Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Exact constraint aggregation with applications to smart grids and resource distribution

Trangbæk, Klaus; Bendtsen, Jan Dimon

Published in:
Decision and Control (CDC), 2012 IEEE 51st Annual Conference on

DOI (link to publication from Publisher):
10.1109/CDC.2012.6426475

Publication date:
2012

Document Version
Early version, also known as pre-print

Link to publication from Aalborg University

Citation for published version (APA):
Trangbæk, K., & Bendtsen, J. D. (2012). Exact constraint aggregation with applications to smart grids and resource distribution. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on* (pp. 4181 - 4186). IEEE Press. https://doi.org/10.1109/CDC.2012.6426475

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: March 29, 2024

Exact constraint aggregation with applications to smart grids and resource distribution

Klaus Trangbaek Jan Bendtsen, *Member, IEEE*

***Abstract*— As hierarchical predictive control of large-scale distributed systems grow in complexity, it eventually becomes necessary to consider *aggregation* of lower-level units into larger groups of units that can be handled efficiently at higher levels in the hierarchy. When aggregating similar units in this manner, it is advantageous if the aggregation maintains a certain degree of genericity, since the higher-level algorithms can then be designed with a higher degree of *modularity*. To achieve this goal, however, it is not only necessary to examine aggregation of models of the underlying units, but also the accompanying constraints. Constraint sets for rate- and storage volume-constrained units can often be represented as polytopes in high-dimensional Euclidean space; unfortunately, adding such polytopic sets in higher dimension than 2 has so far been considered a combinatorial problem. In this paper, we present a novel method for computing such polytopic constraint sets for integrating units, which achieves a much lower computational complexity than previous results. The concept is demonstrated via simulations of a smart grid control scenario.**

I. Introduction

Many complex systems feature closely interacting subsystems that can be difficult to control with a single monolithic control structure [1]. One way of reducing complexity is by employing *hierarchical* control, where controllers manage only a limited set of subsystems, which in turn may manage subsystems of their own, etc. There are plenty of examples of hierarchical control systems, for instance traffic control [2]), wastewater treatment [3], drinking water networks [4] and even production planning [5], to name a few.

A natural aspect of hierarchical control is the issue of *aggregation* of lower-level units into larger groups of units that can be handled efficiently at higher levels in the hierarchy. When aggregating similar units in this manner, it is advantageous if the aggregation maintains a certain generic aspect, since the higher-level algorithms can then be designed with a higher degree of *modularity*, as this makes the overall solutions more scalable and better able to handle modification and replacement of individual subsystems [6].

Most work so far has focused on aggregation of the models of the underlying units. [5], for instance, develops an enhanced integrated model, which includes automatic acquisition of parameters. [7] considers a general threelevel hierarchical control problem and attempts to solve it by recasting the optimisation problem at each level as a multi-parametric programming problem. [8] considers hierarchical control of electricity consumers in a Smart Grid setting and introduces artificial aggregators as a way of dealing with computational complexity.

The authors are with the Department of Electronic Systems, Automation and Control, Aalborg University, Denmark; email: {dimon, ktr}@es.aau.dk

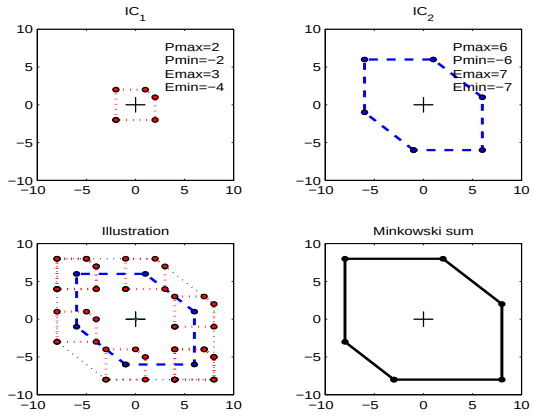


Fig. 1. Minkowski summation of two resource polytopes for a 2-sample horison. P_{max} and P_{min} refer to power limitations, while E_{max} and E_{min} are energy limitations for two "intelligent consumers" in a Smart Grid setting. The polytopes are represented by their vertices (dots) [9].

However, it appears that constraints generally tend to be difficult to handle. In [9], the authors proposed a geometric interpretation of constraint sets involved in distribution of power to controllable consumers in a similar Smart Grid setting. In that application, each consumer would be modelled as a simple integrator constrained with respect to the amount of power (short-term consumption) and energy (long-term consumption), and although the power and energy constraints can *not* immediately be added to describe several consumers, it was explained how these constraints could be mapped to convex polytopes in a way that would allow the polytopes to be added in a meaningful way. The polytopes were observed to be hypercubes with 'corners cut-off' due to the energy constraints. The polytopes could then be added to yield a larger polytope using the so-called *Minkowski sum* operation, as illustrated in Figure 1.

Unfortunately, computing the Minkowski sum of several resource polytopes is quite demanding, because the number of vertices grows combinatorically with the prediction horison. In [9], a procedure for computing the vertices in an ordered sequence was proposed, but it suffered from high complexity. The procedure gives the exact Minkowski sum in a vertex representation, but for optimisation purposes a half-plane representation is more appropriate. The conversion from vertex to half-plane representation is computationally

demanding as well, and a faster over-approximation was therefore suggested.

In this paper, we extend the work in [9] by formally proving that by representing the constraints via polytope *facets* rather than via vertices, the aforementioned combinatorial approach can be replaced by vector addition; furthermore, since the resulting polytopes are represented by half-planes directly, the optimisation can be performed with exact aggregated constraints without the need for costly conversions between representations.

The outline of the rest of the paper is as follows. Section II provides an overview of the hierarchical distribution problem considered in the paper, while Section III describes the geometric representation of flow and storage constraints. Section IV provides the main result of this paper, a formal proof that it is possible to compute Minkowski sums of resource polytopes directly using the aforementioned half-plane formulation. Section V compares the calculation time and accuracy of the proposed half-plane approach with the vertex-based method on a model-predictive control (MPC) simulation example from [9]. Finally, Section VI sums up the work.

Throughout the paper, $\mathbb{R}_+$ and $\mathbb{R}_-$ denote the positive and negative real axis, respectively, including the origin, i.e., $\mathbb{R}_+ = \{x \in \mathbb{R} | x \geq 0\}, \mathbb{R}_- = \{x \in \mathbb{R} | x \leq 0\}$. Similarly, $\mathbb{Z}_+$ and $\mathbb{Z}_-$ indicate the positive and negative integers including 0, respectively.

Furthermore, we make use of certain bit-wise binary operations between integers in $\mathbb{Z}_+$; $\wedge$ represents bitwise AND (e.g., $5 \wedge 6 \mathrel{\hat{=}} 0101_b \wedge 0110_b = 0100_b = 4$, where $(\cdot)_b$ indicates base 2), $\vee$ represents bitwise OR (e.g., $5 \vee 6 \mathrel{\hat{=}} 0101_b \vee 0110_b = 0111_b = 7$) and $\neg$ represents bitwise NOT (e.g., $\neg 13 \mathrel{\hat{=}} \neg 1101_b = 0010_b = 2$). We use $/$ as short-hand for bitwise AND-NOT, i.e., $i/j \mathrel{\hat{=}} i \wedge (\neg j)$.

II. Control hierarchy

We consider a control hierarchy of the form shown in Figure 2. The system to be controlled consists of a main system and a large number of parallel flexible units. The flexible units are not necessarily identical, but their effect on the main system is assumed to be only through the sum of the flows $f_1, f_2, \ldots, f_N$. The system is disturbed by a time-varying load f_{load}, and the control objective is to maintain a certain variable S_{bal} at a desired level, for instance 0. The control system can affect the main system either directly through f_{ext}, or indirectly by using the flexible units $\text{FU}_i, i = 1, \ldots, N$. Typically, the use of f_{ext} is associated with a high cost, however, so it is preferable to use the flexible units as much as possible in order to achieve the control objective.

In a smart grid setting, for instance, the main system would be a power grid and the control objective would be to maintain the energy balance between production and consumption, i.e., the integral of the difference between produced and consumed power. The external flow is power produced at central power plants, whereas the flexible units could be intelligent consumers, consumers that can adjust their power consumption, for instance by postponing the activation of a heat pump in a house (see [10] for a real-life example of this concept).

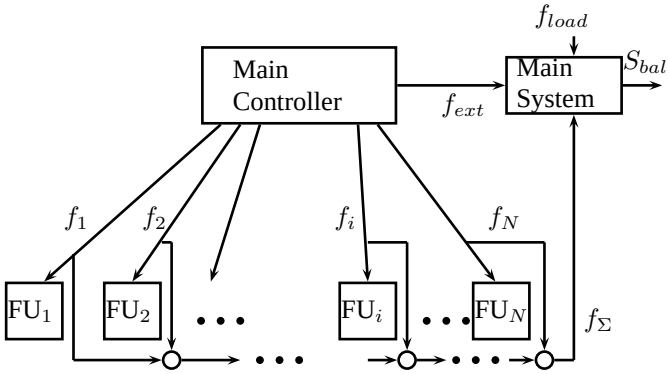


Fig. 2. Control architecture with main control and flexible units. [8]

The control system typically attempts to minimise some appropriate cost function, which tries to reach the design goal (e.g., $S_{bal} = 0$) while penalising the use of f_{ext} over time, subject to system constraints. In particular, $\underline{f}_i$ and $\overline{f}_i$ are constraints on the flows the i'th flexible unit is able to consume at (discrete) time t, whereas $\underline{S}_i$ and $\overline{S}_i$ are constraints on the i'th flexible unit's amount of storage. For simplicity, the storage in the i'th unit is modelled via the simple difference equation

$$S_i(t+1) = S_i(t) + f_i(t) \tag{1}$$

with the implicit assumption that the signals have been normalised and any mean values subtracted *a priori*.

On a finite horison, the above may be cast as a standard optimisation problem, see e.g., [8], which can be tackled using well-established methods, e.g., [11], [12], [13], or [1]. However, as the number of flexible units grows, it becomes increasingly difficult to keep track of the individual units' constraints.

In [8] it was proposed to use dedicated *aggregators* between the main controller and the flexible units to manage the units in smaller groups, allowing the main controller to solve an optimisation problem of lower order. This approach came at the expense of having to aggregate the individual unit constraints, however. As mentioned in the introduction, [9] provided a combinatorial approach to managing the constraints for all units assigned to each aggregator, along with an optimisation-based approach to dispatch of $f_1, f_2, \ldots, f_N$ once the constraints have been determined. [14] offers an alternative dispatch strategy in a similar setting.

Handling the aggregated constraints is the main topic of the rest of this paper. Although the optimal choice of what flexible units to aggregate remains an open question at the time of this writing, we shall simply assume that the assignment of units to aggregators has already been carried out.

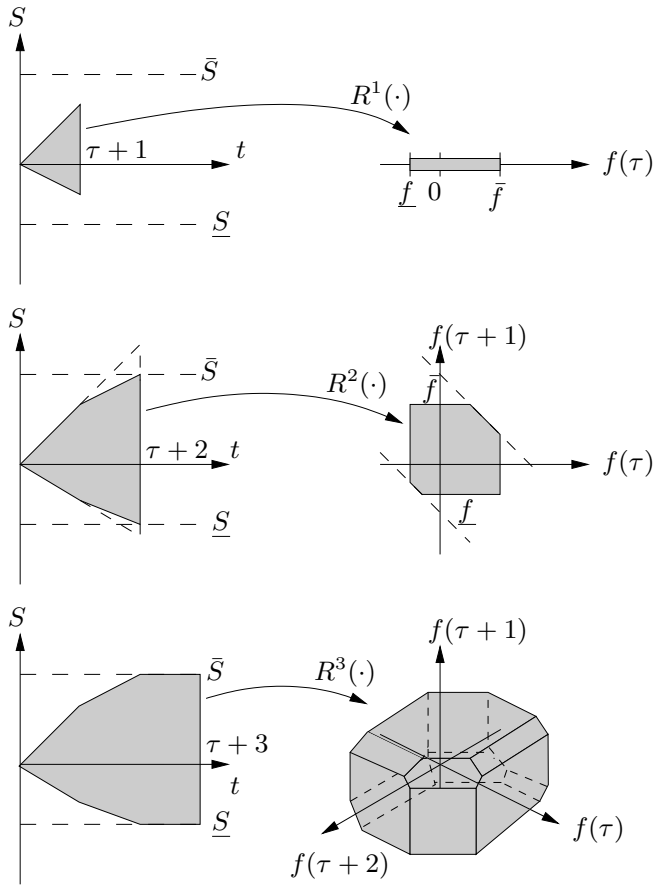


Fig. 3. Graphical representation of the mapping R^h for $h = 1$, $h = 2$ and $h = 3$. In the top figure, only the flow constraints $\underline{f}, \overline{f}$ are active, implying that any value in the interval $[\underline{f}, \overline{f}]$ can be chosen for $f(\tau)$. In the middle and lower figures, the storage constraints $\underline{S} \leq S(\tau+1) = S(\tau)+f(\tau) \leq \overline{S}$ and $\underline{S} \leq S(\tau+2) = S(\tau+1) + f(\tau+1) \leq \overline{S}$ become active, limiting the admissible flow profiles.

III. Representation of resources

In this section, we represent flow and storage constraints geometrically in a way that is amenable to aggregation; the principle is illustrated in Figure 3. At every time step τ, the constraints limit the potential consumption of flexible unit i; hence, at $t = \tau$ the unit can potentially consume any flow $f_i(\tau)$ within the flow limitations $\underline{f}_i, \overline{f}_i$. Doing so would bring the storage in flexible unit i to the level $S_i(\tau+1) = S_i(\tau)+f_i(\tau)$. At time $t = \tau+1$, the unit would again be able to consume any flow $f_i(\tau+1) \in [\underline{f}_i, \overline{f}_i]$, *unless* doing so would bring the storage $S_i(\tau+2) = S_i(\tau+1)+f_i(\tau+1)$ to a level where it would violate either of the constraints $\underline{S}_i, \overline{S}_i$, implying that the storage constraints cause the admissible flow at one time step to depend on the flow at the previous time step. As illustrated in the figure, over a finite horizon of length $h > 0$, the constraints on $f_i(t), t = \tau, \ldots, \tau+h$ can be represented by a polytope in $\mathbb{R}^h$. In fact, as long as the constraints $\underline{S}_i \leq S_i \leq \overline{S}_i$ 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 figure.

Without loss of generalisation, it can be assumed that $\overline{S} \in \mathbb{R}_+, \underline{S} \in \mathbb{R}_-, \overline{f} \in \mathbb{R}_+$ and $\underline{f} \in \mathbb{R}_-$. The convex polytope in $\mathbb{R}^h$ resulting from a set of flow and storage constraints can then be identified with the set-valued map $R^h : \mathbb{R}_+ \times \mathbb{R}_- \times \mathbb{R}_+ \times \mathbb{R}_- \to \mathcal{P}$, where $\mathcal{P}^h \subset \mathbb{R}^h$ denotes the set of bounded convex polytopes in h-dimensional Euclidean space. For the simple integrating units discussed in the previous section, we have

$$R^h(\overline{S}, \underline{S}, \overline{f}, \underline{f}) \triangleq$$
$$\{x \in \mathbb{R}^h | \underline{f} \leq x_j \leq \overline{f}, \underline{S} \leq \sum_{k=1}^{j} x_k \leq \overline{S},\ j = 1, \ldots, h\} \quad (2)$$

where x_j is the j'th coordinate of the vector x. We refer to this set as a *resource polytope*, since, for flexible unit i with constraints $\overline{S}_i, \underline{S}_i, \overline{f}_i, \underline{f}_i$, $R^h(\overline{S}_i, \underline{S}_i, \overline{f}_i, \underline{f}_i)$ is exactly the convex set of all admissible flow profiles over the horizon.

The *Minkowski sum* $\oplus$ of a number of convex polytopes, $\Pi_1, \ldots, \Pi_n \in \mathcal{P}^h$ is defined as the (polytopic) set of all sums of elements from the individual polygons, i.e. $\Pi_\Sigma = \Pi_1 \oplus \ldots \oplus \Pi_n = \{\sum_{i=1}^n x_i | x_i \in \Pi_i, i = 1, \ldots, n\}$. This is exactly what we need to compute for a set of several flexible units in order to provide the control level with a single constraint set; that is, given consumption capacities for a number of flexible units over a horizon, the total capacity will be given by the Minkowski sum of these. Note that this operation is *not* the same as simply adding flow rate and storage capacity limits, however.

Note further that the flow rate and storage capacity constraints can be written on the form

$$Fx \leq g$$

where F and g are matrices of appropriate dimensions; we refer to this form as a *half-plane representation*. [9] treated resource polytopes via their vertices (see Figure 1), which required a costly transformation into the half-plane representation for numerical optimisation purposes.

In the following sections, we will show that the sum can be computed directly in the half-plane representation, thereby avoiding the aforementioned transformation.

IV. Generalised resource polytopes

In order to achieve the desired result, a few tedious definitions are necessary. First, for the pair $h, z \in \mathbb{Z}_+$ satisfying $z < 2^h$, we define the *binary representation vector* of z as the unique vector $b_{h,z} \in \{0,1\}^h$ that satisfies

$$\sum_{i=1}^{h} b_{h,z,i}^T 2^{h-i} = z. \quad (3)$$

Note that, when considering the binary representation as a mapping between $\mathbb{Z}_+$ and $\{0,1\}^h$, it is an isomorphism; for instance, $b_{5,11} = \begin{bmatrix}0 & 1 & 0 & 1 & 1\end{bmatrix}^T \mapsto 11$ and $11 \mapsto b_{5,11}$.

Next, we define the *constraint matrix* $H_W^h \in \mathbb{R}^{(2^{h+1}-2)\times h}$ as follows:

$$H_W^h \triangleq \begin{bmatrix} b_{h,1}^T \\ b_{h,2}^T \\ \vdots \\ b_{h,2^h-1}^T \\ -b_{h,1}^T \\ -b_{h,2}^T \\ \vdots \\ -b_{h,2^h-1}^T \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \dots & 1 & 1 \\ 0 & 0 & \dots & 0 & -1 \\ 0 & 0 & \dots & -1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & -1 & \dots & -1 & -1 \end{bmatrix} \quad (4)$$

The purpose of H_W^h is to link constraints with the sample numbers at which they apply. Individual flow constraints $\underline{f}, \overline{f}$ may for example apply at individual time steps, which can be associated with binary representations of the form $\begin{bmatrix}0 & 1 & 0 & 0\end{bmatrix}$, $\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}$ etc., while storage constraints $\underline{S}, \overline{S}$ correspond to summing over a number of time steps, which can be associated with binary representations of the form $\begin{bmatrix}1 & 1 & 0 & 0\end{bmatrix}$, $\begin{bmatrix}1 & 1 & 1 & 1\end{bmatrix}$ etc. However, when adding two resource polytopes via Minkowski addition, constraints can appear which do not fit into these two patterns. Hence, H_W^h is structured so as to encode constraints for all possible sample combinations.

We are now ready to define the vector-valued function $W^h : \mathcal{P}^h \to \mathbb{R}_+^{2^{h+1}-2}$ as

$$W^h(\Pi) \triangleq \begin{bmatrix} d_1^+ & \cdots & d_{2^h-1}^+ & d_1^- & \cdots & d_{2^h-1}^- \end{bmatrix}^T \quad (5)$$

where Π is a bounded convex polytope in $\mathbb{R}^h$ and

$$d_i^+ \triangleq \max_{x\in\Pi} b_{h,i}^T x,\ i = 1, \dots, 2^h - 1 \quad (6)$$

$$d_i^- \triangleq -\min_{x\in\Pi} b_{h,i}^T x,\ i = 1, \dots, 2^h - 1 \quad (7)$$

We shall refer to the output of this mapping, i.e., $d = W^h(\Pi)$, as an *extended constraint vector*, since it defines the polytope

$$\Pi_W^h(d) = \{x \in \mathbb{R}^h | H_W^h x \le d\}. \quad (8)$$

Inspection shows that if this operation is applied to a resource polytope of the form (2), we end up with exactly the same polytope, i.e. $\Pi_W^h(W^h(R^h(\overline{S}_i, \underline{S}_i, \overline{f}_i, \underline{f}_r))) = R^h(\overline{S}_i, \underline{S}_i, \overline{f}_i, \underline{f}_i)$. However, since H_W^h is structured to encode all possible time step-wise summations over the horison, the representation in (8) is, in fact, a generalisation of the representation in (2).

Finally, let us define the semigroup (under vector addition)

$$\mathcal{W}^h \triangleq \left\{ d = \begin{bmatrix} d_1^+ \\ \vdots \\ d_{2^h-1}^+ \\ d_1^- \\ \vdots \\ d_{2^h-1}^- \end{bmatrix} \right\} \subset \mathbb{R}_+^{2^{h+1}-2} \quad (9)$$

where the entries $d_i^+, d_i^-, i = 1, \dots, 2^h - 1$ must satisfy

$$d_0^+ = d_0^- \triangleq 0 \quad (10)$$

$$d_i^+ + d_j^- \ge d_{i/j}^+ + d_{j/i}^- \quad (11)$$

$$d_i^+ + d_j^+ \ge d_{i\wedge j}^+ + d_{j\vee i}^+ \quad (12)$$

$$d_i^- + d_j^- \ge d_{i\wedge j}^- + d_{j\vee i}^- \quad (13)$$

$$\exists x \in \Pi_W^h(d) : b_{h,i}^T x = d_i^+ \quad (14)$$

$$\exists x \in \Pi_W^h(d) : -b_{h,i}^T x = d_i^- \quad (15)$$

for all $1 \le i, j \le 2^h - 1$ (index 0 may result from some of the bitwise operations). Now, it turns out that if a polytope Π is a resource polytope, the extended constraint vector resulting from the operation $W^h(\Pi)$ belongs to this semigroup:

Theorem 1: Let $\overline{S}, \underline{S}, \overline{f}, \underline{f}$ be flow and storage constraints as given in (2); then

$$W^h(R^h(\overline{S}, \underline{S}, \overline{f}, \underline{f})) \in \mathcal{W}^h \quad (16)$$

Proof: Let $\Pi = R^h(\overline{S}, \underline{S}, \overline{f}, \underline{f})$ and $d = W^h(\Pi)$; (14)-(15) follow trivially from the definition of W^h.

Note that $d_i^+ = \max_{x\in\Pi} b_{h,i}^T x$ is obtained by maximising x_k when $b_{h,i,k}^T = 1$ and minimising when $b_{h,i,k}^T = 0$, and, importantly, the order of these optimisations does not matter.

For $1 \le i, j < 2^h$, pick an $\tilde{x}$ such that $b_{h,i\wedge j}^T \tilde{x} = d_{i\wedge j}^+$. Then, for any a such that $(i \wedge j) \vee a = a$, we have

$$d_a^+ = d_{i\wedge j}^+ + \max_{x\in\tilde{P}} b_{h,a/(i\wedge j)}^T x \quad (17)$$

where $\tilde{P} = \{x \in \Pi | b_{h,i\wedge j,k}^T x_k = b_{h,i\wedge j,k}^T \tilde{x}_k \forall k\}$. Observe

$$\begin{aligned} \max_{x\in\tilde{P}} b_{h,i/(i\wedge j)}^T x \ &+ \ \max_{x\in\tilde{P}} b_{h,j/(i\wedge j)}^T x \\ &= \max_{x\in\tilde{P}} b_{h,i/j}^T x + \max_{x\in\tilde{P}} b_{h,j/i}^T x \\ &\ge \max_{x\in\tilde{P}} b_{h,(j/i)\vee(j/i)}^T x \\ &= \max_{x\in\tilde{P}} b_{h,(i\vee j)/(i\wedge j)}^T x \end{aligned}$$

Applying (17) with $a = i$ and $a = j$ to the left and with $a = i \vee j$ to the right, we have

$$d_i^+ - d_{i\wedge j}^+ + d_j^+ - d_{i\wedge j}^+ \ge d_{i\wedge j}^+ - d_{i\wedge j}^+ + d_{i\vee j}^+ - d_{i\wedge j}^+$$

i.e., (12). Inequality (13) follows in the same manner.

Since $(i/j) \wedge (j/i) = 0\ \forall i, j$, (11) follows from

$$\begin{aligned} \max_{x\in P} b_{h,i/j}^T x + \max_{x\in P} -b_{h,j/i}^T x &= \max_{x\in P} (b_{h,i/j}^T - b_{h,j/i}^T) x \\ &= \max_{x\in P} (b_{h,i}^T - b_{h,j}^T) x \\ &\le \max_{x\in P} b_{h,i}^T x + \max_{x\in P} (-b_{h,j}^T x) \end{aligned}$$

Thus, $W^h(\Pi) = W^h(R^h(\overline{S}, \underline{S}, \overline{f}, \underline{f})) \in \mathcal{W}^h$. ■

As noted above, the extended representation does not affect the resource polytope itself.

We now point out some useful properties of this representation; together, Lemmas 1 and 2 imply that a facet resulting from the Minkowski summation of two facets or ridges in the extended representation will be a facet in the sum polytope.

Lemma 1: Let the integers $0 \leq i, j, m, n < 2^h$ be such that $i \wedge j = m \wedge n = 0$. If there exist a $q \in \mathbb{R}^h$ and $c \in \mathbb{R}$ such that

$$\{v+\hat{v} \in \mathbb{R}^h : \ b_{h,i}^T v = d_i, \ b_{h,j}^T v = d_j, \ b_{h,m}^T \hat{v} = d_m, \ b_{h,n}^T \hat{v} = d_n\} = \{w \in \mathbb{R}^h : q^T w = c\} \quad (18)$$

where $d_i, d_j, d_m, d_n \in \mathbb{R}$ are fixed numbers and $b_{h,i}, b_{h,j}, b_{h,m}$ and $b_{h,n}$ are binary representation vectors, then there exist a scaling factor $\gamma \in \mathbb{R}$ and an integer $l \in [0, 2^h)$ such that $q = \gamma b_{h,l}$.

Lemma 2: Let $d = \begin{bmatrix} d^+ \\ d^- \end{bmatrix} \in \mathcal{W}^h$. If $i/j > 0$ and $j/i > 0$, the intersection of $\Pi_W^h(d)$ and either pair of hyperplanes $b_{h,i}^T x = d_i^+$, $b_{h,j}^T x = d_j^+$ or $-b_{h,i}^T x = d_i^-$, $-b_{h,j}^T x = d_j^-$ is of dimension lower than $h-2$ or empty.

The proofs are omitted due to space constraints.

The key observation to be made here is that because the extended constraint vectors belong to a semigroup that is closed under vector addition, it becomes possible to perform the Minkowski summation as a simple vector addition, as will be shown in the following.

Theorem 2: Let $d, \hat{d} \in \mathcal{W}^h$; then

$$\Pi_W^h(d) \oplus \Pi_W^h(\hat{d}) = \Pi_W^h\left(d + \hat{d}\right). \quad (19)$$

Proof: (19) holds trivially for $h = 1$, so in the following we shall assume that $h \geq 2$.

The facets of $\Pi_W^h(d) \oplus \Pi_W^h(\hat{d})$ are formed by the Minkowski summation of facets and ridges of $\Pi_W^h(d)$ and $\Pi_W^h(\hat{d})$. Let the entries in $d = \begin{bmatrix} d^+ \\ d^- \end{bmatrix}$ be indexed as in (9) and let $F = \{v \in \Pi_W^h(d) : b_{h,i}^T v = d_i, \ b_{h,j}^T v = d_j\}, i > j$, be a ridge or facet (if $j = 0$) of $\Pi_W^h(d)$.

Since F is a ridge or facet of a polytope in $\mathbb{R}^h$, it must be of dimension $h-2$ or $h-1$, respectively. From Lemma 2 we then know that we cannot have either $i/j > 0$ or $j/i > 0$; hence we must have $j/i = 0$, and thus $F = \{v \in \Pi_W^h(d) : b_{h,i/j}^T v = d_i - d_j, b_{h,j}^T v = d_j\}$.

Similarly, let $\hat{d} = \begin{bmatrix} \hat{d}^+ \\ \hat{d}^- \end{bmatrix}$ be indexed as in (9) and, with $m > n$, let $\hat{F} = \{\hat{d} \in \Pi_W^h(\hat{d}) : b_{h,m}^T \hat{v} = d_m, b_{h,n}^T \hat{v} = d_n\}$ be a ridge or facet of $\Pi_W^h(\hat{d})$. In a similar manner as before, we see that $\hat{F} = \{\hat{v} \in \Pi_W^h(\hat{d}) : b_{h,m/n}^T \hat{v} = d_m - d_n, b_{h,n}^T \hat{v} = d_n\}$.

We can now apply Lemma 1 to see that there exist l and c such that $F \oplus \hat{F} = \{v+\hat{v}\} \subseteq \{w : b_{h,l}^T w = c\}$. This claim clearly holds for any sum of corresponding facets and ridges of $\Pi_W^h(d)$ and $\Pi_W^h(\hat{d})$, and, crucially, the corresponding $b_{h,l}^T$ forms a row in H_W^h in (8). Consequently, there exists a $\delta = \begin{bmatrix} \delta^+ \\ \delta^- \end{bmatrix} \in \mathcal{W}^h$ such that $\Pi_W^h(d) \oplus \Pi_W^h(\hat{d}) = \Pi_W^h(\delta)$. Specifically, we can choose

$$\begin{aligned} \delta_i^+ &= \max_{v \in \Pi_W^h(d), \hat{v} \in \Pi_W^h(\hat{d})} b_{h,i}^T (v + \hat{v}) \\ &= \max_{v \in \Pi_W^h(d)} b_{h,i}^T v + \max_{\hat{v} \in \Pi_W^h(\hat{d})} b_{h,i}^T \hat{v} \\ &= d_i^+ + \hat{d}_i^+ \end{aligned}$$

where the last equality follows from (14). Similarly, we can choose $\delta_i^- = d_i^- + \hat{d}_i^-$, and hence $\delta = d + \hat{d}$. ∎

Theorem 1 and 2 allow the generalised constraint vector to be computed separately for each resource (in a distributed manner, if need be), and the total resource polytope of the form (8) can then be used for optimisation purposes. Any flow profile that satisfies the polytopic constraints is guaranteed to be possible to distribute among the aggregated flexible units.

	Refrig. system	Heat pump
$\overline{E}_i = \overline{S}_i$	2.6 kWh	8 kWh
$\underline{E}_i = \underline{S}_i$	0 kWh	0 kWh
$\overline{P}_i = T_s^{-1}\overline{f}_i$	3 kW	1.3 kW
$\underline{P}_i = T_s^{-1}\underline{f}_i$	-7 kW	-3 kW

TABLE I
PARAMETERS FOR THE CONSIDERED ICS.

V. SMART GRID SIMULATION EXAMPLE

In [9], a model predictive control scheme for smart grids was presented and applied to a simulation example involving a grid consisting of wind turbines, heat pumps, cooling facilities and a power plant. The wind turbines exhibit fast fluctuations, which must be balanced by the heat pumps, cooling facilities, and, if necessary, by the power plant. The energy balance is modelled as

$$E(t+1) = E(t) + T_s(P_{plant}(t) - P_{wind}(t) - P_\Sigma(t))$$

where E denotes energy (storage) and P denotes power (flow). The sample time T_s is 15 minutes.

At every sample, the control system solves the problem

$$\min_{P_{\Sigma,t}, P_{plant,t}, t=\tau,\ldots,t+\tau} \sum_{t=\tau}^{t+\tau} y(t)^T Q y(t)$$

where

$$y(t) = \begin{bmatrix} E(t) \\ P_{plant}(t) - P_{wind}(t) \\ P_{plant}(t) - P_{plant}(t-1) \end{bmatrix}, \ Q = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 2.0 \end{bmatrix}$$

Four aggregators are included in the simulations and each aggregator handles 400 heat pumps and 400 refrigeration systems. Flow (power) and storage (energy) constraints for the two types of units can be found in Table I. Refer to [9] for further details.

We repeat the simulation to compare three methods: the vertex method from [9], the half-plane approach presented here and a low-complexity, but conservative, "box" approach. In the box approach, the resource polytopes are inner-approximated by boxes defined by the two vertices found by respectively maximising and minisiming the power at each sample beginning with the first. This means that the early samples have the most flexibility, while the storage constraints will limit the latter ones. This is expected to give the best performance in the receding horizon scheme. This approach has very low complexity since only two vertices must be computed for each resource, and the Minkowski sum of boxes is simply found by adding these vertices.

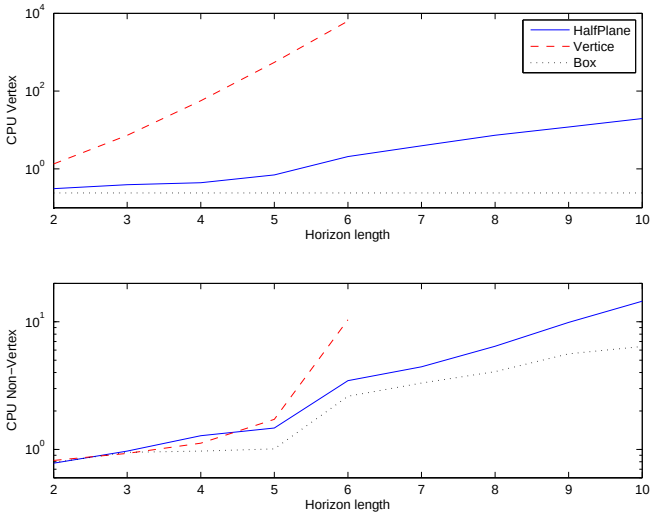


Fig. 4. Computational complexity as a function of optimisation horizon. Top: vertex/half-plane computation. Bottom: Remaining optimisation routines.

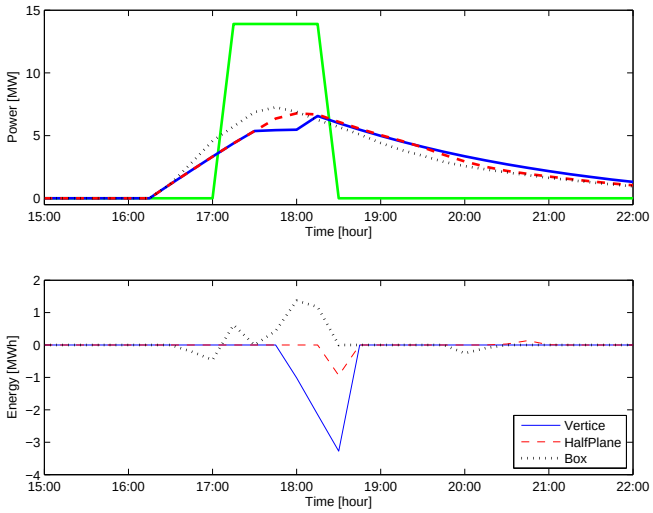


Fig. 5. Simulation of steps up and down in the wind disturbance. Top: The power plant contribution. Bottom: Resulting energy balance.

Figure 4 shows the computational time for the three schemes as the control horizon of the MPC is increased. For the vertex method, the time needed for computing vertices increases drastically, making it impossible to work with long horizons. The performance has also been investigated, and the half-plane approach gives slightly better performance than the vertex method. They are both significantly better than the box method.

The reason for the worse performance of the vertex method is that, even though the computed vertices represent the correct Minkowski sum, the conversion to half-planes needed in the optimisation is an (over-)approximation. This is illustrated by the simulation in Figure 5, where the real wind data have been replaced by a large step. The half-plane method is able to fully utilise the flexible resources, thereby having a smooth output from the power plant, while keeping the energy imbalance small. The conservative box method has to use the power plant more and still gives a worse balance. The vertex method does fine until an infeasible trajectory is produced in the optimisation, resulting in problems keeping the balance.

VI. Conclusions

This paper considered hierarchical predictive control of large-scale distributed systems, in particular systems consisting of flexible units operating in parallel toward a common goal, subject to integrating dynamics and rate- and storage volume constraints. It was shown how such constraints could be represented as polytopes in high-dimensional Euclidean space, so-called resource polytopes. Furthermore, a novel method for aggregating such polytopic constraint sets for integrating units, which achieves a much lower computational complexity than previous results, was proposed. The concept was demonstrated via simulations of a smart grid control scenario, but can potentially be used for entirely different applications as well, for instance warehouse logistics, water distribution etc.

References

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

[2] A. Hegyi, B. D. Schutter, and J. Hellendoorn, "Optimal coordination of variable speed limits to suppress shock waves," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 102–112, March 2005.

[3] M. A. Brdys, M. Grochowski, T. Gminski, K. Konarczak, and K. Drewa, "Hierarchical predictive control of integrated wastewater treatment systems," *Control Engineering Practice*, vol. 16, pp. 751–767, June 2008.

[4] C. Ocampo-Martinez, D. Barcelli, V. Puig, and A. Bemporad, "Hierarchical and decentralised model predictive control of drinking water networks: Application to barcelona case study," *IET Control Theory and Applications*, vol. 6, pp. 62–71, January 2012.

[5] S. Tzafestas and G. Kapsiotis, "Unified hierarchical control algorithm for constrained non-linear multi-delay interconnected systems: application to production planning," *Control, theory and advanced technology*, vol. 10, pp. 1737–1761, November 1995.

[6] K. Edlund, J. D. Bendtsen, and J. B. Jørgensen, "Hierarchical model-based predictive control of a power plant portfolio," *Control Engineering Practice*, vol. 19, pp. 1126–1136, 2011.

[7] N. Faisca, K. Kouramas, and N. Pistikopoulos, "A multi-parametric optimization strategy for multilevel hierarchical control problems," in *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, July 2008.

[8] 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.

[9] K. Trangbaek, M. Petersen, J. Bendtsen, and J. Stoustrup, "Exact power constraints in smart grid control," in *Proc. of 50th IEEE Conference on Decision and Control*, Orlando, Florida, Dec. 2011.

[10] P. Andersen, T. S. Pedersen, and K. M. Nielsen, "Observer based model identification of heat pumps in a smart grid," in *Proc. of Multiconf. Systems and Control*, Dubrovnik, Croatia, Oct. 2012.

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

[12] 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.

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

[14] M. Petersen, J. Bendtsen, and J. Stoustrup, "Optimal dispatch strategy for the agile virtual power plant imbalance compensation problem," in *Proc. of American Control Conference*, Montreal, Canada, June 2012.