

Predictive Control of a Smart Grid: A Distributed Optimization Algorithm with Centralized Performance Properties*

Philipp Braun¹, Lars Grüne¹, Christopher M. Kellett², Steven R. Weller², and Karl Worthmann³

Abstract—The authors recently proposed several model predictive control (MPC) approaches to managing residential level energy generation and storage, including centralized, distributed, and decentralized schemes. As expected, the distributed and decentralized schemes result in a loss of performance but are scalable and more flexible with regards to network topology. In this paper we present a distributed optimization approach which asymptotically recovers the performance of the centralized optimization problem performed in MPC at each time step. Simulations using data from an Australian electricity distribution company, Ausgrid, are provided showing the benefit of a variable step size in the algorithm and the impact of an increasing number of participating residential energy systems. Furthermore, when used in a receding horizon scheme, simulations indicate that terminating the iterative distributed optimization algorithm before convergence does not result in a significant loss of performance.

I. INTRODUCTION

With the proliferation of residential rooftop solar photovoltaics and the increasing availability of cost-effective residential-scale energy storage solutions (e.g., batteries or fuel cells), there is a need to coordinate the storage charge/discharge schedules so as to avoid large demand peaks or troughs. In [14], [13], the authors proposed three different model predictive control (MPC) schemes to smooth the energy demand of a collection of residences. These MPC schemes involved a centralized approach, requiring full communication of all relevant system variables, a distributed approach, requiring limited communication of relevant system variables, and a decentralized approach, requiring no communication of system variables. While all three approaches succeeded in smoothing the aggregate energy demand, unsurprisingly the centralized approach achieved better performance when compared to the distributed and decentralized approaches, but suffered from an inability to scale to a large number of residential systems.

In this paper, we present a distributed optimization algorithm with the goal of recovering the performance of the centralized MPC scheme whilst remaining scalable. In other

words, we focus on the solution of a single, finite time horizon, optimization problem implemented in a distributed fashion.

At least in the control literature, the field of distributed optimization traces its roots to the thesis of Tsitsiklis [11] (see also [2]). Much of the recent work in this field has involved multi-agent systems trying to optimize a global objective function under different conditions; see for example [5], [7], [8], [9], [15] and the references therein. A common feature in many of these references is the assumption that the global cost function can be decomposed as a sum of the cost functions for each individual agent. However, the cost function naturally used to solve the problem of smoothing the energy demand is not decomposable in this way.

In [4], a closely related problem is solved where an electricity retailer aims to minimize the cost due to discrepancies between the power the retailer bids to use and what its customers actually use. Again, this gives rise to a different cost function to that which we propose.

The paper is organized as follows. In Section II we introduce the mathematical model of the Residential Energy System (RES) and define the desired performance metrics. The centralized MPC approach is presented in Section III and our proposed distributed computation algorithm is described in Section IV-A. A brief comparison with primal/dual decomposition is provided in Section IV-B. A simulation study using data from an Australian electricity distribution company, Ausgrid, is undertaken in Section V. In particular, we demonstrate the benefit of a varying step-size in the distributed optimization algorithm (Section V-A), we examine the impact of increasing the number of systems (Section V-B), and the effect of early termination of the distributed optimization algorithm is illustrated (Section V-C). Concluding remarks are provided in Section VI.

II. THE RESIDENTIAL ENERGY SYSTEM

Let $\mathcal{I} \in \mathbb{N}$ be the number of RESs connected in the local area under consideration. We summarize a simple model of RES i , $i \in \{1, \dots, \mathcal{I}\}$, presented in [13]

$$x_i(k+1) = x_i(k) + Tu_i(k), \quad (1)$$

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

where x_i is the state of charge of the battery in [kWh], u_i is the battery charge/discharge rate in [kW], w_i is the static load minus the local generation in [kW], and z_i is the power supplied by/to the grid in [kW]. Here, T represents the length of the sampling interval in [h] (hours); e.g., $T = 0.5$ [h] corresponds to 30 minutes. While the system dynamics (1)

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

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

²C. 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.

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

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 $i \in \{1, \dots, \mathcal{I}\}$, the constraints on the battery capacity and charge/discharge rates are described by the constants $C_i, \bar{u}_i \in \mathbb{R}_{>0}$ and $\underline{u}_i \in \mathbb{R}_{<0}$, i.e.,

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

Our goal is to flatten the performance output z . We introduce two relevant performance metrics. To this end, let

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

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

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

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

$$\sqrt{\frac{1}{\mathcal{N}} \sum_{k=0}^{\mathcal{N}-1} (\Pi(k) - \mathcal{Y})^2} \quad (\text{RMS})$$

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

III. MODEL PREDICTIVE CONTROL APPROACHES

We recall a model predictive control (MPC) algorithm for the control of a network of RESs introduced in [13] and [14], respectively. This 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. In Section IV-A we present a distributed optimization algorithm which is based on local optimization problems, keeping the flexibility of the network topology, while maintaining optimality with respect to the CMPC approach. A corresponding proof of convergence is given in the Appendix.

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., [10] or [6]). 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 $j \in \{k, \dots, k+N-1\}$, where $k \in \mathbb{N}_0$ is the current time.

A. Centralized Model Predictive Control (CMPC)

To implement the CMPC algorithm, we compute the network-wide average demand at every time step k over the prediction horizon by

$$\bar{\zeta}(k) := \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \left(\frac{1}{N} \sum_{j=k}^{k+N-1} w_i(j) \right) \quad (4)$$

and then minimize the joint cost function

$$V(x(k); k) := \min_{\hat{u}(\cdot)} \sum_{j=k}^{k+N-1} \left(\bar{\zeta}(k) - \underbrace{\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} (w_i(j) + \hat{u}_i(j))}_{\hat{z}_i(j)} \right)^2 \quad (5)$$

with respect to the predicted control inputs $\hat{u}(\cdot) = (\hat{u}_1(\cdot), \hat{u}_2(\cdot), \dots, \hat{u}_{\mathcal{I}}(\cdot))^T$ with $\hat{u}_i(\cdot) = (\hat{u}_i(j))_{j=k}^{k+N-1}$, $i \in \{1, 2, \dots, \mathcal{I}\}$, subject to the system dynamics (1), the current state $x(k) = (x_1(k), \dots, x_{\mathcal{I}}(k))^T$, and the constraints (3) for $i \in \{1, \dots, \mathcal{I}\}$. 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 deliver extra information. Additionally we use the notation $u(j) = (u_1(j), \dots, u_{\mathcal{I}}(j))^T$ for a fixed time $j \in \mathbb{N}$. The same holds for the other variables x , w and z .

In Figure 1 the aggregated energy profile and the aggregated battery profile for a simulation of one week ($\mathcal{N} = 336$, $T = 0.5[\text{h}]$) for 100 RESs, initial conditions $x_i(0) = 0.5[\text{kWh}]$, constraints $\bar{u}_i = -\underline{u}_i = 0.3[\text{kW}]$ and battery capacity $C_i = 2[\text{kWh}]$ for all $i \in \{1, \dots, 100\}$ are visualized. The load and generation data for this simulation was collected by an Australian electricity distribution company, Ausgrid, as part of their *Smart Grid, Smart City* project. The figures compare the uncontrolled system dynamics with the closed loop dynamics of CMPC.

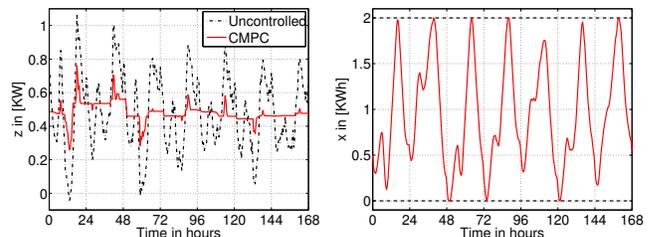


Fig. 1. Performance of CMPC for a simulation length of one week and 100 RES. The left figure shows the average power demand while the right figure shows the average state of charge of the batteries.

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 to achieve the performance of the CMPC algorithm, i.e., a network-wide objective while keeping flexibility. The optimal value returned by the distributed optimization algorithm coincides with the optimal value of the minimization problem (5) (cf. the Appendix for a proof).

A. The Distributed Optimization Algorithm

The distributed optimization algorithm 1 is based on the cost function (5) 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 an appropriate step size θ in every iteration.

At time step k , the algorithm is initialized with $\bar{\zeta} := \bar{\zeta}(k)$ (cf. Equation (4)), $w_i(j) := w_i(k+j)$, $j = 0, \dots, N-1$, $i \in \{1, 2, \dots, \mathcal{I}\}$, and $x(0) := x(k)$.

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 , energy profile $(w_i(j))_{j=0}^{N-1}$, and $\bar{\zeta}$.
- **Central Entity**: Number of RESs \mathcal{I} , N , $\bar{\zeta}$, 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 $(\hat{z}_i^{*1}(j))_{j=0}^{N-1}$ and $(\hat{z}_i^1(j))_{j=0}^{N-1}$.
- **Central Entity**: Set the iteration counter $\ell = 0$ and $V^1 = \infty$, receive $(\hat{z}_i^1(j))_{j=0}^{N-1}$, $i \in \{1, 2, \dots, \mathcal{I}\}$.

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

- Compute the step size θ^ℓ as

$$\operatorname{argmin}_{\theta \in [0,1]} \sum_{j=0}^{N-1} \left(\bar{\zeta} - \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \theta \hat{z}_i^{*\ell}(j) + (1-\theta) \hat{z}_i^\ell(j) \right)^2 \quad (6)$$

- Compute $\hat{z}^{\ell+1}(j) := \theta^\ell \hat{z}^{*\ell}(j) + (1-\theta^\ell) \hat{z}^\ell(j)$ and the predicted average demand $\Pi^\ell(j) := \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \hat{z}_i^{\ell+1}(j)$ for $j \in \{0, 1, \dots, N-1\}$. Then, evaluate the performance index

$$V^{\ell+1} := \sum_{j=0}^{N-1} (\bar{\zeta} - \Pi^\ell(j))^2. \quad (7)$$

- If $|V^{\ell+1} - V^\ell| < \varepsilon$ or $\ell \geq \ell_{\max}$ 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

$$\hat{z}_i^{\ell+1}(j) := \theta^\ell \hat{z}_i^{*\ell}(j) + (1-\theta^\ell) \hat{z}_i^\ell(j) \quad (8)$$

- Solve the (local) minimization problem

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

subject to the system dynamics (1), $\hat{x}_i(0) = x_i(0)$, and the constraints (3) to obtain the unique minimizer $(\hat{z}_i^{*\ell+1}(j))_{j=0}^{N-1} := (w_i(j) + \hat{u}_i^{*\ell+1}(j))_{j=0}^{N-1}$.

- Transmit $(\hat{z}_i^{*\ell+1}(j))_{j=0}^{N-1}$.
-

Note that $\Pi^\ell(\cdot)$ only depends on $\hat{z}^{\ell+1}(\cdot)$. The index is chosen in such a way that in iteration ℓ , the predicted average $\Pi^\ell(\cdot)$ has to be transmitted. A feasible initialization of RES i , $i \in \{1, \dots, \mathcal{I}\}$ is for example given by $\hat{z}_i^{*1}(j) = \hat{z}_i^1(j) := w_i(j)$, which corresponds to the choice $u^1(\cdot) \equiv 0$ and can be replaced by any other admissible initialization. Algorithm 1 is terminated either after a fixed number of iterations ℓ_{\max} or if the stopping criteria $|V^{\ell+1} - V^\ell| < \varepsilon$ is satisfied. The communication structure of Algorithm 1 is visualized in Figure 2.

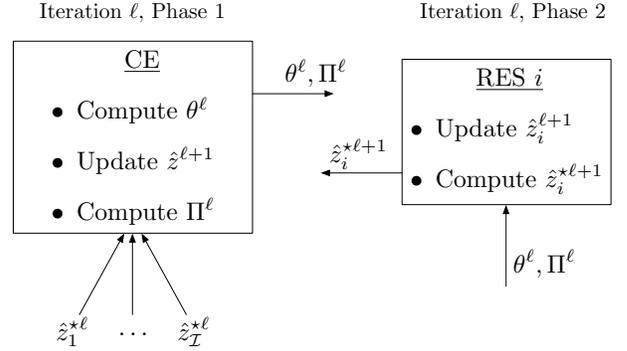


Fig. 2. Communication structure of Algorithm 1.

The input $u(k)$ is defined by the update rule of Equation (8), i.e., as a convex combination of the last two computed inputs. Since the constraints define a convex set it is ensured that $\hat{z}_i^{\ell+1}(\cdot)$ corresponds to an admissible input sequence in every iteration. Theorem 1.4 in the Appendix ensures that the value V^ℓ converges to the unique optimal value if the iteration index ℓ tends to infinity. Moreover, note that solving the minimization problem (6) is equivalent to a simple function evaluation as proven in the following proposition.

Proposition 4.1: If $(\hat{z}^{*\ell}(j))_{j=0}^{N-1} \neq (\hat{z}^\ell(j))_{j=0}^{N-1}$, the parameter θ^ℓ in iteration ℓ is given by the projection of

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

to the interval $[0, 1]$, i.e., $\theta = \max\{0, \min\{\tilde{\theta}, 1\}\}$.

Proof: In order to show the assertion, we define the function

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

Since F is strictly convex, the assertion follows by solving $F'(\tilde{\theta}) = 0$ and projecting the solution on the interval $[0, 1]$. Hence, showing that $\tilde{\theta}$ solves $F'(\tilde{\theta}) = 0$ completes the proof.

This follows by computing $-\mathcal{I}^2/2 \cdot F'(\theta)$:

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

with $\Delta_i^\ell(j) = \hat{z}_i^{\star\ell}(j) - \hat{z}_i^\ell(j)$. ■

Remark 4.2: Alternatively to the variable step size θ^ℓ computed in Equation (6), the fixed step size $\theta^\ell = 1/\mathcal{I}$ leads to a decrease of the optimal value V^ℓ in every iteration and convergence to the optimal value of the CMPC minimization problem which is an immediate consequence of the proof of Lemma 1.2. In Section V the impact of a fixed and a variable step size is illustrated by simulations.

In every iteration, the central entity communicates N values (the average consumption at each time within the prediction window) and the parameter θ to all RESs. In the reverse direction, 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. Since the optimization problems are solved by the 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 nicely scalable — in contrast to CMPC.

B. Comparison to primal and dual decomposition

In this section we compare Algorithm 1 with primal and dual decomposition algorithms described in [3]. 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.

Consider the minimization problem

$$\begin{aligned} & \min_{v, y} f(v, y) \\ & \text{s.t.} \quad (v, y) \in P \end{aligned} \quad (9)$$

given in [3]. Here f denotes a convex function and P a polyhedron of suitable dimension. Assume that the function f and the polyhedron P can be split such that the minimization problem (9) can be equivalently written as

$$\begin{aligned} & \min_{v, y} \sum_{i=1}^{\mathcal{I}} f_i(v_i, y) \\ & \text{s.t.} \quad v_i \in P_i \quad \forall i = 1, \dots, \mathcal{I} \\ & \quad y \in P_y \end{aligned} \quad (10)$$

with convex functions f_i and polyhedra P_y and P_i for $i \in \{1, \dots, \mathcal{I}\}$. 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

$$\begin{aligned} & \min_{v_i} f_i(v_i, y) \\ & \text{s.t.} \quad v_i \in P_i \end{aligned} \quad (11)$$

separately. This technique of rewriting (9) as several problems of the form (11) is called *primal decomposition*. To solve the problem in a distributed way, (11) is solved for all $i \in \{1, \dots, \mathcal{I}\}$ 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 (5) can be written as

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

where $v_i = u_i$ and

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

with constant values $\xi(j)$. Observe that due to the square, the function f is not separable with respect to the variables $v_1, \dots, v_{\mathcal{I}}$. 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. We 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

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

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 . One 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 (10) is written in the form

$$\begin{aligned} & \min_{v_i, y_i} \sum_{i=1}^{\mathcal{I}} f_i(v_i, y_i) \\ & \text{s.t.} \quad v_i \in P_i \quad \forall i = 1, \dots, \mathcal{I} \\ & \quad y_i \in P_y \quad \forall i = 1, \dots, \mathcal{I} \\ & \quad y_i = y_j \quad \forall i, j = 1, \dots, \mathcal{I}. \end{aligned} \quad (12)$$

Instead of fixing the parameter y , y_i is used as an additional optimization variable. The optimization problem (12) 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.

V. A NUMERICAL CASE STUDY

A numerical case study is presented in order to show the benefit of DiMPC compared to CMPC. This case study is based on anonymized load and generation profiles of residential customers provided by an Australian electricity distribution company, Ausgrid, based in New South Wales.

The numerical experiments are conducted using the interior point solver *IPOPT* [12] and the *HSL* mathematical software library [1] to solve the underlying minimization problems and linear systems of equations, respectively. For all numerical experiments we fix the initial values $x_i(0) = 0.5[\text{kWh}]$ and the constraints $C_i = 2[\text{kWh}]$, $\bar{u}_i = -\underline{u}_i = 0.3[\text{kW}]$ for all $i \in \{1, \dots, \mathcal{I}\}$.

A. Choice of the Step Size θ

In this subsection we investigate the role of the step length θ . To this end, 20 RESs are simulated for a duration of 3 days ($\mathcal{N} = 144$, $T = 0.5[\text{h}]$). In Figure 3 we visualize the number of iterations until a certain accuracy $|V^\ell(k) - V^*(k)| \leq 10^{-i}$, $i \in \{1, 2, \dots, 5\}$, is reached.

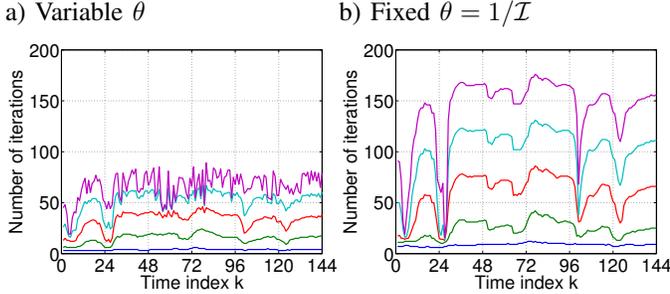


Fig. 3. Number of iterations to obtain a certain accuracy $\varepsilon_i = 10^{-i}$ for $i = 1, \dots, 5$ at time step k , i.e., $|V^\ell(k) - V^*(k)| \leq \varepsilon_i$ where V^* denotes the solution of the centralized MPC algorithm.

If a fixed step size $\theta = 1/\mathcal{I}$ is used instead of a variable θ according to Proposition 4.1 the required number of iterations is, on average, twice as large, see Table I.

Accuracy	average no.		maximum no.		minimum no.	
	$1/\mathcal{I}$	variable	$1/\mathcal{I}$	variable	$1/\mathcal{I}$	variable
$\varepsilon = 10^{-1}$	8.61	3.81	12	6	6	3
$\varepsilon = 10^{-2}$	23.90	15.05	42	24	10	6
$\varepsilon = 10^{-3}$	59.33	33.04	86	46	13	11
$\varepsilon = 10^{-4}$	99.85	51.44	131	67	14	16
$\varepsilon = 10^{-5}$	142.69	65.89	176	89	16	19

TABLE I

Average, minimum, and maximum number of iterations to achieve a certain accuracy for variable and fixed θ .

In Figure 4 the average deviation in iteration ℓ from the benchmark CMPC solution, i.e., $\frac{1}{\mathcal{N}} \sum_{k=0}^{\mathcal{N}-1} |V^\ell(k) - V^*(k)|$, is visualized. The average is taken with respect to each sampling instant k with simulation length $\mathcal{N} = 144$. Hence, the convergence speed of the distributed optimization algorithm with step size θ in accordance with Proposition 4.1 clearly outperforms its counterpart using constant $\theta = 1/\mathcal{I}$. The

constant line after approximately 120 iterations is due to the optimization accuracy of *IPOPT*.

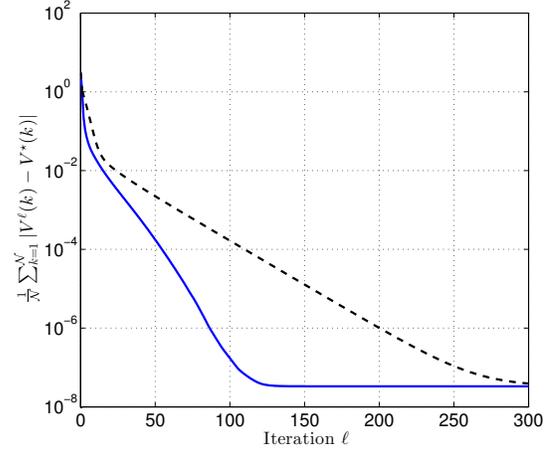


Fig. 4. Average speed of convergence of the distributed optimization algorithm with fixed $\theta = 1/\mathcal{I}$ (black) and variable θ (blue).

B. Impact of the Number of Systems

Next, we analyze the dependence of the (average) number of iterations on the number of RESs. To this end, the number of RESs, \mathcal{I} , is varied within the set $\{10, 20, \dots, 300\}$. Then, the number of iterations is counted until the accuracy $|V^\ell - V^*| \leq 10^{-2}$ is obtained both for variable and fixed step size θ . In Figure 5, we observe a linear growth in the number of iterations for fixed θ while this number is significantly smaller and seems to grow sublinearly in the case of variable θ . In conclusion, the number of iterations stays moderate for variable θ while it may become too large for $\theta = 1/\mathcal{I}$ to make the algorithm applicable for a (very) large number of RESs.

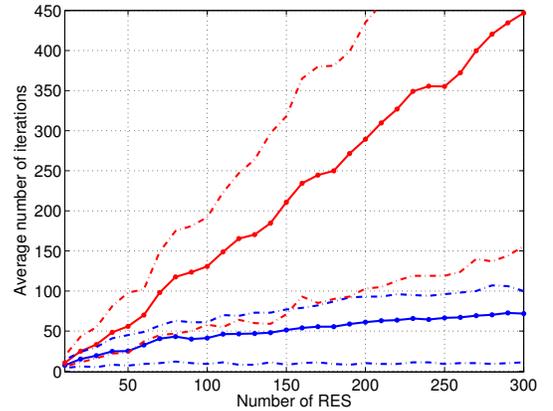


Fig. 5. Average number of iterations needed to ensure the accuracy $|V^\ell - V^*| \leq 10^{-2}$ in dependence of the number of RES with fixed $\theta = 1/\mathcal{I}$ (red) and with variable θ (blue). The dashed lines show the maximal and minimal number of iterations.

C. Imperfect Optimization

Algorithm 1 needs about 42 iterations on average to obtain an accuracy of 10^{-2} in the setting of 100 RESs and

variable θ , cf. Figure 5.

However, in practice, it may be necessary to terminate the algorithm after a fixed number of iterations; e.g., due to a fixed allowable computation time. We examine two issues. The first is merely the performance of Algorithm 1 with a fixed number of iterations. The second is the closed loop performance of Algorithm 1 with a fixed number of iterations when used in a receding horizon fashion. We first compute the deviation $|V^\ell(k) - V^*(k)|$ at each time instant k within the simulation window and, then, we analyze the MPC closed loop performance. If the step size θ is chosen such that (6) is solved in each iteration the total deviation is still large after 10 iterations, but the closed loop performance already looks convincing, see Figure 6.

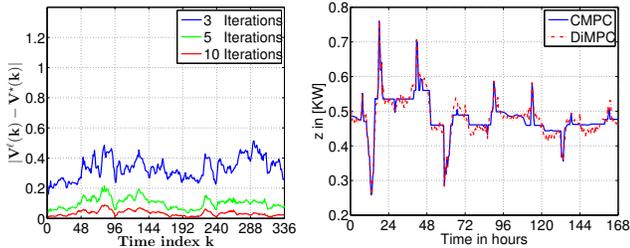


Fig. 6. Deviation and MPC closed loop evolution for 100 RESs using variable θ and incomplete optimization (10 iterations).

On the contrary, the closed loop performance is not satisfactory for fixed θ as seen in Figure 7.

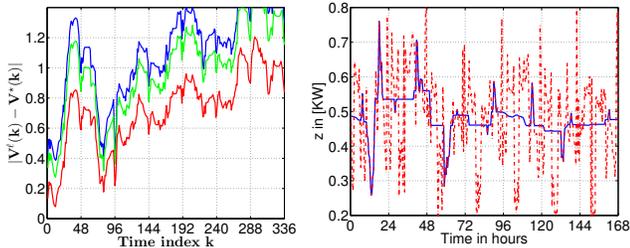


Fig. 7. Deviation and MPC closed loop evolution for 100 RESs using fixed θ and incomplete optimization (10 iterations).

The same conclusions can be drawn for even smaller iteration numbers (see Table II).

Number of iterations	variable θ		$\theta = 1/\mathcal{I}$	
	PTP	RMS	PTP	RMS
DiMPC - CMPC				
$\ell = 3$	0.0000	0.0212	0.2914	0.1033
$\ell = 5$	0.0000	0.0102	0.2592	0.0942
$\ell = 10$	0.0000	0.0041	0.2028	0.0759

TABLE II

Deviation of Distributed MPC with incomplete optimization and CMPC for 100 RES — in dependence of the step size θ .

Remark 5.1: For the considered data set in this section, i.e., the 144 samples and a variable number of RESs, the values of V^* are in the interval $[0.054, 1.850]$. A large (small)

V^* corresponds to a large (small) deviation from the average $\bar{\zeta}$. Therefore, we use the absolute error

$$|V^\ell - V^*| \leq \epsilon$$

instead of the relative error

$$|V^\ell - V^*| \leq \epsilon \cdot V^*$$

as a qualitative measure of the results. If V^* is small the performance with respect to our metrics is good even if the relative error might still be large. The choice $\epsilon = 10^{-2}$ 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 have presented a distributed optimization algorithm for the application to the problem of smoothing energy consumption in a residential electricity network where residences have small scale generation (e.g., rooftop solar photovoltaic panels) and storage (e.g., a battery). This iterative message-passing algorithm asymptotically recovers the optimal value of the centralized optimization problem. Via a simulation study, the distributed optimization algorithm has been shown to scale well with the number of systems and, when used in an MPC scheme, to retain good performance when the algorithm is terminated after a fixed number of iterations. Furthermore, we have demonstrated the benefit of implementing a variable step size.

APPENDIX

In this section, we prove convergence of Algorithm 1 to the optimal value of (5), i.e., we show that the limit $V^* := \lim_{\ell \rightarrow \infty} V^\ell$ corresponding to Algorithm 1 coincides with the optimal value $V^\#$ of the minimization problem

$$\begin{aligned}
 \min_{\hat{z}_1(\cdot), \dots, \hat{z}_{\mathcal{I}}(\cdot)} & \sum_{j=0}^{N-1} \left(\bar{\zeta} - \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \hat{z}_i(j) \right)^2 \\
 \text{s.t.} & x(0) = \hat{x}(0) \\
 & x_i(j+1) = x_i(j) + T u_i(j) \\
 & z_i(j) = w_i(j) + u_i(j) \\
 & \underline{u}_i \leq u_i(j) \leq \bar{u}_i \\
 & 0 \leq x_i(j+1) \leq C_i \\
 & \forall (i, j) \in \{1, \dots, \mathcal{I}\} \times \{0, \dots, N-1\}
 \end{aligned} \tag{13}$$

which has to be solved in every time step of CMPC.

To this end, we define the functions

$$v_i(\hat{z}_i(\cdot); \ell) := \sum_{j=0}^{N-1} \left(\bar{\zeta} - \Pi^\ell(j) + \frac{1}{\mathcal{I}} (\hat{z}_i^\ell(j) - \hat{z}_i(j)) \right)^2 \tag{14}$$

and rewrite the local minimization problem from Phase 2 of

Algorithm 1

$$\begin{aligned}
& \min_{\hat{z}_i(\cdot)} v(\hat{z}_i(\cdot); \ell) \\
& \text{s.t.} \quad x_i(0) = \hat{x}_i(0) \\
& \quad \quad x_i(j+1) = x_i(j) + T u_i(j) \\
& \quad \quad z_i(j) = w_i(j) + u_i(j) \\
& \quad \quad \underline{u}_i \leq u_i(j) \leq \bar{u}_i \\
& \quad \quad 0 \leq x_i(j+1) \leq C_i \\
& \quad \quad \forall j \in \{0, \dots, N-1\}
\end{aligned} \tag{15}$$

for $i \in \{1, \dots, \mathcal{I}\}$. The constraints of (15) define a convex and compact set. The function v_i is strictly convex and continuous in $\hat{z}_i(\cdot)$ and in the parameters $\bar{\zeta} - \Pi^\ell(j) + \hat{z}_i^\ell(j)/\mathcal{I}$, $j \in \{0, \dots, N-1\}$. Hence the optimal value $v_i(\hat{z}_i^*(\cdot); \ell)$, where $\hat{z}_i^*(\cdot)$ denotes the unique minimizer of the local minimization problem, depends continuously on the parameters $\bar{\zeta} - \Pi^\ell(j) + \hat{z}_i^\ell(j)/\mathcal{I}$, $j \in \{0, \dots, N-1\}$. Since we will use this result in the following we will state it in a Lemma.

Lemma 1.1: The optimal value $v_i(\hat{z}_i^*(\cdot); \ell)$ of the local minimization problem (15) of RES $i \in \{1, \dots, \mathcal{I}\}$ is continuous with respect to the parameters $\bar{\zeta} - \Pi^\ell(j) + \hat{z}_i^\ell(j)/\mathcal{I}$, $j \in \{0, \dots, N-1\}$.

Before we can prove the convergence of the sequence $(V^\ell)_{\ell \in \mathbb{N}_0}$ we show the weaker result of monotonicity.

Lemma 1.2: The sequence $(V^\ell)_{\ell \in \mathbb{N}_0}$ generated by Algorithm 1 is monotonically decreasing, i.e., $V^{\ell+1} \leq V^\ell$ holds for all $\ell \in \mathbb{N}$. If, additionally, $\hat{z}^{*\ell}(\cdot) \neq \hat{z}^\ell(\cdot)$, then $V^{\ell+1} < V^\ell$ holds. Hence, the sequence $(V^\ell)_{\ell \in \mathbb{N}_0}$ is strictly monotonically decreasing until Algorithm 1 stops.

Proof: Since $\theta \in [0, 1]$ is chosen such that $F(\theta)$ attains its minimum, see Remark 4.1, replacing θ by \mathcal{I}^{-1} yields a larger value

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

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

$$f\left(\sum_{m=1}^M \alpha_m x_m\right) \leq \sum_{m=1}^M \alpha_m f(x_m), \quad \sum_{m=1}^M \alpha_m = 1, \quad \alpha_m \geq 0$$

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

The proof of Lemma 1.2 shows that $1/\mathcal{I}$ is a possible, fixed, choice for θ in Algorithm 1. Hence, the convergence also holds if the (optimal) step size in Algorithm 1 is replaced by the step size $1/\mathcal{I}$.

Corollary 1.3: For $\ell \rightarrow \infty$ the sequence $(V^\ell)_{\ell \in \mathbb{N}_0} \subset \mathbb{R}$ of Algorithm 1 converges, i.e., $\lim_{\ell \rightarrow \infty} V^\ell = V^* \in \mathbb{R}$.

Proof: Since $V^\ell \geq 0$ and $(V^\ell)_{\ell \in \mathbb{N}_0}$ is monotonically decreasing by Lemma 1.3, $(V^\ell)_{\ell \in \mathbb{N}_0}$ converges to its infimum V^* . ■

In Lemma 1.2 and Corollary 1.3 we have shown that the sequence $(V^\ell)_{\ell \in \mathbb{N}}$ is converging. What is left to show, is the convergence against the value corresponding to the minimization problem (13) which will be done next.

Theorem 1.4: The limit V^* 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 (13).

Proof: Let $z^\sharp(\cdot)$ denote the solution of Problem (13). Since the cost function is continuous and defined on a compact set, there exists an (admissible) accumulation point $z^*(\cdot)$ of the sequence $(\hat{z}^\ell(\cdot))_{\ell=0}^\infty$ satisfying the equality

$$\sum_{j=0}^{N-1} \left(\bar{\zeta} - \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} z_i^*(j) \right)^2 = V^*.$$

We first assume that the limit $\hat{z}_i^*(\cdot)$ is obtained in finitely many iterations, i.e., there exists a $j^* \in \mathbb{N}$ such that $\hat{z}_i^{j^*}(\cdot) = \hat{z}_i^*(\cdot)$. We define the function $F^* : [0, 1]^\mathcal{I} \rightarrow \mathbb{R}$ as

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

To show the assertion, we assume

$$F^*(\mathbf{1}_\mathcal{I}) = V^\sharp < V^* = F^*(0_\mathcal{I}). \tag{16}$$

Since $F^*(\cdot)$ is convex, its directional derivative in $0_\mathcal{I}$ with respect to $\theta = \mathbf{1}_\mathcal{I}$ is less than zero, i.e.,

$$0 > \langle \text{grad } F^*(0_\mathcal{I}), \mathbf{1}_\mathcal{I} \rangle = \sum_{i=1}^{\mathcal{I}} \frac{\partial F^*}{\partial \theta_i}(0_\mathcal{I}). \tag{17}$$

Inequality (16) implies the existence of an index $i \in \{1, \dots, \mathcal{I}\}$ such that $z_i^\sharp(\cdot) \neq z_i^*(\cdot)$ and, thus, $0 > \frac{\partial F^*}{\partial \theta_i}(0_\mathcal{I})$ holds. However, then the i -th RES updates $\hat{z}(\cdot)$, cf. (14) — a contradiction to the assumption that V^* is the limit of $(V^\ell)_{\ell \in \mathbb{N}_0}$ according to Lemma 1.2 since the update $\hat{z}^{j^*+1}(\cdot)$ leads to a better value $V^{j^*+1} < V^*$. If the accumulation point $\hat{z}^*(\cdot)$ is not reached in finitely many steps then there exists a subsequence $(j_k)_{k \in \mathbb{N}}$ such that $\lim_{k \rightarrow \infty} \hat{z}^{j_k}(\cdot) = \hat{z}^*(\cdot)$. Then, due to the continuity of the optimal value function (c.f. Lemma 1.1) there exists a $k^* \in \mathbb{N}$ such that

$V^{j_{k^*}+1} < V^*$ which again contradicts the properties of V^* . ■

REFERENCES

- [1] HSL Mathematical Software Library. A collection of Fortran codes for large-scale scientific computation, 2004. <http://hsl.rl.ac.uk/>.
- [2] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, Belmont, MA, USA, 1989.
- [3] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley. Notes on decomposition methods. Technical report, Stanford University, 2007.
- [4] 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.
- [5] 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.
- [6] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control. Theory and Algorithms*. Springer London, 2011.
- [7] D. Jakovetić, J. Xavier, and J. M. F. Moura. Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146, 2014.
- [8] A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.
- [9] A. Nedić, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.
- [10] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.
- [11] J. N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, MIT, Cambridge, MA, USA, 1984.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] M. Zhu and S. Martínez. On distributed convex optimization under inequality and equality constraints. *IEEE Transactions on Automatic Control*, 57(1):151–164, 2012.