

Distributed Model Predictive Control for Smart Energy Systems

Rasmus Halvgaard, Lieven Vandenberghe, Niels Kjølstad Poulsen,
Henrik Madsen, and John Bagterp Jørgensen

Abstract—Integration of a large number of flexible consumers in a smart grid requires a scalable power balancing strategy. We formulate the control problem as an optimization problem to be solved repeatedly by the aggregator in a model predictive control framework. To solve the large-scale control problem in real-time requires decomposition methods. We propose a decomposition method based on Douglas–Rachford splitting to solve this large-scale control problem. The method decomposes the problem into smaller subproblems that can be solved in parallel, e.g., locally by each unit connected to an aggregator. The total power consumption is controlled through a negotiation procedure between all cooperating units and an aggregator that coordinates the overall objective. For large-scale systems, this method is faster than solving the original problem and can be distributed to include an arbitrary number of units. We show how different aggregator objectives are implemented and provide simulations of the controller including the computational performance.

Index Terms—Smart grid, model predictive control, Douglas–Rachford splitting.

I. INTRODUCTION

A LARGE number of units with flexible power consumption are expected to be part of the future power system [1]. In Denmark, examples of these units are electrical heat pumps for heating in buildings and Electric Vehicles (EVs) with batteries that can be charged and discharged. If pooled together in a large-scale aggregated system these smaller consumption units could potentially offer flexibility to the power system. If the units are controlled and coordinated well, they can help to partially balance the fluctuating power production caused by renewable energy sources such as wind and solar. In real-time electricity markets the aggregated units can help to minimize the imbalances caused by forecast errors and in general provide ancillary services. Controlling a large number of units in real-time requires fast evaluation

of the control algorithm that coordinates the power consumption. Thus methods for solving this large-scale optimization problem in real-time must be developed.

In this paper, we consider the problem of real-time large-scale power balancing. We apply Douglas–Rachford splitting [2] to decompose the general problem into smaller dynamically decoupled subproblems. The subproblems can be distributed and solved in parallel either by a large central computer or locally by each unit. The latter case requires a faster and very reliable two-way communication between the controller and the units since the subproblems must be solved many times at each time step. Each unit has its own model, constraints, and variables, and can even make decisions based on its own local control objective. All units must communicate their predicted consumption plan to an aggregator that coordinates the need for system level flexibility and minimizes imbalances. The aggregator continuously communicates control signals and synchronizes the global negotiation. This negotiation procedure is required to converge in every time step and thus requires fast evaluation of the unit subproblems that can be cast as convex optimal control problems. Convergence is achieved by coordinating all units' consumption through a negotiation procedure that is updated in the coordinating system level referred to as an aggregator.

With a least-squares tracking objective, the aggregator is able to control and deliver the requested combined power consumption of all units in real-time and can provide running forecasts of the consumption by applying a receding horizon control principle. This principle, referred to as Model Predictive Control (MPC), repeatedly solves the control problem online for the predicted development of the system. The first part of the obtained control action is implemented. At the next sample time, the procedure is repeated by using new measurements and by moving the prediction window one step. Any model and forecast uncertainty will impact the control performance. The control algorithm presented in this paper is based on the separation principle and certainty equivalence [3], [4], i.e., all future uncertain values are replaced by their mean values. In this way we avoid the curse-of-dimensionality associated with Dynamic Programming that otherwise optimally includes forecast uncertainty.

The sampling time in real-time markets could be lower than five minutes [5]. The precise sampling time is dictated by the settlement requirements of the regulating power market.

Manuscript received November 15, 2014; revised April 22, 2015, August 3, 2015, and December 1, 2015; accepted January 25, 2016. Date of publication February 24, 2016; date of current version April 19, 2016. This work was supported by the Danish Council for Strategic Research through the CITIES Project [23] under Grant 1305-00027. Paper no. TSG-01129-2014.

R. Halvgaard, N. K. Poulsen, H. Madsen, and J. B. Jørgensen are with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby 2800, Denmark (e-mail: rhal@dtu.dk; nkpo@dtu.dk; hmad@dtu.dk; jbj@dtu.dk).

L. Vandenberghe is with the University of California at Los Angeles, Los Angeles, CA 90095-1594 USA (e-mail: vandenbe@ucla.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSG.2016.2526077

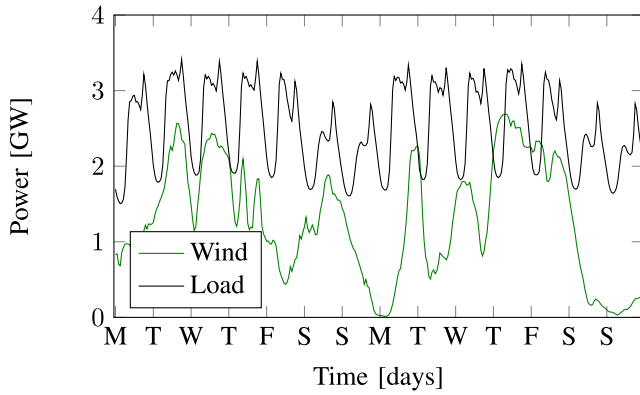


Fig. 1. Wind power and consumption in West Denmark January 2012.

This short sampling time motivates computational efficient optimization algorithms for the MPC that balances the power. For large-scale systems such as power balancing problems, decomposition methods are one computationally attractive option. The computation time for MPC problems grow polynomially in standard solvers as the number of units increase. By decomposing the problem and solving smaller subproblems in parallel, we can handle a much larger amount of units. The fluctuating wind power could be the power reference to be tracked by the aggregator, see Fig. 1. However, in practice the consumption plan comes from the power market. We use the Nordic power market framework, where the goal of the power balancing aggregator would be to minimize the deviation from a consumption plan already negotiated in advance at the day-ahead market. Any deviations from this day-ahead plan cause power imbalances that must be settled in the regulating power market. These imbalances can be minimized by the aggregator by tracking the day-ahead plan as accurately as possible, while trying to eliminate forecast errors and unforeseen disturbances along the way. If the regulating power prices are predicted, the aggregator objective can reflect the actual imbalance costs. Many studies do not consider the imbalances they introduce when only optimizing over the day-ahead market prices [6], [7]. We therefore provide an example of taking regulating power prices into account. Grid capacity constraints on the total active power can also be applied as an aggregator constraint.

Compared to dual decomposition that uses the subgradient projection method with rather slow convergence [8], the Douglas-Rachford splitting used in this paper is often faster. Another advantage of the presented method is that it easily evaluates complicated and even non-linear expressions. Compared to a column generation method like Dantzig-Wolfe decomposition [9], which only handles linear programming problems, this method easily handles non-linear convex functions, and converges under very mild conditions. In our case, the subproblems even reduce to quadratic optimal control problems that can be solved very efficiently using the Riccati recursions [3], [10]. This would not be the case for a similar ADMM formulation [11], where the subproblems are more complicated. Reference [12] use ADMM for Electric Vehicle charging coordination. Reference [13] describe a

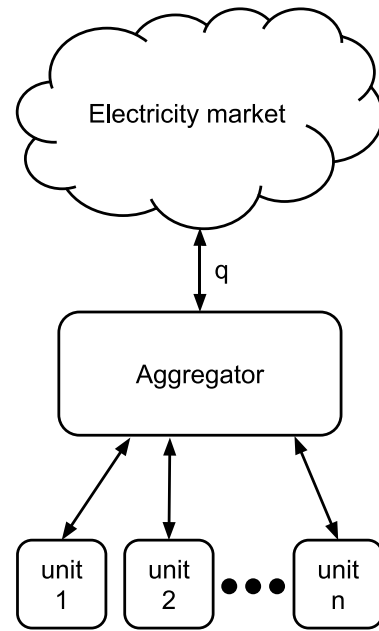


Fig. 2. Aggregator role and portfolio of units. The aggregator gets a consumption plan q to follow from the market.

general approach to Distributed MPC problems, while [14] provides convergence results for the splitting algorithm.

In this paper, we illustrate the advantages of the Douglas-Rachford splitting method and show how an aggregator can use the method for power balancing flexible consumption units, exemplified by thermal storage units, e.g., heat pumps in buildings. The method can be completely decentralized down to communication with neighbors only [15].

This paper is organized as follows. In Section II we formulate the general large-scale optimization problem for an aggregator with a large number of units in its portfolio. In Section III we introduce operator splitting and its convex optimization terminology. This leads to the Douglas Splitting algorithm in Section III-A. Section IV describes how the Douglas-Rachford splitting algorithm can be applied to the aggregator problem just described. Different aggregator objectives are investigated in Section V. Finally in Section VI the method is demonstrated through simulations and its convergence and scalability is discussed. Section VII provides conclusions.

II. THE AGGREGATOR PROBLEM

We wish to control the power consumption of a large number of flexible and controllable units. The motivation for controlling the units is to continuously adapt their consumption to the changing stochastic power production from wind and solar. In the future, this power balancing might be done by solving large-scale control problems. Instead of tracking the wind power directly, it is currently much more realistic to interface and bid into electricity markets. We assume that an *aggregator* controls a large number of flexible consumption units as shown in Fig. 2. Based on predictions of the

aggregated unit behavior, the aggregator bids into the day-ahead power market and buys a certain amount of energy for the coming day. The plan could be a result of solving a unit commitment problem [16], where stochastics and integer variables are taken into account. The resulting consumption plan must be followed to avoid imbalances and in turn economic penalties. So a real-time controller must regulate the power to minimize any imbalances.

The power consumption profile is a vector denoted $q \in \mathbb{R}^N$ and denotes the amount of power to be consumed at each time step k for the entire prediction horizon $k = 1, \dots, N$. This profile must be followed by the aggregator, such that the combined power consumption from all units sum to this at every time instant. The centralized large-scale problem that includes all units and their variables (p, x_j, y_j, u_j) and constraints is

$$\underset{p, u_j}{\text{minimize}} \quad cg(p) + \sum_{j=1}^n \phi_j(u_j) \quad (1a)$$

$$\text{subject to} \quad p = \sum_{j=1}^n u_j \quad (1b)$$

$$x_j^+ = A_j x_j + B_j u_j + E_j d_j \quad (1c)$$

$$y_j = C_j x_j \quad (1d)$$

$$y_j^{\min} \leq y_j \leq y_j^{\max} \quad (1e)$$

$$\Delta u_j^{\min} \leq \Delta u_j \leq \Delta u_j^{\max} \quad (1f)$$

$$u_j^{\min} \leq u_j \leq u_j^{\max} \quad (1g)$$

We model the units with linear discrete-time state-space systems and define \mathcal{U}_j as a closed convex set containing the model and constraints of the j th unit (1c)-(1g). $x_j \in \mathbb{R}^N$ is the time varying state vector in the discrete-time state-space system defined by the matrices (A_j, B_j, E_j, C_j) . y_j is the output, of a linear system with control input u_j . d_j is the modeled and predictable disturbances (e.g., outdoor temperature). The time-varying input and output constraints are superscripted with max and min, and account for the available flexibility for each unit. For thermal storage units, these are the power available and the accepted temperature interval, respectively. $\Delta u_j(k) = u_j(k) - u_j(k-1)$ is the rate of input change, where k is the time step in the time varying input vector $u_j \in \mathbb{R}^N$. The total consumption $p \in \mathbb{R}^N$ is a sum of the predicted consumption profiles u_j for each of the $j = 1, 2, \dots, n$ units. p can also be constrained to reflect capacity constraints in the power grid. g is the aggregator objective function. We investigate various choices of aggregator objectives in Section V but simulate using a traditional least squares LQR objective. This objective contains local control objectives and costs associated with operating the unit. Examples of local objective functions are

$$\phi_j^{eco}(u_j) = c_j u_j \quad (2a)$$

$$\phi_j^{soft}(v_j) = \gamma_j v_j \quad (2b)$$

$$\phi_j^{reg}(u_j) = \|\Delta u_j\| \quad (2c)$$

$$\phi_j^{ref,u}(u_j) = \left\| u_j - u_j^{ref} \right\| \quad (2d)$$

$$\phi_j^{ref,y}(y_j) = \left\| y_j - y_j^{ref} \right\| \quad (2e)$$

(2a): cost c of using power consumption (could be actual economic maintenance or fuel costs). (2b): output slack variable v_j with penalty γ_j (to prevent infeasible problems when subject to noise). (2c): regularization on the rate of input change. (2e): output (temperature) reference tracking. (2d): input reference (day-head plan). v_j is a slack variable on the output, such that infeasible problems are avoided when noise drives the system slightly outside the constraints. The slack variable should be heavily penalized in the local objective function ϕ_j when added. We leave it out for a lighter notation in the remaining part of the paper. The functions ϕ_j and g may be indicator functions that represent constraints on the variables u_j or their sum. We exploit this later when decomposing the problem.

A. Decomposing the Aggregator Problem

With the use of indicator functions (4) the original problem (1) can be written as

$$\underset{u_j}{\text{minimize}} \quad \sum_{j=1}^n f_j(u_j) + g\left(\sum_{j=1}^n u_j\right). \quad (3)$$

Here the constraints in \mathcal{U}_j are hidden in an indicator function f_j

$$f_j(u_j) = \begin{cases} \phi_j(u_j) & \text{if } u_j \in \mathcal{U}_j \\ +\infty & \text{otherwise.} \end{cases} \quad (4)$$

In decomposition literature [11] a common and general formulation of the problem is in the form

$$\underset{u, p}{\text{minimize}} \quad g(p) + f(u) \\ \text{subject to} \quad p = Su. \quad (5)$$

The functions $f_j : \mathbb{R}^N \rightarrow \mathbb{R}$ and $g : \mathbb{R}^N \rightarrow \mathbb{R}$ are assumed to be closed and convex. We define $u = [u_1^T, u_2^T, \dots, u_n^T]^T$ as a stacked vector of individual unit consumption profiles and $f(u)$ as a sum of the unit indicator functions from (4)

$$f(u) = \sum_{j=1}^n f_j(u_j).$$

$S \in \mathbb{R}^{N \times Nn}$ simply sums all contributions to the total power consumption

$$S = [I \ I \ \dots \ I], \quad (6)$$

where I is the identity matrix. S remains constant even if we also add power production units by defining production as negative consumption. We limit S to this simple structure that speeds up computation time in Section III-A. This optimization problem must be solved at every time instant in a Receding Horizon manner [17], [18]. We apply Douglas-Rachford splitting to solve this large-scale control problem in real-time. This splitting method is explained in the following section.

III. OPERATOR SPLITTING

We show how operator splitting can be done for a problem in the form

$$\underset{u}{\text{minimize}} \quad f(u) + g(Su). \quad (7)$$

This is equivalent to the control problem (1) in the form (5). The objective (7) can be split in separate functions with variable $u \in \mathbb{R}^N$ and where $f : \mathbb{R}^N \rightarrow \mathbb{R}$ and $g : \mathbb{R}^N \rightarrow \mathbb{R}$ are closed convex functions with nonempty domains. It captures a large class of constrained and unconstrained optimization problems. For example, if we let g be the indicator function of a convex set \mathcal{U}

$$g(Su) = \begin{cases} 0 & \text{if } Su \in \mathcal{U} \\ +\infty & \text{otherwise,} \end{cases} \quad (8)$$

the problem (7) is equivalent to the following constrained optimization problem

$$\begin{aligned} & \underset{u}{\text{minimize}} && f(u) \\ & \text{subject to} && Su \in \mathcal{U}. \end{aligned}$$

In order to solve the problem we derive the optimality conditions associated with (7). The first-order optimality condition are expressed in inclusion form as

$$0 \in \partial f(u) + S^T \partial g(Su), \quad (9)$$

where $\partial f(u)$ denotes the subdifferential of f at u . The dual optimality condition of (7) is found by first introducing an extra variable p and a consensus constraint, such that

$$\begin{aligned} & \underset{u,p}{\text{minimize}} && f(u) + g(p) \\ & \text{subject to} && Su = p. \end{aligned} \quad (10)$$

The Lagrangian of (10) is

$$L = f(u) + g(p) + z^T(Su - p)$$

where z is the dual variable associated with the equality constraint. The dual function is

$$\inf_{u,p} L = \inf_u (z^T Su + f(u)) + \inf_p (-z^T p + g(p))$$

Two conjugate functions are readily identified in this expression. Since the conjugate function is defined as [19]

$$f^*(p) = \sup_u (p^T u - f(u))$$

the dual of (7) reduces to

$$\underset{z}{\text{maximize}} \quad -f^*(-S^T z) - g^*(z), \quad (11)$$

where f^* and g^* denote the conjugate functions of f and g . The first-order dual optimality condition associated with (11) can now be expressed as

$$0 \in -S \partial f^*(-S^T z) + \partial g^*(z). \quad (12)$$

The subdifferential of a closed convex function f satisfies $(\partial f)^{-1} = \partial f^*$ [20], i.e.,

$$y \in \partial f(x) \Leftrightarrow \partial f^*(y) \in x$$

Hence if we let $u = \partial f^*(-S^T z)$ in (12) then

$$0 \in \begin{bmatrix} 0 & S^T \\ -S & 0 \end{bmatrix} \begin{bmatrix} u \\ z \end{bmatrix} + \begin{bmatrix} \partial f(u) \\ \partial g^*(z) \end{bmatrix}. \quad (13)$$

This is the primal dual optimality conditions that can be solved to find a solution of (7). Because we deal with subgradients

we use the inclusion sign instead of equality in (13). It simply means that zero must be included in the set. The subdifferentials are monotone operators and (13) is a monotone inclusion problem. The solution to this problem has zero included in the sum of the two maximal monotone operators A and B

$$0 \in A(u, z) + B(u, z). \quad (14)$$

A. Douglas-Rachford Splitting

We apply Douglas-Rachford splitting [2] to solve the problem and identify two splitting operators in (13) as

$$A(u, z) = \begin{bmatrix} 0 & S^T \\ -S & 0 \end{bmatrix}, \begin{bmatrix} u \\ z \end{bmatrix} \quad (15)$$

$$B(u, z) = \begin{bmatrix} \partial f(u) \\ \partial g^*(z) \end{bmatrix}. \quad (16)$$

The primal-dual Douglas-Rachford splitting algorithm is a special case of the proximal point algorithm [21] and works by starting at any z and repeating the following iterations:

$$x^+ = (I + tB)^{-1}(v), \quad (17a)$$

$$y^+ = (I + tA)^{-1}(2x^+ - v), \quad (17b)$$

$$v^+ = v + \rho(y^+ - x^+). \quad (17c)$$

The superscript $+$ indicates the next iterate, e.g., $x^+ = x_{i+1}$ if i is the iteration number. This general algorithm fits our problem (15) when $x = [u^T z^T]^T$. The algorithm steps are: 1) optimize over x 2) keep x fixed and optimize over y 3) integrate the error with gain $\rho \in]0; 2]$. Thus the algorithm requires resolvents of A and B , but not their sum. The inverse function of the operators A and B is called a resolvent. The resolvent of an operator F is the operator $(I + tF)^{-1}$ with scaling $t > 0$. In our case we require the resolvents of the functions ∂f and ∂g^* from (13). The resolvent of ∂f is the proximal mapping

$$(I + t\partial f)^{-1}(x) = \text{prox}_{t\partial f}(x). \quad (18)$$

This proximal mapping is defined as the prox-operator of f , if f is convex

$$\text{prox}_{t\partial f}(x) = \underset{u}{\text{argmin}} \left(f(u) + \frac{1}{2t} \|u - x\|^2 \right).$$

If f is an indicator function similar to (8) the prox-operator is equivalent to solving the problem

$$\begin{aligned} & \underset{u}{\text{minimize}} && \frac{1}{2t} \|u - x\|^2 \\ & \text{subject to} && u \in \mathcal{U}. \end{aligned}$$

This can also be simplified to a projection on the convex set \mathcal{U} .

The prox-operator of the conjugate g^* is related to the prox-operator of g via

$$\text{prox}_{t\partial g^*}(s) = s - t \text{prox}_{g/t}(s/t). \quad (19)$$

The algorithm can now be compactly written with prox-operators.

IV. DOUGLAS-RACHFORD SPLITTING ALGORITHM

When the specific operators (15) from our problem are inserted into the general Douglas-Rachford splitting algorithm (17) we get

$$u^+ = \text{prox}_{f^*}(v) \quad (20a)$$

$$z^+ = \text{prox}_{t g^*}(s) \quad (20b)$$

$$\begin{bmatrix} w^+ \\ m^+ \end{bmatrix} = \begin{bmatrix} I & tS^T \\ -tS & I \end{bmatrix}^{-1} \begin{bmatrix} 2u^+ - v \\ 2z^+ - s \end{bmatrix} \quad (20c)$$

$$v^+ = v + \rho(w^+ - u^+) \quad (20d)$$

$$s^+ = s + \rho(m^+ - z^+). \quad (20e)$$

The A -operator from (15) is linear, so the resolvent in (20c) is the matrix inverse.

In our case the algorithm has the following interpretation:

- 1) The aggregator sends suggested consumption profiles v_j to the units
- 2) The units evaluate their subproblems, i.e., the prox-operator in (20a), by solving a QP with their local unit model, costs, constraints, and variables
- 3) The units respond in parallel with their updated consumption profiles u_j^+
- 4) The remaining steps (20b)-(20e) simply updates the other variables and are computed by the aggregator alone

It is important to note that the prox-operators do not have to be evaluated by the units. If the units upload their objective and constraints to the aggregator a large parallel computer could solve all the subproblems. This significantly reduces convergence speed and communication requirements in a practical system.

A. Step (20a)

The prox-operator evaluation in the first step (20a) is defined as

$$\text{prox}_{f^*}(v) = \underset{\tilde{v}}{\text{argmin}} \left(f(\tilde{v}) + \frac{1}{2t} \|\tilde{v} - v\|_2^2 \right).$$

In our case we must solve this subproblem for all units and stack their solutions in u^+ . Since all units are decoupled, separability of $f(u) = \sum_j f_j(u_j)$ implies that

$$\text{prox}_{f^*}(v) = (\text{prox}_{f_1^*}(v_1), \dots, \text{prox}_{f_n^*}(v_n)).$$

Here $v = [v_1^T, v_2^T, \dots, v_n^T]^T$ and each of these prox-operators involve only one unit. We thus have the j th unit subproblem

$$\text{prox}_{f_j^*}(v_j) = \underset{u_j}{\text{argmin}} \left(f_j(u_j) + \frac{1}{2t} \|u_j - v_j\|_2^2 \right).$$

with the standard QP formulation

$$\begin{aligned} & \underset{u_j}{\text{minimize}} && \frac{1}{2t} u_j^T u_j - \frac{1}{t} v_j^T u_j + \phi_j(u_j) \\ & \text{subject to} && u_j \in \mathcal{U}_j. \end{aligned} \quad (21)$$

Note that ϕ_j contains slack variables and regularization (2c). v_j can be interpreted as a tracking reference trajectory or an individual linear coefficient for each unit. In our case, all of these quadratic subproblems reduce to finite horizon constrained LQR problems that can be solved efficiently by methods based on the Riccati recursion [3], [10].

B. Step (20c)

The (w, m) -update in (20c) gathers the unit consumption profiles and involves only multiplications with S and S^T . Due to the simple structure of S , defined in (6), we can simplify this update using the matrix inversion lemma to

$$\begin{bmatrix} I & tS^T \\ -tS & I \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{\tilde{n}} \begin{bmatrix} -tS^T \\ I \end{bmatrix} \begin{bmatrix} tS & I \end{bmatrix},$$

where $\tilde{n} = 1 + nt^2$. Due to the simple structure of S this update reduces to

$$w_j = \tilde{v}_j - \frac{1}{\tilde{n}} \left(t\tilde{s} + t^2 \sum_{j=1}^n \tilde{v}_j \right)$$

$$w^+ = [w_1^T, w_2^T, \dots, w_n^T]^T$$

$$m^+ = \frac{1}{\tilde{n}} \left(\tilde{s} + t \sum_{j=1}^n \tilde{v}_j \right),$$

where $\tilde{v}_j = 2u_j^+ - v_j$ and $\tilde{s} = 2z^+ - s$. For a large-scale system with $nt^2 \gg 1$ we get expressions involving the mean consumption plan $\hat{v} = \frac{1}{n} \sum_{j=1}^n \tilde{v}_j$

$$w_j \simeq \tilde{v}_j - \hat{v} \quad m^+ \simeq \frac{1}{t} \hat{v}$$

w_j is the difference between the unit consumption plan and the average \hat{v} .

C. Convergence

The primal and dual optimality conditions from (13) provide a measure of convergence, i.e., $\|r_p\| \leq \epsilon_p$ and $\|r_d\| \leq \epsilon_d$ with

$$r_p = \frac{v - u^+}{t} + S^T z^+ \quad r_d = \frac{s - z^+}{t} - S u^+. \quad (22)$$

These expressions are obtained from the prox-operators (20a) and (20b) that defines the subgradients $\partial f(u)$ and $\partial g^*(z)$ in (13), respectively. The algorithm is terminated when the norm of these residuals are below some user-defined tolerance. See [14] for more general convergence results.

From theory, it is known that the step size t in the algorithm must remain constant. However, various heuristics provide adaptive strategies, see for instance the references in [11]. In the numerical example provided in this paper, t was found experimentally, based on the observed convergence behavior. Also the scaling gain $\rho \in]0; 2]$ must be selected and usually $\rho = 1.5$ is a good choice.

V. AGGREGATOR OBJECTIVES

In this paper we consider three different aggregator objectives sketched in Fig. 3. All objectives track the given power consumption profile q by minimizing $e = p - q$. First, the traditional tracking MPC with quadratic penalty on the residual e . Second, we also set up a linear economic MPC that includes the actual costs for imbalances. Third, the same economic objective but with asymmetric costs. All objectives are easily evaluated by the prox-operator (20b) that yields simple analytic expressions in all three cases.

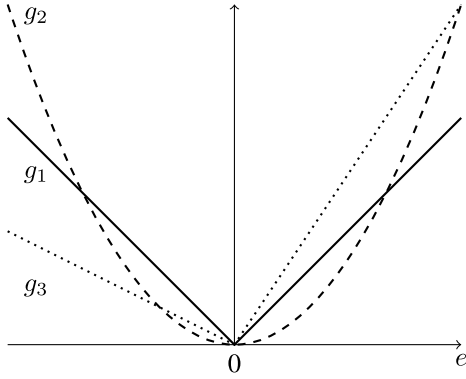


Fig. 3. Aggregator objective functions: g_1 (26), g_2 (23), and g_3 (27). $e = p - q$ is the power imbalance.

A. Tracking Objective

If q represents a desired total power consumption profile to be followed by the aggregated units, we can choose the differentiable quadratic aggregator objective

$$g_2(p) = \frac{1}{2} \|p - q\|_2^2, \quad (23)$$

that is easily evaluated by the prox-operator required in the Douglas-Rachford z^+ update (20b). The conjugate of $g(p)$ is

$$g^*(y) = \sup_p (y^T p - g(p)) = \frac{1}{2} \|y\|_2^2 + q^T y, \quad (24)$$

and the prox-operator simplifies to the analytic expression

$$\text{prox}_{t g^*}(s) = \frac{1}{t+1} s - \frac{t}{t+1} q, \quad (25)$$

to be substituted for (20b).

B. Economic Objectives

When the aggregated power consumption p deviates from the reference plan q , this power imbalance, denoted $e = p - q$, has a penalty cost c equal to the regulating power price

$$g_1(e) = c \|e\|_1. \quad (26)$$

However, the price might depend on the sign of the imbalance and the overall system imbalance. These regulating power prices are difficult to forecast, but in the Nordic power market they have a lower or upper bound depending on imbalance direction, i.e., the sign of e . When the up and down-regulating power market prices, c^+ and c^- respectively, are forecasted individually, we get the following piecewise linear objective function to be minimized

$$g_3(e) = \sum_{k=0}^N \max(-c_k^+ e_k, c_k^- e_k). \quad (27)$$

This aggregator objective is non-differentiable, but is easily handled by the prox-operator required in the algorithm (20). In this case the conjugate function $g_3^*(e)$ is more difficult to find analytically, and we exploit the relation (19).

The objective (27) is both separable in time and units, so the prox-operator of (27) becomes

$$\text{prox}_{g/t}(s/t) = \underset{\bar{s}_k}{\text{argmin}} \sum_{k=0}^N \left(g_k(\bar{s}_k) + \frac{t}{2} \|\bar{s}_k - s_k/t\|_2^2 \right).$$

The prox-operator evaluation can be divided into several cases for the unconstrained minimum to be found, since it is a sum of a piecewise linear function $g(e)$ and a quadratic function. Analytically, this leaves the minimum to be found in the following three cases

$$[\text{prox}_{g/t}(s_k/t)]_k = \begin{cases} (s_k - c_k^-)/t & \text{if } s_k \leq tq_k - c_k^- \\ (s_k + c_k^+)/t & \text{if } s_k \geq tq_k + c_k^+ \\ q_k & \text{otherwise} \end{cases}$$

When inserted into (19) and using the asymmetric economic objective (27) the final analytical expression for (20b) is

$$[\text{prox}_{t g^*}(s_k)]_k = \begin{cases} c_k^- & \text{if } s_k \leq tq_k - c_k^- \\ -c_k^+ & \text{if } s_k \geq tq_k + c_k^+ \\ s_k - tq_k & \text{otherwise} \end{cases} \quad (28)$$

The case with a symmetric price (26) is easily included here when $c^- = c^+$.

C. Grid Constraints

A power capacity constraint on p can also be added to (1). When $g(p) = 0$

$$\begin{aligned} & \text{minimize} \quad \frac{t}{2} \|p - s/t\|_2^2 \\ & \text{subject to} \quad p \geq h \end{aligned} \quad (29)$$

the prox-operator reduces to a simple projection

$$\text{prox}_{t g^*}(s) = (0, s - th)_- = \min(0, s - th) \quad (30)$$

where $h = -p_{\max}$. This upper capacity limit is sufficient p_{\max} as the consumption units already have a zero lower bound on their consumption (1g), i.e., negative consumption is not allowed as this is production.

VI. NUMERICAL EXAMPLE

We model a portfolio of different thermal storage systems as second order systems in the form

$$G_j(s) = \frac{y_j}{u_j} = \frac{K}{(\tau_j^a s + 1)(\tau_j^b s + 1)}.$$

u_j is the consumption and y_j is the output temperature. One time constant is usually much bigger than the other. Realistic values for the dominating time constant in buildings with heat pumps or refrigeration systems is $\tau^a \in \{10; 120\}$ h [22]. In our simulations we also set $\tau^b = \tau^a/5$ and pick τ^a randomly. We use the same dynamics for the disturbances d_j , e.g., ambient temperature. The transfer functions are easily converted to the state space form (1c). The constraints were selected equal for both units: $(y^{\min}, y^{\max}) = (15, 25)$ °C, $(u^{\min}, u^{\max}) = (0, 50)$ W, $(\Delta u^{\min}, \Delta u^{\max}) = (-50, 50)$ W, and output slack variable penalty $\gamma = 10^4$.

We scale the gain K with $1/n$ such that the maximum possible power consumption automatically adds to $p_{\max} = 1$.

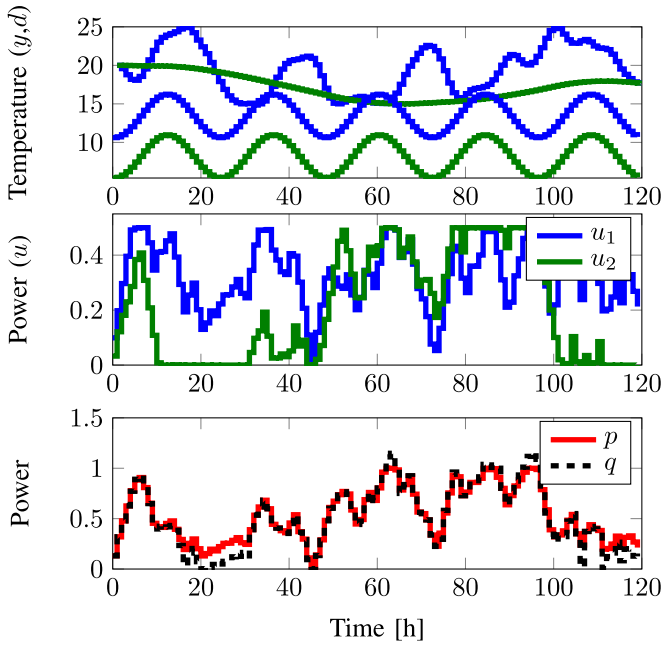


Fig. 4. Closed loop simulation of power balancing with $n = 2$ thermal storage units.

The reference q is also scaled to always lie between 0 and 1. Better numerical performance and sensitivity to the step size t is obtained in this way.

A. $n = 2$

The simulation results with $t = 0.5$ using 25 iterations for every open loop problem is shown in Fig. 4. Full state information was used to produce these closed-loop profiles. The prediction horizon was $N = 24$ and we used the quadratic objective function (23). The first plot shows the output temperatures of the two units in blue and green respectively. The two sinusoids in the same plot are the ambient temperature disturbances. The second plot shows the input power consumption. The bottom plot is the resulting total power p and tracking profile q . All powers are scaled, such that the total maximum power of p is equal to 1. Consequently for $n = 2$ units their maximum power is $1/n = 0.5$. Clearly, both the consumption and the temperatures are kept within their operating intervals. Also the combined consumption p matches the reference q very well. The plot illustrates two cases where it is not always possible to follow the plan q . Obviously in periods where q is larger than the maximum total power. And in periods where q is close to zero and the outputs are near the constraints, e.g., around 20 h. It is simply not possible to follow the plan without violating the output constraints. However, around 45 h there is enough capacity to turn the units completely off for a short period of time. q was a scaled version of the difference between the wind power and the load from Fig. 1, but could be any profile.

B. $n = 100$

To demonstrate that the algorithm works for a larger number of units, we chose $n = 100$ units with uniform randomly

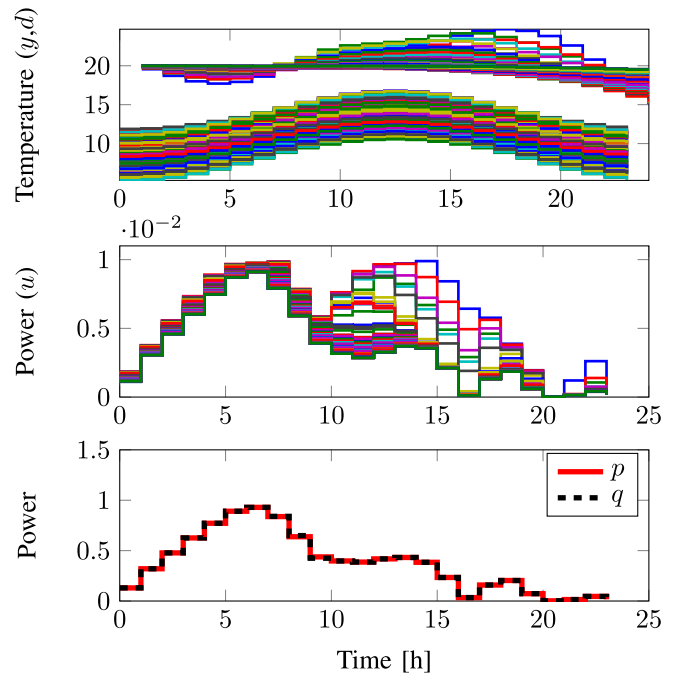


Fig. 5. Open loop simulation of power balancing with $n = 100$ thermal storage units.

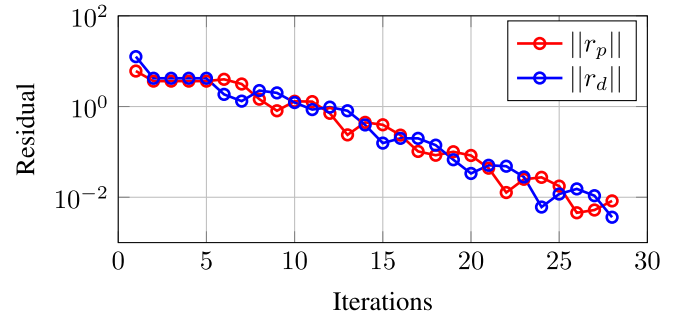


Fig. 6. Convergence for open-loop problem $n = 3$ and $t = 0.8$.

generated parameters as before. In Fig. 5 the same consumption profile q is tracked but we only plotted the first open-loop profile.

C. Convergence

Figure 6 shows how the Douglas-Rachford splitting algorithm converges during the iterations of one open-loop simulation. The optimality conditions (22) both reach a threshold below 10^{-2} after 25 iterations. The step size t was tuned to $t = 0.8$. This was done for different number of units in Fig. 7. For the simulation with $n = 100$ units, the algorithm is seen to converge within 50 iterations. As the number of units increase the computation time also increases. This is shown in Fig. 8. We measured the computation of all the subproblems and took the average (labeled DR parallel) to illustrate the unit scaling behavior. The total computation time for the serial implementation is labeled DR. In general the computation speed is linear for this method. For a large number of units the Douglas-Rachford splitting algorithm is faster than just solving the original problem, even for a serial implementation. We solved all QPs including the large scale problem

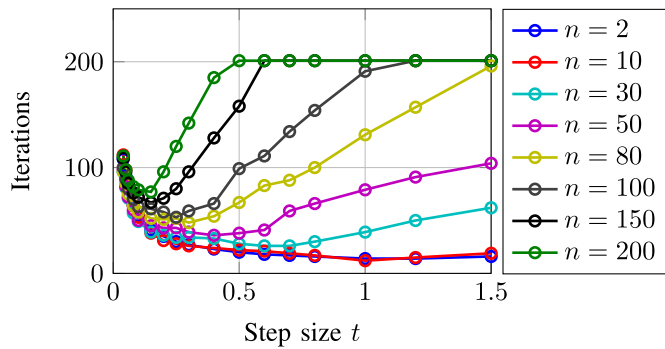


Fig. 7. Tuning of step size t as a function of the number of units n . Maximum number of iterations was limited to 200.

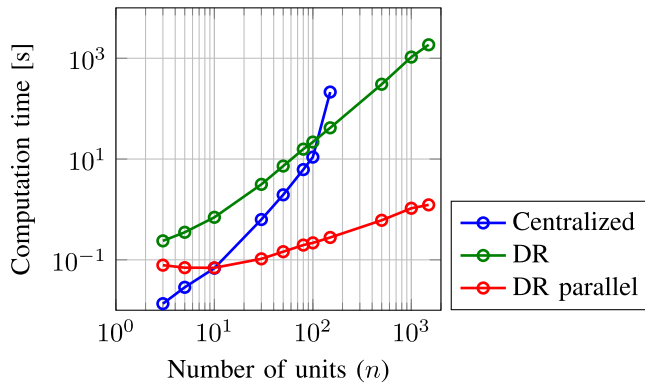


Fig. 8. Convergence for open-loop problem with tuned step sizes t .

using MOSEK in MATLAB on an Intel Core i7 2.67 GHz laptop.

VII. CONCLUSION

We solved the power balancing problem using a constrained model predictive controller with a least squares tracking error criterion. This is an example of a large-scale optimization problem that must be solved reliably and in real-time. We demonstrated how Douglas-Rachford splitting can be applied in solving this problem. By decomposing the original optimization problem thousands of units can be controlled in real-time by computing the problem in a distributed (parallel) manner. We considered a large-scale power balancing problem with flexible thermal storage units. A given power consumption profile can be followed by controlling the total power consumption of all flexible units through a negotiation procedure with the dual variables introduced in the method. An economic aggregator objective that takes the regulating power prices into account was derived. The obtained solution converges towards the original problem solution and requires two-way communication between units and the coordinating level. The resulting power balancing performance runs in closed loop while the local constraints and objectives for each unit are satisfied and aggregator operation costs are reduced. We showed that the decomposition algorithm scales well with the number of units compared to a standard solver, even when solving the subproblems serially.

REFERENCES

- [1] J. Yan, A. J. Conejo, and E. Dahlquist, "Intelligent energy systems," in *Handbook of Clean Energy Systems*, vol. 4. Hoboken, NJ, USA: Wiley, 2015, pp. 2213–2238.
- [2] J. Eckstein and D. P. Bertsekas, "On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, nos. 1–3, pp. 293–318, 1992.
- [3] J. B. Jørgensen, "Moving horizon estimation and control," Ph.D. dissertation, Dept. Chem. Eng., Tech. Univ. Denmark, Lyngby, Denmark, 2005.
- [4] D. P. Bertsekas, *Dynamic Programming and Stochastic Control*. New York, NY, USA: Academic Press, 1976.
- [5] Y. Ding *et al.*, "Real-time market concept architecture for ecogrid EU—A prototype for European smart grids," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 2006–2016, Dec. 2013.
- [6] R. Halvgaard, N. K. Poulsen, H. Madsen, and J. B. Jørgensen, "Economic model predictive control for building climate control in a smart grid," in *Proc. IEEE PES ISGT*, Washington, DC, USA, 2012, pp. 1–6.
- [7] T. G. Hovgaard, L. F. S. Larsen, K. Edlund, and J. B. Jørgensen, "Model predictive control technologies for efficient and flexible power consumption in refrigeration systems," *Energy*, vol. 44, no. 1, pp. 105–116, Aug. 2012.
- [8] G. K. H. Larsen, N. D. van Foreest, and J. M. A. Scherpen, "Distributed control of the power supply-demand balance," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 828–836, Jun. 2013.
- [9] K. Edlund, J. D. Bendtsen, and J. B. Jørgensen, "Hierarchical model-based predictive control of a power plant portfolio," *Control Eng. Pract.*, vol. 19, no. 10, pp. 1126–1136, 2011.
- [10] C. V. Rao, S. J. Wright, and J. B. Rawlings, "Application of interior-point methods to model predictive control," *J. Optim. Theory*, vol. 99, no. 3, pp. 723–757, Dec. 1998.
- [11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
- [12] A. Mercurio, A. Di Giorgio, and F. Purificato, "Optimal fully electric vehicle load balancing with an ADMM algorithm in smartgrids," in *Proc. 21st Mediterr. Conf. Control Autom.*, Chania, Greece, 2013, pp. 119–124.
- [13] T. H. Summers and J. Lygeros, "Distributed model predictive consensus via the alternating direction method of multipliers," in *Proc. 50th Annual Allerton Conf. Commun. Control Comput. (Allerton)*, Monticello, IL, USA, 2012, pp. 79–84.
- [14] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 644–658, Mar. 2013.
- [15] M. Kraning, E. Chu, J. Lavaei, and S. P. Boyd, "Message passing for dynamic network energy management," *Found. Trends Optim.*, vol. 1, no. 2, pp. 70–122, 2014.
- [16] T. Shiina and J. R. Birge, "Stochastic unit commitment problem," *Int. Trans. Oper. Res.*, vol. 11, no. 1, pp. 19–32, Jan. 2004.
- [17] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267–278, Mar. 2010.
- [18] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, "Distributed MPC strategies with application to power system automatic generation control," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 6, pp. 1192–1206, Nov. 2008.
- [19] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [20] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*. New York, NY, USA: Springer, 1998.
- [21] B. D. Rouhani and H. Khatibzadeh, "On the proximal point algorithm," *J. Optim. Theory Appl.*, vol. 137, no. 2, pp. 411–417, Dec. 2008.
- [22] R. Halvgaard, J. B. Jørgensen, N. K. Poulsen, H. Madsen, and L. Vandenberghe, "Decentralized large-scale power balancing," in *Proc. IEEE PES ISGT*, Lyngby, Denmark, 2013, pp. 1–5.
- [23] *CITIES Project*. [Online]. Available: <http://www.smart-cities-centre.org/>, accessed Oct. 12, 2015.