A Distributed Optimization Algorithm for the Predictive Control of Smart Grids

Philipp Braun[1], Lars Grüne[1], Christopher M. Kellett[2], Steven R. Weller[2], and Karl Worthmann[3]

***Abstract*—In this paper, we present a hierarchical, iterative distributed optimization algorithm, and show that the algorithm converges to the solution of a particular global optimization problem. The motivation for the distributed optimization problem is the predictive control of a smart grid, in which the states of charge of a network of residential-scale batteries are optimally coordinated so as to minimize variability in the aggregated power supplied to/from the grid by the residential network. The distributed algorithm developed in this paper calls for communication between a central entity and an optimizing agent associated with each battery, but does not require communication between agents. The distributed algorithm is shown to achieve the performance of a large-scale centralized optimization algorithm, but with greatly reduced communication overhead and computational burden. A numerical case study using data from an Australian electricity distribution network is presented to demonstrate the performance of the distributed optimization algorithm.**

***Index Terms*—Distributed Optimization, Model Predictive Control, Smart Grid**

I. Introduction

The well-known curse of dimensionality encountered when trying to solve large-scale optimization problems has led to several work-arounds including various distributed optimization algorithms. In the control literature, this approach was initiated in the thesis of Tsitsiklis [1] (see also [2]), wherein decentralized optimization schemes with and without communication between agents or processors were studied. Adopting a more modern terminology, we refer to decentralized algorithms with and without communication between agents as *distributed* and *decentralized* algorithms, respectively.

Recent work in the field of multi-agent systems, particularly in the field of consensus, has seen a resurgence of interest in the area of distributed optimization; see for example [3], [4], [5], [6], [7], [8], [9] 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 [10], where the global cost function is not separable. In [10], 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, [10] assumes a global coupled inequality constraint, where each agent knows its (functional) contribution to the global coupled constraint. In this setting, [10] presents a distributed optimization algorithm based on neighbor-to-neighbor communication. Importantly, it is shown that this distributed optimization algorithm converges to the solution of the global optimization problem.

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. This deployment presents various challenges to current electricity networks, including the possibility of large supply-demand power swings if battery charging and discharging is poorly scheduled. This has led to a significant research effort in the area of battery scheduling; see [11], [12], [13], [14], [15] and the references therein.

In [16], we studied the problem of optimally scheduling battery storage at the residential level so as to minimize variability in energy supply and demand at a local level; for example at the level of a residential neighborhood where residences have installed both energy generation and storage technologies such as solar photovoltaic panels and batteries, respectively. Unfortunately, this optimal scheduling problem does not lead to a global cost function that is decomposable as a sum of cost functions at each residence. Hence, in [16], we proposed and compared centralized, distributed, and decentralized algorithms, where a degradation in performance was observed from the centralized solution. Of course, the benefit of the distributed and decentralized algorithms resulted from their scalability.

In this paper, we present a hierarchical, iterative, distributed optimization algorithm that converges to the solution of the centralized optimization problem for the specific cost function and system dynamics used in [16]. In contrast to [10], the algorithm presented herein relies on the presence of a central entity with which all agents communicate. Note that this is the only communication present; i.e., agents do not communicate directly with each other. Naturally, the structural difference between the algorithm presented herein and that in [10] means that a different proof technique is required to show that the distributed algorithm converges to the global optimum.

*C.M. Kellett is supported by ARC Future Fellowship FT1101000746. L. Grüne is supported by the Deutsche Forschungsgemeinschaft, Grant GR 1569/13-1.

[1]P. Braun and L. Grüne are with the Mathematical Institute, Universität Bayreuth, 95440 Bayreuth, Germany, e-mail: {philipp.braun, lars.gruene}@uni-bayreuth.de.

[2]C. M. Kellett and S. R. Weller are with the School of Electrical Engineering and Computer Science at the University of Newcastle, Callaghan, New South Wales 2308, Australia, e-mail: {chris.kellett, steven.weller}@newcastle.edu.au.

[3]K. Worthmann is with the Institute for Mathematics, Technische Universität Ilmenau, 99693 Ilmenau, Germany, e-mail: karl.worthmann@tu-ilmenau.de.

The paper is organized as follows: in Section II we recall the system dynamics and optimization problem considered in [16]. In Section III we recall centralized and decentralized model predictive control schemes from [16]. In Section IV we present a hierarchical distributed optimization algorithm and prove that this algorithm converges to the solution of the centralized optimization problem. We additionally compare our algorithm to the primal-dual decomposition approach and the distributed consensus-based primal-dual perturbation of [10]. In Section V we investigate the performance of our algorithm using data provided by an Australian electricity distribution company. Finally, concluding remarks are made in Section VI.

II. The Residential Energy System

Let $\mathcal{I} \in \mathbb{N}$ be the number of Residential Energy Systems (RESs) connected in the local area under consideration. A simple model of RES i, $i \in \{1, \ldots, \mathcal{I}\}$, presented in [16], is given by

$$x_i(k+1) = x_i(k) + T u_i(k), \qquad (1)$$

$$z_i(k) = w_i(k) + u_i(k) \qquad (2)$$

where x_i is the state of charge of the battery in kilowatt-hours (kWh), u_i is the battery charge/discharge rate in kilowatts (kW), w_i is the static load minus the local generation in kilowatts (kW), and z_i is the power supplied by/to the grid in kilowatts (kW). Here, T represents the length of the sampling interval in hours (h); e.g., $T = 0.5$ corresponds to 30 minutes. While the system dynamics (1) is autonomous, the performance output (2) depends on the time varying quantity $w_i(\cdot)$.

The RES network is then defined by the following discrete-time system

$$\begin{aligned} x(k+1) &= f(x(k), u(k)), \\ z(k) &= h(u(k), w(k)) \end{aligned}$$

where $x, u, w, z \in \mathbb{R}^{\mathcal{I}}$, and the definitions of f and h are given componentwise by (1) and (2), respectively. For each RES, the constraints on the battery capacity and charge/discharge rates are described by the constants $C_i, \overline{u}_i \in \mathbb{R}_{>0}$ and $\underline{u}_i \in \mathbb{R}_{<0}$, i.e.,

$$0 \leq x_i(k) \leq C_i \quad \text{and} \quad \underline{u}_i \leq u_i(k) \leq \overline{u}_i \quad \forall k \in \mathbb{N}_0 \qquad (3)$$

for $i \in \{1, \ldots, \mathcal{I}\}$.

From the point of view of the distribution network, one reasonable performance goal is to reduce variability in usage of the network; i.e., to have a nearly constant power profile. This allows the network operator to avoid expense infrastructure that may only be required to deal with rare peak demand or oversupply events. In our notation, this corresponds to reducing variability in the performance output z. We introduce two relevant performance metrics. To this end, let

$$\Pi(k) := \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} z_i(k) \qquad (4)$$

denote the average power demand at time k and let $\mathcal{N}$ denote the number of samples comprising a simulation length. The performance metric of peak-to-peak (PTP) variation of the average demand of all RESs is given by

$$\left(\max_{k \in \{0,\ldots,\mathcal{N}-1\}} \Pi(k) \right) - \left(\min_{k \in \{0,\ldots,\mathcal{N}-1\}} \Pi(k) \right). \qquad \text{(PTP)}$$

The second performance metric, the root-mean-square (RMS) deviation from the average, is defined as

$$\sqrt{\frac{1}{\mathcal{N}} \sum_{k=0}^{\mathcal{N}-1} \left(\Pi(k) - \Upsilon \right)^2} \qquad \text{(RMS)}$$

with the average demand $\Upsilon := \frac{1}{\mathcal{N}\mathcal{I}} \sum_{k=0}^{\mathcal{N}-1} \sum_{i=1}^{\mathcal{I}} w_i(k)$.

III. Model Predictive Control Approaches

We recall two model predictive control (MPC) algorithms for the control of a network of RESs introduced in [16] and [17]. The first approach is a centralized MPC (CMPC) scheme, in which full communication of all relevant variables for the entire network as well as a known model of the network are required. The second approach is a decentralized MPC (DeMPC) approach where each RES implements its own local MPC controller. This requires no communication or cooperation between RESs. The main contribution of this paper is the presentation and the analysis of a distributed optimization algorithm in Section IV. The distributed optimization algorithm combines the advantages of DeMPC and CMPC, i.e., local optimization with high flexibility of the network topology and reduced communication while maintaining optimality.

MPC iteratively minimizes an optimization criterion with respect to predicted trajectories and implements the first part of the resulting optimal control sequence until the next optimization is performed (see, e.g., [18] or [19]). We propose such a predictive controller. To this end, we assume that we have predictions of the residential load and generation some time into the future that is coincident with the horizon of the predictive controller. In other words, given a prediction horizon $N \in \mathbb{N}$, we assume knowledge of $w_i(j)$ for $j \in \{k, \ldots, k+N-1\}$, where $k \in \mathbb{N}_0$ is the current time.

Note that the focus of this paper is the efficient computation of a solution of a static minimization or optimal control problem, respectively and not the analysis of the closed loop performance of an MPC algorithm using the solution of the minimization problem in a receding horizon fashion.

A. Centralized Model Predictive Control (CMPC)

Define the predicted average power usage for RES i as

$$\zeta_i(k) := \frac{1}{N} \sum_{j=k}^{k+N-1} w_i(j). \qquad (5)$$

To implement the CMPC algorithm, we compute the overall average at every time step k, on the prediction horizon by

$$\bar{\zeta}(k) := \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \zeta_i(k) \qquad (6)$$

and then minimize the joint cost function

$$V(x(k);k) := \min_{\hat{u}(\cdot)} \sum_{j=k}^{k+N-1} \Big(\bar{\zeta}(k) - \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \underbrace{(w_i(j) + \hat{u}_i(j))}_{\hat{z}_i(j)} \Big)^2 \tag{7}$$

with respect to the predicted control inputs $\hat{u}(\cdot) = (\hat{u}_1(\cdot), \hat{u}_2(\cdot), \ldots, \hat{u}_{\mathcal{I}}(\cdot))^T$ with $\hat{u}_i(\cdot) = (\hat{u}_i(j))_{j=k}^{k+N-1}$, $i \in \{1, 2, \ldots, \mathcal{I}\}$, subject to the system dynamics (1), the current state $x(k) = (x_1(k), \ldots, x_{\mathcal{I}}(k))^T$, and the constraints (3) for $i \in \{1, \ldots, \mathcal{I}\}$. The vector of the predicted performance output $\hat{z}(\cdot)$ is defined in the same way as the predicted control $\hat{u}(\cdot)$. To simplify the notation, the current time k is dropped when it does not convey extra information. Additionally we use the notation $u(j) = (u_1(j), \ldots, u_{\mathcal{I}}(j))^T$ for a fixed time $j \in \mathbb{N}$. The same holds for the other variables x, w, and z.

Here, and in what follows, we denote predicted controls and outputs in the MPC algorithm by hats, i.e., for RES i at time j the predicted control is $\hat{u}_i(j)$ and the predicted performance output is $\hat{z}_i(j)$. Optimal solutions of the CMPC minimization problem (7) are indicated by $\sharp$, i.e., the minimum in (7) is attained for $u^\sharp(\cdot)$ and $z^\sharp(\cdot)$, respectively. Note that the minimum of the CMPC minimization problem exists since the objective function is continuous and the constraints define a non-empty compact set. However, the minimizer $u^\sharp(\cdot)$ is, in general, not unique.

B. Decentralized Model Predictive Control (DeMPC)

The CMPC approach presented above requires a significant amount of communication overhead. A further drawback of the CMPC approach is that it requires a central entity or controller with full knowledge of the network model, in particular, constraints (3) for each $i \in \{1, \ldots, \mathcal{I}\}$. Therefore, any change in the network such as the addition of new generation or storage devices or the replacement of such devices requires an update of the central model. As a remedy we recall a decentralized control approach from [16] that reduces the communication overhead and the computational burden of solving a (large-scale) optimization problem.

A straightforward option in order to flatten the energy profile z_i of RES i is to penalize deviations from its (anticipated) average usage defined in (5). This can be achieved by minimizing a local version of the cost function (7), i.e.,

$$\min_{\hat{u}_i(\cdot)} \sum_{j=k}^{k+N-1} (\zeta_i(k) - \underbrace{(w_i(j) + \hat{u}_i(j))}_{\hat{z}_i(j)})^2$$

subject to the system dynamics (1), the current state of charge $x_i(k)$ of the energy storage and the constraints (3). With each RES solving its own optimization problem with no reference to the rest of the network, the aforementioned communication and computation difficulties of the CMPC are no longer present in the DeMPC algorithm.

In Figure 1 the aggregated power profile and the aggregated battery state of charge profile are presented for a simulation of one week ($\mathcal{N} = 336$, $T = 0.5$) for 100 RESs, initial conditions $x_i(0) = 0.5$[kWh], constraints $\overline{u}_i = -\underline{u}_i = 0.3$[kW] and battery capacity $C_i = 2$[kWh] for all $i \in \{1, \ldots, 100\}$. Load and generation data, w, is taken from a dataset provided by the Australian electricity distribution network provider, Ausgrid. This dataset is described in detail in [20].

In previous work, [17] and [16], it was shown that a simple controller, only based on the current state of the battery, does not manage to flatten the profile significantly and that sophisticated algorithms need to be used to have an impact on the performance. Figure 1 compares the uncontrolled system dynamics with the closed loop dynamics of CMPC and DeMPC. Both CMPC and DeMPC improve the performance significantly. Additionally, as a consequence of the central controller having comprehensive information on all parameters for every RES, CMPC outperforms DeMPC.

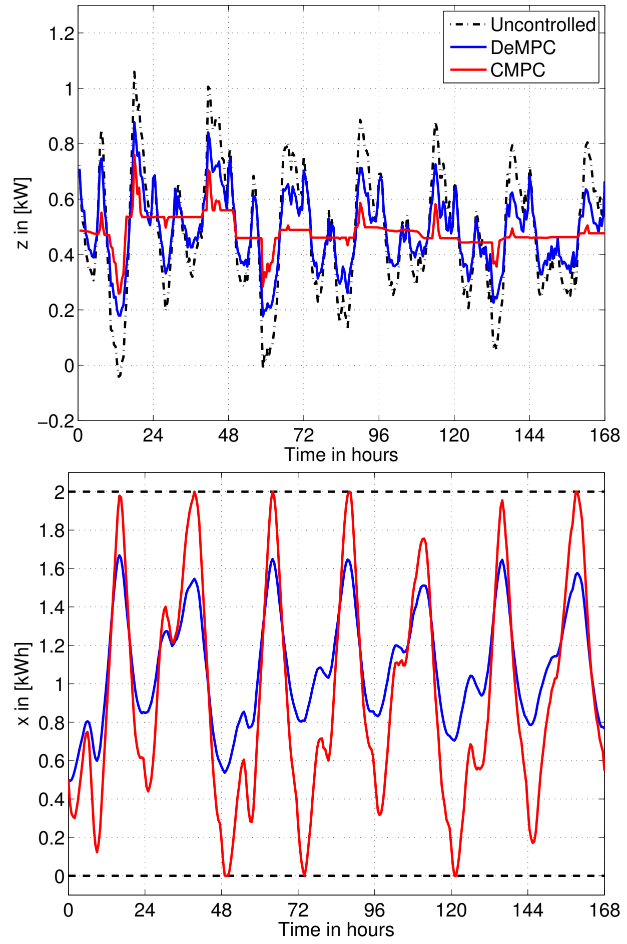


Fig. 1. *Performance of Centralized and Decentralized MPC for a simulation length of one week and 100 RESs. The top plot shows the average power demand, including the uncontrolled or no-battery case for comparison, while the bottom plot shows the average state of charge of the batteries.*

IV. Centralized MPC with Distributed Computation

In this section, we propose a hierarchical distributed model predictive control (DiMPC) approach where each RES can communicate with a central entity with the aim of achieving the performance of the CMPC algorithm, i.e., a network-wide objective while keeping the main advantage of DeMPC: flexibility. After introducing the algorithm we show that the optimal values at a fixed time step of the minimization problems corresponding to DiMPC and CMPC coincide.

A. The Distributed Optimization Algorithm

The distributed optimization algorithm is based on the cost function (7) introduced in the centralized setting. Instead of solving one minimization problem, several iterations are performed at every time step k in which every RES minimizes only over its own control variables. The central entity communicates the aggregated performance output between the systems and computes the variable step size θ in every iteration.

The communication structure of Algorithm 1 is visualized in Figure 2.

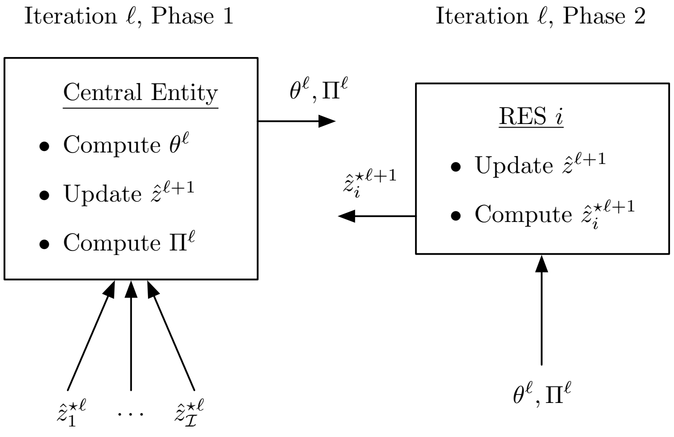

Iteration ℓ, Phase 1
Iteration ℓ, Phase 2
Central Entity
• Compute θ^ℓ
• Update $\hat z^{\ell+1}$
• Compute Π^ℓ
θ^ℓ, Π^ℓ
$\hat z_i^{\star\ell+1}$
RES i
• Update $\hat z^{\ell+1}$
• Compute $\hat z_i^{\star\ell+1}$
$\hat z_1^{\star\ell}$
$\dots$
$\hat z_{\mathcal{I}}^{\star\ell}$
θ^ℓ, Π^ℓ

Fig. 2. *Visualization of the communication structure of Algorithm 1.*

A feasible initialization of RES i, $i \in \{1, \dots, \mathcal{I}\}$ is for example given by $\hat z_i^{\star 1}(j) = \hat z_i^1(j) := w_i(j)$, which corresponds to the choice $u_i^1(\cdot) \equiv 0$ and can be replaced by any other admissible initialization. Additionally the parameters $w_i(j) := w_i(k+j)$, $j = 0, 1, \dots, N-1$, for $i \in \{1, 2, \dots, \mathcal{I}\}$, and $x^0 := x(k)$ have to be given. The average value $\bar\zeta := \bar\zeta(k)$ has to be computed based on the information of $(\hat z_i^1(j))_{j=0}^{N-1}$ by the central entity similar to Equation (6).

Algorithm 1 is terminated either after a fixed number of iterations $\ell_{\max}$ or if the stopping criteria $|V^{\ell+1} - V^\ell| \le \varepsilon$ is satisfied. The input $u(k)$ is defined in a similar way to the update rule (9), i.e., as a convex combination of the last two computed inputs. Alternatively, the input $u_i(k) := \hat u_i^{\star\ell_{\max}}(0)$ for $i \in \{1, 2, \dots, \mathcal{I}\}$ can be used. Since the constraints define a convex set it is ensured that $\hat z^{\ell+1}(\cdot)$ corresponds to an admissible input sequence in every iteration. In Theorem IV.9 we show that, as the iteration index ℓ converges to infinity, Algorithm 1 converges to the unique optimal value of the considered optimal control problem.

Algorithm 1 Distributed Optimization Algorithm

Input:

- **RES** i**,** $i \in \{1, 2, \dots, \mathcal{I}\}$: initial state of charge $x_i(0)$, prediction horizon N, power profile $(w_i(j))_{j=0}^{N-1}$.
- **Central Entity**: Number of RESs $\mathcal{I}$, prediction horizon N, maximal iteration number $\ell_{\max} \in \mathbb{N} \cup \{\infty\}$, desired precision ε.

Initialization:

- **RES** i**,** $i \in \{1, 2, \dots, \mathcal{I}\}$: define and transmit $(\hat z_i^{\star 1}(j))_{j=0}^{N-1}$ and $(\hat z_i^1(j))_{j=0}^{N-1}$.
- **Central Entity**: Set the iteration counter $\ell = 0$ and $V^1 = \infty$, receive $(\hat z_i^1(j))_{j=0}^{N-1}$, $i \in \{1, 2, \dots, \mathcal{I}\}$, transmit $\bar\zeta$.

Phase 1 (Central Entity): Increment the iteration counter ℓ. Then, receive $(\hat z_i^{\star\ell}(j))_{j=0}^{N-1}$, $i = 1, 2, \dots, \mathcal{I}$.

- Compute the step size θ^ℓ as

$$\underset{\theta\in[0,1]}{\operatorname{argmin}} \sum_{j=0}^{N-1} \left(\bar\zeta - \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \left[\theta \hat z_i^{\star\ell}(j) + (1-\theta) \hat z_i^\ell(j) \right] \right)^2 \tag{8}$$

- Compute

$$\hat z^{\ell+1}(j) := \theta^\ell \hat z^{\star\ell}(j) + (1-\theta^\ell) \hat z^\ell(j) \tag{9}$$

  and the predicted average demand

$$\Pi^\ell(j) := \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \hat z_i^{\ell+1}(j)$$

  for $j \in \{0, 1, \dots, N-1\}$. Evaluate the performance index

$$V^{\ell+1} := \sum_{j=0}^{N-1} \left(\bar\zeta - \Pi^\ell(j) \right)^2. \tag{10}$$

- If $|V^{\ell+1} - V^\ell| < \varepsilon$ or $\ell \ge \ell_{\max}$ holds, terminate the algorithm. Otherwise transmit θ^ℓ and $(\Pi^\ell(j))_{j=0}^{N-1}$

Phase 2 (RES i**,** $i \in \{1, 2, \dots, \mathcal{I}\}$**)**: Receive θ^ℓ and $(\Pi^\ell(j))_{j=0}^{N-1}$

- For $j = 0, 1, \dots, N-1$ compute

$$\hat z_i^{\ell+1}(j) := \theta^\ell \hat z_i^{\star\ell}(j) + (1-\theta^\ell) \hat z_i^\ell(j) \tag{11}$$

- Solve the (local) minimization problem

$$\min_{\hat u_i(\cdot)} \sum_{j=0}^{N-1} \left(\bar\zeta - \Pi^\ell(j) + \frac{\hat z_i^{\ell+1}(j)}{\mathcal{I}} - \frac{w_i(j) + \hat u_i(j)}{\mathcal{I}} \right)^2 \tag{12}$$

  subject to the system dynamics (1), $\hat x_i(0) = x_i(0)$, and the constraints (3) to obtain the unique minimizer $(\hat z_i^{\star\ell+1}(j))_{j=0}^{N-1} := (w_i(j) + \hat u_i^{\star\ell+1}(j))_{j=0}^{N-1}$.
- Transmit $(\hat z_i^{\star\ell+1}(j))_{j=0}^{N-1}$.

Remark IV.1. *Rather than solving the optimization problem* (8)*, the parameter* θ *in iteration* ℓ *can be computed by projecting the expression*

$$\tilde\theta = \frac{\sum_{j=0}^{N-1} \left(\sum_{i=1}^{\mathcal{I}} \left(\bar\zeta - \hat z_i^\ell(j) \right) \right) \left(\sum_{i=1}^{\mathcal{I}} \left(\hat z_i^{\star\ell}(j) - \hat z_i^\ell(j) \right) \right)}{\sum_{j=0}^{N-1} \left(\sum_{i=1}^{\mathcal{I}} \left(\hat z_i^{\star\ell}(j) - \hat z_i^\ell(j) \right) \right)^2}$$

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

$$F(\theta) := \sum_{j=0}^{N-1}\left(\bar{\zeta} - \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}(\theta \hat{z}_i^{\star\ell}(j) + (1-\theta)\hat{z}_i^{\ell}(j))\right)^2$$
$$= \sum_{j=0}^{N-1}\left(\bar{\zeta} - \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\hat{z}_i^{\ell}(j) - \frac{\theta}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}(\hat{z}_i^{\star\ell}(j) - \hat{z}_i^{\ell}(j))\right)^2 .$$

Since F is strictly convex, the assertion follows by setting $F'(\theta) = 0$ and projecting the resulting θ on the interval $[0,1]$. Note that the expression $-\mathcal{I}^2/2 \cdot F'(\theta)$ is given by

$$\mathcal{I}\sum_{j=0}^{N-1}\left[\left(\bar{\zeta} - \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\hat{z}_i^{\ell}(j) - \frac{\theta}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\left(\hat{z}_i^{\star\ell}(j) - \hat{z}_i^{\ell}(j)\right)\right) \cdot \sum_{i=1}^{\mathcal{I}}\left(\hat{z}_i^{\star\ell}(j) - \hat{z}_i^{\ell}(j)\right)\right]$$
$$= \sum_{j=0}^{N-1}\left(\sum_{i=1}^{\mathcal{I}}\left(\bar{\zeta} - \hat{z}_i^{\ell}(j)\right)\right)\cdot\left(\sum_{i=1}^{\mathcal{I}}\left(\hat{z}_i^{\star\ell}(j) - \hat{z}_i^{\ell}(j)\right)\right)$$
$$- \theta\sum_{j=0}^{N-1}\left(\sum_{i=1}^{\mathcal{I}}\left(\hat{z}_i^{\star\ell}(j) - \hat{z}_i^{\ell}(j)\right)\right)^2 .$$

In the case where the explicit expression for θ^ℓ is not defined, i.e.,

$$\sum_{j=0}^{N-1}\left(\sum_{i=1}^{\mathcal{I}}\left(\hat{z}_i^{\star\ell}(j) - \hat{z}_i^{\ell}(j)\right)\right)^2 = 0,$$

we have $\hat{z}_i^{\star\ell}(j) = \hat{z}_i^{\ell}(j)$ for all $(i,j) \in \{1,\ldots,\mathcal{I}\} \times \{0,1,\ldots,N-1\}$ which implies that the algorithm already found the minimum.

Remark IV.2. *Instead of using one parameter θ, one can also cluster the RESs in M groups, $1 \le M \le \mathcal{I}$ and define an optimization variable for each cluster. Then, the central entity minimizes*

$$\sum_{j=0}^{N-1}\left(\bar{\zeta} - \frac{1}{M\mathcal{I}_m}\sum_{m=1}^{M}\sum_{i=1}^{\mathcal{I}_m}\left(\theta_m \hat{z}_{m,i}^{\ell}(j) - (1-\theta_m)\hat{z}_{m,i}^{\star\ell}(j)\right)\right)^2 \tag{13}$$

with respect to the update parameter $\theta \in [0,1]^M$ where $\mathcal{I}_m$ denotes the number of RESs in the m-th cluster while the subindices in $\hat{z}_{m,i}^{\star\ell}(\cdot)$ and $\hat{z}_{m,i}^{\ell}(\cdot)$ give a numbering for the $\sum_{m=1}^{M}\mathcal{I}_m = \mathcal{I}$ RESs. Observe that the minimization problem is quadratic with (convex) box constraints and can be solved efficiently if the number of optimization variables is small compared to the number of systems. In the case that $M = 1$, we are in the situation of Remark IV.1 and have an exact solution formula which can be used as a lower bound with respect to performance. Even if the minimization problem (13) *is not solved until convergence, performing several iterations improves the outcome of the ℓ-th iteration if $\theta_i = \theta$ from Remark IV.1 is used as an initial guess.*

In every iteration, the central entity broadcasts N values (the average consumption at each time within the prediction window) and the parameter θ to all RESs. Additionally, each RES transmits N values in each iteration. Hence, the amount of data transmitted by the central entity is independent of the number of systems and the information can be broadcast instead of individually communicated to each RES. Since the optimization problems are solved by all RESs individually, the complexity of the algorithm does not grow with the number of systems. The central entity does not make use of the constraints (3). Changing system dynamics, constraints or adding/removing single systems can be achieved easily on a local level, making the algorithm readily scalable — in contrast to CMPC.

B. Convergence of Algorithm 1

In this section, we prove convergence of the proposed algorithm to the optimal value of CMPC. To this end we first formalize the crucial steps and the corresponding functions of the algorithm. For all RESs $i \in \{1,\ldots,\mathcal{I}\}$, for a fixed w_i, the constraints of the local minimization problems define compact and convex sets

$$D_i = \left\{ z_i(\cdot) \in \mathbb{R}^N \;\middle|\; \begin{array}{l} x_i(k) = x_{0_i} \\ x_i(j+1) = x_i(j) + Tu_i(j) \\ z_i(j) = w_i(j) + u_i(j) \\ \underline{u}_i \le u(j) \le \overline{u}_i \\ 0 \le x(j+1) \le C_i \\ \qquad \forall j = k,\ldots,k+N-1 \end{array} \right\}. \tag{14}$$

The corresponding parameter dependent objective function $v_i : D_i \to \mathbb{R}^+$ can be written as

$$v_i(\hat{z}_i(\cdot); p_i(\cdot)) := \sum_{j=0}^{N-1}\left(p_i(j) - \frac{1}{\mathcal{I}}\hat{z}_i(j)\right)^2 \tag{15}$$

and changes in every iteration ℓ according to the parameter $p_i^\ell(\cdot) \in \mathbb{R}^N$ given by

$$p_i^\ell(j) := \bar{\zeta} - \Pi^{\ell-1}(j) + \frac{1}{\mathcal{I}}\hat{z}_i^\ell(j)$$

for $j = 0,\ldots,N-1$.

Since v_i is strictly convex, the local minimizer

$$\hat{z}_i^\star(\cdot) := \operatorname*{argmin}_{\hat{z}_i(\cdot)\in D_i} v_i(\hat{z}_i(\cdot); p_i(\cdot))$$

is unique and since, additionally, v_i is continuous in $p_i(\cdot)$ it follows that the mapping

$$p_i(\cdot) \mapsto \hat{z}_i^\star(\cdot) \tag{16}$$

from the parameters to the minimizers is continuous.

The solutions of the local minimization problems are used by the central entity to evaluate the global objective function $V : D_1 \times \ldots \times D_{\mathcal{I}} \to \mathbb{R}^+$ defined as

$$V(\hat{z}_1(\cdot),\ldots,\hat{z}_{\mathcal{I}}(\cdot)) = \sum_{j=0}^{N-1}\left(\bar{\zeta} - \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\hat{z}_i(j)\right)^2$$

(cf. the definition of the CMPC cost function, (7)). V is continuous and convex but not necessarily strictly convex, unlike the local objective functions v_i. For the values computed by Algorithm 1 we employ the more compact notation

$$V^\ell = V(\hat{z}_1^{\ell+1}(\cdot),\ldots,\hat{z}_{\mathcal{I}}^{\ell+1}(\cdot)).$$

In the following, we will use the properties of the functions v_i and V to show convergence of Algorithm 1 to the optimal value of CMPC. First, we show that the sequence $(V^\ell)_{\ell\in\mathbb{N}_0}$ computed in (10) is nonincreasing.

Lemma IV.3. *The sequence $(V^\ell)_{\ell\in\mathbb{N}_0}$ generated by Algorithm 1 is nonincreasing, i.e. $V^{\ell+1} \leq V^\ell$ holds for all $\ell \in \mathbb{N}$. If, additionally, $\hat{z}^{\star\ell}(\cdot) \neq \hat{z}^\ell(\cdot)$, then $V^{\ell+1} < V^\ell$ holds. Hence, the sequence $(V^\ell)_{\ell\in\mathbb{N}_0}$ is monotonically decreasing until Algorithm 1 stops.*

Proof: Since $\theta^\ell \in [0,1]$ is chosen such that $F(\theta)$ attains its minimum (see Remark IV.1) replacing θ^ℓ by $\mathcal{I}^{-1}$ yields a larger value

$$\begin{aligned}
V^{\ell+1} &= \sum_{j=0}^{N-1} \left(\bar\zeta - \Pi^\ell(j)\right)^2 \\
&= \sum_{j=0}^{N-1} \left(\bar\zeta - \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} \hat{z}_i^\ell(j) + \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} \theta^\ell\left(\hat{z}_i^\ell(j) - \hat{z}_i^{\star\ell}(j)\right)\right)^2 \\
&= \sum_{j=0}^{N-1} \left(\bar\zeta - \Pi^{\ell-1}(j) + \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} \theta^\ell\left(\hat{z}_i^\ell(j) - \hat{z}_i^{\star\ell}(j)\right)\right)^2 \\
&\leq \sum_{j=0}^{N-1} \left(\frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} \left(\bar\zeta - \Pi^{\ell-1}(j) + \frac{1}{\mathcal{I}}\left(\hat{z}_i^\ell(j) - \hat{z}_i^{\star\ell}(j)\right)\right)\right)^2 \\
&\leq \sum_{i=1}^{\mathcal{I}} \frac{1}{\mathcal{I}} \underbrace{\sum_{j=0}^{N-1} \left(\bar\zeta - \Pi^{\ell-1}(j) + \frac{1}{\mathcal{I}}\left(\hat{z}_i^\ell(j) - \hat{z}_i^{\star\ell}(j)\right)\right)^2}_{=v_i(\hat{z}_i^{\star\ell}(\cdot);p_i^\ell(\cdot))} \\
&\leq \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}} v_i(\hat{z}_i^\ell(\cdot);p_i^\ell(\cdot)) = \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\sum_{j=0}^{N-1}\left(\bar\zeta - \Pi^{\ell-1}(j)\right)^2 \\
&= V^\ell.
\end{aligned}$$

The first inequality follows with $\theta^\ell = 1/\mathcal{I}$. The second inequality follows from the definition of convex functions (or Jensen's inequality), i.e.,

$$f\left(\sum_{m=1}^{M} \alpha_i x_i\right) \leq \sum_{m=1}^{M} \alpha_i f(x_i), \qquad \sum_{m=1}^{M} \alpha_i = 1,\ \alpha_i \geq 0$$

applied to $f(x) = x^2$. The third inequality is a direct consequence of the optimality of $\hat{z}_i^{\star\ell}(\cdot)$. Since $v_i(\cdot;p_i^\ell(\cdot))$ is strictly convex we obtain $\sum_{i=1}^{\mathcal{I}} v_i(\hat{z}_i^{\star\ell}(\cdot);p_i^\ell(\cdot)) < \sum_{i=1}^{\mathcal{I}} v_i(\hat{z}_i^\ell(\cdot);p_i^\ell(\cdot))$ if there exists an index $(i,j) \in \{1,2,\ldots,\mathcal{I}\} \times \{0,1,\ldots,N-1\}$ such that $\hat{z}_i^{\star\ell}(j) \neq \hat{z}_i^\ell(j)$ holds. ■

The proof of Lemma IV.3 shows that $1/\mathcal{I}$ is a possible choice for θ in Algorithm 1. Hence, the subsequent statements in this chapter also hold if the (optimal) stepsize in Algorithm 1 is replaced by the stepsize $1/\mathcal{I}$.

Corollary IV.4. *For $\ell \to \infty$ the sequence $(V^\ell)_{\ell\in\mathbb{N}_0} \subset \mathbb{R}$ of Algorithm 1 converges, i.e., $\lim_{\ell\to\infty} V^\ell = V^\star \in \mathbb{R}$.*

Proof: Since $V^\ell \geq 0$ and $(V^\ell)_{\ell\in\mathbb{N}_0}$ is monotonically decreasing by Lemma IV.3, $(V^\ell)_{\ell\in\mathbb{N}_0}$ converges to its infimum $V^\star$. ■

In Lemma IV.3 and Corollary IV.4 we have shown that the sequence $(V^\ell)_{\ell\in\mathbb{N}}$ is converging. Our remaining task, which is the main result of this paper, is to demonstrate that the limit of the sequence $(V^\ell)_{\ell\in\mathbb{N}}$ is equal to the optimal value of the centralized cost function (7).

Theorem IV.5. *The limit $V^\star$ of the sequence $(V^\ell)_{\ell\in\mathbb{N}_0}$ generated by Algorithm 1 coincides with the optimal value $V^\sharp$ of the minimization problem* (7).

Proof: Let $z^\sharp(\cdot)$ denote a solution of Problem (7), i.e.,

$$V^\sharp := V(z_1^\sharp(\cdot),\ldots,z_{\mathcal{I}}^\sharp(\cdot)).$$

For any $\tilde{z}_i(\cdot) \in D_i$, $i = 1,\ldots,\mathcal{I}$, with

$$V(\tilde{z}_1(\cdot),\ldots,\tilde{z}_{\mathcal{I}}(\cdot)) > V^\sharp \tag{17}$$

one step of Algorithm 1 with $\hat{z}_i^\ell(\cdot) = \tilde{z}_i(\cdot)$ for $i = 1,\ldots,\mathcal{I}$ yields

$$V(\hat{z}_1^{\ell+1}(\cdot),\ldots,\hat{z}_{\mathcal{I}}^{\ell+1}(\cdot)) < V(\tilde{z}_1(\cdot),\ldots,\tilde{z}_{\mathcal{I}}(\cdot)) \tag{18}$$

or

$$V(\hat{z}_1^{\ell+1}(\cdot),\ldots,\hat{z}_{\mathcal{I}}^{\ell+1}(\cdot)) = V(\tilde{z}_1(\cdot),\ldots,\tilde{z}_{\mathcal{I}}(\cdot)) \tag{19}$$

due to Lemma IV.3. If Equation (19) holds, we additionally obtain from Lemma IV.3 that $\hat{z}_i^{\ell+1}(\cdot) = \tilde{z}_i(\cdot)$ for all $i \in \{1,\ldots,\mathcal{I}\}$, i.e., the algorithm is stationary. To show that this case cannot happen while (17) holds simultaneously, we define the function $F : [0,1]^{\mathcal{I}} \to \mathbb{R}$ as

$$\begin{aligned}
F(\eta) &:= \sum_{j=0}^{N-1}\left(\bar\zeta - \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\left((1-\eta_i)\tilde{z}_i(j) + \eta_i z_i^\sharp(j)\right)\right)^2 \\
&= \sum_{j=0}^{N-1}\left(\bar\zeta - \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\tilde{z}_i(j) - \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\eta_i(z_i^\sharp(j) - \tilde{z}_i(j))\right)^2
\end{aligned}$$

for $\eta \in [0,1]^{\mathcal{I}}$. With the notation $\mathbb{1}_{\mathcal{I}} = [1,\ldots,1] \in \mathbb{R}^{\mathcal{I}}$ and $0_{\mathcal{I}} = [0,\ldots,0] \in \mathbb{R}^{\mathcal{I}}$, it holds that

$$F(\mathbb{1}_{\mathcal{I}}) = V^\sharp < V(\tilde{z}_1(\cdot),\ldots,\tilde{z}_{\mathcal{I}}(\cdot)) = F(0_{\mathcal{I}}). \tag{20}$$

Since F is convex, its directional derivative in $0_{\mathcal{I}}$ with respect to $\eta = \mathbb{1}_{\mathcal{I}}$ is less than zero, i.e.,

$$0 > \langle \operatorname{grad} F(0_{\mathcal{I}}), \mathbb{1}_{\mathcal{I}}\rangle = \sum_{i=1}^{\mathcal{I}} \frac{\partial F}{\partial \eta_i}(0_{\mathcal{I}}). \tag{21}$$

Inequality (20) implies the existence of an index $i \in \{1,\ldots,\mathcal{I}\}$ such that $z_i^\sharp(\cdot) \neq \tilde{z}_i(\cdot)$ and, thus, $0 > \frac{\partial F}{\partial \eta_i}(0_{\mathcal{I}})$ holds. However, then the i-th RES updates $\tilde{z}_i(\cdot)$, cf. (15) which contradicts the assumption $\hat{z}_i^{\ell+1}(\cdot) = \tilde{z}_i(\cdot)$. Hence inequality (18) holds for all $\tilde{z}_i(\cdot) \in D_i$, $i = 1,\ldots,\mathcal{I}$, satisfying inequality (17).

The function V is continuous and defined on a compact set. Therefore, there exists an (admissible) accumulation point $z^\star(\cdot)$ of the sequence $(\hat{z}^\ell(\cdot))_{\ell=0}^\infty$ satisfying the equality

$$V(z_1^\star(\cdot),\ldots,z_{\mathcal{I}}^\star(\cdot)) = V^\star.$$

It is clear that $V^\star \geq V^\sharp$. To show that $V^\star = V^\sharp$, assume to the contrary that $V^\star > V^\sharp$. Since the solutions of the local optimization problems depend continuously on the parameters $p_i(\cdot)$, as per (16), and the function V is continuous, the decrease property (18) at the accumulation point $z^\star(\cdot)$ implies the existence of an $\varepsilon > 0$ such that the inequality

$$V(\hat{z}_1^{\ell+1}(\cdot), \dots, \hat{z}_{\mathcal{I}}^{\ell+1}(\cdot)) < V^\star \tag{22}$$

is satisfied whenever $(z_1^\ell(\cdot), \dots, z_{\mathcal{I}}^\ell(\cdot)) \in B_\varepsilon(z_1^\star(\cdot), \dots, z_{\mathcal{I}}^\star(\cdot))$ holds, where $B_\varepsilon(\cdot)$ represents the open ball of radius $\varepsilon > 0$ centered at the specified point. Since $\hat{z}^\star(\cdot)$ is an accumulation point, there exists an index $\ell \in \mathbb{N}$ such that $(z_1^\ell(\cdot), \dots, z_{\mathcal{I}}^\ell(\cdot)) \in B_\varepsilon(z_1^\star(\cdot), \dots, z_{\mathcal{I}}^\star(\cdot))$ holds and, thus, Inequality (22) holds. However, taking the monotonicity of the sequence $(V^\ell)_{\ell \in \mathbb{N}_0}$ into account (Lemma IV.3), this contradicts the definition of $V^\star$. Hence, the assertion $V^\star = V^\sharp$ holds. ■

Theorem IV.5 is the main result of this paper. Since $V^\star$ coincides with $V^\sharp$ the performance of CMPC and DiMPC can be expected to be equal if Algorithm 1 is iterated until convergence. The following results give further insight into some properties of Algorithm 1 that can be used to increase its convergence speed. We show that the values $\Pi^\ell(j)$, $j = 0, 1, \dots, N-1$, converge for $\ell \to \infty$. Hence, the limit can be used as an initial guess in the optimization problem of the algorithm.

Lemma IV.6. *The values* $\Pi^\star(j) = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} z_i(j)$, $j \in \{0, 1, \dots, N-1\}$, *such that*

$$V^\star = \sum_{j=0}^{N-1} \left(\bar{\zeta} - \Pi^\star(j)\right)^2$$

holds are unique.

Proof: Let $\Pi^\star(j)$ and $\bar{\Pi}(j)$ both be optimal. From the definition of $\Pi^\star(j)$, $j = 0, 1, \dots, N-1$, it is straightforward that $(\Pi^\star(j)/2 + \bar{\Pi}(j)/2)_{j=0}^{N-1}$ corresponds to an admissible performance output z. Thus by strict convexity of quadratic functions we obtain

$$\begin{aligned}
&\sum_{j=0}^{N-1} \left(\bar{\zeta} - \frac{1}{2}\left(\Pi^\star(j) + \bar{\Pi}(j)\right)\right)^2 \\
&\quad < \frac{1}{2}\sum_{j=0}^{N-1} \left(\bar{\zeta} - \Pi^\star(j)\right)^2 + \frac{1}{2}\sum_{j=0}^{N-1} \left(\bar{\zeta} - \bar{\Pi}(j)\right)^2 \\
&\quad = \sum_{j=k}^{k+N-1} \left(\bar{\zeta} - \Pi^\star(j)\right)^2
\end{aligned}$$

which contradicts the optimality of $\Pi^\star(\cdot)$. ■

As a consequence of Lemma IV.4, Lemma IV.6, and the strict convexity of the function $f : \mathbb{R}^N \to \mathbb{R}$,

$$f(\Pi) = \sum_{j=0}^{N-1} \left(\bar{\zeta} - \Pi(j)\right)^2$$

we obtain the following corollary.

Corollary IV.7. *The convergence* $\Pi^\ell(j) \to \Pi^\star(j)$, $j = 0, 1, \dots, N-1$, *holds for* $\ell \to \infty$.

So far we have shown that $(V^\ell)_{\ell \in \mathbb{N}}$ and $(\Pi^\ell(\cdot))_{\ell \in \mathbb{N}}$ converge. As a final step, it would be desirable if additionally it could be shown that $(\hat{z}^\ell(\cdot))_{\ell \in \mathbb{N}}$ converges. Unfortunately we can only show that the difference between two solutions, $\hat{z}^\ell(\cdot)$ and $\hat{z}^{\ell-1}(\cdot)$, converges to zero for $\ell \to \infty$. In order to demonstrate that the stepsize $\|\hat{z}^\ell(\cdot) - \hat{z}^{\ell-1}(\cdot)\|$ converges to zero, we need the following theorem.

Theorem IV.8 ([21, Theorem 2])**.** *Let Q be a positive definite matrix and $\Theta \subset \mathbb{R}^m$ be a polyhedron. Furthermore, let $S \subset \mathbb{R}^n$ be a polytope. Consider the function $q : \Theta \to \mathbb{R} \cup \{\infty\}$ defined by the multi-parametric quadratic program*

$$q(\delta) := \left\{ \begin{array}{ll} \min_s & c^T s + \frac{1}{2} s^T Q s \\ \textit{s.t.} & As \le b + F\delta \\ & s \in S \end{array} \right. . \tag{23}$$

Then the set of feasible parameters $\Theta_f \subset \Theta$ is convex, the optimizer $s(\delta) : \Theta_f \to \mathbb{R}^n$ is unique, continuous and piecewise affine, and the value function $q(\delta) : \Theta_f \to \mathbb{R}$ is continuous, convex, and piecewise quadratic.

Theorem IV.9. *Let $(\hat{z}^\ell(\cdot))_{\ell \in \mathbb{N}_0}$be the sequence generated by Algorithm 1. Then, the stepsize $\|\Delta \hat{z}^\ell(\cdot)\|$ with $\Delta \hat{z}^\ell(\cdot) := \hat{z}^\ell(\cdot) - \hat{z}^{\ell-1}(\cdot)$ converges to zero for $\ell \to \infty$.*

Proof: With the definition of $\hat{z}^{\ell+1}(\cdot)$ in Equation (9), $\|\Delta \hat{z}^\ell(\cdot)\|$ can be written as

$$\|\hat{z}^{\ell+1}(\cdot) - \hat{z}^\ell(\cdot)\| = \theta \|\hat{z}^{\star_\ell}(\cdot) - \hat{z}^\ell(\cdot)\|.$$

Hence, showing that $\|\hat{z}^{\star_\ell}(\cdot) - \hat{z}^\ell(\cdot)\|$ converges to zero for $\ell \to \infty$ is sufficient to prove the assertion. We first write the local minimization problems (12) in the form (23). Let $\Pi^\star(j)$, $j = 0, 1, \dots, N-1$, be the unique values from Lemma IV.6. With $\delta(\cdot) := (\delta(0), \dots, \delta(N-1))^T \in \mathbb{R}^N$, the set

$$\Theta = \left\{ \delta(\cdot) \in \mathbb{R}^N | \delta(j) = \Pi^\star(j) - \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} z_i(j), z_i(\cdot) \in D_i \right\},$$

where D_i is as in (14), is compact, convex, and contains the vector $\delta(\cdot) = 0_N$. Then, for an arbitrary iteration index $\ell \in \mathbb{N}_0$, $\Pi^\ell(\cdot)$ from Algorithm 1 can be written as

$$\Pi^\ell(j) = \Pi^\star(j) - \delta^\ell(j), \qquad j \in \{0, 1, \dots, N-1\} \tag{24}$$

for $\delta^\ell(\cdot) \in \Theta$. The objective of the local minimization problem (12), in the form (15), of the RES i can be written as

$$\begin{aligned}
v_i(\hat{z}_i(\cdot); p_i^\ell(\cdot)) &= \sum_{j=0}^{N-1} \left(p_i^\ell(j) - \frac{1}{\mathcal{I}} \hat{z}_i(j)\right)^2 \\
&\overset{(24)}{=} \sum_{j=0}^{N-1} \left(\bar{\zeta} - \Pi^\star(j) + \delta^\ell(j) + \frac{1}{\mathcal{I}}\left(\hat{z}_i^\ell(j) - \hat{z}_i(j)\right)\right)^2.
\end{aligned}$$

If $\delta^\ell(\cdot) = 0_N$, the strict convexity of $v_i(\cdot; p_i(\cdot))$ implies global optimality and thus, $\hat{z}^{\star_\ell}(\cdot) = \hat{z}^\ell(\cdot)$, which shows the assertion. Otherwise, if $\delta^\ell(\cdot) \neq 0_N$, we define the coordinate transformation

$$\begin{aligned}
s_i(\delta^\ell(\cdot)) &= \delta^\ell(\cdot) + \frac{1}{\mathcal{I}}\left(\hat{z}_i^\ell(\cdot) - \hat{z}_i^{\star_\ell}(\cdot)\right) \\
&= \delta^\ell(\cdot) + \frac{1}{\mathcal{I}}\left(\hat{u}_i^\ell(\cdot) - \hat{u}_i^{\star_\ell}(\cdot)\right).
\end{aligned}$$

Then the objective function of the local minimization problem (12) can be equivalently written as

$$\min_{s_i} \frac{1}{2} s_i^T s_i + (\bar{\zeta} - \Pi^\star(0), \dots, \bar{\zeta} - \Pi^\star(N-1)) s_i \tag{25}$$

with $s_i = s_i(\delta^\ell(\cdot)) = (s_i(\delta^\ell(0), s_i(\delta^\ell(1), \ldots, s_i(\delta^\ell(N-1))^T$. (Constant values are omitted in the minimization formulation, since they do not change the optimal solution s_i.) The lower and upper bounds $\underline{u}_i \le u_i(j) \le \overline{u}_i$ in the variables s_i and $\delta^\ell(\cdot)$ read

$$\mathbb{1}_N \underline{u}_i \le \mathcal{I}\delta^\ell(\cdot) + \hat{u}_i^\ell(\cdot) - \mathcal{I}s_i \le \mathbb{1}_N \overline{u}_i \tag{26}$$
$$\Longleftrightarrow \left\{ \begin{array}{rclcl} -\mathcal{I}s_i & \le & (\mathbb{1}_N\overline{u}_i - \hat{u}_i^\ell(\cdot)) & - & \mathcal{I}\delta^\ell(\cdot) \\ \mathcal{I}s_i & \le & -(\mathbb{1}_N\underline{u}_i - \hat{u}_i^\ell(\cdot)) & + & \mathcal{I}\delta^\ell(\cdot) \end{array} \right. .$$

The constraints on the battery $0 \le x_i(j) \le C_i$ and the system dynamics $x_i(j+1) = x_i(j) + Tu_i(j)$ can be written in the form

$$0 \le \mathbb{1}_N x_i^0 + TLu_i(\cdot) \le \mathbb{1}_N C_i \tag{27}$$

where $L \in \mathbb{R}^{N\times N}$ is defined as

$$L = \begin{pmatrix} 1 & & \\ \vdots & \ddots & \\ 1 & \cdots & 1 \end{pmatrix}.$$

In the variables s_i and δ^ℓ the inequality (27) reads

$$0 \le \mathbb{1}_N x_i^0 + \mathcal{I}TL\delta^\ell(\cdot) + TL\hat{u}_i^\ell(\cdot) - \mathcal{I}TLs_i \le \mathbb{1}_N C_i \tag{28}$$
$$\Leftrightarrow \left\{ \begin{array}{rcr} -\mathcal{I}TLs_i & \le & (\mathbb{1}_N(C_i - x_i^0) - TL\hat{u}_i^\ell(\cdot)) - \mathcal{I}TL\delta^\ell(\cdot) \\ \mathcal{I}TLs_i & \le & (\mathbb{1}_N x_i^0 + TL\hat{u}_i^\ell(\cdot)) + \mathcal{I}TL\delta^\ell(\cdot) \end{array} \right.$$

Now we can apply Theorem IV.8 to the optimization problem $q(\delta^\ell)$ defined by the objective function (25) and the constraints (26) and (28) in order to obtain a piecewise affine function $s_i(\delta)$ with $s_i(0_N) = 0_N$ and

$$\frac{1}{\mathcal{I}}(\hat{z}_i^\ell(\cdot) - \hat{z}_i^{\star\ell}(\cdot)) = s_i(\delta^\ell(\cdot)) - \delta^\ell(\cdot). \tag{29}$$

Since $\delta^\ell(\cdot) \to 0_N$ for $\ell \to \infty$ (by Corollary IV.7) we obtain

$$\lim_{\ell\to\infty} \frac{1}{\mathcal{I}}(\hat{z}_i^\ell(\cdot) - \hat{z}_i^{\star\ell}(\cdot)) = 0_{\mathbb{R}^N} \tag{30}$$

for all $i = 1, \ldots, \mathcal{I}$, showing the assertion. ■

The convergence of the sequence $(\Delta\hat{z}^\ell(\cdot))_{\ell\in\mathbb{N}}$ does not imply the convergence of the sequence $(\hat{z}^\ell(\cdot))_{\ell\in\mathbb{N}_0}$. Since the optimal solution of the minimization problem (7) is not unique, the sequence can have several accumulation points with the same performance. The convergence of $(\Delta\hat{z}^\ell(\cdot))_{\ell\in\mathbb{N}}$ can in particular be used to improve the stopping criterion of Algorithm 1. A stopping criterion only based on the value $|V^{\ell-1} - V^\ell|$ might stop the algorithm early. Hence, checking additionally the values $\|\Pi^{\ell-1}(\cdot) - \Pi^\ell(\cdot)\|$ according to Corollary IV.7 and $\|\Delta\hat{z}^\ell(\cdot)\|$ according to Theorem IV.9 can help preventing stopping the algorithm too early, i.e., when the solution is still far from optimal.

Even though we can show that Algorithm 1 yields the same optimal value as the CMPC algorithm at every time step, the closed loop behavior may differ. Since the optimal solution is, in general, not unique, the choice of different optimal solutions can have an impact in the closed loop trajectory. This is not only a problem of the distributed optimization Algorithm 1, but can also appear in the CMPC algorithm if different optimization algorithms (for example active set methods or interior point methods) are used to compute the optimal solution. This behavior is illustrated in the next example.

Example IV.10. *Let $N = \mathcal{I} = 2$, $T = 1$, $w_i = \{1, 1, -1\}$, $x_i(0) = 1$ for $i = 1, 2$, and consider the constraints*

$$0 \le x_i(j) \le 2$$
$$-1 \le u_i(j) \le 1$$

for $j \in \mathbb{N}_0$ and $i = 1, 2$. For the time $k = 0$ we get $\bar{\zeta}(0) = 1$ which leads to the optimal open loop trajectories

$$\hat{u}_1(0) = [0\ 0], \qquad \hat{z}_1(0) = [1\ 1],$$
$$\hat{u}_2(0) = [0\ 0], \qquad \hat{z}_2(0) = [1\ 1].$$

For $k = 1$ we obtain $\bar{\zeta}(1) = 0$ and optimal open loop solutions are given by

$$\hat{u}_1(1) = [-1\ 1], \qquad \hat{z}_1(1) = [0\ 0],$$
$$\hat{u}_2(1) = [-1\ 1], \qquad \hat{z}_2(1) = [0\ 0].$$

For the cost function we obtain the values $V(x(0)) = V(x(1)) = 0$.

On the other hand, the values

$$\hat{u}_1(0) = [1\ 0], \qquad \hat{z}_1(0) = [2\ 1],$$
$$\hat{u}_2(0) = [-1\ 0], \qquad \hat{z}_2(0) = [0\ 1].$$

are also optimal, i.e., $V(x(0)) = 0$. In this case we obtain $x_1(1) = 2$ and $x_2(1) = 0$ but since the second battery is empty we obtain

$$\hat{u}_1(1) = [-1\ 1], \qquad \hat{z}_1(1) = [0\ 0],$$
$$\hat{u}_2(1) = [0\ 1], \qquad \hat{z}_2(1) = [1\ 0].$$

as the optimal open loop solution with $V(x(1)) = 1/4$ for the cost function.

C. Comparison to decomposition algorithms

a) Primal and dual decomposition: In this section we compare Algorithm 1 with primal and dual decomposition algorithms described in [22]. Decomposition approaches describe methods to break a single optimization problem into several optimization problems which are easier to solve. Primal decomposition refers to the decomposition of the original problem, while dual decomposition manipulates the dual formulation. Dual decomposition has been used in distributed MPC algorithms [23], [24], [25], [26] when the cost function is quadratic and separable.

Consider the minimization problem

$$\begin{array}{ll} \min\limits_{v,y} & f(v, y) \\ \text{s.t.} & (v, y) \in P \end{array} \tag{31}$$

given in [22]. Here f denotes a convex function and P a polyhedron of suitable dimension. We assume that the function f and the polyhedron P can be split such that the minimization problem (31) can be equivalently written as

$$\begin{array}{lll} \min\limits_{v,y} & \sum_{i=1}^{\mathcal{I}} f_i(v_i, y) & \\ \text{s.t.} & v_i \in P_i & \forall i = 1, \ldots, \mathcal{I} \\ & y \in P_y & \end{array} \tag{32}$$

with convex functions f_i and polyhedra P_y and P_i for $i \in \{1, \ldots, \mathcal{I}\}$. Hence the objective function is decoupled with respect to the variables v_i, and for a fixed value $y \in P_y$, one can solve the minimization problems

$$\begin{aligned} \min_{v_i} \quad & f_i(v_i, y) \\ \text{s.t.} \quad & v_i \in P_i \end{aligned} \tag{33}$$

separately. This technique of rewriting (31) as several problems of the form (33) is called *primal decomposition*. To solve the problem in a distributed way, (33) is solved for all $i \in \{1, \ldots, \mathcal{I}\}$ and a fixed value $y \in P_y$. Afterwards, the optimization variable y is updated and the process is repeated until an optimal solution is found.

In our case, the minimization problem at a fixed time step can be written as

$$\begin{aligned} \min_{v_1,\ldots,v_\mathcal{I}} \quad & \textstyle\sum_{i=1}^{\mathcal{I}} f_i(v_1, \ldots, v_\mathcal{I}) & \\ \text{s.t.} \quad & v_i \in P_i & \forall i = 1, \ldots, \mathcal{I} \end{aligned}$$

where $v_i = u_i$ and

$$f(v_1, \ldots, v_\mathcal{I}) = \sum_{j=0}^{N-1} \left(\xi(j) - \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} v_i(j) \right)^2$$

with constant values $\xi(j)$. Observe that due to the square, the function f is not separable with respect to the variables $v_1, \ldots, v_\mathcal{I}$. Additionally, an analog of the variable y does not exist in our setting. Nevertheless, it is possible to find similarities between primal decomposition and Algorithm 1. Define the values

$$y_i(j) = \xi(j) - \frac{1}{\mathcal{I}} \sum_{j=1; i \neq j}^{\mathcal{I}} \tilde{v}_i(j)$$

for given values $\tilde{v}_i(j)$. Then we can define the functions

$$f(v_i, y_i) = \frac{1}{\mathcal{I}} \sum_{j=0}^{N-1} \left(y_i(j) - \frac{1}{\mathcal{I}} v_i(j) \right)^2$$

and the corresponding minimization problems

$$\begin{aligned} \min_{v_i} \quad & f_i(v_i, y_i) \\ \text{s.t.} \quad & v_i \in P_i \end{aligned}$$

which are separated for constant values y_i or constant values $\tilde{v}_i$, respectively. Hence, the minimization problems can be solved in a distributed manner by iteratively updating $\tilde{v}_i$. A smart way of updating $\tilde{v}_i$ is given by Algorithm 1. In contrast to primal decomposition, however, we point out that in our case y_i is not an optimization variable and we need an individual y_i for every f_i.

In *dual decomposition*, the minimization problem (32) is written in the form

$$\begin{aligned} \min_{v_i, y_i} \quad & \textstyle\sum_{i=1}^{\mathcal{I}} f_i(v_i, y_i) & \\ \text{s.t.} \quad & v_i \in P_i & \forall i = 1, \ldots, \mathcal{I} \\ & y_i \in P_y & \forall i = 1, \ldots, \mathcal{I} \\ & y_i = y_j & \forall i, j = 1, \ldots, \mathcal{I}. \end{aligned} \tag{34}$$

Instead of fixing the parameter y, y_i is used as an additional optimization variable. The optimization problem (34) can be separated by looking at the Lagrangian and fixing the Lagrange variables. In *dual decomposition*, the minimization problems are solved for the unknowns (x_i, y_i) and fixed Lagrange variables for the next iteration. The Lagrange variables are updated until a solution is found. As emphasized above, the variable y does not exist in our objective function and hence, dual decomposition is not applicable in our context.

b) Distributed consensus-based primal-dual perturbation [10]: In [10] a distributed optimization algorithm applicable to our setting is presented. For our setting, the variables

$$y_j = \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} z_i(j)$$

$j = \{0, \ldots, N-1\}$ and the cost function

$$\mathcal{F}(y) = \sum_{j=0}^{N-1} \left(\bar{\zeta} - y_j \right)^2$$

have to be defined. Every system tries to find the optimal solution y. Convergence against a global optimum is achieved by communicating between neighboring systems and under the assumption that the corresponding graph is strongly connected. The iterative solutions are computed by evaluating the gradient $\nabla \mathcal{F}$, moving in a decreasing direction, and projecting the new solution on the local constraint sets.

Algorithm 1, by contrast, is a hierarchical algorithm due to the central entity, which mediates the information between RESs. The central entity does this by aggregating the information of all RESs and broadcasts only aggregate information. Note that, in the context of the considered smart grid application, customer privacy is an important goal and, using the hierarchical approach proposed herein, customers communicate only with the central entity. Furthermore, customer information must generally be shared with a central entity in this way for billing purposes. The neighbor-to-neighbor approach of [10], however, requires that detailed supply-consumption data be shared among neighbors. Additionally, in Algorithm 1, by computing the variable step size θ, the central entity makes sure, that the received data is used in an optimal way, with respect to a single iteration.

V. A Numerical Case Study

This section starts by describing a common technique in MPC applications, warm-start, as applied to the numerical implementation of Algorithm 1 for achieving the CMPC performance in a distributed fashion. Then, based on real data from an Australian electricity distribution company, Ausgrid, a numerical case study is presented in order to show the benefit of DiMPC compared to CMPC and DeMPC. An overview of the load and generation data provided by Ausgrid can be found in [20].

A. Warm-Start

The computational complexity of Algorithm 1 depends strongly on the number of iterations needed to satisfy the stopping criteria. An initialization of $\hat{z}^1(\cdot)$ with an almost optimal value of the objective function can reduce the number of iterations significantly. Since an optimization problem has to be solved at every time step in MPC, it is natural to use the (optimal) solution of the previous iteration as the initial guess at the current time instant. This method, which is called *warm-start*, can be used in our context in view of Corollary IV.7.

Let $\hat{z}^\star(\cdot;k) = (\hat{z}^\star(k;k), \hat{z}^\star(k{+}1;k), \ldots, \hat{z}^\star(k{+}N{-}1;k))^T$ and $\hat{z}^\star(\cdot;k+1) = (\hat{z}^\star(k+1;k+1), \ldots, \hat{z}^\star(k+N;k+1))^T$ denote optimal solutions at time k and time $k+1$, respectively. Here, the second argument indicates the time index at which the output value is computed. Since the underlying optimization problems only differ in the initial value $x(k)$ and $x(k+1)$, respectively, and in the shifted power demand $w(\cdot)$, we expect that the average value and the optimal performance output only change slightly, i.e., $\overline{\zeta}(k) \approx \overline{\zeta}(k+1)$ and $\hat{z}^\star(k+j;k) \approx \hat{z}^\star(k+j;k+1)$ for all $j \in \{1, \ldots, N-1\}$. Hence, we define

$$\hat{z}_i^1(j) := \hat{z}_i^\star(k+j+1;k)$$
$$\hat{u}_i^1(j) := \hat{u}_i^\star(k+j+1;k)$$

for all $i \in \{0, \ldots, \mathcal{I}\}$ and for all $j \in \{0, \ldots, N-2\}$ in Algorithm 1 instead of $\hat{z}_i^1(j) = w_i(k+j)$. Additionally, we set $\hat{u}_i^1(N-1) = 0$, which implies $\hat{z}_i^1(N-1) = w_i(k+N)$, to make sure that the initial trajectory is feasible.

B. Case Study

All minimization problems involved in the numerical experiments are solved using the Interior Point Optimizer (IPOPT) [27] and the HSL mathematical software library [28] to solve the underlying linear systems of equations.

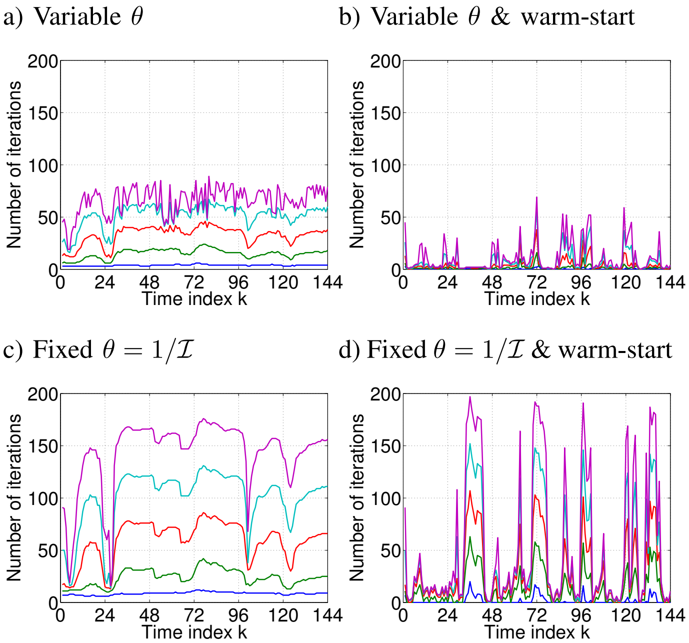
For all numerical experiments we fix the initial values $x_i(0) = 0.5[\text{kWh}]$ and the constraints $C_i = 2[\text{kWh}]$ and $\overline{u}_i = -\underline{u}_i = 0.3[\text{kW}]$ for all $i \in \{1, \ldots, \mathcal{I}\}$.

Benefits of Warm-Start and variable θ: In Lemma IV.5 we showed that the optimal value $V^\star$ obtained by the CMPC algorithm coincides with the value $\lim_{\ell\to\infty} V^\ell$ obtained by the distributed optimization Algorithm 1. In Figure 3 we visualize the number of iterations of this algorithm for a simulation length of 3 days (i.e., $\mathcal{N} = 144$, $T = 0.5[\text{h}]$), and 20 RESs, which are necessary to ensure the accuracy $|V^\ell(k) - V^\star(k)| \leq 10^{-i}$ for $i = 1, \ldots, 5$. Additionally, Figure 3 shows the importance of the step length θ. If the fixed value $\theta = 1/\mathcal{I}$ is used instead of a variable θ according to the minimization problem (8), approximately twice as many iterations are necessary to obtain a certain accuracy. A variable θ in combination with warm-start reduces the number of iterations even further.

In Figure 4 we visualize

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

i.e., the average deviation from the optimal solution computed by the centralized MPC algorithm in iteration ℓ. The average is with respect to the simulation length $\mathcal{N} = 144$. The figure illustrates the rate of convergence of the distributed optimization algorithm using fixed $\theta = 1/\mathcal{I}$ and variable θ with and without warm-start. We can observe that the method consists of two phases with a linear rate of convergence. 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.


a) Variable θ

b) Variable θ & warm-start

c) Fixed $\theta = 1/\mathcal{I}$

d) Fixed $\theta = 1/\mathcal{I}$ & warm-start

Fig. 3. *Number of iterations needed to ensure* $|V^\ell(k) - V^\star(k)| \leq 10^{-i}$ *for* $i = 1, \ldots, 5$ *at time* k. $V^\star$ *denotes the solution of CMPC.*

	without warm-start		with warm-start	
	$\theta = 1/\mathcal{I}$	variable θ	$\theta = 1/\mathcal{I}$	variable θ
av. no. 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
max. no. 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
min. no. 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 I

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

Impact of the Number of Systems: In Figure 5 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 until the accuracy $|V^\ell - V^\star| \leq \varepsilon$ 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 systems. 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

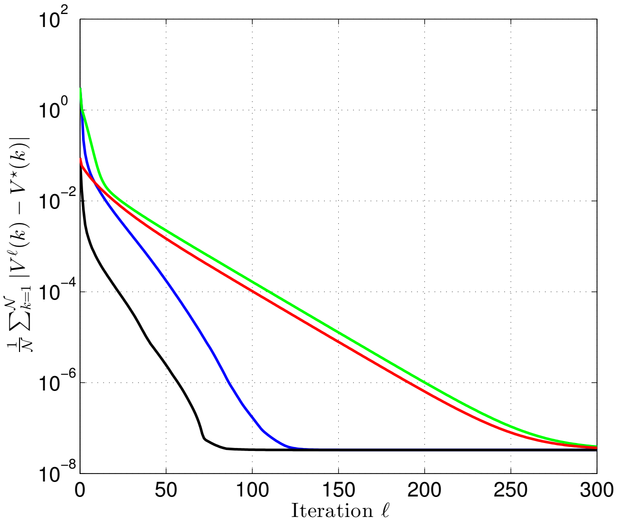


Fig. 4. *Average speed of convergence of the distributed optimization algorithm with fixed $\theta = 1/\mathcal{I}$ with (red) and without (green) warm-start and variable θ with (black) and without (blue) warm-start.*

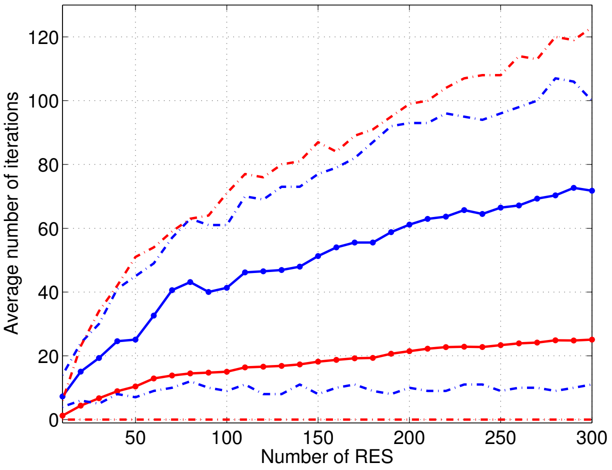


Fig. 5. *Average number of iterations needed to obtain the accuracy $|V^{\ell} - V^{\star}| \leq 10^{-2}$ for a different number of RESs without warm-start (blue) and with warm-start (red). The dashed lines show the maximal and minimal number of iterations.*

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).

Imperfect Optimization: In Figure 5 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 always perform a fixed number of iterations at every time step, we can conclude that 3 iterations are sufficient to obtain a closed loop performance which is close to CMPC (cf. Figure 7 and Table II), even though the difference to the optimal value is still large (cf. Figure 6) and, in most of the cases, greater than 10^{-2}. Note that Figure 6 only shows the first 144 time steps. The missing time steps show a similar behavior and are not visualized to keep the description clear.)

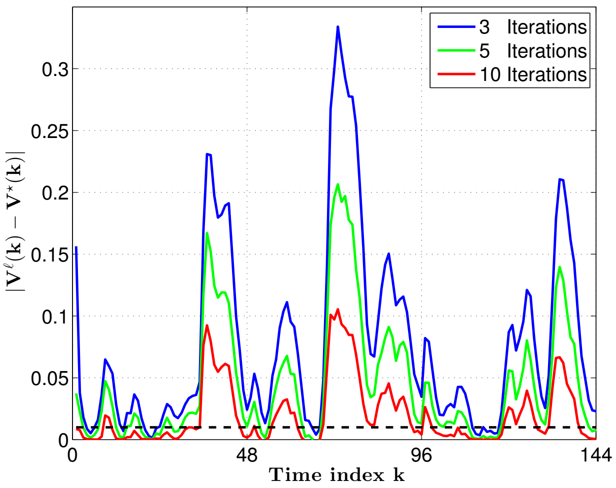


Fig. 6. *Comparison of the optimal value function of centralized MPC and distributed MPC for 100 RESs using warm-start and incomplete optimization.*

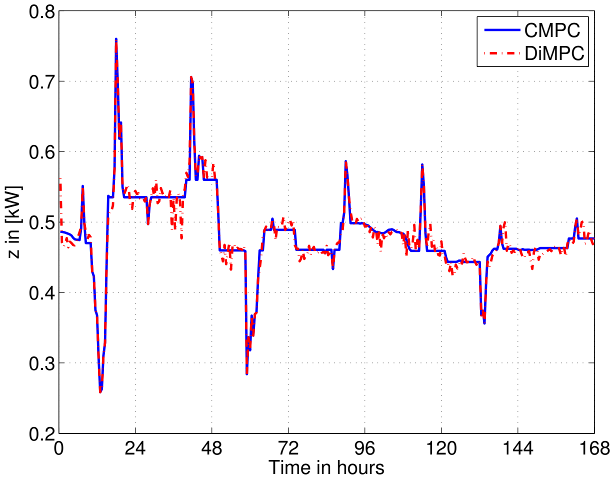


Fig. 7. *Comparison of the closed loop behavior of CMPC and DMPC for 100 RESs using warm-start and 3 iterations at every time step.*

	PTP	RMS
Centralized MPC		
	0.5016	0.0568
Dist MPC without warm-start		
$\ell = 3$	0.5016	0.0780
$\ell = 5$	0.5016	0.0670
$\ell = 10$	0.5016	0.0609
Dist MPC with warm-start		
$\ell = 3$	0.5016	0.0583
$\ell = 5$	0.5016	0.0574
$\ell = 10$	0.5016	0.0569

TABLE II
CMPC performance and performance of distributed MPC for 100 RESs and incomplete optimization.

Remark V.1. *For the considered dataset in this section, i.e., the 144 samples and a variable number of RESs, the values $V^{\star}$ are in the interval $[0.054, 1.850]$. A large (small) $V^{\star}$ corresponds to a large (small) deviation from the average $\overline{\zeta}$.*

Therefore, we use the absolute error

$$\left|V^{\ell} - V^{\star}\right| \leq \epsilon$$

instead of the relative error

$$\left|V^{\ell} - V^{\star}\right| \leq \epsilon \cdot V^{\star}$$

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

VI. Conclusion

In this paper we presented a novel distributed optimization algorithm for the receding horizon control of a distributed electricity network, based on communication with a central entity. Optimality was established for the corresponding centralized optimization problem. The communication structure of the algorithm is independent of the number of RESs connected to the grid and the number of iterations needed at every time step is reduced significantly by optimizing the step length in every iteration. Additionally, we showed that the number of iterations increases very moderately with the number of RESs if the algorithm is combined with a warm start method. Indeed, already very few iterations are sufficient to obtain a result comparable to CMPC if the algorithm is applied in an MPC context.

Future work will concentrate on the benefit to individual RESs. While in the presented algorithm the motivation of an RES in minimizing the fluctuation in the network are not clear, we will investigate algorithms with a similar communication structure, but with cost functions representing real energy prices. Hence, in this future work, the goal of the grid operator (minimizing the fluctuations in the power demand) can be interpreted as a minimization of the electricity costs for each individual RES.

References

[1] J. N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, MIT, Cambridge, MA, USA, 1984.

[2] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, Belmont, MA, USA, 1989.

[3] 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.

[4] D. Jakovetić, J. Xavier, and J. M. F. Moura. Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146, 2014.

[5] 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.

[6] 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.

[7] A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[8] 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.

[9] M. Zhu and S. Martínez. On distributed convex optimization under inequality and equality constraints. *IEEE Transactions on Automatic Control*, 57(1):151–164, 2012.

[10] 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.

[11] 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.

[12] 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*, 2012.

[13] 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.

[14] 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.

[15] 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, March 2015.

[16] 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*, 2015. Doi: 10.1109/TSG.2015.2392081.

[17] K. Worthmann, C. M. Kellett, L. Grüne, and S. R. Weller. Distributed control of residential energy systems using a market maker. In *19th IFAC World Congress, South Africa*, pages 11641–11646, 2014.

[18] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.

[19] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control. Theory and Algorithms*. Springer London, 2011.

[20] E. L. Ratnam, S. R. Weller, C. M. Kellett, and A. Murray. Residential load and rooftop PV generation: An Australian distribution network dataset. submitted to *Energy Reports*, May 2015.

[21] V. Sakizlis, K. I. Kouramas, and E. N. Pistikopoulos. *Linear Model Predictive Control via Multiparametric Programming*, volume 1. Wiley-VCH Verlag GmbH & Co. KGaA, 2007. Eds. E. N. Pistikopoulos and M. C. Georgiadis and V. Dua.

[22] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley. Notes on decomposition methods. Technical report, Stanford University, 2007.

[23] 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*, 49:829–833, 2013.

[24] 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.

[25] 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.

[26] P. Giselsson and A. Rantzer. Distributed model predictive control with suboptimality and stability guarantees. In *Proc. 49th IEEE Conference on Decision and Control*, pages 7272–7277, Atlanta, GA, USA, December 2010.

[27] 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.

[28] HSL Mathematical Software Library. A collection of Fortran codes for large-scale scientific computation, 2004. http://hsl.rl.ac.uk/.