A distributed-in-time NMPC-based coordination mechanism for resource sharing problems

Mohamed Yacine Lamoudi, Mazen Alamir and Patrick Béguery

Abstract In this chapter, a hierarchical model predictive control framework is presented for a network of subsystems that are submitted to general resource sharing constraints. The method is based on a primal decomposition of the centralized open-loop optimization problem over several subsystems. A coordinator is responsible of adjusting the parameters of the problems that are to be solved by each subsystem. A distributed-in-time feature is combined with a bundle method at the coordination layer that enables to enhance the performance and the real-time implementability of the proposed approach. The scheme performance is assessed using a real-life energy coordination problem in a building involving 20 zones that have to share a limited amount of total power.

1 Introduction

The principle of Distributed Model Predictive Control (DMPC) [3, 10] is to design local predictive controllers responsible of local decision making. In iterative schemes, these local controllers have to come up with an agreement throughout "negotiation" iterations in order to recover the solution (or to achieve a relevant sub-optimal solution) of the original centralized problem. DMPC appears to be an interesting approach for large scale systems. Indeed, for such systems the centralized optimization problem is generally very hard (or even impossible) to solve given restrictions on the computational time and computational/communication resources. Moreover, the non scalability of the centralized solution is not the only reason of

Mohamed Yacine Lamoudi and Patrick Béguery
Schneider-Electric Industries -Strategy & Innovation-. 37, quai Paul Louis Merlin-38000 Grenoble-France. e-mail: {mohamed-yacine.lamoudi,patrick.beguery}@fr.schneider-electric.com

Mazen Alamir
CNRS-University of Grenoble, Gipsa-lab, BP 46, Domaine Universitaire, 38400 Saint Martin d'Hères, France. e-mail: Mazen.alamir@gipsa-lab.grenoble-inp.fr

making the distributed approach interesting, indeed centralizing the decision process on one physical controller is generally unsafe because any failure affects the whole system and leads to non-modular schemes which may be impractical. Several DMPC approaches have been proposed in the recent literature. They are generally based either on a primal or a dual decomposition of the centralized optimization problem [11]. As pointed out by [4], the convergence of the distributed solution to the centralized problem generally requires an important number of iterations even for simple case studies. This may be prohibitive for real-time implementability issues and for communication network saturation concern. Consequently, despite the apparent similarity of the many distributed MPC-based solutions, a relevant comparison should focus, among other indicators, on the specific issues of communication and computation time saving.

In order to deal with this issue, a general framework based on the combination of an efficient distributed optimization technique (*disaggregated bundle method*) and an original *distributed-in-time computation* feature is presented. The technique is based on a primal decomposition of the centralized optimization problem and leads to a hierarchical structure.

The original works leading to the scheme proposed here address the issue of energy management in buildings with a high number of zones and limited communication rate. As far as the building case is concerned, the results have been reported in [6, 9]. In the present chapter, a general setting of the underlying framework is proposed and some properties of the resulting algorithms are analyzed in deeper details.

The chapter is organized as follows: section 2 presents succinctly the kind of problems that can be addressed by the proposed approach. Section 3 gives a step by step description of the framework. In section 4, a brief discussion of the theoretical rationale that underlines the distributed-in-time iterations is given. In section 5, a real-life example consisting of a 20 zones building submitted to global power restriction and time varying energy tariffs is analyzed using different algorithm parameters. Finally, section 6 concludes the chapter and describes future research directions.

2 Boundary conditions

In this section a general description of the class of problems targeted by the proposed scheme is given. Roughly speaking, a network of subsystems sharing limited resources is considered. This is detailed in the present section.

2.1 Subsystems

Consider a set of N_s subsystems that are dynamically uncoupled[1]. Each subsystem $\ell \in \mathscr{S} := \{1,\dots,N_s\}$ is governed by the following general nonlinear dynamic:

$$x_{\ell,k+1} = f(x_{\ell,k}, u_{\ell,k}) \tag{1}$$

where:

- $x_{\ell,k} \in \mathbb{R}^{n_\ell^x}$ is the state vector of the subsystem ℓ at instant k,
- $u_{\ell,k} \in \mathbb{R}^{n_\ell^u}$ is the input vector of the subsystem ℓ at instant k,

In the sequel, the following notation is extensively used: given a vector quantity $v_\ell \in \mathbb{R}^{n_v}$ related to subsystem ℓ, the boldfaced vector $\mathbf{v}_{\ell,k}$ represents the future profiles v_ℓ over the prediction horizon of length N_p, namely:

$$\mathbf{v}_{\ell,k} := [v_{\ell,k}^T, \dots, v_{\ell,k+N_p-1}^T]^T \tag{2}$$

when no ambiguity results, the time index k is dropped and the predicted profile $\mathbf{v}_{\ell,k}$ is simply denoted by $\mathbf{v}_\ell$. Moreover, all the quantities indexed by $\cdot_{\ell,k}$ refers to quantities related to subsystem ℓ at instant k.

Let us now define for each subsystem $\ell \in \mathscr{S}$ the resource vector $r_{\ell,k} \in \mathbb{R}^{n_\ell^s}$ and its future profile $\mathbf{r}_{\ell,k}$. The resource limitation for subsystems $\ell \in \mathscr{S}$ over the prediction horizon are expressed through the following local constraints inequality:

$$\forall \ell \in \mathscr{S}, \quad \mathbf{h}_\ell(\mathbf{x}_{\ell,k}, \mathbf{u}_{\ell,k}, \mathbf{r}_{\ell,k}) \leq 0 \tag{3}$$

It is assumed that each subsystem $\ell \in \mathscr{S}$ is locally controlled by a model predictive controller referred to hereafter by $\mathrm{MPC}_\ell(\mathbf{r}_{\ell,k})$ where $\mathbf{r}_{\ell,k}$ is the available resource profile allocated to subsystem ℓ over the prediction horizon. Therefore, at each sampling time k and given a prescribed available resource profile $\mathbf{r}_{\ell,k}$ over the prediction horizon, the system ℓ has to solve the following optimization problem:

$$\mathrm{MPC}_\ell(\mathbf{r}_{\ell,k}): \underset{\mathbf{x}_{\ell,k}\in\mathscr{X}_{\ell,k}, \mathbf{u}_{\ell,k}\in\mathscr{U}_{\ell,k}}{\text{Minimize}} \quad L_\ell(\mathbf{x}_{\ell,k}, \mathbf{u}_{\ell,k}) \tag{4a}$$

$$\text{Subject to:} \quad \mathbf{h}_\ell(\mathbf{x}_{\ell,k}, \mathbf{u}_{\ell,k}, \mathbf{r}_{\ell,k}) \leq 0 \tag{4b}$$

where $L_\ell(\mathbf{x}_{\ell,k}, \mathbf{u}_{\ell,k}) \geq 0$ is the cost function related to the subsystem ℓ and the domains $\mathscr{X}_{\ell,k}$ and $\mathscr{U}_{\ell,k}$ denote respectively state and input constraints that are possibly time-varying.

Once the problem (4) is solved at time k, the optimal predicted input and state trajectories $\mathbf{u}_{\ell,k}^\star$ and $\mathbf{x}_{\ell,k}^\star$ are obtained. The first component $u_{\ell,k}^\star$ of $\mathbf{u}_{\ell,k}^\star$ is applied to the subsystem during the time interval $[k, k+1]$. This operation is renewed at the next

[1] Potential coupling can be handled through dedicated observers as it is shown for instance in [6, 9] regarding the building energy management context

instant $k+1$ based on new measurement or estimation of the state $x_{\ell,k+1}$ and a new available resource profile $\mathbf{r}_{\ell,k+1}$ and so on. The next section details how the profile $\mathbf{r}_{\ell,k}$ is managed by a coordination level.

2.2 Resource sharing

Consider now a global constraint on the available resource for the whole network of subsystems. Assume that it is expressed through the following inequality:

$$\mathbf{H}(\mathbf{r}_{1,k},\ldots,\mathbf{r}_{N_s,k}) \leq 0 \tag{5}$$

The centralized formulation of the optimization problem becomes:

$$\underset{\{\mathbf{x}_{\ell,k}\in\mathscr{X}_{\ell,k},\mathbf{u}_{\ell,k}\in\mathscr{U}_{\ell,k}\}_{\ell\in\mathscr{S}}}{\text{Minimize}} \quad \sum_{\ell\in\mathscr{S}} L_\ell(\mathbf{x}_{\ell,k},\mathbf{u}_{\ell,k}) \tag{6a}$$

Subject to:

$$\mathbf{H}(\mathbf{r}_{1,k},\ldots,\mathbf{r}_{N_s,k}) \leq 0 \tag{6b}$$

$$\mathbf{h}_\ell(\mathbf{x}_{\ell,k},\mathbf{u}_{\ell,k},\mathbf{r}_{\ell,k}) \leq 0, \quad \forall \ell \in \mathscr{S} \tag{6c}$$

When the number of subsystems N_s is high, the centralized optimization problem (6) becomes a large scale optimization problem which may become intractable. This is why a coordination layer is introduced in order to keep the global constraint (5) satisfied despite the fact that each subsystem still solves its own optimization problem. The coordinator is responsible of adjusting the available resources profiles $\{\mathbf{r}_{\ell,k}\}_{\ell\in\mathscr{S}}$ of each subsystem $\ell \in \mathscr{S}$. The communication is supposed take place between each subsystem and the coordinator but is not available between the subsystems, leading to a classical hierarchical scheme. The kind of information exchanged between these entities as well as an algorithmic description of the optimization taking place at the coordination layer are presented in the next section.

3 Description of the approach

In order to coordinate the subsystems, an efficient iterative procedure is designed in this section. Throughout the negotiation iterations taking place between the coordinator and subsystems, the coordinator refines the optimal resource allocation in order to achieve a suitable repartition of the resources between the subsystems. To this end, let us first decompose the centralized problem (6) and introduce some notations and assumptions.

3.1 Problem Decomposition

Let $J_{\ell,k}(\mathbf{r}_{\ell,k})$ denotes the achieved optimal value for a given resource allocation $\mathbf{r}_{\ell,k}$:

$$J_{\ell,k}(\mathbf{r}_{\ell,k}) := L_\ell(\mathbf{x}^\star_{\ell,k}, \mathbf{u}^\star_{\ell,k}) \tag{7}$$

Moreover, consider that the (sub-)gradient $\mathbf{g}_{\ell,k}(\mathbf{r}_{\ell,k})$ of $J_{\ell,k}$ is available, namely:

$$\mathbf{g}_{\ell,k}(\mathbf{r}_{\ell,k}) \in \partial J_{\ell,k}(\mathbf{r}_{\ell,k}) \tag{8}$$

where $\partial J_{\ell,k}(\mathbf{r}_{\ell,k})$ is the subdifferential set of the function $J_{\ell,k}$ at $\mathbf{r}_{\ell,k}$.

Assumption 1 *For all ℓ and all k, the function $J_{\ell,k}(\mathbf{r}_{\ell,k})$ is convex and its subgradient $\mathbf{g}_{\ell,k}(\mathbf{r}_{\ell,k}) \in \partial J_{\ell,k}(\mathbf{r}_{\ell,k})$ is available.*

Note also that the local problems (4) are not always feasible. However, let us consider that $\mathbf{r}_{\ell,k} \in \mathscr{F}_{\ell,k}$ is a sufficient condition to ensure its feasibility and that $\mathscr{F}_{\ell,k}$ is available for all the subsystems $\ell \in \mathscr{S}$.

Assumption 2 *The local sub-problems* (4) *are feasible $\forall \mathbf{r}_{\ell,k} \in \mathscr{F}_{\ell,k}$. Moreover, the domains $\mathscr{F}_{\ell,k}, \forall \ell \in \mathscr{S}$ are known by the coordinator.*

Based on the previous definitions and assumptions, one can rewrite the optimization problem (6) in the following form:

$$\underset{\{\mathbf{r}_{\ell,k} \in \mathscr{F}_{\ell,k}\}_{\ell \in \mathscr{S}}}{\text{Minimize}} \quad J(\mathbf{r}_{\cdot,k}) := \sum_{\ell \in \mathscr{S}} J_{\ell,k}(\mathbf{r}_{\ell,k}) \quad \text{S.t.} \quad \mathbf{H}(\mathbf{r}_{1,k}, \ldots, \mathbf{r}_{N_s,k}) \leq 0 \tag{9}$$

or more shortly

$$\underset{\mathbf{r}_{\cdot,k} \in \mathscr{D}}{\text{Minimize}} \quad J(\mathbf{r}_{\cdot,k}) \tag{10}$$

where $\mathbf{r}_{\cdot,k} := [\mathbf{r}^T_{1,k}, \ldots, \mathbf{r}^T_{N_s,k}]^T$. The notation $\mathbf{r}_{\cdot,k} \in \mathscr{D}$ means the fulfillment of both the feasibility conditions $\{\mathbf{r}_{\ell,k} \in \mathscr{F}_{\ell,k}\}_{\ell \in \mathscr{S}}$ and global resources constraints (5).

The optimization problem (10) is called the *master problem* and its resolution is performed at the coordination layer as explained in the following section.

3.2 Solving the master problem - disaggregated bundle method

In order to solve the master problem, the coordinator successively approximates the functions $\{J_{\ell,k}\}_{\ell \in \mathscr{S}}$. Indeed, let us remind that the coordinator does not have any information regarding the dynamics and current states of the subsystems. However, the coordinator is supposed to be able to affect any resource $\mathbf{r}_{\ell,k}$ to the subsystems $\ell \in \mathscr{S}$ and to request the values of the objective function of each subsystem ℓ as well as their corresponding sub-gradients (assumption 1). Bundle method for this kind of

problems appears to be an interesting choice, it relies on iteratively approximating the objective function $J = \sum_{\ell \in \mathscr{S}} J_{\ell,k}$ by a so called *cutting plane model*. Since the objective function J is separable, an individual *cutting plane model* $\check{J}_{\ell,k}$ of each function $J_{\ell,k}$ is built up. In this section, a presentation of the bundle method is given, the reader may refer to [5] for more details. For the sake of conciseness, the time index k is dropped in the notations (e.g : $\mathbf{r}_\ell \equiv \mathbf{r}_{\ell,k}$) since the procedure described here takes place at a given instant k.

In disaggregated bundle method, individual cutting plane approximations of the functions $\{J_\ell\}_{\ell \in \mathscr{S}}$ are built-up thanks to a memory $\mathfrak{B}_\ell^{(s)}$ dedicated to each sub-function ℓ, updated at each iteration s and defined as follows:

$$\mathfrak{B}_\ell^{(s)} := \{\mathbf{s}_\ell^{(i)}, \varepsilon_\ell^{(i)}\}_{i=1,\dots,n_{\mathfrak{B}}} \tag{11}$$

The bundle $\mathfrak{B}_\ell^{(s)}$ keeps only the $n_{\mathfrak{B}}$ last elements and behaves like a FIFO register in which the first element ($i = 1$) of the bundle is updated at the current iterate s after the whole bundle has been shifted and its last element dropped, namely:

$$\mathfrak{B}_\ell^{(s)} = \text{Update}(\mathfrak{B}_\ell^{(s-1)}, J_\ell(\mathbf{r}_\ell^{(s)}), \mathbf{g}_\ell^{(s)}, \mathbf{r}_\ell^{(s)}) \tag{12}$$

where the function update is defined by the algorithm (1).

Algorithm 1 Bundle update

1: **for** $i \leftarrow n_{\mathfrak{B}} - 1, \dots, 1$ **do** ▷ Shift the bundle memory
2: $\quad \mathbf{s}_\ell^{(i+1)}, \varepsilon_\ell^{(i+1)} \leftarrow \mathbf{s}_\ell^{(i)}, \varepsilon_\ell^{(i)}$
3: **end for**
4: $\mathbf{s}_\ell^{(1)} \leftarrow \mathbf{g}_\ell^{(s)}$
5: $\varepsilon_\ell^{(1)} \leftarrow J_\ell(\mathbf{r}_\ell^{(s)}) - \langle \mathbf{g}_\ell^{(s)}, \mathbf{r}_\ell^{(s)} \rangle$

The (sub-)gradients $\mathbf{s}_\ell^{(i)}$ and their corresponding linearization errors $\varepsilon_\ell^{(i)}$ (:5) are computed thanks to the values of the function $J_\ell(\mathbf{r}_\ell^{(s)})$ and (sub-)gradients $\mathbf{g}_\ell^{(s)}$ returned by each the subsystem ℓ when the coordinator requests an evaluation at the current iterate $\mathbf{r}_\ell^{(s)}$. Each bundle of information $\mathfrak{B}_\ell^{(s)}$ enables the so called *cutting plane approximation* $\check{J}_\ell^{(s)}(\cdot)$ to be defined according to:

$$\check{J}_\ell^{(s)}(\cdot) := \underset{i=1,\dots,n_{\mathfrak{B}}}{\text{Max}} \ \langle \mathbf{s}_\ell^{(i)}, \cdot \rangle + \varepsilon_\ell^{(i)} \tag{13}$$

where each linear piece $\langle \mathbf{s}_\ell^{(i)}, \cdot \rangle + \varepsilon_\ell^{(i)}, i = 1, \dots, n_{\mathfrak{B}}$ defines a half space as depicted on figure 1. Indeed, each *cutting plane i* is a supporting hyperplane of the epigraph $\mathbf{epi}(J_\ell)$ of the function J_ℓ and constitutes, since J_ℓ is supposed convex, a global under-estimator of J_ℓ.

Given the approximations $\check{J}_\ell^{(s)}, \ell \in \mathscr{S}$, an approximation $\check{J}^{(s)}$ of the centralized cost function can be obtained according to:

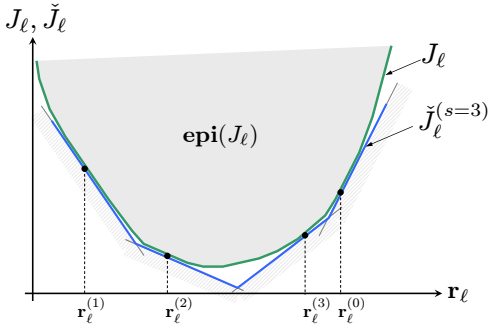

Fig. 1 Representation of J_ℓ and its piece-wise linear approximation $\check{J}_\ell^{(3)}$ after the third iteration. The approximate function is a global under-estimator of J_ℓ.

$$\check{J}^{(s)}(\mathbf{r}_\cdot) := \sum_{\ell \in \mathscr{S}} \check{J}_\ell^{(s)}(\mathbf{r}_\ell) \tag{14}$$

Instead of minimizing $\check{J}^{(s)}(\cdot)$, the following regularized optimization problem, denoted Master$^{(s)}$, is considered:

$$\text{Master}^{(s)}: \quad \mathbf{r}_\cdot^{(s+1)} := \underset{\mathbf{r}_\cdot \in \mathscr{D}}{\text{Argmin}} \left[\check{J}^{(s)}(\mathbf{r}_\cdot) + D_{\gamma^{(s)}}(\mathbf{r}_\cdot - \overline{\mathbf{r}}_\cdot^{(s)})\right] \tag{15}$$

where the stabilization term $D_{\gamma^{(s)}}(\mathbf{r}_\cdot - \overline{\mathbf{r}}_\cdot^{(s)})$ is introduced in order to prevent any drastic movement from the current best candidate point $\overline{\mathbf{r}}_\cdot^{(s)}$, which is called the stability center (or central point). Quite weak assumptions on the properties of $D_{\gamma^{(s)}}(\cdot)$ are necessary in order to ensure the convergence of the algorithm [5], however the most common choice is the following (euclidean) measure:

$$D_{\gamma^{(s)}}(\mathbf{r}_\cdot - \overline{\mathbf{r}}_\cdot^{(s)}) := \frac{1}{2 \cdot \gamma^{(s)}} \|\mathbf{r}_\cdot - \overline{\mathbf{r}}_\cdot^{(s)}\|_{L_2} \tag{16}$$

The parameter $\gamma^{(s)} > 0$ is the proximity parameter, it is updated at each iteration s. The central point $\overline{\mathbf{r}}_\cdot^{(s)}$ plays a crucial role in the bundle method since it enables, contrarily to pure cutting planes techniques [2], to keep track of the best known solution until iteration s and to help avoiding eventual oscillations resulting from a potentially poor approximation of the functions $\{J_\ell\}_{\ell \in \mathscr{S}}$.

In order to derive updating rules for the central point $\overline{\mathbf{r}}_\cdot^{(s)}$ and the parameter $\gamma^{(s)}$ let us define on one hand the predicted decrease at iteration s:

$$\hat{d}^{(s)} := J(\overline{\mathbf{r}}_\cdot^{(s)}) - \check{J}^{(s)}(\mathbf{r}_\cdot^{(s+1)}) \geq 0 \tag{17}$$

and on the other hand the real decrease $d(s)$:

$$d^{(s)} := J(\overline{\mathbf{r}}_{\cdot}^{(s)}) - J(\mathbf{r}_{\cdot}^{(s+1)}) \tag{18}$$

If the real decrease is greater than a certain fraction $f \in [0,1]$ of the predicted decrease then the current iterate s is called a *Serious Step*, in this case the coordinator enhanced the current solution : the central point $\overline{\mathbf{r}}^{(s)}$ is updated, and the proximity parameter $\gamma^{(s)}$ is increased. Otherwise ($d^{(s)} < f \cdot \hat{d}^{(s)}$), the step s is called a *Null Step*: the central point in kept unchanged and the proximity parameter $\gamma^{(s)}$ is decreased. Note however that in both situations the accuracy of the approximation $\check{J}^{(s)}$ is improved each time a new element is incorporated in the bundle (12).

The algorithm achieves the optimal solution with an accuracy η_J when the predicted decrease $\hat{d}^{(s)}$ is lower than a predefined accuracy on the objective function $\hat{d}^{(s)} \leq \eta_J$. Since the number of iterations allowed in our framework is very limited the condition $\hat{d}^{(s)} \leq \eta_J$ is rarely achieved. Thus, the algorithm is generally stopped if the iteration counter s reaches the maximum allowed number of iterations s_{max}.

Let us finally emphasize the fact that all iterates are feasible in the sense of respecting all global and also local constraints. This feature is very interesting since the algorithm can be stopped, if necessary, at any iteration. To find a feasible starting point, a projection of the best known solution at instant k, denoted $\mathbf{r}_{\cdot}^{\sharp}$, on the domain $\mathscr{D}$ is performed at the initialization phase. This operation is performed centrally, namely:

$$\mathbf{r}_{\cdot}^{(0)} := \text{Init}(\mathbf{r}_{\cdot}^{\sharp}) = \underset{\mathbf{r}_{\cdot} \in \mathscr{D}}{\text{Argmin}} \, \|\mathbf{r}_{\cdot} - \mathbf{r}_{\cdot}^{\sharp}\| \tag{19}$$

Some other technical details regarding, bundle compression techniques and update strategies of $\gamma^{(s)}$ have been omitted but are available in [5]

3.3 Distributing the optimization over time

Ideally, all the iterations described in the previous section have to take place during the sampling period $[k-1,k]$. Moreover, many of these iterations have to be performed in order to achieve a sufficiently good approximation of cost functions $J_{\ell,k}$. This may be incompatible with the sampling period. The idea is then to still use a modified version of the past information contained in the previous bundles $\mathfrak{B}_{\ell,k-j}, j = 1,2,\ldots$. This operation relies on the assumption that the functions $J_{\ell,k}(\cdot)$ don't change drastically and therefore the approximations of these functions, stored in the bundle $\mathfrak{B}_{\ell,k-1}$, can be used as an initialization at instant k of the bundle but "corrected" in the following way :

$$\mathfrak{B}_{\ell,k}^{(0)} = \{m_{\ell,k} \cdot \overleftarrow{\mathbf{s}}_{\ell}^{(i)}, m_{\ell,k} \cdot \varepsilon_{\ell}^{(i)}\}_{i=1,\ldots,n_{\mathfrak{B}}} \tag{20}$$

where the approximated (sub-)gradient $\overleftarrow{\mathbf{s}}_{\ell}^{(i)}$ is obtained by conveniently shifting and completing the (sub-)gradient vectors obtained over the last instants. The operator shift $\overleftarrow{\mathbf{v}}_k$ of any profile $\mathbf{v}_k = [v_k^T, \ldots, v_{k+N-1}^T]^T$ is defined by:

$$\overleftarrow{\mathbf{v}}_k := [v_{k+1}^T, \ldots, v_{k+N-1}^T, v_{k+N-1}^T]^T \tag{21}$$

The positive parameter $m_{\ell,k} \in [0\,, 1-\varepsilon]$ $(0 < \varepsilon \ll 1)$ is the memory factor and plays a central role in the scheme, it enables to "forget" a certain part of the information. The memory parameter expresses a trade-off between the quantity of information that one wants to keep and the fact the cuts used as initialization for the bundle at instant k should be under-estimators of the function $J_{\ell,k}$ in order to prevent the optimal solution from begin excluded from the search domain. Remark that when $m_{\ell,k} = 0$, no information of the previous instants is kept at the current instant and the whole bundle information is equivalent to $J_{\ell,k} > 0$. For these reasons, the memory factors $m_{\ell,k}$ are adjusted in accordance with the quality of the initial approximation $\check{J}_{\ell,k}^{\text{init}}$ computed at the initial point $\mathbf{r}_{\ell,k}^{(0)}$ and generalized over the whole bundle (figure 2).

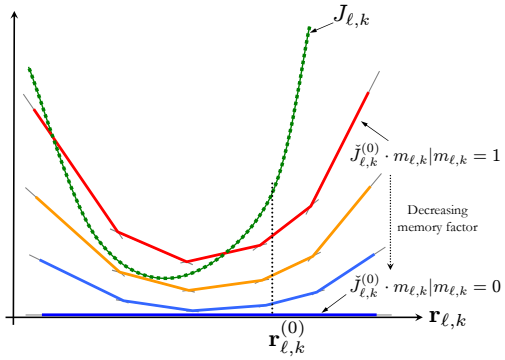

Fig. 2 Illustration of the role played by the memory factor $m_{\ell,k}$ enabling previously computed cutting planes to remain relevant as lower bound of the function to be approximated.

$$m_{\ell,k} := 1 - \text{Sat}_{[0\,,1-\varepsilon]} \left(\frac{[\check{J}_{\ell,k}^{\text{init}}(\mathbf{r}_{\ell,k}^{(0)}) - J_{\ell,k}(\mathbf{r}_{\ell,k}^{(0)})]^2}{[J_{\ell,k}(\mathbf{r}_{\ell,k}^{(0)})]^2} \right) \tag{22}$$

where the initial approximation $\check{J}_{\ell,k}^{init}$ results from the bundle $\mathfrak{B}_{\ell,k}^{(0)} | m_{\ell,k} = 1$ (with no forgetting factor). The function $\text{Sat}_{[a\,b]}(\cdot)$ is the saturation function.

3.4 The correction mechanism

It is essential to understand that the memory capability starts from the assumption that the functions $J_{\ell,k}$ from one decision instant to the next are supposed to have a

slow change. In such conditions the initial rough approximation (20) is supposed to give a rough yet valuable starting approximation at instant k. However, one has to mention that this assumption is very hard to check. Moreover, note that the gradient approximation $\overleftarrow{\mathbf{s}}_\ell$ introduced previously introduces itself an intrinsic error and that no knowledge about the sub-systems respective states and/or disturbances is neither available nor taken into account.

This is why a correction mechanism has to be introduce in order to ensure, a priori, that the linear pieces $i = 1,\dots,n_{\mathfrak{B}}$ remain under-estimators of the functions $\{J_\ell\}_{\ell\in\mathscr{S}}$ in order to prevent the optimal point to be excluded from the current search domain. Therefore, each time a new evaluation of the function $J_{\ell,k}(\mathbf{r}_{\ell,k}^{(s)})$ is performed, it is easy to check that each linear piece composing the current approximation $\check{J}_\ell^{(s)}$ under-estimates the function $J_{\ell,k}$ when compute at $(\mathbf{r}_{\ell,k}^{(s)})$, defining $\beta^{(i)}$ as the difference between the *true* value of the function at the current iterate s (returned by subsystems) and the value of the i^{th} linear piece.

$$\beta^{(i)} := J_{\ell,k}(\mathbf{r}_{\ell,k}^{(s)}) - (\langle \mathbf{s}_\ell^{(i)}, \mathbf{r}_{\ell,k}^{(s)}\rangle + \varepsilon_\ell^{(i)}) \quad i = 1,\dots,n_{\mathfrak{B}} \tag{23}$$

If $\beta^{(i)} \geq 0$ then the linear piece i is a priori a valid under-estimator of the function and it is kept in the bundle. Otherwise $\beta^{(i)} < 0$ and the linear piece i in the bundle has to be vertically translated in order to correct its position, namely:

$$\text{if} \quad \beta^{(i)} < 0 \quad \text{than} \quad \varepsilon^{(i)} = \varepsilon^{(i)} + 1.1\cdot\beta^{(i)} \tag{24}$$

For the sake of clarity, the correction mechanism is defined as a function:

$$\text{Correct}(\mathfrak{B}_{\ell,k}^{(s)}, J_{\ell,k}(\mathbf{r}_{\ell,k}^{(s)}), \mathbf{r}_{\ell,k}^{(s)}) \tag{25}$$

where the function Correct$(\cdot)$ is defined as follows:

Algorithm 2 Bundle correction mechanism

for $i \leftarrow 1,\dots,n_{\mathfrak{B}}$ **do**
  $\beta^{(i)} = J_{\ell,k}(\mathbf{r}_{\ell,k}^{(s)}) - (\langle \mathbf{s}_\ell^{(i)}, \mathbf{r}_{\ell,k}^{(s)}\rangle + \varepsilon_\ell^{(i)})$
  if $\beta^{(i)} < 0$ **then**
    $\varepsilon^{(i)} = \varepsilon^{(i)} + 1.1\cdot\beta^{(i)}$
  end if
  $\mathfrak{B}_{\ell,k}^{(s)} = \{\mathbf{s}_\ell^{(i)}, \varepsilon_\ell^{(i)}\}_{i=1,\dots,n_{\mathfrak{B}}}$ ▷ Update the bundle information
end for

Finally, in order to make as clear as possible the presentation of the technique, the complete framework is presented on figure 3, in which the elements $\{\text{BM}_\ell\}_{\ell\in\mathscr{S}}$ are the bundle managers units, they perform the operations related to bundle storage, correction and memory factors described previously. The algorithm (3)[2] (see page 12) exhaustively summarizes the proposed framework.

[2] Contraction and dilatation parameters of $\gamma^{(s)}$, here 1.1 and 0.8 are provided as indications.

4 Theoretical results availability

Convergence results for the classical bundle method (for an optimization problem defined once for all) can be found in [5] when the cost function is convex. The difficulties (in carrying the convergence analysis) associated to the dynamic character of the problem (that changes because of the state variation during a sampling period) can be overcome following the same guidelines recently used in [1]. Roughly speaking, it can be shown that the decrease in the cost function that is guaranteed in the ideal static case are disturbed by a term which is $O(\tau^2)$ where τ is the control updating period. As the latter is precisely reduced by the distributed in time scheme (requiring fewer iterations to be needed at each updating period) a sort of virtuous circle in favor of stability circle is favor of stability assessment takes place.

5 Application results availability

The proposed approach has been assessed in [6, 9]. The case study consists of a 20 zones building (20 subsystems). The building is able to store electrical energy in batteries in order to redistribute it to zones. This storage capability offers the benefit of shifting energy consumption of the building to periods in which electricity is cheaper since the final objective here is to minimize the energy invoice while respecting comfort constraints. The power grid can provide to the building a limited amount of power and constitute in this sense a hard constraint that must be respected. The local MPC controllers enable to handle local actuators at the least energetic cost besides ensuring occupants comfort[7, 8]. Therefore, the building should take into account this power limitation in advance in order to store energy in an electrical form or a thermal form in the zones in order to maintain occupants comfort within the prescribed level. Remind that the zones computation units (local MPCs) are

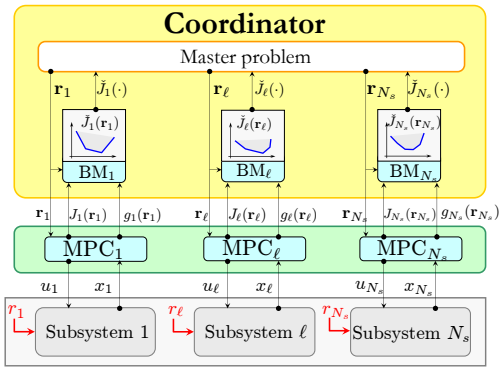

Fig. 3 Hierarchical Distributed Model Predictive Control scheme.

Algorithm 3 Bundle method based DMPC with memory

for each decision instant k **do**
  $s \leftarrow 0$ ▷ Initialize iteration counter
  $\mathbf{r}_{\cdot,k}^{(0)} \leftarrow \text{Init}(\overleftarrow{\mathbf{r}^{\star}_{\cdot,k-1}})$ ▷ Sends feasible candidate profiles for subsystems
  for $\ell \leftarrow 1,\dots,N_s$ **do** ▷ Parallel operation performed by the subsystems
    $J_{\ell,k}(\mathbf{r}_{\ell,k}^{(0)}), \mathbf{g}_{\ell,k}^{(0)} \leftarrow \text{MPC}_\ell(\mathbf{r}_{\ell,k}^{(0)})$ ▷ ↑↑ The subsystems solve their optimization
    $u_{\ell,k}^{\sharp} \leftarrow (u_{\ell,k}^{\star})^{(0)}$ ▷ ↓↓ Each subsystems stores the current optimal control solution
    $m_{\ell,k} := 1 - \text{Sat}_{[0,1-\varepsilon]}\left([\check{J}_\ell^{\text{init}}(\mathbf{r}_{\ell,k}^{(0)}) - J_{\ell,k}(\mathbf{r}_{\ell,k}^{(0)})]^2 / [J_{\ell,k}(\mathbf{r}_{\ell,k}^{(0)})]^2\right)$ ▷ Compute memory fact.
    $\mathfrak{B}_{\ell,k}^{(0)} = \{m_{\ell,k} \cdot \overleftarrow{\mathbf{s}}_\ell^{(i)}, m_{\ell,k} \cdot \boldsymbol{\varepsilon}_\ell^{(i)}\}_{i=1,\dots,n_{\mathfrak{B}}}$ ▷ Forgetting operation
    $\mathfrak{B}_{\ell,k}^{(0)} \leftarrow \text{Update}(\mathfrak{B}_{\ell,k}^{(0)}, J_{\ell,k}(\mathbf{r}_{\ell,k}^{(0)}), \mathbf{g}_{\ell,k}^{(0)}, \mathbf{r}_{\ell,k}^{(0)})$ ▷ Add new cut to the bundle
    $\mathfrak{B}_{\ell,k}^{(0)} \leftarrow \text{Correct}(\mathfrak{B}_{\ell,k}^{(0)}, J_{\ell,k}(\mathbf{r}_{\ell,k}^{(0)}), \mathbf{r}_{\ell,k}^{(0)})$ ▷ Correct bundles information
  end for
  $\bar{\mathbf{r}}_{\cdot}^{(0)} \leftarrow \mathbf{r}_{\cdot}^{(0)}, \quad \hat{d}^{(0)} \leftarrow \infty$ ▷ Initialize the central point
  while $s \leq s_{max}$ **and** $\hat{d}^{(s)} \geq \eta_{\mathbf{J}}$ **do**
    $\mathbf{r}_{\cdot,k}^{(s+1)} \leftarrow \text{Master}^{(s)}$ ▷ ↓↓ Coordinator solves the master problem
    $\hat{d}^{(s)} = J(\bar{\mathbf{r}}_{\cdot,k}^{(s)}) - \check{J}^{(s)}(\mathbf{r}_{\cdot,k}^{(s+1)})$ ▷ Compute predicted decrease
    for $\ell \leftarrow 1,\dots,N_s$ **do**
      $J_{\ell,k}(\mathbf{r}_{\ell,k}^{(s+1)}), \mathbf{g}_{\ell,k}^{(s+1)} \leftarrow \text{MPC}_\ell(\mathbf{r}_{\ell,k}^{(s+1)})$ ▷ ↑↑ The subsystems solve their optimization
      $d^{(s)} = J(\bar{\mathbf{r}}_{\cdot,k}^{(s)}) - \check{J}^{(s)}(\mathbf{r}_{\cdot,k}^{(s+1)})$ ▷ Compute real decrease
      if $\hat{d}^{(s)} > f \cdot d^{(s)}$ **then** ▷ Serious step
        $\gamma^{(s+1)} \leftarrow 1.1 \cdot \gamma^{(s)}$, $\bar{\mathbf{r}}_{\cdot,k}^{(s+1)} \leftarrow \mathbf{r}_{\cdot,k}^{(s+1)}$ ▷ Increase γ, Update the central point
        $u_{\ell,k}^{\sharp} \leftarrow (u_{\ell,k}^{\star})^{(s+1)}$ ▷ ↓↓ The subsystems update optimal solutions
      else
        $\gamma^{(s+1)} \leftarrow 0.8 \cdot \gamma^{(s)}$, $\bar{\mathbf{r}}_{\cdot,k}^{(s+1)} \leftarrow \bar{\mathbf{r}}_{\cdot,k}^{(s)}$ ▷ Decrease γ, central point is unchanged
      end if
      $\mathfrak{B}_{\ell,k}^{(s+1)} \leftarrow \text{Update}(\mathfrak{B}_{\ell,k}^{(s)}, J_{\ell,k}(\mathbf{r}_{\ell,k}^{(s+1)}), \mathbf{g}_{\ell,k}^{(s+1)}, \mathbf{r}_{\ell,k}^{(s+1)})$ ▷ Add new cut to the bundle
      $\mathfrak{B}_{\ell,k}^{(s+1)} \leftarrow \text{Correct}(\mathfrak{B}_{\ell,k}^{(s+1)}, J_{\ell,k}(\mathbf{r}_{\ell,k}^{(s+1)}), \mathbf{r}_{\ell,k}^{(s+1)})$ ▷ Correct bundles information
    end for
    $s \leftarrow s+1$ ▷ Increment iteration counter
  end while
  $\mathbf{r}_{\cdot,k}^{\star} \leftarrow \bar{\mathbf{r}}_{\cdot,k}^{(s+1)}$
  for $\ell \leftarrow 1,\dots,N_s$ **do** ▷ Parallel operation performed by the subsystems
    Subsystem ℓ Applies $u_{\ell,k}^{\sharp}$
  end for
end for

much less powerful than the computation unit of the coordinator therefore the number iterations s_{max} is the most important criterion concerning the assessment of the scheme. Nevertheless, one has to mention that the introduction of memory feature should not result in an excessive increase of the computational burden of the coordinator. This is why, the computational time of the coordinator is examined in this section.

In order to illustrate the benefit of the memory mechanism presented in this chapter a set of simulations on 12 hours has been performed for different combinations

of s_{max} (maximum number of iterations) and $n_{\mathfrak{B}}$ (bundle memory size), the results are depicted on figure 5. Let however distinguish two main phases:

- **(12p.m - 7a.m)**. During this phase the objective function is decreasing in order to enable the system to reach its steady state. Notice that the introduction of the memory feature enhances considerably the overall performance of the scheme. For instance, using only 3 iterations with memory enables to perform a fastest decrease of the objective function (excluding the very first decision instants) at a lower computational time compared to 10 iterations without memory. Note also that the computational time can be higher when using memory with 4 iterations than with 5 iterations, this results from the fact that the coordinator compensate the lack of new information by using a higher number of cuts previously stored in the bundle, this results in harder optimization problems to solve, implying a larger computational burden -which may be as high as in the case of 10 iterations without memory-.
- **(7a.m -12a.m)** during this phase the system reaches its steady state. The schemes exhibit quite similar performances. Nevertheless, the computational burden is still lower in the cases with memory -since less iterations are allowed-

6 Conclusion

In this chapter, a distributed-in-time algorithm for distributed model predictive control has been presented. The main advantages lie in decreasing drastically the number negotiation iterations required in order to achieve relevant solution at each decision instant. The scheme is particularly adapted to situations in which the computation units at the subsystem layer are much more ineffective compared to the one at coordination layer. In the presented case study, this enabled to reduce the computational time by more than 60 % for quite similar performances. Future investigations concern the possible on-line tuning of the bundle size as well as the number of iterations. Moreover, a deeper and more rigorous analysis of the convergence is also undergoing. Finally, while the local subsystem MPC controller has been already implemented in the Schneider-Electric RoomBox device (see [8] for more details), implementation of the distributed scheme is currently investigated.

Acknowledgements This work is part of **HOMES** collaborative program (http://www.homesprogramme.com). This program is funded by **OSEO** (http://www.oseo.fr).

References

1. M. Alamir. A framework for real-time implementation of low dimensional parameterized NMPC. *Automatica*, 48:198–204, 2011.

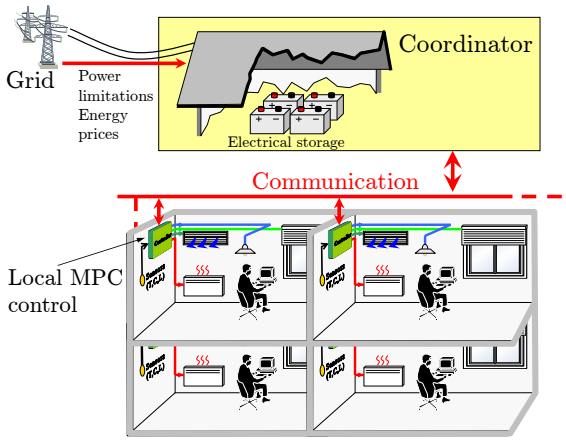

Fig. 4 Distributed Model Predictive in a multizone building. The coordinator gathers global information and ensures coordination of local controllers besides battery management.

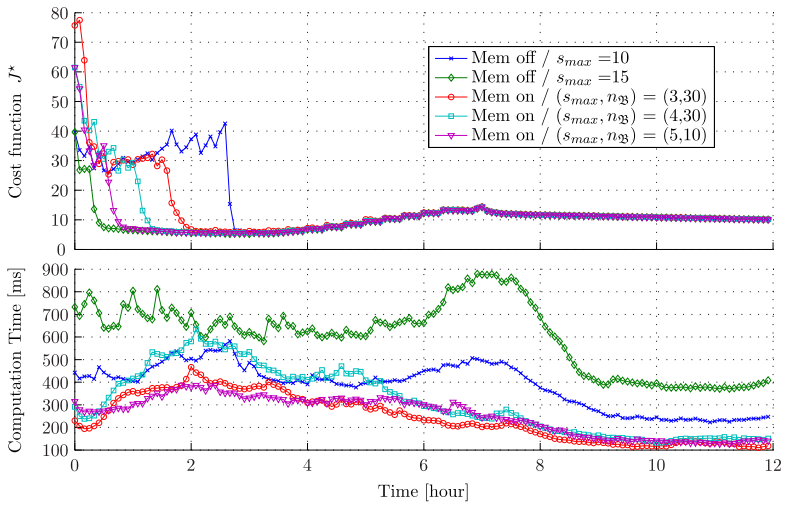

Fig. 5 Assessment of the distributed-in-time mechanism.

2. O. Briant, C. Lemaréchal, Ph. Meurdesoif, S. Michel, N. Perrot, and F. Vanderbeck. Comparison of bundle and classical column generation. *Mathematical programming*, 113(2):299–344, 2008.
3. E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, February:44–52, 2002.
4. M. Diehl. Report of literature survey, analysis, and comparison of on-line optimization methods for hierarchical and distributed MPC. Hierarchical and Distributed Model Predictive Control of Large-Scale Systems-Deliverable Number: 4.1.1/ Senventh Framework programme theme -ICT, 2009.

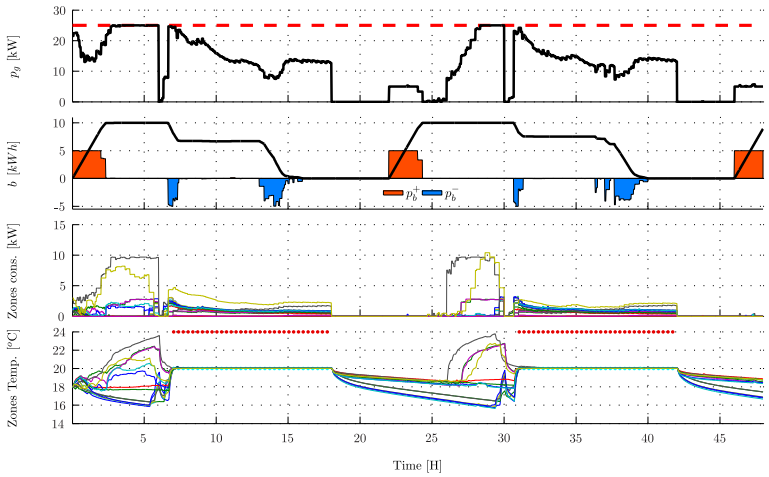

Fig. 6 Simulation results with $N_s = 20, s_{max} = 3, n_{\mathfrak{B}} = 30$. Notice that the zones temperatures lie within their respective bounds during occupied hours of the building (CO_2 levels and lighting levels are also respected in all zones but not displayed for lack of space)

5. A. Frangioni. Generalized bundle methods. *SIAM Journal of Optimization*, 13 (1):117–156, 2002.
6. M. Y. Lamoudi, M. Alamir, and P. Béguery. Distributed constrained model predictive control based on bundle method for building energy management. In *50th IEEE Conference on Decision and Control and European Control Conference- Orlando*, 2011.
7. M. Y. Lamoudi, M. Alamir, and P. Béguery. Unified NMPC for multi-variable control in smart buildings. In *IFAC 18th World Congress, Milano, Itlay*, 2011.
8. M. Y. Lamoudi, M. Alamir, and P. Béguery. Model predictive control for energy management in buildings- part 1: zone model predictive control. In *IFAC workshop on Nonlinear Model Predictive Control. The Netherlands.*, 2012.
9. M. Y. Lamoudi, M. Alamir, and P. Béguery. Model predictive control for energy management in buildings- part 2: Distributed model predictive control. In *IFAC workshop on Nonlinear Model Predictive Control. The Netherlands*, 2012.
10. R.R. Negenborn, B. De Schutter, and J. Hellendoorn. Multi-agent model predictive control: A survey. In *Technical report 04-010 -Delft Center for Systems and Control -Delft University of Technology*, 2004.
11. H. Scheu, J. C. Calderon, Dang Doan, J. F. Garcia, R. Negenborn, A. Tarau, F. V. Arroyave, B. D. Schutter, J. J. Espinosa, and W. Marquardt. Report on assessment of existing coordination mechanisms for simple case studies, and on possible options for improving and extending these coordination mechanisms. Hierarchical and Distributed Model Predictive Control of Large-Scale Systems-Deliverable Number: 3.3.1/ Senventh Framework programme theme - ICT, 2009.