No-Regret Learning for Coalitional Model Predictive Control [⋆]

Chanfreut, P. [*] Maestre, J.M. [*] Zhu, Q. [**] Camacho, E.F. [*]

[*] *Systems and Automation Engineering Department, University of Seville, Spain (e-mail: {pchanfreut,pepemaestre,efcamacho}@us.es).*
[**] *Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University (e-mail: qz494@nyu.edu)*

Abstract: In this paper, we introduce a learning approach for the controller structure in coalitional model predictive control (MPC) schemes. In this context, the local control entities can dynamically perform in a decentralized manner or assemble into groups of controllers that coordinate their control actions, i.e., *coalitions*. Such control strategy aims at maximizing system performance while reducing the coordination and computation burden. In this paper, we pose a multi-armed bandit problem where the *arms* are a set of possible controller structures and the *player* performs as a supervisory layer that can periodically change the composition of the coalitions. The goal is to use real-time observations to progressively learn the controller structure that best suits the needs of the system. A heuristic learning algorithm and illustrative results are provided.

Keywords: Coalitional model predictive control, multi-armed bandits.

1. INTRODUCTION

Distributed model predictive control (DMPC) has received significant attention by the research community in the last years (Christofides et al. (2013); Negenborn and Maestre (2014)). The inherent properties of DMPC schemes, such as its modularity and scalability, make it an attractive control methodology to address the challenges presented by large-scale networked systems, where the application of the classical centralized MPC is limited. In this context, the decision-making capacity is distributed among a set of local controllers or agents which operate with a certain degree of autonomy. Due to the dynamical couplings, the optimality of the local control actions hinges on the rest of the system behavior and other agents' decisions, hence coordination allows for increasing the global performance. However, the benefits of greater coordination come at the expense of a higher communication and computational burden. Recently, controllers of variable structure have gained importance for addressing these drawbacks, e.g., Jain et al. (2018), Zheng et al. (2018). Within this framework, we focus on coalitional MPC (Fele et al. (2017)), a novel strategy that allows transitions from dense to sparse communication scenarios by fostering the formation of *coalitions* of control entities, i.e., clusters of agents that can share data and coordinate their actions, normally for a control objective that comprises the individual goals of the merged controllers.

Optimizing the coordination scenario needs for further computation and requires the existence of a performance index able to predict the future suitability of each possibility. See Fele et al. (2014) for an example using an upper bound on the cost-to-go. In this paper, we aim at introducing online learning solutions for the adaptation of the controller structure in coalitional MPC schemes. In particular, we pose a multi-armed bandit problem (MAB) (see Bubeck et al. (2012); Auer et al. (2002) among many others), where the *arms* are a set of possible coordination scenarios, and the *player* (or *learner*) acts as a supervisory layer that should learn to optimize the controller structure. In its simplest formulation, a MAB problem is modeled as a sequential game where, each round, the *player* pulls one arm from a finite decision set and subsequently received a loss, which depends on the optimality of its choice. Just the loss associated with the chosen arm is observed, while the behavior of the others remains unknown. The learner's goal is minimizing its long-term loss while balancing exploration, i.e., trying out different actions to gather information, and exploitation, i.e., playing the arm believed to be optimal. Within this class of learning problems, we focus on *contextual* bandits (Slivkins (2014)). In this setting, the player receives some relevant data in the form of a *context* vector to make its decision. One of its main applications is the selection of web's content (e.g., Chu et al. (2011)), where the *context* contains meaningful information such as the users previous Internet queries.

The rest of the paper is organized as follows. Section II introduces the system and states the control problem. Section III focuses on the learning procedure for selecting the communication topologies. Section IV describes the control scheme and Section V illustrates the proposed scheme through an academic example. Finally, conclusions and further research are provided in Section VI.

[⋆] This work is partially supported by the Spanish Training program for Academic Staff (FPU17/02653), the H2020 ADG-ERC project OCONTSOLAR (ID 789051), the MINECO-Spain project DPI2017-86918-R, and the US National Science Foundation (NSF) under grant ECCS-1847056 and CNS-1544782.

2. PROBLEM FORMULATION.

In this section, we present the model describing the system dynamics, the communication infrastructure, and the control problem we address throughout this work.

2.1 System dynamics.

Consider a class of linear systems that can be partitioned into a set $\mathcal{N} = \{1, 2, \ldots, N\}$ of coupled subsystems, whose dynamics are modelled as follows

$$\begin{aligned} x_i(k+1) &= A_{ii}x_i(k) + B_{ii}u_i(k) + d_i(k), \\ d_i(k) &= \sum_{j\in\mathcal{N}_i} \left[A_{ij}x_j(k) + B_{ij}u_j(k)\right], \end{aligned} \tag{1}$$

where $x_i \in \mathbb{R}^n$ and $u_i \in \mathbb{R}^m$ are respectively the state and input vector of subsystem $i \in \mathcal{N}$, and d_i describes the coupling among subsystem i and its set of neighbours, defined as $\mathcal{N}_i = \{j \in \mathcal{N} \setminus \{i\} \mid [A_{ij}, B_{ij}] \neq \mathbf{0}\}$ [1] . Matrices $A_{ij} \in \mathbb{R}^{n\times n}$ and $B_{ij} \in \mathbb{R}^{n\times m}$ are, respectively, the state transition and the input-to-state matrices for all $i, j \in \mathcal{N}$.

2.2 Network structure

Consider a communication network described by graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where the nodes represent local agents and the set of edges $\mathcal{L}$ contains communication links. Hereafter, we consider that the state of these links can dynamically switch between *enabled* and *disabled.* In particular, symbol Λ denotes different communication *topologies*, i.e., combinations of enabled links. Note that if the cardinality of $\mathcal{L}$ is $|\mathcal{L}|$, then, we can derive a set $\mathcal{T}$ of $2^{|\mathcal{L}|}$ possible communication topologies:

$$\mathcal{T} = \{\Lambda_0, \Lambda_1, \cdots, \Lambda_{2^{|\mathcal{L}|}-1}\}, \tag{2}$$

where each Λ_i partitions the set of agents into a set $\mathcal{N}/\Lambda_i$ of *coalitions*, i.e., group of agents connected by a path of enabled links. The coalitions are denoted by $\mathcal{C}$ and we assume the following:

Assumption 1. Inside each coalition $\mathcal{C}$, the controllers share data and coordinate their actions while operating in a decentralized manner from the rest of the system, i.e., agents $i \in \mathcal{C}$ do not communicate with any agent $j \notin \mathcal{C}$.

Dinamically, any coalition can be considered as a single system modelled by

$$\begin{aligned} x_\mathcal{C}(k+1) &= A_{\mathcal{C}\mathcal{C}}x_\mathcal{C}(k) + B_{\mathcal{C}\mathcal{C}}u_\mathcal{C}(k) + d_\mathcal{C}(k), \\ d_\mathcal{C}(k) &= \sum_{\mathcal{D}\in\mathcal{N}_\mathcal{C}} \left[A_{\mathcal{C}\mathcal{D}}x_\mathcal{D}(k) + B_{\mathcal{C}\mathcal{D}}u_\mathcal{D}(k)\right], \end{aligned} \tag{3}$$

where $x_\mathcal{C} = [x_i]_{i\in\mathcal{C}}$ and $u_\mathcal{C} = [u_i]_{i\in\mathcal{C}}$ are respectively the aggregates of the states and inputs of the subsystems $i \in \mathcal{C}$, and matrices $A_{\mathcal{C}\mathcal{C}}$ and $B_{\mathcal{C}\mathcal{C}}$ map the current coalition state and inputs to its successor state. Similarly, $d_\mathcal{C}$ models the effect of neighboring coalitions $\mathcal{D} \in \mathcal{N}_\mathcal{C}$, where matrices $A_{\mathcal{C}\mathcal{D}}$, $B_{\mathcal{C}\mathcal{D}}$ and set $\mathcal{N}_\mathcal{C}$ are defined analogously to the case of single interacting subsystems. Note that if $\mathcal{C} = \mathcal{N}$ (i.e., $\Lambda = \mathcal{L}$), then $d_\mathcal{C} = \mathbf{0}$ because coupling is accounted for in the overall system model. In this respect, we will use $x_\mathcal{N}$ and $u_\mathcal{N}$ to refer to the global system state and input respectively.

2.3 Control objective

The control objective considered in this work is two-fold: optimizing the system performance and reducing coordination costs. Following an MPC approach, the global control problem at each time instant can be posed as

$$\begin{aligned} \min_{\mathbf{\Lambda}(k),\mathbf{u}_\mathcal{C}(k)} \quad & \sum_{n=0}^{N_\mathrm{p}-1} \sum_{\mathcal{C}\in\mathcal{N}/\Lambda(k+n)} \ell_\mathcal{C}(k+n) + \sum_{n=0}^{N_\mathrm{p}-1} c\,|\Lambda(k+n)| \\ \text{s.t.} \quad & x_\mathcal{C}(k+n+1) = A_{\mathcal{C}\mathcal{C}}x_\mathcal{C}(k+n) + \\ & \qquad B_{\mathcal{C}\mathcal{C}}u_\mathcal{C}(k+n) + d_\mathcal{C}(k+n), \\ & u_\mathcal{C}(k+n) \in \mathcal{U}_\mathcal{C}, \\ & x_\mathcal{C}(k+n+1) \in \mathcal{X}_\mathcal{C}, \\ & \forall \mathcal{C} \in \mathcal{N}/\Lambda(k+n), \\ & \forall n = 0, ..., N_\mathrm{p}-1. \end{aligned} \tag{4}$$

where function $\ell_\mathcal{C}(\cdot)$ is the stage performance index of coalition $\mathcal{C} \in \mathcal{N}/\Lambda(\cdot)$, $c > 0$ is the cost of enabling a link, and $|\Lambda(\cdot)|$ denotes the cardinality of $\Lambda(\cdot)$. Optimization variables $\mathbf{\Lambda}$ and $\mathbf{u}_\mathcal{C}$ are respectively the sequence of communication topologies and the coalitions control actions for a future time horizon of N_p instants. Additionally, $\mathcal{X}_\mathcal{C}$ and $\mathcal{U}_\mathcal{C}$ are respectively the state and input constraints sets of coalition $\mathcal{C}$. For simplicity, we assume that all subsystems should be regulated towards the origin. Hence, we can define the stage cost as

$$\ell_\mathcal{C}(k) = x_\mathcal{C}^\mathrm{T}(k)Q_\mathcal{C}x_\mathcal{C}(k) + u_\mathcal{C}^\mathrm{T}(k)R_\mathcal{C}u_\mathcal{C}(k), \tag{5}$$

Remark 1. Problem (4) constitutes a dynamic optimization problem with mixed integer variables, which restricts its applicability for real time control unless some simplifications are introduced. Note that variable $\mathbf{\Lambda}$ determines the composition of the coalitions and thus the definition of functions $\ell_\mathcal{C}(\cdot)$, where variables $\mathbf{u}_\mathcal{C}$ come into play.

Previous works on coalitional MPC provide algorithms that approximate the solution of Problem (4). Following Fele et al. (2014), we use a double sample rate strategy that decouples the optimization of $\mathbf{\Lambda}$ and $\mathbf{u}_\mathcal{C}$. In particular, decisions on the communication topology are just taken periodically. Once a certain Λ_i is imposed, the coalitions $\mathcal{C} \in \mathcal{N}/\Lambda_i$ calculate their control inputs through the following optimization problem:

$$\begin{aligned} \min_{\mathbf{u}_\mathcal{C}(k)} \quad & \sum_{n=0}^{N_\mathrm{p}-1} \ell_\mathcal{C}(k+n) \\ \text{s.t.} \quad & x_\mathcal{C}(k+n+1) = A_{\mathcal{C}\mathcal{C}}x_\mathcal{C}(k+n) + B_{\mathcal{C}\mathcal{C}}u_\mathcal{C}(k+n), \\ & u_\mathcal{C}(k+n) \in \mathcal{U}_\mathcal{C}, \\ & x_\mathcal{C}(k+n+1) \in \mathcal{X}_\mathcal{C}, \\ & \forall n = 0, ..., N_\mathrm{p}-1. \end{aligned} \tag{6}$$

The first action of the optimal inputs sequences $\mathbf{u}_\mathcal{C}^*(k)$ consitute the global input vector implemented at time k. In this paper, the couplings among different coalitions are neglected in the dynamic model used in (6), however, an estimate of $d_\mathcal{C}$ could be equally considered. Note that a good choice of the controller structure will minimize the inter-coalitions couplings and hence $d_\mathcal{C}$ will be small.

[1] External disturbances are not considered for simplicity. It is straight forward to extend the results of this paper to account for them.

3. ONLINE LEARNING FOR COALITIONS FORMATION

Hereon, we use the game-theoretic notions of *player* (or *learner*) and *arms* to denote respectively the decision-making entity and the set of possible topologies. The problem of switchings between communication topologies is modelled as a contextual multi-armed bandit problem, where at each game-round t, the player should choose, based on a given *context*, one communication topology out of the $2^{|\mathcal{L}|}$ possibilities in (2). The learner receives as *context* the system state at the time instant corresponding to round t, hereon, $x_{\mathcal{N},t}$. Each decision reports a *loss* that considers the suitability of the selected topology for operating the system. The goal is to learn from previous choices to improve current and future decisions.

3.1 Loss function

The *loss* function weights system performance and coordination costs. Note that the sum of (5) for a certain amount of steps can be expressed in terms of the current system state and of the sequence of control actions, and let us define $J_{\Lambda_i}(x_{\mathcal{N}}(k), \mathbf{u}_{\mathcal{N}}(k)) = \sum_{n=0}^{N'_\text{p}-1} \sum_{\mathcal{C}\in\mathcal{N}/\Lambda_i} \ell_{\mathcal{C}}(k+n)$, where $N'_\text{p} \geq N_\text{p}$ is the value of the prediction horizon and $\mathbf{u}_{\mathcal{N}}(k) = [\mathbf{u}_{\mathcal{C}}(k)]_{\mathcal{C}\in\mathcal{N}/\Lambda_i}$ is the aggregation of the coalitions' sequences of inputs. The window of time N'_p is set longer than the prediction horizon used by the local controllers because decisions on the communication topology remain unchaged for $T_\text{top} > N_\text{p}$ time instants; also, if the prediction horizon is too short, differences between topologies benefits may go unnoticed.

Definition 1. The loss reported when choosing topology Λ_i on round t is given by

$$L_{\Lambda_i,t} = J_{\Lambda_i}(x_{\mathcal{N},t}, [\mathbf{u}^*_{\mathcal{C},t}]_{\mathcal{C}\in\mathcal{N}/\Lambda_i}) + N'_\text{p}c|\Lambda_i|. \tag{7}$$

where $[\mathbf{u}^*_{\mathcal{C},t}]_{\mathcal{C}\in\mathcal{N}/\Lambda_i}$ is the optimal sequence of control actions resulting when solving Problem (6) for all coalitions $\mathcal{C} \in \mathcal{N}/\Lambda_i$ with prediction horizon N'_p.

Remark 2. Note that at each point $x_{\mathcal{N}}$ of the state space, the loss of any arm Λ_i takes a constant value, i.e., if $x_{\mathcal{N},t} = x_{\mathcal{N},t'}$, then, $L_{\Lambda_i,t} = L_{\Lambda_i,t'}$ (with $t \neq t'$).

3.2 Historical data

Let Λ_I be the player's decision on round t. Then, each round, the player observes context vector $x_{\mathcal{N},t}$, his decision Λ_I, and the incurred loss $L_{\Lambda_I,t}$. Hence, previous observations can help to learn the optimal mapping from state space $\mathcal{X}$ and the set of arms $\mathcal{T}$ (see Remark 2). To this end, we consider the data history

$$\mathcal{H}_t = \{x_{\mathcal{N},\tau}, \Lambda_{I_\tau}, L_{\Lambda_I,\tau}\}_{\tau=1}^{t}, \tag{8}$$

which is updated accordingly across game-rounds to bring together the data observed up to current time.

3.3 Evaluation of the player's performance

The learner performance is measured by *regret*, i.e., the diffrence between the player cumulative loss and the loss of playing the best action in hindsight, that is,

$$R_T = \sum_{t=1}^{T} L_{\Lambda_I,t} - \sum_{t=1}^{T} \min_{\Lambda_j\in\mathcal{T}} L_{\Lambda_j,t} \tag{9}$$

where T represents the total number of played rounds.

3.4 Optimal arm selection policy

In this paper, the dependence between the arm's benefits and the system state is used to establish some rules to guide the learning process and reduce the regret.

Firstly, based on context $x_{\mathcal{N},t}$, the player pre-selects a subset of arms from set $\mathcal{T}$ that are more likely to minimize the loss, say $\mathcal{T}_t$. For example, if the system state is close to the origin, we foster the choice of sparse controller structures. On the contrary case, denser communication scenarios are pre-selected.

Subsequently, the learner uses history $\mathcal{H}_{t-1}$ to estimate, when possible, the loss of the pre-selected arms. To this end, it searches for previous observations in the surroundings of point $x_{\mathcal{N},t}$, which, for simplicity, will be defined as the space enclosed by an hypersphere centered at $x_{\mathcal{N},t}$ with radius r, that is,

$$\mathcal{S}(x_{\mathcal{N},t}, r) = \{x \in \mathcal{X} \ : \ \|x - x_{\mathcal{N},t}\| \leq r\}. \tag{10}$$

Assumption 2. Given context $x_{\mathcal{N},t}$, the loss of topology Λ_i can be approximated by a quadratic function on the system state, i.e., $\hat{L}_{\Lambda_i} = x_{\mathcal{N},t}^\text{T} M_{\Lambda_i} x_{\mathcal{N},t} + m_{\Lambda_i} x_{\mathcal{N},t} + q_{\Lambda_i} + N'_\text{p}c|\Lambda_i|$, where M_{Λ_i} is a diagonal matrix, m_{Λ_i} a vector, and q_{Λ_i} a scalar that are calculated using previous observations in $\mathcal{S}(x_{\mathcal{N},t}, r)$.

Note that the accuracy of the estimation $\hat{L}_{\Lambda_i}$ decreases as the differences between the real context and the points used to calculate the estimated loss increase. Hence, reducing the searching radio r will improve the accuracy of learning and minimize long-term regret, but it will also reduce notably the learning rate, especially when the contexts received are too dissimilar and/or the number of arms is large. As a compromise, we consider two radius r_1 and r_2, with $r_1 < r_2$.

Assumption 3. Loss estimates based on observations within set $\mathcal{S}(x_{\mathcal{N},t}, r_1)$ are accurate enough to reliably identify the optimal topology.

Normally, we set r_2 as our searching radius, however, with probability p_{r_1}, it is reduced to r_1. Hence, across game-rounds, the player will be able to discard misleading loss estimates. Hereon, all topologies in $\mathcal{T}_t$ whose loss can be predicted with the information gathered up to round t will be grouped in set $\mathcal{T}_\text{o}$, where the subindex stands for *observed* topologies. Likewise, we will use $\mathcal{T}_\text{no}$ in the contrary case (*non-observed* topologies).

Additionally, considering the physical meaning of the arms, the player can capture further information. Let $\hat{L}^*_t$ be the minimum of the loss estimates on round t, and $\hat{\Lambda}^*_t$ the corresponding topology. Then, if $\mathcal{T}_\text{no} \neq \emptyset$, we use the following rules to foster or hinder the choice of certain topologies:

a) In case the player knows an estimation for the centralized topology, i.e., $\hat{L}_{\Lambda_\text{cen}}$, it uses the perfomance optimality in terms of system behaviour of this topology. Then, all Λ_i for which

$$\hat{L}_{\Lambda_\text{cen}} - N'_\text{p}c(|\Lambda_\text{cen}| - |\Lambda_i|) \geq \hat{L}^*_t \tag{11}$$

are discarded from $\mathcal{T}_\text{no}$.

b) In line with a), we consider a more heuristic criterion that uses similarities between arms and that assumes that as the system evolves to the origin, the needs for communication decrease. In particular, the probabilities of exploring arms that derive from $\hat{\Lambda}_t^*$, i.e., arms Λ_j in set $\tau_{\hat{\Lambda}_t^*} = \{\Lambda_j : \Lambda_j \in \mathcal{T}_{\text{no}}, \Lambda_j \subset \hat{\Lambda}_t^*\}$, are slightly increased over other options in $\mathcal{T}_{\text{no}}$.

Finally, the loss estimates and the hypothesis above are used to build a probability distribution

$$\mathbf{p}_t = [p_{\Lambda_0}, p_{\Lambda_1}, ..., p_{2^{|\mathcal{L}|}-1}]^{\mathrm{T}}, \tag{12}$$

where each p_{Λ_i} indicates the probability of playing arm Λ_i on round t. Then, higher values of p_{Λ_i} should be assigned to those arms that are more likely to minimize losses on the basis of the data observed up to round t.

Multi-class classification:

Over time, the player will discover regions in state space where some topology dominates the others with high probability, i.e., sets of points in $\mathcal{X}$ where a certain arm minimizes the incurred loss. Here, we consider multi-class classification techniques (Mayoraz and Alpaydin (1999)) to build in real-time a model that properly associates any context to one of the $2^{|\mathcal{L}|}$ classes, i.e., the communication topologies. In particular, this multiclass classification problem is reduced to multiple binary classification problems that should distinguish regions where each topology Λ_i is optimal. To build the classification models, we store the states where some p_{Λ_i} is notably greater than any other p_{Λ_j} (with $\Lambda_j \neq \Lambda_i$). The classification models are only updated periodically to reduce the computational burden. Hereafter, we will use the notation $\mathcal{D}_{\Lambda_i,t} \subseteq \mathcal{X}$ to denote the sets where topology Λ_i is dominant according the classification up to round t.

4. CONTROL SCHEME

In this section, the pseudocode of the coalitional control scheme is provided (Algorithm 1). Also, the logic for switching between topologies is described in Algorithm 2. In this respect, with some small probability p_{class} a classified context is considered as non-classified to review at some game-rounds the truthfulness of the classification models.

Algorithm 1. Control scheme: Let T_{top} be the number of time steps beween game rounds. Consider also set $\mathcal{K} = \{tT_{\text{top}} \mid t \in \mathbb{N}_{\geq 0}\}$ of time instants. Then, starting from round $t = 0$ with communication topology $\Lambda_0 \in \mathcal{T}$, at each sample time k:

```
if $k \in \mathcal{K}$ then
    $t = t + 1$
    All agents share their state so that context $x_{\mathcal{N},t}$ is communicated to the decision entity, i.e., the player.
    The player chooses a topology $\Lambda_t$ according to Algorithm 2.
end if
Within coalitions $\mathcal{C} \in \mathcal{N}/\Lambda_t$, the local controllers share their state and jointly solve Problem 6.
The resulting control actions $u^*_{\mathcal{C}}(k)$ are implemented by all $\mathcal{C} \in \mathcal{N}/\Lambda_t$.
```

Algorithm 2. Topology Λ_t selection: Each round t, the player follows the steps below:

```
Initialize $\Lambda_t = \emptyset$.
Check if $x_{\mathcal{N},t}$ is classified within some set $\mathcal{D}_{\Lambda_i}$, and with probability $1 - p_{\text{class}}$, select the corresponding topology, i.e., $\Lambda_t \leftarrow \Lambda_i$.
if $\Lambda_t = \emptyset$ then
    Pre-select subset of topologies $\mathcal{T}_t$.
    Search $\mathcal{H}_{t-1}$ for observations in set $\mathcal{S}(x_{\mathcal{N},t}, r_2)$.
    With probability $p_{r_1}$, discard observations outside $\mathcal{S}(x_{\mathcal{N},t}, r_1)$.
    Determine sets $\mathcal{T}_{\text{o}}$ and $\mathcal{T}_{\text{no}}$.
    For all $\Lambda_i \in \mathcal{T}_{\text{o}}$, calculate loss estimations $\hat{L}_{\Lambda_i}$ using previous observations to compute parameters $M_{\Lambda_i}$, $m_{\Lambda_i}$ and $q_{\Lambda_i}$.
    If $\hat{L}_{\Lambda_{\text{cen}}}$ is known, apply rule a) and update set $\mathcal{T}_{\text{no}}$.
    Construct probability vector $\mathbf{p}_t$ as
        $p_{\Lambda_i,t} = \frac{1}{|\mathcal{T}_{\text{no}}| + 1}, \quad \forall \Lambda_i \in \hat{\Lambda}_t^* \cup \mathcal{T}_{\text{no}},$   (13)
    and $p_{\Lambda_i,t} = 0$ for the rest of arms.
    Apply rule b). That is, $\mathbf{p}_t \leftarrow (1-\gamma)\mathbf{p}_t + \gamma \mathbf{v}$, where $\mathbf{v}$ is the uniform distribution among topologies $\Lambda_j \in \tau_{\hat{\Lambda}_t}$, and $\gamma \in [0, 1]$ is a weighting factor.
    Draw $\Lambda_t \sim \mathbf{p}_t$.
end if
Update history $\mathcal{H}_t$ and store context if $\max \mathbf{p}_t > \overline{p}$.
With period $T_{\text{class}} \geq T_{\text{top}}$, update the models for classification.
```

5. ACADEMIC EXAMPLE

In this section, we apply the coalitional scheme to a modified version of the system proposed in Farina and Scattolini (2012). It consists on five trucks, i.e., $\mathcal{N} = \{1, 2, 3, 4, 5\}$, coupled via springs and dumpers as shown in Fig. 1. The dynamic of each truck is modelled by:

$$\begin{bmatrix} \dot{s}_i \\ \dot{v}_i \end{bmatrix} = A_{ii} \begin{bmatrix} s_i \\ v_i \end{bmatrix} + B_{ii} u_i + \sum_{j \in \mathcal{N}_i} A_{ij} \begin{bmatrix} s_j \\ v_j \end{bmatrix} \tag{14}$$

where

$$A_{ii} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{m_i} \sum_{j \in \mathcal{N}_i} k_{ij} & -\frac{1}{m_i} \sum_{j \in \mathcal{N}_i} h_{ij} \end{bmatrix},$$

$$A_{ij} = \begin{bmatrix} 0 & 1 \\ \frac{1}{m_i} \sum_{j \in \mathcal{N}_i} k_{ij} & \frac{1}{m_i} \sum_{j \in \mathcal{N}_i} h_{ij} \end{bmatrix} \text{ and } B_{ii} = \begin{bmatrix} 0 \\ 50 \end{bmatrix}, \tag{15}$$

for all $i \in \mathcal{N}$. The state x_i of each subsystem is formed by the displacement s_i from the equilibrium point and by the instantaneous velocity v_i. Additionally, the agents can apply a longitudinal force $F_i = 50u_i$, where u_i is the control action. The parameters that characterize the system are given in Table 1. The continuous-time dynamics are discretized using zero-order hold and a sampling time of 0.1s. The goal is regulating the five trucks towards the origin while minimizing communication costs. In this

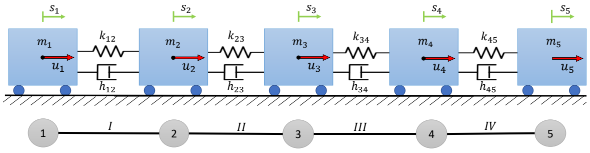


Fig. 1. Five trucks system and its communication network.

Table 1. Masses, spring stiffnesses, damping factors and further parameters used in the simulation.

Masses [kg]		Spring stiffnesses [N/m]		Dumping factors [N/(m·s)]	
m_1	3				
m_2	4	k_{12}	1.2	h_{12}	0.5
m_3	2	k_{23}	2.4	h_{23}	0.6
m_4	3	k_{34}	2.1	h_{34}	0.5
m_5	4	k_{45}	2	h_{45}	0.4

Further parameters									
N_p	1 s	r_1	0.05	T_top	1 s	p_{r_1}	0.2	p_class	0.2
N'_p	2 s	r_2	0.2	T_class	7 s	$\overline{p}$	0.8	γ	0.1

respect, the stage performance function $\ell_\mathcal{C}$ is defined by weighting matrices $Q_\mathcal{C} = \mathrm{diag}\,(Q_i)_{\forall i \in \mathcal{C}}$ and $R_\mathcal{C} = \mathrm{diag}\,(R_i)_{\forall i \in \mathcal{C}}$, where $Q_i = \mathbf{I}_2$ and $R_i = 100$, for all $i \in \mathcal{N}$, and the cost per enabled link c has been set at 0.01. The five agents are connected by a network of four bidirectional links, i.e., $\mathcal{L} = \{I, II, III, IV\}$ (see Fig. 1 and Table 2).

The control scheme described in Section III has been implemented using the parameters specified in Table 1 considering the following constraints: $|s_i| \leq 4$ and $|u_i| \leq 2$. For simplicity, to illustrate the learner performance, the system's behavior has been repeatedly simulated from a set of initial states distributed within the constraints set. For the classifier design and for the neighbours points search in $\mathbb{R}^{10}$, we have used `Matlab Machine Learning Toolbox` and, in particular, Support Vector Machine solutions (Scholkopf and Smola (2001)) and function `rangesearch`. In this respect, we have first applied the density based clustering algorithm (Ester et al. (1996)) to discard outliers that may lead to a misleading classification.

In this example, to preselect the set of topologies in sets $\mathcal{T}_t$, we consider as criterion the infinity norm of $x_{\mathcal{N},t}$. Specifically, if $\|x_{\mathcal{N},t}\|_\infty \geq 3$, then we pre-select those topologies with 3 or 4 enabled links; if $1.5 \leq \|x_{\mathcal{N},t}\|_\infty < 3$, also those with 2 enabled links come into play; if $0.5 \leq \|x_{\mathcal{N},t}\|_\infty < 1.5$, we consider topologies with 2, 1 or 0 enabled links; and if $\|x_{\mathcal{N},t}\|_\infty \leq 0.5$, just topologies with 1 or 0 enabled links can be chosen.

Hereon, we will use the term *cycle* to denote a sequence of simulations from the set of initial points, and *game-round* will refer to each time the player makes a choice. In Fig. 2, we show the evolution of the learning regret together with the percentage of optimal decisions on the communication topology. It can be seen how previous observations help to learn which are the more suitable communication topologies to operate the system. At the beginning, the lack of information motivates arbitrary decisions and hence high regret and low level of optimal choices are observed. In particular, the percentage of optimal decisions remains below 30%, while it is notably increased in the future and ends around 80% after 20 cycles. As the system is always regulated to the origin, the player quickly gathers information in this area and hence learns faster the topology to be used. The latter is the main cause of having a relatively high percentage of optimal choices just after a few cycles. However, learning the optimal topology in the rest of the state space is more challenging. Note that the regret is varying for the same error percentage, as some topologies may be closer to be optimal than others. Fig. 3 illustrates the loss of optimality of the coalitional controller. To evaluate the performance, we use the average cumulative cost, i.e., the sum of the stage cost (5) for all time steps averaged over the number of initial states. In particular, in Fig. 3, we show the difference in cost between the coalitional controller and centralized MPC evaluated for each cycle. The figure illustrates the benefits of choosing suitable topologies on the system behavior. In particular, the cost for the centralized MPC controller is 560.6, while a complete decentralized structure leads to 582.2. Likewise, it is 560.9 for the coalitional controller with the optimal topology trajectory, however, the learning procedure entails higher losses especially at the beginning, when it reaches a maximum of 571.7. As example, in Fig. 4 we show the system state temporal evolution from a certain initial state and in Fig. 5 we show the sequences of topology that were chosen at two different cycles. Note that the state trajectory depends on the topologies selection and, hence, the optimal arms sequence may vary despite starting at the same state.

Table 2. Network topologies for the five trucks system.

Λ_0	$\emptyset$	Λ_4	$\{IV\}$	Λ_8	$\{II, III\}$	Λ_{12}	$\{I, II, IV\}$
Λ_1	$\{I\}$	Λ_5	$\{I, II\}$	Λ_9	$\{II, IV\}$	Λ_{13}	$\{I, III, IV\}$
Λ_2	$\{II\}$	Λ_6	$\{I, III\}$	Λ_{10}	$\{III, IV\}$	Λ_{14}	$\{II, III, IV\}$
Λ_3	$\{III\}$	Λ_7	$\{I, IV\}$	Λ_{11}	$\{I, II, III\}$	Λ_{15}	$\mathcal{L}$

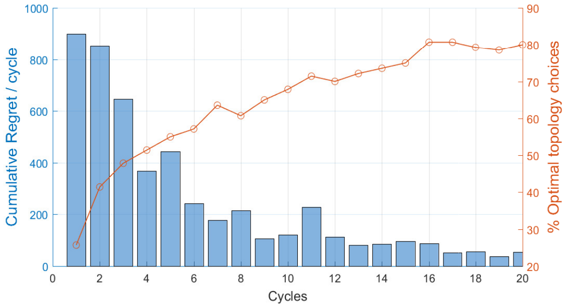


Fig. 2. Learning regret and percentage of optimal communication topology choices with respect to the whole number of game-rounds per cycle.

Finally, Fig. 6 illustrates the contexts classification. As the system evolves in the ten-dimensional space, we have selected a plane to allow for a bidimensional representation. To obtain this figure, we have trained the classifier with states points at the corresponding plane, that is, forcing the rest of the states to be zero.

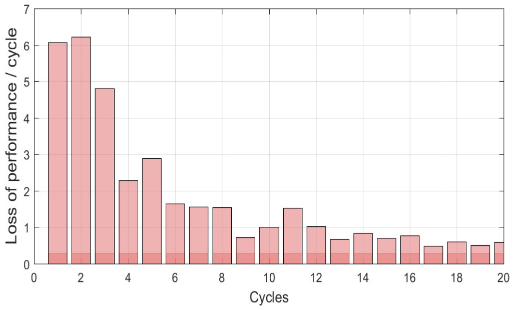


Fig. 3. Loss of performance of the coalitonal controller with respect to centralized MPC

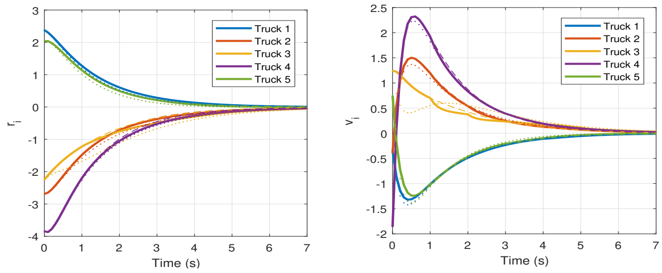
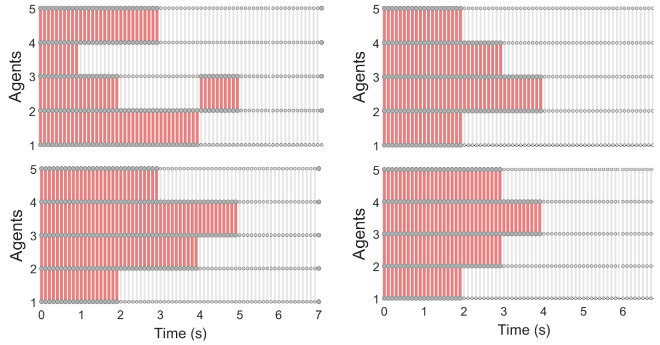


Fig. 4. Effect of the coordination scenario on the system state. The solid lines show the behaviour of the coalitional controller with the sequence of topologies shown in Fig. 5(a)(top). The centralized and decentralized behaviour are shown respectively in dashed and dotted lines.



(a) Cycle 3

(b) Cycle 15

Fig. 5. Selected (top) and optimal (bottom) communication topologies for two simulations from the same initial state at different cycles.

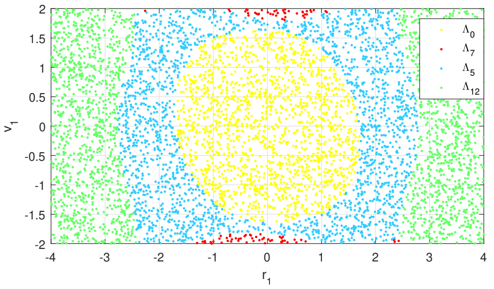


Fig. 6. Regions in the plane $[s_1, v_1]$ where different Λ dominates according to the classification of contexts.

6. CONCLUSIONS

In this paper, we deal with a multi-agent network where the communication structure among the set of control entities varies over time, aiming at maximizing the benefits provided by their mode of interaction. The switchings between topologies are based on a learning procedure where previous observations and current system state are used to estimate the performance of each coordination scenario. The problem is modeled as a sequential decision-game where the coalitions' composition can be periodically changed by a learning entity. We have proposed a heuristic algorithm for the topology adjustment where different hypothesis are introduced to improve the speed of convergence. In this respect, we have shown that the players' performance and, thus, the system optimality, increase over time. In particular, the system has been simulated periodically from a set of initial points and it is illustrated how the learner's regret notably decreases along with trials.

Further research will study stability issues on the proposed controller and explore other online-learning solutions to improve our results. Finally, we plan to investigate how these ideas can be applied from a bottom-up approach, that is, considering the links as decision-making entities that can enable/disable themselves.

REFERENCES

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3), 235–256.

Bubeck, S., Cesa-Bianchi, N., et al. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1), 1–122.

Christofides, P.D., Scattolini, R., de la Pena, D.M., and Liu, J. (2013). Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51, 21–41.

Chu, W., Li, L., Reyzin, L., and Schapire, R. (2011). Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 208–214.

Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, 226–231.

Farina, M. and Scattolini, R. (2012). Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems. *Automatica*, 48(6), 1088–1096.

Fele, F., Maestre, J.M., and Camacho, E.F. (2017). Coalitional control: Cooperative game theory and control. *IEEE Control Systems*, 37(1), 53–69.

Fele, F., Maestre, J.M., Hashemy, S.M., de la Peña, D.M., and Camacho, E.F. (2014). Coalitional model predictive control of an irrigation canal. *Journal of Process Control*, 24(4), 314–325.

Jain, A., Chakrabortty, A., and Biyik, E. (2018). Distributed wide-area control of power system oscillations under communication and actuation constraints. *Control Engineering Practice*, 74, 132–143.

Mayoraz, E. and Alpaydin, E. (1999). Support vector machines for multi-class classification. In *International Work-Conference on Artificial Neural Networks*, 833–842. Springer.

Negenborn, R.R. and Maestre, J. (2014). Distributed model predictive control: An overview and roadmap of future research opportunities. *IEEE Control Systems Magazine*, 34(4), 87–97.

Scholkopf, B. and Smola, A.J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Slivkins, A. (2014). Contextual bandits with similarity information. *The Journal of Machine Learning Research*, 15(1), 2533–2568.

Zheng, Y., Wei, Y., and Li, S. (2018). Coupling degree clustering-based distributed model predictive control network design. *IEEE Transactions on Automation Science and Engineering*.