An Iterative Design Method for Coalitional Control Networks with Constraints on the Shapley Value ⋆

F. J. Muros * **J. M. Maestre** * **E. Algaba** ** **T. Álamo** *
E. F. Camacho *

* Dep. of Systems and Automation Engineering, University of Seville,
Spain (e-mail: {franmuros, pepemaestre, talamo, efcamacho}@us.es)
** Dep. of Applied Mathematics II, University of Seville, Spain
(e-mail: ealgaba@us.es)

Abstract: In this work, we introduce a new iterative design method for a coalitional control scheme for linear systems recently proposed. In this scheme, the links in the network infrastructure are enabled or disabled depending on their contribution to the overall system performance. As a consequence, the local controllers are divided dynamically into sets or coalitions that cooperate in order to attain their control tasks. The new design method allows the control system designer to include new constraints regarding the game theoretical tools of the control architecture, while optimizing the matrices that define the controller.

Keywords: Coalitional networks, distributed control, cooperative game theory, Shapley value.

1. INTRODUCTION

During the last years, a lot of effort has been devoted to the development of non-centralized control techniques in order to take advantage of the new capabilities provided by information and communication technologies. This type of schemes presents very interesting features such as their inherent scalability and modularity, which make them appropriate for the control of large-scale systems such as traffic, water, or power networks (Negenborn et al., 2006).

The rationale behind non-centralized control is to partition a global control problem into several pieces that are assigned to local controllers –also known as agents– which may communicate to coordinate their control tasks. In case the controllers communicate, we will refer to distributed control. Otherwise, i.e., if there is no communication at all, the term decentralized control is used.

Literally, many schemes have been proposed during the last decade, very particularly under the framework of model based control (see for example Maestre and Negenborn (2014) for a survey of distributed model predictive control techniques). The key idea is to use a model of the system to predict and optimize a behavior according to the designer's goals. In this way, it is possible to deal explicitly with the interactions between the different controllers. Given that these interactions are not static, i.e., the groups of controllers whose control tasks are coupled vary with time, control schemes that also govern the communication between controllers have been developed lately. Examples of this type of control schemes are: Jilg and Stursberg (2013), where the coupling of the plant is used to partition it into hierarchically coupled clusters; Trodden

and Richards (2009), where groups of cooperative agents are formed using the set of active constraints; or Nuñez et al. (2013), where several possible hierarchical control structures are considered in order to implement the most appropriate one. Recently, this setting has been extended to a MPC framework in Fele et al. (2013).

In this work, we develop the properties of the coalitional scheme proposed by Maestre et al. (2011, 2013). In particular, in these works it is shown that the links of the network can be interpreted as players of a cooperative game whose characteristic function is given by a bound on the cost-to-go of the closed-loop system. Therefore, a solution concept of the game can provide information regarding the cost associated to the use of each link. In particular, we focus on the Shapley value of the game (Shapley, 1953) because it is the best known solution concept and it deals with the averaged contribution of each link, which is interesting in order to obtain information when considering all the different network topologies by the link game. In addition, if the cost function has an economical meaning, the Shapley value also provides a reasonable way of distributing the costs among the agents through the position value (Borm et al., 1992).

The main contribution of this work is an improvement of the controller design procedure proposed by Maestre et al. (2011, 2013) so that restrictions regarding the Shapley value can be considered. In this way, it is possible to design a coalitional control scheme that guarantees that the cost assigned to a given link is greater or lower than a certain threshold, or assures that a certain link will assume more (or less) cost than another one. In addition, we propose an iterative algorithm that allows us to improve the performance of the matrices calculated during the controller design phase.

The rest of the paper is organized as follows. First, the problem setting is provided. Second, an iterative controller design procedure based on linear matrix inequalities (LMIs), which includes conditions on the Shapley value, is presented. Next, a simple academical example is used to illustrate the proposed approach. Finally, conclusions and comments about future research are presented.

2. PROBLEM SETTING

In this section, we first present the model used to represent the system dynamics. Next, a description of the control scheme used is introduced. Finally, connections of this scheme with cooperative game theory are provided.

2.1 System Description

Consider the class of distributed linear systems composed of a set of $\mathcal{N} = \{1, 2, \ldots, N\}$ interconnected subsystems. The dynamics of the subsystem $i \in \mathcal{N}$ can be described mathematically as

$$\mathbf{x}_i(k+1) = \mathbf{A}_{ii}\mathbf{x}_i(k) + \mathbf{B}_{ii}\mathbf{u}_i(k) + \mathbf{d}_i(k),$$
$$\mathbf{d}_i(k) = \sum_{j \neq i} [\mathbf{A}_{ij}\mathbf{x}_j(k) + \mathbf{B}_{ij}\mathbf{u}_j(k)], \quad (1)$$

where $\mathbf{x}_i(k) \in \mathbb{R}^{n_{\mathbf{x}_i}}$ is the state of the subsystem i, $\mathbf{u}_i(k) \in \mathbb{R}^{n_{\mathbf{u}_i}}$ is its corresponding input, and $\mathbf{A}_{ij} \in \mathbb{R}^{n_{\mathbf{x}_i} \times n_{\mathbf{x}_j}}, \mathbf{B}_{ij} \in \mathbb{R}^{n_{\mathbf{x}_i} \times n_{\mathbf{u}_j}}$ are, respectively, the state transition and the input to state matrices. Notice that we use $\mathbf{d}_i(k)$ to denote the influence of other subsystems in the subsystem i.

The overall system can be described as

$$\mathbf{x}_{\mathcal{N}}(k+1) = \mathbf{A}_{\mathcal{N}}\mathbf{x}_{\mathcal{N}}(k) + \mathbf{B}_{\mathcal{N}}\mathbf{u}_{\mathcal{N}}(k), \quad (2)$$

where $\mathbf{x}_{\mathcal{N}}(k) = \begin{bmatrix} \mathbf{x}_1(k), \ldots, \mathbf{x}_{|\mathcal{N}|}(k) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{n_{\mathbf{x}_{\mathcal{N}}}}$, $\mathbf{u}_{\mathcal{N}}(k) = \begin{bmatrix} \mathbf{u}_1(k), \ldots, \mathbf{u}_{|\mathcal{N}|}(k) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{n_{\mathbf{u}_{\mathcal{N}}}}$ are, respectively, the global state and global input vectors, and $\mathbf{A}_{\mathcal{N}} = [\mathbf{A}_{ij}]_{i,j \in \mathcal{N}} \in \mathbb{R}^{n_{\mathbf{x}_{\mathcal{N}}} \times n_{\mathbf{x}_{\mathcal{N}}}}$, $\mathbf{B}_{\mathcal{N}} = [\mathbf{B}_{ij}]_{i,j \in \mathcal{N}} \in \mathbb{R}^{n_{\mathbf{x}_{\mathcal{N}}} \times n_{\mathbf{u}_{\mathcal{N}}}}$ are the state transition and the input to state global matrices.

2.2 Control Architecture

Each subsystem is governed by a local controller that has complete knowledge of its local model and state $\mathbf{x}_i(k)$, and it is able to manipulate the control action $\mathbf{u}_i(k)$. The agents can communicate by means of a network whose topology is described by a graph $(\mathcal{N}, \mathcal{E})$, with $\mathcal{E} \subseteq \mathcal{E}^{\mathcal{N}} = \mathcal{N} \times \mathcal{N}$ the set of edges corresponding to the physical paths between the agents in $\mathcal{N}$ (notice that ij and ji represent the same communication link). We assume that each link $l \in \mathcal{E}$ can be enabled or disabled at a time step, assuming a fixed cost $c \in \mathbb{R}^+$ in case it is enabled.

Definition 1. We call network mode or network topology denoted by $\Lambda(k)$ the set of enabled links in a time step k. Notice that there are $2^{|\mathcal{E}|}$ different network modes in a given network $(\mathcal{N}, \mathcal{E})$, which will be denoted as $\Lambda_0(k), \Lambda_1(k), \ldots, \Lambda_{2^{|\mathcal{E}|}-1}(k)$.

Example 1. Consider the 4-link networks shown in Figure 1, where we use arabic numbers for the agents and roman letters for the links. The network modes and the corresponding maximal connected agent coalitions or *components* for each network are represented in Table 1.
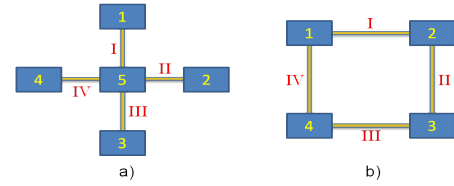

Fig. 1. Two different cases of 4-link networks $(\mathcal{N}, \mathcal{E})$

Table 1. Network modes and components

Network modes		Components	
		$(\mathcal{N}, \mathcal{E})_a$	$(\mathcal{N}, \mathcal{E})_b$
$\Lambda_0(k)$	$\{\emptyset\}$	$\{1\},\{2\},\{3\},\{4\},\{5\}$	$\{1\},\{2\},\{3\},\{4\}$
$\Lambda_1(k)$	$\{I\}$	$\{1,5\},\{2\},\{3\},\{4\}$	$\{1,2\},\{3\},\{4\}$
$\Lambda_2(k)$	$\{II\}$	$\{1\},\{2,5\},\{3\},\{4\}$	$\{1\},\{2,3\},\{4\}$
$\Lambda_3(k)$	$\{III\}$	$\{1\},\{2\},\{3,5\},\{4\}$	$\{1\},\{2\},\{3,4\}$
$\Lambda_4(k)$	$\{IV\}$	$\{1\},\{2\},\{3\},\{4,5\}$	$\{1,4\},\{2\},\{3\}$
$\Lambda_5(k)$	$\{I,II\}$	$\{1,2,5\},\{3\},\{4\}$	$\{1,2,3\},\{4\}$
$\Lambda_6(k)$	$\{I,III\}$	$\{1,3,5\},\{2\},\{4\}$	$\{1,2\},\{3,4\}$
$\Lambda_7(k)$	$\{I,IV\}$	$\{1,4,5\},\{2\},\{3\}$	$\{1,2,4\},\{3\}$
$\Lambda_8(k)$	$\{II,III\}$	$\{1\},\{2,3,5\},\{4\}$	$\{1\},\{2,3,4\}$
$\Lambda_9(k)$	$\{II,IV\}$	$\{1\},\{2,4,5\},\{3\}$	$\{1,4\},\{2,3\}$
$\Lambda_{10}(k)$	$\{III,IV\}$	$\{1\},\{2\},\{3,4,5\}$	$\{1,3,4\},\{2\}$
$\Lambda_{11}(k)$	$\{I,II,III\}$	$\{1,2,3,5\},\{4\}$	$\mathcal{N}$
$\Lambda_{12}(k)$	$\{I,II,IV\}$	$\{1,2,4,5\},\{3\}$	$\mathcal{N}$
$\Lambda_{13}(k)$	$\{I,III,IV\}$	$\{1,3,4,5\},\{2\}$	$\mathcal{N}$
$\Lambda_{14}(k)$	$\{II,III,IV\}$	$\{1\},\{2,3,4,5\}$	$\mathcal{N}$
$\Lambda_{15}(k)$	$\mathcal{E}$	$\mathcal{N}$	$\mathcal{N}$

We assume that, at each time step, the control goal is to minimize the following cost function:

$$J(k) = \overbrace{\left(\sum_{j=0}^{\infty} [\mathbf{x}_{\mathcal{N}}^{\mathrm{T}}(k+j)\mathbf{Q}_{\mathcal{N}}\mathbf{x}_{\mathcal{N}}(k+j) + \mathbf{u}_{\mathcal{N}}^{\mathrm{T}}(k+j)\mathbf{R}_{\mathcal{N}}\mathbf{u}_{\mathcal{N}}(k+j)] \right)}^{J_s(k)} + \overbrace{c|\Lambda(k)|}^{J_c(k)},$$
$$(3)$$

with $J_s(k)$, $J_c(k) \in \mathbb{R}^+$ being, respectively, the cost-to-go and the communication cost, and $\mathbf{Q}_{\mathcal{N}} \in \mathbb{R}^{n_{\mathbf{x}_{\mathcal{N}}} \times n_{\mathbf{x}_{\mathcal{N}}}}$, $\mathbf{R}_{\mathcal{N}} \in \mathbb{R}^{n_{\mathbf{u}_{\mathcal{N}}} \times n_{\mathbf{u}_{\mathcal{N}}}}$ positive definite and semi-definite weighting matrices[1]. To this end, the following topology-dependant overall control law is applied

$$\mathbf{u}_{\mathcal{N}} = \mathbf{K}_{\Lambda}\mathbf{x}_{\mathcal{N}}, \quad (4)$$

where $\mathbf{K}_{\Lambda} \in \mathbb{R}^{n_{\mathbf{u}_{\mathcal{N}}} \times n_{\mathbf{x}_{\mathcal{N}}}}$.

The problem of minimizing (3) can be formulated as a dynamic programming problem with mixed-integer optimization variables, which belongs to the class of NP-complete problems. In general, it is not possible to solve this problem easily because it is not convex. Hence, the goal is to calculate at least a suboptimal solution of the original problem. In this way, we introduce the following assumption.

Assumption 1. There exists a positive definite matrix $\mathbf{P}_{\Lambda} \in \mathbb{R}^{n_{\mathbf{x}_{\mathcal{N}}} \times n_{\mathbf{x}_{\mathcal{N}}}}$ that provides a Lyapunov function $f(\mathbf{x}_{\mathcal{N}}) = \mathbf{x}_{\mathcal{N}}^{\mathrm{T}}\mathbf{P}_{\Lambda}\mathbf{x}_{\mathcal{N}}$ of the closed-loop system (when the feedback law (4) is applied), and it also satisfies, $\forall \mathbf{x}_{\mathcal{N}}, \Lambda$

$$\mathbf{x}_{\mathcal{N}}^{\mathrm{T}}\mathbf{P}_{\Lambda}\mathbf{x}_{\mathcal{N}} \geq J_s. \quad (5)$$

Note that, following Maestre et al. (2013), $\mathbf{K}_{\Lambda}$ and $\mathbf{P}_{\Lambda}$ are related by

$$\overbrace{\mathbf{x}_{\mathcal{N}}^{\mathrm{T}+}\mathbf{P}_{\Lambda}\mathbf{x}_{\mathcal{N}}^{+}}^{\geq J_s^+} + \overbrace{\mathbf{x}_{\mathcal{N}}^{\mathrm{T}}\mathbf{Q}_{\mathcal{N}}\mathbf{x}_{\mathcal{N}} + \mathbf{x}_{\mathcal{N}}^{\mathrm{T}}\mathbf{K}_{\Lambda}^{\mathrm{T}}\mathbf{R}_{\mathcal{N}}\mathbf{K}_{\Lambda}\mathbf{x}_{\mathcal{N}}}^{\text{stage cost}} \leq \overbrace{\mathbf{x}_{\mathcal{N}}^{\mathrm{T}}\mathbf{P}_{\Lambda}\mathbf{x}_{\mathcal{N}}}^{\geq J_s}, \quad (6)$$

where superindex $+$ refers to dependence on $k+1$.

---

[1] In the following, we will omit the dependence of states and inputs on time step k for the sake of clarity.

Notice that a different $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda$ have to be found for each network topology Λ. Likewise, if there is no physical path between two local controllers i and j in the topology defined by Λ, which is denoted by $i \overset{\Lambda}{\nleftrightarrow} j$, then the sub-blocks of $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda$ that connect agents i and j, denoted respectively as $\mathbf{K}_\Lambda^{ij}, \mathbf{K}_\Lambda^{ji}$ and $\mathbf{P}_\Lambda^{ij}, \mathbf{P}_\Lambda^{ji}$, are zero.

Based on Assumption 1, and according to Maestre et al. (2013), we can define the following upper bound on the cost function J:

$$r^v(\Lambda, \mathbf{x}_{\mathcal{N}}) = \mathbf{x}_{\mathcal{N}}^{\mathrm{T}} \mathbf{P}_\Lambda \mathbf{x}_{\mathcal{N}} + c|\Lambda|, \qquad (7)$$

which can be minimized with respect to Λ in order to find out the most appropriate network topology at state $\mathbf{x}_{\mathcal{N}}$, according to the improvement of the system's performance. In this way, the following two-layer control scheme is proposed:

Let $D \in \mathbb{N}^+$ be a number of time samples. At each time step k,

(1) If k is a multiple of D, all the agents broadcast their state to calculate the network mode Λ that minimizes (7). Otherwise, each agent sends his state only to those agents that are connected to it.
(2) Each agent uses the state information received in order to update its control action. Globally, this implies that the linear controller $\mathbf{u}_{\mathcal{N}} = \mathbf{K}_\Lambda \mathbf{x}_{\mathcal{N}}$ is applied.

This control scheme is proved to be stable in Maestre et al. (2013). To be precise, this scheme works properly for small or medium scale networks. The combinatorial explosion problems make it inadequate for big scale networks.

2.3 Game Theoretical Perspective

In this subsection, we briefly introduce how game theoretical tools are used in Maestre et al. (2011, 2013) to find out what links and agents are more important in the proposed control scheme. In this context, it is possible to use the pair $(\mathcal{E}, \boldsymbol{r^v})$ to define a cooperative game with transferable utility. Notice that the key to connect the fields of control and cooperative game theory is to interpret (7) as the characteristic function of a cooperative game in which the set of edges $\mathcal{E}$ is the set of players.

Once that the game is built, it is possible to use a payoff rule to find a vector that specifies the benefit or cost that each player can reasonably expect from the game. Given that the characteristic function is based on the cost-to-go and the communication cost associated to each coalition of links, a payoff rule will provide the corresponding cost of each link. In general, useful links will be associated to lower costs in the payoff rule. In this work, we will use the Shapley value (Shapley, 1953), which assigns to the game $(\mathcal{E}, \boldsymbol{r^v})$ the vector $\boldsymbol{\phi}(\mathcal{E}, \boldsymbol{r^v})$, which is defined $\forall l \in \mathcal{E}$ as

$$\phi_l(\mathcal{E}, \boldsymbol{r^v}) = \sum_{\Lambda \subseteq \mathcal{E}, l \notin \Lambda} \frac{|\Lambda|!(|\mathcal{E}| - |\Lambda| - 1)!}{|\mathcal{E}|!} [r^v(\Lambda \cup \{l\}, \mathbf{x}_{\mathcal{N}}) - r^v(\Lambda, \mathbf{x}_{\mathcal{N}})], \qquad (8)$$

that is, the marginal contribution of each link l is averaged for all the possible permutations it can be part of.

Following the work of Xu et al. (2008), it is possible to find a matricial expression for the Shapley value. Consider a matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{E}| \times 2^{|\mathcal{E}|}}$, where the rows correspond to each link $l \in \mathcal{E}$ and the columns to the different network modes $\Lambda \subseteq \mathcal{E}$, in the lexicographic order. Given a link-game $(\mathcal{E}, \boldsymbol{r^v})$, the Shapley value $\boldsymbol{\phi}(\mathcal{E}, \boldsymbol{r^v})$ can be represented by the Shapley standard matrix $\mathbf{M}$ as

$$\boldsymbol{\phi}(\mathcal{E}, \boldsymbol{r^v}) = \begin{bmatrix} \phi_I \\ \phi_{II} \\ \vdots \\ \phi_{|\mathcal{E}|} \end{bmatrix} = \mathbf{M} \begin{bmatrix} r^v(\Lambda_0, \mathbf{x}_{\mathcal{N}}) \\ r^v(\Lambda_1, \mathbf{x}_{\mathcal{N}}) \\ r^v(\Lambda_2, \mathbf{x}_{\mathcal{N}}) \\ \vdots \\ r^v(\Lambda_{2^{|\mathcal{E}|}-1}, \mathbf{x}_{\mathcal{N}}) \end{bmatrix} = \mathbf{M}\boldsymbol{r^v}, \qquad (9)$$

with each component of $\boldsymbol{r^v}$ given by (7), and where each element of $\mathbf{M}$, denoted by $m_{l\Lambda}$, is defined as

$$m_{l\Lambda} = \begin{cases} \dfrac{(|\Lambda| - 1)!(|\mathcal{E}| - |\Lambda|)!}{|\mathcal{E}|!} & l \in \Lambda, \\[2mm] -\dfrac{|\Lambda|!(|\mathcal{E}| - |\Lambda| - 1)!}{|\mathcal{E}|!} & l \notin \Lambda. \end{cases} \qquad (10)$$

Example 2. Consider for example any link-game with 4 links, as represented in Figure 1. The Shapley standard matrix $\mathbf{M}$ can be easily obtained by using (10)

$$\mathbf{M}_4 = \begin{bmatrix} -\frac{1}{4} & \frac{1}{4} & -\frac{1}{12} & -\frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & -\frac{1}{12} & -\frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & -\frac{1}{4} & \frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{12} & \frac{1}{4} & -\frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & -\frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & \frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & \frac{1}{12} & -\frac{1}{4} & \frac{1}{12} & \frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{12} & -\frac{1}{12} & \frac{1}{4} & -\frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & \frac{1}{12} & -\frac{1}{4} & \frac{1}{12} & \frac{1}{12} & \frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{12} & -\frac{1}{12} & -\frac{1}{12} & \frac{1}{4} & -\frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & \frac{1}{12} & -\frac{1}{4} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{4} \end{bmatrix}.$$

Notice that by means of (7) and (10) it is possible to rewrite the Shapley value of each link $l \in \mathcal{E}$ as

$$\phi_l(\mathcal{E}, \boldsymbol{r^v}) = c + \sum_{\Lambda \subseteq \mathcal{E}} m_{l\Lambda} \left[\mathbf{x}_{\mathcal{N}}^{\mathrm{T}} \mathbf{P}_\Lambda \mathbf{x}_{\mathcal{N}}\right]. \qquad (11)$$

Note that an analysis by agents from the link-game can be obtained through the position value (Borm et al., 1992), which provides a payoff vector for the agents $i \in \mathcal{N}$ using the Shapley value of each link $l \in \mathcal{E}$, according to

$$\pi_i(\mathcal{N}, \boldsymbol{v}, \mathcal{E}) = \frac{1}{2} \sum_{l \in \mathcal{E}_i} \phi_l(\mathcal{E}, \boldsymbol{r^v}), \quad \forall i \in \mathcal{N}, \qquad (12)$$

where $\mathcal{E}_i$ represents the subset of links connected to agent i.

3. CONTROLLER DESIGN PROCEDURE

In this section, we present a method to design all the matrices for the control scheme using LMIs. We first introduce very briefly the original design method used in Maestre et al. (2011, 2013). Next, we develop new LMIs that can be added in order to guarantee that the Shapley value of the cooperative game satisfies certain conditions. Finally, we describe an iterative procedure in order to optimize the design method proposed.

3.1 Original Design Method

The problem of finding matrices $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda$ can be solved using linear matrix inequalities. In particular, in Maestre et al. (2013) it is shown that if the following conditions hold for each topology, then $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda$ stabilize the closed loop system, provide a bound on the cost-to-go and satisfy the communication constraints

$$\mathbf{P}_\Lambda > 0,$$

$$\mathbf{P}_\Lambda - (\mathbf{A}_{\mathcal{N}} + \mathbf{B}_{\mathcal{N}} \mathbf{K}_\Lambda)^{\mathrm{T}} \mathbf{P}_\Lambda (\mathbf{A}_{\mathcal{N}} + \mathbf{B}_{\mathcal{N}} \mathbf{K}_\Lambda) - \mathbf{Q}_{\mathcal{N}} - \mathbf{K}_\Lambda^{\mathrm{T}} \mathbf{R}_{\mathcal{N}} \mathbf{K}_\Lambda > 0, \qquad (13a)$$

$$i \stackrel{\Lambda}{\leftrightarrow} j \implies \begin{cases} \mathbf{K}_\Lambda^{ij} = \mathbf{K}_\Lambda^{ji} = 0, \\ \mathbf{P}_\Lambda^{ij} = \mathbf{P}_\Lambda^{ji} = 0. \end{cases} \quad (13b)$$

Note that the second inequality of (13a) does not verify the affinity property that is a necessary condition to consider the previous system as an LMI. Using the Schur complement it is possible to rewrite (13) as the following LMI[2]

$$\begin{bmatrix} \mathbf{W}_\Lambda & \mathbf{W}_\Lambda \mathbf{A}_\mathcal{N}^T + \mathbf{Y}_\Lambda^T \mathbf{B}_\mathcal{N}^T & \mathbf{W}_\Lambda \mathbf{Q}_\mathcal{N}^{1\backslash 2} & \mathbf{Y}_\Lambda^T \mathbf{R}_\mathcal{N}^{1\backslash 2} \\ \mathbf{A}_\mathcal{N} \mathbf{W}_\Lambda + \mathbf{B}_\mathcal{N} \mathbf{Y}_\Lambda & \mathbf{W}_\Lambda & 0 & 0 \\ \mathbf{Q}_\mathcal{N}^{1\backslash 2} \mathbf{W}_\Lambda & 0 & \mathbf{I} & 0 \\ \mathbf{R}_\mathcal{N}^{1\backslash 2} \mathbf{Y}_\Lambda & 0 & 0 & \mathbf{I} \end{bmatrix} > 0,$$
$$(14a)$$

$$i \stackrel{\Lambda}{\leftrightarrow} j \implies \begin{cases} \mathbf{Y}_\Lambda^{ij} = \mathbf{Y}_\Lambda^{ji} = 0, \\ \mathbf{W}_\Lambda^{ij} = \mathbf{W}_\Lambda^{ji} = 0, \end{cases} \quad (14b)$$

where $\mathbf{W}_\Lambda = \mathbf{P}_\Lambda^{-1}$ and $\mathbf{Y}_\Lambda = \mathbf{K}_\Lambda \mathbf{P}_\Lambda^{-1}$ are the decision variables.

Note that $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda$ can be rearranged as block diagonal matrices, which guarantees that (13b) and (14b) are equivalent. See Maestre et al. (2013) for further details.

3.2 Constraints on the Shapley Value

The Shapley value of the link-game provides the cost of each link when it joins arbitrarily to any coalition, i.e., this value distributes the costs of the grand coalition among the links that take part in the game. Hence, the higher value a link has, the more costly for the system it is. Keeping the Shapley value of certain links under/over some specific limits will force the overall system to consider them as more critical or unnecessary. To this end, different types of Shapley value constraints and the obtention of the corresponding LMI conditions, in order to integrate them into the design algorithm, are presented below.

3.2.1 Absolute Shapley Value Constraints

We can force the Shapley value of a certain link $l \in \mathcal{E}$ to keep under/over given constant thresholds $\mathcal{V}_l, \mathcal{W}_l \in \mathbb{R}$, i.e.,

$$\phi_l(\mathcal{E}, \boldsymbol{r^v}) < \mathcal{V}_l, \quad (15)$$

$$\phi_l(\mathcal{E}, \boldsymbol{r^v}) > \mathcal{W}_l. \quad (16)$$

Using (11) we can rewrite (15) as

$$\sum_{\Lambda \subseteq \mathcal{E}} m_{l\Lambda} \left[\mathbf{x}_\mathcal{N}^T \mathbf{P}_\Lambda \mathbf{x}_\mathcal{N} \right] < \mathcal{V}_l - c. \quad (17)$$

Next, we reformulate (17) as $\begin{bmatrix} 1 & \mathbf{x}_\mathcal{N}^T \end{bmatrix} \mathbf{D_a} \begin{bmatrix} 1 \\ \mathbf{x}_\mathcal{N} \end{bmatrix} > 0$. Thus, solving (15) is equivalent to finding a solution of

$$\mathbf{D_a} > 0, \quad \mathbf{D_a} = \begin{bmatrix} \mathcal{V}_l - c & 0 \\ 0 & -\sum_{\Lambda \subseteq \mathcal{E}} m_{l\Lambda} \mathbf{P}_\Lambda \end{bmatrix}. \quad (18)$$

By using constraint (16), and operating in the same way, we obtain the following LMI condition

---
[2] In the following, matrix $\mathbf{I}$ will denote the unit matrix of the corresponding size.

$$\mathbf{D_b} > 0, \quad \mathbf{D_b} = \begin{bmatrix} c - \mathcal{W}_l & 0 \\ 0 & \sum_{\Lambda \subseteq \mathcal{E}} m_{l\Lambda} \mathbf{P}_\Lambda \end{bmatrix}. \quad (19)$$

3.2.2 Relative Shapley Value Constraints

We can force the Shapley value of a certain link $l_p \in \mathcal{E}$ to be greater (lower) than the Shapley value of another link $l_q \in \mathcal{E}$, i.e.,

$$\phi_{l_p}(\mathcal{E}, \boldsymbol{r^v}) > \phi_{l_q}(\mathcal{E}, \boldsymbol{r^v}), \quad (20)$$

and by using (11), the above is equivalent to

$$\mathbf{D_c} > 0, \quad \mathbf{D_c} = \sum_{\Lambda \subseteq \mathcal{E}} \left(m_{l_p\Lambda} - m_{l_q\Lambda} \right) \mathbf{P}_\Lambda. \quad (21)$$

Therefore, we have obtained different LMI conditions depending on the Shapley value constraints considered.

Definition 2. We call Shapley condition set, denoted by $\mathcal{P}$, the set of different LMI conditions (18), (19) and (21), corresponding with the Shapley value constraints that may be imposed in a specific control problem.

3.3 Computation Procedure

The problem of the aforementioned Shapley LMI conditions is that they do not depend on the same variables as (14). In order to deal with this issue, we propose an iterative optimization procedure that is similar to that of *DK*-iteration (Skogestad and Postlethwaite, 2001), i.e., we alternate the optimization with respect to $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda$ (keeping the other fixed). To this end, we provide the following theorem, that we will use in the design algorithm.

Theorem 1. Let $\Lambda \subseteq \mathcal{E}$ and $\mathbf{O}_\Lambda \in \mathbb{R}^{n_{x_\mathcal{N}} \times n_{x_\mathcal{N}}}$ be a network mode and a positive definite constant matrix, respectively, such that $\mathbf{O}_\Lambda^{ij} = \mathbf{O}_\Lambda^{ji} = 0$ when $i \stackrel{\Lambda}{\leftrightarrow} j$ holds. Let the dynamics of the overall system be given by (2) and (4), and the cost function by (3). If there exist a matrix $\mathbf{K}_\Lambda \in \mathbb{R}^{n_{u_\mathcal{N}} \times n_{x_\mathcal{N}}}$ and a scalar $\xi_\Lambda \in \mathbb{R}^+$, such that the following constraints are satisfied

$$\begin{bmatrix} \mathbf{O}_\Lambda & \mathbf{Q}_\mathcal{N}^{1\backslash 2} & \mathbf{K}_\Lambda^T \mathbf{R}_\mathcal{N}^{1\backslash 2} & (\mathbf{A}_\mathcal{N} + \mathbf{B}_\mathcal{N} \mathbf{K}_\Lambda)^T \\ \mathbf{Q}_\mathcal{N}^{1\backslash 2} & \xi_\Lambda \mathbf{I} & 0 & 0 \\ \mathbf{R}_\mathcal{N}^{1\backslash 2} \mathbf{K}_\Lambda & 0 & \xi_\Lambda \mathbf{I} & 0 \\ \mathbf{A}_\mathcal{N} + \mathbf{B}_\mathcal{N} \mathbf{K}_\Lambda & 0 & 0 & \mathbf{O}_\Lambda^{-1} \end{bmatrix} > 0, \quad (22a)$$

$$i \stackrel{\Lambda}{\leftrightarrow} j \implies \mathbf{K}_\Lambda^{ij} = \mathbf{K}_\Lambda^{ji} = 0, \quad (22b)$$

then the matrices $\mathbf{P}_\Lambda = \xi_\Lambda \mathbf{O}_\Lambda$ and $\mathbf{K}_\Lambda$ verify (5), all the communication constraints imposed by the network mode Λ and stabilize the whole system.

Theorem 1 can be proved using standard techniques based on the application of Schur complement, as it is done in Maestre et al. (2011, 2013).

Next, we introduce the optimization algorithm. The goal of this procedure is to obtain the minimum bound on the cost-to-go, i.e., to minimize $\mathbf{P}_\Lambda$, while satisfying the Shapley constraints.

Design Algorithm

Let l be the iteration index and r be a counter variable, starting with $l = 1$ and $r = 0$, respectively

(1) In order to get an initial value of $\mathbf{K}_\Lambda$, $\mathbf{P}_\Lambda$, solve, $\forall \Lambda \subseteq \mathcal{E}$

$$\max_{\mathbf{W}_\Lambda, \mathbf{Y}_\Lambda} \text{Tr}(\mathbf{W}_\Lambda), \qquad (23)$$

subject to (14), from where we obtain the matrices $\mathbf{W}_\Lambda^{(r)}$ and $\mathbf{Y}_\Lambda^{(r)}$, and, consequently, $\mathbf{K}_\Lambda^{(r)}$ and $\mathbf{P}_\Lambda^{(r)}$.

(2) Let $\mathbf{K}_\Lambda^{(r+1)} = \mathbf{K}_\Lambda^{(r)}$, and solve

$$\min_{\mathbf{P}_\Lambda} \left(\sum_\Lambda \text{Tr}(\mathbf{P}_\Lambda) \right), \qquad (24)$$

subject to (13), $\forall \Lambda \subseteq \mathcal{E}$, and the chosen set $\mathcal{P}$. Therefore, we obtain $\mathbf{P}_\Lambda^{(r+1)}$.

(3) Let $\mathbf{P}_\Lambda^{(r+2)} = \xi_\Lambda \mathbf{P}_\Lambda^{(r+1)}$, and solve

$$\min_{\xi, \mathbf{K}_\Lambda} \left(\sum_\Lambda \xi_\Lambda \right), \qquad (25)$$

subject to (22), $\forall \Lambda \subseteq \mathcal{E}$, and $\mathcal{P}$. Hence, we get $\mathbf{K}_\Lambda^{(r+2)}$.

(4) Make $r = r + 2$, $l = l + 1$ and go to step 2, while $l < l_{\max}$ (with $l_{\max}$ the maximum number of iterations) or until convergence has been attained.

The key of the algorithm proposed is to consider, alternatively, constant solution of previous steps for $\mathbf{K}_\Lambda$ (step 2) and $\mathbf{P}_\Lambda$ (step 3). Hence, it is possible to include Shapley value constraints, and also to optimize the value of the matrices that define the controller. These are the main advantages of this procedure with respect to the one proposed in Maestre et al. (2013).

Note that both (24) and (25) improve the sum of the traces of the set $\mathbf{P}_\Lambda$. Given that this sum is lower bounded, it can be deduced that the algorithm converges.

3.4 Suboptimality Index

In order to determine the impact of satisfying the additional Shapley constraints, we define the following index, which will be calculated once $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda$ are obtained.

Definition 3. The suboptimality index of a set of matrices $\mathbf{P}_\Lambda, \Lambda \subseteq \mathcal{E}$ is defined as

$$\eta = \max_k \left| \frac{\mathbf{x}_\mathcal{N}^{\text{T}} \mathbf{P}_\Lambda \mathbf{x}_\mathcal{N}}{\mathbf{x}_\mathcal{N}^{\text{T}} \mathbf{P}_\Lambda^* \mathbf{x}_\mathcal{N}} \right|, \qquad (26)$$

where

$$\mathbf{P}_\Lambda^* = \lim_{n \to \infty} \left(\sum_{j=0}^n (\mathbf{A}_{CL}^{\text{T}})^j \left[\mathbf{Q}_\mathcal{N} + \mathbf{K}_\Lambda^{\text{T}} \mathbf{R}_\mathcal{N} \mathbf{K}_\Lambda \right] \mathbf{A}_{CL}^j \right), \qquad (27)$$

is obtained by using (2) and (4), with $\mathbf{A}_{CL} = \mathbf{A}_\mathcal{N} + \mathbf{B}_\mathcal{N} \mathbf{K}_\Lambda$.

Notice that we take, each D time samples, the matrix $\mathbf{P}_\Lambda$ corresponding to the mode Λ that minimizes (7). Likewise, the expression (27) can be considered the optimal value for the corresponding $\mathbf{P}_\Lambda$. Furthermore, a value of η closer to value 1 implies less degradation of the set of matrices $\mathbf{P}_\Lambda, \Lambda \subseteq \mathcal{E}$ from the theoretical optimal value. As it will be seen in the simulation section, the index η decreases with the number of algorithm iterations applied.

4. SIMULATION RESULTS

In this section, we show an academic example with four agents and four links, i.e., $\mathcal{N} = \{1, 2, 3, 4\}, \mathcal{E} = \{I, II, III, IV\}$, corresponding to the configuration shown in Figure 1b. The sixteen different network modes and their respective components are specified in Table 1. In the following, we denote as Λ_C the network modes equivalent to the grand coalition of agents $\mathcal{N}$. The matrices that define the subsystem dynamics are

$$\mathbf{A}_{11} = \begin{bmatrix} 1 & 0.8 \\ 0 & 0.7 \end{bmatrix}, \mathbf{A}_{22} = \begin{bmatrix} 1 & 0.9 \\ 0 & -2.5 \end{bmatrix}, \mathbf{A}_{33} = \begin{bmatrix} 1 & -2 \\ 0 & 3 \end{bmatrix}, \mathbf{A}_{44} = \begin{bmatrix} 1 & 2.2 \\ 0 & 0.5 \end{bmatrix},$$

$$\mathbf{B}_{ii} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{A}_{ij} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{B}_{ij} = \begin{bmatrix} 0 \\ 0.15 \end{bmatrix}, i \neq j, \qquad (28)$$

where $\mathbf{x}_i \in \mathbb{R}^2$ and $\mathbf{u}_i \in \mathbb{R}$ are, respectively, the states and the input of each subsystem $i \in \mathcal{N}$. The cost-to-go J_s of all the subsystems is defined by matrices $\mathbf{Q} = \mathbf{I} \in \mathbb{R}^{8 \times 8}$ and $\mathbf{R} = \mathbf{I} \in \mathbb{R}^{4 \times 4}$. We also suppose $c = 0.5$ and $l_{\max} = 15$.

In order to demonstrate the design procedure, we will impose the following Shapley condition set $\mathcal{P}$ to the overall problem

$$\phi_{III}(\mathcal{E}, \boldsymbol{r^v}) > 0, \quad \phi_{II}(\mathcal{E}, \boldsymbol{r^v}) < 1. \qquad (29a)$$

$$\begin{aligned} \phi_I(\mathcal{E}, \boldsymbol{r^v}) &> \phi_{IV}(\mathcal{E}, \boldsymbol{r^v}), \\ \phi_{II}(\mathcal{E}, \boldsymbol{r^v}) &> \phi_{IV}(\mathcal{E}, \boldsymbol{r^v}), \\ \phi_{III}(\mathcal{E}, \boldsymbol{r^v}) &> \phi_{IV}(\mathcal{E}, \boldsymbol{r^v}), \end{aligned} \qquad (29b)$$

The design algorithm of Subsection 3.3 has been implemented using Matlab® LMI Control Toolbox (Gahinet et al., 1995). In Figure 2a, it is possible to see the decrease of η with the number of iterations l. Hence, the design algorithm improves the control matrices, as expected. As a result of the considered algorithm, the matrices $\mathbf{K}_\Lambda \in \mathbb{R}^{4 \times 8}$ and $\mathbf{P}_\Lambda \in \mathbb{R}^{8 \times 8}$, $\forall \Lambda \subseteq \mathcal{E}$, can be obtained.

Once the design problem is solved, we can test the control scheme proposed in Subsection 2.2 with $D = 3$. Let the initial state

$$\mathbf{x}_1 = \begin{bmatrix} 5 \\ 2 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}, \mathbf{x}_3 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \mathbf{x}_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \qquad (30)$$

Next, we present some simulations of the given controller. In Figure 2b the cumulated cost of the proposed coalitional algorithm is compared with the cumulated costs of considering full communication (centralized system) and no communication (decentralized system). As expected, the coalitional scheme outperforms the decentralized one, and is not far away from the centralized controller during the initial steps. Later, the communication cost makes the coalitional scheme to be the most appropriate one. Figure 3 shows the input and state trajectories as a function of time.
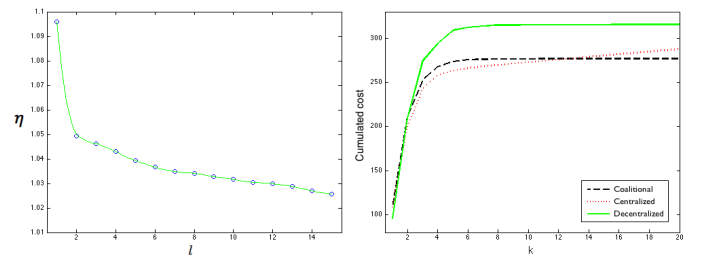

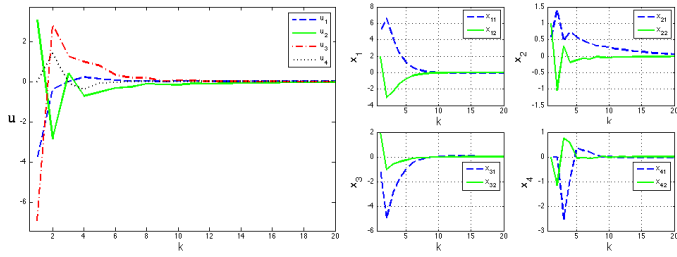Fig. 2. Suboptimality index evolution and cumulated cost comparative study

Fig. 3. Input and state trajectories

At this point we will analyze two scenarios resulting of omitting (Figure 4a) and considering (Figure 4b) the Shapley condition set (29). In the first scenario the initial network mode Λ_C evolves to an intermediate mode $\Lambda_2 = \{II\}$ and then it reaches the decentralized mode Λ_0. Notice that the link IV is the most expensive one –the highest value– in this initial case. In the second scenario, the initial mode is $\Lambda_5 = \{I, II\}$ and the intermediate one is $\Lambda_9 = \{II, IV\}$. Hence, link III is deactivated, giving priority to link II, according to (29a), and it holds an increase in the use of link IV, now the cheapest one, according to (29b). Therefore, the network mode evolves in line with the required conditions on the Shapley value. Notice that the steady state does not depend on the constraints imposed, that is, the Shapley value tends to $c = 0.5$, as expected according to (11), and the network mode tends to the decentralized mode Λ_0 in both scenarios.
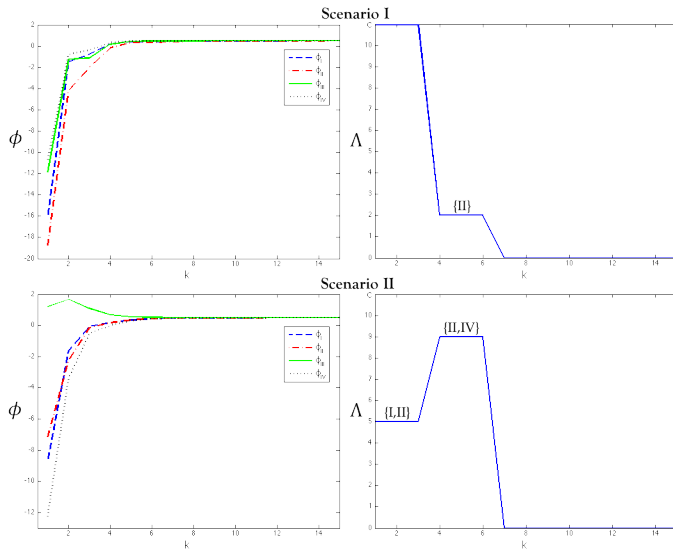

Fig. 4. Shapley values and network modes

As it has been shown in this section, we can conclude that the numerical results illustrate the feasibility of the considered approach.

5. CONCLUSIONS

In this paper, we have enhanced the design method proposed in Maestre et al. (2011, 2013) for a coalitional control scheme. In particular, we have focused on how to include constraints regarding the Shapley value at the design phase. Moreover, these conditions allow the controller to analyze if a certain coalitional control scheme verifies the constraints. On top of that, we have proposed an iterative design method in order to improve the perfomance of the

matrices that define the controller. Finally, the example illustrates the feasibility of the proposed scheme.

Future work will include the development of a more dynamical approach for the cooperative games that results from the application of this control architecture. In addition, the explicit inclusion of constraints for the agents by using the position value will be considered. Likewise, other solution concepts from game theory could be take into account for the design procedure.

REFERENCES

Borm, P., Owen, G., and Tijs, S. (1992). On the position value for communication situations. *SIAM Journal on Discrete Mathematics*, 5(3), 305–320.

Fele, F., Maestre, J.M., Muros, F.J., and Camacho, E.F. (2013). Coalitional control: An irrigation canal case study. In *Proceedings of the* 10*th IEEE ICNSC*, 759–764.

Gahinet, P., Nemirovskii, A.S., Laub, A.J., and Chilali, M. (1995). *LMI Control Toolbox For Use with MATLAB®*. The MathWorks, Inc.

Jilg, M. and Stursberg, O. (2013). Optimized distributed control and topology design for hierarchically interconnected systems. In *Proceedings of the* 12*th European Control Conference*, 4340–4346. Zurich, Switzerland.

Maestre, J.M., Muñoz de la Peña, D., Jiménez Losada, A., Algaba, E., and Camacho, E.F. (2013). A coalitional control scheme with applications to cooperative game theory. *Optimal Control Applications and Methods*. DOI: 10.1002/oca.2090.

Maestre, J.M., Muñoz de la Peña, D., Jiménez Losada, A., Algaba, E., and Camacho, E.F. (2011). An application of cooperative game theory to distributed control. In *Proceedings of the* 18*th IFAC World Congress*, 9121–9126. Milano, Italy.

Maestre, J.M. and Negenborn, R.R. (2014). *Distributed MPC Made Easy*. Intelligent Systems, Control and Automation: Science and Engineering. Springer-Verlag, Vol. 69, Dordrecht, The Netherlands.

Negenborn, R.R., De Schutter, B., and Hellendoorn, J. (2006). Multi–agent model predictive control of transportation networks. In *Proceedings of the* 3*th IEEE ICNSC*, 296–301. Ft. Lauderdale, Florida, USA.

Nuñez, A., Ocampo-Martínez, C., De Schutter, B., Valencia, F., López, J., and Espinosa, J. (2013). A multiobjective-based switching topology for hierarchical model predictive control applied to a hydro-power valley. In *Proceedings of the* 3*th IFAC ICONS*. Chengdu, China.

Shapley, L.S. (1953). A value for n-person games. *Contributions to the Theory of Games II. Edited by H. W. Kuhn y A. W. Tucker. Annals of Mathematics Studies*, 28, 307–317.

Skogestad, S. and Postlethwaite, I. (2001). *Multivariable Feedback Control - Analysis and Design*. John Wiley and Sons, New York, USA. 2*nd* edition.

Trodden, P. and Richards, A.G. (2009). Adaptive cooperation in robust distributed model predictive control. In *Proceedings of the* 24*th IEEE ISIC*, 896–901. St. Petersburg, Russia.

Xu, G., Driessen, T.S.H., and Sun, H. (2008). Matrix analysis for associated consistency in cooperative game theory. *Linear Algebra and its Applications*, 428(7), 1571–1586.