Chapter 21
Predictive Smart Grid Control with Exact Aggregated Power Constraints

Klaus Trangbaek[1], Mette Pedersen[2], Jan Bendtsen[1], and Jakob Stoustrup[1]

[1] Dept. of Electronic Systems, Automation and Control, Aalborg University, Fr. Bajers Vej 7C, 9220 Aalborg, Denmark
{ktr,dimon,jakob}@es.aau.dk
[2] DONG Energy Power, A. C. Meyersvænge 9, 2450 Copenhagen, Denmark
mehpe@dongenergy.com

Abstract. This chapter deals with hierarchical model predictive control (MPC) of smart grid systems. The design consists of a high-level MPC controller, a second level of so-called *aggregators*, which reduces the computational and communication-related load on the high-level control, and a lower level of autonomous consumers.

The control system is tasked with balancing electric power production and consumption within the smart grid, and makes active use of the flexibility of a large number of power producing and/or power consuming units. The load variations on the grid arise on one hand from varying consumption, and on the other hand from natural variations in power production from e.g. wind turbines.

The consumers represent energy-consuming units such as heat pumps, car batteries etc. These units obviously have limits on how much power and energy they can consume at any given time, which impose constraints on the optimisation taking place at the higher levels of the control system. This chapter presents a novel method for computing the aggregated constraints without approximation, yielding better utilisation of the units when the load variations are large.

The method is demonstrated through simulation of a smart grid containing consumers with very different characteristics. It is demonstrated how the novel aggregation method makes it possible for the top level controller to treat all these as one big consumer, significantly simplifying the optimisation.

21.1 Introduction

One of the greatest challenges in introducing large ratios of renewable energy into existing electric power systems, is the fluctuating and unpredictable nature of power

A. Keyhani and M. Marwali (Eds.): Smart Power Grids 2011, POWSYS, pp. 649–668.
springerlink.com

sources that harvest energy from wind, waves and sunlight. One of the main approaches to dealing with this difficulty, is a gradual shift toward so-called "smart grid" infrastructures, where both producers and consumers are equipped with control capabilities that allow them to participate in balancing efforts, etc. [6], and where discrepancies between supply and demand can be evened out via (short-term) storage of energy [15] or by voluntarily displacing consumption in time, so-called *demand-side management* [9]. One way to achieve this in practise is to exploit large thermal time constants in deep freezers, refrigerators, local heat pumps etc.; extra energy can be stored during off-peak hours, and the accumulated extra cooling can then be used by turning compressors and similar devices on less frequently during peak hours—see e.g., [1] and [10].

Since power systems are multi-variable and subject to constraints, and future reference estimates are often known in advance, e.g., from 24-hour power consumption traces, weather forecasts, etc., a natural choice for the top-level controller will in most cases be some sort of model-predictive controller (MPC)—see for instance [8], [11], [12], or [13].

In an earlier paper [14], the authors proposed a hierarchical control architecture for this type of system, which distributes power to consumers in such a way that the consumers participate actively in balancing external load disturbances on the grid while at the same time consuming a pre-determined amount of energy over a given time interval.

The proposed solution

- is based on a standard MPC solution at the top level
- is able to accommodate new units without requiring modifications of the top-level controller
- remains stable for an increasing number of units

The design could for example act like a simple "Virtual Power Plant"; see Figure 21.1.

However, the consumers are obviously subject to both power and energy constraints; hardware such as compressors can only consume certain amounts of power at any given time, and it is only allowed to store or release certain amounts of energy within a given interval, thus avoiding spoiling stored food in deep freezers, ensuring comfortable temperatures in housing, etc. The problem is that these constraints depend in a complicated manner on the current energy stored in the consumers, as well as how the power is distributed between them. Simple constraint estimates will therefore not allow the consumers to be used to their full potential in case of rapid load fluctuations.

In this chapter, we present a method for computing non-conservative future constraints for a (possibly large) number of consumers, based on knowledge of their individual power and energy constraints. The method is illustrated on a simulation example that uses measured wind data as load variation, and simple, but reasonably realistic consumer models. The example illustrates that the exact bounds computed

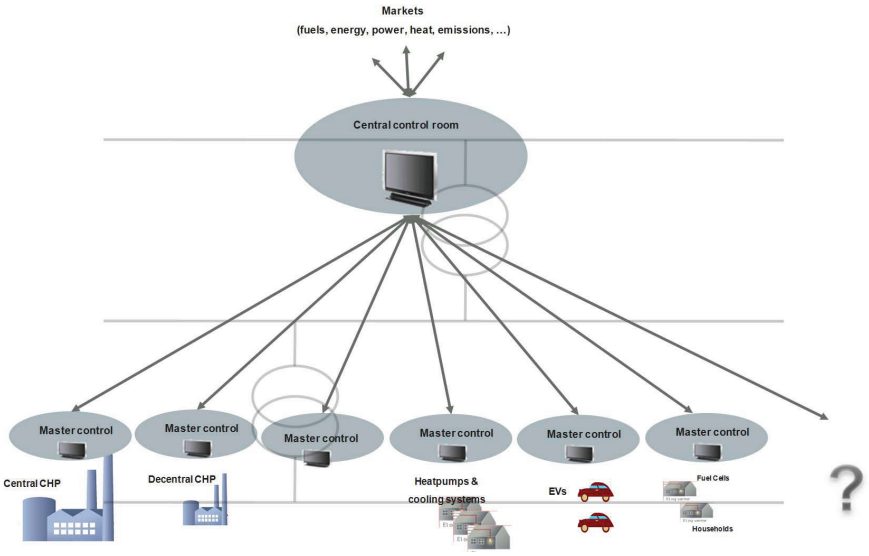

Fig. 21.1 A vision for smart grids: Virtual Power Plants which aggregate producing or consuming units

by the proposed method leads to a more efficient power distribution than a conventional, conservative approach in case of fast changes in the load.

We begin by describing the general hierarchical control setup in Section 21.2 and then go into more details with respect to the consumers in Section 21.3, in which we argue that the power and energy constraints give rise to well-defined convex polytopes. Section 21.4 presents the main contribution of this chapter, an efficient method for computing the optimisation constraints utilising these polytopes. Next, Section 21.5 illustrates how the hierarchical predictive control scheme should be implemented to make use of the polytopes, whereupon Section 21.6 demonstrates the efficiency of the constraint computation on a smart grid simulation example where fast fluctuations in wind power are absorbed by intelligent consumers.

Our notation is mostly standard. Signals are scalars, unless otherwise noted. In case of vectors and matrices, it is stated which spaces the objects belong to, unless it is obvious from context. Subscript $(\cdot)_k$ denotes discrete-time sample No. k. We denote by $(\cdot)^*$ a stacked vector on a finite horizon of length N_h, e.g $P^*_{i,k} = \left[P^T_{i,k}\ P^T_{i,k+1}\ \cdots\ P^T_{i,k+N_h-1}\right]^T$. Furthermore, for compactness of notation, we define sequences and symbols with zero or negative index as empty symbols; for example, for $i = 1$, we define

$$[P^*_1, \ldots, P^*_{i-1}, P^*_{i+1}, \ldots, \overline{P}^*_N] := [P^*_2, \ldots, P^*_N]$$

since the first sequence, $P_1^*, \ldots, \overline{P}_0^*$, should be interpreted as an empty vector (the index is counting "up from 1 to 0", which is not possible). I and 0 denote identity and zero matrices (scalars) of appropriate dimensions. Finally, $\mathbf{1}$ indicates a vector consisting entirely of entries with the value 1.

21.2 System Description

We consider a setup of the form shown in Figure 21.2. The controller is given the task of following a specific externally generated trajectory of consumption and/or production of power, represented by P_{load}. The objective is to maintain a certain system-level *balance* (between demand and production); the error in the balance is represented by the scalar signal E_{bal}, which must be driven to 0. Over time the demand and production must match, however, and the disturbance P_{load} hence enters as short-time changes in the balance, whereas E_{bal} is an integrated error signal.

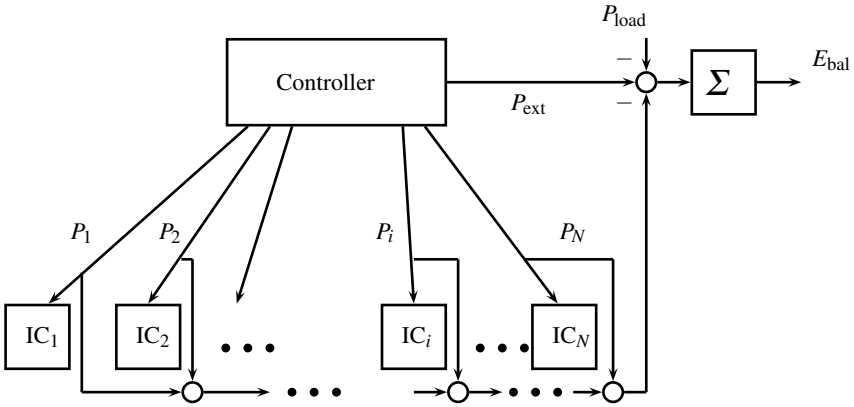

Fig. 21.2 Control architecture with grid-level control and intelligent consumers [14]

The control system can affect the power/energy balance by either distributing power to "intelligent consumers" $\text{IC}_i, i = 1, \ldots, N$, or by requesting external power production units (typically thermal power plants) to increase/decrease their production, thereby providing the required amount of production adjustment P_{ext}. Obviously, it is preferable to distribute the incoming power to the ICs, whereas utilising P_{ext} incurs a significant cost. The high-level controller is able to direct power to the consumers, but must ensure that each consumer *on average* receives a specific amount of energy, and certain upper and lower bounds $\underline{P}_i$ and $\overline{P}_i$ may not be exceeded. Note that these bounds are unique to each consumer.

In most cases, it is furthermore desirable to minimise rapid changes in P_{ext} by distributing extra power to the ICs instead, as it tends to be costly to rapidly increase/decrease production in many types of plants.

Finally, the consumers can only receive a certain range of energy, which will be represented by upper and lower bounds $\underline{E}_i$ and $\overline{E}_i$, respectively.

As a natural result of the considerations above, the controller needs to solve an optimisation problem of the form

$$\begin{aligned} \min_{P^*_{i,k}, P^*_{\text{ext},k}} \quad & \sum_{l=k+1}^{N_h} J(E_{\text{bal},l}, P_{\text{load},l}, P_{\text{ext},l}, \Delta P_{\text{ext},l}) \\ \text{s. t.} \quad & \underline{P}_i \le P_{i,l} \le \overline{P}_i, \quad i = 1,\dots,N \\ & \underline{E}_i \le E_{i,l} \le \overline{E}_i, \quad i = 1,\dots,N \end{aligned}$$

at each sample k for some pre-specified prediction horizon N_h, where $J(\cdot)$ is an appropriate cost function and $\Delta P_{\text{ext},k} = P_{\text{ext},k} - P_{\text{ext},k-1}$.

This is a quite standard optimisation problem, which as mentioned earlier can be tackled using well-established methods, e.g., [8], [11], [12], or [13]. However, as the number of ICs grows, it becomes increasingly difficult to keep track of the individual units' constraints. We address this in the following.

21.3 Intelligent Consumers

In our simplified setup, each IC can be described as an integrator with the energy storage equation

$$E_{i,k+1} = E_{i,k} + T_s(P_{i,k} - P_{c,i,k}) \tag{21.1}$$

where E_i is the energy stored in the i-th IC, P_i is the power delivered to the IC, $P_{c,i}$ is the (constant) power consumed, and T_s is the size of the time step. For ease of notation, we will assume $T_s = 1$ in the following. We further assume that $P_{c,i} = 0$. If this is not the case, the average consumption can be considered as part of the external load, and the IC seen as only the controllable consumption around this average.

The consumption rate and energy storage levels are limited by physical constraints:

$$\underline{P}_i \le P_i \le \overline{P}_i, \quad \underline{E}_i \le E_i \le \overline{E}_i. \tag{21.2}$$

As indicated above, in order to be of use in predictive control, these constraints must be formulated as constraints on the control variable P_i over a finite horizon.

Starting at time step k, the constraints limit the potential consumption of IC_i as illustrated in Figure 21.3. As shown in the figure, over a finite horizon of length $N_h > 0$, the constraints on $P^*_{i,k} = \begin{bmatrix} P_{i,k} & P_{i,k+1} & \dots & P_{i,k+N_h-1} \end{bmatrix}^T$ can also be represented by a polytope in $\mathbb{R}^{N_h}$, where the coordinate axis directions correspond to future sample numbers $k+1, k+2$, etc. Clearly, the energy storage constraints mean that the constraints at one time step depend on the consumption rate at the previous time step; that is, as long as the constraints $\underline{E}_i \le E_i \le \overline{E}_i$ are not active, the polytope is a hypercube, but when the level constraints do become active, they 'cut off' convex

subsets of the hypercube via intersection by hyperplanes, as indicated in the right part of the figure.

We shall denote these polytopes *resource polytopes* in the following. In short, the power a given consumer can receive in future time steps is bounded by these polytopes. The question then becomes, how to aggregate the many constraints in a meaningful, non-conservative manner.

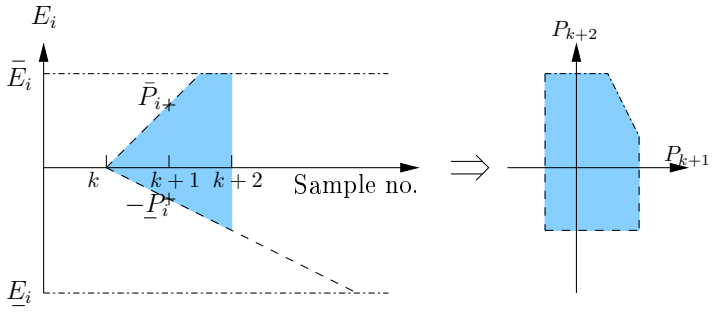

Fig. 21.3 At a given time step k, power $(---)$ and energy $(-\cdot-)$ constraints limit the future consumption for consumer i. The feasible power consumption that can be consumed by the IC within the next two time steps is indicated by the shaded area. The feasible set can also be represented as a polytope (right).

21.4 Minkowski Addition of Resource Polytopes

Along with Section 21.5, this section presents the main contribution of this chapter, an exact method for computing aggregated constraints on the consumption of the ICs.

As shown above, the feasible consumption rates are constrained to a convex polytope in $\mathbb{R}^{N_h}$, so it is relevant to investigate the geometric properties of these sets. Generally, a convex polytope Π can be represented by a half plane description:

$$\Pi = \left\{ x \in \mathbb{R}^{N_h} \middle| \Phi x \leq \gamma \right\}$$

where $\Phi \in \mathbb{R}^{M \times N_h}$ is a matrix, $\gamma \in \mathbb{R}^M$ is a vector, and $\leq$ is taken element-wise. Equivalently, Π can be represented by a vertex description:

$$\Pi = \left\{ x = \sum_{i=1}^{N_v} \alpha_i v_i \middle| 0 \leq \alpha_i, \sum_{i=1}^{N_v} \alpha_i \leq 1 \right\}$$

where $v_i \in \mathbb{R}^{N_h}$ are the N_v vertex vectors of the convex polytope [5]. In the first description, Π is parameterised by matrices Φ, γ and in the second description by the vertex set $V = \{v_i, i = 1, \ldots, N_v\}$.

Tools for automatic conversion between the two representations exist, for instance the Multi-Parametric Toolbox for Matlab [7]. Conversion from vertex to half plane representation for convex polytopes of high dimension is computationally heavy and numerically challenging, while converting the other way is generally easier. However, as we shall discuss below, the nature of the resource distribution problem leads to convex polytopes with a particular structure, which makes it possible to do the conversions in a direct way.

The *Minkowski sum* of a number of convex polytopes, $\Pi_1, \ldots, \Pi_n$, is defined as the (polytopic) set of all sums of elements from the individual polygons:

$$\Pi_\Sigma = \left\{ x_\Sigma = \sum_{i=1}^{n} x_i \middle| \forall x_i \in \Pi_i \right\}$$

This is exactly what we need to compute for the ICs, in order to provide the control level with a single constraint set; that is, given consumption capacities for a number of ICs over a horizon, the total capacity will be given by the Minkowski sum of these.

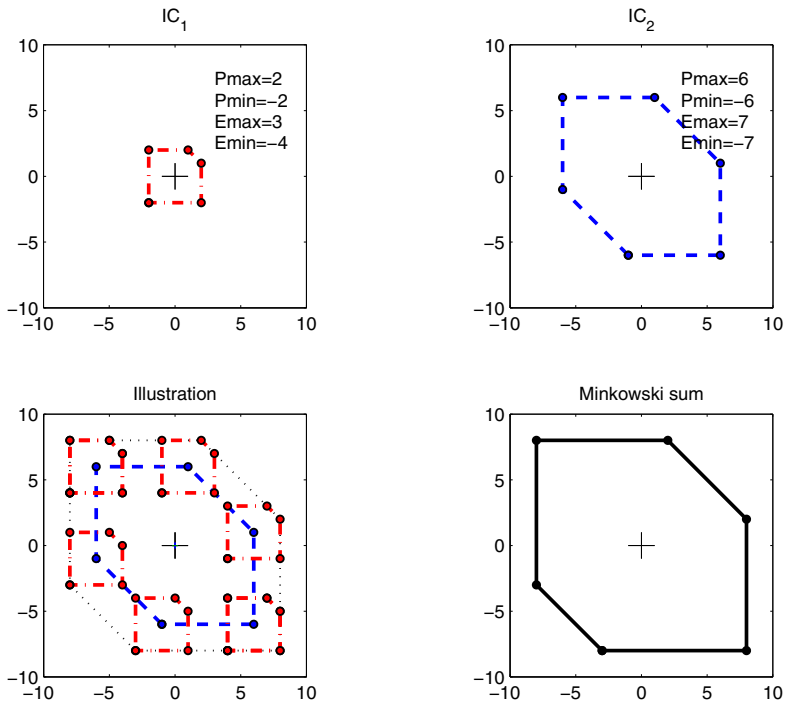

Fig. 21.4 Minkowski summation of two resource polytopes. The two polytopes in the top plots are Minkowski added, resulting in the lower right polytope.

Consider the example convex polytopes in Figure 21.4. The horizon length is 2, implying that the resource polytope exists in $\mathbb{R}^2$, and the axes correspond to the possible consumption rates in the first and second time steps. The top plots represent two ICs, while the bottom plots show the resulting sum. We make the following observations:

- consumption rate and capacity limits cannot simply be added to give the Minkowski sum.
- if we allow some of the vertices to be identical, the convex polytopes all have the same basic shape (i.e., we consider them as having *identical numbers of vertices*).
- the polytopic sum is in itself a polytope with the corresponding number of vertices and edges that represent resource constraints; hence, it can be considered as a consumer in itself with power/energy limits that vary over the horizon.

These observations hold for any number of consumers and horizon lengths, and will be helpful in obtaining computational feasibility.

Computing the Minkowski sum of a large number of general polytopes is computationally demanding [4]. Hence, although it is technically possible for a given vertex representation to compute all possible combinations of vertices via the Minkowski sum above, for a high number of consumers this is not feasible. However, if the vertices can somehow be ordered by their direction, so that a one-to-one correspondence is achieved, then it is only necessary to perform the summation vertex by vertex.

Achieving such an ordering is in general not possible. Defining a direction in dimensions higher than 2 is not straight-forward, and the number of vertices may in general not be the same for all polytopes. However, as indicated above, we can allow to the resource polytopes to have multiple identical vertices, which enables us to form the vertex representation in a way that is ordered. We simply consider all possible orders of minimisation and maximisation of consumption rates. Then the Minkowski sum of two such convex polytopes can be computed by a straightforward summation.

For instance, given a horizon length of 3, we may compute the maximum consumption on time step 2 if the consumption at the two other time steps can be chosen freely. Given this maximum consumption, we can find the minimal consumption on the third time step, and finally, given this, it is possible to find the maximum consumption on the first time step. Assuming a prediction horizon of N_h steps, there are $N_h!$ orders and for each order there are 2^{N_h} combinations of maximisation and minimisation, resulting in $N_h!2^{N_h}$ vertices. Many of the vertices will be identical, but they will all be on the boundary of the convex polytope.

Algorithm 1 is a recursive algorithm that produces a $N_h!2^{N_h} \times N_h$ matrix V, where each row v_i is a vertex vector representing minimum or maximum power consumption rates at time step i. The initial energy storage level of each IC is assumed to be zero, so the current level should be subtracted from the limits before invoking the algorithm.

Algorithm 1. PowerVertices($\overline{E}^*, \underline{E}^*, \overline{P}^*, \underline{P}^*$)

$N_h \leftarrow \dim(\overline{E}^*)$
$L \leftarrow (N_h - 1)!2^{N_h-1}$
Initialize $V \in \mathbb{R}^{2N_hL \times N_h}$
$f_{max} \leftarrow 0, f_{min} \leftarrow 0$
for $i = 1$ **to** N_h **do**
  $v_{max,i} \leftarrow \min(\overline{E}_i^* - f_{min}, \overline{P}_i^*)$
  $v_{min,i} \leftarrow \max(\overline{E}_i^* - f_{max}, \underline{P}_i^*)$
  $f_{max} \leftarrow \min(f_{max} + \overline{P}_i^*, \overline{E}_i^*)$
  $f_{min} \leftarrow \max(f_{min} + \underline{P}_i^*, \underline{E}_i^*)$
end for
for $i = 1$ **to** N_h **do**
  $\overline{\rho} \leftarrow [\overline{P}_1^*, \ldots, \overline{P}_{i-1}^*, \overline{P}_{i+1}^*, \ldots, \overline{P}_{N_h}^*]$
  $\underline{\rho} \leftarrow [\underline{P}_1^*, \ldots, \underline{P}_{i-1}^*, \underline{P}_{i+1}^*, \ldots, \underline{P}_{N_h}^*]$
  if $i = 1$ **then**
    $\rho_1 \leftarrow \overline{E}_i^* - v_{max,i}$
  else
    for $j = 1$ **to** $i - 1$ **do**
      $\rho_j \leftarrow \overline{E}_i^* - v_{max,i} - \sum_{k=j}^{i-1} \underline{P}_k^*$
    end for
  end if
  $\overline{\varepsilon} \leftarrow [\min(\overline{E}_1^*, \rho_1), \ldots, \min(\overline{E}_{i-1}^*, \rho_{i-1}), \overline{E}_{i+1}^* - v_{max,i}, \ldots, \overline{E}_{N_h}^* - v_{max,i}]$
  $\underline{\varepsilon} \leftarrow [\underline{E}_1^*, \ldots, \underline{E}_{i-1}^*, \underline{E}_{i+1}^* - v_{max,i}, \ldots, \underline{E}_{N_h}^* - v_{max,i}]$
  for $l = 1$ **to** L **do**
    $V_{l+2(i-1)L,i} \leftarrow v_{max,i}$
  end for
  if $N_h > 1$ **then**
    $\Lambda = \text{PowerVertices}(\overline{\varepsilon}, \underline{\varepsilon}, \overline{\rho}, \underline{\rho})$ {*Recursive call where* $\overline{\varepsilon}$ *replaces* $\overline{E}^*$*, etc.*}
    for $l = 1$ **to** L **do**
      for $m = 1$ **to** N_h, $m \neq i$ **do**
        $V_{l+2(i-1)L,m} \leftarrow \Lambda_{l,m}$
      end for
    end for
  end if
  The steps above are now repeated for the lower-bound vertices, essentially replacing $\overline{(\cdot)}$ *by* $\underline{(\cdot)}$ *and* $\min(\cdot)$ *by* $\max(\cdot)$*, storing the result in* $V_{l+2(i-1)L+L,\cdot}$*,* $l = 1, \ldots, L$*.*
end for
return V

The number of vertices may seem excessive, but the advantage is that it results in a meaningful ordering, and because of the special structure of the resource polytopes, the Minkowski sum results from a simple vertex by vertex vector summation.

Note that, for optimisation purposes, a half plane representation is more suitable than the vertex representation resulting from the above procedure. As mentioned, the conversion can be performed automatically by existing tools, but for a large number of vertices, it becomes computationally heavy. Luckily, the structure of the resource polytopes makes it possible to do a very simple conversion by considering power and energy constraints directly. Consider a resource polytope spanned by vertex

vectors v_i, $i = 1 \ldots N_v$, and assume that the j'th element of v_i, $v_{i,j}$, is the coordinate corresponding to the j'th time step. Then energy constraints are then simply found as

$$\overline{E} = \max_i \sum_{j=1}^{N_h} v_{i,j} \qquad \text{and} \qquad \underline{E} = \min_i \sum_{j=1}^{N_h} v_{i,j}$$

respectively. Correspondingly, the time-varying power constraints are computed as

$$\overline{P}^*_j = \max_i v_{i,j} \qquad \text{and} \qquad \underline{P}^*_j = \min_i v_{i,j}$$

respectively. We then have the half plane representation

$$\begin{bmatrix} I \\ -I \\ T \\ -T \end{bmatrix} P^* \leq \begin{bmatrix} \overline{P}^* \\ -\underline{P}^* \\ \overline{E}\mathbf{1} \\ -\underline{E}\mathbf{1} \end{bmatrix} \tag{21.3}$$

where T is a Toeplitz matrix with the i-th row consisting of i ones followed by $N_h - i$ zeros, and $\mathbf{1}$ is a column vector consisting of N_h 1-entries.

21.5 Distribution

We now return to the original problem described in Section 21.2.

When controlling a large number of ICs, having a decision variable and a set of constraints for each is not computationally feasible. To alleviate this, we introduce a number of so-called *aggregators* $A_j, 1 \leq j \leq N_A$. An aggregator serves as an interface to a subset $\mathscr{J}^j$ of ICs, aggregating their capacities into one constraint set. In the following we will present how to aggregate the resource polytope representations for the ICs under an aggregator's jurisdiction.

The proposed scheme is shown in Figure 21.5. On both top and aggregator levels, everything is computed over a receding horizon of N_h time steps.

Starting from the lowest level, the ICs provide the aggregators with their current energy levels, E_i. From these, the aggregators compute the consumption constraints of each IC over the horizon. Using the vertex representation described in Section 21.4, these constraints are then added to provide constraints to the top level.

Given aggregated constraints from each aggregator, the top level sums all these constraints to obtain the total capacity of all the ICs. Given this constraint, the top level can then optimise a performance function using the sum of consumptions $P^*_\Sigma = \sum_{i=1}^{N} P^*_{i,k}$ as a decision variable. The optimisation problem at the top level depends on the specific system. In a smart grid setting, the task is usually to balance a time varying load using some central production unit (power plant), and the ICs are then used for handling fast load variations, minimising the rate of change for the central unit.

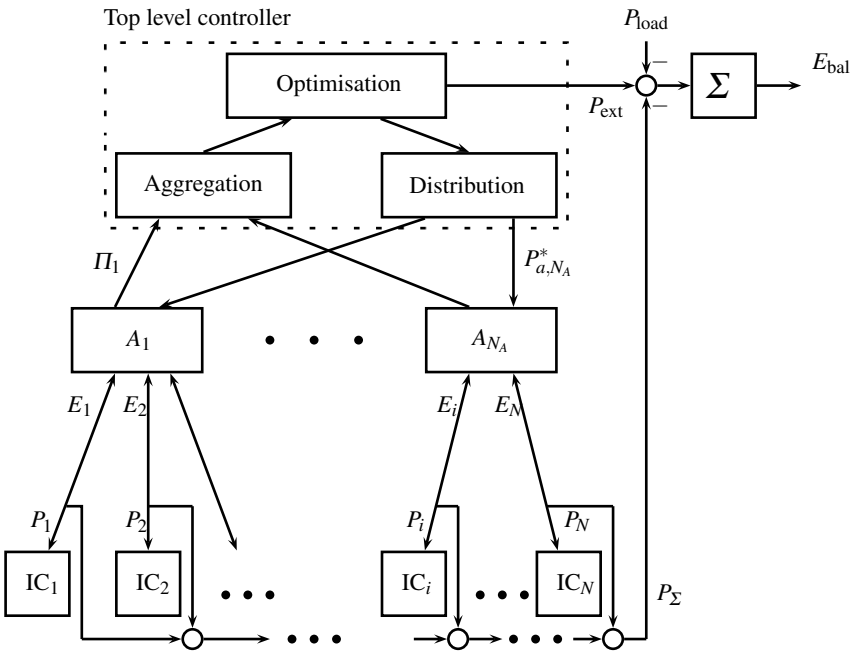

Fig. 21.5 Control architecture with grid-level control, aggregators and intelligent consumers. At each sample instant, consumer IC_i receives power P_i and returns information of its own energy storage level E_i.

The consumption sum P^*_Σ is then distributed between the aggregators, which further distribute it among the ICs. The distribution is performed in a way that ensures that constraints can be met over the entire horizon. This still leaves some freedom to distribute the consumption in a way that brings the individual consumers closer to some reference energy level, $E_{ref,i}$. This level could be picked in a way that increases flexibility, for instance in the middle between the capacity constraints. The desired consumption from the IC point of view is then $\tilde{E}_{i,k} = E_{ref,i} - E_{i,k}$.

At the top level, a half plane representation of the constraints $\Phi_{a,i} P^*_{a,i} \leq \gamma_{a,i}, i = 1,\ldots,N_A$ is computed for each aggregator. The optimisation and distribution is then performed by solving a standard quadratic program at sample k:

$$\begin{aligned} \min_{P^*_{a,1},\ldots,P^*_{a,N_A}} \quad & \sum_{i=1}^{N_A} \beta_{da,i} \Big(\sum_{j=k}^{k+N_h-1} P_{a,i,j} - \sum_{j \in \mathscr{J} i} \tilde{E}_{j,k} \Big)^2 \\ \text{s.t.} \quad & \Phi_{a,i} P^*_{a,i} \leq \gamma_{a,i},\ i = 1,\ldots,N_A \\ & \sum_{i=1}^{N_A} P^*_{a,i} = P^*_\Sigma \end{aligned} \tag{21.4}$$

Here $\beta_{da,i} = \sum_{j \in \mathscr{J}^i} \beta_{d,j}$ and $\beta_{d,j}$ is explained below. These consumption rates can then be distributed by the aggregators among their associated ICs in a similar manner, i.e., for aggregator j at sample k we solve the optimisation problem:

$$\begin{aligned} \min_{w^*_{m,k}, m \in \mathscr{J}^j} \quad & \sum_{i \in \mathscr{J}^j} \beta_{d,i} \Big(\sum_{j=k}^{k+N_h-1} P_{i,j} - \tilde{E}_{i,k} \Big)^2 \\ \text{s.t.} \quad & \Phi_i P_i^* \leq \gamma_i, \; i \in \mathscr{J}^j \\ & \sum_{i \in \mathscr{J}^j} P_i^* = P_{a,i}^* \end{aligned} \tag{21.5}$$

where Φ_i, γ_i represent constraints for the individual ICs. The parameter $\beta_{d,i}$ is an IC specific penalty on distributing load to the IC. The value of $\beta_{d,i}$ can be chosen in different ways, but here it is set to $\overline{P}_i - \underline{P}_i$. This means, that in the distribution the aggregator first distributes load to relatively slow ICs (slow meaning narrow power limits) and thus alleviating faster units.

With the power distributed, the ICs will then absorb their assigned consumption during a time step, after which the entire procedure is repeated for the new energy levels.

21.5.1 Mid-Ranging

At times when the disturbances are steady, the top level should attempt to keep the energy at the consumers at a level that ensures wide manoeuvrability in response to future disturbances, i.e. by keeping the levels away from the capacity limits. By adding a small term, e.g. $\beta_r(\sum_{j=0}^{N} \tilde{E}_{j,k} - \sum_{i=0}^{N_h-1} P^*_{\Sigma,k,i})^2$, to the performance function at the top level, the optimisation will bring the levels closer to the references when there is no need to use the consumers in the overall balancing.

Note that the aggregators are not required to submit all energy levels of the consumers to the top level. Only the sum associated with each aggregator needs to be communicated in order to compute the total sum.

21.6 Simulation Example

The example described in this section simulates a small isolated power grid consisting of a wind farm, a set of intelligent consumers (ICs), namely heat pumps and refrigeration systems, a set of regular consumers (RCs) and a power plant.

The wind farm and the power plant are the production capacities. The RCs must be supplied a constant power at all times. The average production of the wind farm and power plant thus corresponds to the base consumption of the RCs. The wind farm production however exhibits fast fluctuations, which must be balanced by the ICs and if necessary by the power plant.

21.6.1 Problem Formulation

The ICs at the lowest level are modelled as simple power and energy constraint units (constraint limits are denoted $\overline{P}_i$, $\underline{P}_i$, $\overline{E}_i$, $\underline{E}_i$), which are governed by their own energy balance as explained in Section 21.3.

The top level controller has three objectives. As explained earlier the energy supplied to the RCs is the average production of the wind farm and power plant. Fluctuations of the wind power production however must be balanced by the ICs and power plant. The first objective is therefore to keep the energy balance

$$E_{k+1} = E_k + T_s(P_{plant,k} - P_{wind,k} - P_{\Sigma,k})$$

close to zero, where $P_{\Sigma,k} = \sum_{i=1}^{N} P_{i,k}$ is the power absorbed by the ICs. P_{wind} are fluctuations of the wind farm power production and P_{plant} is the power plant deviation from the base line production, respectively corresponding to P_{load} and P_{ext} in Section 21.5.

The second objective at the top level is to reduce the strain on the power plant by limiting the size of changes in production. This means that fast fluctuations of P_{wind} should be handled by the ICs and only slower fluctuations should be left to the power plant.

The last objective of the top level controller is to bring the ICs close to their reference levels, as explained in Section 21.5.1.

With these three objectives the optimisation problem at the top level for a prediction horizon N_h is

$$\min_{P^*_{\Sigma,k}, P^*_{\text{plant, k}}} \sum_{i=1}^{N_h} z_{k+i}^{\mathrm{T}} Q_i z_{k+i} + \beta_r \Big(\sum_{j=0}^{N} \tilde{E}_{j,k} - \sum_{i=0}^{N_h-1} P^*_{\Sigma,k,i}\Big)^2$$

where

$$z_k = \begin{bmatrix} E_k \\ P_{\text{plant, k}} - P_{\text{wind, k}} \\ P_{\text{plant, k}} - P_{\text{plant, k-1}} \end{bmatrix}$$

and $Q_i = diag(\beta_e, \beta_q, \beta_p)$ for $i = 1, 2, \ldots, N_h - 1$. The penalty on $P_{\text{plant}} - P_{\text{wind}}$ is included to improve closed loop performance. The terminal weight Q_{N_h} is found by dual mode analysis: Assuming that the constraints are not active after $k_0 + N_h$, the optimal trajectory will be described by the dynamics $z_{k+1} = \tilde{A} z_k$, where $\tilde{A}$ can be found as the closed loop matrix resulting from a standard LQR problem. By construction, all eigenvalues of $\tilde{A}$ have modulus less than one, so we can find a symmetric positive definite matrix Q_{N_h} that solves the discrete-time Lyapunov equation

$$\tilde{A}^T Q_{N_h} \tilde{A} = Q_{N_h} - Q$$

which yields the required weight on the terminal state.

Table 21.1 Parameters for the considered ICs

	Refrigeration system [2]	Heat pump [3]
Maximum power	10 kW	4.3 kW
ΔT	1 K	3 K
Volume	5 m^3	12 m^3
Volumetric heat capacity	1.9 $\frac{MJ}{m^3K}$ (ice)	2.4 $\frac{MJ}{m^3K}$ (concrete)
Average power	7 kW	3 kW
$\overline{E_i}$	2.6 kWh	8 kWh
$\underline{E_i}$	0 kWh	0 kWh
$\overline{P_i}$	3 kW	1.3 kW
$\underline{P_i}$	-7 kW	-3 kW

21.6.2 Simulation Data and Parameters

The wind farm power production, which has to be balanced in the simulations, consist of production data from Horns Rev 2, a 209 MW offshore wind farm in the North Sea owned and operated by DONG Energy. The period covered extends from 15:00 to 22:00 of February 1st 2011 and the average production value has been subtracted, since this is used to supply the RCs.

Two types of ICs are considered, namely a domestic heat pump and a supermarket refrigeration system. The ICs are modelled as simple energy and power constraint units, so the internal dynamic is not modelled in detail. These ICs each have a primary purpose, which must be met, namely to keep the house and frozen goods within a certain acceptable temperature interval. The main mediums, which the heat pump and cooling system must respectively heat and cool, are the concrete floor of a single family home and the freezer content. The thermal energy resource available is thus given by $C \cdot V \cdot \Delta T$, where V and C are the volume and volumetric heat capacity of the main medium and ΔT is the acceptable temperature interval. The heat pump is assumed to have a coefficient of performance of 3.0, so the electrical energy resource is one third of the thermal resource.

The ICs have an average power consumption, which balances the outside influences on the system. For the heat pump this will depend on the outside temperature and the insulation of the house and similarly for the refrigeration system it will depend on the supermarket temperature and insulation of the freezer.

Additional simulation parameters are $\beta_e = 1$, $\beta_p = 0.001$, $\beta_r = 0.0002$, $\beta_q = 0.001$ and $N_h = 4$. These parameters corresponds to a terminal cost of

$$Q_{N_h} = \begin{bmatrix} 1 & -4 \times 10^{-3} & 0 \\ -4 \times 10^{-3} & 1 & 0 \\ 0 & 0 & 1 \times 10^{-3} \end{bmatrix}$$

Four aggregators are included in the simulations and each aggregator handles 160 heat pumps and 160 refrigeration systems. The time step is 15 minutes and P_{wind} is assumed known over the horizon N_h.

21.6.3 The Cautious Method

The advantage of the resource polytopes is, that the full flexibility of the ICs are communicated to the top level. To illustrate the impact of this, the method is compared to a setup, where the interface between top level and aggregator level does not allow power constraints to vary over the horizon N_h. This method is denoted *the cautious method*.

As described earlier everything at the top and aggregator level is computed over a horizon of N_h time steps. Given a horizon of N_h at time step k the cautious aggregators determine $\overline{P}_{i,\,Cautious} = \min(\overline{P}_i, \frac{\overline{E}_i - E_{i,k}}{N_h})$ and $\underline{P}_{i,\,Cautious} = \max(\underline{P}_i, \frac{\underline{E}_i - E_{i,k}}{N_h})$. These values are communicated to the top level, which should provide at most $\overline{P}_{i,\,Cautious}$ and at least $\underline{P}_{i,\,Cautious}$ to each IC at each time step over the next prediction horizon. This insures that the energy constraints of the IC's are not violated.

Note that the point of this approach is that the power constraints over the horizon become boxes, which can then easily be Minkowski added.

21.6.4 Simulation Results

Simulation results are depicted in Figure 21.6. The top graph depicts P_{wind} and the power plant production P_{plant}. The bottom graph depicts $E_{Cautious}$ and $E_{Resource\ Polytopes}$. Even though the two methods have to balance the exact same oscillations with the exact same resource available at the bottom level, the resource polytope method is noticeably better at maintaining the energy balance and limiting fluctuations of P_{plant}. The reason for this can be seen in Figure 21.7. For the relatively slow heat pump the performance of the two methods is as expected quite similar. For the faster refrigeration system, however, the polytope method is able to get closer $\overline{P}$, when it is needed, which means a better utilisation of the flexibly resource.

Notice that there are two occasions, where both methods have trouble maintaining the energy balance, namely between 16.00 and 17.00 and again between 20:00 and 21:00. The explanation for this is found in Figure 21.7. Between 16.00 and 17.00 and again between 20:00 and 21:00 the polytope method touches the upper bound on both IC power constraint. This means that flexible resource is exhausted, which causes the energy imbalance.

21.6.5 Parameter Dependency

The parameters in the model must be relatively balanced in order to reap the benefits of the resource polytopes. For instance if the IC's have very large energy

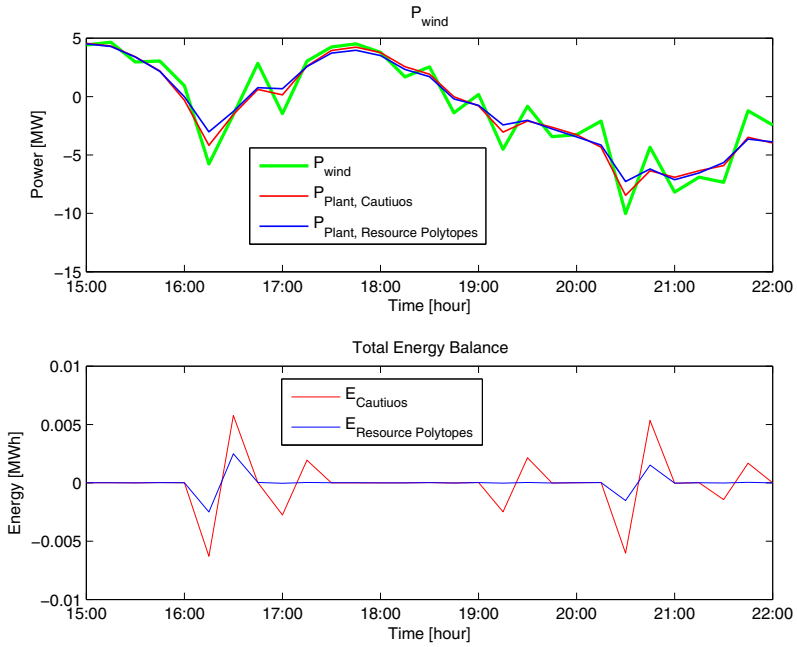

Fig. 21.6 Simulation example, P_{wind} and total energy balance. Notice that the resource polytope method is noticeably better at maintaining the energy balance

resources or if fluctuations of P_{ext} are very cheap the two method will have similar performance. To illustrate this simulations have been performed with an increasing number of IC's. When plenty of flexible resource is available at the lowest level the extra capacity available by using the resource polytope method is not needed and it is therefore not utilized. As discussed earlier the main tasks at the top level is to keep the energy balance at zero and limit fluctuations of P_{plant}. Key performance indicators are therefore given by

$$A = \sum_{k=1}^{N_{sim}} E_k^2$$

and

$$B = \sum_{k=1}^{N_{sim}-1} (P_{plant,k} - P_{plant,k+1})^2.$$

where N_{sim} is the entire simulation horizon. Results of the investigation is given in Figure 21.8 where A and B are given as a function of IC's per aggregator. All other parameters are as in Section 21.6.2, which means that four aggregators are

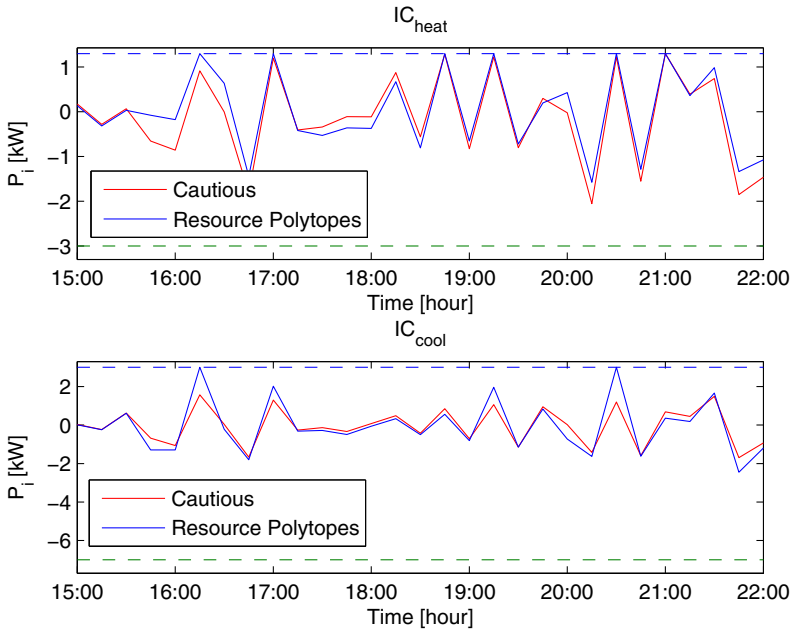

Fig. 21.7 Simulation example. Power consumption for an IC_{heat} and an IC_{cool} with resource polytopes and the cautious method

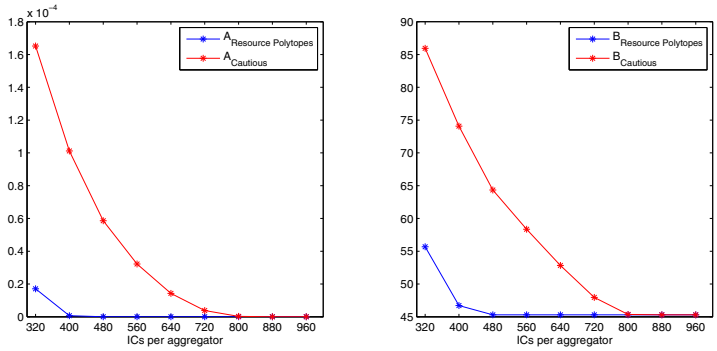

Fig. 21.8 Simulation example. Key performance indicators a as function of the number of IC's per aggregator. The difference in performance decreases as more units are included in the simulation.

included and each aggregator handles half cooling sustem and half heat pumps. As expected the performance indicators of the two methods decrease as more units are included in the simulations and for more than 800 IC's per aggregator the difference in performance is negligible.

21.6.6 Computational Burden

The main computational burdens are the vertex generation, the top level distribution and the aggregator level distribution, all of which must be performed at each time step. Summation of vertices, the top level optimisation and conversion to half plane representations are not major burdens.

The vertex generation can be performed separately for each IC and is thus easily distributed among aggregators. It could even be performed by the ICs, but then a large amount of data would need to be transferred upwards.

Increasing the number of aggregators leaves fewer ICs for each and thus makes the aggregator level distribution easier, but on the other hand it makes the top level distribution heavier. If this becomes a problem, extra levels could be inserted in the distribution hierarchy with no significant effect on the resulting performance.

The complexity of the quadratic programming of a distribution task increases approximately with the square of the number of receivers and approximately doubles with an increase in the horizon length. In the current implementation, the computational burden of performing the distribution at the top level or by an aggregator can then fairly accurately be approximated by

$$\text{distribution load: } \beta_d 2^{N_h} N_d^2 + \beta_0, \tag{21.6}$$

where N_d is the number of associated consumers or aggregators in the layer directly below.

The vertex computation burden is linear in the number of vertices and in the number of ICs

$$\text{vertex load: } \beta_v N_h! 2^{N_h} N_d, \tag{21.7}$$

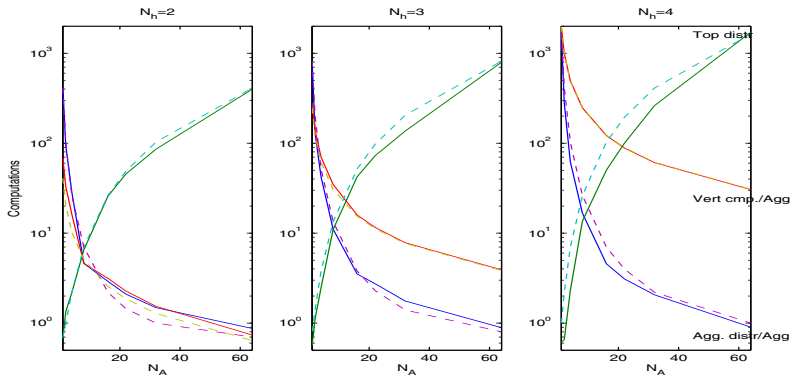

Fig. 21.9 Computational complexity with 64 ICs as a function of the number of aggregators. The plots show horion lengths of 2,3 and 4, respectively. The dashed lines show the model in Equations (21.6) and (21.7)

Simulation studies provide estimates $\beta_v \approx 0.125\beta_0$ and $\beta_d \approx 0.015\beta_0$, where the value of β_0 of course depends on the computer platform. This means that with N_h equal to 2, 3, 4 or 5, for an aggregator controlling respectively 16, 50, 200 or 1000 consumers, the computational burdens of vertex computation and power distribution are approximately equal.

Figure 21.9 illustrates the computational burden with 64 ICs distributed between a varying number of aggregators. As the number of aggregators grows, each aggregator handles fewer ICs and their vertex and distribution loads fall, while the top level has to handle more aggregators causing an increase in the computational burden.

21.6.7 Communication Load

The biggest data flow results from the vertex tables being communicated upward in the hierarchy. Power consumption profiles over the horizon are communicated downwards, but these are quite small in comparison. The vertex tables are of size $N_h!2^{N_h} \times N_h$ and an upper layer must receive one from each of its associated aggregators at each time step. If the aggregators (rather than the consumers) perform the vertex computation, then only the current energy level must be communicated from each consumer, and each consumer need only be given a consumption demand for each time step. There is no need for communication between aggregators on the same level or between consumers.

21.7 Discussion

A novel way to represent resource storage capacity has been presented. The representation has the following useful properties:

- the main constraint computation can be performed separately for each consumer,
- the aggregated constraints of a set of consumers can be computed without approximation by a simple summation,
- conversion to half plane representations, useful for optimisation, can be performed at low cost.

Since the constraint aggregation is exact, the scheme is nearly optimal. With respect to the distribution, it is possible that, in terms of future flexibility, a slightly better distribution could be obtained by a direct distribution at the top level, but for a high number of consumers this does not seem feasible to implement.

Although this chapter focuses on smart electrical grids, the resource polytopes could prove useful in representing other resource distribition applications, for instance in supply chain management.

References

1. Banakar, H., Luo, C., Ooi, B.: Impacts of Wind Power Minute-to-Minute Variations on Power System Operation. IEEE Trans. Power Syst. 23, 150–160 (2008)
2. Cai, J.: Control of Refrigeration Systems for Trade-off between Energy Consumption and Food Quality Loss. PhD thesis, Aalborg University (2008)
3. Energi, D.: Smart grid i Danmark. Energinet.dk (2010)
4. Gritzmann, P., Sturmfels, B.: Minkowski addition of polytopes: computational complexity and applications to Gröbner bases. SIAM J. Discrete Math. 6, 246–269 (1993)
5. Grünbaum: Convex Polytopes, 2nd edn. Springer, Heidelberg (2003)
6. Jarventausta, P., Repo, S., Rautiainen, A., Partanen, J.: Smart grid power system control in distributed generation environment. Annual Rev. Control 34, 277–286 (2010)
7. Kvasnica, M., Grieder, P., Baotic, M.: Multi-Parametric Toolbox (MPT) (2004), `http://control.ee.ethz.ch/~mpt/` (accessed April 01, 2011)
8. Maciejowski, J.: Predictive Control with Constraints. Prentice-Hall, Englewood Cliffs (2002)
9. Mohsenian-Rad, A., Wong, W., Jatskevich, J., Schober, R., Leon-Garcia, A.: Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. IEEE Trans. Smart Grid 1, 320–331 (2010)
10. Moslehi, K., Kumar, R.: A reliability perspective of the smart grid. IEEE Trans. Smart Grid 1, 57–64 (2010)
11. Picasso, B., De Vito, D., Scattolini, R., Colaneri, P.: An MPC approach to the design of two-layer hierarchical control systems. Automatica 46, 823–831 (2010)
12. Rossiter, J.: Model-based predictive control. CRC Press, Boca Raton (2003)
13. Scattolini, R.: Architectures for distributed and hierarchical model predictive control – a review. J. Process Control 19, 723–731 (2009)
14. Trangbaek, K., Bendtsen, J., Stoustrup, J.: Hierarchical model predictive control for resource distribution. In: Proc. of 49th IEEE Conf Decision and Control (2010)
15. Wade, N., Taylor, P., Lang, P., Jones, P.: Evaluating the benefits of an electrical energy storage system in a future smart grid. Energy Policy 38, 7180–7188 (2010)