2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Exact Power Constraints in Smart Grid Control

Klaus Trangbaek Mette Petersen Jan Bendtsen Jakob Stoustrup, *Senior Member, IEEE*

***Abstract*— This paper deals with hierarchical model predictive control (MPC) of smart grid systems. The objective is to accommodate load variations on the grid, arising from varying consumption and natural variations in the power production e.g. from wind turbines. This balancing between supply and demand is performed by distributing power to consumers in an optimal manner, subject to the requirement that each consumer receives the specific amount of energy the consumer is entitled to within a specific time horizon. However, in order to do so, the high-level controller requires knowledge of how much energy the consumers can receive within a given time horizon. In this paper, we present a method for computing these bounds as convex constraints that can be used directly in the optimisation. The method is illustrated on a simulation example that uses actual wind data as load variation, and fairly realistic consumer models. The example illustrates that the exact bounds computed by the proposed method leads to a better power distribution than a conventional, conservative approach in case of fast changes in the load.**

I. 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 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. [1], and where discrepancies between supply and demand can be evened out via (short-term) storage of energy [2] or by voluntarily displacing consumption in time, so-called *demand-side management* [3]. 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., [4] and [5].

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 case be some sort of model-predictive controller (MPC)—see for instance [6], [7], [8] or [9].

In an earlier paper [10], 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. 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 paper, 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 actual wind data as load variation, and fairly realistic consumer models. The example illustrates that the exact bounds computed 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 consumers in more detail in Section II, and argue that the power and energy constraints give rise to well-defined convex polytopes. Section III presents the main contribution, a method for computing the optimisation constraints. Section IV then goes into more detail with a hierarchical predictive control scheme for smart grids. Finally, Section V demonstrates the efficiency of the constraint computation.

Our notation is mostly standard. We denote by $(\cdot)^*$ a stacked vector on a finite horizon of length N_h, e.g $P^*_{i,k} = \begin{bmatrix} P^T_{i,k} & P^T_{i,k+1} & \cdots & P^T_{i,k+N_h-1} \end{bmatrix}^T$.

II. Resource polytopes

We consider a setup of the form shown in Figure 1, where P_{load} is a load imbalance that must be compensated for. The control system is able to do so by either distributing power to consumers $\text{IC}_i, i = 1,\ldots,N$, or by requesting external power units to produce the required amount P_{ext}. In most cases, it is desirable to minimise rapid changes in P_{ext} by distributing these to the ICs instead. Further details about the setup can be found in [10]; for now, we concentrate on the intelligent consumers.

We consider a simple discrete time setting, where each IC can be described as an integrator with the energy storage

The authors are with the Department of Electronic Systems, Automation and Control, Aalborg University, Denmark; M. Petersen is also affiliated with DONG Energy, Denmark. Email: {ktr, dimon, jakob}@es.aau.dk, mehpe@dongenergy.com

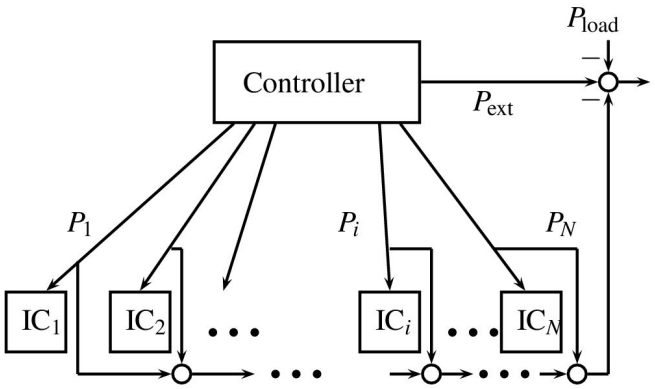

Fig. 1. Control architecture with grid-level control and intelligent consumers [10].

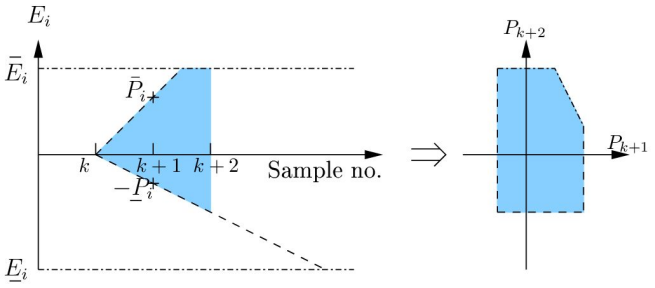

Fig. 2. At a given time step k, power ($- - -$) and energy ($- \cdot -$) constraints limit the future consumption for consumer i. The feasible power consumption that can be consumed by the IC within the next two time steps is indicated by the shaded area. The feasible set can also be represented as a polytope (right).

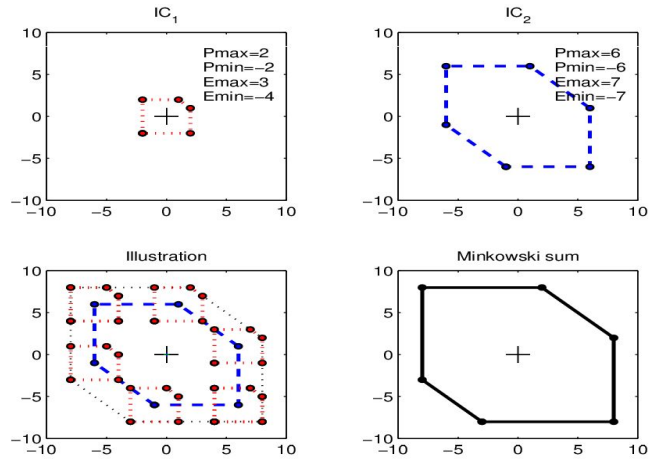

Fig. 3. Minkowski summation of two resource polytopes.

equation

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

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:

$$\underline{P}_i \le P_i \le \overline{P}_i, \quad \underline{E}_i \le E_i \le \overline{E}_i. \tag{2}$$

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 2. As shown in the figure, over a finite horizon of length $N_h > 0$, the constraints on P_i^* can also be represented by a polytope in $\mathbb{R}^{N_h}$. Clearly, the level 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 upper right part of the figure.

III. Minkowski Addition of Resource Polytopes

As shown above, the feasible consumption rates are constrained to a convex polytope in $\mathbb{R}^{N_h}$. Generally, a convex polytope can be represented by a half plane description, $\Pi = \{x \in \mathbb{R}^{N_h} | \Phi x \le \gamma\}$, where $\Phi \in \mathbb{R}^{M \times N_h}$ is a matrix, $\gamma \in \mathbb{R}^M$ is a vector, and $\le$ is taken element-wise, or by a vertex description, $\Pi = \{\sum_{i=1}^{N_v} \alpha_i v_i | 0 \le \alpha_i,\ \sum_{i=1}^{N_v} \alpha_i \le 1\}$, where v_i are the N_v vertex vectors of the convex polytope [11].

Tools for automatic conversion between the two representations exist, for instance MPT [12], but the computations remain challenging, so it is important to choose the most appropriate representation. In this case, the nature of the resource distribution problem leads to convex polytopes with a particular structure, which makes the vertex representation the more attractive choice. Furthermore, conversion to half-plane representation becomes simple and well-conditioned.

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, i.e. $\Pi_\Sigma = \{\sum_{i=1}^{n} x_i | x_i, i = 1, \ldots, n\}$. This is exactly what we need to compute for the ICs, in order to provide the control level with a single constraint set; given consumption capacities for a number of ICs over a horizon, the total capacity will be given by the Minkowski sum of these.

Consider the example convex polytopes in Figure 3. The horizon length is 2, and the axes represent the consumption rates in the first and second time steps. The top plots represent two ICs, the bottom plots the resulting sum. We make the following observations:

- consumption rate and capacity limits cannot simply be added to give the Minkowski sum.
- if we view some of the vertices as multiple, the convex polytopes all have the same basic shape.
- the sum can be considered as a consumer in itself with consumption rate and capacity limits, only now

the rate limits vary over the horizon, and for $N_h > 2$, capacity constraints may be needed for all combinations of samples.

Computing the Minkowski sum of a large number of general polytopes is computationally demanding [13]. Given vertex representations, all possible combinations of vertices can be computed and the Minkowski sum will then be spanned by all these vertex sums. For a high number of consumers, this is not feasible; however, if the vertices can somehow be ordered in the same sequence for each consumer, then it is only necessary to perform the summation vertex by vertex. This is indeed possible for resource polytopes, by permitting some vertices to be identical.

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 $V \in \mathbb{R}^{2N_hL \times N_h}$

for $k = 1$ **to** N_h **do**
  $v_{max,k} \leftarrow \min(\overline{P}, \overline{E}^*_k - \max(\underline{P} \cdot k, \underline{E}^*_k))$
  for $l = 1$ **to** L **do**
    $V_{l+2(k-1)L,k} \leftarrow v_{max,k}$
  end for
end for

for $k = 1$ **to** N_h **do**
  if $k = 1$ **then**
    $\underline{\Omega}^* \leftarrow [\underline{E}^*_2 - v_{max,1}, \ldots, \underline{E}^*_{N_h} - v_{max,1}]$
    $\overline{\Omega}^* \leftarrow [\overline{E}^*_2 - v_{max,1}, \ldots, \overline{E}^*_{N_h} - v_{max,1}]$
  else
    $\underline{\Omega}^* \leftarrow [\underline{E}^*_1, \ldots, \underline{E}^*_{k-1},$
      $\underline{E}^*_{k+1} - v_{max,k}, \ldots, \underline{E}^*_{N_h} - v_{max,k}]$
    $\overline{\Omega}^* \leftarrow [\min(\overline{E}^*_1, \overline{E}^*_k - v_{max,k} - ((k-1)-1)\cdot\underline{P}),$
      $\min(\overline{E}^*_2, \overline{E}^*_k - v_{max,k} - ((k-1)-2)\cdot\underline{P}),$
      $\ldots,$
      $\min(\overline{E}^*_{k-1}, \overline{E}^*_k - v_{max,k}),$
      $\overline{E}^*_{k+1} - v_{max,k}, \ldots, \overline{E}^*_{N_h} - v_{max,k}]$
  end if
  if $N_h > 1$ **then**
    $\Lambda =$ PowerVertices($\overline{\Omega}^*, \underline{\Omega}^*, \overline{P}, \underline{P}$)
    for $l = 1$ **to** L **do**
      for $m = 1$ **to** N_h, $m \neq k$ **do**
        $V_{l+2(k-1)L,m} \leftarrow \Lambda_{l,m}$
      end for
    end for
  end if
end for

The steps above are now repeated for the lower-bound vertices, replacing $\overline{(\cdot)}$ *by* $\underline{(\cdot)}$ *and* $\min(\cdot)$ *by* $\max(\cdot)$*, storing the result in* $V_{l+2(k-1)L+L,\cdot}$, $l = 1, \ldots, L$.

return V

Algorithm 1 is a recursive algorithm that produces a $N_h!2^{N_h} \times N_h$ matrix V, where each row is a vertex vector representing a possible combination of minimum and maximum consumption over the horizon N_h. The initial resource level is assumed to be zero.

Each call to the algorithm consists of three main operations. First, variables are initialized for storage; next, we iterate through the prediction horizon and compute extreme upper bounds for each sample by assuming that the absolute minimum power has been pulled from the unit at all previous samples. Thirdly, for each sample in turn, we assume that the considered sample is at the extreme upper bound and compute new upper and lower energy bounds according to this assumption. The computation is then repeated with the new set of constraints until the end of the horizon is reach. Finally equivalent computations are made for the lower-bound vertices essentially by exchanging minimisation and maximisation.

When completed, the algorithm has generated all potential $N_h!2^{N_h}$ vertices, many of which will be identical, but all of which will be on the boundary of the convex polytope.

For optimisation purposes, a half plane representation is more suitable than the vertex representation resulting from Algorithm 1. As mentioned, the conversion can be performed automatically by existing tools, but for a large number of vertices, it becomes computationally complex. An approximate (over-bounding) but fairly accurate conversion can be performed by considering rate and capacity constraints directly. Assume that a resource polytope is spanned by vertex vectors v_i, $i = 1..N_v$, and that the j-th element of v_i, $v_{i,j}$, is the coordinate for the j-th time step. Then capacity constraints are simply found as $\overline{E} = \max_i \sum_{j=1}^{N_h} v_{i,j}$ and $\underline{E} = \min_i \sum_{j=1}^{N_h} v_{i,j}$. Time varying consumption constraints are $\overline{P}_j = \max_i v_{i,j}$ and $\underline{P}_j = \min_i v_{i,j}$. We then have the half plane representation

$$\begin{bmatrix} I \\ -I \\ T \\ -T \end{bmatrix} P \leq \begin{bmatrix} \overline{w} \\ -\underline{w} \\ \overline{W}\overline{1} \\ -\underline{W}\overline{1} \end{bmatrix} \tag{3}$$

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

IV. Distribution

When controlling a large number of ICs, having a decision variable and a set of constraints for each is not computationally feasible (see [14] for more details).

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 4. 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 resource levels, P_i. From these, the

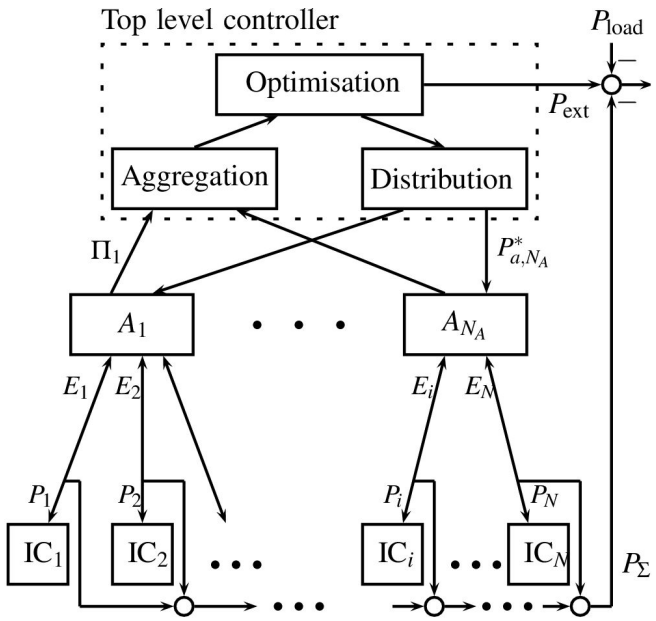

Fig. 4. Control architecture with grid-level control, aggregators and intelligent consumers. At each sample instant, consumer IC_i receives power P_i and returns information of its own energy storage level E_i.

aggregators compute the consumption constraints of each IC over the horizon. Using the vertex representation described in Section III, these constraints are then added to provide constraints provided to the top level.

Once the top level has received the resource polytopes it has full knowledge of the flexibility available in the system. This means that no "iterative price coordination [8]" is needed between the two upper layers of the hierarchy.

Given these constraints, 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 consumption sum P^*_Σ is then distributed between the aggregators, which further distribute it among the ICs. The distribution should be performed in a way that ensures that constraints can be met over the entire horizon.

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

$$\begin{aligned} \min_{P^*_{a,1},\dots,P^*_{a,N_A}} \quad & \sum_{i=1}^{N_A}\Big(\sum_{j=k}^{k+N_h-1} P_{a,i,j} - \sum_{j\in\mathscr{J}^i}\tilde{E}_{j,k}\Big)^2 \\ \text{s.t.} \quad & \Phi_{a,i}P^*_{a,i} \le \gamma_{a,i},\ i = 1,\dots,N_A \\ & \sum_{i=1}^{N_A} P^*_{a,i} = P^*_\Sigma \end{aligned} \tag{4}$$

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:

$$\begin{aligned} \min_{w^*_{m,k},m\in\mathscr{J}^j} \quad & \sum_{i\in\mathscr{J}^j}\Big(\sum_{j=k}^{k+N_h-1} P_{i,j} - \tilde{E}_{i,k}\Big)^2 \\ \text{s.t.} \quad & \Phi_i P^*_i \le \gamma_i,\ i\in\mathscr{J}^j \\ & \sum_{i\in\mathscr{J}^j} P^*_i = P^*_{a,i} \end{aligned} \tag{5}$$

where Φ_i, γ_i represent constraints for the individual ICs.

The ICs will then absorb their assigned consumption during a time step, after which the entire procedure is repeated for the new resource levels.

A. 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.

In the current implementation, the computational burden of performing the distribution at the top level or by an aggregator can fairly accurately be approximated by

$$\text{distribution load: } \beta_d 2^{N_h} N_d 2 + \beta_0, \tag{6}$$

and the vertex computation at an aggregator by

$$\text{vertex load: } \beta_v N_h! 2^{N_h} N_d, \tag{7}$$

where N_d is the number of associated consumers or aggregators in the layer directly below. Using cpu running times as provided by Matlab simulation studies indicate $\beta_v \approx 0.125\beta_0$ and $\beta_d \approx 0.015\beta_0$ on a standard PC.

B. Communication load

The biggest data flow results from the vertex tables being communicated upward in the hierarchy. Resource flow 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 resource 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.

C. Mid-ranging

At times when the disturbances are steady, the top level should attempt to keep the resources 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. Let $E_{ref,i}$ denote a reference level for consumer i. The desired consumption from the IC point of

view is then $\tilde{E}_{i,k} = E_{ref,i} - E_{i,k}$. By adding a term taking this into account, 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.

V. Simulation example

The simulation example considers a 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.

The ICs at the lowest level are modeled as simple power and energy constraint units governed by their own energy balance as explained in Section II.

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.

The second objective is to reduce the strain on the power plant by limiting the size of changes in production and the last objective of the top level controller is to bring the ICs close to their reference levels. With these three objectives the optimisation problem at the top level for a prediction horizon N_h is

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

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 $i = 1, 2, ..., N_h - 1$. The terminal weight Q_{N_h} is found by standard dual mode MPC methods [7]. The penalty on $P_{\text{plant}} - P_{\text{wind}}$ is included to improve closed loop performance.

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 it is used to supply the RCs.

	Refr. syst. [15]	Heat pump [16]
Maximum power	10 kW	4.3 kW
ΔT	1 K	3 K
Volume	5 m^3	12 m^3
Volm. heat cap.	1.9 $\frac{MJ}{m^3K}$ (ice)	2.4 $\frac{MJ}{m^3K}$ (concr.)
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

TABLE I

Parameters for the considered ICs.

Two types of ICs are considered, namely a domestic heat pump and a supermarket refrigeration system. The ICs are modeled as simple energy and power constraint units, so the internal dynamic is not modeled 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. Additional parameters are $\beta_e = 1$, $\beta_p = 2$, $\beta_r = 0.002$, $\beta_q = 0.1$ and $N_h = 4$, corresponding to

$$Q_{N_h} = \begin{bmatrix} 7.4 & -1.7 & 0 \\ -1.7 & 2.1 & 0 \\ 0 & 0 & 10 \end{bmatrix}.$$

Four aggregators are included in the simulations and each aggregator handles 400 heat pumps and 400 refrigeration systems. The time step is 15 minutes and P_{wind} is assumed known over the horizon N_h.

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, denoted the cautious method, where the interface between top- and aggregator level does not allow constraints to vary over the horizon N_h.

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.

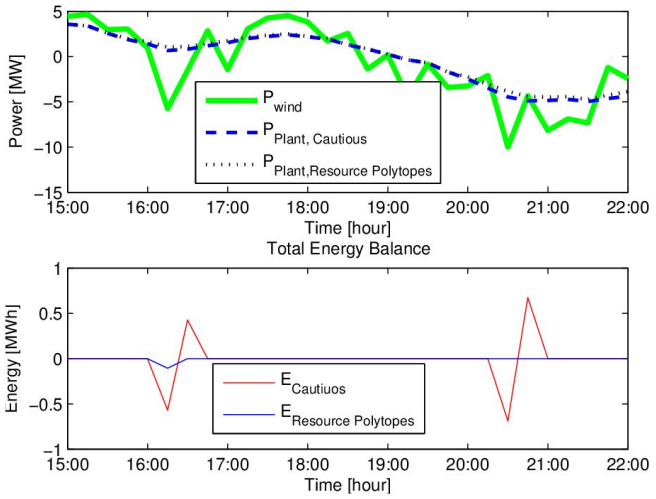

Fig. 5. P_{wind}, P_{plant} and total energy balance. The resource polytope method is noticeably better at maintaining the energy balance.

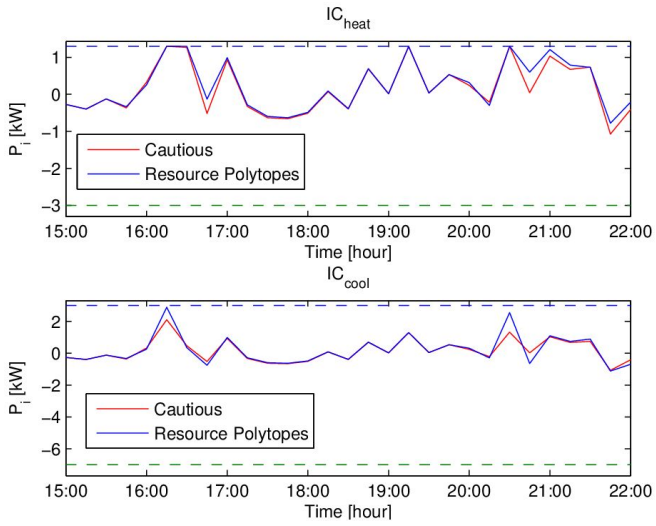

Fig. 6. Power consumption for an IC_{heat} and an IC_{cool} with resource polytopes and the cautious method.

Simulation results are give in Figure 5, which depicts P_{wind}, P_{plant}, $E_{Cautious}$ and $E_{Resource\ Polytopes}$. The two methods have to balance the exact same oscillations with the exact same resource available, but the resource polytope method is noticeably better at maintaining the energy balance. The reason for this can be seen in Figure 6. 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 needed, which means a better utilisation of the flexibility.

VI. Discussion

We have presented a novel way to represent resource storage capacity, which has the useful properties that:

- 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.

References

[1] P. Jarventausta, S. Repo, A. Rautiainen, and J. Partanen, "Smart grid power system control in distributed generation environment," *Annual Reviews in Control*, vol. 34, pp. 277–286, 2010.

[2] N. Wade, P. Taylor, P. Lang, and P. Jones, "Evaluating the benefits of an electrical energy storage system in a future smart grid," *Energy Policy*, vol. 38, pp. 7180–7188, 2010.

[3] A. Mohsenian-Rad, v.W.S. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *IEEE Transactions on Smart Grid*, vol. 1, no. 3, pp. 320–331, 2010.

[4] H. Banakar, C. Luo, and B. T. Ooi, "Impacts of wind power minute-to-minute variations on power system operation," *IEEE Transactions on Power Systems*, vol. 23, pp. 150–160, 2008.

[5] K. Moslehi and R. Kumar, "A reliability perspective of the smart grid," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 57–64, 2010.

[6] J. A. Rossiter, *Model-based predictive control*. CRC Press, 2003.

[7] J. M. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, 2002.

[8] R. Scattolini, "Architectures for distributed and hierarchical model predictive control a review," *Journal of Process Control*, vol. 19, pp. 723–731, 2009.

[9] B. Picasso, D. De Vito, R. Scattolini, and P. Colaneri, "An MPC approach to the design of two-layer hierarchical control systems," *Automatica*, vol. 46, no. 5, pp. 823–831, 2010.

[10] K. Trangbaek, J. Bendtsen, and J. Stoustrup, "Hierarchical model predictive control for resource distribution," in *Proc. of 49th IEEE Conference on Decision and Control*, Atlanta, Georgia, Dec. 2010.

[11] B. Grünbaum, *Convex Polytopes, 2nd edition*. Springer, 2003.

[12] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-Parametric Toolbox (MPT)," 2004. [Online]. Available: http://control.ee.ethz.ch/ mpt/

[13] P. Gritzmann and B. Sturmfels, "Minkowski addition of polytopes: computational complexity and applications to Gröbner bases," *SIAM J. Discrete Math.*, vol. 6, pp. 246–269, 1993.

[14] K. Trangbaek, M. Petersen, J. Bendtsen, and J. Stoustrup, "Predictive smart grid control with exact aggregated power constraints," *Submitted for publication.*

[15] J. Cai, "Control of refrigeration systems for trade-off between energy consumption and food quality loss," Ph.D. dissertation, Aalborg University, 2008.

[16] Dansk Energi and Energinet.dk, "Smart grid i danmark, bilagsrapport," 2010.