Automatica 78 (2017) 320–332

Contents lists available at ScienceDirect

Automatica

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

Networked control design for coalitional schemes using game-theoretic methods☆

Francisco Javier Muros [a], José María Maestre [a], Encarnación Algaba [b], Teodoro Alamo [a], Eduardo F. Camacho [a]

[a] Department of Systems and Automation Engineering, University of Seville, Spain
[b] Department of Applied Mathematics II, University of Seville, Spain

ARTICLE INFO

Article history:
Received 4 September 2014
Received in revised form
25 August 2016
Accepted 25 November 2016
Available online 3 February 2017

Keywords:
Coalitional schemes
Networked control
Distributed control
Cooperative game theory
Shapley value
Position value

ABSTRACT

In this work, we present an iterative design method for a coalitional networked control scheme for linear systems. In this scheme, the links in the communication network are enabled or disabled depending on their contribution to the overall system performance. Likewise, the control law is adapted to these changes. In particular, new conditions are included at the design phase, in order to consider constraints on the links and the agents regarding the game theoretical tools utilized while optimizing the matrices that define the controller.

1. Introduction

Non-centralized control techniques have been well addressed by the control community: their well-known advantages such as scalability and modularity, are suitable to control large-scale systems such as traffic, water or power networks (Negenborn, De Schutter, & Hellendoorn, 2006). The main idea of these schemes (in comparison with centralized ones) is to divide the overall system into several pieces, each of them governed by a local controller or *agent*. In this way, we refer to decentralized control, if there is no communication among the agents, i.e., the subsystems are isolated; or distributed control, in case the controllers share information to improve the overall system performance.

Focusing on distributed control, it is possible to find in the literature many examples – very specifically under the framework of model based control – which consider that the groups of connected agents, also called *coalitions*, do not vary along the time. In other words, there is no possibility of modifying the way in that the agents are grouped. Examples of these approaches are: Maestre, Muñoz de la Peña, Camacho, and Alamo (2011a), where the agents always send proposals regarding the control actions to the same neighbors; or Lagrangian prices based schemes as Negenborn, van Overloop, Keviczky, and De Schutter (2009), because prices are always updated by the set of agents that share the common resource. See Maestre and Negenborn (2014) and Scattolini (2009) for surveys of these techniques in a distributed model predictive control context.

Recently, different works that consider explicitly interactions among the agents that evolve dynamically with time to reduce the communication burden without compromising the system performance have appeared. To this end, groups of cooperating controllers are merged into dynamic neighborhoods or coalitions that behave as a single agent. Examples of this type of schemes, known as *coalitional control* schemes, can be found in: Jilg and Stursberg (2013), where the coupling of the plant is used to

☆ Financial support by the MINECO-Spain Projects DPI2013-46912-C2-1-R (COOPERA), DPI2013-48243-C2-2-R and ECO2015-68856-P, and the European Union Project FP7-ICT-ICT-2013.3.4-611281 (DYMASOS) is gratefully acknowledged. The material in this paper was partially presented at the 13th European Control Conference, June 24–27, 2014, Strasbourg, France and at the 19th World Congress of the International Federation of Automatic Control (IFAC 2014), August 24–29, 2014, Cape Town, South Africa. This paper was recommended for publication in revised form by Associate Editor Valery Ugrinovskii under the direction of Editor Ian R. Petersen.

E-mail addresses: franmuros@us.es (F.J. Muros), pepemaestre@us.es (J.M. Maestre), ealgaba@us.es (E. Algaba), talamo@us.es (T. Alamo), efcamacho@us.es (E.F. Camacho).

http://dx.doi.org/10.1016/j.automatica.2016.12.010
0005-1098/

partition it into hierarchically coupled clusters; Trodden and Richards (2009), where coalitions are formed using the set of active constraints; Núñez, Ocampo-Martinez, De Schutter, Valencia, López, and Espinosa (2013), Núñez, Ocampo-Martinez, Maestre, and De Schutter (2015), where several possible hierarchical control structures are considered to implement the most appropriate one; or Maestre, Muñoz de la Peña, Jiménez Losada, Algaba, and Camacho (2011b); Maestre, Muñoz de la Peña, Jiménez Losada, Algaba, and Camacho (2014), where the control scheme enables or disables links depending on their contribution to the overall system performance. Recently, this setting has been extended to a MPC framework in Fele, Maestre, Muros, and Camacho (2013), Fele, Maestre, Shahdany, Muñoz de la Peña, and Camacho (2014) and Maestre, Muros, Fele, and Camacho (2015).

The applications of cooperative game theory into engineering problems are becoming more common. For example, Saad, Han, Debbah, Hjørungnes, and Başar (2009) present a tutorial regarding coalitional game theory and its applications in communication networks. Some interesting applications of this framework into control problems have been presented by Bauso and his coworkers, who proposed in Bauso and Timmer (2012) a robust dynamic scheme in which instantaneous and averaged games are analyzed and allocation rules are presented. The problem of robust allocation rules for cooperative games is also considered in Bauso and Timmer (2009). Finally, this line of work is enhanced in Nedić and Bauso (2013), where a distributed bargaining protocol is developed so that an allocation inside the core of the game is provided in different cases.

In particular, in Maestre et al. (2011b, 2014) some game theoretical tools are introduced to consider a bound on the cost function minimized by the control scheme, as the characteristic function of a cooperative game where the players are the links that connect the agents. Once the game is defined in that way, it is necessary to choose a payoff rule to distribute the benefit or cost of the grand coalition among the players (links). From the different solution concepts, there are some recent works Muros, Maestre, Algaba, Alamo, and Camacho (2014a,b) and Muros, Maestre, Algaba, Ocampo-Martinez, and Camacho (2015) that have focused on the Shapley value of the game (Shapley, 1953). This value deals with the averaged contribution of each link, which is interesting to obtain information when considering all the different network configurations by the link-game. In addition, if the cost function has an economical meaning, the position value (Borm, Owen, & Tijs, 1992) also provides a reasonable way of distributing the costs among the agents.

In this work, we enhance and present in a more formal way the preliminary results given in Muros et al. (2014a,b). More specifically, we will focus on the following directions:

- We derive conditions to consider Shapley and position value constraints in the overall control problem. In particular, we introduce a matrix notation that extends the Shapley standard matrix concept (Xu, Driessen, & Sun, 2008) to the position value. Ultimately, this setting will make possible to bound or establish comparisons for each link or agent inside the network and also combine constraints for the links and the agents.
- We propose an iterative design algorithm which optimizes the matrices that define the controller. In addition, we present a new suboptimality index, which gives a measure of the convergence achieved.

In order to achieve the objectives mentioned above, we will use linear matrix algebra and linear matrix inequalities (LMIs) to model the optimization problem. The key idea is to minimize a linear objective under LMI constraints. In this way, if there exists a set of matrices that simultaneously satisfy all the LMIs, this set is convex and hence the *interior point methods* (IPMs) find a solution of the optimization problem with an affordable computational time (Alamo, Normey-Rico, Arahal, Limon, & Camacho, 2006; Boyd, El Ghaoui, Feron, & Balakrishnan, 1994). In this work, we will use a Matlab® solver which implements the IPMs proposed in Nesterov and Nemirovskii (1994). Nevertheless, in the context of design, there are other solvers in the literature, such as the *active set methods*. In this sense, the choice of the optimization method will depend on the specific problem to solve in each case. A comparative analysis of the different solvers available can be found in Bartlett, Wätchet, and Biegler (2000), Geletu (2007), Leyffer and Mahajan (2010) and Wright (1997).

Another relevant topic in this context is that of networked control systems (NCSs), which consider spatially distributed systems for which the communication between their parts is supported by a shared communication network. These control systems deal with the issues derived from the communication, e.g., packet data rates, networking technology, sampling, network security, packet dropout or network delays (see Gupta & Chow, 2010; Hespanha, Naghshtabrizi, & Xu, 2007 for surveys about these topics). Nevertheless, in this particular work, we will not deal with these challenges, but we will focus on analyzing control architecture changes that depend on the state, by considering game theory tools. Another noteworthy point in this regard is that some schemes proposed in the NCSs literature are also designed by means of LMIs (Millán,Orihuela,Vivas,Rubio,Dimarogonas & Johansson, 2013), which may simplify its integration with the design method presented in this article.

Note that this work contains significant differences with respect to Muros et al. (2014a,b). First, the design algorithm has been generalized to consider constraints on the agents by the position value, and the corresponding position value LMI conditions have been derived. In addition, an extension for multiplayer constraints is also considered. Likewise, the steady state is studied, the limit case conditions for the LMIs are also given and the decentralized case has been analyzed more explicitly. Moreover, performance indices have been redefined and proofs for the theorems have been provided. Finally, new examples and figures are presented to enhance and illustrate the explanation of the scheme proposed.

The rest of the paper is organized as follows. In Section 2 the problem setting and the game theory tools are presented. In Section 3, a controller design procedure based on LMIs, which integrates conditions on the Shapley and the position values, is introduced. In Section 4, a very simple numerical example is used to illustrate the proposed approach. Finally, conclusions and comments about future research are presented in Section 5.

2. Problem formulation

In this section, we present the model used to represent the system dynamics, a description of the control scheme and how some cooperative game theory tools can be applied to distributed control.

2.1. System description

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

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

[1] In Section 3.4 the possibility of including extra state or input constraints is considered.

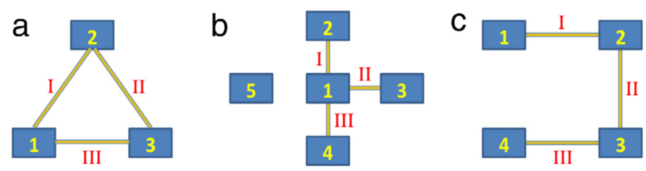

Fig. 1. Three different cases of 3-link networks $(\mathcal{N}, \mathcal{E})$.

where $\mathbf{x}_i(k) \in \mathbb{R}^{n_{\mathrm{x}_i}}$ is the state of subsystem i, $\mathbf{u}_i(k) \in \mathbb{R}^{n_{\mathrm{u}_i}}$ is its corresponding input, and $\mathbf{A}_{ij} \in \mathbb{R}^{n_{\mathrm{x}_i} \times n_{\mathrm{x}_j}}$, $\mathbf{B}_{ij} \in \mathbb{R}^{n_{\mathrm{x}_i} \times n_{\mathrm{u}_j}}$ 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 on subsystem i.

The overall system dynamics can be described as

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

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

2.2. Networked control architecture

The agents in $\mathcal{N}$ are connected by a network whose topology is characterized by an undirected graph $(\mathcal{N}, \mathcal{E})$, where $\mathcal{E} \subseteq \mathcal{E}^{\mathcal{N}} = \mathcal{N} \times \mathcal{N}$ is the set of edges or links $l \in \mathcal{E}$ corresponding to the physical connections among the subsystems. We assume that each link l can be enabled or disabled at each time step, with a corresponding cost per link $c \in \mathbb{R}^+ \setminus \{0\}$ in case it is enabled. Notice that this cost could be different for each link as well. Therefore, the network described by graph $(\mathcal{N}, \mathcal{E})$ has a dynamic configuration depending on its enabled (or disabled) links at each time step.

Definition 1. Consider a network described by an undirected graph $(\mathcal{N}, \mathcal{E})$. The set of enabled links in a time step k, denoted by $\Lambda(k)$, is named network topology and it verifies $\Lambda(k) \subseteq \mathcal{E}$. The $2^{|\mathcal{E}|}$ possible network topologies in $(\mathcal{N}, \mathcal{E})$ will be symbolized as $\Lambda_0, \Lambda_1, \ldots, \Lambda_{2^{|\mathcal{E}|}-1}$.

Definition 2. Consider a network described by an undirected graph $(\mathcal{N}, \mathcal{E})$. The set that contains all the possible network topologies that can be taken by network $(\mathcal{N}, \mathcal{E})$ in any time step k, denoted by $\mathbf{\Lambda} = \{\Lambda_0, \Lambda_1, \ldots, \Lambda_{2^{|\mathcal{E}|}-1}\} \in \mathbb{R}^{2^{|\mathcal{E}|}}$, is named topologies set.

Remark 1. The set of possible topologies described by $\mathbf{\Lambda}$ is static and only depends on network $(\mathcal{N}, \mathcal{E})$ considered. The network topology $\Lambda(k)$, which describes the dynamic graph configuration at each time step k, will take one specific value from the different elements $\Lambda_0, \Lambda_1, \ldots, \Lambda_{2^{|\mathcal{E}|}-1}$ that belong to $\mathbf{\Lambda}$. In other words, $\Lambda(k) \in \mathbf{\Lambda}$.

Example 1. Consider the 3-link networks shown in Fig. 1, where we use arabic numbers for the agents and roman letters for the links. The network topologies, the enabled/disabled links, and the corresponding maximal connected agent coalitions or communication components for each network are represented in Table 1.

In this work, we will consider that the control purpose is to minimize the following cost function:

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

with $J_{\mathrm{s}}(\mathbf{x}_{\mathcal{N}}(k), \mathbf{u}_{\mathcal{N}}(k), \Lambda(k)), J_{\mathrm{c}}(\Lambda(k)) \in \mathbb{R}^+$ being, respectively, the cost-to-go and the communication cost, where $\mathbf{Q}_{\mathcal{N}} \in \mathbb{R}^{n_{\mathrm{x}_{\mathcal{N}}} \times n_{\mathrm{x}_{\mathcal{N}}}}$, $\mathbf{R}_{\mathcal{N}} \in \mathbb{R}^{n_{\mathrm{u}_{\mathcal{N}}} \times n_{\mathrm{u}_{\mathcal{N}}}}$ are positive semi-definite and definite weighting matrices, respectively, and with $c \in \mathbb{R}^+ \setminus \{0\}$ being the cost per enabled link that we introduced previously. We assume that the following linear topology-dependent feedback is used to control the system

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

with $\mathbf{K}_{\Lambda(k)} \in \mathbb{R}^{n_{\mathrm{u}_{\mathcal{N}}} \times n_{\mathrm{x}_{\mathcal{N}}}}$.

As can be seen, Eq. (3) sums a cost related to the system performance from a control viewpoint, i.e., $J_{\mathrm{s}}(\mathbf{x}_{\mathcal{N}}(k), \mathbf{u}_{\mathcal{N}}(k), \Lambda(k))$, and another related with the number of communication links used by the control system, i.e., $J_{\mathrm{c}}(\Lambda(k))$. Notice that $J_{\mathrm{s}}(\mathbf{x}_{\mathcal{N}}(k), \mathbf{u}_{\mathcal{N}}(k), \Lambda(k))$ is the classical cost minimized by an LQR and it is also affected by the topology of the control system. Hence, the topology used has an indirect effect on the evolution of the stage cost of the system, i.e., the control law changes with the network topology. Term $J_{\mathrm{c}}(\Lambda(k))$ is then introduced to explicitly penalize the use of the communication network. This way the control system designer can attain a trade-off between control performance and communication burden. Otherwise, full communication at each time step would be used because it provides optimal control performance.

In general, it is not possible to solve the problem of minimizing (3) in a straightforward way because it belongs to the class of NP-complete problems (Fletcher & Leyffer, 1998). The choice regarding the state of each link can be modeled as a binary decision-variable. With the exception of particular structures, mixed-integer programming problems involving 0–1 variables are classified as NP-complete (Bemporad & Morari, 1999; Raman & Grossmann, 1991). Note that the nature of NP-complete problems seriously compromises their numerical solution. Therefore, from now on, we provide a heuristic solution of the original problem. To this end, we make the following assumption.

Assumption 1. Let (4) be the feedback control law used for system (2). There exists a Lyapunov function $f(\mathbf{x}_{\mathcal{N}}(k)) = \mathbf{x}_{\mathcal{N}}^{\mathrm{T}}(k)\mathbf{P}_{\Lambda(k)}\mathbf{x}_{\mathcal{N}}(k)$ of the closed-loop system, which satisfies, $\forall \mathbf{x}_{\mathcal{N}}(k), \Lambda(k)$, the following point-wise inequality

$$\mathbf{x}_{\mathcal{N}}^{\mathrm{T}}(k)\mathbf{P}_{\Lambda(k)}\mathbf{x}_{\mathcal{N}}(k) \geq J_{\mathrm{s}}(\mathbf{x}_{\mathcal{N}}(k), \mathbf{u}_{\mathcal{N}}(k), \Lambda(k)), \tag{5}$$

with $J_{\mathrm{s}}(\mathbf{x}_{\mathcal{N}}(k), \mathbf{u}_{\mathcal{N}}(k), \Lambda(k))$ being the cost-to-go of the overall cost $J(\mathbf{x}_{\mathcal{N}}(k), \mathbf{u}_{\mathcal{N}}(k), \Lambda(k))$, which is defined in (3), and where $\mathbf{P}_{\Lambda(k)} \in \mathbb{R}^{n_{\mathrm{x}_{\mathcal{N}}} \times n_{\mathrm{x}_{\mathcal{N}}}}$ is a positive definite matrix.

Following Maestre et al. (2014), $\mathbf{K}_{\Lambda(k)}$ and $\mathbf{P}_{\Lambda(k)}$ are related by

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

where $\mathbf{x}_{\mathcal{N}}^{+}(k)$ is the successor state.

Table 1
Network topologies and communication components.

Set $\mathbf{\Lambda}$		Links			Communication components		
		I	*II*	*III*	$(\mathcal{N},\mathcal{E})_a$	$(\mathcal{N},\mathcal{E})_b$	$(\mathcal{N},\mathcal{E})_c$
Λ_0	$\emptyset$	×	×	×	{1}, {2}, {3}	{1}, {2}, {3}, {4}, {5}	{1}, {2}, {3}, {4}
Λ_1	{*I*}	✓	×	×	{1, 2}, {3}	{1, 2}, {3}, {4}, {5}	{1, 2}, {3}, {4}
Λ_2	{*II*}	×	✓	×	{1}, {2, 3}	{1, 3}, {2}, {4}, {5}	{1}, {2, 3}, {4}
Λ_3	{*III*}	×	×	✓	{1, 3}, {2}	{1, 4}, {2}, {3}, {5}	{1}, {2}, {3, 4}
Λ_4	{*I*, *II*}	✓	✓	×	$\mathcal{N}$	{1, 2, 3}, {4}, {5}	{1, 2, 3}, {4}
Λ_5	{*I*, *III*}	✓	×	✓	$\mathcal{N}$	{1, 2, 4}, {3}, {5}	{1, 2}, {3, 4}
Λ_6	{*II*, *III*}	×	✓	✓	$\mathcal{N}$	{1, 3, 4}, {2}, {5}	{1}, {2, 3, 4}
Λ_7	$\mathcal{E}$	✓	✓	✓	$\mathcal{N}$	{1, 2, 3, 4}, {5}	$\mathcal{N}$

Notice that there is a different $\mathbf{K}_{\Lambda(k)}$ and $\mathbf{P}_{\Lambda(k)}$ for each network topology $\Lambda(k)$. Likewise, if there is no physical path between two agents i and j in a particular topology defined by $\Lambda(k)$, which will be denoted as $i \overset{\Lambda(k)}{\nleftrightarrow} j$, the sub-blocks of $\mathbf{K}_{\Lambda(k)}$ and $\mathbf{P}_{\Lambda(k)}$ that connect both agents, denoted as $\mathbf{K}^{ij}_{\Lambda(k)}$, $\mathbf{K}^{ji}_{\Lambda(k)}$ and $\mathbf{P}^{ij}_{\Lambda(k)}$, $\mathbf{P}^{ji}_{\Lambda(k)}$, respectively, are zero (see for instance agent 5 in the network of Fig. 1(b)).

Assumption 2. It is possible to find a feasible solution for each $\mathbf{K}_{\Lambda(k)}$ and $\mathbf{P}_{\Lambda(k)}$ that verify (6), with $\Lambda(k) \in \mathbf{\Lambda}$.

Remark 2. In this work, we consider that all the topologies are feasible, i.e., $|\mathbf{\Lambda}| = 2^{|\mathcal{E}|}$. Nevertheless, it may be possible to limit set $\mathbf{\Lambda}$, in order to satisfy Assumption 2, to include only those topologies for which $\mathbf{K}_{\Lambda(k)}$ and $\mathbf{P}_{\Lambda(k)}$ exist.

Remark 3. The Lyapunov function $f(\mathbf{x}_{\mathcal{N}}(k))$ establishes a finite upper bound on the cost-to-go $J_s(\mathbf{x}_{\mathcal{N}}(k), \mathbf{u}_{\mathcal{N}}(k), \Lambda(k))$ by satisfying (5).

Based on Assumptions 1 and 2, and according to Maestre et al. (2014), we can define the following upper bound on the cost function $J(\mathbf{x}_{\mathcal{N}}(k), \mathbf{u}_{\mathcal{N}}(k), \Lambda(k))$

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

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

Control Scheme 1

Let $k_s \in \mathbb{N}^+ \setminus \{0\}$ be a number of time samples. At each time step k,

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

As it can be seen, this scheme is appropriate for small or medium scale networks. The combinatorial explosion problems make it inadequate for large scale networks.[2]

Remark 4. In this work, we assume an implementation of the control scheme in a hierarchical fashion, with a centralized top layer with only one decision-maker that calculates the optimal network topology by using information from all the agents, and a coalitional bottom layer, where there are multiple decision-makers that, depending on the network topology commanded by the upper layer, enable or disable the corresponding links in a dynamical way, and also apply the corresponding optimal control law.

It may also be possible to consider a centralized implementation. Hence, there would not be independent agents, i.e., only one decision-maker. Finally, the control scheme may also be implemented in a distributed fashion so that the agents calculate the best network topology by exchanging information using standard distributed optimization methods (Maestre & Negenborn, 2014).

2.3. Game theoretical perspective

In Maestre et al. (2011b, 2014) the key to incorporate game theory results to distributed control is the interpretation of pair $(\mathcal{E}, \boldsymbol{r}^{\boldsymbol{v}})$ as a cooperative game with transferable utility, where the set of edges $\mathcal{E}$ is the set of players and with $\boldsymbol{r}^{\boldsymbol{v}}$ defined by (7). Once the game is defined, we need to choose a payoff rule to give us the corresponding cost or benefit that each player expects from the game. In general, useful players will be associated to lower costs in the payoff rule. In this work, we will use the Shapley value (Shapley, 1953), which assigns to game $(\mathcal{E}, \boldsymbol{r}^{\boldsymbol{v}})$ the vector $\boldsymbol{\phi}(\mathcal{E}, \boldsymbol{r}^{\boldsymbol{v}})$, which is defined $\forall l \in \mathcal{E}$ as

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

that is, the marginal contribution of each link l is averaged for all the possible network permutations it can be part of. Without any doubt, the Shapley value is the most studied solution concept in cooperative games because of its reasonable properties. Moreover, the position value defined by the Shapley value of the link-game is completely characterized (Borm et al., 1992). In fact, the original axiomatization of the Shapley value (Shapley, 1953) is by the properties of *linearity*, *efficiency*, *dummy player* and *symmetry*.

Based on Xu et al. (2008), it is possible to find a matrix 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 topologies $\Lambda \subseteq \mathcal{E}$, in the lexicographic order. Given a link-game $(\mathcal{E}, \boldsymbol{r}^{\boldsymbol{v}})$, the Shapley value $\boldsymbol{\phi}(\mathcal{E}, \boldsymbol{r}^{\boldsymbol{v}})$ can be represented by the Shapley standard matrix $\mathbf{M}$ as

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

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

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

[2] From now on, states, inputs and topologies dependence on time step k will be omitted, for simplicity.

Example 2. Consider any link-game with 3 links, as represented in Fig. 1. The Shapley standard matrix can be easily obtained by using (10)

$$\mathbf{M}_3 = \begin{bmatrix} -1/3 & 1/3 & -1/6 & -1/6 & 1/6 & 1/6 & -1/3 & 1/3 \\ -1/3 & -1/6 & 1/3 & -1/6 & 1/6 & -1/3 & 1/6 & 1/3 \\ -1/3 & -1/6 & -1/6 & 1/3 & -1/3 & 1/6 & 1/6 & 1/3 \end{bmatrix}.$$

By combining (7) and (10) with (9) and taking into account that

$$\sum_{\Lambda\subseteq\mathcal{E}} m_{l\Lambda} c|\Lambda| = c, \quad \forall l \in \mathcal{E}, \tag{11}$$

it is possible to obtain the Shapley value of each link $l \in \mathcal{E}$ as Muros et al. (2014a,b)

$$\phi_l(\mathcal{E}, \mathbf{r}^{\mathbf{v}}) = c + \sum_{\Lambda\subseteq\mathcal{E}} m_{l\Lambda} \left[\mathbf{x}_{\mathcal{N}}^{\mathrm{T}} \mathbf{P}_\Lambda \mathbf{x}_{\mathcal{N}}\right]. \tag{12}$$

Remark 5. The Shapley value of a link $l \in \mathcal{E}$ given by (12) analyzes the behavior of this link inside the network from a control perspective. In this sense, this value provides us information about link l (by means of its communication cost c) but also takes into account its interdependence on other links inside the network (by means of elements $m_{l\Lambda}$ of the Shapley standard matrix) and the system dynamics (by means of matrices $\mathbf{P}_\Lambda$).

An analysis by agents from the link-game is obtained in Borm et al. (1992) through the position value, which gives a payoff for each agent $i \in \mathcal{N}$ using the Shapley value of the link-game, according to

$$\pi_i(\mathcal{N}, \boldsymbol{v}, \mathcal{E}) = \frac{1}{2}\sum_{l\in\mathcal{E}_i} \phi_l(\mathcal{E}, \mathbf{r}^{\mathbf{v}}), \quad \forall i \in \mathcal{N}, \tag{13}$$

where $\mathcal{E}_i$ represents the subset of links connected to agent i. This allocation rule is the only one that satisfies the properties of *efficiency by components* and *balanced total threats* (Slikker, 2005). In fact, we can calculate a matrix expression that connects the position and Shapley values.

Definition 3. Consider a network described by an undirected graph $(\mathcal{N}, \mathcal{E})$. Let $\boldsymbol{\Pi} \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{E}|}$ be a matrix, where the rows refer to each agent $i \in \mathcal{N}$ and the columns to each link $l \in \mathcal{E}$. The Π_{il} element of $\boldsymbol{\Pi}$ is defined as

$$\Pi_{il} = \begin{cases} 1/2, & l \in \mathcal{E}_i, \\ 0, & l \notin \mathcal{E}_i. \end{cases} \tag{14}$$

In other words, matrix $\boldsymbol{\Pi}$ is related to the incidence matrix (Diestel, 2005) of a graph $(\mathcal{N}, \mathcal{E})$. In this sense, matrix $\boldsymbol{\Pi}$, with its Π_{il} elements defined by (14), satisfies

$$\boldsymbol{\pi}(\mathcal{N}, \boldsymbol{v}, \mathcal{E}) = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_{|\mathcal{N}|} \end{bmatrix} = \boldsymbol{\Pi}\boldsymbol{\phi}(\mathcal{E}, \mathbf{r}^{\mathbf{v}}) = \boldsymbol{\Pi}\mathbf{M}\mathbf{r}^{\mathbf{v}}. \tag{15}$$

Example 3. Consider the different networks represented in Fig. 1. Matrix $\boldsymbol{\Pi}$ for each case, is given by

$$\boldsymbol{\Pi}_{\mathrm{a}} = \frac{1}{2}\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \qquad \boldsymbol{\Pi}_{\mathrm{b}} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{\Pi}_{\mathrm{c}} = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Finally, it is possible to reach an expression for the position value of each agent $i \in \mathcal{N}$ if we rewrite (15) by using (12)–(14), and rearranging

$$\begin{aligned} \pi_i(\mathcal{N}, \boldsymbol{v}, \mathcal{E}) &= \sum_{l\in\mathcal{E}} \Pi_{il}\phi_l(\mathcal{E}, \mathbf{r}^{\mathbf{v}}) \\ &= c\sum_{l\in\mathcal{E}} \Pi_{il} + \sum_{l\in\mathcal{E}}\sum_{\Lambda\subseteq\mathcal{E}} \Pi_{il} m_{l\Lambda} \left[\mathbf{x}_{\mathcal{N}}^{\mathrm{T}} \mathbf{P}_\Lambda \mathbf{x}_{\mathcal{N}}\right]. \end{aligned} \tag{16}$$

Remark 6. If there are no links connected to an agent $j \in \mathcal{N}$ in a given network $(\mathcal{N}, \mathcal{E})$, i.e., agent j is isolated, then $\Pi_{jl} = 0, \forall l \in \mathcal{E}$, and consequently $\pi_j(\mathcal{N}, \boldsymbol{v}, \mathcal{E}) = 0$. See for example agent 5 for the network shown in Fig. 1(b).

Remark 7. Matrix $\mathbf{M}$ only depends on the number of players in the link-game. Therefore, we will have a unique matrix $\mathbf{M}_{|\mathcal{E}|}$ for all the possible combinations of link-games with $|\mathcal{E}|$ links. However, matrix $\boldsymbol{\Pi}$ depends on the topology of network $(\mathcal{N}, \mathcal{E})$ considered. Hence, we have a univocal matrix $\boldsymbol{\Pi}_{(\mathcal{N},\mathcal{E})}$ that defines every network $(\mathcal{N}, \mathcal{E})$.

Remark 8. If we consider the decentralized network topology, i.e., $\Lambda_0 = \emptyset$ (isolated subsystems), the corresponding cost $r^v(\Lambda_0, \mathbf{x}_{\mathcal{N}})$ could not be other than zero. Thus, in order to use (7) as a cost function of a game, it has to be redefined as

$$r^{v\prime}(\Lambda, \mathbf{x}_{\mathcal{N}}) = r^v(\Lambda, \mathbf{x}_{\mathcal{N}}) - r^v(\Lambda_0, \mathbf{x}_{\mathcal{N}}), \quad \forall \Lambda \subseteq \mathcal{E}. \tag{17}$$

Nevertheless, the Shapley value for the redefined game (and consequently, the position value) remains constant, according to (8), because both cost functions $r^{v\prime}(\Lambda, \mathbf{x}_{\mathcal{N}})$ and $r^v(\Lambda, \mathbf{x}_{\mathcal{N}})$ only differ in a term that does not depend on Λ.

Remark 9. In the steady state, the Shapley value of a link $l \in \mathcal{E}$ and the position value of an agent $i \in \mathcal{N}$ are, respectively

$$\phi_l^{ss}(\mathcal{E}, \mathbf{r}^{\mathbf{v}}) = c, \tag{18a}$$

$$\pi_i^{ss}(\mathcal{N}, \boldsymbol{v}, \mathcal{E}) = c\sum_{l\in\mathcal{E}} \Pi_{il}. \tag{18b}$$

In other words, the Shapley value in the steady state does not depend on the link considered. However, the position value in the steady state will be affected by the number of links connected to the agent under study.

3. Controller design procedure

In this section, we present an offline method to design the matrices for the controller using LMIs. We first introduce very briefly the original design method used in Maestre et al. (2011b, 2014). Next, we develop new LMIs that can be added to guarantee that the Shapley and the position values of the cooperative game satisfy certain conditions. Finally, we describe an iterative procedure to optimize the design method proposed.

3.1. Original design method

The problem of finding matrices $\mathbf{K}_\Lambda$, which stabilize the overall system, and $\mathbf{P}_\Lambda$, which provide us with a bound on the cost to go, can be solved via

$$\begin{aligned} \mathbf{P}_\Lambda &> 0, \\ \mathbf{P}_\Lambda - (\mathbf{A}_{\mathcal{N}} + \mathbf{B}_{\mathcal{N}}\mathbf{K}_\Lambda)^{\mathrm{T}} \mathbf{P}_\Lambda (\mathbf{A}_{\mathcal{N}} + \mathbf{B}_{\mathcal{N}}\mathbf{K}_\Lambda) & \\ - \mathbf{Q}_{\mathcal{N}} - \mathbf{K}_\Lambda^{\mathrm{T}}\mathbf{R}_{\mathcal{N}}\mathbf{K}_\Lambda &> 0, \end{aligned} \tag{19a}$$

$$i \overset{\Lambda}{\nleftrightarrow} j \Longrightarrow \begin{cases} \mathbf{K}_\Lambda^{ij} = \mathbf{K}_\Lambda^{ji} = 0, \\ \mathbf{P}_\Lambda^{ij} = \mathbf{P}_\Lambda^{ji} = 0. \end{cases} \tag{19b}$$

Using the Schur complement (Zhang, 2005) it is possible to rewrite (19) as the following LMI[3] (see Maestre et al., 2014)

$$\begin{bmatrix} \mathbf{W}_\Lambda & \mathbf{W}_\Lambda \mathbf{A}_\mathcal{N}^{\mathrm{T}} + \mathbf{Y}_\Lambda^{\mathrm{T}} \mathbf{B}_\mathcal{N}^{\mathrm{T}} & \mathbf{W}_\Lambda \mathbf{Q}_\mathcal{N}^{1/2} & \mathbf{Y}_\Lambda^{\mathrm{T}} \mathbf{R}_\mathcal{N}^{1/2} \\ \mathbf{A}_\mathcal{N} \mathbf{W}_\Lambda + \mathbf{B}_\mathcal{N} \mathbf{Y}_\Lambda & \mathbf{W}_\Lambda & 0 & 0 \\ \mathbf{Q}_\mathcal{N}^{1/2} \mathbf{W}_\Lambda & 0 & \mathbf{I} & 0 \\ \mathbf{R}_\mathcal{N}^{1/2} \mathbf{Y}_\Lambda & 0 & 0 & \mathbf{I} \end{bmatrix} > 0, \tag{20a}$$

$$i \overset{\Lambda}{\nleftrightarrow} j \Longrightarrow \begin{cases} \mathbf{Y}_\Lambda^{ij} = \mathbf{Y}_\Lambda^{ji} = 0, \\ \mathbf{W}_\Lambda^{ij} = \mathbf{W}_\Lambda^{ji} = 0 \end{cases} \tag{20b}$$

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

Remark 10. Control matrices $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda$ can be rearranged as block diagonal matrices, which guarantees that (19b) and (20b) are equivalent. See Maestre et al. (2014) for further details.

Remark 11. In order to optimize the design of the controller, the LMI conditions (20) are satisfied aiming to minimize the trace of $\mathbf{P}_\Lambda, \forall \Lambda \subseteq \mathcal{E}$, by means of the maximization of the trace of $\mathbf{W}_\Lambda$, which is the inverse of $\mathbf{P}_\Lambda$.

3.2. Constraints on the Shapley and Position values

The Shapley value satisfies efficiency which means that the cost of the grand coalition in the link-game is allocated among the links participating in the game. Hence, the higher value a link has, the more costly for the system it is. Moreover, if the Shapley value of certain links is bounded under/over certain limits, then the overall system will be forced to consider these links as more critical/dispensable. Likewise, if the cost function is economical, the constraints allow the designer to include limits in the payoff of the players. In fact, we can interpret the position value of an agent i as a weighted measure of the Shapley value on all the links that are connected to this agent. In this way, different types of Shapley and position value constraints – from now on, shortly called, value constraints – and the obtention of the corresponding LMI conditions are next presented, in order to integrate them into the design algorithm.

3.2.1. Absolute constraints

We impose that the Shapley value of a certain link $l \in \mathcal{E}$ is kept under/over given constant thresholds $\mathcal{V}_l, \mathcal{W}_l \in \mathbb{R}$, i.e.,

$$\phi_l(\mathcal{E}, \boldsymbol{r^v}) < \mathcal{V}_l, \tag{21}$$

$$\phi_l(\mathcal{E}, \boldsymbol{r^v}) > \mathcal{W}_l. \tag{22}$$

Following Muros et al. (2014b), it can be concluded that, solving (21) and (22) is equivalent to finding, respectively, a solution of

$$\mathbf{D_a} > 0, \quad \text{with } \mathbf{D_a} = \begin{bmatrix} \mathcal{V}_l - c & 0 \\ 0 & -\sum_{\Lambda \subseteq \mathcal{E}} m_{l\Lambda} \mathbf{P}_\Lambda \end{bmatrix}, \tag{23}$$

$$\mathbf{D_b} > 0, \quad \text{with } \mathbf{D_b} = \begin{bmatrix} c - \mathcal{W}_l & 0 \\ 0 & \sum_{\Lambda \subseteq \mathcal{E}} m_{l\Lambda} \mathbf{P}_\Lambda \end{bmatrix}. \tag{24}$$

Analogously, we can force the position value of a certain agent $i \in \mathcal{N}$ to be kept under/over given constant thresholds $\mathcal{Y}_i, \mathcal{Z}_i \in \mathbb{R}$, i.e.,

$$\pi_i(\mathcal{N}, \boldsymbol{v}, \mathcal{E}) < \mathcal{Y}_i, \tag{25}$$

$$\pi_i(\mathcal{N}, \boldsymbol{v}, \mathcal{E}) > \mathcal{Z}_i. \tag{26}$$

[3] From now on, matrix $\mathbf{I}$ will denote the identity matrix of the corresponding size.

By combining (16) with (25) and (26) it is possible to obtain the following LMI conditions

$$\mathbf{E_a} > 0, \quad \text{with } \mathbf{E_a} = \begin{bmatrix} \mathcal{Y}_i - c \sum_{l \in \mathcal{E}} \Pi_{il} & 0 \\ 0 & -\sum_{l \in \mathcal{E}} \sum_{\Lambda \subseteq \mathcal{E}} \Pi_{il} m_{l\Lambda} \mathbf{P}_\Lambda \end{bmatrix}, \tag{27}$$

$$\mathbf{E_b} > 0, \quad \text{with } \mathbf{E_b} = \begin{bmatrix} c \sum_{l \in \mathcal{E}} \Pi_{il} - \mathcal{Z}_i & 0 \\ 0 & \sum_{l \in \mathcal{E}} \sum_{\Lambda \subseteq \mathcal{E}} \Pi_{il} m_{l\Lambda} \mathbf{P}_\Lambda \end{bmatrix}. \tag{28}$$

Remark 12. In order to fulfill the LMI requirements, the first principal minors of (23), (24), (27) and (28) have to be equal to or greater than zero, and this depends on the constant thresholds $\mathcal{V}_l$, $\mathcal{W}_l$, $\mathcal{Y}_i$, $\mathcal{Z}_i$. Hence, according to (18), it is necessary to satisfy the following additional steady state constraints

$$\begin{aligned} \mathcal{V}_l &\geq \phi_l^{ss}(\mathcal{E}, \boldsymbol{r^v}), & \mathcal{W}_l &\leq \phi_l^{ss}(\mathcal{E}, \boldsymbol{r^v}), \\ \mathcal{Y}_i &\geq \pi_i^{ss}(\mathcal{N}, \boldsymbol{v}, \mathcal{E}), & \mathcal{Z}_i &\leq \pi_i^{ss}(\mathcal{N}, \boldsymbol{v}, \mathcal{E}). \end{aligned} \tag{29}$$

In the limit case, the principal minors are equal to zero and the resulting LMI conditions are

$$\begin{aligned} \mathbf{D_a^0} &> 0, \quad \text{with } \mathbf{D_a^0} = -\sum_{\Lambda \subseteq \mathcal{E}} m_{l\Lambda} \mathbf{P}_\Lambda, \\ \mathbf{D_b^0} &> 0, \quad \text{with } \mathbf{D_b^0} = \sum_{\Lambda \subseteq \mathcal{E}} m_{l\Lambda} \mathbf{P}_\Lambda, \\ \mathbf{E_a^0} &> 0, \quad \text{with } \mathbf{E_a^0} = -\sum_{l \in \mathcal{E}} \sum_{\Lambda \subseteq \mathcal{E}} \Pi_{il} m_{l\Lambda} \mathbf{P}_\Lambda, \\ \mathbf{E_b^0} &> 0, \quad \text{with } \mathbf{E_b^0} = \sum_{l \in \mathcal{E}} \sum_{\Lambda \subseteq \mathcal{E}} \Pi_{il} m_{l\Lambda} \mathbf{P}_\Lambda. \end{aligned} \tag{30}$$

3.2.2. Relative constraints

We may require that the Shapley value of a certain link $l_\mathrm{p} \in \mathcal{E}$ is greater (lower) than the Shapley value of another link $l_\mathrm{q} \in \mathcal{E}$, i.e.,

$$\phi_{l_\mathrm{p}}(\mathcal{E}, \boldsymbol{r^v}) > \phi_{l_\mathrm{q}}(\mathcal{E}, \boldsymbol{r^v}). \tag{31}$$

By means of (12), we can obtain the following LMI condition (Muros et al., 2014a,b)

$$\mathbf{D_c} > 0, \quad \text{with } \mathbf{D_c} = \sum_{\Lambda \subseteq \mathcal{E}} \left(m_{l_\mathrm{p}\Lambda} - m_{l_\mathrm{q}\Lambda} \right) \mathbf{P}_\Lambda. \tag{32}$$

We can also force the position value of a certain agent $i_\mathrm{p} \in \mathcal{N}$ to be greater (lower) than the position value of another agent $i_\mathrm{q} \in \mathcal{N}$, i.e.,

$$\pi_{i_\mathrm{p}}(\mathcal{N}, \boldsymbol{v}, \mathcal{E}) > \pi_{i_\mathrm{q}}(\mathcal{N}, \boldsymbol{v}, \mathcal{E}). \tag{33}$$

By using (16), the following relation is satisfied

$$\mathbf{E_c} > 0, \quad \text{with } \mathbf{E_c} = \begin{bmatrix} c \sum_{l \in \mathcal{E}} \left(\Pi_{i_\mathrm{p}l} - \Pi_{i_\mathrm{q}l} \right) & 0 \\ 0 & \sum_{l \in \mathcal{E}} \sum_{\Lambda \subseteq \mathcal{E}} \left(\Pi_{i_\mathrm{p}l} - \Pi_{i_\mathrm{q}l} \right) m_{l\Lambda} \mathbf{P}_\Lambda \end{bmatrix}. \tag{34}$$

Remark 13. In order to fulfill the LMI requirements, the first principal minor of (34) has to be equal to or greater than zero. However, in this case, it does not depend on given thresholds but on the position value steady state defined by (18b). In other words, it is necessary to satisfy

$$\pi_{i_\mathrm{p}}^{ss}(\mathcal{N}, \boldsymbol{v}, \mathcal{E}) \geq \pi_{i_\mathrm{q}}^{ss}(\mathcal{N}, \boldsymbol{v}, \mathcal{E}). \tag{35}$$

Moreover, for the limit case, i.e., $\pi_{i_p}^{ss}(\mathcal{N}, \boldsymbol{v}, \mathcal{E}) = \pi_{i_q}^{ss}(\mathcal{N}, \boldsymbol{v}, \mathcal{E})$, the resulting LMI condition is

$$\mathbf{E_c^0} > 0, \quad \text{with } \mathbf{E_c^0} = \sum_{l\in\mathcal{E}}\sum_{\Lambda\subseteq\mathcal{E}} \left(\Pi_{i_pl} - \Pi_{i_ql}\right) m_{l\Lambda}\mathbf{P}_\Lambda, \tag{36}$$

which is similar to (32). This fact is because the steady state Shapley value $\phi_l^{ss}(\mathcal{E}, \mathbf{r}^{\boldsymbol{v}})$, according to (18a), does not depend on the link considered.

So far, we have obtained LMI conditions depending on the different value constraints.

3.3. Computation procedure

In order to group the LMI conditions associated to the value constraints and for the sake of clarity we introduce the following definition.

Definition 4. We call value constraint set, denoted by $\mathcal{G}$, to the set of different LMI conditions (23), (24), (27), (28), (32) and (34), corresponding to the Shapley and position value constraints that may be imposed in a specific control problem.

The problem of the aforementioned LMI conditions is that they do not depend on the same variables as (20). In order to deal with this issue, we propose an iterative optimization procedure that is similar to that of *DK*-iterations (Skogestad & 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, which was introduced without proof in Muros et al. (2014b).

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 topology and a positive definite constant matrix, respectively, such that $\mathbf{O}_\Lambda^{ij} = \mathbf{O}_\Lambda^{ji} = 0$ when $i \overset{\Lambda}{\nleftrightarrow} j$ 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 $\xi_\Lambda \in \mathbb{R}^+ \setminus \{0\}$, such that the following constraints are satisfied*

$$\begin{bmatrix} \mathbf{O}_\Lambda & \mathbf{Q}_\mathcal{N}^{1/2} & \mathbf{K}_\Lambda^T\mathbf{R}_\mathcal{N}^{1/2} & (\mathbf{A}_\mathcal{N}+\mathbf{B}_\mathcal{N}\mathbf{K}_\Lambda)^T \\ \mathbf{Q}_\mathcal{N}^{1/2} & \xi_\Lambda\mathbf{I} & 0 & 0 \\ \mathbf{R}_\mathcal{N}^{1/2}\mathbf{K}_\Lambda & 0 & \xi_\Lambda\mathbf{I} & 0 \\ \mathbf{A}_\mathcal{N}+\mathbf{B}_\mathcal{N}\mathbf{K}_\Lambda & 0 & 0 & \mathbf{O}_\Lambda^{-1} \end{bmatrix} > 0, \tag{37a}$$

$$i \overset{\Lambda}{\nleftrightarrow} j \Longrightarrow \mathbf{K}_\Lambda^{ij} = \mathbf{K}_\Lambda^{ji} = 0, \tag{37b}$$

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

Proof. Applying iteratively backward the Schur's complement (Zhang, 2005) to LMI (37a) it can be seen that if (37a) is satisfied, then the following inequality holds

$$\mathbf{O}_\Lambda - (\mathbf{A}_\mathcal{N}+\mathbf{B}_\mathcal{N}\mathbf{K}_\Lambda)^\mathrm{T}\mathbf{O}_\Lambda(\mathbf{A}_\mathcal{N}+\mathbf{B}_\mathcal{N}\mathbf{K}_\Lambda) - \frac{\mathbf{Q}_\mathcal{N}}{\xi_\Lambda} - \frac{\mathbf{K}_\Lambda^\mathrm{T}\mathbf{R}_\mathcal{N}\mathbf{K}_\Lambda}{\xi_\Lambda} > 0. \tag{38}$$

Multiplying by ξ_Λ and taking into account that $\mathbf{P}_\Lambda = \xi_\Lambda\mathbf{O}_\Lambda$, where $\xi_\Lambda \in \mathbb{R}^+ \setminus \{0\}$ and $\mathbf{O}_\Lambda$ is a positive definite matrix, we obtain a similar LMI system to that of (19a)

$$\begin{aligned} \mathbf{P}_\Lambda &> 0, \\ \mathbf{P}_\Lambda - (\mathbf{A}_\mathcal{N}+\mathbf{B}_\mathcal{N}\mathbf{K}_\Lambda)^\mathrm{T}\mathbf{P}_\Lambda(\mathbf{A}_\mathcal{N}+\mathbf{B}_\mathcal{N}\mathbf{K}_\Lambda) - \mathbf{Q}_\mathcal{N} - \mathbf{K}_\Lambda^\mathrm{T}\mathbf{R}_\mathcal{N}\mathbf{K}_\Lambda &> 0. \end{aligned} \tag{39}$$

Finally, considering that $\mathbf{O}_\Lambda^{ij} = \mathbf{O}_\Lambda^{ji} = 0$ when $i \overset{\Lambda}{\nleftrightarrow} j$ holds, then the following topology conditions, similar to those of (19b), are trivially satisfied

$$i \overset{\Lambda}{\nleftrightarrow} j \Longrightarrow \begin{cases} \mathbf{K}_\Lambda^{ij} = \mathbf{K}_\Lambda^{ji} = 0, \\ \mathbf{P}_\Lambda^{ij} = \mathbf{P}_\Lambda^{ji} = 0. \end{cases} \tag{40}$$

Consequently, matrices $\mathbf{P}_\Lambda = \xi_\Lambda\mathbf{O}_\Lambda$ and $\mathbf{K}_\Lambda$ stabilize the overall system, provide us with a bound on the cost-to-go and satisfy all the communication constraints. ■

Next, we present the optimization algorithm, which generalizes the algorithm given in Muros et al. (2014b) by including constraints in the position value. 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 value constraints.

Design Algorithm 1

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 $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda$, solve, $\forall\Lambda \subseteq \mathcal{E}$

$$\max_{\mathbf{W}_\Lambda,\mathbf{Y}_\Lambda} \mathrm{Tr}(\mathbf{W}_\Lambda), \tag{41}$$

subject to (20), from where we obtain 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

$$\min_{\mathbf{P}_\Lambda}\left(\sum_\Lambda \mathrm{Tr}(\mathbf{P}_\Lambda)\right), \tag{42}$$

subject to (19), $\forall\Lambda \subseteq \mathcal{E}$, and the value constraint set given by $\mathcal{G}$. Therefore, we obtain $\mathbf{P}_\Lambda^{(r+1)}$.

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

$$\min_{\xi,\mathbf{K}_\Lambda}\left(\sum_\Lambda \xi_\Lambda\right), \tag{43}$$

subject to (37), $\forall\Lambda \subseteq \mathcal{E}$, and set $\mathcal{G}$. 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.

Remark 14. In (41) we solve one optimization problem per network topology, because it is more efficient in terms of time complexity. However, in (42) and (43) we have to solve a multiple-topology problem since different network topologies are present in the value constraint set given by $\mathcal{G}$.

Remark 15. Both (42) and (43) improve the sum of the traces of the set of matrices $\mathbf{P}_\Lambda$. Given that this sum is lower bounded, it can be deduced that the algorithm converges in a finite number of iterations.

Remark 16. In Step 1, we solve the optimization problem by using variables $(\mathbf{Y}_\Lambda, \mathbf{W}_\Lambda)$, and without considering value constraints, which are included afterwards. In Steps 2 and 3 we need to solve the problem by using $(\mathbf{K}_\Lambda, \mathbf{P}_\Lambda)$, in order to introduce the value constraint set $\mathcal{G}$ that is formulated in these variables. Hence, in Step 2, the affinity property required to consider (19) as an LMI is reached by taking $\mathbf{K}_\Lambda$ as the solution obtained in Step 1. Likewise, in Step 3, we utilize (37), with $\mathbf{O}_\Lambda$ the solution of $\mathbf{P}_\Lambda$ from Step 2, to modify $\mathbf{P}_\Lambda$ proportionally to the previous step.

The key of the algorithm proposed is to consider information of the previous steps for the control matrices to use (19) and (37) as LMI conditions. Hence, it is possible to include 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. (2014).

Theorem 2. *Let $(\mathcal{N}, \mathcal{E})$ be a network controlled by Control Scheme 1. If matrices $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda, \forall \Lambda \subseteq \mathcal{E}$, have been obtained by Design Algorithm 1, then the closed-loop system is asymptotically stable.*

Proof. The proof is built following Maestre et al. (2014), and it is based on the fact that function $r^v(\Lambda, \mathbf{x}_\mathcal{N})$, given by (7), is a decreasing function with a lower bound for the state trajectories of the closed loop with the proposed controller. Note that an upper bound on the cost-to-go of the closed-loop system is given by $\mathbf{x}_\mathcal{N}^{\mathrm{T}} \mathbf{P}_\Lambda \mathbf{x}_\mathcal{N}$. Then, according to Maestre et al. (2014) and Theorem 1, if matrices $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda$ are designed subject to (19), (20) and (37), as it is done in Design Algorithm 1, we can affirm that the cost-to-go of the closed-loop system controlled by linear feedback $\mathbf{K}_\Lambda$ decreases in time as long as topology Λ does not change.

Next, let us suppose that at a given multiple of k_s time steps, there is a switch of the topology. According to Control Scheme 1, this happens only if the new topology offers a lower value for the overall cost function $r^v(\Lambda, \mathbf{x}_\mathcal{N})$ that includes both control and communication costs. Otherwise, the topology is kept fixed for another k_s time steps. During these k_s time steps the control cost decreases and the communication cost remains constant, which again lead to a decrement of $r^v(\Lambda, \mathbf{x}_\mathcal{N})$. If we apply this argument recursively, it can be concluded that $r^v(\Lambda, \mathbf{x}_\mathcal{N})$ decreases in time. Eventually, the communication topology $\Lambda \in \boldsymbol{\Lambda}$ with the lowest communication cost is implemented and the cost-to-go of the closed-loop system becomes zero, which means that the overall state has been regulated to the origin. ■

Notice that the control performance of the system is affected by the network topology. This is a direct result of the zeros imposed on matrices $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda$, which reduce the degrees of freedom of the design problem. Given that this problem aims at optimizing the controller performance, it can be considered that

$$\mathbf{x}_\mathcal{N}^{\mathrm{T}} \mathbf{P}_{\Lambda_0} \mathbf{x}_\mathcal{N} \geq \mathbf{x}_\mathcal{N}^{\mathrm{T}} \mathbf{P}_\Lambda \mathbf{x}_\mathcal{N}, \tag{44a}$$

$$\mathbf{x}_\mathcal{N}^{\mathrm{T}} \mathbf{P}_\Lambda \mathbf{x}_\mathcal{N} \geq \mathbf{x}_\mathcal{N}^{\mathrm{T}} \mathbf{P}_{\mathrm{LQR}} \mathbf{x}_\mathcal{N}, \tag{44b}$$

where $\mathbf{P}_{\Lambda_0}$ and $\mathbf{P}_{\mathrm{LQR}}$ are the matrices corresponding to, respectively, the decentralized topology and the LQR solution for the centralized case, i.e., that with full communication.

Remark 17. Matrix $\mathbf{P}_{\mathrm{LQR}}$ represents a theoretical minimum and hence, condition (44b) is always verified. Likewise, it makes sense to assume matrix $\mathbf{P}_{\Lambda_0}$ as the more expensive one in terms of control and it can be guaranteed by simply adding to the design procedure an LMI condition equivalent to (44a).

Note that (44b) can be used to obtain a bound on the suboptimality of the scheme from a control perspective. In this sense, in order to determine the impact of satisfying the additional value constraints, we introduce the following index, which will be calculated once $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda$ are obtained.

Definition 5. The suboptimality index of a set of matrices $\mathbf{P}_\Lambda$, $\Lambda \subseteq \mathcal{E}$ is defined as

$$\eta = \frac{\sum\limits_{\Lambda \subseteq \mathcal{E}} \mathrm{Tr}(\mathbf{P}_\Lambda)}{2^{|\mathcal{E}|} \cdot \mathrm{Tr}(\mathbf{P}_{\mathrm{LQR}})}. \tag{45}$$

Notice that 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, index η decreases with the number of algorithm iterations applied.

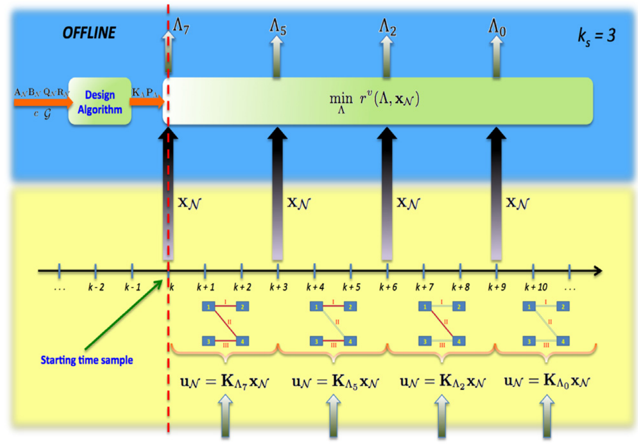

Fig. 2. Scheme overview.

To finish this subsection and summarize the behavior of the proposed scheme, an overview for the particular case of a 3-link network is presented in Fig. 2. First, control matrices $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda$ are calculated offline. Then, each k_s time steps (in this example $k_s = 3$), the bottom layer sends the states to the top one that calculates the optimal network topology for the following k_s time steps. Next, during these k_s time steps the corresponding control law is applied and the unnecessary links are disabled in a dynamical way. Finally, note that the relevance of the local controllers during the networked control scheme implementation can be dynamically measured at each time step by using the position value (Borm et al., 1992).

Remark 18. The design phase, that is, the offline obtention of matrices $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda$ that define the controller, is made in a centralized manner because centralized system information is needed, i.e., by solving the different optimization problems given for each possible topology. Once these matrices are calculated, in this work, we assume an implementation of the control scheme at each time step in a hierarchical fashion.

3.4. Additional constraints

In Sections 3.2.1 and 3.2.2, we have studied cases that consider the Shapley or position values of a *single* player with respect to a constant threshold (absolute constraints) or to another single player (relative ones). In this section, we extend the previous analysis to consider constraints that take into account the lineal combination of Shapley or position values of several players. Hence, focusing on the first type of absolute Shapley constraints (21), and given h players (links) that belong to set $\mathcal{E}$, the multiplayer constraint generalization is given by

$$\sum_{j=1}^{h} \kappa_j \phi_{l_j}(\mathcal{E}, \mathbf{r}^{\boldsymbol{v}}) < \mathcal{V}_\Sigma, \tag{46}$$

with $\kappa_j \in \mathbb{R}, j = 1, \ldots, h$ being the value weights and $\mathcal{V}_\Sigma \in \mathbb{R}$ a global threshold.

Then, by using again (12), operating with matrices and rearranging terms, it is possible to obtain the following LMI condition

$$\mathbf{D}_\Sigma > 0, \quad \text{with } \mathbf{D}_\Sigma = \begin{bmatrix} \mathcal{V}_\Sigma - c\sum\limits_{j=1}^{h}\kappa_j & 0 \\ 0 & -\sum\limits_{\Lambda\subseteq\mathcal{E}}\sum\limits_{j=1}^{h}\kappa_j m_{l_j\Lambda}\mathbf{P}_\Lambda \end{bmatrix}, \tag{47}$$

where the corresponding steady state constraint is given by

$$\mathcal{V}_\Sigma \geq \sum_{j=1}^{h} \kappa_j \phi_{l_j}^{ss}(\mathcal{E}, \mathbf{r}^{\boldsymbol{\nu}}). \tag{48}$$

Finally, note that it is possible to include state or input constraints by adding and/or modifying the LMI conditions, as it is shown in Alamo et al. (2006) and Kothare, Balakrishnan, and Morari (1996). For example, if we assume that the set of state constraints of the overall problem is defined by

$$\mathcal{X}_\mathcal{N} = \left\{ \mathbf{x}_\mathcal{N} : \mathbf{x}_\mathcal{N}^{\mathrm{T}} \mathbf{G} \mathbf{x}_\mathcal{N} \leq \rho \right\}, \tag{49}$$

with $\mathbf{G} > 0, \mathbf{G} \in \mathbb{R}^{n_{x_\mathcal{N}} \times n_{x_\mathcal{N}}}$ and $\rho \in \mathbb{R}^+ \setminus \{0\}$, we can rewrite the Shapley value constraint (21) as the following LMI condition (Muros et al., 2014a)

$$\mathbf{D}_\mathbf{a}' > 0, \quad \text{with } \mathbf{D}_\mathbf{a}' = \begin{bmatrix} \mathcal{V}_l - c - \gamma\rho & 0 \\ 0 & \gamma \mathbf{G} - \sum_{\Lambda \subseteq \mathcal{E}} m_{l\Lambda} \mathbf{P}_\Lambda \end{bmatrix}, \tag{50}$$

with $\gamma \in \mathbb{R}^+ \setminus \{0\}$ an optimization variable.

Operating in the same way with other value constraints, it is possible to obtain analogous LMIs for other multiplayer cases, or those that consider implicitly state constraints, whenever the resulting expression fulfills the requirements to be an LMI. Nevertheless, note that as more LMIs are introduced in the design phase, the system would be more conservative and it would be more difficult to find a feasible solution for matrices $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda$.

4. Simulation results

In this section, we show an academic example with four agents and three links, i.e., $\mathcal{N} = \{1, 2, 3, 4\}$, $\mathcal{E} = \{I, II, III\}$, corresponding to the configuration shown in Fig. 1(c). The eight different network topologies and their respective components are specified in Table 1. The matrices that define the subsystem dynamics are the following

$$\begin{aligned} &\mathbf{A}_{11} = \begin{bmatrix} 1 & 0.8 \\ 0 & 0.7 \end{bmatrix}, \quad \mathbf{A}_{22} = \begin{bmatrix} 1 & 0.9 \\ 0 & -2.5 \end{bmatrix}, \\ &\mathbf{A}_{33} = \begin{bmatrix} 1 & -2 \\ 0 & 3 \end{bmatrix}, \quad \mathbf{A}_{44} = \begin{bmatrix} 1 & 2.2 \\ 0 & 0.5 \end{bmatrix}, \\ &\mathbf{B}_{ii} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{A}_{ij} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B}_{ij} = \begin{bmatrix} 0 \\ 0.15 \end{bmatrix}, \quad i \neq j, \end{aligned} \tag{51}$$

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 stage cost 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 $l_{\max} = 20$.

In order to demonstrate the feasibility of the design procedure, we will consider the following two scenarios of value constraints that will be imposed on the overall problem. Note that the multiplayer constraint case has been taken into account in Scenario II.

- *Scenario I*:

$$\begin{aligned} &\phi_{II}(\mathcal{E}, \mathbf{r}^{\boldsymbol{\nu}}) > 0, \\ &\phi_{III}(\mathcal{E}, \mathbf{r}^{\boldsymbol{\nu}}) < 1, \end{aligned} \tag{52a}$$

$$\begin{aligned} &\pi_1(\mathcal{N}, \boldsymbol{\nu}, \mathcal{E}) < 1, \\ &\pi_1(\mathcal{N}, \boldsymbol{\nu}, \mathcal{E}) > \pi_4(\mathcal{N}, \boldsymbol{\nu}, \mathcal{E}). \end{aligned} \tag{52b}$$

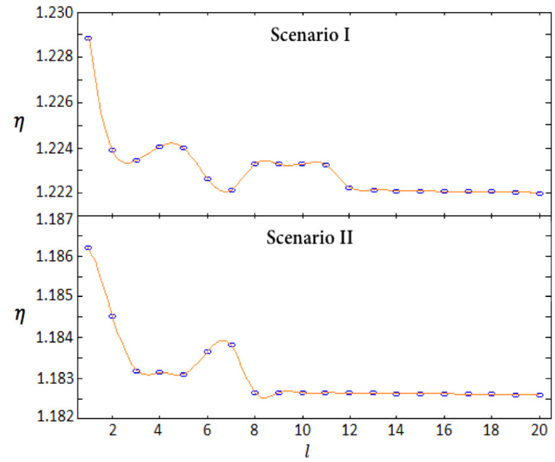

Fig. 3. Suboptimality index evolution of the design procedure with the number of iterations *l*.

- *Scenario II*:

$$\begin{aligned} &\phi_{II}(\mathcal{E}, \mathbf{r}^{\boldsymbol{\nu}}) < 0.8, \\ &\phi_{I}(\mathcal{E}, \mathbf{r}^{\boldsymbol{\nu}}) + \phi_{III}(\mathcal{E}, \mathbf{r}^{\boldsymbol{\nu}}) > 0.7, \end{aligned} \tag{53a}$$

$$\pi_1(\mathcal{N}, \boldsymbol{\nu}, \mathcal{E}) + \pi_4(\mathcal{N}, \boldsymbol{\nu}, \mathcal{E}) > \pi_3(\mathcal{N}, \boldsymbol{\nu}, \mathcal{E}). \tag{53b}$$

Notice that the constraints defined in (52) and (53) verify (29), (35) and (48). From both scenarios the corresponding value constraint sets $\mathcal{G}$ have been derived.

The design algorithm of Section 3.3 has been implemented using Matlab® LMI Control Toolbox (Gahinet, Nemirovskii, Laub, & Chilali, 1995) in a 2.7 GHz quad-core Intel® Core™ i5/4 GB RAM computer. More specifically, we have used the solver *mincx* which implements the *interior point methods* proposed in Nesterov and Nemirovskii (1994). In Fig. 3, it is possible to check 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, we have obtained matrices $\mathbf{K}_\Lambda$ and $\mathbf{P}_\Lambda, \forall \Lambda \subseteq \mathcal{E}$. For example, the resulting matrices for network topology Λ_2, for the case of Scenario I, are

$$\mathbf{K}_{\Lambda_2} = \begin{bmatrix} -0.3306 & -0.6582 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.2583 & 2.2856 & -0.0134 & 0.3015 & 0 & 0 \\ 0 & 0 & 0.0103 & -0.2080 & 0.1904 & -3.5048 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.2765 & -0.9987 \end{bmatrix},$$

$$\mathbf{P}_{\Lambda_2} = \begin{bmatrix} 3.5656 & 2.7105 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2.7105 & 4.6089 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7.9322 & 4.6286 & 0.1044 & -0.0001 & 0 & 0 \\ 0 & 0 & 4.6286 & 18.0402 & 0.1998 & -1.4256 & 0 & 0 \\ 0 & 0 & 0.1044 & 0.1998 & 3.4475 & -5.8143 & 0 & 0 \\ 0 & 0 & -0.0001 & -1.4256 & -5.8143 & 31.7684 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2.4712 & 3.5220 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3.5220 & 9.9426 \end{bmatrix}.$$

Note that these control matrices satisfy the communication constraints imposed by the network topology.

Once the design problem is solved, we test the two-layer networked control scheme proposed taking $k_s = 3$. Consider the initial state

$$\mathbf{x}_1 = \begin{bmatrix} 5 \\ 2 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}, \quad \mathbf{x}_3 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \quad \mathbf{x}_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{54}$$

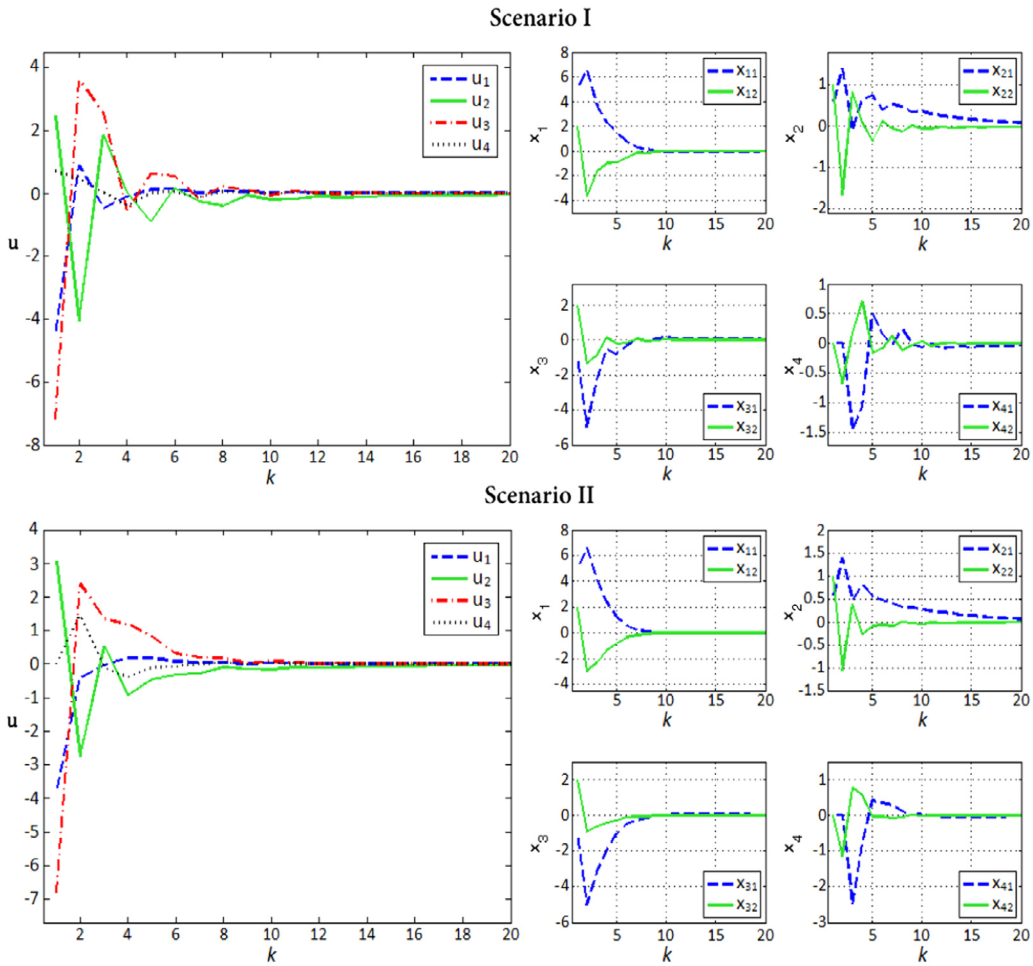

Fig. 4. Input and state trajectories.

At this point, we present some simulations of the given controller. First, Fig. 4 shows the input and state trajectories as a function of time after considering, respectively, constraints (52) and (53). Next, the evolution of the Shapley and position values and the network topologies are shown, respectively, in Figs. 5 and 6, for both scenarios and also without considering any value constraints. It is possible to see that both Shapley and position values satisfy the specifications. Furthermore, depending on the scenario considered, the evolution of the network topologies denotes a deactivation/predominance of link II, the most expensive/cheapest one due to constraints (52)/(53).

Note that both the Shapley and position values steady state do not depend on the constraints imposed. More specifically, the Shapley value of each link tends to the cost c per enabled link, but the position value steady state has a dependence on the number of links connected to each agent, as expected according to (18). In the network analyzed in this example, the position value tends to $c/2$ for agents 1 and 4, and tends to c for agents 2 and 3, because these agents are the end-points of one and two links, respectively. Finally, the network topology always tends to the one with the least communicational costs, i.e., the decentralized configuration.

Finally, in Fig. 7 the cumulated cost of the proposed coalitional algorithm for both scenarios is compared with the cumulated cost of considering full communication (centralized system) and no communication (decentralized system). As expected, the hierarchical-coalitional schemes outperform the decentralized one, and they are not far away from the centralized controller during the initial steps. Later, the communication cost makes the coalitional schemes to be the most appropriate ones.

5. Conclusions

In this paper, we have enhanced the design method proposed in Maestre et al. (2011b, 2014) for a coalitional networked control scheme. In particular, we have focused on how to include constraints on the links and the agents regarding the Shapley and position values, respectively, at the design phase. Moreover, the new conditions allow the designer to analyze if a certain coalitional control scheme verifies the constraints. In addition to this, we have proposed an iterative design method that improves the performance of the matrices that define the controller. In this sense, the simulation results have shown the good performance of the proposed scheme.

Future research will include the possibility of considering other game theory solution concepts for the design procedure. Moreover, scenarios with a limited topologies set $\mathbf{\Lambda}$, i.e., verifying $|\mathbf{\Lambda}| < 2^{|\mathcal{E}|}$, and methods for the distributed design of the local feedback controllers will be researched. Finally, the issues derived from the combinatorial explosion in systems with a large number of agents will also be addressed. In fact, possible applications to traffic, power, water networks and smart grids are currently object of study, by using randomized methods as (Castro, Gómez, & Tejada, 2009) to estimate the Shapley value.

Acknowledgments

The authors would like to thank the Editor and the anonymous reviewers for their valuable suggestions and comments that have substantially helped to improve this manuscript.

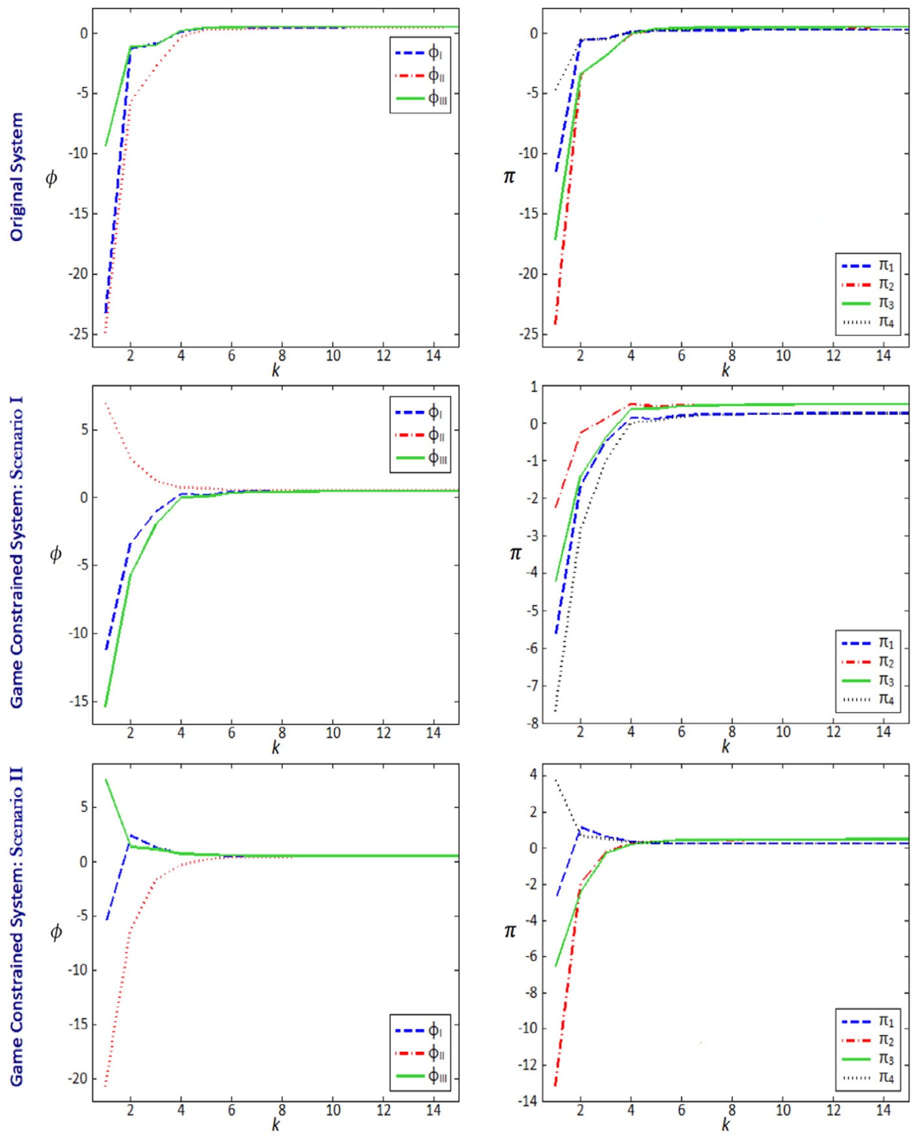

Fig. 5. Shapley and position values evolution.

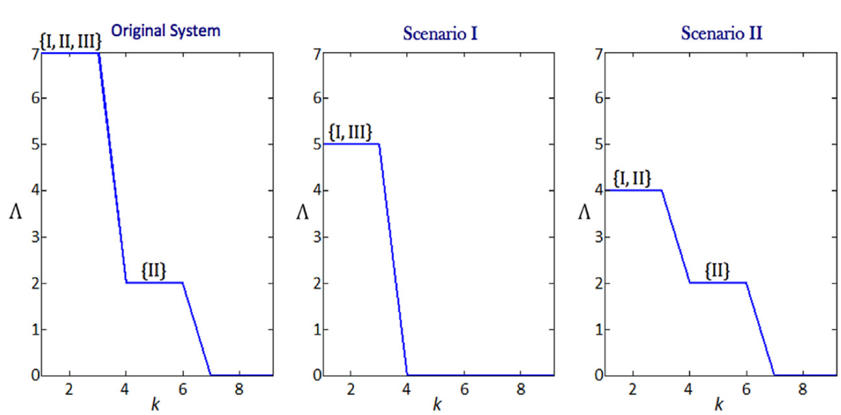

Fig. 6. Network topology evolution.

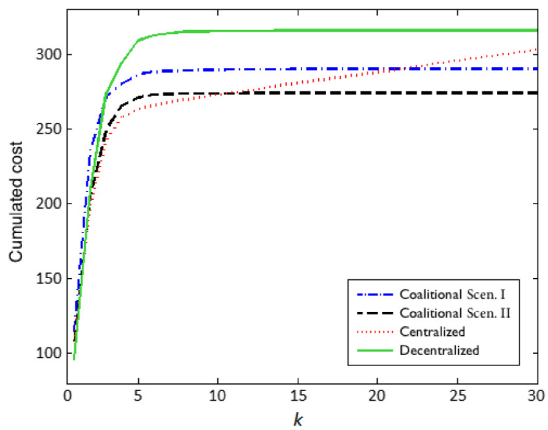

Fig. 7. Cumulated cost comparative study.

References

Alamo, T., Normey-Rico, J. E., Arahal, M. R., Limon, D., & Camacho, E. F. (2006). Introducing linear matrix inequalities in a control course. In *Proceedings of the 7th IFAC symposium on advances in control education, ACE 2006* (pp. 205–2010).

Bartlett, R. A., Wätchet, A., & Biegler, L. T. (2000). Active sets vs. interior point strategies for model predictive control. In *Proceedings of the 18st American control conference, ACC 2003* (pp. 4229–4233).

Bauso, D., & Timmer, J. (2009). Robust dynamic cooperative games. *International Journal of Game Theory*, *38*(1), 23–36.

Bauso, D., & Timmer, J. (2012). On robustness and dynamics in (un)balanced coalitional games. *Automatica*, *48*(10), 2592–2596.

Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, *35*(3), 407–427.

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

Boyd, S., El Ghaoui, L., Feron, E., & Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory*. Philadelphia, Pennsylvania, USA: Society for Industrial and Applied Mathematics.

Castro, J., Gómez, D., & Tejada, J. (2009). Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research*, *36*(5), 1726–1730.

Diestel, R. (2005). *Graph theory* (3rd ed.). Heidelberg, Germany: Springer-Verlag.

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

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

Fletcher, R., & Leyffer, S. (1998). Numerical experience with lower bounds for MIQP branch-and-bound. *SIAM Journal on Optimization*, *8*(2), 604–616.

Gahinet, P., Nemirovskii, A. S., Laub, A. J., & Chilali, M. (1995). *LMI control toolbox for use with MATLAB®*. The MathWorks, Inc.

Geletu, A. (2007). *Solving optimization problems using the matlab optimization toolbox — a tutorial*. TU-Ilmenau, Fakultät für Mathematik und Naturwissenschaften.

Gupta, R. A., & Chow, M.-Y. (2010). Networked control system: Overview and research trends. *IEEE Transactions on Industrial Electronics*, *57*(7), 2527–2535.

Hespanha, J. P., Naghshtabrizi, P., & Xu, Y. (2007). A survey of recent results in networked control systems. *Proceedings of the IEEE*, *95*(1), 138–162.

Jilg, M., & Stursberg, O. (2013). Optimized distributed control and topology design for hierarchically interconnected systems. In *Proceedings of the 12nd European control conference, ECC 2013* (pp. 4340–4346).

Kothare, M. V., Balakrishnan, V., & Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica*, *32*(10), 1361–1379.

Leyffer, S., & Mahajan, A. (2010). *Nonlinear constrained optimization: Methods and software, Technical report*. Argonne National Laboratory.

Maestre, J. M., Muñoz de la Peña, D., Camacho, E. F., & Alamo, T. (2011a). Distributed model predictive control based on agent negotiation. *Journal of Process Control*, *21*(5), 685–697.

Maestre, J. M., Muñoz de la Peña, D., Jiménez Losada, A., Algaba, E., & Camacho, E. F. (2011b). An application of cooperative game theory to distributed control. In *Proceedings of the 18th IFAC world congress* (pp. 9121–9126).

Maestre, J. M., Muñoz de la Peña, D., Jiménez Losada, A., Algaba, E., & Camacho, E. F. (2014). A coalitional control scheme with applications to cooperative game theory. *Optimal Control Applications & Methods*, *35*(5), 592–608.

Maestre, J. M., Muros, F. J., Fele, F., & Camacho, E. F. (2015). An assessment of coalitional control in water systems. In *Proceedings of the 14th European control conference, ECC 2015* (pp. 3291–3296).

Maestre, J. M., & Negenborn, R. R. (Eds.) (2014). *Intelligent systems, control and automation: science and engineering*: *vol. 69. Distributed model predictive control made easy*. Springer.

Millán, P., Orihuela, L., Vivas, C., Rubio, F. R., Dimarogonas, D. V., & Johansson, K. H. (2013). Sensor-network-based robust distributed control and estimation. *Control Engineering Practice*, *21*(9), 1238–1249.

Muros, F. J., Maestre, J. M., Algaba, E., Alamo, T., & Camacho, E. F. (2014a). Constraints on the Shapley value for a coalitional control system. In *Proceedings of the 13rd European control conference, ECC 2014* (pp. 1223–1228).

Muros, F. J., Maestre, J. M., Algaba, E., Alamo, T., & Camacho, E. F. (2014b). An iterative design method for coalitional control networks with constraints on the Shapley value. In *Proceedings of the 19th IFAC world congress* (pp. 1188–1193).

Muros, F. J., Maestre, J. M., Algaba, E., Ocampo-Martinez, C., & Camacho, E. F. (2015). An application of the Shapley value to perform system partitioning. In *Proceedings of the 33rd American control conference, ACC 2015* (pp. 2143–2148).

Nedić, A., & Bauso, D. (2013). Dynamic coalitional TU games: Distributed bargaining among players' neighbors. *IEEE Transactions on Automatic Control*, *58*(6), 1363–1376.

Negenborn, R. R., De Schutter, B., & Hellendoorn, J. (2006). Multi-agent model predictive control of transportation networks. In *Proceedings of the 3rd IEEE international conference on networking, sensing and control, ICNSC 2006* (pp. 296–301).

Negenborn, R. R., van Overloop, P.-J., Keviczky, T., & De Schutter, B. (2009). Distributed model predictive control of irrigation canals. *Networks and Heterogeneous Media*, *4*(2), 359–380.

Nesterov, Y., & Nemirovskii, A. S. (1994). *Interior-point polynomial algorithms in convex programming*. Philadelphia, Pennsylvania, USA: Society for Industrial and Applied Mathematics.

Núñez, A., Ocampo-Martinez, C., De Schutter, B., Valencia, F., López, J., & Espinosa, J. (2013). A multiobjective-based switching topology for hierarchical model predictive control applied to a hydro-power valley. In *Proceedings of the 3rd IFAC international conference on intelligent control and automation science, ICONS 2013* (pp. 534–539).

Núñez, A., Ocampo-Martinez, C., Maestre, J. M., & De Schutter, B. (2015). Time-varying scheme for non-centralized model predictive control of large-scale systems. *Mathematical Problems in Engineering*,.

Raman, R., & Grossmann, I. E. (1991). Relation between MILP modelling and logical inference for chemical process synthesis. *Computers & Chemical Engineering*, *15*(2), 73–84.

Saad, W., Han, Z., Debbah, M., Hjørungnes, A., & Başar, T. (2009). Coalitional game theory for communication networks. *IEEE Signal Processing Magazine*, *26*(5), 77–97.

Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control — a review. *Journal of Process Control*, *19*(5), 723–731.

Shapley, L. S. (1953). A value for *n*-person games. In H. W. Kuhn, & A. W. Tucker (Eds.), *Annals of Mathematics Studies*: *28. Contributions to the theory of games II* (pp. 307–317).

Skogestad, S., & Postlethwaite, I. (2001). *Multivariable feedback control — analysis and design* (2nd ed.). New York, USA: John Wiley and Sons.

Slikker, M. (2005). A characterization of the position value. *International Journal of Game Theory*, *33*(4), 505–514.

Trodden, P., & Richards, A. G. (2009). Adaptive cooperation in robust distributed model predictive control. In *Proceedings of the 24th IEEE International symposium on intelligent control, ISIC 2009* (pp. 896–901).

Wright, S. J. (1997). Applying new optimization algorithms to model predictive control. In J. C. Kantor, C. E. Garcia, & B. Carnahan (Eds.), *Fifth international conference on chemical process control*, CPC V. (pp. 147–155). AIChE and CACHE.

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

Zhang, F. (2005). *Numerical methods and algorithms*: *vol. 4. The schur complement and its applications*. New York, USA: Springer Verlag.

Francisco Javier Muros received the M.S. degree in Telecommunications Engineering from the University of Seville in 2004. Later, he acquired a wide experience in several works related to the electricity sector. He received a Master's degree in Design, Construction and Maintenance of High Voltage Electrical Transmission Infrastructure from the Comillas Pontifical University, Madrid in 2014, being currently Coordinator of this Master. He performed a Ph.D.-stay in Maastricht, The Netherlands, in 2016. At present he is with the Department of Systems and Automation Engineering at the University of Seville.

He is author and co-author of more than twenty publications including book chapters, journal papers, and conference papers. He has participated in the European Union Project FP7-ICT-ICT-2013.3.4-611281 (DYMASOS) and in several MINECO-Spain Projects. His research interests focus on cooperative game theory, coalitional and distributed control.

José María Maestre received the joint B.Eng. and M.Eng. degree in telecommunication engineering from the University of Seville in 2005, and the Ph.D. on automation and robotics in 2010 by the same university, where he also works as associate professor. He has also worked in LTH at Lund University (guest researcher), TU Delft (postdoc), University of Cadiz (assistant professor), and Tokyo Institute of Technology (visiting professor). Besides his Ph.D., which was awarded with the extraordinary prize of the University of Seville, he has a master degree in smart homes and intelligent buildings from the Universidad Politécnica de Madrid, a master in telecommunications economics from the Universidad Nacional de Educación a Distancia, and a master in economics and development from the University of Seville. His main research interests are the control of distributed systems and the integration of service robots in the smart home. He has authored and coauthored more than one hundred conference and journal papers regarding these topics. He is also editor of the books *Service Robotics within the Digital Home: Applications and Future Prospects* (Springer, 2011), *Distributed Model Predictive Control Made Easy* (Springer, 2014), and *Domótica para Ingenieros* (Paraninfo, 2015). Finally, he is one of the founders of the technological firms *Idener*, *Alacarta Tecnologías Integradas*, and *Eskesso*.

Encarnación Algaba is Associate Professor in the Department of Applied Mathematics II of the Advanced Engineering School at the University of Seville and member of the Mathematics Research Institute at the University of Seville (IMUS). She received her Ph.D. degree in Mathematical Sciences from the University of Seville. Her main research line is about Cooperative Games on Combinatorial Structures with relevant contributions in the area. She is one of the Spanish representatives in the SING (Spain, Italy, Netherlands Game Theory) Committee from 2012 and member of the Scientific Committee in SING 10 (Cracow, Poland 2014), European meeting on Game Theory SING 11-GTM 2015 (St. Petersburg, Russia 2015), SING 12 (Odense, Denmark 2016) and SING 13 (Paris, France 2017). She has acted as evaluator of projects at national and European level and she belongs to the national expert evaluators board from 2014. She has organized diverse courses, streams in International Conferences and workshops on Game Theory. She has been invited speaker and guest researcher in various universities, has given invited Summer Courses and numerous talks in International Conferences.

Teodoro Alamo was born in Spain in 1968. He received his M.Eng. degree in Telecommunications Engineering from the Polytechnic University of Madrid (Spain) in 1993 and his Ph.D. in Telecommunications Engineering from the University of Seville in 1998. From 1993 to 2000 he was assistant professor in the Department of Systems and Automation Engineering at the University of Seville. He was associate professor from 2001 to 2010. Since March 2010 he is full professor in the same department. He was at the École Nationale Supérieure des Télécommunications (Télécom Paris) from September 1991 to May 1993. Part of his Ph.D. was done at RWTH Aachen, Alemania, June–September 1995.

He is the author or coauthor of more than 160 publications including books, book chapters, journal papers, conference proceedings and educational books (google scholar profile available at http://scholar.google.es/citations?user=W3ZDTkIAAAAJ&hl=en). His current research interests are in decision making, model predictive control, machine learning, randomized algorithms and optimization strategies. He has co-founded the spin off company *Optimal Performance* (University of Seville, Spain).

Eduardo F. Camacho (F11) received the Ph.D. degree in electrical engineering from the University of Seville, Seville, Spain.

He is now a Full Professor with the Department of System and Automation Engineering, University of Seville. He is author of *Model Predictive Control in the Process Industry* (1995), *Advanced Control of Solar Plants* (1997), *Model Predictive Control* (1999), (Springer, 2004, 2nd edition), *Control e Instrumentación de Procesos Químicos* (Ed. Síntesis), *Control of Dead-time Processes* (Springer, 2007) and *Control of Solar Systems* (Springer, 2011). He has served on various IFAC technical committees and chaired the IFAC publication Committee from 2002–2005. He was the president of the European Control Association (2005–2007) and chaired the IEEE/CSS International Affairs Committee (2003–2006), Chair of the IFAC Policy Committee and a member of the IEEE/CSS Board of Governors. He has acted as evaluator of projects at national and European level and was appointed Manager of the Advanced Production Technology Program of the Spanish National R&D Program (1996–2000). He was one of the Spanish representatives on the Program Committee of the Growth Research program and expert for the Program Committee of the NMP research priority of the European Union. He has carried out reviews and editorial work for various technical journals and many conferences. At present he is one of the Editors of the *IFAC Journal, Control Engineering Practice*, Editor-at-Large of the *European Journal of Control* and Subject Editor of *Optimal Control: Methods and Applications*.

Dr. Camacho is an IEEE and IFAC Fellow. He was Publication Chair for the IFAC World Congress 2002, General Chair of the joint 44th IEEE CDC-ECC 2005, and co-General Chair of the joint 50th IEEE CDC-ECC 2011.