European Journal of Control 68 (2022) 100676

Contents lists available at ScienceDirect

European Journal of Control

journal homepage: www.elsevier.com/locate/ejcon

Coalitional model predictive control with different inter-agent interaction modes☆

A. Sánchez-Amores*, P. Chanfreut, J.M. Maestre, E.F. Camacho

Department of Systems and Automation Engineering, University of Seville, Camino de los Descubrimientos, Seville, 41092, Spain

ARTICLE INFO

Article history:
Received 12 April 2022
Accepted 6 June 2022
Available online 14 June 2022

Recommended by Prof. T Parisini

Keywords:
Coalitional control
Distributed control
Model predictive control
Networked control systems

ABSTRACT

Coalitional control is a type of distributed control characterized by the dynamic adjustment of the overall controller structure so that only strongly coupled agents interact with each other. In particular, local controllers merge into cooperative coalitions (or clusters) only when it improves global performance, thus reducing the overall cooperation burden. This paper proposes a novel coalitional model predictive control (MPC) approach in which coupled variables are decomposed into a *public* part, which is optimized by the neighboring agents, and a *private* one, which is locally controlled by the agent that owns it. The bounds on these variables are negotiated in a distributed manner, and a threshold is established to trigger different interaction modes, including the classical decentralized and distributed approaches, and flexible modes of cooperation. Finally, to illustrate the benefits of this control scheme, results on a simulated eight input-coupled tanks plant are provided.

1. Introduction

During the last years, different distributed control approaches that promote the formation of dynamic clusters of local controllers have been proposed [10,12–14]. The underlying idea is to use coordination mechanisms based on information exchange only when they generate a meaningful increment of performance. Otherwise, it might be preferable to have controllers working in a decentralized manner, i.e., without communication, to save communication and coordination costs. Nevertheless, this approach leads to a problem formulation that is typically casted as a mixed-integer optimization problem, for the state of the links of the communication network is discrete. In this article, a new method where local controllers solve a quadratic program is proposed. To this end, coupling variables are partitioned and assigned to the corresponding agents, leading to different types of inter-agent interaction modes that range from decentralized to distributed control. The switching between these modes of interaction is regulated by decision variables whose value is decided in a distributed fashion.

From the different control frameworks available, we focus on model predictive control (MPC), which is a continuous replanning method that recalculates control actions at each time step in a receding horizon fashion. This framework has been applied in multiple industrial applications for decades [21–23]. The need for distributed implementations of MPC stems from problems where there are constraints that impede or limit the implementation of a centralized control strategy. For example, it may not be possible to solve the optimization problem for a large-scale system such an industrial plant within the timing constraints imposed by the sampling time. Other times the system is naturally distributed, e.g., power grids [25] and traffic networks [5], and it is preferable to have several independent controllers that coordinate their actions for reasons such as scalability and redundancy. Whatever the reason is, the overall control problem is partitioned into a set of coupled smaller pieces, which are assigned to local controllers or agents that communicate to agree on how they mutually interact with each other. Multiple methods to achieve coordination have been introduced during the last years and are surveyed in [17,20].

A significant amount of work has also been dedicated to minimize communication burden, for the amount of data transferred between controllers can be considerable. For example, as it is shown in [16], a distributed multiple shooting scheme is required to exchange more than 6 million floats per time instant. Whether

☆ This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (OCONTSOLAR, grant agreement No 789051), the Spanish Training Program for Academic Staff (FPU17/02653), and project C3PO-R2D2 (Grant PID2020-119476RB-I00 funded by MCIN/AEI/10.13039/501100011033).

* Corresponding author.
E-mail addresses: asamores@us.es (A. Sánchez-Amores), pchanfreut@us.es (P. Chanfreut), pepemaestre@us.es (J.M. Maestre), efcamacho@us.es (E.F. Camacho).

https://doi.org/10.1016/j.ejcon.2022.100676
0947-3580/

this is a limiting issue depends on the application considered, but it invites questioning if that amount of information exchanged is worth to increase performance. For example, flexibility regarding the extent of coupling is exploited in [24], where local controllers broadcast information regarding how much of their constraint space they are using to improve performance in a distributed tube-based MPC. Coalitional control promotes data exchange only when it leads to a significant increment of performance, for communication is no longer assumed to be costless [1,6,9–11]. To this end, a cost is assigned either to the use of links of the communication network or to the effort necessary to coordinate a certain number of cooperating agents. As a consequence, the control network topology changes dynamically, leading to dynamic partitioning of agents into disjoint clusters known as coalitions [7]. This goal can be achieved either in a top-down, e.g., [6,11], or bottom-up manner, e.g., [1,9].

The additional degree of freedom gained with the network topology, which becomes another variable into the optimization problem, has a cost in terms of additional complexity. For example, top-down coalitional strategies usually resort to solving mixed-integer optimization problems that consider the network links state as a discrete variable. Alternatively, it is also possible to consider that links are activated randomly and only those leading to an increasing performance remain active. Both approaches are not without their problems. The first type of approach increases the computation burden and usually requires to change the network topology at a lower rate by a supervisory control layer to keep the problem feasible. As for bottom-up methods, they incorporate a certain degree of suboptimality into the problem due to the heuristics used in the activation of links.

In this paper, we propose a new approach to coalitional MPC for systems coupled through system variables. The idea is to distribute coupling variables sharing a common constraint space between agents. Each part of the coupling variable is associated with a different mode of interaction, and the amount of constraint space for each of these modes is to be decided by agents in a decentralized manner. In this way, agents can switch from decentralized to full communication control using a convex optimization problem. This decomposition into *private* and *shared* variables is also proposed in [2] to reject large disturbances, where the upper layer of a hierarchical MPC may force agents to share part of their local inputs to assist other subsystems. Similarly, a hierarchical MPC for electricity networks where a set of generators, storage systems, and loads are dynamically partitioned into locally controlled clusters is presented in [14]. In this case, the supervisory layer comes into play when some of these clusters need support from their neighbors to achieve the control goals. Unlike [2,14], our approach does not rely on a supervisory layer and seeks to optimize in real time the coordination structure between local controllers so as to balance performance and communication costs.

The outline of the rest of this work is organized as follows. Section II presents the model of the system and the basic problem setting for coalitional MPC. Section III describes the decomposition in private and public variables. Moreover, the dual decomposition algorithm is integrated within the negotiation of shared variables. In Section IV, the proposed control scheme is introduced. Section V includes simulation results on a simulated eight input-coupled tanks benchmark. Finally, conclusions are given in Section VI.

2. Problem setting

Without of loss of generality, we will focus on input-coupled systems, but the same approach can be applied in case of state coupling. Consider a system divided into a set $\mathcal{N}=\{1,2,\dots,N\}$ of input-coupled subsystems with LTI dynamics:

$$\begin{aligned} &x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i(k) + d_i(k),\\ &\text{with } \; d_i(k) = \textstyle\sum_{j\in\mathcal{N}_i} B_{ij}u_j(k), \end{aligned} \tag{1}$$

where $x_i \in \mathbb{R}^{n_{x_i}}$ and $u_i \in \mathbb{R}^{n_{u_i}}$ are respectively the state and input vector of subsystem $i \in \mathcal{N}$, and variable $d_i \in \mathbb{R}^{n_{x_i}}$ represents the input coupling with neighboring subsystems $\mathcal{N}_i = \{j \in \mathcal{N} : B_{ij} \neq \mathbf{0},\ j \neq i\}$. Also, assume that the subsystems' states and inputs are subject to the following constraints: $x_i(k) \in \mathcal{X}_i$ and $u_i(k) \in \mathcal{U}_i$, for all $k \geq 0$, where $\mathcal{X}_i$ and $\mathcal{U}_i$ are convex sets.

To describe the centralized model, the overall state and input vectors are defined as the aggregation of all subsystems' states and inputs, i.e., $x = [x_i]_{i\in\mathcal{N}} = [x_1^{\mathrm{T}}, \dots, x_N^{\mathrm{T}}]^{\mathrm{T}}$ and $u = [u_i]_{i\in\mathcal{N}} = [u_1^{\mathrm{T}}, \dots, u_N^{\mathrm{T}}]^{\mathrm{T}}$, leading to:

$$x(k+1) = Ax(k) + Bu(k), \tag{2}$$

where global matrices $A = [A_{ii}]_{i\in\mathcal{N}}$ and $B = [B_{ij}]_{i,j\in\mathcal{N}}$ are defined by aggregating (1) for all subsystems.

Hereon, assume that the subsystems in $\mathcal{N}$ are managed by a set of local controllers or agents, which, in turn, can communicate through a data network. This network is described as an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{L}$ is the set of links, i.e., $\mathcal{L} \subseteq \mathcal{L}^{\mathcal{N}} = \{\{i, j\} | i, j \in \mathcal{N}\}$, and the set of nodes $\mathcal{N}$ represents the agents. In coalitional control, the state of the links are dynamically switched between *enabled* and *disabled* so that the set of agents can be *partitioned* into disjoint communication components or coalitions. In particular, let $\mathcal{P}(k) = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{|\mathcal{P}(k)|}\}$ be the partition imposed at time instant k, where $\mathcal{C}_i$ represents each of the resulting coalitions (or clusters) and $|\mathcal{P}(k)|$ denotes the cardinality of $\mathcal{P}(k)$. Note that $\mathcal{C}_i$ can represent from a singleton to the grand coalition. For example, in the decentralized configuration there are $|\mathcal{N}|$ *coalitions* that are singletons, whereas the centralized configuration groups all agents into the grand coalition, and hence $|\mathcal{P}(k)| = 1$. Inside each coalition, agents share data and coordinate their actions, but note that there is no inter-cluster communication.

3. Sharing coupling variables

In this paper, we follow a decomposition approach where shared variables are partitioned into public and private parts to allow different interaction modes. As will be seen, this decomposition changes dynamically, allowing the system to work using different levels of partial cooperation. In particular, a local variable u_i is decomposed as

$$u_i(k) = u_i^{\mathrm{pr}}(k) + \sum_{j\in\mathcal{M}_i} u_{ij}^{\mathrm{pu}}(k), \tag{3}$$

where

(i) $u_i^{\mathrm{pr}}(k)$ is the private part of the variable, which is controlled exclusively by the agent that *owns* it, i.e., i, and it must verify $u_i^{\mathrm{pr}}(k) \in \alpha_i(k)\mathcal{U}_i$ with $\alpha_i(k) \in [0, 1]$ for all $k \geq 0$.

(ii) $u_{ij}^{\mathrm{pu}}(k)$ is the public part of $u_i(k)$ controlled by agent j and must verify $u_{ij}^{\mathrm{pu}}(k) \in \alpha_{ij}(k)\mathcal{U}_i$. The set of agents j that can manipulate the public part of $u_i(k)$ are defined as *affected* subsystems $\mathcal{M}_i = \{j \in \mathcal{N} : B_{ji} \neq \mathbf{0},\ j \neq i\}$. Notice that sets $\mathcal{N}_i$ and $\mathcal{M}_i$ are not necessarily the same and depend on the dynamics of the system, i.e., while $\mathcal{N}_i$ contains the set of agents j which affect i, $\mathcal{M}_i$ defines the set of agents affected by i.

Hereafter, it is considered that at each time instant k the agents should decide the values of $u_i^{\mathrm{pr}}(k)$ and $u_{ij}^{\mathrm{pu}}(k)$, for all $i \in \mathcal{N}$ and $j \in \mathcal{M}_i$, in a manner that the following inequality is satisfied:

$$\alpha_i(k) + \sum_{j\in\mathcal{M}_i} \alpha_{ij}(k) \leq 1. \tag{4}$$

Note that if (4) holds, then input $u_i(k)$ computed as in (3) will satisfy the constraint $u_i(k) \in \mathcal{U}_i$ given in the previous section. Also, we assume that local variables $u_i^{\rm pr}$ and $u_{ji}^{\rm pu}$ are treated respectively by the neighboring agents $\mathcal{N}_i$ and by the *affected* agents $\mathcal{M}_i$ as bounded disturbances, whose scale factor $\alpha_i(k)$ and $\alpha_{ji}(k)$ can be negotiated in case the controllers share information.

Let us also define $\bar{\alpha}$ as the threshold that enables the agents communication. In this regard, depending on the values of $\alpha_{ij}(k)$, agents can switch between different modes of operation ranging between these two extreme cases:

- All agents work in a decentralized manner, i.e., there is no communication, which happens when $\alpha_{ij}(k) < \bar{\alpha}$ for all $u_{ij}^{\rm pu}(k) \in \alpha_{ij}(k)\mathcal{U}_i$.
- All agents work in a distributed manner, i.e., all of them share data if $\alpha_{ij}(k) \geq \bar{\alpha}$ for all $u_{ij}^{\rm pu}(k) \in \alpha_{ij}(k)\mathcal{U}_i$. Unlike standard distributed schemes, here the boundaries (and not the values) of coupling variables are negotiated. The rationale of this choice is to reduce the cooperation burden.

Hence, agents work in a flexible manner that admits partial modes of cooperation. That is, there may be clusters of agents that communicate, i.e., *coalitions* [10], whereas others may be operating in a decentralized manner.

In what follows, we introduce a dual decomposition-based algorithm that exploits the concept of private and public variables above described. Each agent $i \in \mathcal{N}$ will optimize the private part of its input variable $u_i^{\rm pr}$ and the variable α_i that bounds its constraints, as well as the public part of neighboring input variables $u_{ji}^{\rm pu}$, and the variable α_{ji} that determines its constraints. To deter a fully cooperative operation, the use of public variables will be penalized with a higher cost than the private ones. Also, to *protect* the agents against the neighboring uncertainty, we introduce a set of scenario-based constraints that aim satisfying state and input constraints while reducing the degree of conservativeness. The latter is detailed in the following subsections.

3.1. Scenario-based MPC

Considering (1) and (3), the model of each subsystem $i \in \mathcal{N}$ can be rewritten as follows:

$$x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i^{\rm pr}(k) + \sum_{j\in\mathcal{N}_i} B_{ij}u_{ji}^{\rm pu}(k) + \underbrace{\sum_{j\in\mathcal{M}_i} B_{ii}u_{ij}^{\rm pu}(k) + \sum_{j\in\mathcal{N}_i} B_{ij}u_j^{\rm pr}(k)}_{w_i(k)}, \tag{5}$$

where $w_i(k)$ is decomposed into a first term that contains the public part of $u_i(k)$ that is managed by agents $\mathcal{M}_i$ that are affected by it, and the second one includes the private variables of neighboring agents $\mathcal{N}_i$. Due to the definition of private and public variables, $w_i(k)$ satisfies:

$$w_i(k) \in \bigoplus_{j\in\mathcal{M}_i} B_{ii}\mathcal{W}_{ij}^{\rm pu} \oplus \bigoplus_{j\in\mathcal{N}_i} B_{ij}\mathcal{W}_j^{\rm pr} \tag{6}$$

where $\mathcal{W}_{ij}^{\rm pu}$ and $\mathcal{W}_j^{\rm pr}$ are polyhedral sets such that

$$\begin{aligned} u_{ij}^{\rm pu}(k) &\in \mathcal{W}_{ij}^{\rm pu} = \alpha_{ij}(k)\mathcal{U}_i, \\ u_j^{\rm pr}(k) &\in \mathcal{W}_j^{\rm pr} = \alpha_j(k)\mathcal{U}_j. \end{aligned} \tag{7}$$

To take into account these uncertainties, it is possible to follow a conservative robust approach, e.g., tube-based MPC [15,19]. However, in order to reduce conservativeness, we propose a stochastic approach that generates a fixed number n_s of equiprobable random scenarios considering different possible realizations of the disturbance.

Let $\bar{x}_i$ be the part of equation (5) that does not depend on the disturbances, i.e., $\bar{x}_i(k) = A_{ii}x_i(k) + B_{ii}u_i^{\rm pr}(k) + \sum_{j\in\mathcal{N}_i} B_{ij}u_{ji}^{\rm pu}(k)$. Regarding n_s possible realizations of the uncertainty, the state prediction using model (5) can be rewritten as

$$\mathbf{x}_i(k+1) = \bar{x}_i(k) + \begin{cases} \sum_{j\in\mathcal{M}_i} B_{ii}w_{ij}^1(k) + \sum_{j\in\mathcal{N}_i} B_{ij}w_j^1(k) \\ \sum_{j\in\mathcal{M}_i} B_{ii}w_{ij}^2(k) + \sum_{j\in\mathcal{N}_i} B_{ij}w_j^2(k) \\ \quad\vdots \qquad\qquad\qquad \vdots \\ \sum_{j\in\mathcal{M}_i} B_{ii}w_{ij}^{n_s}(k) + \sum_{j\in\mathcal{N}_i} B_{ij}w_j^{n_s}(k). \end{cases} \tag{8}$$

That is, $\mathbf{x}_i(k)$ aggregates subsystems' i state at time instant k for all possible scenarios, i.e., $\mathbf{x}_i(k) = [x_i^d(k)]_{d\in[1,\ldots,n_s]}$.

The control action computed in scenario-based MPC is optimized to account for possible scenarios [3,18], which can be used to obtain robustness regarding closed-loop constraint satisfaction, e.g., following [4]. For this purpose, the performance of each scenario is defined as a quadratic function $\ell_i^d(\cdot)$. Accordingly, the stage cost $\ell_i(\cdot)$ for each subsystem i is calculated as the weighted sum of the performance cost of each of the scenarios, where the weights are the probabilities of their occurrence. As mentioned before, we assume that all scenarios have the same probability $p_i = 1/n_s$:

$$\ell_i(k) = \sum_{d=1}^{n_s} p_i\ell_i^d(k), \tag{9}$$

where $\sum_{\forall n_s} p_i = 1$ and

$$\begin{aligned} \ell_i^d(k) &= u_i^{\rm pr^T}(k)R_i^{\rm pr}u_i^{\rm pr}(k) + \textstyle\sum_{j\in\mathcal{N}_i} u_{ji}^{\rm pu^T}(k)R_i^{\rm pu}u_{ji}^{\rm pu}(k) + \\ &\left(x_i^d(k+1) - x_i^{\rm ref}(k+1)\right)^{\rm T} Q_i\left(x_i^d(k+1) - x_i^{\rm ref}(k+1)\right). \end{aligned} \tag{10}$$

Also, weighting matrices Q_i, $R_i^{\rm pr}$, $R_i^{\rm pu}$ are positive definite and satisfy $R_i^{\rm pu} >> R_i^{\rm pr}$ to incentivize the use of local resources first. Finally, $x_i^{\rm ref}(k)$ represents the desired reference for subsystem i in time instant k.

Considering (9), the centralized MPC optimization problem at each time instant k is expressed by:

$$\left[\mathbf{U}_i^*(k)\right]_{i\in\mathcal{N}} = \arg\min_{[\mathbf{U}_i(k)]_{i\in\mathcal{N}}} \sum_{t=0}^{N_{\rm p}-1}\sum_{i\in\mathcal{N}} \ell_i(t) + \sum_{i\in\mathcal{N}} f_i$$

s.t.

$$x_i^d(0) = x_i(k), \tag{11a}$$

$$x_i^d(t+1) = \bar{x}_i(t) + \sum_{j\in\mathcal{M}_i} B_{ii}w_{ij}^d(t) + \sum_{j\in\mathcal{N}_i} B_{ij}w_j^d(t), \tag{11b}$$

$$u_i(t) = u_i^{\rm pr}(t) + \sum_{j\in\mathcal{M}_i} u_{ij}^{\rm pu}(t) \tag{11c}$$

$$u_i^{\rm pr}(t) \in \alpha_i\mathcal{U}_i, \tag{11d}$$

$$u_{ji}^{\rm pu}(t) \in \alpha_{ji}\mathcal{U}_j, \ \forall j \in \mathcal{N}_i, \tag{11e}$$

$$x_i^d(t) \in \mathcal{X}_i, \tag{11f}$$

$$\forall i \in \mathcal{N}, \tag{11g}$$

$$\forall d \in [1, 2, \ldots, n_s], \tag{11h}$$

$$\forall t = 0, \ldots, N_{\rm p} - 1. \tag{11i}$$

where $N_{\rm p}$ is the prediction horizon, and $\mathbf{U}_i(k)$ is formed by the sequence of control actions $u_i^{\rm pr}(\cdot)$ and $u_{ji}^{\rm pu}(\cdot)$ from instants t to $t + N_{\rm p} - 1$ and scale factors α_i and α_{ji}. Notice that variables $u_i^{\rm pr}(\cdot)$ and $u_{ji}^{\rm pu}(\cdot)$ vary along the prediction horizon while α_i and α_{ji} are kept constant in the optimization. Additionally, function f_i penalizes the values α_i and α_{ji}, for all $i \in \mathcal{N}$ and $\forall j \in \mathcal{N}_i$, i.e.,

$$f_i = \rho^{\rm pr}\alpha_i + \sum_{j\in\mathcal{N}_i} \rho^{\rm pu}\alpha_{ji}, \tag{12}$$

where $\rho^{\rm pr}$ and $\rho^{\rm pu}$ are positive weighting factors. The solution of (11) provides for each $i \in \mathcal{N}$ the value of the private and public inputs to be implemented at instant k, i.e., $u_i^{\rm pr}(k) = u_i^{\rm pr,*}(0)$ and $u_{ji}^{\rm pu}(k) = u_{ji}^{\rm pu,*}(0)$, and the corresponding optimal scale factors, i.e., $\alpha_i(k) = \alpha_i^*$ and $\alpha_{ji}(k) = \alpha_{ji}^*$, for all $j \in \mathcal{N}_i$. At the same time, according to Eq. (3), agents will take into account the private part of neighboring agents $u_j^{\rm pr}(k)$ and the public part of their local variable $u_{ij}^{\rm pu}(k)$ that is controlled by the agents $j \in \mathcal{M}_i$ affected by it as bounded disturbances. Thanks to the scenario-based MPC approach, $n_{\rm s}$ possible realizations of the disturbances will be considered in the optimization problem according to (8). Below, optimization problem (11) is distributed among the set of MPC agents by using the dual decomposition method. That is, the centralized problem is not meant to be directly solved.

3.2. Agents negotiation based on dual decomposition

Agents affected by the same input $u_i(k)$ need to negotiate the value of the local variables $\{\alpha_i(k), \alpha_{ij}(k)\}$ that scale the input constraints sets. Condition (4) must be satisfied by agents $\{i, \forall j \in \mathcal{M}_i\}$ that carry out the negotiation. To compute the solution in a distributed manner we apply the dual decomposition algorithm described in [8]. Convergence to the centralized solution is attained throughout an iterative procedure in which Lagrange multipliers $\lambda_i(k)$ are applied. The satisfaction of constraint (4) is achieved by incorporating Lagrange multipliers into local objective functions, which are constant along the prediction horizon but vary over time. The goal of the agents involved in the negotiation is to satisfy:

$$\lambda_i(k)\left(\alpha_i(k) + \sum_{j\in\mathcal{M}_i} \alpha_{ij}(k) - 1\right) \le 0 \quad \text{with} \quad \lambda_i(k) \ge 0. \tag{13}$$

Notice that every local objective function will contain as many Lagrange multipliers as the number of neighbors' inputs that affect their dynamics, as well as the one that characterizes its own input. In other words, the corresponding Lagrange multiplier for its private and public input variables. Eq. (14) describes how these auxiliary variables are considered in the performance function of subsystem i:

$$\Lambda_i = \lambda_i\alpha_i + \sum_{j\in\mathcal{N}_i} \lambda_j\alpha_{ji}. \tag{14}$$

Finally, we formulate the objective function of each subsystem i as the sum of the stage cost $\ell_i(\cdot)$, penalization of scale factors f_i and the corresponding terms for the dual decomposition algorithm Λ_i, i.e.,

$$\mathbf{J}(x_i, \mathbf{U}_i, [\lambda_m]_{m\in\mathcal{S}}) = \sum_{t=0}^{N_{\rm p}-1} \ell_i(t) + f_i + \Lambda_i. \tag{15}$$

Therefore, at each iteration, agent i solves the following problem:

$$\min_{\mathbf{U}_i} \max_{[\lambda_m]_{m\in\mathcal{S}_i}} \quad \mathbf{J}(x_i, \mathbf{U}_i, [\lambda_m]_{m\in\mathcal{S}_i})$$

where $\mathcal{S}_i = i \cup \mathcal{N}_i$, subject to the following constraints:

$$x_i^d(0) = x_i(k), \tag{16a}$$

$$(3), (8) \tag{16b}$$

$$u_i^{\rm pr}(t) \in \alpha_i\mathcal{U}_i, \;\; u_{ji}^{\rm pu}(t) \in \alpha_{ji}\mathcal{U}_j, \;\; \forall j \in \mathcal{N}_i, \tag{16c}$$

$$x_i^d(t) \in \mathcal{X}_i, \tag{16d}$$

$$\forall d \in [1, 2, \ldots, n_{\rm s}], \tag{16e}$$

$$\forall t = 0, \ldots, N_{\rm p} - 1, \tag{16f}$$

$$\lambda_m \ge 0, \;\; \forall m \in \mathcal{S}_i. \tag{16g}$$

Agents carry out an iterative negotiation where the values of the variables involved in (14) are compared to those at the previous iteration. Convergence is attained when the values in two consecutive iterations p and $p-1$ are similar enough, measuring the difference with: $\Delta = \alpha^p(k) - \alpha^{p-1}(k)$. We establish a threshold ϵ where the negotiation will stop: $\Delta \le \epsilon$. Lagrange multipliers must be updated in every step of the iterative procedure according to the following expression:

$$\lambda_i^{p+1}(k) = \lambda_i^p(k) + \gamma\left(\alpha_i^p(k) + \sum_{j\in\mathcal{M}_i} \alpha_{ij}^p(k) - 1\right), \tag{17}$$

being $\gamma > 0$ the step size. Note that the scale factors $\alpha_i(k)$ and $\alpha_{ij}(k)$ in (13) are the optimal solution of (16) at instant k and iteration p.

Considering the above, Algorithm 1 presents a pseudo-code with the steps that have to be followed by the different agents to implement the proposed coalitional MPC scheme. The idea behind it is to reduce communication in comparison with the fully distributed MPC approach. Coordination between agents will be event-based, as communication links will be disabled if Step 2 of the algorithm is verified. We consider that if the public part that an agent can manipulate is small in two consecutive time steps, there is no need for communication. To avoid recomputing an optimization, the implementation assumes that $u_{ji}^{\rm pu}(k) = 0$, but note that an extra step could be included here to re-optimize imposing that the public variable is zero. However, if at a single time step an

Algorithm 1 Control Scheme.

At each sample time k, each agent $i \in \mathcal{N}$ proceeds as follows:

1: Solve the optimization problem (16) for $n_{\rm s}$ equiprobable scenarios and find optimal sequence $u_i^{\rm pr}(k), u_{ji}^{\rm pu}(k)$ and optimal bounds of the variables $\alpha_i(k), \alpha_{ji}(k)$, $\forall j \in \mathcal{N}_i$.
2: **if** $\alpha_{ji}(k) < \bar{\alpha}$ and $\alpha_{ji}(k-1) < \bar{\alpha}$ **then**
3:     Disable communication flag: $f_{ji} = 0$.
4:     Ignore the value of the public variable: $u_{ji}^{\rm pu}(k) = 0$.
5: **else**
6:     Enable communication flag: $f_{ji} = 1$.
7: **end if**
8: **if** $f_{ji} = 1$ **then**
9:     **while** $\Delta > \epsilon$ **do**
10:         Update local and neighboring Lagrange multipliers $\{\lambda_i(k),\ \lambda_j(k) :\ \forall j \in \mathcal{N}_i\}$ according to (17).
11:         Compute $\Delta = \max\{\|\alpha_j^p(k) - \alpha_j^{p-1}(k)\|, \|\alpha_{ji}^p(k) - \alpha_{ji}^{p-1}(k)\|\}\ \forall j \in \mathcal{N}_i$.
12:         Set $p \leftarrow p+1$.
13:     **end while**
14: **end if**

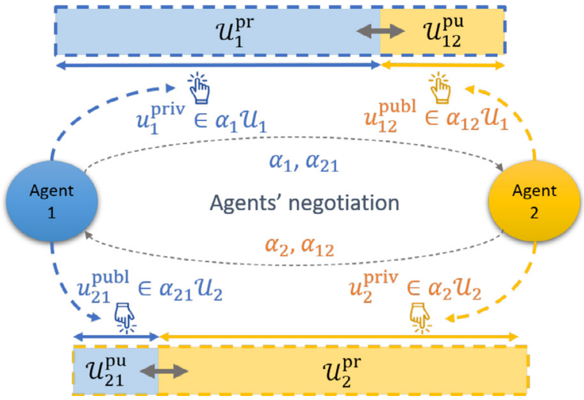


Fig. 1. Scheme of the proposed variable decomposition for two agents. Each agent i manipulates its private variable u_i^{pr} and the public input u_{ji}^{pu} ceded by its neighbor in such a way that all optimized variables belong to scaled constraints sets to guarantee overall constraints satisfaction. Likewise, the agents can negotiate these scale factors.

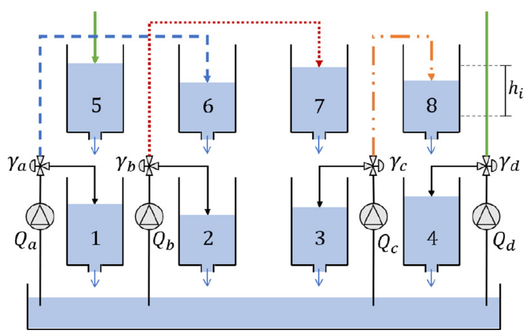


Fig. 2. Scheme of the eight tanks system. The colored pipes represent the coupling connections.

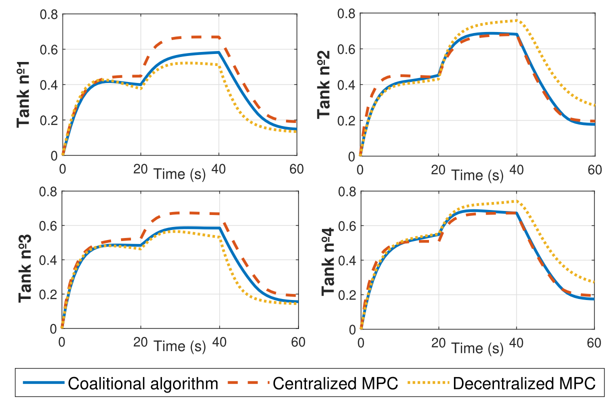
agent demands a greater use of the public variable, being its value above the threshold $\bar{\alpha}$, communication is retaken.

When communication occurs, agents follow a distributed approach according to the dual decomposition algorithm. They share the values of their local variables, updating the corresponding Lagrange multiplier until convergence is attained. It is worth mentioning that agents keep calculating public variables and bounds $\alpha_{ji}(k)$ even when they are not communicating with their neighbors in case negotiation should be resumed.

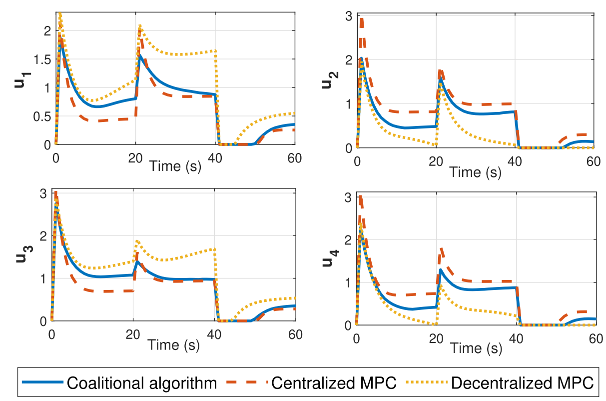
4. Simulation results

An eight input-coupled tanks plant has been used as an example to illustrate the behavior of the proposed technique. The system is composed by four top tanks (5, 6, 7, and 8) that discharge into four bottom tanks (1, 2, 3, and 4). In turn, tanks in the lower level discharge into a shared storage tank. The eight tanks are filled thanks to 4 pumps (Q_a, Q_b, Q_c and Q_d) that carry water from the storage tank. Moreover, four three-way valves are employed to regulate the pumped flow, dividing it into two ways to fill the upper and lower tanks.

We can distinguish $N = 4$ subsystems formed by a top and bottom tank. In this way, tanks 1 and 5 define the first subsystem; the second is composed of tanks 2 and 6; tanks 3 and 7 describe the third subsystem; finally, tanks 4 and 8 form the fourth one. These subsystems are physically coupled through the colored pipes that interconnect the tanks in Fig. 2. Also, the following bidirectional links, i.e., data connections between the corresponding local controllers, are considered: (1,2), (2,3), (3,4), and (4,1). Here, link (i, j) denotes that local controllers i and j can exchange information.



(a) Subsystems' states



(b) Subsystems' inputs

Fig. 3. Evolution of the states of the lower tanks and subsystems' inputs using $\bar{\alpha} = 0.05$ and $n_s = 20$. Solids lines represent the result using the proposed coalitional algorithm, while dashed and dotted lines represent the evolution using a centralized and decentralized MPC respectively.

The objective is to regulate the four lower tanks towards their desired reference in the water level. For this purpose, we define the state of each subsystem as the water level in meters of the two tanks that represent it, e.g., for subsystem one: $x_1 = [h_1 \; h_5]^T$. Furthermore, inputs are given by the value of the pump flow in cubic meters per hour $u_i = Q_k$, with $i = 1, \ldots, N$ and $k \in \{a, b, c, d\}$. Besides, the system is subject to the following constraints: $0.2 \leq x_i \leq 1.3$, $[0, 0, 0, 0] \leq u_i \leq [3.26, 4, 3.6, 4]$, with $i = 1, \ldots, N$. Lastly, each subsystem is characterized by the following matrices:

$$A_{11} = \begin{bmatrix} 0.8257 & 0.1178 \\ 0 & 0.8703 \end{bmatrix}, B_{11} = \begin{bmatrix} 0.0379 \\ 0 \end{bmatrix}, B_{14} = \begin{bmatrix} 0.0056 \\ 0.0843 \end{bmatrix},$$

$$A_{22} = \begin{bmatrix} 0.8163 & 0.1023 \\ 0 & 0.8867 \end{bmatrix}, B_{22} = \begin{bmatrix} 0.0503 \\ 0 \end{bmatrix}, B_{21} = \begin{bmatrix} 0.0053 \\ 0.0916 \end{bmatrix},$$

$$A_{33} = \begin{bmatrix} 0.8232 & 0.1077 \\ 0 & 0.8813 \end{bmatrix}, B_{33} = \begin{bmatrix} 0.0442 \\ 0 \end{bmatrix}, B_{32} = \begin{bmatrix} 0.0047 \\ 0.0783 \end{bmatrix},$$

$$A_{44} = \begin{bmatrix} 0.8194 & 0.1050 \\ 0 & 0.8840 \end{bmatrix}, B_{44} = \begin{bmatrix} 0.0441 \\ 0 \end{bmatrix}, B_{43} = \begin{bmatrix} 0.0050 \\ 0.0849 \end{bmatrix}.$$

The proposed scheme has been simulated using as weighting matrices: $Q_i = \text{diag}(4, 1)$, $R_i^{pr} = 0.1$, $R_i^{pu} = 5R_i^{pr}$. As well, the penalization of scale factors in (12) has been defined as $\rho^{pr} = 0.1R^{pr}$ and $\rho^{pu} = 0.1R^{pu}$. The prediction horizon has been set to $N_p = 10$, and $n_s = 20$ scenarios have been generated following a random uniform distribution.

Fig. 3 a shows the evolution of the states towards their desired references. In Fig. 3b, the performance of Algorithm 1 is shown in

Table 1
Comparison of the performance costs for different modes of operation.

	Performance cost **P**
Centralized MPC	39.4458
Coalitional algorithm with $\bar{\alpha} = 0.05$	43.4423
Coalitional algorithm with $\bar{\alpha} = 0.10$	46.1769
Decentralized MPC	78.5676

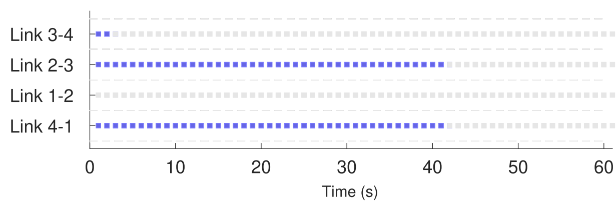


Fig. 4. Communication topology by means of the enable/disable of links using $\bar{\alpha} = 0.05$.

terms of the evolution of subsystems' inputs. As can be seen, when using $\bar{\alpha} = 0.05$ input and state trajectories with the proposed coalitional algorithm follow up closely the centralized solution. To allow a comparison between different MPC schemes, we consider the following index to evaluate the system performance:

$$\mathbf{P} = \sum_{k=1}^{T} \sum_{i \in \mathcal{N}} \|u_i(k)\|_{R_i^{\text{pr}}} + \|x_i(k+1) - x_i^{\text{ref}}(k+1)\|_{Q_i}$$

being T the simulation length and defining $u_i(k)$ in the coalitional approach according to (3). Table 1 shows a comparison of the overall performance when using various control schemes. We have provided two results for the coalitional algorithm using different values of $\bar{\alpha}$. In particular, for this simulation is considered that the optimum value of $\bar{\alpha}$ is 0.05, resulting in a decrease of the overall performance of a 10.13% concerning the centralized solution. At the same time, when using $\bar{\alpha} = 0.10$ the loss in performance is a 17.06%. As we want to reduce communication compared to the centralized approach, a value of the communication cost has been provided. This cost is calculated as the total number of active links during the entire simulation. For this purpose, all links are always active in the centralized solution, while in the coalitional scheme they enable or disable according to Step 2 in Algorithm 1. The value of this communication cost for centralized MPC is 240, while for the coalitional algorithm is 84 using $\bar{\alpha} = 0.05$ and 71 when $\bar{\alpha} = 0.10$. Using $\bar{\alpha} = 0.05$, communication cost reduces in a 65% with respect the centralized approach, while with $\bar{\alpha} = 0.10$ drops to a 70%. For this simulation, choosing a higher value of $\bar{\alpha}$ results in worse performance, but lower communication exchange. Furthermore, Fig. 4 represents the status of the communication links throughout the simulation for the optimal value of $\bar{\alpha} = 0.05$.

Fig. 5 shows the evolution of the optimization variables of subsystem 1 to clarify the concepts of public and private variables. Variable u_1^{pr} is represented with a continuous blue line, while α_1 is represented below it in purple; note that α_1 scales the constraints of the private part of u_1. For this reason, the product $\alpha_1 \mathcal{U}_1$ has been represented in the graph with a purple dashed line. The same applies to u_{41}^{pu}, which is the public part of input u_4 that can be manipulated by subsystem 1. Variable α_{41} scales the constraints of the public variable, and the product $\alpha_{41}\mathcal{U}_4$ is represented in a gray dashed line together with the evolution of u_{41}^{pu}. In this case, when the value of α_{41} is below the red dotted line $\bar{\alpha} = 0.05$, the value of u_{41}^{pu} is set to zero because the communication link is disabled. In accordance with Fig. 4, it can be seen that link 4-1 disables at time instant $k = 41$.

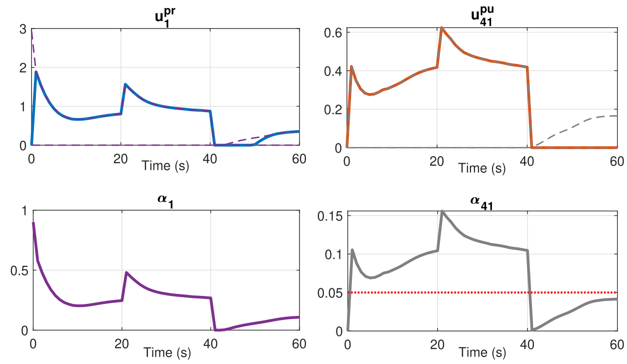


Fig. 5. Evolution of the optimization variables of the first subsystem $\mathbf{u_1} = \{u_1^{\text{pr}}, u_{41}^{\text{pu}}, \alpha_1, \alpha_{41}\}$.

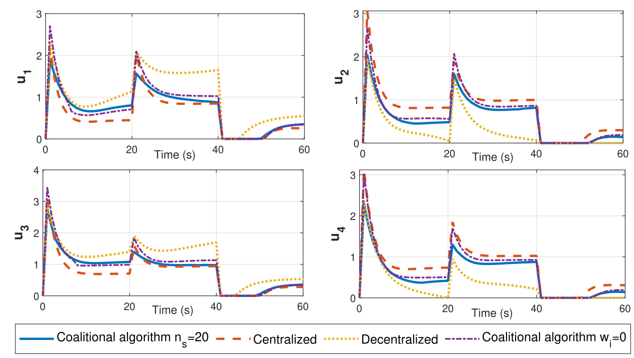


Fig. 6. Comparison of the evolution of subsystems' inputs when not considering scenarios for the fixed value of $\bar{\alpha} = 0.05$. Regarding to Fig. 3b the purple dot-dashed line represents the case where $w_i = 0$.

Finally, Fig. 6 illustrates a less conservative approach where agents neglect the possible coupling with their neighbors, meaning that they use (5) as prediction model with w_i set to zero during the entire prediction horizon. Inputs reach higher values than before, increasing the need for communication. This was expected since in Step 2 of Algorithm 1 we are applying the same threshold $\bar{\alpha} = 0.05$ and inputs values are greater than with $n_s = 20$. Communication cost is 61.9% higher than the one obtained when using scenarios. Furthermore, the cumulative cost reaches a value of $\mathbf{P} = 46.1624$, which supposes a 6.26% increase.

Being less conservative leads to an evolution of the input that also follows closely the centralized trajectory, but it increases communication. (see Fig. 6). As we were considering random equiprobable scenarios, the controller had to calculate a trajectory suited to all the possibilities. The risk of not considering uncertainties is that a sudden input value may violate the system's constraints.

5. Conclusions

In this paper, we propose a coalitional MPC approach by partitioning coupling variables. These variables share a common constraint space, where agents can negotiate in a distributed manner their limits if needed. The different inter-agent interaction modes permit flexible event-based communication, where agents coordinate only when they require it for the sake of better overall performance. To illustrate the proposed scheme, numerical results in an eight-input-coupled tanks benchmark show that this approach results in a performance close to the centralized solution while saving communication burden. Further research will focus on adapting this heuristic method to robust control, providing conditions for stability.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] P.R. Baldivieso-Monasterios, P.A. Trodden, Coalitional predictive control: consensus-based coalition forming with robust regulation, Automatica 125 (2021) 109380.
[2] A.L. Bella, F. Bonassi, M. Farina, R. Scattolini, Two-layer model predictive control of systems with independent dynamics and shared control resources, in: 15th IFAC Symposium on Large Scale Complex Systems LSS 2019, volume 52, 2019, pp. 96–101.
[3] D. Bernardini, A. Bemporad, Scenario-based model predictive control of stochastic constrained linear systems, in: Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, 2009, pp. 6333–6338.
[4] G. Calafiore, L. Fagiano, Robust model predictive control via scenario optimization, IEEE Trans. Autom. Control 58 (2013) 219–224.
[5] G. Chaloulos, P. Hokayem, J. Lygeros, Distributed hierarchical MPC for conflict resolution in air traffic control, in: Proceedings of the 2010 American Control Conference, 2010, pp. 3945–3950.
[6] P. Chanfreut, J.M. Maestre, E.F. Camacho, Coalitional model predictive control on freeways traffic networks, IEEE Trans. Intell. Transp. Syst. (2020).
[7] P. Chanfreut, J.M. Maestre, E.F. Camacho, A survey on clustering methods for distributed and networked control systems, Ann. Rev. Control 52 (2021) 75–90, doi:10.1016/j.arcontrol.2021.08.002.
[8] F. Farokhi, I. Shames, K.H. Johansson, Distributed MPC via dual decomposition and alternative direction method of multipliers, in: Distributed model predictive control made easy, Springer, 2014, pp. 115–131.
[9] F. Fele, E. Debada, J.M. Maestre, E.F. Camacho, Coalitional control for self-organizing agents, IEEE Trans. Autom. Control 63 (9) (2018) 2883–2897.
[10] F. Fele, J.M. Maestre, E.F. Camacho, Coalitional control: cooperative game theory and control, IEEE Control Syst. 37 (1) (2017) 53–69.
[11] F. Fele, J.M. Maestre, S.M. Hashemy, D.M. de la Peña, E.F. Camacho, Coalitional model predictive control of an irrigation canal, J. Process Control 24 (4) (2014) 314–325.
[12] R. Guicherd, P.A. Trodden, A.R. Mills, V. Kadirkamanathan, Supervised-distributed control with joint performance and communication optimisation, Int. J. Control (2020) 1–12.
[13] A. Jain, A. Chakrabortty, E. Biyik, Distributed wide-area control of power system oscillations under communication and actuation constraints, Control Eng. Pract. 74 (2018) 132–143.
[14] A. La Bella, P. Klaus, G. Ferrari-Trecate, R. Scattolini, Supervised model predictive control of large–scale electricity networks via clustering methods, Optim. Control Appl. Methods 43 (2021) 44–64.
[15] W. Langson, I. Chryssochoos, S. Raković, D. Mayne, Robust model predictive control using tubes, Automatica 40 (1) (2004) 125–133.
[16] J. Maestre, M. Ridao, A. Kozma, C. Savorgnan, M. Diehl, M. Doan, A. Sadowska, T. Keviczky, B. De Schutter, H. Scheu, et al., A comparison of distributed MPC schemes on a hydro-power plant benchmark, Optim. Control Appl. Methods 36 (3) (2015) 306–332.
[17] Distributed Model Predictive Control Made Easy, in: J.M. Maestre, R.R. Negenborn (Eds.), Intelligent Systems, Control and Automation: Science and Engineering, Springer, 2014.
[18] M. Maiworm, T. Bäthge, R. Findeisen, Scenario-based model predictive control: recursive feasibility and stability, IFAC-PapersOnLine 48 (8) (2015) 50–56. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015
[19] D.Q. Mayne, M.M. Seron, S.V. Raković, Robust model predictive control of constrained linear systems with bounded disturbances, Automatica 41 (2) (2005) 219–224.
[20] R.R. Negenborn, J.M. Maestre, Distributed model predictive control: an overview and roadmap of future research opportunities, IEEE Control Syst. 34 (4) (2014) 87–97.
[21] P. Neumann, Communication in industrial automation–what is going on? Control Eng. Pract. 15 (11) (2007) 1332–1347.
[22] S.J. Qin, T.A. Badgwell, A survey of industrial model predictive control technology, Control Eng. Pract. 11 (7) (2003) 733–764.
[23] J. Richalet, A. Rault, J.L. Testud, J. Papon, Model predictive heuristic control: applications to industrial processes, Automatica 14 (2) (1978) 413–428.
[24] P.A. Trodden, J.M. Maestre, Distributed predictive control with minimization of mutual disturbances, Automatica 77 (2017) 31–43.
[25] A.N. Venkat, I. Hiskens, J.B. Rawlings, S.J. Wright, et al., Distributed MPC strategies with application to power system automatic generation control, IEEE Trans. Control Syst. Technol. 16 (6) (2008) 1192–1206.