lambda_i^{\ell+1} - \lambda_i^\ell\|^2 - \rho \sum_{i=1}^{\mathcal{I}} \|\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^\ell\|^2, \tag{7.7a}$$

$$K^{\ell+1} - K^\star \leq \sum_{i=1}^{\mathcal{I}} \left(\rho \left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^\ell \right) \left(\mathbf{a}_i^\star - \mathbf{a}_i^{\ell+1} - \mathbf{r}_i^{\ell+1} \right)^T - \lambda_i^{\ell+1} \mathbf{r}_i^{\ell+1^T} \right), \tag{7.7b}$$

$$K^\star - K^{\ell+1} \leq \sum_{i=1}^{\mathcal{I}} \lambda_i^{\star^T} \mathbf{r}_i^{\ell+1}. \tag{7.7c}$$

Then inequality (7.7a) guarantees a decrease of L^ℓ in every iteration if $(\mathbf{a}^{\ell+1}, \lambda^{\ell+1}) \neq (\mathbf{a}^\ell, \lambda^\ell)$ is satisfied. Since $L^\ell \leq L^0$, from the definition of L, it follows that $(\lambda^\ell)_{\ell\in\mathbb{N}}$ and $(\mathbf{a}^\ell)_{\ell\in\mathbb{N}}$ are bounded sequences. Summing up the Inequalities (7.7a) over $\ell \in \mathbb{N}$ leads to

$$L^0 \geq \sum_{\ell=0}^{\infty} \left(\frac{1}{\rho} \sum_{i=1}^{\mathcal{I}} \|\lambda_i^{\ell+1} - \lambda_i^\ell\|^2 + \rho \sum_{i=1}^{\mathcal{I}} \|\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^\ell\|^2 \right), \tag{7.8}$$

which implies that $(\lambda^\ell)_{\ell\in\mathbb{N}}$ and $(\mathbf{a}^\ell)_{\ell\in\mathbb{N}}$ are Cauchy sequences and in particular, there exists $\lambda^\sharp \in \mathbb{R}^{\mathcal{I}\times N}$, such that $\lambda^\ell \to \lambda^\sharp$ holds for $\ell \to \infty$. Since $\lambda^\sharp$ is a stationary point of the ADMM iterations (7.5), it follows from the saddle point condition (7.6) that $\lambda^\sharp = \lambda^\star$.
Inequality (7.8) additionally implies the convergence

$$\mathbf{r}_i^{\ell+1} = \frac{1}{\rho}\left(\lambda_i^{\ell+1} - \lambda_i^\ell\right) \to 0, \qquad \mathbf{a}_i^{\ell+1} - \mathbf{a}_i^\ell \to 0 \tag{7.9}$$

for $\ell \to \infty$ for all $i \in \mathbb{N}_{\mathcal{I}}$. Applying these results to the Inequalities (7.7b) and (7.7c) leads to

$$\left| K^\ell - K^\star \right| \to 0 \tag{7.10}$$

for $\ell \to \infty$. From Equation (7.10), we obtain the convergence of the objective function.
It remains to show that the Inequalities (7.7a) to (7.7c) hold. Inequality (7.7c) can be derived from Assumption 7.1.10 (ii). It holds that

$$\begin{aligned} K^\star &= \mathcal{L}_0(\mathbf{z}^\star, \mathbf{a}^\star, \sigma^\star, \lambda^\star) \\ &\leq \mathcal{L}_0(\mathbf{z}^{\ell+1}, \mathbf{a}^{\ell+1}, \sigma^{\ell+1}, \lambda^\star) \\ &= \sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i^{\ell+1}) + \overline{G}\left(\hat{\mathbf{a}}^{\ell+1}\right) + H(\sigma^{\ell+1}) + \sum_{i=1}^{\mathcal{I}} \lambda_i^\star (\mathbf{z}_i^{\ell+1} - \mathbf{a}_i^{\ell+1})^T \\ &= K^{\ell+1} + \sum_{i=1}^{\mathcal{I}} \lambda_i^\star \left(\mathbf{r}_i^{\ell+1}\right)^T. \end{aligned}$$

To show inequality (7.7b), we use Equation (7.4a) defining the $\mathbf{z}_i$ update

$$\mathbf{z}_i^{\ell+1} = \underset{\mathbf{z}_i \in \mathbb{D}_i}{\operatorname{argmin}}\ \mathcal{L}_\rho(\mathbf{z}, \mathbf{a}^\ell, \sigma^\ell, \lambda^\ell).$$

(Observe that the minimizer $\mathbf{z}^{\ell+1}$ is unique according to Remark 7.1.1.) With Remark 7.1.6, a necessary and sufficient condition for $\mathbf{z}_m^{\ell+1}$, $m \in \mathbb{N}_{\mathcal{I}}$, to be a minimizer of the Lagrangian $\mathcal{L}$ is

$$0 \in \partial_{\mathbf{z}_m} \mathcal{L}_\rho(\mathbf{z}^{\ell+1}, \mathbf{a}^\ell, \sigma^\ell, \lambda^\ell).$$

Together with Theorem 7.1.9, this implies

$$\begin{aligned}
0 \in & \partial_{\mathbf{z}_m} \mathcal{L}_\rho(\mathbf{z}^{\ell+1}, \mathbf{a}^\ell, \sigma^\ell, \lambda^\ell) \\
= & \partial_{\mathbf{z}_m} \left(\sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i^{\ell+1}) + \overline{G}\left(\hat{\mathbf{a}}^\ell\right) + H(\sigma^\ell) + \sum_{i=1}^{\mathcal{I}} \lambda_i^\ell (\mathbf{z}_i^{\ell+1} - \mathbf{a}_i^\ell)^T + \frac{\rho}{2} \sum_{i=1}^{\mathcal{I}} \left\| \mathbf{z}_i^{\ell+1} - \mathbf{a}_i^\ell \right\|^2 \right) \\
= & \sum_{i=1}^{\mathcal{I}} \partial_{\mathbf{z}_m} F_i(\mathbf{z}_i^{\ell+1}) + \partial_{\mathbf{z}_m} \overline{G}\left(\hat{\mathbf{a}}^\ell\right) + \partial_{\mathbf{z}_m} H(\sigma^\ell) \\
& + \sum_{i=1}^{\mathcal{I}} \partial_{\mathbf{z}_m} \left(\lambda_i^\ell (\mathbf{z}_i^{\ell+1} - \mathbf{a}_i^\ell)^T \right) + \frac{\rho}{2} \sum_{i=1}^{\mathcal{I}} \partial_{\mathbf{z}_m} \left(\left\| \mathbf{z}_i^{\ell+1} - \mathbf{a}_i^\ell \right\|^2 \right) \\
= & \partial_{\mathbf{z}_m} F_m(\mathbf{z}_m^{\ell+1}) + \partial_{\mathbf{z}_m} \left(\lambda_m^\ell (\mathbf{z}_m^{\ell+1} - \mathbf{a}_m^\ell)^T \right) + \partial_{\mathbf{z}_m} \left(\frac{\rho}{2} \left\| \mathbf{z}_m^{\ell+1} - \mathbf{a}_m^\ell \right\|^2 \right) \\
= & \partial_{\mathbf{z}_m} F_m(\mathbf{z}_m^{\ell+1}) + \lambda_m^\ell + \rho \left(\mathbf{z}_m^{\ell+1} - \mathbf{a}_m^\ell \right).
\end{aligned}$$

The update for the dual variable (7.5c) (together with the scaling $\lambda_m = \rho \nu_m$)

$$\lambda_m^{\ell+1} = \lambda_m^\ell + \rho(\mathbf{z}_m^{\ell+1} - \mathbf{a}_m^{\ell+1}) \quad \Longleftrightarrow \quad \lambda_m^\ell = \lambda_m^{\ell+1} - \rho(\mathbf{z}_m^{\ell+1} - \mathbf{a}_m^{\ell+1})$$

implies

$$\begin{aligned}
& 0 \in \partial_{\mathbf{z}_m} F_m(\mathbf{z}_m^{\ell+1}) + \lambda_m^{\ell+1} - \rho(\mathbf{z}_m^{\ell+1} - \mathbf{a}_m^{\ell+1}) + \rho \left(\mathbf{z}_m^{\ell+1} - \mathbf{a}_m^\ell \right) \\
\Longleftrightarrow \quad & 0 \in \partial_{\mathbf{z}_m} F_m(\mathbf{z}_m^{\ell+1}) + \lambda_m^{\ell+1} + \rho \left(\mathbf{a}_m^{\ell+1} - \mathbf{a}_m^\ell \right).
\end{aligned}$$

Hence, $\mathbf{z}_m^{\ell+1} \in \mathbb{D}_m$ minimizes the expression

$$F_m(\mathbf{z}_m) + \left(\lambda_m^{\ell+1} + \rho \left(\mathbf{a}_m^{\ell+1} - \mathbf{a}_m^\ell \right) \right) \mathbf{z}_m^T.$$

We repeat the same arguments for the pair $(\mathbf{a}, \sigma)$ by considering the minimization problem

$$(\mathbf{a}^{\ell+1}, \sigma^{\ell+1}) \in \operatorname*{argmin}_{(\mathbf{a},\sigma) \in \mathbb{P}} \mathcal{L}_\rho(\mathbf{z}^{\ell+1}, \mathbf{a}, \sigma, \lambda^\ell).$$

(According to Remark 7.1.1, $\mathbf{a}^{\ell+1}$ is unique but $\sigma^{\ell+1}$ does not have to be unique.) In this case, we obtain the condition on the subdifferential

$$\begin{aligned}
& 0 \in \partial_{(\mathbf{a},\sigma)} \mathcal{L}_\rho(\mathbf{z}^{\ell+1}, \mathbf{a}^{\ell+1}, \sigma^{\ell+1}, \lambda^\ell) \\
\Longleftrightarrow \quad & 0 \in \partial_{(\mathbf{a},\sigma)} \overline{G}\left(\hat{\mathbf{a}}^{\ell+1}\right) + \partial_{(\mathbf{a},\sigma)} H(\sigma^{\ell+1}) - \begin{pmatrix} \lambda_1^\ell \\ \vdots \\ \lambda_{\mathcal{I}}^\ell \\ 0 \end{pmatrix} - \rho \begin{pmatrix} \mathbf{z}_1^{\ell+1} - \mathbf{a}_1^{\ell+1} \\ \vdots \\ \mathbf{z}_{\mathcal{I}}^{\ell+1} - \mathbf{a}_{\mathcal{I}}^{\ell+1} \\ 0 \end{pmatrix} \\
\Longleftrightarrow \quad & 0 \in \begin{pmatrix} \partial_{\mathbf{a}} \overline{G}\left(\hat{\mathbf{a}}^{\ell+1}\right) \\ \partial_\sigma H(\sigma^{\ell+1}) \end{pmatrix} - \begin{pmatrix} \lambda_1^\ell \\ \vdots \\ \lambda_{\mathcal{I}}^\ell \\ 0 \end{pmatrix} - \rho \begin{pmatrix} \mathbf{z}_1^{\ell+1} - \mathbf{a}_1^{\ell+1} \\ \vdots \\ \mathbf{z}_{\mathcal{I}}^{\ell+1} - \mathbf{a}_{\mathcal{I}}^{\ell+1} \\ 0 \end{pmatrix}.
\end{aligned}$$

With $\lambda^{\ell} = \lambda^{\ell+1} - \rho\left(\mathbf{z}^{\ell+1} - \mathbf{a}^{\ell+1}\right)$, we obtain

$$0 \in \begin{pmatrix} \partial_{\mathbf{a}}\overline{G}\left(\hat{\mathbf{a}}^{\ell+1}\right) \\ \partial_{\sigma}H(\sigma^{\ell+1}) \end{pmatrix} - \begin{pmatrix} \lambda_1^{\ell+1} - \rho\left(\mathbf{z}_1^{\ell+1} - \mathbf{a}_1^{\ell+1}\right) \\ \vdots \\ \lambda_{\mathcal{I}}^{\ell+1} - \rho\left(\mathbf{z}_{\mathcal{I}}^{\ell+1} - \mathbf{a}_{\mathcal{I}}^{\ell+1}\right) \\ 0 \end{pmatrix} - \rho \begin{pmatrix} \mathbf{z}_1^{\ell+1} - \mathbf{a}_1^{\ell+1} \\ \vdots \\ \mathbf{z}_{\mathcal{I}}^{\ell+1} - \mathbf{a}_{\mathcal{I}}^{\ell+1} \\ 0 \end{pmatrix}$$

$$\iff \quad 0 \in \begin{pmatrix} \partial_{\mathbf{a}}\overline{G}\left(\hat{\mathbf{a}}^{\ell+1}\right) \\ \partial_{\sigma}H(\sigma^{\ell+1}) \end{pmatrix} - \begin{pmatrix} \lambda_1^{\ell+1} \\ \vdots \\ \lambda_{\mathcal{I}}^{\ell+1} \\ 0 \end{pmatrix}$$

which means $(\mathbf{a}^{\ell+1}, \sigma^{\ell+1})$ minimizes

$$(\mathbf{a}^{\ell+1}, \sigma^{\ell+1}) \in \underset{(\mathbf{a},\sigma)\in\mathbb{P}}{\operatorname{argmin}}\ \overline{G}\left(\hat{\mathbf{a}}\right) + H(\sigma) - \sum_{i=1}^{\mathcal{I}} \lambda^{\ell+1}\mathbf{a}_i^T. \tag{7.11}$$

Summing up the results, we obtain the two inequalities

$$\overline{G}\left(\hat{\mathbf{a}}^{\ell+1}\right) + H(\sigma^{\ell+1}) - \sum_{i=1}^{\mathcal{I}} \lambda_i^{\ell+1}\mathbf{a}_i^{\ell+1^T} \leq \overline{G}\left(\hat{\mathbf{a}}^{\star}\right) + H(\sigma^{\star}) - \sum_{i=1}^{\mathcal{I}} \lambda_i^{\ell+1}\mathbf{a}_i^{\star^T}$$

and

$$\begin{aligned} &\sum_{i=1}^{\mathcal{I}} \left(F_i(\mathbf{z}_i^{\ell+1}) + \left(\lambda_i^{\ell+1} + \rho\left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell}\right)\right)\mathbf{z}_i^{\ell+1^T}\right) \\ &\qquad \leq \sum_{i=1}^{\mathcal{I}} \left(F_i(\mathbf{z}_i^{\star}) + \left(\lambda_i^{\ell+1} + \rho\left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell}\right)\right)\mathbf{z}_i^{\star^T}\right). \end{aligned}$$

With the notation

$$\begin{aligned} K^{\ell+1} &= \sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i^{\ell+1}) + \overline{G}\left(\hat{\mathbf{a}}^{\ell+1}\right) + H(\sigma^{\ell+1}), \\ K^{\star} &= \sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i^{\star}) + \overline{G}\left(\hat{\mathbf{a}}^{\star}\right) + H(\sigma^{\star}), \end{aligned}$$

the two inequalities add up to

$$\begin{aligned} K^{\ell+1} + &\sum_{i=1}^{\mathcal{I}} \left(\left(\lambda_i^{\ell+1} + \rho\left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell}\right)\right)\mathbf{z}_i^{\ell+1^T}\right) - \sum_{i=1}^{\mathcal{I}} \lambda_i^{\ell+1}\mathbf{a}_i^{\ell+1^T} \\ \leq K^{\star} + &\sum_{i=1}^{\mathcal{I}} \left(\left(\lambda_i^{\ell+1} + \rho\left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell}\right)\right)\mathbf{z}_i^{\star^T}\right) - \sum_{i=1}^{\mathcal{I}} \lambda_i^{\ell+1}\mathbf{a}_i^{\star^T}, \end{aligned}$$

which is equivalent to the expression

$$\begin{aligned} K^{\ell+1} - K^\star &\leq \sum_{i=1}^{\mathcal{I}} \left(\left(\lambda_i^{\ell+1} + \rho \left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right) \right) \left(\mathbf{z}_i^\star - \mathbf{z}_i^{\ell+1} \right)^T \right) - \sum_{i=1}^{\mathcal{I}} \lambda_i^{\ell+1} \left(\mathbf{a}_i^\star - \mathbf{a}_i^{\ell+1} \right)^T \\ &= \sum_{i=1}^{\mathcal{I}} \left(\rho \left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right) \left(\mathbf{z}_i^\star - \mathbf{z}_i^{\ell+1} \right)^T + \lambda_i^{\ell+1} \left(\mathbf{z}_i^\star - \mathbf{a}_i^\star - \mathbf{z}_i^{\ell+1} + \mathbf{a}_i^{\ell+1} \right)^T \right). \end{aligned}$$

In the limit, $\mathbf{z}_i^\star = \mathbf{a}_i^\star$ holds for all $i \in \mathbb{N}_{\mathcal{I}}$. Additionally, with the definition of the residuals $\mathbf{r}_i^{\ell+1} = \mathbf{z}_i^{\ell+1} - \mathbf{a}_i^{\ell+1}$ the inequality simplifies to

$$K^{\ell+1} - K^\star \leq \sum_{i=1}^{\mathcal{I}} \left(\rho \left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right) \left(\mathbf{z}_i^\star - \mathbf{z}_i^{\ell+1} \right)^T - \lambda_i^{\ell+1} \mathbf{r}_i^{\ell+1^T} \right).$$

Using $\mathbf{z}_i^\star = \mathbf{a}_i^\star$ and $\mathbf{z}_i^{\ell+1} = \mathbf{r}_i^{\ell+1} + \mathbf{a}_i^{\ell+1}$ leads to the expression

$$K^{\ell+1} - K^\star \leq \sum_{i=1}^{\mathcal{I}} \left(\rho \left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right) \left(\mathbf{a}_i^\star - \mathbf{a}_i^{\ell+1} - \mathbf{r}_i^{\ell+1} \right)^T - \lambda_i^{\ell+1} \mathbf{r}_i^{\ell+1^T} \right)$$

which shows that inequality (7.7b) holds.
As a last step, we show that (7.7a) is satisfied. Adding the Inequalities (7.7b) and (7.7c) and multiplying by 2 leads to

$$0 \geq 2 \sum_{i=1}^{\mathcal{I}} \left(\lambda_i^{\ell+1} - \lambda_i^\star \right) \mathbf{r}_i^{\ell+1^T} + 2 \sum_{i=1}^{\mathcal{I}} \rho \left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right) \left(\mathbf{r}_i^{\ell+1} + \left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^\star \right) \right)^T. \tag{7.12}$$

We first concentrate on the term

$$2 \left(\lambda_i^{\ell+1} - \lambda_i^\star \right) \mathbf{r}_i^{\ell+1^T}. \tag{7.13}$$

The updates of the dual variables are given by

$$\lambda_i^{\ell+1} = \lambda_i^\ell + \rho \mathbf{r}_i^{\ell+1}, \qquad \mathbf{r}_i^{\ell+1} = \frac{1}{\rho} \left(\lambda_i^{\ell+1} - \lambda_i^\ell \right).$$

With these definitions, the expression (7.13) can be expanded to

$$\begin{aligned} 2 \left(\lambda_i^{\ell+1} - \lambda_i^\star \right) \mathbf{r}_i^{\ell+1^T} &= 2 \left(\lambda_i^\ell + \rho \mathbf{r}_i^{\ell+1} - \lambda_i^\star \right) \mathbf{r}_i^{\ell+1^T} \\ &= 2 \left(\lambda_i^\ell - \lambda_i^\star \right) \mathbf{r}_i^{\ell+1^T} + 2\rho \left\| \mathbf{r}_i^{\ell+1} \right\|^2 \\ &= \frac{2}{\rho} \left(\lambda_i^\ell - \lambda_i^\star \right) \left(\lambda_i^{\ell+1} - \lambda_i^\ell \right)^T + 2\rho \left\| \mathbf{r}_i^{\ell+1} \right\|^2 \\ &= \frac{2}{\rho} \left(\lambda_i^\ell - \lambda_i^\star \right) \left(\lambda_i^{\ell+1} - \lambda_i^\ell \right)^T + \frac{1}{\rho} \left\| \lambda_i^{\ell+1} - \lambda_i^\ell \right\|^2 + \rho \left\| \mathbf{r}_i^{\ell+1} \right\|^2 \\ &= \frac{2}{\rho} \left(\lambda_i^\ell - \lambda_i^\star \right) \left((\lambda_i^{\ell+1} - \lambda_i^\star) - (\lambda_i^\ell - \lambda_i^\star) \right)^T \\ &\quad + \frac{1}{\rho} \left\| (\lambda_i^{\ell+1} - \lambda_i^\star) - (\lambda_i^\ell - \lambda_i^\star) \right\|^2 + \rho \left\| \mathbf{r}_i^{\ell+1} \right\|^2 \\ &= \frac{1}{\rho} \left\| \lambda_i^{\ell+1} - \lambda_i^\star \right\|^2 - \frac{1}{\rho} \left\| \lambda_i^\ell - \lambda_i^\star \right\|^2 + \rho \left\| \mathbf{r}_i^{\ell+1} \right\|^2. \end{aligned}$$

As a second step, we rewrite the second term of inequality (7.12) plus the additional term $\rho \left\| \mathbf{r}_i^{\ell+1} \right\|^2$ obtained in the last equation:

$$
\begin{aligned}
&\rho \left\| \mathbf{r}_i^{\ell+1} \right\|^2 + 2\rho \left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right) \left(\mathbf{r}_i^{\ell+1} + \left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\star} \right) \right)^T \\
&\quad = \rho \left\| \mathbf{r}_i^{\ell+1} \right\|^2 + 2\rho \left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right) \left(\mathbf{r}_i^{\ell+1} + \left((\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell}) + (\mathbf{a}_i^{\ell} - \mathbf{a}_i^{\star}) \right) \right)^T \\
&\quad = \rho \left\| \mathbf{r}_i^{\ell+1} \right\|^2 + 2\rho \left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right) \mathbf{r}_i^{\ell+1^T} + 2\rho \left\| \mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right\|^2 + 2\rho \left(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right) (\mathbf{a}_i^{\ell} - \mathbf{a}_i^{\star})^T \\
&\quad = \rho \left\| \mathbf{r}_i^{\ell+1} + \mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right\|^2 + \rho \left\| \mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right\|^2 + 2\rho \left((\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\star}) - (\mathbf{a}_i^{\ell} - \mathbf{a}_i^{\star}) \right) (\mathbf{a}_i^{\ell} - \mathbf{a}_i^{\star})^T \\
&\quad = \rho \left\| \mathbf{r}_i^{\ell+1} + \mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right\|^2 + \rho \left\| \mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right\|^2 + 2\rho (\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\star})(\mathbf{a}_i^{\ell} - \mathbf{a}_i^{\star})^T - 2\rho \left\| \mathbf{a}_i^{\ell} - \mathbf{a}_i^{\star} \right\|^2 \\
&\quad = \rho \left\| \mathbf{r}_i^{\ell+1} + \mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right\|^2 + \rho \left\| (\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\star}) - (\mathbf{a}_i^{\ell} - \mathbf{a}_i^{\star}) \right\|^2 \\
&\qquad + 2\rho (\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\star})^T (\mathbf{a}_i^{\ell} - \mathbf{a}_i^{\star}) - 2\rho \left\| \mathbf{a}_i^{\ell} - \mathbf{a}_i^{\star} \right\|^2 \\
&\quad = \rho \left\| \mathbf{r}_i^{\ell+1} + \mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right\|^2 + \rho \left\| \mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\star} \right\|^2 - \rho \left\| \mathbf{a}_i^{\ell} - \mathbf{a}_i^{\star} \right\|^2 .
\end{aligned}
$$

Summing up the results leads to

$$
\begin{aligned}
0 \geq & \frac{1}{\rho} \sum_{i=1}^{\mathcal{I}} \left\| \lambda_i^{\ell+1} - \lambda_i^{\star} \right\|^2 - \frac{1}{\rho} \sum_{i=1}^{\mathcal{I}} \left\| \lambda_i^{\ell} - \lambda_i^{\star} \right\|^2 \\
& + \rho \sum_{i=1}^{\mathcal{I}} \left\| \mathbf{r}_i^{\ell+1} + \mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right\|^2 + \sum_{i=1}^{\mathcal{I}} \rho \left\| \mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\star} \right\|^2 - \rho \sum_{i=1}^{\mathcal{I}} \left\| \mathbf{a}_i^{\ell} - \mathbf{a}_i^{\star} \right\|^2 .
\end{aligned}
$$

Here, we can identify the function L, and obtain the estimate

$$
L(\mathbf{a}^{\ell}, \lambda^{\ell}) - L(\mathbf{a}^{\ell+1}, \lambda^{\ell+1}) \geq \rho \sum_{i=1}^{\mathcal{I}} \left\| \mathbf{r}_i^{\ell+1} + \mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right\|^2
$$

or equivalently

$$
L(\mathbf{a}^{\ell+1}, \lambda^{\ell+1}) \leq L(\mathbf{a}^{\ell}, \lambda^{\ell}) + \sum_{i=1}^{\mathcal{I}} \left(-\rho \left\| \mathbf{r}_i^{\ell+1} \right\|^2 - \rho \left\| \mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right\|^2 - 2\rho \mathbf{r}_i^{\ell+1} (\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell})^T \right). \tag{7.14}
$$

From the minimization problem (7.11), we obtain the two inequalities

$$
\begin{aligned}
\overline{G}\left(\hat{\mathbf{a}}^{\ell+1} \right) + H(\sigma^{\ell+1}) - \sum_{i=1}^{\mathcal{I}} \lambda_i^{\ell+1} \mathbf{a}_i^{\ell+1^T} &\leq \overline{G}\left(\hat{\mathbf{a}}^{\ell} \right) + H(\sigma^{\ell}) - \sum_{i=1}^{\mathcal{I}} \lambda_i^{\ell+1} \mathbf{a}_i^{\ell^T} \\
\overline{G}\left(\hat{\mathbf{a}}^{\ell} \right) + H(\sigma^{\ell}) - \sum_{i=1}^{\mathcal{I}} \lambda_i^{\ell} \mathbf{a}_i^{\ell^T} &\leq \overline{G}\left(\hat{\mathbf{a}}^{\ell+1} \right) + H(\sigma^{\ell+1}) - \sum_{i=1}^{\mathcal{I}} \lambda_i^{\ell} \mathbf{a}_i^{\ell+1^T},
\end{aligned}
$$

which lead to the estimate

$$-\sum_{i=1}^{\mathcal{I}} \lambda_i^{\ell+1} \mathbf{a}_i^{\ell+1^T} - \sum_{i=1}^{\mathcal{I}} \lambda_i^{\ell} \mathbf{a}_i^{\ell^T} \leq -\sum_{i=1}^{\mathcal{I}} \lambda_i^{\ell+1} \mathbf{a}_i^{\ell^T} - \sum_{i=1}^{\mathcal{I}} \lambda_i^{\ell} \mathbf{a}_i^{\ell+1^T}$$

or

$$-\sum_{i=1}^{\mathcal{I}} (\lambda_i^{\ell+1} - \lambda_i^{\ell})(\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell})^T \leq 0$$

equivalently. With the residual $\rho \mathbf{r}_i^{\ell+1} = \lambda_i^{\ell+1} - \lambda_i^{\ell}$ we obtain

$$-\rho \sum_{i=1}^{\mathcal{I}} \mathbf{r}_i^{\ell+1} (\mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell})^T \leq 0$$

which can be used in (7.14)

$$\begin{aligned} L(\mathbf{a}^{\ell+1}, \lambda^{\ell+1}) &\leq L(\mathbf{a}^{\ell}, \lambda^{\ell}) - \sum_{i=1}^{\mathcal{I}} \rho \left\| \mathbf{r}_i^{\ell+1} \right\|^2 - \sum_{i=1}^{\mathcal{I}} \rho \left\| \mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right\|^2 \\ &= L(\mathbf{a}^{\ell}, \lambda^{\ell}) - \sum_{i=1}^{\mathcal{I}} \frac{1}{\rho} \left\| \lambda^{\ell+1} - \lambda^{\ell} \right\|^2 - \sum_{i=1}^{\mathcal{I}} \rho \left\| \mathbf{a}_i^{\ell+1} - \mathbf{a}_i^{\ell} \right\|^2 \end{aligned}$$

and shows that inequality (7.7a) holds. □

Theorem 7.1.12 provides convergence of the ADMM iterates to the optimal solution $K^\star$ of the optimization problem (7.1). The rate of convergence is not considered here. We refer to [29], where the authors investigate computational aspects of ADMM and examine the number of iterations to satisfy a given stopping criteria in the MPC context.
In the next section, we show how the steps (7.5) can be implemented efficiently such that the number of unknowns in the optimization problems are independent of the number of RESs.

7.1.4 Simplification of the ADMM formulation

The ADMM iterates can be solved in a distributed manner since Equation (7.5a) decomposes into $\mathcal{I}$ separable optimization problems which can individually be solved by their respective RESs in parallel. The optimization problem (7.5b)

$$(\mathbf{a}^{\ell+1}, \sigma^{\ell+1}) \in \underset{(\mathbf{a},\sigma)\in\mathbb{P}}{\operatorname{argmin}} \, \overline{G}(\hat{\mathbf{a}}) + H(\sigma) + \frac{\rho}{2} \sum_{i=1}^{\mathcal{I}} \left\| \mathbf{z}_i^{\ell+1} - \mathbf{a}_i + \nu_i^{\ell} \right\|^2$$

cannot be separated due to the coupling in the function $\overline{G}$. However, it is possible to make the number of unknowns in this problem independent of the number of RESs $\mathcal{I}$ by using

the averaged variables $\hat{\mathbf{a}} \in \mathbb{R}^N$ instead of $\mathbf{a} \in \mathbb{R}^{\mathcal{I}\times N}$. In order to show this, we rewrite the minimization problem (7.5b) in the form

$$\begin{aligned}(\mathbf{a}^{\ell+1}, \sigma^{\ell+1}) &\in \operatorname*{argmin}_{(\mathbf{a},\sigma)\in\mathbb{P}} \mathcal{L}_\rho\left(\mathbf{z}^{\ell+1}, \mathbf{a}, \sigma, \lambda^\ell\right) \\ &= \operatorname*{argmin}_{(\hat{\mathbf{a}},\sigma)\in\mathbb{P}} \overline{G}\left(\hat{\mathbf{a}}\right) + H(\sigma) + \frac{\rho}{2}\left\|\mathbf{z}^{\ell+1} - \mathbf{a}\right\|^2 + \sum_{i=1}^{\mathcal{I}} \lambda_i^{\ell^T}(\mathbf{z}_i^{\ell+1} - \mathbf{a}_i) \\ &= \operatorname*{argmin}_{(\mathbf{a},\sigma)\in\mathbb{P}} \overline{G}\left(\hat{\mathbf{a}}\right) + H(\sigma) + \sum_{i=1}^{\mathcal{I}} \frac{\rho}{2}\left\|\mathbf{z}_i^{\ell+1} - \mathbf{a}_i\right\|^2 + \lambda_i^{\ell^T}(\mathbf{z}_i^{\ell+1} - \mathbf{a}_i) \\ &= \operatorname*{argmin}_{(\mathbf{a},\sigma)\in\mathbb{P}} \overline{G}\left(\hat{\mathbf{a}}\right) + H(\sigma) + \frac{\rho}{2}\sum_{i=1}^{\mathcal{I}}\left\|\frac{\lambda_i^\ell}{\rho} + \mathbf{z}_i^{\ell+1} - \mathbf{a}_i\right\|^2. \end{aligned} \tag{7.15}$$

To continue, we require the following lemma.

Lemma 7.1.13. *For $\mathbf{c}, \mathbf{y}_i \in \mathbb{R}^N$, $i = 1, \ldots, \mathcal{I}$, the minimizer of the optimization problem*

$$\begin{aligned} \min_{\mathbf{v}_i\in\mathbb{R}^N} \quad & \sum_{i=1}^{\mathcal{I}} \|\mathbf{v}_i - \mathbf{y}_i\| \\ s.t. \quad & \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} \mathbf{v}_i = \mathbf{c} \end{aligned} \tag{7.16}$$

is given by $\mathbf{v}_i = \mathbf{y}_i + \mathbf{c} - \hat{\mathbf{y}}$ for all $i \in \mathbb{N}_\mathcal{I}$ where $\hat{\mathbf{y}} = (1/\mathcal{I})\sum_{i=1}^{\mathcal{I}} \mathbf{y}_i$.

Proof. For $\mathbf{y}_i = 0$ for all $i \in \mathbb{N}_\mathcal{I}$, the triangular inequality implies

$$\|\mathcal{I}\cdot\mathbf{c}\| = \min_{\mathbf{v}_i, \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\mathbf{v}_i = \mathbf{c}} \left\|\sum_{i=1}^{\mathcal{I}} \mathbf{v}_i\right\| \leq \min_{\mathbf{v}_i, \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\mathbf{v}_i = \mathbf{c}} \sum_{i=1}^{\mathcal{I}} \|\mathbf{v}_i\|$$

and equality is obtained for $\mathbf{v}_i = \mathbf{c}$ for all $i \in \mathbb{N}_\mathcal{I}$. For the general case, we use the coordinate transformation $\overline{\mathbf{v}}_i = \mathbf{v}_i - \mathbf{y}_i$. Then the equality constraint (7.16) reads

$$\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} \overline{\mathbf{v}}_i = \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} \mathbf{v}_i - \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} \mathbf{y}_i = \mathbf{c} - \hat{\mathbf{y}}$$

which shows the assertion. □

Applying this result to the minimization problem (7.15) and fixing the variables $\hat{\mathbf{a}} = \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} \mathbf{a}_i$ and σ, we see that the optimal solution satisfies

$$\frac{\hat{\lambda}^\ell}{\rho} + \hat{\mathbf{z}}^{\ell+1} - \hat{\mathbf{a}} = \frac{\lambda_i^\ell}{\rho} + \mathbf{z}_i^{\ell+1} - \mathbf{a}_i \tag{7.17}$$

for all $i \in \mathbb{N}_{\mathcal{I}}$ with the definition $\hat{\lambda} = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \lambda_i$. Hence, the minimization problem (7.15) can be rewritten as

$$\begin{aligned} \min_{(\hat{\mathbf{a}},\sigma)} \quad & \overline{G}(\hat{\mathbf{a}}) + H(\sigma) + \tfrac{\rho}{2} \textstyle\sum_{i=1}^{\mathcal{I}} \left\| \frac{\hat{\lambda}^\ell}{\rho} + \hat{\mathbf{z}}^{\ell+1} - \hat{\mathbf{a}} \right\|^2 \\ \text{s.t.} \quad & (\hat{\mathbf{a}}, \sigma) \in \hat{\mathbb{P}}, \end{aligned} \tag{7.18}$$

where the number of optimization variables is independent of the number of RESs and the set $\hat{\mathbb{P}}$ is defined as

$$\hat{\mathbb{P}} = \left\{ (\hat{\mathbf{a}}, \sigma) \in \mathbb{R}^N \times \mathbb{S} \,|\, A\hat{\mathbf{a}} + B\sigma - b = 0 \right\}. \tag{7.19}$$

Furthermore, if we use Equation (7.17) together with the scaling $\lambda = \rho\nu$ in the update of the Lagrange multipliers (7.5c) we obtain

$$\begin{aligned} \lambda_i^{\ell+1} &= \lambda_i^\ell + \rho \left(\mathbf{z}_i^{\ell+1} - \mathbf{a}_i^{\ell+1} \right) \\ &= \hat{\lambda}^\ell + \rho \left(\hat{\mathbf{z}}^{\ell+1} - \hat{\mathbf{a}}^{\ell+1} \right), \end{aligned}$$

which implies that $\lambda_i = \lambda_j$ holds for all $i, j \in \mathbb{N}_{\mathcal{I}}$ after the first iteration. Hence, the Lagrange multiplier $\lambda \in \mathbb{R}^{\mathcal{I} \times N}$ can be replaced by a Lagrange multiplier $\hat{\lambda} \in \mathbb{R}^N$ and the update of the dual variable $\hat{\lambda}$ is also independent of the number of RESs.
With these considerations, which follow the arguments given in [16, Chapter 7.3], the updates of Equation (7.5b) and (7.5c) reduce to the minimization problem

$$\operatorname*{argmin}_{(\hat{\mathbf{a}},\sigma) \in \hat{\mathbb{P}}} \overline{G}(\hat{\mathbf{a}}) + H(\sigma) + \frac{\rho \cdot \mathcal{I}}{2} \left\| \frac{\hat{\lambda}^\ell}{\rho} + \hat{\mathbf{z}}^{\ell+1} - \hat{\mathbf{a}} \right\|^2$$

and the update

$$\hat{\lambda}^{\ell+1} = \hat{\lambda}^\ell + \rho(\hat{\mathbf{z}}^{\ell+1} - \hat{\mathbf{a}}^{\ell+1}).$$

The update (7.5a) of the variables $\mathbf{z}_i$ of the individual RESs given by the solution of the minimization problem

$$\operatorname*{argmin}_{\mathbf{z}_i \in \mathbb{D}_i} F_i(\mathbf{z}_i) + \frac{\rho}{2} \left\| \mathbf{z}_i + \frac{\hat{\lambda}^\ell}{\rho} - \mathbf{a}_i^\ell \right\|^2$$

involves the variable $\mathbf{a}_i$ which differs for all $i \in \mathbb{N}_{\mathcal{I}}$ and as a consequence has to be transmitted to every RES individually. To avoid the need to communicate the individual $\mathbf{a}_i$, $i \in \mathbb{N}_{\mathcal{I}}$, we define

$$\Pi^\ell := \frac{\hat{\lambda}^\ell}{\rho} + \hat{\mathbf{z}}^\ell - \hat{\mathbf{a}}^\ell.$$

Now, Equation (7.17) can be rewritten in the form

$$\frac{\lambda_i^\ell}{\rho} - \mathbf{a}_i^\ell = -\mathbf{z}_i^\ell + \frac{\hat{\lambda}^\ell}{\rho} + \hat{\mathbf{z}}^\ell - \hat{\mathbf{a}}^\ell = -\mathbf{z}_i^\ell + \Pi^\ell.$$

If the variable Π^ℓ is known by the individual systems then the update $\mathbf{z}_i^{\ell+1}$ can be computed by

$$\underset{\mathbf{z}_i \in \mathbb{D}_i}{\operatorname{argmin}} \; F_i(\mathbf{z}_i) + \frac{\rho}{2} \left\| \mathbf{z}_i - \mathbf{z}_i^\ell + \Pi^\ell \right\|^2 \tag{7.20}$$

without the knowledge of individual variables specific to RES i. We now have all the necessary components to be in the position to summarize the ADMM algorithm and construct the DiMPC algorithm using ADMM at every time instant in the context of a network of RESs.

7.2 ADMM for a network of residential energy systems

7.2.1 The hierarchical distributed optimization algorithm

Algorithm 10 summarizes the ideas presented in the previous section and breaks down the ADMM iterates into tasks which have to be carried out by the individual RESs in parallel, and tasks which have to be executed by the CE or the grid operator.
Algorithm 10 provides several properties that are desirable for our application including:

- Only the parameter Π is transmitted to the RESs and not the energy demand $\mathbf{z}_i$. This implies that privacy of data between the individual RESs is maintained. Furthermore, the dimension of Π is independent of the number of RESs. Thus, the communication overhead scales well with the size of the network.
- The number of unknowns in the optimization problem of the grid operator is independent of the number of RESs. Hence, the computational complexity of the CE and the RESs is independent of $\mathcal{I}$.
- The RESs do not need to know the objective functions $\overline{G}$ and H defined by the CE. This allows the CE to modify the objective function without communicating this to the RESs.

7.2.2 ADMM in the receding horizon context

The last point in the preceding list is of particular interest when Algorithm 10 is embedded in a receding horizon scheme described in Algorithm 11, since it allows the grid operator to change the objective function at every time instant $k \in \mathbb{N}$ without changing the network or communication structure and without changing the optimization problem on the local level. Moreover, the grid operator does not need to react to changes in the local system dynamics or to changes in the constraints, i.e., changes in the set $\mathbb{D}_i$, $i \in \mathbb{N}_\mathcal{I}$.

7.3 Situation-based control of a network of RESs

The degree of freedom in choosing the objective function $\overline{G}$ and H and the linear coupling constraints in every MPC step offer different possibilities to apply the ADMM algorithm.

Algorithm 10 Hierarchical distributed optimization algorithm

Initialization: Set $\ell = 0$ and define $\Pi^0 \in \mathbb{R}^N$, $\lambda^0 \in \mathbb{R}^N$ and $\mathbf{z}_i^0 \in \mathbb{D}_i$ arbitrarily.
Main loop:
Phase 1 (RES i, $i \in \mathbb{N}_{\mathcal{I}}$): Receive Π^ℓ.

- Solve the minimization problem

$$\mathbf{z}_i^{\ell+1} = \underset{\mathbf{z}_i \in \mathbb{D}_i}{\operatorname{argmin}} \; F_i(\mathbf{z}_i) + \frac{\rho}{2} \left\| \mathbf{z}_i - \mathbf{z}_i^\ell + \Pi^\ell \right\|^2$$

  and send $\mathbf{z}_i^{\ell+1}$ to the CE.

Phase 2 (CE): Receive $\mathbf{z}_i^{\ell+1}$, $i = 1, \ldots, \mathcal{I}$.

- Compute the average $\hat{\mathbf{z}}^{\ell+1} = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i^{\ell+1}$.
- Solve the minimization problem

$$(\hat{\mathbf{a}}^{\ell+1}, \sigma^{\ell+1}) \in \underset{(\hat{\mathbf{a}},\sigma)\in\hat{\mathbb{P}}}{\operatorname{argmin}} \quad \left(\overline{G}\,(\hat{\mathbf{a}}) + H(\sigma) + \tfrac{\rho\cdot\mathcal{I}}{2} \left\| \hat{\mathbf{z}}^{\ell+1} - \hat{\mathbf{a}} + \tfrac{\hat{\lambda}^\ell}{\rho} \right\|^2 \right).$$

- Update the Lagrange multiplier

$$\hat{\lambda}^{\ell+1} = \hat{\lambda}^\ell + \rho \left(\hat{\mathbf{z}}^{\ell+1} - \hat{\mathbf{a}}^{\ell+1} \right).$$

- Compute and broadcast

$$\Pi^{\ell+1} = \hat{\mathbf{z}}^{\ell+1} - \hat{\mathbf{a}}^{\ell+1} + \frac{\hat{\lambda}^{\ell+1}}{\rho}.$$

Increment the iteration counter $\ell = \ell + 1$ and repeat the loop.

In this section, we discuss several applications for the network of RESs introduced in Chapter 3.
Similar to the degree of freedom of the CE, the RESs can choose their cost functions F_i, $i \in \mathbb{N}_{\mathcal{I}}$, individually, as long as the assumptions of Theorem 7.1.12 (i.e., the functions F_i have to be convex for all $i \in \mathbb{N}_{\mathcal{I}}$) are satisfied. F_i could for example be defined such that the usage of a storage device is penalized to extend its lifetime. Alternatively, the operational costs of a storage device could be modeled in the function F_i. Since we are interested in the performance of the network of RESs from the point of view of the CE and since the functions F_i have an impact on the overall cost function

$$K(\mathbf{z}, \mathbf{a}, \sigma) = \sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i) + \overline{G} \left(\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{a}_i \right) + H(\sigma)$$

defined in Equation (7.3), we use the functions $F_i \equiv 0$ in the following if not mentioned

Algorithm 11 Situation-based distributed model predictive control

1. **Initialization:**
   RES i, **(**$i \in \mathbb{N}_{\mathcal{I}}$**)**:

   - Measure the initial state of charge of the battery $x_i(k) = x_{i,0}$ and predict the values $\mathbf{s}_i(k; N)$.

   CE:

   - Define the objective functions $\overline{G}$ and H.

2. **Distributed optimization:** Apply Algorithm 10 to compute the optimal input $\mathbf{u}_i^\star(k; N)$ for $i = 1, \ldots, \mathcal{I}$.

3. Apply $\mathbf{u}_i^\star(k)$ for $i = 1, \ldots, \mathcal{I}$.

Increment the time index $k = k + 1$ and go to step 1.

otherwise. We focus on the dynamics

$$x_i(k+1) = \alpha_i x_i(k) + T\left(\beta_i u_i^+(k) + u_i^-(k)\right), \tag{7.21a}$$
$$z_i(k) = s_i(k) + u_i^+(k) + \gamma_i u_i^-(k) \tag{7.21b}$$

subject to the constraints

$$0 \leq x_i(k) \leq C_i, \tag{7.22a}$$
$$\underline{u}_i \leq u_i^-(k) \leq 0, \tag{7.22b}$$
$$0 \leq u_i^+(k) \leq \overline{u}_i, \tag{7.22c}$$
$$0 \leq \frac{u_i^-(k)}{\underline{u}_i} + \frac{u_i^+(k)}{\overline{u}_i} \leq 1. \tag{7.22d}$$

7.3.1 Vertical fluctuations

The standard setting considered in this thesis, penalizing the deviation from the average demand, is realized by the function $\overline{G} : \mathbb{R}^N \to \mathbb{R}$,

$$\overline{G}(\hat{\mathbf{a}}) = \left\| \hat{\mathbf{a}} - \mathbb{1}\hat{\zeta} \right\|^2. \tag{7.23}$$

Introducing the artificial variables σ is not necessary and hence, H is not needed. Similar, the functions F_i are defined as $F_i \equiv 0$ for all $i \in \mathbb{N}_{\mathcal{I}}$. Since in this case the variable $\hat{\mathbf{a}} \in \mathbb{R}^N$ is unconstrained, the update of the CE is simply given by

$$\hat{\mathbf{a}}^{\ell+1} = \frac{1}{2 + \rho\mathcal{I}} \left(2\hat{\zeta} + \rho\mathcal{I} \left(\hat{\mathbf{z}}^{\ell+1} + \frac{1}{\rho}\lambda^\ell \right) \right).$$

7.3.2 Temporary islanded operation of a network of RESs

We consider the case that the network of RESs represents a residential neighborhood connected to the main grid through a point of common coupling visualized in Figure 7.1. One of the benefits of a network of RESs, also called microgrid in this context, is the

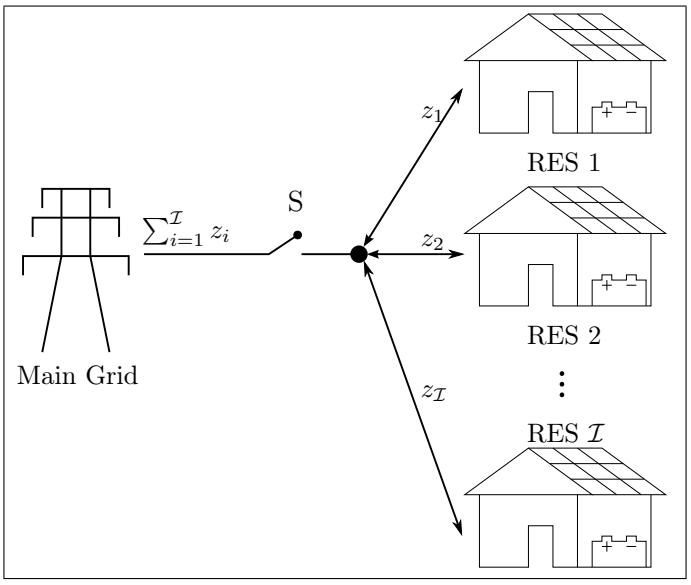

Figure 7.1: *Visualization of a network of RESs. A number of $\mathcal{I} \in \mathbb{N}$ RESs connected to the main grid through a point of common coupling. We assume that the RESs can be disconnected from the main grid through the switch S.*

potential to disconnect it from the main distribution network represented by the switch S in Figure 7.1. This is referred to as *islanding*, whereby the microgrid maintains normal operation using only the locally stored energy and energy locally generated by renewable generation units.

In this section, we propose an optimization problem to cover the problem of islanded operation of a microgrid with limited or even no conventional generation. At the start of and during islanded operation, it is vital to know the maximal allowable time window for which the microgrid is able to locally maintain supply on its own without any need for conventional generation or connection to the grid. More precisely, we set up an optimal control problem which serves two purposes. When solved once at time k with initial battery state $x(k)$, its solution tells us the number of time steps $\bar{k} \in \mathbb{N}$ the grid can be operated in islanded mode after a given time instant $k + k^\star$, $k^\star \in \mathbb{N}$. When solved iteratively within Algorithm 11, it yields the control strategy for keeping the microgrid in islanded mode from $k + k^\star$ to $k + k^\star + \bar{k}$.

We have the following two distinct applications in mind:

- $k^\star \geq 1$: A scheduled disconnection from the grid for an a priori specified time window.
- $k^\star = 0$: An unscheduled disconnection.

The main difference between the two cases is that in the first scenario the microgrid can specifically prepare itself in advance by charging the batteries until time $k+k^\star$, neglecting (possibly conflicting) other objectives. Despite this difference, both settings can be handled with the proposed methodology by adequately adapting the objective function and the constraints in Algorithm 10. Scheduled or forced islanded operation of a microgrid is for example also discussed in the paper [62]. Here, our approach can be used to calculate when the isolated microgrid will run out of energy and when additional generators have to be used to ensure a power supply without outages.
The possibility of disconnecting the grid at time $k^\star$ is equivalent to the existence of $\mathbf{z}_i \in \mathbb{D}_i$ (for $i \in \mathbb{N}_{\mathcal{I}}$) such that $\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i(k+k^\star) \leq 0$ is satisfied, i.e., the overall power demand is less or equal to zero at time $k^\star$. To find the maximal consecutive number of time steps from $k+k^\star$ to $k^\star+q^\star$ ($q^\star \geq 0$) such that $\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i(k+k^\star+q) \leq 0$ holds for all $q \in \{0,\dots,q^\star\}$, we define the following minimization problem.

Definition 7.3.1. *For a given time index $k^\star \in \{0,\dots,N-1\}$, set $M = N-k^\star$ and define the grid disconnection problem as*

$$(\hat{\mathbf{a}}^\star, \sigma^\star) \in \underset{(\hat{\mathbf{a}},\sigma)\in\hat{\mathbb{P}}}{\operatorname{argmin}} \; H(\sigma)$$

with

$$\hat{\mathbb{P}} = \left\{ (\hat{\mathbf{a}}, \sigma) \in \mathbb{R}^N \times \mathbb{R}^M \,\middle|\, (0\ I)\hat{\mathbf{a}} - \sigma \leq 0 \ \wedge \ \sigma \in \mathbb{S} = \mathbb{R}^M_{\geq 0} \right\} \tag{7.24}$$

and the objective function $H : \mathbb{R}^M_{\geq 0} \to \mathbb{R}$ is defined as

$$H(\sigma) = \sum_{q=1}^{M} (M+1-q)^\kappa \cdot \sigma(q)$$

for a positive constant $\kappa > 0$.

Remark 7.3.2. *In Definition 7.3.1 the functions $\overline{G}$ and F_i are tacitly defined as $\overline{G} \equiv 0$ and $F_i \equiv 0$ for all $i \in \mathbb{N}_{\mathcal{I}}$.*

Remark 7.3.3. *To be consistent with the definition of the minimization problem* (7.1)*, one has to introduce slack variables to obtain an equivalent formulation with equality constraints instead of inequality constraints. For simplicity, we stick to the formulation using inequality constraints but point out, that a reformulation with equality constraints can be obtained easily.*

Now we show that the number of leading zeros of a possibly non-unique optimal solution $\sigma^\star$ provides the maximal disconnection time if the weighting factor $\kappa > 0$ is chosen appropriately. Observe that the objective function H is linear and places a heavier penalty on the smaller indices of σ. To give an illustrative motivation for the choice of the objective function and the choice of κ, we assume that for all $i \in \mathbb{N}_{\mathcal{I}}$, $\alpha_i = 1$ in the system dynamics (7.21) before we prove the general case in Theorem 7.3.4.

Since the weighting parameters in the objective function H are positive, the constraint $\sigma \in \mathbb{R}^M_{\geq 0}$ implies $\hat{\mathbf{a}}(k+k^\star-1+q) = \sigma^\star(q)$ for all optimal $\sigma^\star$ with $\hat{\mathbf{a}}(k+k^\star-1+q) \geq 0$, $q \in \mathbb{N}_M$. For simplicity, consider an isolated (power) exchange between $\sigma(q_1) > 0$ and $\sigma(q_2)$ ($q_1 < q_2$, $q_1, q_2 \in \mathbb{N}_M$) of a feasible solution σ. Due to the linear system dynamics, reducing $\sigma(q_1)$ by $\gamma_i\varepsilon$ leads to an increase of $\sigma(q_2)$ by $\varepsilon\beta_i^{-1}$ in the case that losses have maximal impact. This is, for example, the case if

$$\hat{\mathbf{a}}(k+k^\star-1+q_1) = \sigma(q_1) \qquad \text{and} \qquad \hat{\mathbf{a}}(k+k^\star-1+q_2) = \sigma(q_2)$$

and $\hat{\mathbf{a}}(k+k^\star-1+q_1)$ can only be decreased by using a smaller $u_i^-(k+k^\star-1+q_1)$ by at least one RES i (i.e., discharge ε more from the battery at time $k+k^\star-1+q_1$) and simultaneously increase $u_i^+(k+k^\star-1+q_2)$ (i.e., charge ε more at time $k+k^\star-1+q_2$). Charging more at time $k+k^\star-1+q_2$ could, for example, be necessary to prevent that the battery constraints (6.20a) are violated at time steps $k+k^\star-1+q$ for $q > q_2$. Moreover, these considerations show that decreasing $\sigma(q_1)$ by $\gamma_i\varepsilon$ can always be compensated by maximally increasing $\sigma(q_2)$ by ε/β_i.
If this power exchange results in a new feasible $\tilde{\sigma}$ with

$$\tilde{\sigma}(q_1) = \sigma(q_1) - \gamma_i\varepsilon \quad \text{and} \quad \tilde{\sigma}(q_2) = \sigma(q_2) + \varepsilon/\beta_i$$

and $\tilde{\sigma}(q) = \sigma(q)$ for all $q \in \mathbb{N}_M \backslash \{q_1, q_2\}$, and since $q_1 < q_2$ by assumption, κ has to be chosen such that the value of the objective function decreases, i.e., $H(\tilde{\sigma}) < H(\sigma)$ holds. Hence, we obtain the estimate

$$0 > H(\tilde{\sigma}) - H(\sigma) = -(M+1-q_1)^\kappa \gamma_i\varepsilon + (M+1-q_2)^\kappa \varepsilon/\beta_i \tag{7.25}$$

or equivalently,

$$\gamma_i \cdot \beta_i > \left(\frac{M+1-q_2}{M+1-q_1}\right)^\kappa.$$

Since this inequality has to hold for arbitrary $q_1 < q_2$, $q_1, q_2 \in \mathbb{N}_M$ the estimate

$$\gamma_i \cdot \beta_i > \left(\frac{M-1}{M}\right)^\kappa > \left(\frac{M+1-q_2}{M+1-q_1}\right)^\kappa$$

has to be satisfied and leads to the condition

$$\kappa > \frac{\log(\gamma \cdot \beta)}{\log\left(\frac{M-1}{M}\right)}.$$

That this condition indeed suffices for the system dynamics (7.21) subject to the constraints (7.22) is rigorously shown in the following theorem.

Theorem 7.3.4. *Consider the grid disconnection problem defined in Definition 7.3.1. For an arbitrary optimal solution $(\hat{\mathbf{a}}^\star, \sigma^\star) \in \hat{\mathbb{P}}$ and an arbitrary feasible solution $(\hat{\mathbf{a}}^\sharp, \sigma^\sharp) \in \hat{\mathbb{P}}$, we define $q^\star, q^\sharp \in \{1, \ldots, M+1\}$ as the maximal indices such that, for all $q < q^\star$, $\sigma^\star(q) = 0$*

and, for all $q < q^\sharp$, $\sigma^\sharp(q) = 0$. *Let* $\beta := \min_{i=1,\dots,\mathcal{I}}\{\beta_i\}$ *and* $\gamma := \min_{i=1,\dots,\mathcal{I}}\{\gamma_i\}$ *define the maximal losses of the battery models.*
If κ *is chosen such that*

$$\kappa > \log(\beta \cdot \gamma) \Big/ \log\left(\frac{M-1}{M}\right) \tag{7.26}$$

then $q^\star \geq q^\sharp$ *holds, i.e., the grid can be disconnected for at most* $q^\star - 1$ *time steps.*

Proof. Assume that $\alpha_i = 1$ and let $(\hat{\mathbf{a}}^\star, \sigma^\star) \in \hat{\mathbb{P}}$ be an optimal solution of the minimization problem with κ chosen according to Equation (7.26). Let $q^\star$ denote the index of the first entry of $\sigma^\star$ which is unequal to zero, i.e., we have $\sigma^\star(q) = 0$ for all $q < q^\star$ and $\sigma^\star(q^\star) > 0$. (In the case $\sigma^\star = 0$, the statement of the theorem is trivially satisfied and hence, we can assume that $q^\star \leq M$.)
Assume there exists a feasible solution $(\hat{\mathbf{a}}^\sharp, \sigma^\sharp) \in \hat{\mathbb{P}}$ such that $\sigma^\sharp(q) = 0$ holds for all $q < q^\sharp$ and $q^\star < q^\sharp$. We will show that the existence of $(\hat{\mathbf{a}}^\sharp, \sigma^\sharp)$ contradicts the optimality of $(\hat{\mathbf{a}}^\star, \sigma^\star)$. We assume without loss of generality that

$$\hat{\mathbf{a}}^\star(k + k^\star - 1 + q) = \sigma^\star(q) \qquad \text{and} \qquad \hat{\mathbf{a}}^\sharp(k + k^\star - 1 + q) = \sigma^\sharp(q)$$

holds for all q with $\hat{\mathbf{a}}^\star(k + k^\star - 1 + q) \geq 0$ and $\hat{\mathbf{a}}^\sharp(k + k^\star - 1 + q) \geq 0$, respectively.
Since $\sigma^\sharp(q^\star) = 0$ and $\sigma^\star(q^\star) = \hat{\mathbf{a}}(k + k^\star - 1 + q^\star) > 0$ there is an index $i \in \mathbb{N}_\mathcal{I}$ such that

$$u_i^{+\star}(k + k^\star - 1 + q^\star) > 0 \qquad \text{or} \qquad u_i^{-\star}(k + k^\star - 1 + q^\star) > \underline{u}_i,$$

i.e., one of the constraints (7.22b) and (7.22c) is not active and it is possible to decrease

$$u_i^{+\star}(k + k^\star - 1 + q^\star) \qquad \text{and/or} \qquad u_i^{-\star}(k + k^\star - 1 + q^\star)$$

to reduce $\mathbf{z}_i^\star(k + k^\star - 1 + q^\star)$ and consequently also $\hat{\mathbf{a}}(k + k^\star - 1 + q^\star)$ and $\sigma^\star(q^\star)$. If we reduce $\mathbf{u}_i^{-\star}(k + k^\star - 1 + q^\star)$ by $\varepsilon > 0$, i.e.,

$$\tilde{u}_i^{-\star}(k + k^\star - 1 + q^\star) := u_i^{-\star}(k + k^\star - 1 + q^\star) - \varepsilon \tag{7.27}$$

then the state x_i decreases to

$$\tilde{x}_i^\star(k + k^\star - 1 + q) := x_i^{-\star}(k + k^\star - 1 + q) - \varepsilon \tag{7.28}$$

for all $q > q^\star$. If

$$x_i^{-\star}(k + k^\star + q) > 0 \qquad \text{for all } q > q^\star$$

then $u_i^{-\star}(k + k^\star - 1 + q^\star)$ can be changed without violating the constraints (7.22) and the variable $\sigma^\star(q^\star)$ can be reduced by $\gamma_i \cdot \varepsilon / \mathcal{I}$ which contradicts the optimality of $\sigma^\star$. The same argument applies if it is possible to change $u_i^{+\star}(k + k^\star - 1 + q^\star)$.
Hence, we can assume, that it is only possible to change

$$u_i^{-\star}(k + k^\star - 1 + q^\star) \qquad (\text{or } u_i^{+\star}(k + k^\star - 1 + q^\star), \text{ respectively})$$

by simultaneously changing

$$u_i^{-\star}(q) \qquad (\text{or } u_i^{+\star}(q))$$

at a time instant $q < k + k^\star - 1 + q^\star$ or $q > k + k^\star - 1 + q^\star$. Note that one of these options needs to be possible due to the existence of the solution $(\hat{\mathbf{a}}^\sharp, \sigma^\sharp)$ and since the constraints of the systems are decoupled, one can concentrate on one RES $i \in \mathbb{N}_\mathcal{I}$ only.
If it is possible to increase $x_i^\star(q)$ by increasing $u_i^{-\star}(q)$ or $u_i^{+\star}(q)$ at time steps $q < k + k^\star - 1 + q^\star$ without increasing $\sigma^\star(q)$ for all $q < k + k^\star - 1 + q^\star$ (i.e., there exist time steps $q < k + k^\star - 1 + q^\star$ such that $\hat{\mathbf{a}}^\star(q)$ can be increased without changing $\sigma^\star$), then this strategy increases in particular $x_i^\star(k + k^\star - 1 + q^\star)$. Hence, it is possible to decrease

$$u_i^{+\star}(k + k^\star - 1 + q^\star) \qquad \text{or} \qquad u_i^{-\star}(k + k^\star - 1 + q^\star)$$

without violating the constraints $x_i^\star(q) \geq 0$ for $q > k + k^\star - 1 + q^\star$, i.e., $\sigma^\star(q^\star)$ can be reduced which violates the optimality of $\sigma^\star$.
If the strategy (7.27) leads to $x_i^{-\star}(k+k^\star-1+q) < 0$ for some $q > q^\star$, again a contradiction to optimality can be derived based on the estimate (7.25) and the choice of κ by decreasing $\sigma^\star(q^\star)$ and increasing $\sigma^\star(q)$ for $q > q^\star$. Hence, $(\hat{\mathbf{a}}^\sharp, \sigma^\sharp) \in \hat{\mathbb{P}}$ does not exist, which completes the proof for $\alpha_i = 1$ for all $i \in \mathbb{N}_\mathcal{I}$.
Assume that $\alpha_i < 1$ for at least one $i \in \mathbb{N}_\mathcal{I}$. The strategy of reducing $\sigma(q^\star)$ by increasing $\hat{\mathbf{a}}^\star(q)$ for $q < q^\star$ is applicable in the same way as in the case $\alpha_i = 1$. Moreover, if $\sigma(q^\star)$ can be decreased using the idea of Equation (7.28) and simultaneously increasing $\sigma(q)$ for $q > q^\star$, then the amount of energy which is lost due to self-discharge for $q > q^\star$ decreases (i.e., the corresponding $u_i^{+\star}(k+k^\star-1+q^\star)$ or $u_i^{-\star}(k+k^\star-1+q^\star)$ can be decreased more before $x_i^{+\star}(k + k^\star - 1 + q) - \varepsilon = 0$ becomes active) which increases the amount $\sigma(q^\star)$ can be reduced. □

It has been shown in Theorem 7.3.4 that an optimal pair $(\hat{\mathbf{a}}^\star, \hat{\sigma}^\star)$ provides the maximal disconnection time if κ is chosen such that Condition (7.26) holds. However, for very large M, large values of κ are required, which lead to a numerically unstable scaling of the cost function H. Nevertheless, if a maximal disconnection time can be estimated, the presented approach can be easily generalized such that κ remains reasonably sized, e.g. if the maintenance work requires at most eight hours of the 24 hours within the prediction horizon N. Moreover, numerical experiments indicate that $\kappa = 1$ works well even if Condition (7.26) is violated.

Remark 7.3.5. *If no losses are considered, i.e., $\beta = \gamma = 0$, then any value $\kappa > 0$ can be used in the objective function H. For the values $M = 48$ and $\beta = \gamma = 0.95$, we obtain $\kappa > 4.88$ from Condition* (7.26).

Remark 7.3.6. *If the maximal desired duration of the islanded mode is a priori specified, the constraints $\begin{pmatrix} 0 & I & 0 \end{pmatrix} \hat{\mathbf{a}} - \sigma \leq 0$ can be used instead of $\begin{pmatrix} 0 & I \end{pmatrix} \hat{\mathbf{a}} - \sigma \leq 0$ to obtain a smaller value M and hence, a smaller κ.*

Remark 7.3.7. *Since the objective function is convex and defined on a convex and compact set (compactness of $\mathbb{S}$ can be easily enforced), Assumptions 7.1.10 hold according to Remark 7.1.11 and convergence of Algorithm 10 can be concluded from Theorem 7.1.12.*

7.3.3 Peak-detection

In this subsection, we propose a minimization problem to detect and to flatten the maximal and minimal peaks in the average power demand in the prediction horizon, i.e., we want to minimize

$$\max\left\{\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\mathbf{z}_i\right\} - \min\left\{\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\mathbf{z}_i\right\}.$$

If the maximal and minimal peaks are known to the energy provider, it can estimate an upper and a lower bound of power it needs to be able to provide in the next hours. If the interval of the power demand is smaller, then the interval containing the security margin of the energy provider also becomes smaller. The corresponding optimization problem is given by the formulation

$$\begin{aligned}(\mathbf{z}^\star,\mathbf{a}^\star) \quad \in \quad & \underset{(\mathbf{z}_i\mathbf{a})}{\operatorname{argmin}} \quad (\max\{\hat{\mathbf{a}}\} - \min\{\hat{\mathbf{a}}\}) \\ & \text{s.t.} \qquad \mathbf{a}_i \in \mathbb{R}^N \qquad \forall\, i = 1,\dots,\mathcal{I} \\ & \qquad\quad \mathbf{z}_i \in \mathbb{D}_i \qquad \forall\, i = 1,\dots,\mathcal{I} \\ & \qquad\quad \mathbf{z}_i - \mathbf{a}_i = 0 \qquad \forall\, i = 1,\dots,\mathcal{I}.\end{aligned}$$

To obtain a differentiable objective function, we introduce the variables $\sigma^+, \sigma^- \in \mathbb{R}$, $\sigma = (\sigma^+\ \sigma^-)$, and rewrite the optimization problem

$$\begin{aligned}(\mathbf{z}^\star,\mathbf{a}^\star,\sigma^\star) \quad \in \quad & \underset{(\mathbf{z},\mathbf{a},\sigma)}{\operatorname{argmin}} \quad s^+ - s^- \\ & \text{s.t.} \qquad \mathbf{a}_i \in \mathbb{R}^N \qquad \forall\, i = 1,\dots,\mathcal{I} \\ & \qquad\quad \mathbf{z}_i \in \mathbb{D}_i \qquad \forall\, i = 1,\dots,\mathcal{I} \\ & \qquad\quad \mathbf{z}_i - \mathbf{a}_i = 0 \qquad \forall\, i = 1,\dots,\mathcal{I} \\ & \qquad\quad \hat{\mathbf{a}} \le \mathbb{1}\sigma^+, \quad \hat{\mathbf{a}} \ge \mathbb{1}\sigma^-.\end{aligned}$$

which can be solved by Algorithm 10 using the definitions $F_i \equiv 0$ for all $i \in \mathbb{N}_{\mathcal{I}}$, $\overline{G} \equiv 0$, $H(\sigma) = \sigma(1) - \sigma(2)$ and

$$\hat{\mathbb{P}} = \left\{(\hat{\mathbf{a}},\sigma) \in \mathbb{R}^N \times \mathbb{R}^2 \,\middle|\, \begin{pmatrix} I \\ -I \end{pmatrix}\hat{\mathbf{a}}^T + \begin{pmatrix} \mathbb{1}^T & 0 \\ 0 & -\mathbb{1}^T \end{pmatrix}\sigma^T \le \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right\}.$$

7.3.4 Power balance

If a sufficient number of local generators are considered in the network of RESs the minimization problem

$$\begin{aligned}\mathbf{z}^\star \quad \in \quad & \underset{\mathbf{z}}{\operatorname{argmin}} \quad \textstyle\sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i) \\ & \text{s.t.} \qquad \tfrac{1}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}} \mathbf{z}_i = 0 \\ & \qquad\quad \mathbf{z}_i \in \mathbb{D}_i \qquad \forall\, i = 1,\dots,\mathcal{I}\end{aligned} \tag{7.29}$$

can be considered. Here, the focus is on the minimization of the costs for energy generation and hence, the functions $F_i : \mathbb{R}^N \to \mathbb{R}$, $i \in \mathbb{N}_{\mathcal{I}}$, need to be defined accordingly. The

minimization problem can be interpreted as minimizing individual costs or maximizing individual profit while maintaining power balance and consequently stability of the grid. By introducing the variable $\mathbf{a}$ and the function $\overline{G} \equiv 0$, we obtain the minimization problem

$$\begin{aligned}(\mathbf{z}^\star, \mathbf{a}^\star) \quad \in \quad & \underset{(\mathbf{z},\mathbf{a})}{\operatorname{argmin}} \quad \textstyle\sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i) + \overline{G}(\hat{\mathbf{a}}) \\ & \text{s.t.} \qquad \hat{\mathbf{a}} = 0 \\ & \qquad\quad \mathbf{z}_i \in \mathbb{D}_i \qquad\quad \forall\, i = 1, \dots, \mathcal{I} \\ & \qquad\quad \mathbf{z}_i - \mathbf{a}_i = 0 \qquad \forall\, i = 1, \dots, \mathcal{I}\end{aligned}$$

which implies that the optimal solution of the minimization problem of the CE in every iteration is given by $\hat{\mathbf{a}} = 0$ and hence, the update of the Lagrange multiplier $\hat{\lambda}$ reduces to

$$\hat{\lambda}^{\ell+1} = \hat{\lambda}^{\ell} + \rho \hat{\mathbf{z}}^{\ell+1}$$

for all $\ell \in \mathbb{N}$ and $\Pi^{\ell+1}$ is defined as

$$\Pi^{\ell+1} = \hat{\mathbf{z}}^{\ell+1} + \frac{\hat{\lambda}^{\ell+1}}{\rho}.$$

In the power balance problem one has to make sure that an optimal solution of the optimization problem (7.29) exists, i.e., local generators need to be able to provide enough power at every instant k.

7.3.5 Time-varying tube constraints

Instead of a power balance requiring that $\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i = 0$ holds, we can also consider a scenario where the energy provider wants to keep the power demand within a certain interval, i.e.,

$$\underline{\mathbf{c}} \leq \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \mathbf{z}_i \leq \overline{\mathbf{c}}$$

for possibly time-varying bounds $\underline{\mathbf{c}}, \overline{\mathbf{c}} \in \mathbb{R}^N$ while the RESs minimize their costs using the functions F_i, $i \in \mathbb{N}_{\mathcal{I}}$. With $\overline{G} \equiv 0$, the optimization problem

$$\begin{aligned}(\mathbf{z}^\star, \mathbf{a}^\star) \quad \in \quad & \underset{(\mathbf{z},\mathbf{a})}{\operatorname{argmin}} \quad \textstyle\sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i) + \overline{G}(\hat{\mathbf{a}}) \\ & \text{s.t.} \qquad \underline{\mathbf{c}} \leq \hat{\mathbf{a}} \leq \overline{\mathbf{c}} \\ & \qquad\quad \mathbf{z}_i \in \mathbb{D}_i \qquad\quad \forall\, i = 1, \dots, \mathcal{I} \\ & \qquad\quad \mathbf{z}_i - \mathbf{a}_i = 0 \qquad \forall\, i = 1, \dots, \mathcal{I}\end{aligned}$$

can be used to capture this setting. To avoid infeasibility of the minimization problem we relax the inequality constraints to

$$\underline{\mathbf{c}} - \underline{\sigma} \leq \hat{\mathbf{a}} \leq \overline{\mathbf{c}} + \overline{\sigma}$$

for $\underline{\sigma}, \overline{\sigma} \in \mathbb{R}^N_{\geq 0}$, $\sigma := (\underline{\sigma}\ \overline{\sigma})$ and include a function $H : \mathbb{R}^{2N}_{\geq 0} \to \mathbb{R}$ penalizing the deviation from $\sigma = 0$, e.g. $H(\sigma) = \|\sigma\|^2$. The overall optimization problem reads

$$\begin{aligned} (\mathbf{z}^\star, \mathbf{a}^\star, \sigma^\star) \quad \in \quad & \underset{(\mathbf{z},\mathbf{a},\sigma)}{\operatorname{argmin}} \quad \textstyle\sum_{i=1}^{\mathcal{I}} F_i(\mathbf{z}_i) + H(\sigma) \\ & \text{s.t.} \qquad (\hat{\mathbf{a}}, \sigma) \in \hat{\mathbb{P}} \\ & \qquad\quad \mathbf{z}_i \in \mathbb{D}_i \qquad\quad \forall\, i = 1, \dots, \mathcal{I} \\ & \qquad\quad \mathbf{z}_i - \mathbf{a}_i = 0 \qquad \forall\, i = 1, \dots, \mathcal{I} \end{aligned} \tag{7.30}$$

with

$$\hat{\mathbb{P}} = \left\{ (\hat{\mathbf{a}}, \sigma) \in \mathbb{R}^N \times \mathbb{R}^{2N} \,\middle|\, \begin{pmatrix} -I \\ I \end{pmatrix} \hat{\mathbf{a}}^T - \sigma^T \leq \begin{pmatrix} -\underline{\mathbf{c}}^T \\ \overline{\mathbf{c}}^T \end{pmatrix} \right\}.$$

The optimal variables $\sigma^\star$ can be used to change the bounds $\underline{\mathbf{c}}$ and $\overline{\mathbf{c}}$ at the next time step. If $\sigma^\star = 0$ holds, the bounds $-\underline{\mathbf{c}}$ and $\overline{\mathbf{c}}$ can be decreased. If $\sigma^\star \neq 0$ holds, $\sigma^\star$ can be an indicator for the energy provider to decide at which time steps in the prediction horizon an increased or decreased external power generation is necessary. To additionally smoothen the power demand profile within the bounds, the function

$$\overline{G}(\hat{\mathbf{a}}) = \hat{\mathbf{a}} \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 2 & -1 \\ & & & -1 & 1 \end{pmatrix} \hat{\mathbf{a}}^T \tag{7.31}$$

with an appropriate weighting can be included in the objective function. The function $\overline{G}$ penalizes the quadratic deviation between two consecutive elements $\hat{\mathbf{a}}(k+j) - \hat{\mathbf{a}}(k+j-1)$ for $j = 1, \dots, N-2$.

7.4 Numerical simulations

In this section, we visualize the results obtained by Algorithm 10 and Algorithm 11. We consider a setting of 300 RESs using the system dynamics (7.21) and the constraints (7.22). We concentrate on the temporary islanded operation of the RESs introduced in Section 7.3.2 and on the setting with time-varying tube constraints introduced in Section 7.3.5. All numerical simulations in this chapter are performed using MATLAB.

7.4.1 Islanded operation of a microgrid

In Figure 7.2, the open-loop solution of Algorithm 10 using the objective functions

$$\overline{G}(\hat{\mathbf{a}}(0;48)) + H(\sigma) = \eta_p \sum_{j=0}^{23} \left(\hat{a}(j) - \hat{\zeta} \right)^2 + \frac{\mathcal{I}}{20} \sum_{q=1}^{24} (25-q)^\kappa \sigma(q)$$

and $F_i(\mathbf{z}(0;48)) \equiv 0$ is visualized for $\eta_1 = 0$ and $\eta_2 = 10^3$ and $\kappa = 2.5$. Considering the definitions given in Section 7.3.2, the objective function can be used to calculate the length of the time interval the microgrid can operate in islanded mode with a disconnection time $k^\star = 24$. In the case $\eta_p = 0$, only the maximal disconnection time is computed and in the case $\eta_p = 10^3$, the deviation from the average demand $\hat{\zeta} = \mathbf{s}(0;48)\mathbb{1}^T/48$ is additionally penalized before the RESs are disconnected from the main grid.
For the simulation, the constants defining the dynamics and the constraints of the RESs are set to $C_i = 4$[kWh], $-\underline{u}_i = \overline{u}_i = 0.9$, $(\alpha_i, \beta_i, \gamma_i) = (0.96, 0.94, 0.98)$ and $x_i^0 = 2$[kWh], for $i \in \mathbb{N}_{300}$. Moreover, we use a discretization of $T = 0.5$[h] and a prediction horizon of $N = 48$. The parameter ρ in the ADMM formulation is set to $\rho = 10$. For the given initial

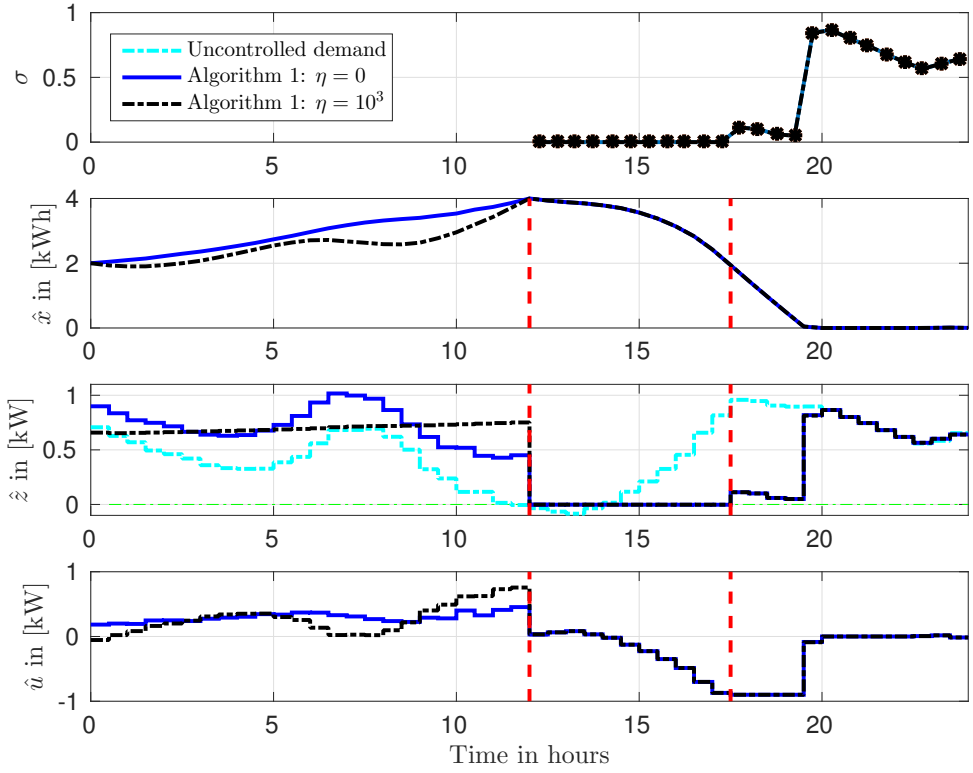

Figure 7.2: *Visualization of the variable σ and the average values $\hat{x}$, $\hat{z}$, and $\hat{\mathbf{u}} = \hat{\mathbf{u}}^+ + \hat{\mathbf{u}}^-$ for a single minimization problem with different weights for the function $\overline{G}$. The microgrid is disconnected after 12 hours and can stay islanded for 5 hours. Additionally, the uncontrolled power demand without storage devices is shown for comparison.*

state and the given parameters, the grid can be disconnected for 5 hours. In the case $\eta \neq 0$, the vertical grid load is additionally minimized in the first 12 hours. Observe that at the time the microgrid must be reconnected, the average state of charge of the batteries is still at 50%. This implies that the requirement that the microgrid is reconnected is not due to a shortage of locally stored energy but rather due to the maximal discharging rate being too small to satisfy the demand.
In Figure 7.3, the closed-loop performance of the receding horizon Algorithm 11 is visual-

ized. The grid operator wants to disconnect the grid after 48 time steps, i.e., $k^\star(k) = 47-k$

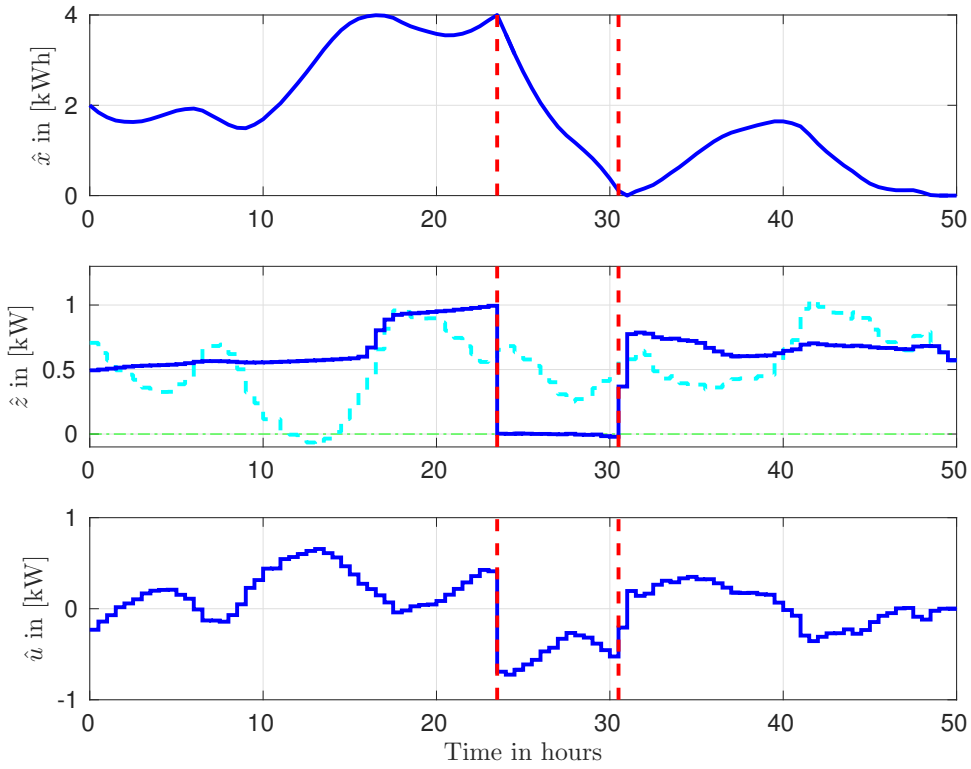

Figure 7.3: *Visualization of the average values $\hat{x}$, $\hat{z}$, and $\hat{u} = \hat{u}^+ + \hat{u}^-$ for a simulation of 50 hours. During the first 24 hours, the deviation with respect to the average demand is penalized and the maximal disconnection time is computed. Afterwards, the microgrid is disconnected for 14 time steps (7 hours) before the deviation from the average is penalized again. The controlled power demand (blue) can be compared to the uncontrolled power demand (cyan) in the second graph.*

for $k = 0, 1, \ldots, 47$. Therefore, the peak-to-peak variation is penalized and the disconnection time is maximized using the functions

$$\overline{G}(\hat{\mathbf{a}}(k;48)) + H(\sigma) = 10 \sum_{j=k}^{47} \Big(\hat{a}(j) - \hat{\zeta}(k)\Big)^2 + \frac{\mathcal{I}}{20} \sum_{q=1}^{k} (k-q+1)\sigma(q)$$

(and $F_i(\mathbf{z}(k;48)) \equiv 0$ for all $i \in \mathbb{N}_\mathcal{I}$ and for all $k \in \mathbb{N}$).
After 24 hours (or $k = 47$), the grid is disconnected and stays disconnected for 14 time steps, the maximal time in islanded mode. After the RESs are connected again, the simulation is continued by minimizing the deviation from the average using the objective function

$$\overline{G}(\hat{\mathbf{a}}(k;48)) = 10 \sum_{j=k}^{k+47} \Big(\hat{a}(j) - \hat{\zeta}(k)\Big)^2 .$$

As already pointed out, since only the cost function of the grid operator changes in the receding horizon scheme, the RESs do not need to change their setting on the local level. Even though $\kappa = 1$ does not satisfy condition (7.26) of Theorem 7.3.4, the maximal disconnection time is returned, which shows that condition (7.26) is very conservative in our application.

7.4.2 Time-varying tube constraints

In Figure 7.4, the closed-loop solution corresponding to the setting introduced in Section 7.3.5 is visualized. For the simulation, the constants of the 300 RESs are set to

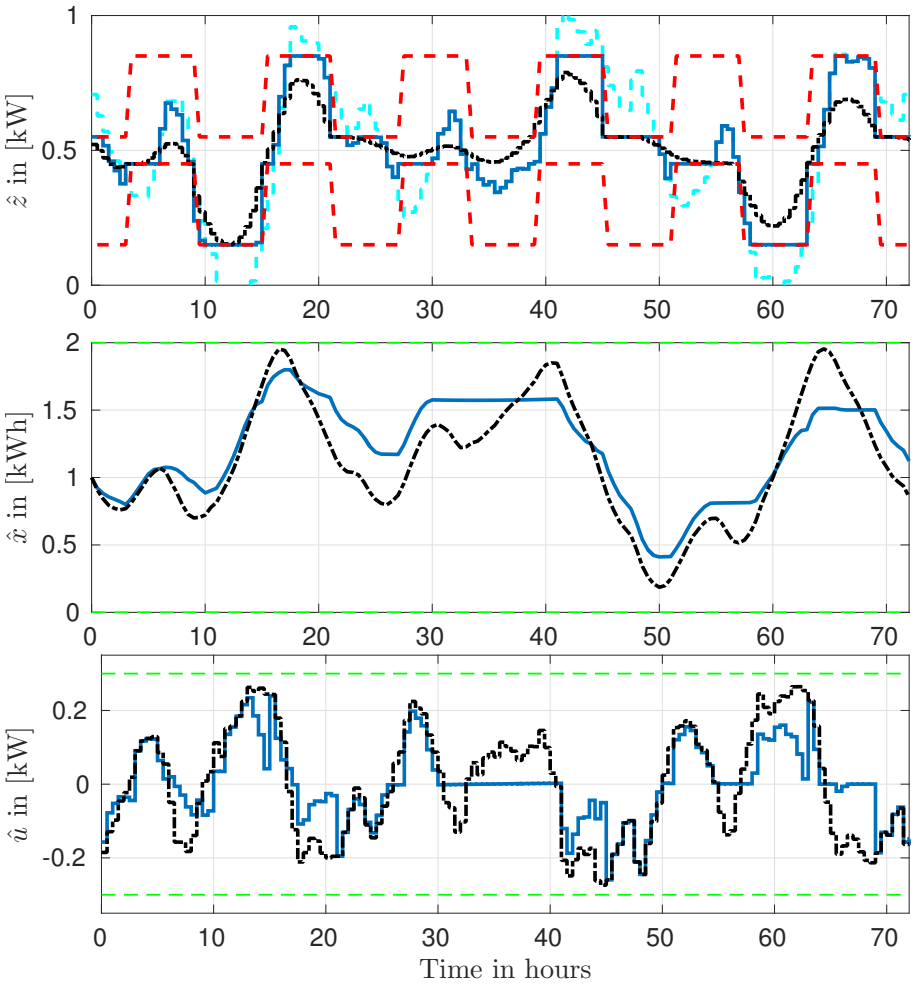

Figure 7.4: *The visualization of the closed-loop solution for time-varying tube constraints, introduced in Section 7.3.5, is shown in red. The graph on top shows the average power demand $\hat{z}$ of the uncontrolled setting (cyan) compared to the controlled setting with smoothing (black) and without smoothing (blue). The corresponding states $\hat{x}$ and the inputs $\hat{u} = \hat{u}^+ + \hat{u}^-$ are shown in the middle and in the lower graphic.*

$C_i = 2$[kWh], $-\underline{u}_i = \overline{u}_i = 0.3$[kW], $(\alpha_i, \beta_i, \gamma_i) = (1, 1, 1)$ (i.e., we consider the case without

losses) and $x_i^0 = 1$[kWh]. The discretization parameter is set to $T = 0.5$[h], the prediction horizon $N = 48$ is used and the MPC algorithm is simulated for three days, i.e., $\mathcal{N} = 144$. The lower bound of the tube, which can be seen in Figure 7.4 (top) in red switches between the values 0.15[kW] and 0.55[kW] and the upper bound switches between 0.45[kW] and 0.85[kW].[3] The uncontrolled setting is compared to the receding horizon solution of the minimization problem (7.30), with and without the use of the function $\overline{G}$ defined in (7.31). (In the setting with $\overline{G}$, the parameter $\rho = 0.05$ and in the setting without $\overline{G}$, the parameter $\rho = 0.5$ are used, respectively.)
We observe that with the chosen time-varying bounds, the closed loop solutions are inside the tube for all time steps. In the case where $\overline{G}$ is used according to Equation (7.31), we obtain a smoother trajectory $\hat{\mathbf{z}}(0; 144)$.
Again, we point out that only the CE has to change its minimization problem at every time step. The RESs have no information about the tube constraints and do not need to adapt their respective minimization problems.

[3]The definition of the tubes reflect the four daily peaks in the uncontrolled setting (see Figure 3.3) and are chosen to illustrate our approach. They can be replaced by any other definition of time-varying lower and upper bounds if desired.

Chapter 8

Conclusions

In this chapter, we compare the hierarchical distributed optimization algorithms presented in Chapters 5 to 7 and discuss possible topics for future research.

8.1 Comparison of the distributed optimization algorithms

In Chapters 5 to 7, we discussed three hierarchical distributed optimization algorithms to iteratively solve a single optimization problem at a fixed time instant $k \in \mathbb{N}$. All three algorithms have a similar structure regarding the distribution of the tasks between the RESs and the CE. The RESs solve an optimization problem solely based on the knowledge of their local system dynamics, their local constraints and the information communicated by the CE. The CE solves a simple optimization problem in every iteration, i.e., the number of unknowns and the number of constraints in the optimization problem are independent of the size of the network. Additionally, the CE is responsible for communication of necessary information between the RESs. Similar to the unknowns in the optimization problem of the CE, the number of variables sent by the CE to the RESs are independent of the number of RESs. Furthermore, personal data of a single RES remains private and cannot be recovered from the communicated data.
In the algorithms considered in Chapter 5, an average of the primal solution $\hat{\mathbf{z}}^\ell$ and the step-size θ^ℓ is transmitted and the primal problem is solved directly, whereas in the algorithms in Chapters 6 and 7, the dual variables are communicated and the optimal primal solution is recovered from the optimal dual solution. This makes the algorithms based on dual decomposition in Chapters 6 and 7 more flexible, since the RESs do not need to know the objective function or the goal of the minimization problem. However, it also makes the optimization problem less transparent for the RESs. The flexibility offers the CE several degrees of freedom, but the non-transparency might prevent the RESs to join the network coordinated by the CE and being willing to share their data with the CE. Instead, in the distributed optimization problem in Chapter 5, the RESs have access to the overall objective function and thus, the algorithm provides more transparency while keeping local information private among the RESs.
The dual ascent algorithm (Chapter 6) is only able to solve a relaxed formulation of the

original optimization problem if the objective function is not strongly convex. Moreover, the stepsize to update the Lagrange multipliers in every iteration has to be chosen carefully to ensure convergence. Furthermore, it decreases when the size of the network grows. In the primal algorithm (Chapter 5), the stepsize is optimized in every iteration. Thus, an a priori stepsize selection depending on the number of RESs in the network is not necessary. Nonetheless, the dual ascent algorithm offers a non-cooperative price-based interpretation of the optimization process and is more flexible than the primal algorithm with respect to the choice of the objective function and additional constraints.
The ADMM algorithm (Chapter 6) circumvents the problems of the dual ascent algorithm by including an additional term in the Lagrangian. Hence, the stepsize, which is defined through the parameter ρ, is independent of the size of the network. However, the speed of convergence strongly depends on the choice of ρ and stepsize rules in the context of ADMM are still a topic of ongoing research. The ADMM algorithm is the most flexible algorithm considered in this thesis. Yet, a non-cooperative price-based interpretation introduced for the dual ascent algorithm is not applicable due to the additional coupling, which ensures strong convexity properties of the augmented Lagrangian.

8.2 Future work

In Chapter 4, MPC schemes for time-varying discrete-time systems were introduced. While the numerical simulations in the Chapters 4 to 7 indicate that the MPC closed-loop costs

$$V_\infty^\mu(x(0)) = \liminf_{k\to\infty} \frac{1}{k} \sum_{j=0}^{\infty} \ell(x^\mu(j), \mu(j))$$

perform well compared to the (unknown) infinite horizon costs

$$V_\infty(x(0)) = \lim_{k\to\infty} \lim_{\mathbf{u}(0;k)\in\mathbb{U}^{0,\infty}(x(0))} \frac{1}{k} \sum_{j=0}^{k} \ell(x^{\mathbf{u}}(j), u(j)),$$

a rigorous proof for the performance of MPC for time-varying discrete-time systems could not be derived. In the recent paper [30], the authors derive performance estimates for linear discrete-time systems along with strictly convex running costs ℓ, by showing that the corresponding OCP is strictly dissipative. In [73], the authors provide performance guarantees, based on dissipativity, for economic MPC for the periodic operation of discrete-time system. Combining both results provides promising ideas for the derivation of closed-loop performance guarantees of time-varying discrete-time systems.
In the Chapters 5 to 7, we presented three different hierarchical distributed optimization algorithms, each of which has its own strengths and advantages over the others. However, several other distributed optimization algorithms are discussed in the literature and could also be considered in this thesis. Proximal point algorithms (or proximal methods) compose another class of algorithms which could also be considered additionally (see for

example [81],[59],[92]). A comprehensive study and comparison of distributed optimization algorithms and their equivalences is beyond the scope of this thesis but nevertheless, constitutes to a goal of future research in this area.
Most of the distributed optimization algorithms can only handle convex optimization problems, i.e., optimization problems with convex objective functions defined on convex sets. In Chapter 5, the proposed hierarchical algorithm is extended to handle non-convex constraints. Even though the convergence properties of the algorithm are not investigated in this thesis, the numerical results look promising. A rigorous proof of convergence can be a next step towards the direction of non-convex distributed optimization.
In [53], the authors recently proposed a distributed optimization algorithm for non-convex optimization problems for non-convex objective functions and non-convex constraints, which converges to a local optimal solution. The proposed algorithm allows for an extension of the network of RESs to more complicated dynamics and more complicated objective functions and will be investigated in one of our next steps.

Acronyms

ADMM	alternating direction method of multipliers
ASF	averaged smoothing factor
CE	central entity
CMPC	centralized model predictive control
DeMPC	decentralized model predictive control
DiMPC	distributed model predictive control
IPOPT	Interior Point Optimizer
LOE	loss of energy
MPC	model predictive control
MQD	mean-quadratic-deviation
OCP	optimal control problem
PTP	peak-to-peak
RES	residential energy system
SOC	state of charge

Glossary

0	zero vector of appropriate dimension, i.e., $0 = (0, \ldots, 0)$
$\mathbb{1}$	all one vector of appropriate dimension i.e., $\mathbb{1} = (1, \ldots, 1)$
$B_\varepsilon(\mathbf{y})$	open ball of radius $\varepsilon > 0$ around $\mathbf{y} \in \mathbb{R}^n$, i.e., $B_\varepsilon(\mathbf{y}) = \{\overline{\mathbf{y}} \,\vert\, \vert\vert\overline{\mathbf{y}} - \mathbf{y}\vert\vert < \varepsilon\}$
$\mathbb{D}_i^{k,N}(x_{i,0})$	$\mathbf{z}_i \in \mathbb{D}_i^{k,N}(x_{i,0}) \subset \mathbb{R}^{p \times N}$ (admissible set with respect to z_i)
I	identity matrix of appropriate dimension
$\mathcal{I}$	number of residential energy system $\mathcal{I} \in \mathbb{N}$
N	prediction horizon $N \in \mathbb{N}$
$\mathbb{N}$	natural numbers (including zero), i.e., $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$
$\mathbb{N}_\mathcal{I}$	natural numbers $\leq \mathcal{I}$, i.e., $\mathbb{N}_\mathcal{I} = \{1, \ldots, \mathcal{I}\}$
$\vert\vert \cdot \vert\vert$	2-norm of a vector
$\mathbb{R}$	real numbers
$\mathbb{R}_{\geq 0}$	non-negative real numbers, i.e., $\mathbb{R}_{\geq 0} = \{y \in \mathbb{R} \vert y \geq 0\}$
$\mathbb{R}_{>0}$	positive real numbers, i.e., $\mathbb{R}_{>0} = \{y \in \mathbb{R} \vert y > 0\}$
s	exogenous data of the overall system; $s \in \mathbb{R}^d$
s_i	exogenous data of system $i \in \mathbb{N}_\mathcal{I}$; $s_i \in \mathbb{R}^{d_i}$
$\mathbb{U}_i$	input constraints $u_i \in \mathbb{U}_i \subset \mathbb{R}^{m_i}$
u	input of the overall system; $u \in \mathbb{R}^m$
$\mathbb{U}_i^{k,N}(x_{i,0})$	$\mathbf{u}_i \in \mathbb{U}_i^{k,N}(x_{i,0}) \subset \mathbb{U}_i^N$ (admissible set with respect to u_i)
u_i	input of system $i \in \mathbb{N}_\mathcal{I}$; $u_i \in \mathbb{R}^{m_i}$
$\mathbb{X}_i$	state constraints $x_i \in \mathbb{X}_i \subset \mathbb{R}^{n_i}$
x	state of the overall system; $x \in \mathbb{R}^n$
$\mathbb{X}_i^N$	$\mathbf{x}_i \in \mathbb{X}_i^N \subset \mathbb{R}^{n_i \times N}$ (admissible set with respect to x_i)
x_i	state of system $i \in \mathbb{N}_\mathcal{I}$; $x_i \in \mathbb{R}^{n_i}$
z	communication variable of the overall system; $z \in \mathbb{R}^{p\mathcal{I}}$
z_i	communication variable of system $i \in \mathbb{N}_\mathcal{I}$; $z_i \in \mathbb{R}^p$

Bibliography

[1] Ausgrid dataset. http://www.ausgrid.com.au/Common/About-us/Corporate-information/Data-to-share/Solar-household-data.aspx#.Vyx_aqo1enw, accessed: 2016-05-06.

[2] pv magazine. http://www.pv-magazine.de/marktuebersichten/batteriespeicher. accessed: 2016-05-06.

[3] HSL Mathematical Software Library. A collection of Fortran codes for large-scale scientific computation, 2004. http://hsl.rl.ac.uk/.

[4] Erneuerbare Energien in Deutschland, Daten zur Entwicklung 2015. Bundesministerium für Wirtschaft und Energie (BMWI), 2016.

[5] N. Amjady, F. Keynia, and H. Zareipour. Short-term load forecast of microgrids by a new bilevel prediction strategy. *IEEE Transactions on Smart Grid*, 1(3):286–294, 2010.

[6] R. Amrit, J. B. Rawlings, and D. Angeli. Economic optimization using model predictive control with a terminal cost. *Annual Reviews in Control*, 35(2):178–186, 2011.

[7] D. Angeli, R. Amrit, and J. B. Rawlings. On average performance and stability of economic model predictive control. *IEEE Transactions on Automatic Control*, 57(7):1615–1626, 2012.

[8] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar, and J. R. Fonollosa. Demand-side management via distributed energy generation and storage optimization. *IEEE Transactions on Smart Grid*, 4(2):866–876, 2013.

[9] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar, and J. R. Fonollosa. Noncooperative and cooperative optimization of distributed energy generation and storage in the demand-side of the smart grid. *IEEE Transactions on Signal Processing*, 61(10):2454–2472, 2013.

[10] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar, and J. R. Fonollosa. Noncooperative day-ahead bidding strategies for demand-side expected cost minimization with real-time adjustments: A GNEP approach. *IEEE Transactions on Signal Processing*, 62(9):2397–2412, 2014.

[11] J.-P. Aubin and A. Cellina. *Differential inclusions: Set-valued maps and viability theory*, volume 264. Springer, 1984.

[12] S. Barker, A. Mishra, D. Irwin, P. Shenoy, and J. Albrecht. SmartCap: Flattening peak electricity demand in smart homes. In *Proc. of the International IEEE Conference on Pervasive Computing and Communications*, 2012.

[13] A. Beja and M. B. Goldman. On the dynamic behavior of prices in disequilibrium. *The Journal of Finance*, 35(2):235–248, 1980.

[14] D. P. Bertsekas. *Nonlinear programming*. Athena scientific, 1999.

[15] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, Belmont, MA, USA, 1989.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[17] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[18] P. Braun, T. Faulwasser, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann. Maximal islanding time for microgrids via distributed predictive control. In *Proc. of 22nd International Symposium on Mathematical Theory of Networks and Systems*, pages 652–659, 2016.

[19] P. Braun, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann. Predictive control of a smart grid: A distributed optimization algorithm with centralized performance properties. In *Proc. of the 54th IEEE Conference on Decision and Control (CDC)*, pages 5593–5598, 2015.

[20] P. Braun, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann. A real-time pricing scheme for residential energy systems using a market maker. In *Proc. of the 5th Australian Control Conference (AUCC)*, pages 259–262, 2015.

[21] P. Braun, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann. A distributed optimization algorithm for the predictive control of smart grids. *IEEE Transactions on Automatic Control*, 2016. DOI:10.1109/TAC.2016.2525808.

[22] P. Braun, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann. Model predictive control of residential energy systems using energy storage & controllable loads. In *Progress in Industrial Mathematics at ECMI 2014*, pages 1–7, 2016. (accepted for publication).

[23] P. Braun, E. Hernández, and D. Kalise. Reduced-order LQG control of a Timoshenko beam model. *Bulletin of the Brazilian Mathematical Society, New Series*, 47(1):143–155, 2016.

[24] P. Braun, J. Pannek, and K. Worthmann. Predictive control algorithms: Stability despite shortened optimization horizons. In *Proc. of the 15th IFAC Workshop on Control Applications of Optimization (CAO)*, pages 274–279, 2012.

[25] E. F. Camacho and C. Bordons. *Model predictive control*. Springer-Verlag, 1999.

[26] T.-H. Chang, A. Nedić, and A. Scaglione. Distributed constrained optimization by consensus-based primal-dual perturbation method. *IEEE Transactions on Automatic Control*, 59(6):1524–1538, 2014.

[27] H. Chen, T. N. Cong, W. Yang, C. Tan, Y. Li, and Y. Ding. Progress in electrical energy storage system: A critical review. *Progress in Natural Science*, 19(3):291–312, 2009.

[28] Y. C. Cheng. Dual gradient method for linearly constrained, strongly convex, separable mathematical programming problems. *Journal of Optimization Theory and Applications*, 53(2):237–246, 1987.

[29] C. Conte, T. Summers, M. N. Zeilinger, M. Morari, and C. N. Jones. Computational aspects of distributed optimization in model predictive control. In *Proc. of the 51st IEEE Conference on Decision and Control (CDC)*, pages 6819–6824, 2012.

[30] T. Damm, L. Grüne, M. Stieler, and K. Worthmann. An exponential turnpike theorem for dissipative discrete time optimal control problems. *SIAM Journal on Control and Optimization*, 52(3):1935–1957, 2014.

[31] R. Deng, Z. Yang, M.-Y. Chow, and J. Chen. A survey on demand response in smart grids: Mathematical models and approaches. *IEEE Transactions on Industrial Informatics*, 3:570–582, 2015.

[32] M. Diehl, R. Amrit, and J. B. Rawlings. A lyapunov function for economic optimizing model predictive control. *IEEE Transactions on Automatic Control*, 56(3):703–707, 2011.

[33] M. D. Doan, T. Keviczky, I. Necoara, M. Diehl, and B. De Schutter. A distributed version of Han's method for DMPC using local communications only. *Control Engineering and Applied Informatics*, 11(3):6–15, 2009.

[34] M. D. Doan, T. Keviczky, and B. De Schutter. An iterative scheme for distributed model predictive control using Fenchel's duality. *Journal of Process Control*, 21(5):746–755, 2011.

[35] J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 57(3):592–606, 2012.

[36] J. Eckstein. Augmented lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results. *RUTCOR Research Reports*, 32, 2012.

[37] European Photovoltaic Industry Association (EPIA). Global market outlook for photovoltaics 2013–2017, May 2013.

[38] C. Le Floch, F. Belletti, S. Saxena, A. M. Bayen, and S. Moura. Distributed optimal charging of electric vehicles for demand response and load shaping. In *Proc. of the 54th Conference on Decision and Control (CDC)*, pages 6570–6576, 2015.

[39] M. B. Garman. Market microstructure. *Journal of financial Economics*, 3(3):257–275, 1976.

[40] P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer. Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica*, 49(3):829–833, 2013.

[41] P. Giselsson and A. Rantzer. Distributed model predictive control with suboptimality and stability guarantees. In *Proc. 49th IEEE Conference on Decision and Control (CDC)*, pages 7272–7277, 2010.

[42] C. Gouveia, J. Moreira, C. L. Moreira, and J. A. Peças Lopes. Coordinating storage and demand response for microgrid emergency operation. *IEEE Transactions on Smart Grid*, 4(4):1898–1908, 2013.

[43] L. Grüne. Economic receding horizon control without terminal constraints. *Automatica*, 49(3):725–734, 2013.

[44] L. Grüne. Numerische Methoden der Finanzmathematik. lecture notes, 2015.

[45] L. Grüne. Approximation properties of receding horizon optimal control. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 118(1):3–37, 2016.

[46] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control. Theory and Algorithms*. Springer London, 2011.

[47] L. Grüne and M. Stieler. Asymptotic stability and transient optimality of economic mpc without terminal conditions. *Journal of Process Control*, 24(8):1187–1196, 2014.

[48] Y. Guo, M. Pan, Y. Fang, and P. P. Khargonekar. Decentralized coordination of energy utilization for residential households in the smart grid. 4(3):1341–1350, 2013.

[49] R. Halvgaard, N. K. Poulsen, H. Madsen, and J. B. Jørgensen. Economic model predictive control for building climate control in a smart grid. In *Proc. of the IEEE PES Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–6, 2012.

[50] C. Hans, P. Sopasakis, A. Bemporad, J. Raisch, and J. Collon. Scenario-based model predictive operation control of islanded microgrids. In *Proc. of the 54th IEEE Conference on Decision and Control (CDC)*, pages 3272–3277, 2015.

[51] D. Heinemann, E. Lorenz, and M. Girodo. Forecasting of solar radiation. In E. D. Dunlop, L. Wald, and M. Suri, editors, *Solar Resource Management for Electricity Generation from Local Level to Global Scale*, pages 83–94. Nova Science Publishers, New York, 2006.

[52] C. A. Hill, M. C. Such, D. Chen, J. Gonzalez, and W. M. Grady. Battery energy storage for enabling integration of distributed solar power generation. *IEEE Transactions on Smart Grid*, 3(2):850–857, 2012.

[53] B. Houska, J. Frasch, and M. Diehl. An augmented Lagrangian based algorithm for distributed non-convex optimization. *SIAM Journal on Optimization*, pages 1–23, 2016. (accepted for publication).

[54] K. M. M. Huq, M. E. Baran, S. Lukic, and O. E. Nare. An energy management system for a community energy storage system. In *Proc. IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 2759–2763, September 2012.

[55] D. Jakovetić, J. Xavier, and J. M. F. Moura. Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146, 2014.

[56] F. Katiraei and J. R. Aguero. Solar PV integration challenges. *IEEE Power and Energy Magazine*, 9(3):62–71, 2011.

[57] W. Kempton and J. Tomić. Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy. *Journal of power sources*, 144(1):280–294, 2005.

[58] S. S. Kia, J. Cortés, and S. Martínez. Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication. *Automatica*, 55:254–264, 2015.

[59] M. Kraning, E. Chu, J. Lavaei, and Stephen S. P. Boyd. Dynamic network energy management via proximal message passing. *Foundations and Trends in Optimization*, 1(2):70–122, 2013.

[60] G. K. H. Larsen, N. D. van Foreest, and J. M. A. Scherpen. Distributed control of the power supply-demand balance. *IEEE Transactions on Smart Grid*, 4(2):828–836, 2013.

[61] G. K. H. Larsen, N. D. van Foreest, and J. M. A. Scherpen. Distributed MPC applied to a network of households with micro-CHP and heat storage. *IEEE Transactions on Smart Grid*, 5(4):2106–2114, 2014.

[62] J. A. P. Lopes, C. L. Moreira, and A. G. Madureira. Defining control strategies for microgrids islanded operation. *IEEE Transactions on Power Systems*, 21(2):916–924, 2006.

[63] E. Lorenz, J. Hurka, D. Heinemann, and H. G. Beyer. Irradiance forecasting for the power prediction of grid-connected photovoltaic systems. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2(1):2–10, 2009.

[64] Z. Ma, D. S. Callaway, and I. A. Hiskens. Decentralized charging control of large populations of plug-in electric vehicles. *IEEE Transactions on Control Systems Technology*, 21(1):67–78, 2013.

[65] Z. Ma, S. Zou, L. Ran, X. Shi, and I. A. Hiskens. Decentralized coordination for large-scale plug-in electric vehicles in smart grid: An efficient real-time price approach. In *Proc. of the 54th IEEE Conference on Decision and Control (CDC)*, pages 5877–5882, 2015.

[66] J. M. Maciejowski. *Predictive control: with constraints*. Prentice Hall, 2002.

[67] I. Moghram and S. Rahman. Analysis and evaluation of five short-term load forecasting techniques. *IEEE Transactions on Power Systems*, 4(4):1484–1491, 1989.

[68] A.-H. Mohsenian-Rad and A. Leon-Garcia. Optimal residential load control with price prediction in real-time electricity pricing environments. *IEEE Tansactions on Smart Grid*, 1(2):120–133, 2010.

[69] A.-H. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia. Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. *IEEE Transactions on Smart Grid*, 1(3):320–331, 2010.

[70] A. Molderink, V. Bakker, M. G. C. Bosman, J. L. Hurink, and G. J. M. Smit. Domestic energy management methodology for optimizing efficiency in smart grids. In *Proc. of IEEE PowerTech Conference*, pages 1–7, 2009.

[71] A. Molina-Garciá, F. Bouffard, and D. S. Kirschen. Decentralized demand-side contribution to primary frequency control. *IEEE Transactions on Power Systems*, 26(1):411–419, 2011.

[72] M. A. Müller. *Distributed and economic model predictive control: beyond setpoint stabilization*. Logos Verlag Berlin GmbH, 2014.

[73] M. A. Müller and L. Grüne. Economic model predictive control without terminal constraints: optimal periodic operation. *Automatica*, pages 1–6, 2015. (accepted for publication).

[74] N.-K. C. Nair and N. Garimella. Battery energy storage systems: Assessment for small-scale renewable energy integration. *Energy and Buildings*, 42(11):2124–2130, 2010.

[75] I. Necoara and V. Nedelcu. On linear convergence of a distributed dual gradient algorithm for linearly constrained separable convex problems. *Automatica*, 55:209–216, 2015.

[76] A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[77] A. Nedić, A. Ozdaglar, and P. A. Parillo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.

[78] J. Nocedal and S. Wright. *Numerical optimization*. Springer, 2006.

[79] A. Nottrott, J. Kleissl, and B. Washom. Energy dispatch schedule optimization and cost benefit analysis for grid-connected, photovoltaic-battery storage systems. *Renewable Energy*, 55:230–240, 2013.

[80] V. G. Palma. *Robust updated MPC schemes*. PhD thesis, University of Bayreuth, 2014.

[81] N. Parikh and S. P. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.

[82] E. L. Ratnam, S. R. Weller, and C. M. Kellett. An optimization-based approach for assessing the benefits of residential battery storage in conjunction with solar PV. In *Proc. of the IREP Symp. Bulk Power System Dynamics and Control–IX*, pages 1–8, 2013.

[83] E. L. Ratnam, S. R. Weller, and C. M. Kellett. An optimization-based approach to scheduling residential battery storage with solar PV: Assessing customer benefit. *Renewable Energy*, 75:123–134, 2015.

[84] E. L. Ratnam, S. R. Weller, C. M. Kellett, and A. T. Murray. Residential load and rooftop PV generation: An Australian distribution network dataset. *International Journal of Sustainable Energy*, pages 1–20, 2015.

[85] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.

[86] A. J. R. Reis and A. P. A. da Silva. Feature extraction via multiresolution analysis for short-term load forecasting. *IEEE Transactions on Power Systems*, 20(1):189–198, 2005.

[87] J. Rivera, P. Wolfrum, S. Hirche, C. Goebel, and H.-A. Jacobsen. Alternating direction method of multipliers for decentralized electric vehicle charging control. In *Proc. of the 52nd IEEE Conference on Decision and Control (CDC)*, pages 6960–6965, 2013.

[88] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[89] P. Samadi, A.-H. Mohsenian-Rad, R. Schober, W. W. S. Wong, and J. Jatskevich. Optimal real-time pricing algorithm based on utility maximization for smart grid. In *Proc. of the first IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 415–420, 2010.

[90] P. Samadi, A.-H. Mohsenian-Rad, V. W. S. Wong, and R. Schober. Tackling the load uncertainty challenges for energy consumption scheduling in smart grid. *IEEE Transactions on Smart Grid*, 4(2):1007–1016, 2013.

[91] R. Scattolini. Architectures for distributed and hierarchical model predictive control – A review. *Journal of Process Control*, 19(5):723–731, 2009.

[92] G. Scutari, F. Facchinei, J.-S. Pang, and D. P. Palomar. Real and complex monotone communication games. *IEEE Transactions on Information Theory*, 60(7):4197–4231, 2014.

[93] S. Shao, F. Jahanbakhsh, J. R. Agüero, and L. Xu. Integration of PEVs and PV-DG in power distribution systems using distributed energy storage – Dynamic analyses. In *Proc. IEEE PES Innovative Smart Grid Technologies (IGST)*, pages 1–6, 2013.

[94] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy. Predicting solar generation from weather forecasts using machine learning. In *Proc. of the IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 528–533, 2011.

[95] C. Slamka, B. Skiera, and M. Spann. Prediction market performance and market liquidity: A comparison of automated market makers. *IEEE Transactions on Engineering Management*, 60(1):169–185, 2013.

[96] E. D. Sontag. *Mathematical control theory: deterministic finite dimensional systems*. Springer-Verlag, 1998.

[97] M. Sterner and I. Stadler. *Energiespeicher für erneuerbare Energiesysteme*. Springer, 2014.

[98] D. Tran and A. M. Khambadkone. Energy management for lifetime extension of energy storage system in micro-grid applications. *IEEE Tansactions on Smart Grid*, 4(3):1289–1296, 2013.

[99] J. N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, MIT, Cambridge, MA, USA, 1984.

[100] J. S. Vardakas, N. Zorba, and C. V. Verikoukis. A survey on demand response programs in smart grids: Pricing methods and optimization algorithms. *IEEE Communications Surveys & Tutorials*, 17(1):152–178, 2015.

[101] A. Veit, Y. Xu, R. Zheng, N. Chakraborty, and K. Sycara. Demand side energy management via multiagent coordination in consumer cooperatives. *Journal of Artificial Intelligence Research*, 50:885–922, 2014.

[102] P. Vytelingum, T. D. Voice, S. D. Ramchurn, A. Rogers, and N. R. Jennings. Agent-based micro-storage management for the smart grid. In *Proc. of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 39–46. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[103] A. Wächter and L. T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[104] Y. Wang, Q. Xia, and C. Kang. Secondary forecasting based on deviation analysis for short-term load forecasting. *IEEE Transactions on Power Systems*, 26(2):500–507, 2011.

[105] J. C. Willems. Dissipative dynamical systems part I: General theory. *Archive for Rational Mechanics and Analysis*, 45(5):321–351, 1972.

[106] J. C. Willems. Dissipative dynamical systems part II: Linear systems with quadratic supply rates. *Archive for Rational Mechanics and Analysis*, 45(5):352–393, 1972.

[107] K. Worthmann, P. Braun, M. Proch, J. Schlüchtermann, and J. Pannek. On contractual periods in supplier development. In *Proc. of the 7th IFAC Conference on Management and Control of Production and Logistics (MCPL)*, pages 60–65, 2016.

[108] K. Worthmann, C. M. Kellett, P. Braun, L. Grüne, and S. R. Weller. Distributed and decentralized control of residential energy systems incorporating battery storage. *IEEE Transactions on Smart Grid*, 6(4):1914–1923, 2015.

[109] K. Worthmann, C. M. Kellett, L. Grüne, and S. R. Weller. Distributed control of residential energy systems using a market maker. In *Proc. of the 19th IFAC World Congress*, pages 11641–11646, 2014.

[110] K. Worthmann, M. Proch, P. Braun, J. Schlüchtermann, and J. Pannek. Towards dynamic contract extension in supplier development. *Logistics Research*, 9(14):1–12, 2016.

[111] C. Wu, H. Mohsenian-Rad, and J. Huang. Vehicle-to-aggregator interaction game. *IEEE Transactions on Smart Grid*, 3(1):434–442, 2012.

[112] P. Yang, P. Chavali, and A. Nehorai. Parallel autonomous optimization of demand response with renewable distributed generators. In *Proc. of the 3rd IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 55–60, 2012.

[113] A. Yona, T. Senjyu, A. Y. Saber, T. Funabashi, H. Sekine, and C.-H. Kim. Application of neural network to one-day-ahead 24 hours generating power forecasting for photovoltaic system. In *Proc. of the International Conference on Intelligent Systems Applications to Power Systems (ISAP)*, pages 1–6, 2007.

[114] M. Zhu and S. Martínez. On distributed convex optimization under inequality and equality constraints. *IEEE Transactions on Automatic Control*, 57(1):151–164, 2012.

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die von mir angegebenen Quellen und Hilfsmittel verwendet habe.

Weiterhin erkläre ich, dass ich die Hilfe von gewerblichen Promotionsberatern bzw. Promotionsvermittlern oder ähnlichen Dienstleistern weder bisher in Anspruch genommen habe, noch künftig in Anspruch nehmen werde.

Zusätzlich erkläre ich hiermit, dass ich keinerlei frühere Promotionsversuche unternommen habe.

Bayreuth, den

_____

(Philipp Braun)