Distributed optimization based control on the example of microgrids

Philipp Braun, Philipp Sauerteig, and Karl Worthmann

Abstract Model Predictive Control (MPC) is nowadays one of the most successful advanced process control methodologies and is used in a wide range of applications. While originally limited to processes with slow dynamics and a limited number of states, the applicability of MPC schemes increased dramatically over the last years due the performance of modern microchips and the concurrent advancements of mathematical optimization, in particular distributed optimization. In this paper we outline the ideas of distributed optimization schemes embedded in MPC implementations on the example of the dual ascent algorithm and the alternating direction method of multipliers. The performance and the properties of the resulting distributed optimization based control schemes are illustrated on the example of a network of distributed energy systems. In particular, the overall power demand of the network is optimized by using flexibilities resulting from distributed storage devices and controllable loads.

1 Introduction

Model Predictive Control (MPC) is nowadays the most successful advanced process control methodology, which can be concluded from its wide range of

Philipp Braun (philipp.braun@newcastle.edu.au) is with the School of Electrical Engineering, University of Newcastle, Newcastle, Australia, and with the Department of Applied Mathematics, University of Bayreuth, Bayreuth, Germany.
Philipp Sauerteig (philipp.sauerteig@tu-ilmenau.de) and Karl Worthmann (karl.worthmann@tu-ilmenau.de) are with the Institute for Mathematics, Technische Universität Ilmenau, Ilmenau, Germany.

P. Sauerteig and K. Worthmann are supported by the Federal Ministry of Education and Research within the subproject *Consistent Optimization and Stabilization of electrical networked systems* of the program *Mathematik für Innovationen* als Beitrag zur Energiewende.

applications [34, 11]. The key factors for its success are the simplicity of the basic idea – measure/estimate the current state, predict and optimize the future plant behavior to compute an input signal, and repeat this procedure ad infinitum – and its capability to deal with constrained nonlinear multi-input multi-output systems. For details on MPC as well as the development of different MPC concepts over the last decades we refer to the monographs [22, 37], the review article [28], and the references therein.

While the range of applications was limited due to the real-time requirements on the optimization step in MPC, see, e.g. [15], recent developments [40, 26] in mathematical programming allowed to further extend the scope of application. Herein, distributed optimization, see [3, 5, 7], plays a major role to overcome the computational limitations. To this end, new distributed MPC schemes have been developed [38, 39] to embed iterative distributed optimization algorithms within the MPC loop such that performance guarantees can be concluded [19, 20, 21]. The potential of distributed optimization in prediction based control is outlined in the review article [13] and in the book [30].

The interplay of MPC and iterative distributed optimization schemes is both the starting point and the focus of our work. After a brief introduction, we give a conceptual review of dual decomposition to explicate the main ideas behind distributed optimization. The section is concluded by presenting and discussing a dual ascent algorithm [10], specified in terms of an optimal control problem for distributed dynamical systems. While dual ascent already allows for a significant speed-up, see, e.g. [19], and is very flexible with respect to the structure of the interconnection of dynamical systems, it requires – in general – restrictive assumptions for convergence to the (global) optimum. This drawback can, e.g., be mitigated by using the Alternating Direction Method of Multiplies (ADMM), which is concisely outlined in the successor section. We refer to [7] for a recently published more detailed exposition. The key point demonstrated by doing so is that distributed optimization algorithms are typically tailored for a particular setting, in which their performance – in combination with other features like plug-and-play capability – is extremely competitive with respect to scalability, which ensures that the algorithm remains computationally tractable also for large-scale systems.

In this paper, the proposed algorithms are tailored to a smart-grid application to illustrate their properties within the MPC closed loop. In the context of the application, the overall energy demand of a network of RESs is optimized. In this field of application distributed optimization is nowadays very popular, see, e.g. [12, 42] or the review article [36] for further comments on the application. We consider a setting in which Residential Energy Systems (RESs) are connected to the grid via a Central Entity (CE), an operator of a transmission grid, for example. Since the number of RESs is typically large, distributed optimization is used to alleviate the computational burden resulting from solving the optimization problems in MPC, and to maintain flexibility with respect to changes in the network structure of the smart grid. Here, we stick to the term Optimal Control Problem (OCP) despite the fact that we consider discrete time systems, which implies that the OCP is a finite dimensional optimization problem. This work continues our first modeling approach (accompanied by first numerical results) on the integration of controllable loads [9]. Using ADMM, we rigorously show that the proposed distributed optimization approach converges to the global minimum and, thus, lay the foundation for its embedding in distributed MPC. From an application point of view exploiting flexibilities resulting from both energy storage devices *and* controllable loads further improves the contribution to load shaving necessary due to the integration of renewable (and, thus, highly fluctuating) energy sources. Moreover, we investigate the distributed MPC scheme applied to optimize the *energy exchange* between microgrids coupled through transmission lines. In particular, we numerically show that an exchange, on top of the optimization within the individual microgrids, is beneficial with respect to peak shaving even if losses due to the transmission are taken into account. Again, distributed optimization (ADMM) turns out to be the right tool to leverage the untapped potential resulting from this additional *flexibility.*

Throughout the paper, $\mathbb{N}_0$ and $\mathbb{N}$ denote the natural numbers including and excluding 0, and $\mathbb{Z}$ and $\mathbb{R}$ denote the integers and the real numbers, respectively. For given $n, m \in \mathbb{Z}$, $n \leq m$, we use $[n : m] = \{n, n+1, \ldots, m\}$ to define the integers from n to m to shorten the expressions. The identity matrix of appropriate dimension is denoted by I. For a sequence $\{a_1, a_2, a_3, \ldots\} \subset \mathbb{R}^n$ the shorter notation $(a^l)_{l\in\mathbb{N}} \subset \mathbb{R}^n$ is used.

2 Model Predictive Control

To illustrate the MPC concept, we consider a discrete time control system governed by the dynamics

$$x(k+1) = f(x(k), u(k)), \qquad x(0) = x^0, \tag{1}$$

with a continuous map $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$. Here, $x(k)$ and $u(k)$ denote the state of the system and the control input at time instant k, $k \in \mathbb{N}_0$, respectively. In addition, we assume that the states and the control have to respect certain constraints described through compact sets with non-empty interior, i.e. $x(k) \in \mathbb{X} \subset \mathbb{R}^{n_x}$ and $u(k) \in \mathbb{U} \subset \mathbb{R}^{n_u}$, $k \in \mathbb{N}_0$.

With these definitions, MPC is used to determine a state feedback law $\mu : \mathbb{N}_0 \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$, through stage costs $\ell : \mathbb{N}_0 \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ defining a performance measure, such that the closed-loop system

$$x^{\mathrm{cl}}(k+1) = f(x^{\mathrm{cl}}(k), \mu(k, x^{\mathrm{cl}}(k))), \qquad x^{\mathrm{cl}}(0) = x^0 \tag{2}$$

is *recursively feasible* and satisfies *infinite horizon performance* properties.

- *Recursive feasibility* refers to properties of the closed-loop solution $x^{\mathrm{cl}}(k)$, $k \in \mathbb{N}_0$. If $x^{\mathrm{cl}}(k)$ is feasible, i.e. $x^{\mathrm{cl}}(k) \in \mathbb{X}$ holds, the current feedback value and the successor state are also feasible, i.e.

$$\mu(k, x^{\mathrm{cl}}(k)) \in \mathbb{U} \qquad \text{and} \qquad x^{\mathrm{cl}}(k+1) \in \mathbb{X}.$$

  Thus, recursive feasibility implies well-posedness of the closed loop (2) provided initial feasibility $x^{\mathrm{cl}}(0) = x^0 \in \mathbb{X}$ holds, see, e.g. [6, 37].
- The *infinite horizon performance* sums up the stage cost along the MPC closed-loop trajectory

$$\sum_{k=0}^{\infty} \ell(k, x^{\mathrm{cl}}(k), \mu(k, x^{\mathrm{cl}}(k))) \tag{3}$$

  in cases where the limit exists. If the limit does not exist, or in applications where the average is more meaningful, the infinite horizon performance can alternatively be measured by

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} \ell(k, x^{\mathrm{cl}}(k), \mu(k, x^{\mathrm{cl}}(k))). \tag{4}$$

  The performance index (3) is typically used in set point stabilization, see, e.g. [22, 37], while the average version (4) is often used in *economic MPC*, see, e.g. [2, 31]. The main difference is that the stage cost is positive definite with respect to a steady state for set point stabilization while in economic MPC turnpike properties [14] in combination with (strict) dissipativity [18] are used to deduce rigorous assertions with respect to the closed-loop performance.

In general, the problem of finding a feedback law that is optimal regarding the infinite horizon performances (3) or (4) is computationally intractable. MPC is a technique to approximately solve this task. To this end, an optimization problem on a predefined finite time horizon $N \in \mathbb{N}_{\geq 2}$ is considered in a receding horizon fashion, leading to the following basic MPC scheme. Here, the term *receding horizon* means that the feedback law μ is computed iteratively at each time step $k \in \mathbb{N}_0$ (and only for the current state $x(k)$).

Basic MPC Scheme

1. For $k \in \mathbb{N}_0$, measure the current state $\hat{x} := x(k)$.
2. Minimize the cost functional

$$J_N(k, \hat{x}, \mathbf{u}) := \sum_{n=k}^{k+N-1} \ell(n, x(n), u(n))$$

where $\mathbf{u} = (u(k), u(k+1), \dots, u(k+N-1))^T$ and subject to the initial condition $x(k) = \hat{x}$, the system dynamics

$$x(n+1) = f(x(n), u(n)), \qquad n \in [k : k+N-1],$$

and the state and control constraints

$$x(n+1) \in \mathbb{X}, \quad u(n) \in \mathbb{U}, \qquad n \in [k : k+N-1],$$

to compute a minimizing sequence $\mathbf{u}^\star$.

3. Implement $\mu(k, \hat{x}) := u^\star(k)$, shift the horizon forward in time, i.e. set $k = k+1$, and go to Step 1.

While the basic MPC algorithm is well understood, already time varying stage costs ℓ lead to a significantly more involved analysis, see, e.g. [23, 27]. Moreover, a careful design of the stage cost ℓ is important to rigorously deduce closed-loop properties as shown in [35, 32].

In this paper, we want to apply the idea of the MPC scheme to a distributed setting. To be more precise, let $\mathcal{I}$, $\mathcal{I} \in \mathbb{N}_{\geq 2}$, dynamically decoupled systems

$$x_i(k+1) = f_i(x_i(k), u_i(k)), \qquad x_i(0) = x_i^0, \qquad i \in [1 : \mathcal{I}], \tag{5}$$

be given where $x_i \in \mathbb{X}_i \subset \mathbb{R}^{n_{x_i}}$, $u_i \in \mathbb{U}_i \subset \mathbb{R}^{n_{u_i}}$, and the dimensions of the overall system are given by $n_x = \sum_{i=1}^{\mathcal{I}} n_{x_i}$, and $n_u = \sum_{i=1}^{\mathcal{I}} n_{u_i}$, respectively.

Even though the individual systems are not coupled through their dynamics, we assume that the performance of the overall dynamics depends on the individual decisions taken by the subsystems $i \in [1 : \mathcal{I}]$. To incorporate this into the model, we assume the presence of a Central Entity (CE) such that each subsystem may communicate with the CE (star shaped topology) via

$$z_i(k) = h_i(k, x_i(k), u_i(k)) \tag{6}$$

with communication variables $z_i \in \mathbb{R}^p$ for all $i \in [1 : \mathcal{I}]$. The functions $h_i : \mathbb{N}_0 \times \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \to \mathbb{R}^p$ are assumed to be continuous and can depend on the current time step $k \in \mathbb{N}_0$ as well as the local states and the local control, for all $i \in [1 : \mathcal{I}]$. With these definitions the coupling between the individual systems can be described through global stage costs

$$\hat{\ell} : \mathbb{N}_0 \times \mathbb{R}^{p\mathcal{I}} \to \mathbb{R}, \qquad (k, z) \mapsto \hat{\ell}(k, z) \tag{7a}$$

involving the communication variables $z := (z_1, \dots, z_{\mathcal{I}})^T$. The stage costs (7a) capture the performance of the overall system, or equivalently, they capture the costs enforced by the CE. In addition, every system $i \in [1 : \mathcal{I}]$ can have local performance measures defined through local stage costs

$$\ell_i : \mathbb{N}_0 \times \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \to \mathbb{R}, \qquad (k, x_i, u_i) \mapsto \ell_i(k, x_i, u_i) \quad \forall i \in [1 : \mathcal{I}], \tag{7b}$$

which in combination with the global stage costs (7a) lead to multi-objective optimization problems [17] or multi-objective MPC formulations [24] in the control context.

The overall optimal control problem (OCP) to be solved at every time instant $k \in \mathbb{N}_0$ in Step 2 of the MPC scheme can be summarized by the following optimization problem where the notation $\mathbf{x} := (x(k), x(k+1), \dots, x(k+N-1))^T$, $\mathbf{u} := (u(k), u(k+1), \dots, u(k+N-1))^T$ is used.

$$\boxed{\begin{array}{l}\text{Minimize the cost functional}\\ J_N(k,\hat{x},\mathbf{u}) := \displaystyle\sum_{n=k}^{k+N-1} \left(\alpha \cdot \hat{\ell}(n, z(n)) + \frac{1-\alpha}{\mathcal{I}} \cdot \sum_{i=1}^{\mathcal{I}} \ell_i(n, x_i(n), u_i(n)) \right)\\ \text{w.r.t. } u(k), u(k+1), \dots, u(k+N-1) \text{ subject to } x(k) = \hat{x},\\ \text{the equalities (5) and (6), the state and control constraints}\\ x_i(n+1) \in \mathbb{X}_i(n),\ u_i(n) \in \mathbb{U}_i(n) \text{ for } n \in \{k, k+1, \dots, k+N-1\}\end{array}} \tag{8}$$

The parameter $\alpha \in [0,1]$, which we assume to be set and defined by the CE, shifts the emphasis on the performance of the individual subsystems encoded in ℓ_i, $i \in [1 : \mathcal{I}]$, to a network-wide objective $\hat{\ell}$ by choosing α between 0 and 1. The overall distributed MPC algorithm incorporating the OCP (8) is summarized in Algorithm 1. While Algorithm 1 can be easily written down,

Algorithm 1 Distributed MPC algorithm

```
Input: Time horizon $N \in \mathbb{N}$, number of systems $\mathcal{I} \in \mathbb{N}$, stage costs $\hat{\ell}$, $\ell_i$, $i \in [1 : \mathcal{I}]$,
and a weighting parameter $\alpha \in [0,1]$.
Initialization: Set $k = 0$.
Main loop: For $k \in \mathbb{N}_0$
(1) For all $i \in [1 : \mathcal{I}]$, measure the current states $\hat{x}_i := x_i(k)$.
(2) Solve the OCP (8) to obtain minimizing sequences
        $\mathbf{u}_i^\star = (u_i^\star(k), \dots, u_i^\star(k+N-1))^T \quad \forall\, i \in [1 : \mathcal{I}]$.
(3) For all $i \in [1 : \mathcal{I}]$, implement $\mu_i(k, \hat{x}_i) := u_i^\star(k)$, shift the horizon forward in time,
    i.e. set $k = k + 1$, and go to Step (1).
```

Step (2) requires particular care due to the real-time requirements in MPC and to maintain the decoupled structure of the individual subsystems. Hence, the goal of the next section is to outline algorithms, which can be suitably adapted such that Step (2) is computationally tractable and flexible with respect to changes in local system dynamics. Here, iterative distributed optimization schemes play a major role to mitigate the computational burden on the one hand and, on the other hand, allow for a premature stop after a few iterations while still ensuring feasibility and achieving a competitive performance.

3 Distributed optimization

To compute an MPC feedback law, the OCP (8) of the form

$$\min_{u_i(n),x_i(n),z_i(n)} \sum_{n=k}^{k+N-1} \Big(\alpha \cdot \hat{\ell}(n,(z_1,\ldots,z_{\mathcal{I}})^T(n)) + \tfrac{1-\alpha}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \ell_i(n,x_i(n),u_i(n))\Big) \tag{9a}$$

$$\begin{aligned}
\text{s.t.}\quad & x_i(k) = \hat{x}_i, && \text{(9b)}\\
& x_i(n+1) = f_i(x_i(n),u_i(n)), && \text{(9c)}\\
& z_i(n) = h_i(n,x_i(n),u_i(n)), && \text{(9d)}\\
& x_i(n+1) \in \mathbb{X}_i(n+1), \quad u_i(n) \in \mathbb{U}_i(n), && \text{(9e)}\\
& \qquad \forall\, i \in [1:\mathcal{I}], \quad \forall\, n \in [k:k+N-1].
\end{aligned}$$

has to be solved at every time instant k, $k \in \mathbb{N}_0$. To this end, an efficient optimization algorithm is essential to meet the real-time requirements of the MPC scheme proposed in Algorithm 1. For applications with a large number of systems $\mathcal{I}$, $\mathcal{I} \in \mathbb{N}$, scalability and flexibility with respect to $\mathcal{I}$ are indispensable properties for the algorithm design, see also [43, 29]. The latter also includes changes in the particular system dynamics of the individual systems. In this context iterative distributed optimization schemes regained considerable attention over the last years.

In this section, we discuss distributed optimization schemes, and in particular hierarchical distributed optimization schemes, and their embedding within the MPC Algorithm 1. In Section 3.1, we discuss the basic dual decomposition algorithm. While the convergence results are restrictive, dual decomposition nicely illustrates the main idea of distributed optimization in general. The Alternating Direction Method of Multipliers (ADMM) presented in Section 3.2 extends the ideas of dual decomposition and is less restrictive with respect to the stage costs ℓ. We conclude this chapter in Section 3.3 by discussing beneficial properties of iterative optimization schemes within MPC.

Since we are interested in the solution of a static optimization problem (9) in this section, we simplify the notion before we continue. We collect the information of the variables x_i, u_i, z_i, $i \in [1:\mathcal{I}]$, over the prediction horizon at time $k \in \mathbb{N}$ in a single variable

$$\mathbf{y} = \begin{pmatrix} \mathbf{y_1} \\ \vdots \\ \mathbf{y}_\mathcal{I} \end{pmatrix} \quad \text{with} \quad \mathbf{y}_i = \begin{pmatrix} x_i(k) & u_i(k) & z_i(k) \\ \vdots & \vdots & \vdots \\ x_i(k+N-1) & u_i(k+N-1) & z_i(k+N-1) \end{pmatrix}.$$

Even though the variables $z_i(n)$, $i \in [1 : \mathcal{I}]$, $n \in [k : k+N-1]$ are contained in the definition of the variable $\mathbf{y}$ we additionally define

$$\mathbf{z} = \begin{pmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_\mathcal{I} \end{pmatrix} \in \mathbb{R}^{pN\mathcal{I}} \quad \text{with} \quad \mathbf{z}_i = \begin{pmatrix} z_i(k) \\ \vdots \\ z_i(k+N-1) \end{pmatrix}$$

to collect the data of the shared variables. For simplicity of exposition, the k-dependency is dropped in the variables $\mathbf{y}$ and $\mathbf{z}$ in this section.

Remark 1. Note that the variables $\mathbf{y}_i$ only need to contain the information of the variables x_i, u_i, z_i. By doing so, we do not restrict ourselves to a particular formulation of the optimization problem. For example in MPC problems with linear dynamics (1) it is common to optimize with respect to $\mathbf{u}$ and to remove the state vector $\mathbf{x}$ from the set of unknowns, see, e.g. [22, Chapter 12.1].

With these definitions of the vector of unknowns, the stage costs can be summarized in the functions

$$\phi_i(\mathbf{y}_i; k) = \sum_{n=k}^{k+N-1} \ell_i(n, x_i(n), u_i(n)) \qquad \forall\, i \in [1 : \mathcal{I}] \tag{10}$$

$$\text{and} \quad \hat{\phi}(\mathbf{z}; k) = \sum_{n=k}^{k+N-1} \hat{\ell}(n, (z_1, \ldots, z_\mathcal{I})^T(n)). \tag{11}$$

Similarly, the constraints (9b) to (9e) are summarized in the sets

$$\mathbb{D}_i(k) = \left\{ \mathbf{y}_i \;\middle|\; \begin{array}{l} x_i(k) = \hat{x}_i, \\ x_i(n+1) = f_i(x_i(n), u_i(n)), \\ z_i(n) = h_i(n, x_i(n), u_i(n)), \\ x_i(n+1) \in \mathbb{X}_i(n+1), \quad u_i(n) \in \mathbb{U}_i(n), \\ \qquad\quad \forall\, n \in [k : k+N-1]. \end{array} \right\} \tag{12}$$

for all $i \in [1 : \mathcal{I}]$ and $\mathbb{D}(k) = \mathbb{D}_1(k) \times \ldots \times \mathbb{D}_\mathcal{I}(k)$. Combining all these definitions, the OCP (9) can simply be written as

$$\min_{\mathbf{y}_i \in \mathbb{D}_i} \quad \alpha \cdot \hat{\phi}(\mathbf{z}; k) + \tfrac{1-\alpha}{\mathcal{I}} \cdot \textstyle\sum_{i=1}^{\mathcal{I}} \phi_i(\mathbf{y}_i; k). \tag{13}$$

Coupling of the variables takes place in the function $\hat{\phi}$. Otherwise, the optimization problem (13) could be split up into $\mathcal{I}$ independent optimization problems. Next, we present a solution technique for the optimization problem based on iterative distributed optimization and explicit conditions on the function ϕ and the set $\mathbb{D}(k)$ such that convergence to the global solution is guaranteed.

3.1 Dual decomposition

Dual decomposition is a relatively old concept with its roots in the 1960s and is discussed in many papers and monographs. Here, we follow the exposition given in the monographs [5, 4] to introduce concepts for our particular setting and refer to these books for details and references.

The coupling in the objective function (9a) can be eliminated by introducing a new variable $\mathbf{a} = \mathbf{z}$ (clearly also $\mathbf{a}_i = \mathbf{z}_i$ holds for all $i \in [1 : \mathcal{I}]$) linked to the Central Entity (CE) and, then, shifting the coupling from the objective function to the constraints

$$\min_{\mathbf{y}_i \in \mathbb{D}_i(k),\, \mathbf{a} \in \mathbb{R}^{\mathcal{I}pN}} \quad \alpha \hat{\phi}(\mathbf{a}; k) + \tfrac{1-\alpha}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \phi_i(\mathbf{y}_i; k) \tag{14a}$$

$$\text{s.t.} \quad \mathbf{z}_i - \mathbf{a}_i = 0 \qquad \forall\, i \in [1 : \mathcal{I}]. \tag{14b}$$

The advantage of the reformulation is that in the Lagrangian

$$\begin{aligned} \mathcal{L}(\mathbf{y}, \mathbf{a}, \lambda; k) &= \alpha \hat{\phi}(\mathbf{a}; k) + \tfrac{1-\alpha}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \phi_i(\mathbf{y}_i; k) + \textstyle\sum_{i=1}^{\mathcal{I}} \lambda_i^T (\mathbf{z}_i - \mathbf{a}_i) \\ &= \Big(\alpha \hat{\phi}(\mathbf{a}; k) - \textstyle\sum_{i=1}^{\mathcal{I}} \lambda_i^T \mathbf{a}_i \Big) + \textstyle\sum_{i=1}^{\mathcal{I}} \big(\tfrac{1-\alpha}{\mathcal{I}} \phi_i(\mathbf{y}_i; k) + \lambda_i^T \mathbf{z}_i \big) \end{aligned} \tag{15}$$

the optimization variables $\mathbf{a}$ and $\mathbf{y}$ are separated in independent functions. Thus, under appropriate conditions on the objective functions and the local constraints, an optimal solution to the primal problem (14) can be obtained by solving the (unrestricted) dual problem

$$\max_{\lambda \in \mathbb{R}^{p\mathcal{I}N}} \ \psi(\lambda; k) \quad \text{with dual function } \psi(\lambda; k) = \inf_{\substack{\mathbf{y} \in \mathbb{D}(k) \\ \mathbf{a} \in \mathbb{R}^{\mathcal{I}pN}}} \mathcal{L}(\mathbf{y}, \mathbf{a}, \lambda; k). \tag{16}$$

The following theorem characterizes the interplay of the primal and the dual problem, see, e.g. [5, Appendix C].

Theorem 1. *If the primal problem* (14) *has an optimal solution* $(\mathbf{y}^\star, \mathbf{a}^\star)$, *then also the dual problem* (16) *has an optimal solution* $\lambda^\star$ *(and vice versa) and the optimal values coincide, i.e.,*

$$\alpha \hat{\phi}(\mathbf{a}^\star; k) + \tfrac{1-\alpha}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \phi_i(\mathbf{y}_i^\star; k) = \psi(\lambda^\star; k).$$

Moreover, the pair $(\mathbf{y}^\star, \mathbf{a}^\star, \lambda^\star)$ *is optimal for* (14) *and* (16) *if and only if the saddle point condition*

$$\mathcal{L}(\mathbf{y}^\star, \mathbf{a}^\star, \lambda; k) \leq \mathcal{L}(\mathbf{y}^\star, \mathbf{a}^\star, \lambda^\star; k) \leq \mathcal{L}(\mathbf{y}, \mathbf{a}, \lambda^\star; k) \tag{17}$$

is satisfied for all $\mathbf{y} \in \mathbb{D}(k)$*,* $\mathbf{a} \in \mathbb{R}^{p\mathcal{I}N}$ *and for all* $\lambda \in \mathbb{R}^{p\mathcal{I}N}$*.*

As a consequence of Theorem 1, if an optimal solution to the dual problem (16) is known, an optimal solution of the primal problem can be computed by solving the optimization problems

$$\mathbf{y}_i^\star \in \operatorname*{arg\,min}_{\mathbf{y}_i \in \mathbb{D}_i(k)} \tfrac{1-\alpha}{\mathcal{I}} \phi_i(\mathbf{y}_i; k) + \mathbf{z}_i^T \lambda_i^\star$$

for all $i \in [1 : \mathcal{I}]$ in parallel. An optimal solution of the dual problem can be found by iteratively updating the Lagrange multipliers. Hence, under appropriate conditions on the primal problem (14) an optimal solution $(\mathbf{y}^\star, \mathbf{a}^\star)$ and $\lambda^\star$ can be computed through the hierarchical dual ascent Algorithm 2.

Algorithm 2 Hierarchical dual ascent algorithm

Input: Step size $c \in \mathbb{R}> 0$, number of systems $\mathcal{I} \in \mathbb{N}$, objective functions $\hat{\phi}$, ϕ_i, $i \in [1 : \mathcal{I}]$, weighting parameter $\alpha \in [0, 1]$, and maximal number of iterations $l_{\max} \in \mathbb{N}$.
Initialization: Set $l = 0$ and initialize $\lambda^0 \in \mathbb{R}^{p\mathcal{I}\mathbb{N}}$ (arbitrarily).
Main loop: For $l \leq l_{\max}$

1. The individual systems compute an optimal solution of

$$\mathbf{y}_i^l \in \operatorname*{arg\,min}_{\mathbf{y}_i \in \mathbb{D}_i(k)} \tfrac{1-\alpha}{\mathcal{I}} \phi_i(\mathbf{y}_i; k) + \mathbf{z}_i^T \lambda_i^l \qquad \forall\, i \in [1 : \mathcal{I}], \tag{18a}$$

   in parallel and send $\mathbf{z}_i^l$ to the CE.

2. The CE computes an optimal solution of

$$\mathbf{a}^l \in \operatorname*{arg\,min}_{\mathbf{a} \in \mathbb{R}^{p\mathcal{I}N}} \alpha\hat{\phi}(\mathbf{a}; k) - \textstyle\sum_{i=1}^{\mathcal{I}} \mathbf{a}_i^T \lambda_i^l. \tag{18b}$$

3. The CE updates the Lagrange multipliers

$$\lambda_i^{l+1} = \lambda_i^l + c(\mathbf{z}_i^l - \mathbf{a}_i^l) \qquad \forall\, i \in [1 : \mathcal{I}], \tag{18c}$$

   and broadcasts λ_i^{l+1}. Afterwards, set $l = l + 1$ and go to Step 1.

Convergence of Algorithm 2 can, e.g., be guaranteed by Theorem 2.

Theorem 2 ([4, Prop. 1.2.3]). *Consider the primal problem* (14) *and assume that the objective function is strongly convex*[1] *with respect to the variables* $(\mathbf{y}, \mathbf{a})$ *with parameter* $\chi > 0$*. Moreover, assume that the primal problem* (14) *is feasible, and the sets* $\mathbb{D}_i(k)$ *are convex and closed for all* $i \in [1 : \mathcal{I}]$*. Then, for a constant stepsize* $c \in (0, \chi)$ *the iterates* $(\mathbf{y}^l)_{l\in\mathbb{N}}$*,* $(\mathbf{a}^l)_{l\in\mathbb{N}}$*,* $(\lambda^l)_{l\in\mathbb{N}}$ *computed according to Algorithm* (18) *satisfy*

[1] A function $F : \mathbb{R}^n \to \mathbb{R}$ is said to be *strongly convex* with parameter $\chi > 0$ if $F(\mu x + (1-\mu)y) \leq \mu F(x) + (1-\mu)F(y) - \frac{\chi}{2}\mu(1-\mu)\|x-y\|_2^2$ holds for all $x, y \in \mathbb{R}^n$ and all $\mu \in [0, 1]$.

$$\lim_{l\to\infty} \mathbf{z}^l = \mathbf{z}^\star, \qquad \lim_{l\to\infty} \mathbf{a}^l \to \mathbf{a}^\star, \qquad \lim_{l\to\infty} \lambda^l \to \lambda^\star.$$

for arbitrary initial values $\mathbf{y}^0 \in \mathbb{D}(k)$*,* $\mathbf{a}^0 \in \mathbb{R}^{p\mathcal{I}N}$*,* $\lambda^0 \in \mathbb{R}^{p\mathcal{I}N}$*.*

The name *dual ascent* refers to the fact that (18c) is a gradient step in the direction of the dual function ψ, where $\psi(\lambda^{l+1}) > \psi(\lambda^l)$ holds if the condition on the stepsize is satisfied. The maximal stepsize c depends on the convexity of the objective function as well as on the definition of the linear coupling constraints (14b), see [5, Prop. 6.1.1] for details.

The dual ascent Algorithm 2 allows to split the optimization problem in tasks performed by the individual systems in parallel (see Algorithm 2, Step 1), and tasks performed by a CE (Step 2 and 3). The local optimization problems (18a) depend on local information and on the dual variables λ_i, $i \in [1 : \mathcal{I}]$. Thus, the dimension and the complexity of the optimization problem solely depends on the parameters defining the i-th system, $i \in [1 : \mathcal{I}]$. Note that also the number of systems $\mathcal{I} \in \mathbb{N}$ does not need to be known by the individual systems. To achieve this, the scaled Lagrange multipliers $\widetilde{\lambda}_i = \lambda_i \mathcal{I}$, $i \in [1 : \mathcal{I}]$, are broadcast by the CE and individual systems need to solve the optimization problem

$$\mathbf{y}_i^l \in \operatorname*{arg\,min}_{\mathbf{y}_i \in \mathbb{D}_i(k)} \quad (1-\alpha)\phi_i(\mathbf{y}_i; k) + \mathbf{z}_i^T \widetilde{\lambda}_i^l \tag{19}$$

for $i \in [1 : \mathcal{I}]$, instead of (18a).

The dimension of the optimization problem (18b) linked to the CE depends on the number of systems $\mathcal{I} \in \mathbb{N}$. Nevertheless, note that the variables $\mathbf{a} \in \mathbb{R}^{p\mathcal{I}N}$ are unconstrained and thus an optimal solution can be computed efficiently or even explicitly by the CE.

Note that depending on the network structure and the interconnection of the individual systems, the dimension of the variable $\mathbf{a}$ and the dimension of the Lagrange multipliers λ can be reduced (see [5, 7], for example), or in special cases they can even be made independent of the number of systems $\mathcal{I} \in \mathbb{N}$. Consider for example stage costs $\hat{\ell}$, which can be written in the form

$$\hat{\ell}\left(n; (z_1, \ldots, z_{\mathcal{I}})^T\right) = \overline{\ell}\left(n; \tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} z_i\right)$$

and thus, the stage costs only depend on the average taken over the communication variables z_i, $i \in [1 : \mathcal{I}]$. Hence, we can define the variables $\overline{a}(n) = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} z_i(n)$ for $n \in [k : k + N - 1]$, which are the elements of the vector $\overline{\mathbf{a}}$ and modify the cost function (11):

$$\overline{\phi}(\overline{\mathbf{a}}; k) = \hat{\phi}(\mathbf{z}; k) = \textstyle\sum_{n=k}^{k+N-1} \overline{\ell}\left(n; \tfrac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} z_i(n)\right). \tag{20}$$

Then, the primal optimization problem (14) becomes

$$\min_{\substack{\mathbf{y}_i\in\mathbb{D}_i(k)\\ \overline{\mathbf{a}}\in\mathbb{R}^{pN}}} \alpha\overline{\phi}(\overline{\mathbf{a}};k) + \tfrac{1-\alpha}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}}\phi_i(\mathbf{y}_i;k) \qquad \text{s.t.} \quad \tfrac{1}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}}\mathbf{z}_i - \overline{\mathbf{a}} = 0 \tag{21}$$

and the dual ascent scheme (18) is of the form

$$\mathbf{y}_i^l \in \operatorname*{arg\,min}_{\mathbf{y}_i\in\mathbb{D}_i(k)} (1-\alpha)\phi_i(\mathbf{y}_i;k) + \mathbf{z}_i^T\overline{\lambda}^l \qquad \forall\, i\in[1:\mathcal{I}], \tag{22a}$$

$$\overline{\mathbf{a}}^l \in \operatorname*{arg\,min}_{\overline{\mathbf{a}}\in\mathbb{R}^{pN}} \alpha\overline{\phi}(\overline{\mathbf{a}};k) - \textstyle\sum_{i=1}^{\mathcal{I}}\overline{\mathbf{a}}_i^T\overline{\lambda}^l, \tag{22b}$$

$$\overline{\lambda}^{l+1} = \overline{\lambda}^l + c(\tfrac{1}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}}\mathbf{z}_i^l - \overline{\mathbf{a}}^l). \tag{22c}$$

Here, the dimension of $\overline{\mathbf{a}}\in\mathbb{R}^{pN}$ as well as the dimension of the Lagrange multipliers $\overline{\lambda}\in\mathbb{R}^{p\mathcal{I}}$ is independent of the the number of systems. Moreover, the same information is communicated from the CE to the individual systems. In this case, the stepsize c needs to be chosen such that $c\in(0, 2\mathcal{I}\chi/[2+\mathcal{I}])$ is satisfied to guarantee convergence [16].

Overall, the dual ascent Algorithm 2 thus provides a very flexible scheme to solve OCPs embedded in the MPC Algorithm 1. However, the assumed strong convexity of the objective function is very restrictive.

3.2 The alternating direction method of multipliers

To weaken the assumption on strong convexity in Theorem 2, and thus to extend the applicability of iterative distributed optimization schemes, alternative algorithms based on the augmented Lagrangian $\mathcal{L}_\rho(\mathbf{y},\mathbf{a},\lambda;k)$ defined by

$$\begin{aligned}\mathcal{L}_\rho(\mathbf{y},\mathbf{a},\lambda;k) = {}& \alpha\hat{\phi}(\mathbf{a};k) + \tfrac{1-\alpha}{\mathcal{I}}\textstyle\sum_{i=1}^{\mathcal{I}}\phi_i(\mathbf{y}_i;k) \\ &+ \textstyle\sum_{i=1}^{\mathcal{I}}\left(\lambda_i^T(\mathbf{z}_i-\mathbf{a}_i) + \tfrac{\rho}{2}\left\|\mathbf{z}_i-\mathbf{a}_i\right\|_2^2\right),\end{aligned} \tag{23}$$

for a positive parameter $\rho>0$, have been proposed. For $\rho=0$ the original definition of the Lagrangian (15) is recovered. Observe that for a feasible solution of the optimization problem (14) the quadratic terms vanish. In contrast to the Lagrangian, the variables $\mathbf{z}$ and $\mathbf{a}$ are not decoupled (or separable) in the augmented Lagrangian. However, it is still possible, to optimize the variables $\mathbf{z}$ and $\mathbf{a}$ sequentially, which leads to the ADMM Algorithm 3 and in particular to the iteration scheme (24).

The ADMM scheme received a lot of attention over the last years, especially due to the exposition and discussion of the algorithm in [7] and [33]. The advantage of ADMM compared to dual ascent is that convergence can be shown under weaker assumptions and independently of the stepsize.

Algorithm 3 Alternating Direction Method of Multipliers (ADMM)

Input: Step size $\rho \in \mathbb{R}_{>0}$, number of systems $\mathcal{I} \in \mathbb{N}$, objective functions $\hat{\phi}$, ϕ_i, $i \in [1:\mathcal{I}]$, a weighting parameter $\alpha \in [0,1]$, and a maximal number of iterations $l_{\max} \in \mathbb{N}$.
Initialization: Set $l = 0$ and initialize $\lambda^0, \mathbf{a}^0 \in \mathbb{R}^{p\mathcal{I}\mathbb{N}}$ (arbitrarily).
Main loop: For $l \leq l_{\max}$

1. The individual systems compute an optimal solution of

$$\mathbf{y}_i^{l+1} \in \operatorname*{arg\,min}_{\mathbf{y}_i \in \mathbb{D}_i(k)} \tfrac{1-\alpha}{\mathcal{I}} \phi_i(\mathbf{y}_i;k) + \mathbf{z}_i^T \lambda_i^l + \tfrac{\rho}{2} \left\| \mathbf{z}_i - \mathbf{a}_i^l \right\|_2^2 \qquad \forall\, i \in [1:\mathcal{I}] \tag{24a}$$

   in parallel and broadcast $\mathbf{z}_i^{l+1}$ to the CE.
2. The CE computes an optimal solution of

$$\mathbf{a}^{l+1} \in \operatorname*{arg\,min}_{\mathbf{a} \in \mathbb{R}^{p\mathcal{I}N}} \alpha\hat{\phi}(\mathbf{a};k) + \textstyle\sum_{i=1}^{\mathcal{I}} -\mathbf{a}_i^T \lambda_i^l + \tfrac{\rho}{2} \left\| \mathbf{z}_i^{l+1} - \mathbf{a}_i \right\|_2^2 . \tag{24b}$$

3. The CE updates the Lagrange multipliers

$$\lambda_i^{l+1} = \lambda_i^l + \rho(\mathbf{z}_i^{l+1} - \mathbf{a}_i^{l+1}) \qquad \forall\, i \in [1:\mathcal{I}] \tag{24c}$$

   and broadcasts $(\lambda_i^{l+1}, \mathbf{a}_i^{l+1})$ to system $i \in [1:\mathcal{I}]$. Afterwards, set $l = l+1$ and go to Step 1.

Theorem 3 ([7, Sec. 3.2.1]). *Let the functions $\hat{\phi}$, ϕ_i, $i = [1:\mathcal{I}]$, be convex. Suppose there exists a saddle point $(\mathbf{y}^\star, \mathbf{a}^\star, \lambda^\star)$ of the unaugmented Lagrangian $\mathcal{L}_0$, i.e., $(\mathbf{y}^\star, \mathbf{a}^\star, \lambda^\star)$ satisfies (17) for all $\mathbf{y} \in \mathbb{D}(k)$, $\mathbf{a}, \lambda \in \mathbb{R}^{p\mathcal{I}N}$. Then, for $\mathbf{y}^0 \in \mathbb{D}(k)$, $\mathbf{a}^0 \in \mathbb{R}^{p\mathcal{I}N}$ and $\lambda^0 \in \mathbb{R}^{p\mathcal{I}N}$ and fixed stepsizes $\rho > 0$, Algorithm 3 satisfies the following properties:*

1. *The sequence $(\mathbf{z}^l - \mathbf{a}^l)_{l\in\mathbb{N}}$ converges to zero, ensuring feasibility of the optimization problem (14).*
2. *The sequence $\left(\alpha\hat{\phi}(\mathbf{a}^l;k) + \frac{1-\alpha}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \phi_i(\mathbf{y}_i^l;k)\right)_{l\in\mathbb{N}}$ converges to the optimal value of the optimization problem (14).*
3. *The dual variables $(\lambda^l)_{l\in\mathbb{N}}$ converge to the optimal dual point $\lambda^\star$.*

Theorem 3 weakens the assumptions on the objective function of Theorem 2 from strong convexity to convexity. The saddle point condition in Theorem 3 is, e.g., satisfied if the sets $\mathbb{D}_i(k)$, $i \in [1:\mathcal{I}]$, are convex and compact. Compared to the dual ascent Algorithm 2, Step 1 and Step 2 of Algorithm 3 need to be performed sequentially and cannot be performed in parallel.

Remark 2. In Algorithm 3, also the primal variables $\mathbf{a} \in \mathbb{R}^{p\mathcal{I}N}$ need to be communicated. However, the information contained in λ and $\mathbf{a}$ can be compressed in a single communication variable of dimension $p\mathcal{I}N$, which allows to achieve the same communication costs as in Algorithm 2, see the scaled version of ADMM proposed in [7, Sec. 3.1.1].

Even though it is not as obvious as in the dual ascent Algorithm 2, also the ADMM scheme allows a formulation of an Algorithm 3 where the dimension of $\mathbf{a}$ and λ is independent of the number of systems $\mathcal{I} \in \mathbb{N}$ if an objective function of the form (20) is used. In this case, the iteration scheme (24) becomes

$$\begin{aligned}
\mathbf{y}_i^{l+1} &\in \operatorname*{arg\,min}_{\mathbf{y}_i \in \mathbb{D}_i(k)} \tfrac{1-\alpha}{\mathcal{I}} \phi_i(\mathbf{y}_i; k) + \tfrac{\rho}{2} \left\| \mathbf{z}_i - \mathbf{z}_i^l + \Pi^l \right\|_2^2 \qquad \forall\, i \in [1:\mathcal{I}] && \text{(25a)}\\
\overline{\mathbf{a}}^{l+1} &\in \operatorname*{arg\,min}_{\overline{\mathbf{a}} \in \mathbb{R}^{pN}} \; \alpha \overline{\phi}(\overline{\mathbf{a}}; k) + \tfrac{\rho \mathcal{I}}{2} \left\| \tfrac{\overline{\lambda}^l}{\rho} + \tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \mathbf{z}_i^{l+1} - \overline{\mathbf{a}} \right\|_2^2 && \text{(25b)}\\
\overline{\lambda}^{l+1} &= \overline{\lambda}^l + \rho \left(\tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \mathbf{z}_i^{l+1} - \overline{\mathbf{a}}^{l+1} \right) && \text{(25c)}\\
\Pi^{l+1} &= \tfrac{\overline{\lambda}^{l+1}}{\rho} + \left(\tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \mathbf{z}_i^{l+1} - \overline{\mathbf{a}}^{l+1} \right). && \text{(25d)}
\end{aligned}$$

See [7, Section 7] or [8] for a derivation of the iteration scheme, for example.

3.3 Properties within the MPC closed loop

Algorithm 2 and 3 enable us to solve the OCPs (8) by iteratively solving smaller or less complex optimization problems independent of the number of systems. Thus, the numerical complexity of the distributed optimization algorithms strongly depends on the number of iterations needed to compute a solution within a predefined tolerance. If a good *initial guess*, e.g. initial values λ^0 and $\mathbf{a}^0$ close to the optimal solution, are available, the number of iterations is typically much smaller. Here, MPC provides a natural way to initialize λ^0 and $\mathbf{a}^0$ since at two consecutive time instants k and $k+1$, $k \in \mathbb{N}_0$, similar optimization problems are solved and only the first piece of the optimal trajectory is used to define a feedback law, while the rest is discarded. To illustrate this fact let the last iteration of the distributed optimization algorithm at time k be denoted by $\lambda_i^{l_{\max}}[k]$ and the initial value at time $k+1$ by $\lambda_i^0[k+1]$ for $i \in [1:\mathcal{I}]$. Then, we get

$$\lambda_i^{l_{\max}}[k] = \begin{pmatrix} \lambda_i^{l_{\max}}(k|k) \\ \lambda_i^{l_{\max}}(k+1|k) \\ \vdots \\ \lambda_i^{l_{\max}}(k+N-2|k) \\ \lambda_i^{l_{\max}}(k+N-1|k) \end{pmatrix} \quad \rightsquigarrow \quad \begin{pmatrix} \lambda_i^{l_{\max}}(k+1|k) \\ \vdots \\ \lambda_i^{l_{\max}}(k+N-2|k) \\ \lambda_i^{l_{\max}}(k+N-1|k) \\ 0 \end{pmatrix} = \lambda_i^0[k+1].$$

The variable $\mathbf{a}$ can be initialized in the same way. Observe that in Algorithm 2 the variables $\mathbf{a}$ and $\mathbf{y}$ do not need to be initialized since Step 1 and 2 in the first iteration $l = 0$ only depend on the initialization of λ^0. The same holds for the variable $\mathbf{y}$ in the ADMM algorithm. Typically, already a few iterations are enough to obtain a closed-loop performance, which is close to optimal.

Moreover, note that even though the iterates $\mathbf{z}_i^l$, $i \in [1 : \mathcal{I}]$, $l \in [0 : l_{\max}]$ might not be optimal, feasibility (i.e., $\mathbf{z}_i \in \mathbb{D}_i(k)$, $i \in [1 : \mathcal{I}]$) is assured in every iteration since by assumption the systems are physically decoupled. Thus, an early termination of the distributed optimization algorithm always provides a feasible solution if $\mathbb{D}(k) \neq \emptyset$ holds.

4 Case Study: Distributed MPC for small scale electricity networks

In this section we apply the distributed MPC Algorithm 1, where the optimization Step (2) is carried out with the ADMM Algorithm 3 to a small scale electricity network. The input variables (controls) represent the flexibility present in the system. In Section 4.1, locally installed energy storage devices are used for load shaping. Here, we are particularly interested in reducing fluctuations in the network-wide energy consumption. To this end, we first recapitulate the basic model of a network of Residential Energy Systems (RESs) introduced and extended in [41, 10]. Then, in Section 4.2 controllable loads are added to further increase the flexibility with respect to the demand patterns of the RESs. Here, we rigorously prove global convergence of a setting (very) similar to the one presented in [9]. Last, in Section 4.3, the model is extended to additionally optimize the energy exchange between individual microgrids (MGs) coupled through transmission lines.

In the simulations we concentrate on the performance of the overall network instead of individual performance measures. We thus set $\alpha = 1$, i.e., we use the coupling stage costs $\hat{\ell}$ (or $\overline{\ell}$, respecitvely) and neglect the individual stage costs ℓ_i, $i \in \mathbb{N}_{\mathcal{I}}$. However, the additional consideration of convex stage costs is straightforward. The numerical simulations are based on a dataset provided by the electricity distribution company Ausgrid and publicly available online [1]. From the dataset consisting of data collected from 300 residential customers with a resolution of half-hour windows we use the information of power consumption and power generation using solar photovoltaic (PV) panels and the information of controllable loads, which is available for some customers. The power consumption and the power generation is combined to obtain the power demand of a single customer at a particular time.

4.1 The basic model and its MPC formulation

Several models describing RESs have been introduced in the literature. We focus on extensions to the model discussed in [41, 10]. The dynamics of the

i-th RES, $i \in [1:\mathcal{I}]$, are described by

$$x_i(k+1) = f_i(x_i(k), u_i(k)) \quad = \varrho_i x_i(k) + T(\beta_i u_i^+(k) + u_i^-(k)) \tag{26}$$

$$z_i(k) = h_i(k, x_i(k), u_i(k)) = w_i(k) + u_i^+(k) + \gamma_i u_i^-(k), \tag{27}$$

compare (5) and (6) in Section 2. In the particular setting of an electricity grid, the state $x_i \in \mathbb{R}$ represents the state of charge of a battery in kWh, containing local information of the i-th RES and the variable $z_i \in \mathbb{R}$ denotes the power demand in kW of system i, which needs to be shared with or communicated to the CE. The local power demand z_i depends on the net consumption $w_i \in \mathbb{R}$ in kW, i.e., the energy demand minus the generation of solar PV panels, and can be manipulated by charging or discharging the storage device using the input signals $u_i = (u_i^+, u_i^-)^T \in \mathbb{R}^2$ (in kW). The net consumption $w_i(k)$, which is assumed to be a known exogenous signal for all $i \in [1:\mathcal{I}]$ and for all $n \in [k : k+N-1]$, causes the time dependency of the function h in (27). Hence, the prediction horizon corresponds to the time interval, on which *reliable* data is available, see, e.g. [41] for the impact of forecast errors. The additional parameters in (26)-(27) are constants, where $T > 0$ represents the length of a sampling interval in hours (h), while ϱ_i, β_i, $\gamma_i \in (0,1]$ are used to model losses due to energy transformation.

Additionally, for constants C_i, $\underline{u}_i$, $\overline{u}_i \in \mathbb{R}_{>0}$, $i \in [1:\mathcal{I}]$, constraints on the state of charge of the storage device and maximal charging and discharging rates are defined in [41, 10], which result in the state constraints

$$x_i(k) \in \mathbb{X}_i = \{x_i(k) \in \mathbb{R} | 0 \leq x_i(k) \leq C_i\} \tag{28}$$

and the input constraints

$$u_i(k) = \begin{pmatrix} u_i^+(k) \\ u_i^-(k) \end{pmatrix} \in \mathbb{U}_i = \left\{ u_i \in \mathbb{R}^2 \;\middle|\; \begin{array}{rcl} -\underline{u}_i \leq & u_i^- & \leq 0 \\ 0 \leq & u_i^+ & \leq \overline{u}_i \\ 0 \leq & \frac{u_i^-}{-\underline{u}_i} + \frac{u_i^+}{\overline{u}_i} & \leq 1 \end{array} \right\} \tag{29}$$

for all $i \in [1:\mathcal{I}]$ and for all $k \in \mathbb{N}$. Note that in this setting neither the state constraints $\mathbb{X}_i$ nor the input constraints $\mathbb{U}_i$ are time dependent.

In [41, 8] the objective function in the MPC closed-loop formulation is defined in such a way, that for all $k \in \mathbb{N}$ in the prediction horizon, the deviation of the aggregated power demand $\sum_{i=1}^{\mathcal{I}} z_i(k)$ from the aggregated net consumption computed over the prediction horizon is minimized. Here, we use a slightly different approach to be able to use the definition (7) of the stage costs. For a given prediction horizon $N \in \mathbb{N}$, we denote the average net consumption over the prediction horizon of a single RES $i \in [1:\mathcal{I}]$ by

$$\zeta_i(k) = \begin{cases} \frac{1}{k+1} \sum_{n=0}^{k} w_i(n), & \text{if } k \leq N-1, \\ \frac{1}{N} \sum_{n=k-N+1}^{k} w_i(n), & \text{if } k \geq N-1. \end{cases} \tag{30}$$

For the first $N-1$ time steps, where the past data of $w_i(k)$, for $k<0$ is not available, only the average over the available data is used. Analogously the average net consumption of all the systems is defined as

$$\overline{\zeta}(k) = \tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \zeta_i(k) \tag{31}$$

for all $k \in \mathbb{N}$. With this definition the coupled stage costs, tracking the average net consumption of the RESs can be written in the form

$$\overline{\ell}(k, \tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} z_i(k)) = \left(\overline{\zeta}(k) - \tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} z_i(k)\right)^2 \tag{32}$$

and the overall objective function of the OCP (8) in the distributed MPC Algorithm 1 at time $k \in \mathbb{N}$ is defined by

$$\phi(\mathbf{y};k) = \textstyle\sum_{n=k}^{k+N-1} \left(\overline{\zeta}(n) - \tfrac{1}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} z_i(n)\right)^2 .$$

Here, the local terms in the objective function are not present since the weighting parameter is set to $\alpha = 1$. The objective function is convex but not strictly or strongly convex. The sets $\mathbb{D}_i(k)$, $i \in [1:\mathcal{I}]$ and $k \in \mathbb{N}$, are defined by the initial state of charge, the system dynamics (26)-(27) and the constraints (28)-(29), see (12). Since the dynamics and the constraints are linear, $\mathbb{D}_i(k)$ is convex and closed. Additionally, compactness of $\mathbb{D}_i(k)$ can be concluded from the fact that $\mathbb{U}_i$ and $\mathbb{X}_i$, $i \in [1:\mathcal{I}]$ are compact and the boundedness of $\mathbb{U}_i$ implicitly limits the power demand z_i for all $i \in [1:\mathcal{I}]$. Thus, the distributed optimization Algorithm 3 can be used to compute an optimal solution of the OCP embedded in the MPC Algorithm 1.

Numerical simulations

For the numerical simulations throughout this section a setting of $\mathcal{I} = 100$ RESs is used. The parameter T is set to $T = 0.5$, representing half-hour windows, and the energy demand is predicted for 24 hours into the future, i.e., we set $N = 48$. Additionally the constants $\underline{u}_i = \overline{u}_i = 0.5$ are fixed for the maximal discharging/charging rates for all $i \in [1:\mathcal{I}]$. The constants modeling the losses are set to $\varrho_i = 0.99$ and $\beta_i = \gamma_i = 0.95$ for all $i \in [1:\mathcal{I}]$. The battery capacities of the storage devices are set to $C_i = 2$ kWh with initial state of charge $x_i(0) = 0.5$ kWh for all $i \in [1:\mathcal{I}]$. For the ADMM Algorithm 3 embedded in the MPC scheme the parameter ρ is set to $\rho = 0.1$.

The closed-loop results of the MPC Algorithm 1 over a simulation length of one week (i.e., $k = 0, \ldots, 335$) are visualized in Figure 1. Due to the definition of the reference value $\overline{\zeta}(k)$ in (31), $\overline{\zeta}(k)$ takes 24 hours before it becomes a reliable reference value which only changes slowly over time (see the green line in the Figure 1, top). In addition to the reference value, Figure 1 (top)

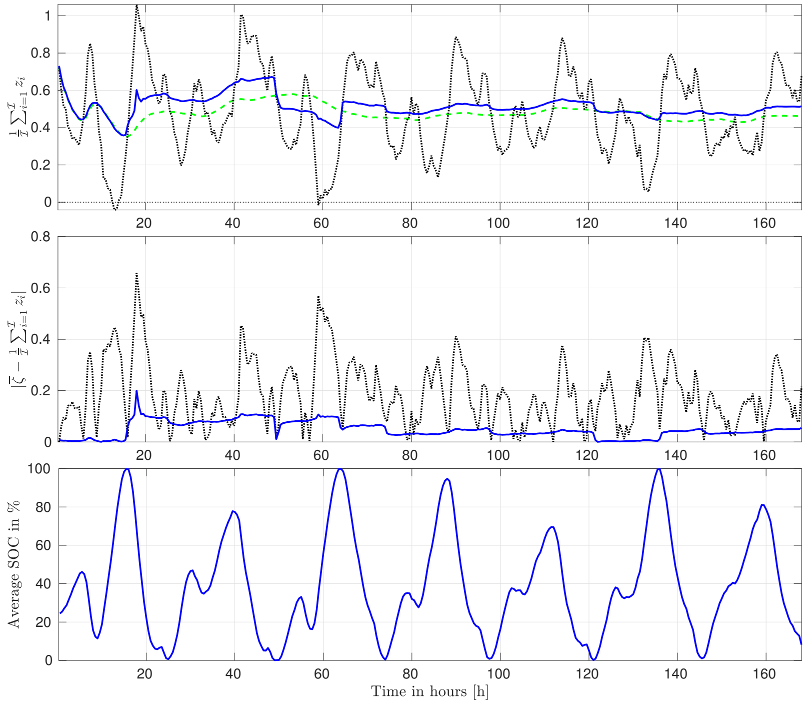

Fig. 1: Closed-loop results of the MPC Algorithm 1 optimizing the use of storage devices. The first figure shows the reference value $\overline{\zeta}(k)$ (green) as well as the uncontrolled (black) and controlled (blue) average power demand, $\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} w_i(k)$ and $\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} z_i(k)$, respectively. The second plot shows the deviation of the uncontrolled (black) and the controlled (blue) average power demand from the reference value, while the last figure visualizes the average State of Charge (SOC) of the storage devices.

visualizes the uncontrolled average power demand $\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} w_i(k)$ (black) and the optimized average power demand $\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} z_i(k)$ (blue). The storage devices help to reduce the the peaks in the average power demand significantly. The deviation of the uncontrolled power demand (black) and the controlled power demand (blue) from the reference value is shown in Figure 1 (middle). Even though the peaks are reduced, the MPC algorithm is not able to track the reference value perfectly due to the maximal capacity of the storage devices and due to the bounds on the charging/discharging rates. This can be observed in the last plot in Figure 1, where the average State of Charge (SOC) of the storage devices is visualized.

4.2 Storage devices and controllable loads

In this section we establish an additional degree of freedom in the model of the electricity network, i.e, we consider the concept of controllable loads introduced for this particular setting in [9]. To this end, the net consumption is split into two parts: the static load $w_i(k)$ and the controllable load $w_i^c(k)$. While we cannot influence the static load, controllable load can be shifted in time. To schedule the controllable load we extend the control input $u_i = (u_i^+, u_i^-, u_i^c)^T \in \mathbb{R}^3$ and modify the dynamics (27) to

$$z_i(k) = w_i(k) + u_i^+(k) + \gamma_i u_i^-(k) + u_i^c(k) \tag{33}$$

for all $i \in [1 : \mathcal{I}]$. Here, $u_i^c(k)$ in kW represents the amount of controllable load scheduled at time k.

We make the assumption that controllable load becomes available at a certain time $j \leq k$ and then needs to be scheduled until (its deadline at) time k, i.e., $w_i^c(k+\overline{N}-1)$, $k \in \mathbb{Z}_{\geq 1-\overline{N}}$, has to be planned during the time interval from $\max\{0, k\}$ to $k+\overline{N}-1$ for a given $\overline{N} \in \mathbb{N}$. To this end, the time-varying constraints

$$\sum_{j=0}^{k} w_i^c(j) - \sum_{j=0}^{k-1} u_i^c(j) \quad \leq \quad u_i^c(k) \quad \leq \quad \sum_{j=0}^{k+\overline{N}-1} w_i^c(j) - \sum_{j=0}^{k-1} u_i^c(j) \tag{34}$$

are introduced. Further, we assume that only a certain amount of the controllable load can be scheduled during one time step, which leads to

$$0 \quad \leq \quad u_i^c(k) \quad \leq \quad \overline{w}_i^c \tag{35}$$

for a constant $\overline{w}_i^c \geq 0$, which is assumed to be chosen such that (34) can be fulfilled simultaneously. Note that in (34), at a fixed time instant $\overline{k} \in \mathbb{N}_0$, $u_i^c(k)$ for $k \geq \overline{k}$ represent control variables which need to be optimized, whereas $u_i^c(k)$ for $k < \overline{k}$ are constants which have been fixed at previous time steps. To be able to update the input constraints (29) capturing the controllable loads, we rewrite the constraints (34) as time-dependent upper and lower bounds

$$\lambda_i^q(k) := \sum_{j=0}^{k+q} w_i^c(j) - \sum_{j=0}^{k-1} u_i^c(j) \leq \sum_{j=k}^{k+q} u_i^c(j) \tag{36a}$$

$$\Lambda_i^q(k) := \sum_{j=0}^{k+\min\{q+\overline{N},N\}-1} w_i^c(j) - \sum_{j=0}^{k-1} u_i^c(j) \geq \sum_{j=k}^{k+q} u_i^c(j) \tag{36b}$$

for the input variables u_i^c, for $i \in [1 : N]$, $k \in \mathbb{N}$ and over the prediction horizon $q \in [0 : N-1]$. Note that the time-dependency is due to the time dependent controllable loads $w_i^c(k)$, $k \in \mathbb{N}$, as well as the decisions made by the controller to define $u_i^c(j)$ for $j < k$ for all $i \in \mathbb{N}_{\mathcal{I}}$. Since we assume that the system dynamics and in particular the load and the controllable load can only be estimated over the prediction horizon, the first sum in (36b) stops at the increment $k + \min\{q + \overline{N}, N\} - 1$.

Due to the linearity of the dynamics (33) and the constraints (35)-(36), convexity and compactness of the (now time-dependent) sets $\mathbb{D}_i(k)$, $i \in [1 : \mathcal{I}]$, $k \in \mathbb{N}$, follows the same arguments as in Section (4.1). For the stage costs, we additionally have to take the controllable loads into account in the computation of the average net consumption, i.e., $\zeta_i(k)$, $i \in \mathbb{N}_{\mathcal{I}}$, is defined as

$$\zeta_i(k) = \begin{cases} \frac{1}{k+1} \sum_{n=0}^{k} w_i(n) + w_i^c(n), & \text{if } k \leq N-1, \\ \frac{1}{N} \sum_{n=k-N+1}^{k} w_i(n) + w_i^c(n), & \text{if } k \geq N-1 \end{cases}$$

in this setting.

Numerical simulations

In addition to the parameters used in the setting without controllable loads we set $\overline{N} = 12$, which represents a six hour time window for scheduling the controllable loads, and bound the controllable load for each RES $i \in [1 : \mathcal{I}]$ at time $k \in \mathbb{N}_0$ by $w_i^c(k) \leq \overline{w}_i^c = 1.25$ [kW].

The results comparing the setting with and without controllable loads can be found in Figure 2. Here, in addition to the results shown in Figure 1, the simulations including controllable loads are visualized in red. The additional degree of freedom in the controllable loads clearly improves the performance of the MPC scheme where only stroage devices are used. The deviation of the average power demand from the reference signal is hardly visible at most of the time steps in the simulation. To obtain a perfect tracking however, i.e., to obtain $|\overline{\zeta}(k) - \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} z_i(k)| = 0$ for all $k = 0, 1, \ldots, 335$, either the capacity of the storage devices or the percentage of controllable load from the overall load needs to be increased.

4.3 Optimal operation of coupled microgrids

So far in this paper, and also in preceding publications, we have concentrated on the optimal operation of a single electricity network. Here, we extend these results to the optimal operation of $\varXi$ microgrids (MGs), $\varXi \in \mathbb{N}$, coupled through a network of transmission lines as visualized in Figure 3, for example. Here, a network of $\varXi = 4$ MGs is shown where MG_1 is only connected to

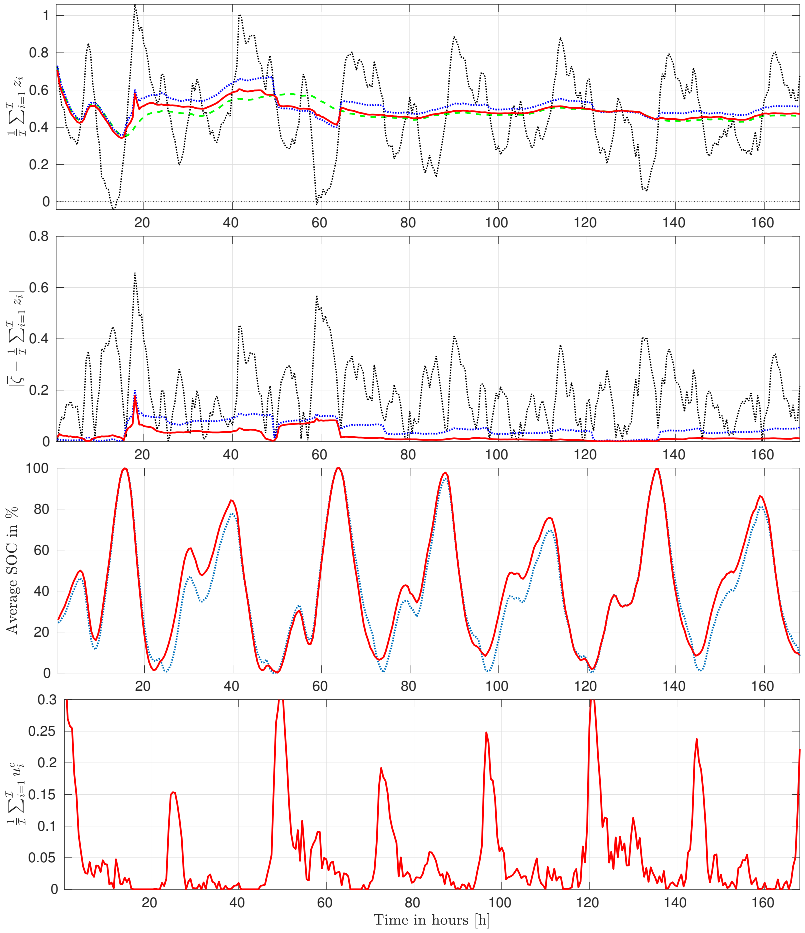

Fig. 2: Impact of controllable loads in the closed-loop simulation of Algorithm 1. Compared to Figure 1 additionally the closed-loop results with controllable loads are shown in red. The last plot shows the extended average control input $\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} u_i^c(k)$, $k \in [0:335]$.

MG$_2$, and MG$_2$, MG$_3$ and MG$_4$ are fully connected through transmission lines.

The individual MGs can be defined as discussed in Section 4.1 without controllable loads or as in Section 4.2 with controllable loads. We use $\mathcal{I}_\kappa \in \mathbb{N}$, to denote the number of RESs in MG$_\kappa$, $\kappa \in [1:\Xi]$. In the case where the MGs are not connected a straightforward extension of the stage costs defined in (32) is to consider the costs

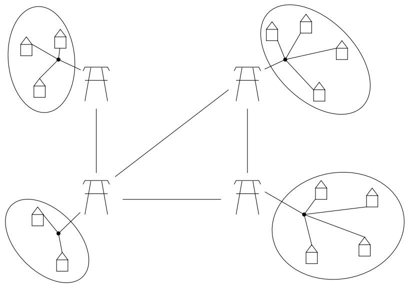

Fig. 3: Example of four MGs coupled through transmission lines. Each MG consists of a number of $\mathcal{I}_\kappa \in \mathbb{N}$, $\kappa = 1, 2, 3, 4$, RESs.

$$\hat{\ell}(k, (z_1, \ldots, z_{\mathcal{I}})^T(k)) = \textstyle\sum_{\kappa=1}^{\varXi} \left(\mathcal{I}_\kappa \overline{\zeta}_\kappa(k) - \sum_{i=1}^{\mathcal{I}_\kappa} z_{\kappa_i}(k) \right)^2, \tag{37}$$

where in contrast to (32) the deviation of the average net consumption $\overline{\zeta}_\kappa(k)$, $\kappa \in \mathbb{N}_{\varXi}$, in the MGs is penalized. This means, $\overline{\zeta}_\kappa(k)$ is defined as

$$\overline{\zeta}_\kappa(k) = \tfrac{1}{\mathcal{I}_\kappa} \textstyle\sum_{i=1}^{\mathcal{I}_\kappa} \zeta_{\kappa_i}(k). \tag{38}$$

We are interested in the case where the MGs are coupled via transmission lines. In particular, we want to investigate whether the MGs can benefit from an energy exchange even if the energy exchange involves losses. We thus consider stage costs of the form

$$\hat{\ell}(k, (z_1, \ldots, z_{\mathcal{I}})^T(k), \delta(k)) = \sum_{\kappa=1}^{\varXi} \left(\mathcal{I}_\kappa \overline{\zeta}_\kappa(k) - \sum_{\nu=1}^{\varXi} (\delta_{\kappa,\nu}(k) \eta_{\kappa,\nu}) \sum_{i=1}^{\mathcal{I}_\nu} z_{i_\nu}(k) \right)^2 \tag{39}$$

with additional variables $\delta(k) \in [0,1]^{\varXi \times \varXi}$ and constants $\eta \in [0,1]^{\varXi \times \varXi}$. The matrix entry $\delta_{\kappa,\nu}(k)$, $\kappa, \nu \in [1 : \varXi]$, represents the fraction of the power demand of MG$_\kappa$ at time k, which is used to manipulate the average $\overline{\zeta}_\nu(k)$ of MG$_\nu$. To define $\delta_{\kappa,\nu}(k)$ as a fraction of the overall power demand of MG$_\kappa$ the linear constraints

$$\textstyle\sum_{\nu=1}^{\varXi} \delta_{\kappa,\nu}(k) = 1 \quad \forall\, \kappa \in [1 : \varXi] \tag{40}$$

are introduced. Moreover, to ensure that power can be exchanged over a transmission line only in one direction at a fixed time index $k \in \mathbb{N}$ we use the nonlinear constraints

$$\delta_{\kappa,\nu}(k) \cdot \delta_{\nu,\kappa}(k) \leq 0 \qquad \forall\, \kappa, \nu \in [1 : \varXi], \;\; \kappa \neq \nu. \tag{41}$$

The matrix η is assumed to be symmetric, i.e. $\eta_{\kappa,\nu} = \eta_{\nu,\kappa}$ for all κ, $\nu \in [1:\Xi]$, where a *zero-entry* ($\eta_{\kappa,\nu} = \eta_{\nu,\kappa} = 0$) indicates that no transmission line between MG_ν and MG_κ exists and values $\eta_{\kappa,\nu} \in (0,1)$ correspond to losses. In addition, we set $\eta_{\kappa,\kappa} = 1$, $\kappa \in [1:\Xi]$. For the MGs visualized in Figure 3 the constants η can be defined as

$$\eta^{\text{loss}} = \begin{pmatrix} 1.0\ 0.0\ 0.8\ 0.0 \\ 0.0\ 1.0\ 0.7\ 0.8 \\ 0.8\ 0.7\ 1.0\ 0.9 \\ 0.0\ 0.8\ 0.9\ 1.0 \end{pmatrix} \qquad \text{and} \qquad \eta^{\text{no loss}} = \begin{pmatrix} 1\ 0\ 1\ 0 \\ 0\ 1\ 1\ 1 \\ 1\ 1\ 1\ 1 \\ 0\ 1\ 1\ 1 \end{pmatrix} \tag{42}$$

with losses and without losses, for example.

To handle the additional set of variables δ in the MPC closed-loop we extend Algorithm 1. Note that the coupling of the variables δ and $(z_1, \dots, z_{\mathcal{I}})$ leads to a non-convex objective function. Additionally, the constraints (41) are nonlinear and non-convex. Hence, δ is only optimized in the MPC algorithm, but not in the distributed optimization conducted in the second Step (2) of the MPC algorithm.

Distributed optimization of the energy exchange

Using the definitions introduced in Section 3 we define the objective function

$$\hat{\phi}(\mathbf{z}, \delta; k) = \textstyle\sum_{n=k}^{k+N-1} \hat{\ell}(k, \mathbf{z}(k), \delta(k)) \tag{43}$$

with $\delta \in \mathbb{R}^{\Xi \times \Xi \times N}$. For the constraints we define the set

$$\Delta = \left\{ \delta \in [0,1]^{\Xi \times \Xi \times N} \;\middle|\; \begin{array}{c} \sum_{\nu=1}^{\Xi} \delta_{\kappa,\nu}(n) = 1 \ \ \forall \kappa \in \Xi \\ \delta_{\kappa,\nu}(n) \cdot \delta_{\nu,\kappa}(n) \leq 0 \ \ \forall \kappa, \nu \in [1:\Xi], \kappa \neq \nu \\ \forall\ n \in [k : k+N-1] \end{array} \right\}.$$

Hence, the optimal control problem involved in the MPC algorithm at time $k \in \mathbb{N}$ is defined as

$$\min_{\substack{\mathbf{y} \in \mathbb{D}(k) \\ \delta \in \Delta}} \alpha \hat{\phi}(\mathbf{z}, \delta; k) + \tfrac{1-\alpha}{\mathcal{I}} \textstyle\sum_{i=1}^{\mathcal{I}} \phi_i(\mathbf{y}_i; k). \tag{44}$$

As already pointed out, due to the additional variables δ, the distributed optimization scheme (24) is not applicable in the current form. Thus, we update the variables δ only once at every time step $k \in \mathbb{N}$ within the MPC algorithm and keep δ fixed in the iterates of the distributed optimization algorithm. The corresponding MPC algorithm is given in Algorithm 4.

In (46), δ is updated for the next time step using the ideas discussed in Section 3.3 and $I \in \mathbb{R}^{\Xi \times \Xi}$ denotes the identity matrix. Optimization problem (45) is not convex and thus it is not guaranteed, that an optimal

Algorithm 4 Distributed MPC for energy exchange

```
Input: Time horizon $N \in \mathbb{N}$, number of MGs $\Xi \in \mathbb{N}$, transmission grid parameters
$\nu \in [0,1]^{\Xi \times \Xi}$, number of systems $\mathcal{I}_\kappa \in \mathbb{N}$ in MG$_\kappa$, $\kappa \in [1 : \Xi]$, satisfying $\sum_{\kappa=1}^{\Xi} \mathcal{I}_\kappa =$
$\mathcal{I}$, stage costs $\hat{\ell}$, $\ell_i$, $i \in [1 : \mathcal{I}]$, and a weighting parameter $\alpha \in [0,1]$.
Initialization: Set $k = 0$.
Main loop: For $k \in \mathbb{N}_0$
  (1) For all $i \in [1 : \mathcal{I}]$, measure the current states $\hat{x}_i := x_i(k)$.
  (2) Solve the OCP (8) for fixed $\delta$ to obtain minimizing sequences
          $\mathbf{u}_i^\star = (u_i^\star(k), \ldots, u_i^\star(k+N-1))^T \qquad \forall\, i \in [1 : \mathcal{I}].$
  (3) Obtain the optimal energy exchange $\delta^*$ by solving the optimization problem
          $\delta^* \in \operatorname*{arg\,min}_{\delta \in \Delta} \alpha \hat{\phi}(\mathbf{z}^*, \delta; k) + \frac{1-\alpha}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \phi_i(\mathbf{y}_i^*; k).$   (45)
  (4) For all $i \in [1 : \mathcal{I}]$, implement $\mu_i(k, \hat{x}_i) := u_i^\star(k)$ and $\delta^*(k)$, define
          $\delta^+ = (\delta^*(k+1) \times \ldots \times \delta^*(k+N-1) \times I),$   (46)
      shift the horizon forward in time, i.e. set $k = k+1$, and go to Step (1).
```

solution is found at every time step. However, observe that the optimization problem (45) does not involve a coupling over the prediction horizon and $\delta(n)$, $n \in [k : k+N-1]$ can be computed independently, leading to N optimization problems, where the number of unknowns is upper bounded by Ξ^2. Even though optimality with respect to the energy exchange cannot be shown in our setting, simulation results indicate the potential benefit of the additional Step (3) in Algorithm 4.

The power exchange of coupled MGs is also discussed in [25]. In [25] however, the power flow over the transmission lines results from linearized DC power flow equations and cannot be optimized separately. In contrast, Algorithm 4 includes the optimization of the power exchange through the additional variable δ.

Numerical simulations

To show the numerical properties of Algorithm 4 we consider a network of $\Xi = 4$ MGs visualized in Figure 3. Additionally, we consider the set of parameters η defined in (42) with and without losses. The individual MGs consist of $\mathcal{I}_1 = 20$, $\mathcal{I}_2 = \mathcal{I}_3 = 25$ and $\mathcal{I}_4 = 30$ RESs. Each RES is defined through the parameters introduced in Section 4.1.

The deviation of the power demand of the closed-loop solution from the reference values in the individual MGs and for the different settings is visualized in Figure 4 and 5. Figure 4 compares the uncontrolled power demand with the controlled power demand without energy exchange (see Section 4.1).

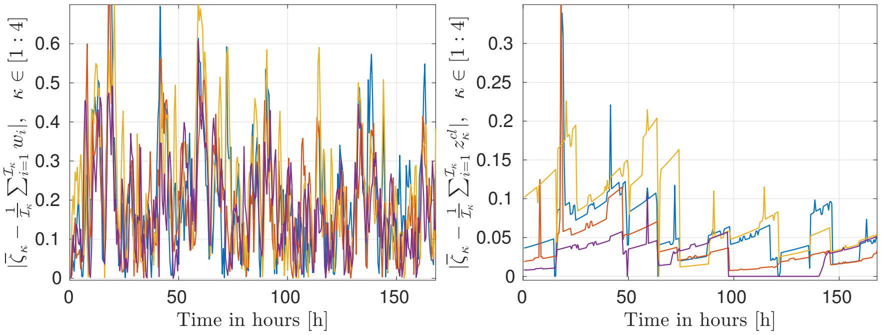

Fig. 4: Deviation of the uncontrolled average power demand in the MGs from the reference values in the left figure and the deviation of the closed-loop solution without power exchange (i.e. optimization with respect to (37)) on the right. Note the different scaling on the y-axis.

In Figure 5 additionally the power exchange is taken into account. The exchange of energy without losses over the transmission lines (Figure 5, right) clearly improves the results of the uncoupled MGs (Figure 4, right). The benefit of the energy exchange with losses η^{loss} over the transmission lines is not that obvious. It can however be observed by evaluating the MPC closed-loop performance (4), taking the average over 336 iterations. Here, the uncontrolled setting in Figure 4 (left) leads to average costs of 148.66 compared to 10.60 in the controlled case without energy exchange (Figure 4, right).

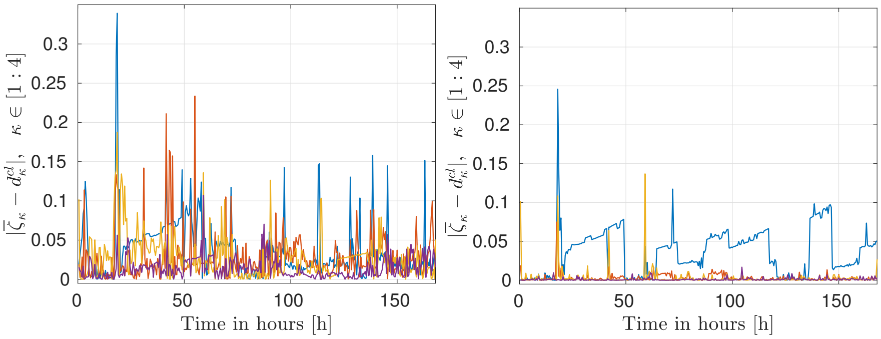

Fig. 5: Visualization of the deviation of the closed-loop solution from the reference value with power exchange (i.e. optimization with respect to (39)), with (left) and without (right) losses are visualized. Here, d_κ^{cl} denotes the closed-loop average power demand after the power exchange $d_\kappa^{\text{cl}}(k) = \frac{1}{\mathcal{I}_\kappa}\sum_{\nu=1}^{\Xi}\delta_{\kappa,\nu}^{\text{cl}}(k)\eta_{\kappa,\nu}\sum_{i=1}^{\mathcal{I}_\nu} z_{i_\nu}(k)$. The matrices η are defined in (42).

The setting with energy exchange and with losses over the transmission lines (Figure 5, left) decreases the average costs to 3.50, which shows the improvements compared to the uncoupled simulations. As one might expect from Figure 5 (right) the setting with energy exchange and without losses outperforms the other simulation results with a value of 1.00 for the average closed-loop performance over 336 iterations.

5 Conclusions

In this paper we gave a conceptual review of dual decomposition – in particular dual ascent – and ADMM as representatives of the class of iterative distributed optimization algorithms. Furthermore, we thoroughly discussed their embedding within distributed MPC schemes. Here, we emphasized the importance to tailor the distributed optimization scheme to the particular application to ensure essential properties like scalability and plug-and-play capability. To demonstrate the effectiveness of the proposed combination, we considered an application, in which flexibilities (energy storage devices, controllable loads, and energy exchange between microgrids) were exploited for load shaving. In particular, we rigorously showed global convergence for the setting with controllable loads to extend our previous work [9] and presented a new model, which allows to (numerically) assess the potential of energy exchange between several microgrids while taking transmission losses into account.

References

1. Ausgrid dataset. https://www.ausgrid.com.au/Common/About-us/Corporate-information/Data-to-share/Solar-home-electricity-data.aspx accessed: 18/05/2018.
2. 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.
3. J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.
4. D. P. Bertsekas. *Nonlinear programming.* Athena scientific, 1999.
5. D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods.* Athena Scientific, Belmont, MA, USA, 1989.
6. A. Boccia, L. Grüne, and K. Worthmann. Stability and feasibility of state constrained mpc without stabilizing terminal constraints. *Systems & Control Letters*, 72(Supplement C):14 – 21, 2014.
7. 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.

8. P. Braun, T. Faulwasser, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann. Hierarchical distributed ADMM for predictive control with applications in power networks. *IFAC Journal of Systems and Control*, 3:10 – 22, 2018.
9. P. Braun, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann. *Progress in Industrial Mathematics at ECMI 2014*, volume 22 of *Mathematics in Industry*, chapter Model predictive control of residential energy systems using energy storage & controllable loads, pages 617–623. Springer, 2016.
10. P. Braun, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann. Towards price-based predictive control of small-scale electricity network. *International Journal of Control*, 2017. DOI:10.1080/00207179.2017.1339329.
11. E. F. Camacho and C. A. Bordons. *Model predictive control in the process industry.* Springer Science & Business Media, 2012.
12. N. Chatzipanagiotis, D. Dentcheva, and M. M. Zavlanos. An augmented Lagrangian method for distributed optimization. *Mathematical Programming*, 152(1-2):405–434, 2015.
13. P. D. Christofides, R. Scattolini, D. M. de la Pena, and J. Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51:21–41, 2013.
14. 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.
15. M. Diehl, H. G. Bock, J.P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4):577–585, 2002.
16. 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.
17. G. Eichfelder. *Adaptive scalarization methods in multiobjective optimization*, volume 436. Springer, 2008.
18. T. Faulwasser and D. Bonvin. Exact turnpike properties and economic NMPC. *European Journal of Control*, 35:34–41, 2017.
19. 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.
20. 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.
21. P. Giselsson and A. Rantzer. On feasibility, stability and performance in distributed model predictive control. *IEEE Transactions on Automatic Control*, 59(4):1031–1036, 2014.
22. L. Grüne and J. Pannek. *Nonlinear Model Predictive Control. Theory and Algorithms.* Springer, London, 2nd edition, 2017.
23. L. Grüne and S. Pirkelmann. Closed-loop performance analysis for economic model predictive control of time-varying systems. In *Proc. of the 54th IEEE Conference on Decision and Control (CDC)*, pages 5563–5569, 2017.
24. L. Grüne and M. Stieler. Performance guarantees for multiobjective model predictive control. In *Proc. 56th IEEE Conference on Decision and Control (CDC)*, pages 5545–5550, 2017.
25. C. A. Hans, P. Braun, J. Raisch, L. Grüne, and C. Reincke-Collon. Hierarchical distributed model predictive control of interconnected microgrids. *IEEE Transactions on Sustainable Energy*, 2018. DOI:10.1109/TSTE.2018.2802922.
26. B. Houska, J. Frasch, and M. Diehl. An augmented Lagrangian based algorithm for distributed nonconvex optimization. *SIAM Journal on Optimization*, 26(2):1101–1127, 2016.

27. J. Köhler, M. A. Müller, and F. Allgöwer. Nonlinear reference tracking: An economic model predictive control perspective. *IEEE Transactions on Automatic Control*, 2018.
28. J. H. Lee. Model predictive control: Review of the three decades of development. *International Journal of Control, Automation and Systems*, 9(3):415, 2011.
29. S. Lucia, M. Kögel, and R. Findeisen. Contract-based predictive control of distributed systems with plug and play capabilities. *IFAC-PapersOnLine*, 48(23):205–211, 2015.
30. J. M. Maestre and R. R. Negenborn. *Distributed model predictive control made easy*, volume 69. Springer, 2014.
31. M. A. Müller and L. Grüne. Economic model predictive control without terminal constraints: optimal periodic operation. *Automatica*, 70(C):128–139, 2016.
32. M. A. Müller and K. Worthmann. Quadratic costs do not always work in MPC. *Automatica*, 82:269 – 277, 2017.
33. N. Parikh and S.P. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.
34. S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.
35. T. Raff, S. Huber, Z. K. Nagy, and F. Allgöwer. Nonlinear model predictive control of a four tank system: An experimental stability study. In *Proc. IEEE Conf. Control Applicat.*, pages 237–242, 2006.
36. H. A. Rahman, M. S. Majid, A. R. Jordehi, C. K. Gan, M. Y. Hassan, and S. O. Fadhl. Operation and control strategies of integrated distributed energy resources: A review. *Renewable and Sustainable Energy Reviews*, 51:1412–1420, 2015.
37. J. B. Rawlings, D. Q. Mayne, and M.M. Diehl. *Model Predictive Control: Theory, Computation, and Design.* Nob Hill Publishing, 2017.
38. B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia. Cooperative distributed model predictive control. *Systems & Control Letters*, 59(8):460 – 469, 2010.
39. A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright. Distributed MPC strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, 16(6):1192–1206, 2008.
40. Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on control systems technology*, 18(2):267–278, 2010.
41. 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.
42. P. Yi, Y. Hong, and F. Liu. Distributed gradient algorithm for constrained optimization with application to load sharing in power systems. *Systems & Control Letters*, 83:45–52, 2015.
43. M. N. Zeilinger, Y. Pu, S. Riverso, G. Ferrari-Trecate, and C. N. Jones. Plug and play distributed model predictive control based on distributed invariance and optimization. In *Proc. 52nd IEEE Conf. Decision and Control (CDC)*, pages 5770–5776, 2013.