

This is a repository copy of *Hierarchical distributed model predictive control based on fuzzy negotiation* in the IRIS Institutional Repository of the Politecnico di Milano.

**Version:** Author Accepted Version.

**Citation:** Masero, E., Francisco, M., Maestre, J. M., Revollar, S., & Vega, P. (2021). Hierarchical distributed model predictive control based on fuzzy negotiation. *Expert Systems with Applications*, 176, 114836. DOI: <https://doi.org/10.1016/j.eswa.2021.114836>

To cite this publication, please use the final published version (if applicable). Please check the document version above.

**Copyright:** Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy:** Please contact us ([catalogoricerca@polimi.it](mailto:catalogoricerca@polimi.it)) and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

# Hierarchical distributed model predictive control based on fuzzy negotiation

Eva Masero<sup>a,\*</sup>, Mario Francisco<sup>b</sup>, José M. Maestre<sup>a</sup>, Silvana Revollar<sup>b</sup>, Pastora Vega<sup>c</sup>

<sup>a</sup>*Department of Systems Engineering and Automation, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Av. Camino de los Descubrimientos, s/n., 41092 Seville, Spain.*

<sup>b</sup>*Department of Computing and Automation, Escuela Técnica Superior de Ingeniería Industrial, Universidad de Salamanca, Av. Fernando Ballesteros, Béjar, 37700 Salamanca, Spain.*

<sup>c</sup>*Department of Computing and Automation, Facultad de Ciencias, Universidad de Salamanca, Plaza de la Merced, s/n, 37008 Salamanca, Spain.*

---

## Abstract

This work presents a hierarchical distributed model predictive control approach for multiple agents with cooperative negotiations based on fuzzy inference. Specifically, a fuzzy-based two-layer control architecture is proposed. In the lower control layer, there are pairwise negotiations between agents according to the couplings and the communication network. The resulting pairwise control sequences are sent to a coordinator in the upper control layer, which merges them to compute the final ones. Furthermore, conditions to guarantee feasibility and stability in the closed-loop system are provided. The proposed control algorithm has been tested on an eight-coupled tank plant via simulation.

*Keywords:* Model predictive control, Hierarchical distributed control, Pairwise negotiations, Fuzzy logic, Multi-agent systems, Stability.

---

## 1. Introduction

Centralized control methods generally achieve the best performance, for all the measurements are gathered at a single point where decisions are taken using full information. Nevertheless, there are compelling reasons to consider distributed control architectures. Firstly, for complex large-scale processes, it is merely not feasible to compute control actions in this manner due to timing constraints. Additionally, other systems such as traffic, power, and water networks are geographically spread, i.e., there are several independent entities with decision-making capabilities and possibly conflicting goals. In these situations, it might be necessary (or preferable) to provide each subsystem with a local controller with communication capabilities to attain a coordinated solution with the rest of the system.

---

\*Corresponding author.

*Email addresses:* [evamasero@us.es](mailto:evamasero@us.es) (Eva Masero), [mfs@usal.es](mailto:mfs@usal.es) (Mario Francisco), [pepemaestre@us.es](mailto:pepemaestre@us.es) (José M. Maestre), [srevolla@usal.es](mailto:srevolla@usal.es) (Silvana Revollar), [pvega@usal.es](mailto:pvega@usal.es) (Pastora Vega)

In general, it is better to carry out local controller negotiations in a proactive fashion for the sake of coordination, i.e., based on sequences of future states and inputs that provide local subsystems with future coupling information. For this reason, Model Predictive Control (MPC), a family of control methods that uses a system model to predict its evolution and calculates optimal control actions for a given horizon (Rossiter, 2003), has become a popular choice for distributed implementations. The MPC framework allows dealing explicitly with multiple variables, constraints, and disturbance information, which are very convenient features in this context.

During the last decade, numerous Distributed Model Predictive Control (DMPC) schemes have been proposed with significant differences on issues such as system decomposition, coupling sources, and control goals, to name a few. Depending on the degree of cooperation, the DMPC method can be decentralized, non-cooperative, or cooperative. Hierarchical architectures are also used for coordination, e.g., to unify the control signals proposed by different agents for some interconnecting variables, with iterative and ‘price-based’ algorithms. See (Scattolini, 2009; Negenborn & Maestre, 2014) for reviews on this topic.

However, coordination comes at a price in terms of communication overhead. For example, Maestre et al. (2015) show that some DMPC methods may require each agent to exchange thousands and even millions of floats per time instant to attain an optimal solution. Whether this is a limiting issue depends on the particular application, but it might be preferable to decrease the degree of optimality if the coordination burden is reduced. Numerous studies can be found in the literature about this research line, i.e., Berglind et al. (2012) and Mi et al. (2019) propose a self-triggered controller that decides the time to update the control input, thus reducing the communication costs. Other studies such as (Maestre et al., 2009; Xing et al., 2007) also focus on reducing the communication steps and power.

Another approach to tackle excessive communication burden is fuzzy logic, which can be used as a tool for agent negotiations. The fuzzy logic theory was introduced by Zadeh (1973), allowing for the consideration of imprecise or uncertain information in the same way as human cognition and perception. One of the advantages of considering fuzzy logic in the negotiation framework is the possibility to characterize a suitable control action employing a large variety of criteria that, otherwise, would produce unwanted results or would be very difficult to quantify. For example, given a constraint, the control action can be considered ‘good’ or ‘bad’ concerning that constraint, instead of considering only a feasible or non-feasible action. Other criteria that can be considered for assessing control actions in the fuzzy framework are economy, process safety, and environmental effects.

The use of fuzzy logic as a negotiation tool is scarce in the literature. Kosonen (2003) presents a distributed traffic signal control system with a fuzzy inference system considering the economy, fluency, environment, and safety as imprecise criteria for negotiation. The results are discrete control actions for traffic lights. Sahebjamnia et al. (2016) propose a fuzzy Q-learning algorithm in the distributed control of chemical plants. Negotiation with fuzzy constraints for the planning and scheduling of supply chains is presented by Hsu et al. (2016). Other studies such as (Kowalczyk, 2002) and (Thibodeau et al., 2013) include decision-making process using fuzzy logic in the field of e-commerce, where this technique is popular.

This work proposes a two-layer hierarchical DMPC control architecture that uses fuzzy logic in the negotiation process. In the low-level control layer, agents negotiate in pairs according to the couplings and the possibilities offered by the communication network, assuming that the variables of interest that belong to other players will follow their current evolution. The pairwise negotiation is based on a modification of the work proposed by Maestre et al. (2011a), where a two-communication-step mechanism is considered to generate sub-optimal yet stable solutions with very low communication burden. One of the disadvantages of that method is the combinatorial explosion of possible inputs when considering two or more subsystems. In general, the more agents are involved in a negotiation, the more communication steps are required to attain an agreement. In this context, fuzzy negotiation presents a remarkable advantage over other methods since computing the final input does not require as many communication steps, but just merging all the inputs proposed by the involved agents. Likewise, the proposed scheme allows agents to preserve their state and even their internal model private, which can be useful in problems where agents are willing to cooperate but may have concerns in this regard. Thus, the original negotiation scheme is modified to deal with multi-agent systems by adding an extra fuzzy step that reduces the communication burden. Moreover, it is considered additional criteria in the decision-making process to soften the trajectory of control signals, which are prone to abrupt changes in the original method.

As a result of the pairwise negotiations, a set of stabilizing sequences or proposals for the system is obtained. In particular, we consider three possible input trajectories for each subsystem: i) its shifted input (computed as an extension of the input applied at the previous time step), which guarantees stability in the closed-loop subsystem when feedback is selected properly; ii) its selfish input, which is the best control sequence locally computed; and iii) the altruistic input, which consists of the action wished and computed by its neighbor. This triplet seems a reasonable heuristic choice to obtain the final inputs via a fuzzification process, as proposed by Maestre et al. (2011a). Note that the referred method was compared with other well-known schemes in a four-tank benchmark, proving good performance despite its heuristic nature (Alvarado et al., 2011). Afterwards, the set of proposal is sent to the fuzzy-based supervisory layer, which merges them to obtain new control sequences for the system. Finally, stability is guaranteed for this fast decision-making scheme. Note that hybrid Fuzzy-MPC approaches are rare in the literature, except for (Francisco et al., 2019), a previous work where fuzzy negotiation was applied to a four-coupled-tank system, but without stability guarantees and limited to systems with only two agents. We would also like to stress that hierarchical-distributed architectures are common in this context. Likewise, some schemes rely on coordinators or supervisory layers (Zafra-Cabeza et al., 2011; Doan et al., 2014; Saad et al., 2018).

The approach benefits are shown via simulation of a non-linear eight-coupled tanks system, designed as an extension of the quadruple-tank process (Johansson, 2000). The quadruple-tank process has significantly been used as a benchmark to analyze distributed MPC techniques (Alvarado et al., 2011), to test a robust tube-based MPC for tracking (Limón et al., 2010), to study multi-variable dead times (Shneiderman & Palmor, 2010), and for the stabilization of linear systems with delays (El Haoussi et al., 2011).

The rest of the article is organized as follows. Section 2 introduces the problem formulation and the control goal. In Section 3, the proposed control architecture, the control algorithm, and the fuzzy pairwise negotiation are detailed. Section 4 displays stability properties and the procedure used to design the controller. Section 5 illustrates the proposed fuzzy DMPC scheme and the fuzzy negotiation criteria employed in the eight-coupled tanks benchmark. The results obtained from the simulation of this plant are shown in Section 6. Finally, conclusions are summarized in Section 7.

*Notation:*  $\mathbb{N}_{0+}$  and  $\mathbb{R}_+$  are, respectively, the sets of non-negative integers and positive real numbers.  $\mathbb{R}^n$  refers to an  $n$ -dimension Euclidean space. The scalar product of vectors  $a, b \in \mathbb{R}^n$  is denoted as  $ab^\top$  or  $a \cdot b$ . Given sets  $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^n$ , the Cartesian product is  $\mathcal{X} \times \mathcal{Y} \triangleq \{(x, y) : x \in \mathcal{X}, y \in \mathcal{Y}\}$ . If  $\{\mathcal{X}_i\}_{i \in \mathcal{N}}$  is a family of sets indexed by  $\mathcal{N}$ , then the Cartesian product is  $\times_{i \in \mathcal{N}} \mathcal{X}_i \triangleq \mathcal{X}_1 \times \cdots \times \mathcal{X}_N = \{(x_1, \dots, x_N) : x_1 \in \mathcal{X}_1, \dots, x_N \in \mathcal{X}_N\}$ . Moreover, the Minkowski sum is  $\mathcal{X} \oplus \mathcal{Y} \triangleq \{x + y : x \in \mathcal{X}, y \in \mathcal{Y}\}$ . The set subtraction operation is symbolized by  $\setminus$ . The image of a set  $\mathcal{X} \subseteq \mathbb{R}^n$  under a linear mapping  $A : \mathbb{R}^n \mapsto \mathbb{R}^m$  is  $A\mathcal{X} \triangleq \{Ax : x \in \mathcal{X}\}$ . The integral of the squared error  $\varepsilon$  is defined as  $ISE = \int_0^\infty \varepsilon^2 dt$ . The  $N_p$ -long future input sequence at time instant  $k$  is defined as  $U(k) = [u(k), u(k+1), \dots, u(k+N_p-1)]^\top$ , and the optimal sequence is denoted as  $U^*(k) = [u^*(k), u^*(k+1), \dots, u^*(k+N_p-1)]^\top$ .

## 2. Problem Formulation

Let the system be composed of a set  $\mathcal{N} = \{1, 2, \dots, N\}$  of input-coupled subsystems whose state evolution is

$$x_i(k+1) = A_i x_i(k) + B_{ii} u_i(k) + w_i(k), \quad (1)$$

where  $k \in \mathbb{N}_{0+}$  denotes the time instant;  $x_i \in \mathbb{R}^{q_i}$  and  $u_i \in \mathbb{R}^{r_i}$  are, respectively, the state and input vectors of each subsystem  $i \in \mathcal{N}$ , constrained in the convex sets containing the origin in their interior  $\mathcal{X}_i \triangleq \{x_i \in \mathbb{R}^{q_i} : A_{x,i} x_i \leq b_{x,i}\}$  and  $\mathcal{U}_i \triangleq \{u_i \in \mathbb{R}^{r_i} : A_{u,i} u_i \leq b_{u,i}\}$ , respectively; and  $A_i \in \mathbb{R}^{q_i \times q_i}$  and  $B_{ii} \in \mathbb{R}^{q_i \times r_i}$  are matrices of proper dimensions. The measurable disturbances vector  $w_i \in \mathbb{R}^{q_i}$  represents the coupling with other subsystems  $j$  belonging to the set of neighbors  $\mathcal{N}_i \triangleq \{j \in \mathcal{N} \setminus \{i\} : B_{ij} \neq 0\}$ , i.e.,

$$w_i(k) = \sum_{j \in \mathcal{N}_i} B_{ij} u_j(k), \quad (2)$$

where  $u_j \in \mathbb{R}^{r_j}$  is the input vector of subsystem  $j \in \mathcal{N}_i$ , and matrix  $B_{ij} \in \mathbb{R}^{q_i \times r_j}$  models the input coupling between  $i$  and  $j$ . Moreover,  $w_i$  is bounded in a convex set  $\mathcal{W}_i \triangleq \bigoplus_{j \in \mathcal{N}_i} B_{ij} \mathcal{U}_j$  due to the system constraints. The neighborhood affected by agent  $i$  is defined as  $\mathcal{M}_i \triangleq \{j \in \mathcal{N} \setminus \{i\} : B_{ji} \neq 0\}$ .

From the global viewpoint, the overall system evolution can be aggregated as

$$x_{\mathcal{N}}(k+1) = A_{\mathcal{N}} x_{\mathcal{N}}(k) + B_{\mathcal{N}} u_{\mathcal{N}}(k), \quad (3)$$

where  $A_{\mathcal{N}} = [A_{ij}]_{i,j \in \mathcal{N}}$  and  $B_{\mathcal{N}} = [B_{ij}]_{i,j \in \mathcal{N}}$  are, respectively, the state and input-to-state matrices of the overall system. Global state and input vectors are constrained in  $\mathcal{X}_{\mathcal{N}} = \times_{i \in \mathcal{N}} \mathcal{X}_i$  and  $\mathcal{U}_{\mathcal{N}} = \times_{i \in \mathcal{N}} \mathcal{U}_i$ , respectively. Regarding mutual interaction,  $w_{\mathcal{N}}$  is implicitly included in (3).

### 2.1. Control objective

The control objective is to drive the system towards the origin of the state space guaranteeing that constraints are satisfied, and minimizing the sum of the local cost functions. At each time instant  $k$ , the cost function of subsystem  $i \in \mathcal{N}$  is calculated based on the predicted trajectories of its states and inputs over a future window of length  $N_p$  (the so-called prediction horizon):

$$J_i(x_i(k), U_i(k), U_j(k)) = \sum_{t=0}^{N_p-1} L_i(x_i(k+t), u_i(k+t)) + F_i(x_i(k+N_p)), \quad (4)$$

where  $L_i(\cdot)$  is the stage cost function, and  $F_i(\cdot)$  is the terminal cost function defined as

$$L_i(x_i(k+t), u_i(k+t)) = x_i(k+t)^\top Q_i x_i(k+t) + u_i(k+t)^\top R_i u_i(k+t),$$

$$F_i(x_i(k+N_p)) = x_i(k+N_p)^\top P_i x_i(k+N_p),$$

with  $Q_i$  being a semi-positive definite matrix, and  $R_i, P_i$  being positive-definite matrices.

## 3. Proposed Control Architecture

Highly coupled agents communicate using a network that can be modeled as the indirect graph  $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ , where  $\mathcal{N}$  is the set of agents, and  $\mathcal{L}$  is the set of bidirectional links  $\mathcal{L} \subseteq \mathcal{L}^{\mathcal{N}} = \{\{i, j\} : \{i, j\} \subseteq \mathcal{N}, i \neq j\}$ , i.e., a link  $l_{ij} \in \mathcal{L}$  connects agents  $i$  and  $j$  providing a bidirectional information flow. If two agents are connected by a communication link  $l_{ij}$ , they follow a pairwise cooperative scheme to find a consensus on their control sequences via a multi-layer fuzzy negotiation algorithm.

The multi-layer fuzzy-based control architecture is shown in Fig. 1. The low-level control layer uses DMPC and fuzzy techniques to deal with the pairwise-agent negotiations assuming that interest variables that belong to other agents follow their current evolutions. Then, the sequences resulting from the pairwise negotiations are sent to a supervisory layer, which merges and *fuzzifies* them to compute the final control sequence for the whole system.

### 3.1. Low-level control layer

A DMPC algorithm for multiple agents is performed by the lower control layer. Specifically, fuzzy-based negotiations are made in pairs considering the couplings with their neighboring subsystems, which are assumed to hold their current trajectories. To this end, it is

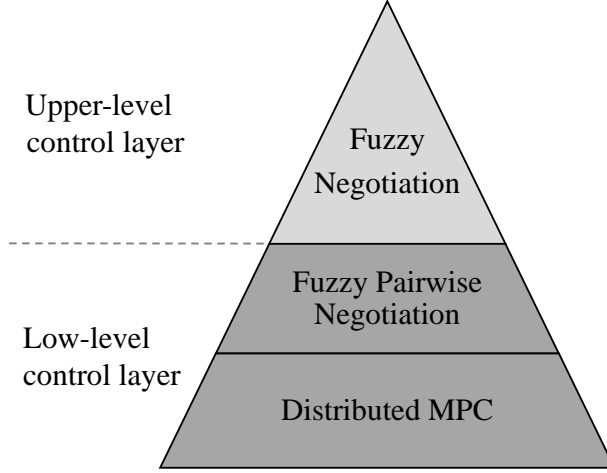


Figure 1: The multi-layer fuzzy-based control architecture.

used a shifted sequence of agent  $i$ , which is defined by adding  $K_i x_i(k + N_p)$  to the sequence chosen at the previous time step  $U_i(k - 1)$ :

$$U_i^s(k) = \begin{bmatrix} u_i(k + 1|k - 1) \\ u_i(k + 2|k - 1) \\ \vdots \\ u_i(k + N_p - 1|k - 1) \\ K_i x_i(k + N_p|k - 1) \end{bmatrix} = \begin{bmatrix} u_i^s(k) \\ u_i^s(k + 1) \\ \vdots \\ u_i^s(k + N_p - 2) \\ u_i^s(k + N_p - 1) \end{bmatrix}. \quad (5)$$

### 3.1.1. Distributed MPC algorithm for multiple agents

Algorithm 1 extends the DMPC scheme proposed by Maestre et al. (2011a) for  $N$  subsystems without combinatorial explosion.

#### Algorithm 1 (Stage 1): Agent Proposals.

At each time step  $k$ :

1. Firstly, agent  $i$  measures its local state  $\tilde{x}_i(k)$ .
2. Agent  $i$  calculates its shifted trajectory  $U_i^s(k)$ , and sends it to its neighbors.
3. Agent  $i$  minimizes its cost function considering that neighbor  $j \in \mathcal{N}_i$  applies its shifted trajectory  $U_j^s(k)$ . It is assumed that the rest of the neighboring subsystems  $l \in \mathcal{N}_i \setminus \{j\}$  follows their current control trajectories  $U_l^s(k)$ . Specifically, agent  $i$  solves

$$U_i^*(k) = \arg \min_{U_i(k)} J_i(x_i(k), U_i(k), U_j^s(k), U_l^s(k)), \quad (6)$$

subject to

$$x_i(k + t + 1) = A_i x_i(k + t) + B_{ii} u_i(k + t) + B_{ij} u_j(k + t) + \sum_{l \in \mathcal{N}_i \setminus \{j\}} B_{il} u_l(k + t),$$

$$\begin{aligned}
x_i(k) &= \tilde{x}_i(k), \quad i \in \mathcal{N}, \\
x_i(k+t) &\in \mathcal{X}_i, \quad t = 0, \dots, N_p - 1, \\
x_i(k+N_p) &\in \Omega_i, \\
u_i(k+t) &\in \mathcal{U}_i, \quad t = 0, \dots, N_p - 1, \\
u_j(k+t) &= u_j^s(k+t), \quad t = 0, \dots, N_p - 1, \\
u_l(k+t) &= u_l^s(k+t), \quad t = 0, \dots, N_p - 1,
\end{aligned}$$

where set  $\Omega_i$  is imposed as terminal state constraint of agent  $i$ . Details regarding the calculation of  $\Omega_i$  are given in Section 4.

4. Agent  $i$  optimizes again its cost  $J_i(\cdot)$  maintaining its optimal input sequence  $U_i^*(k)$  fixed to find the input sequence wished for its neighbor  $j$ , i.e.,  $U_j^{wi}(k)$ . Here, it is also assumed that subsystems  $l$  follow their current trajectories. To this end, agent  $i$  solves

$$U_j^{wi}(k) = \arg \min_{U_j(k)} J_i(x_i(k), U_i^*(k), U_j(k), U_l^s(k)), \quad (7)$$

subject to

$$x_i(k+t+1) = A_i x_i(k+t) + B_{ii} u_i(k+t) + B_{ij} u_j(k+t) + \sum_{l \in \mathcal{N}_i \setminus \{j\}} B_{il} u_l(k+t),$$

$$\begin{aligned}
x_i(k) &= \tilde{x}_i(k), \quad i \in \mathcal{N} \\
x_i(k+t) &\in \mathcal{X}_i, \quad t = 0, \dots, N_p - 1, \\
x_i(k+N_p) &\in \Omega_i, \\
u_i(k+t) &= u_i^*(k+t), \quad t = 0, \dots, N_p - 1, \\
u_j(k+t) &\in \mathcal{U}_j, \quad j \in \mathcal{N}_i \quad t = 0, \dots, N_p - 1, \\
u_l(k+t) &= u_l^s(k+t), \quad t = 0, \dots, N_p - 1.
\end{aligned}$$

5. Finally, agent  $i$  sends  $U_j^{wi}(k)$  to agent  $j$ , and receives  $U_i^{wj}(k)$ .

### 3.1.2. Fuzzy pairwise negotiations

Following Stage 1 of the proposed algorithm, the idea is to apply fuzzy negotiation to obtain a control solution that decreases the performance index and guarantees stability. To this end, each pair of agents  $\{i, j\}$  has control actions  $\{U_i^*(k), U_i^s(k), U_i^{wj}(k)\}$  and  $\{U_j^*(k), U_j^s(k), U_j^{wi}(k)\}$ . A fuzzy inference system for negotiation generates the final control actions  $U_i^f(k)$  and  $U_j^f(k)$ , considering some operational and economic constraints. Fig. 2 depicts the fuzzy negotiation scheme (Uddin & Rahman, 1999) that will be implemented to control the closed-loop system.

The main three steps of the negotiation process are:

**1) Fuzzification:** It consists of converting a numerical variable into a linguistic variable. In this way, the imprecise process knowledge determined by the membership functions transforms a crisp numerical value into fuzzy degrees of membership for each linguistic variable (Raviraj & Sen, 1997). For example, the temperature value in a boiler can be characterized by the degree of ‘high’ and ‘low’. If more linguistic labels are available, the temperature can be classified by: ‘dangerous’, ‘high’, ‘medium’, and ‘low’.



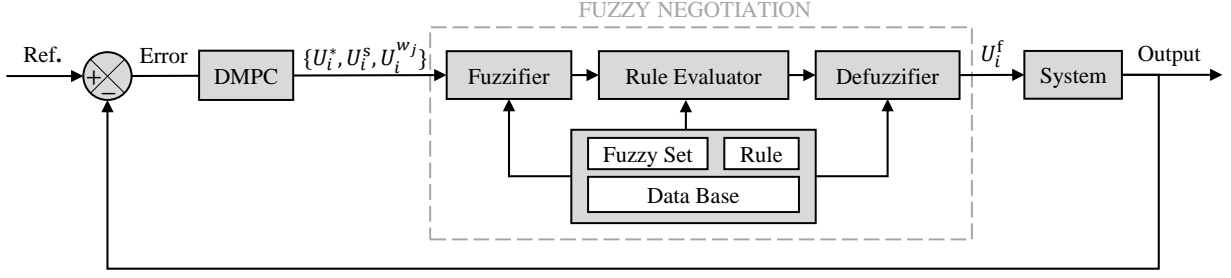


Figure 2: Scheme of the low-level control layer.

Let  $x$  be the state, input, or other algebraic variable characterizing the process (e.g., residence time in a water tank, energy consumption, and temperature). If only two linguistic labels are available for describing the variable, two fuzzy sets  $T_1$  and  $T_2$  can be considered. Their membership functions  $\mu_{T_1}(x)$  and  $\mu_{T_2}(x)$ :

$$\mu_{T_1}(x) = \begin{cases} 1 & \text{for } x < a \\ \frac{b-x}{b-a} & \text{for } a \leq x < b, \\ 0 & \text{for } b \leq x \end{cases}, \quad \mu_{T_2}(x) = \begin{cases} 0 & \text{for } x < a \\ \frac{a-x}{a-b} & \text{for } a \leq x < b, \\ 1 & \text{for } b \leq x \end{cases} \quad (8)$$

are, respectively, associated with the linguistic labels ‘high’ and ‘low’, as shown in Fig. 3.

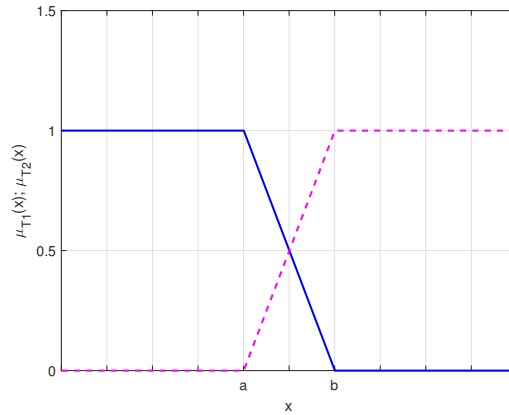


Figure 3: Membership functions (blue solid line:  $\mu_{T_1}(x)$ , and magenta dashed line:  $\mu_{T_2}(x)$ ).

The process knowledge can often allow for the definition of more than two linguistic labels. Although the methodology presented here is for three linguistic labels, its extension is straightforward. If three linguistic labels are considered, three fuzzy sets  $T_3$ ,  $T_4$ , and  $T_5$  must be defined. Let  $y$  be another state, input, or algebraic variable of the process, the

membership functions  $\mu_{T_3}(y)$ ,  $\mu_{T_4}(y)$ , and  $\mu_{T_5}(y)$ :

$$\mu_{T_3}(y) = \begin{cases} 1 & \text{for } y < a \\ \frac{c-y}{c-a} & \text{for } a \leq y < b \\ 0 & \text{for } c \leq y \end{cases}, \quad \mu_{T_4}(y) = \begin{cases} 0 & \text{for } y < a \\ \frac{y-a}{b-a} & \text{for } a \leq y < c \\ \frac{c-y}{b-c} & \text{for } c \leq y < b \\ 0 & \text{for } y > b \end{cases}, \quad (9)$$

$$\mu_{T_5}(y) = \begin{cases} 0 & \text{for } y < c \\ \frac{c-y}{c-b} & \text{for } c \leq y < b \\ 1 & \text{for } b \leq y \end{cases}$$

are, respectively, associated with the linguistic labels ‘high’, ‘medium’, and ‘low’, as displayed in Fig. 4.

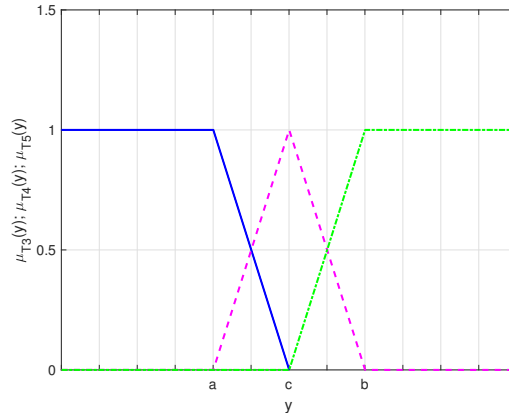


Figure 4: Membership functions (blue solid line:  $\mu_{T_3}(y)$ , magenta dashed line:  $\mu_{T_4}(y)$ , and green dashed dotted line:  $\mu_{T_5}(y)$ ).

**2) Rule evaluation:** Once the fuzzification has been performed, the next step in the fuzzy logic negotiation is the rule evaluation, which links the imprecise value of different relevant variables determined by the membership degrees with the suitability of control action. In general, the more ‘good’ or ‘safe’ the relevant variables are, the better the control action becomes. The fuzzy rules must represent all the possible combinations to provide results for all the universe of discourse. For example, in the case of a process with only two relevant variables for negotiation, a rule  $R_x$  can be written linguistically (Raviraj & Sen, 1997):

Rule  $R_x$ : If  $L_A$  is  $T_{L_A}$  and  $L_B$  is  $T_{L_B}$ , then output  $L_C$  is  $T_{L_C}$ ,

where  $T_{L_A}$ ,  $T_{L_B}$ ,  $T_{L_C}$  are, respectively, the fuzzy sets corresponding to some linguistic labels for variables  $L_A$ ,  $L_B$ ,  $L_C$ , with  $L_C$  being the control action that is evaluated in the negotiation. The use of rules with more than two antecedents or variables is straightforward, and

it depends on the particular case of study.

**3) Defuzzification:** On applying the fuzzy rules, a fuzzy characterization of the control action is generated according to each rule. In defuzzification, it is obtained a crisp number representing the global suitability of a control action taking into account all rules. As proposed in (Bai & Wang, 2006), there are several defuzzification methods, but Sugeno-type fuzzy inference (Sugeno, 1985) has been used in this work.

Each rule weighs the different antecedents regarding the considered linguistic variables:

$$w_r^x = \begin{cases} w_1 & \text{if } x \text{ is } T_1 \text{ (high)} \\ w_2 & \text{if } x \text{ is } T_2 \text{ (low)} \end{cases}, \quad w_r^y = \begin{cases} w_3 & \text{if } y \text{ is } T_3 \text{ (high)} \\ w_4 & \text{if } y \text{ is } T_4 \text{ (medium)} \\ w_5 & \text{if } y \text{ is } T_5 \text{ (low)} \end{cases}.$$

The suitability (fitness) of a control sequence  $U$  for rule  $R_r$  is defined as

$$\alpha_{R_r}(U) = w_r^x \cdot \mu_{T_{r1}}(x) \cdot w_r^y \cdot \mu_{T_{r2}}(y), \quad (10)$$

where  $r$  is the  $r$ -th rule, the variables  $x$  and  $y$  are those associated with the corresponding  $U$ , and  $\mu_{T_{r1}}(x)$ ,  $\mu_{T_{r2}}(y)$  are the fuzzy sets considered in rule  $r \in N_r$ . Hence, the total fitness of the control signal for the set of  $N_r$  rules is

$$T\alpha(U) = \sum_{r=1}^{N_r} \alpha_{R_r}(U). \quad (11)$$

The last part of the procedure consists of merging all proposed control sequences to obtain the final one. Particularly, for the proposed DMPC, the input of agent  $i$  obtained from the  $m$ -th fuzzy negotiation is

$$U_i^{f_m}(k) = \frac{U_i^s(k) \cdot T\alpha(U_i^s(k)) + U_i^*(k) \cdot T\alpha(U_i^*(k)) + U_i^{w_j}(k) \cdot T\alpha(U_i^{w_j}(k))}{T\alpha(U_i^s(k)) + T\alpha(U_i^*(k)) + T\alpha(U_i^{w_j}(k))}, \quad (12)$$

and likewise,  $U_j^{f_m}(k)$  is calculated.

Bear in mind that control sequence  $U_i^{w_j}(k)$  can be excluded from the fuzzification (12) if it leads the subsystem to unfeasibility. Moreover, the feasibility of control sequences obtained after the fuzzification process is also ensured because optimization problems (6) and (7) are convex and, hence,  $U_i^{f_m}$  is a linear combination of feasible sequences.

The  $m$ -th fuzzy-pairwise negotiation procedure is summarized as follows.

**Algorithm 1 (Stage 2):** Proposals for fuzzification.

At each time step  $k$ :

1. For each agent  $i \in \mathcal{N}$ , the triplet of possible inputs is  $\{U_i^s(k), U_i^{w_j}(k), U_i^*(k)\}$ . Since the wished control sequence  $U_i^{w_j}(k)$  is computed by neighbor  $j$  without considering state constraints of agent  $i$ , it is needed to check whether state constraint satisfaction of agent  $i$  holds after applying  $U_i^{w_j}(k)$ . Otherwise, it is excluded from the

fuzzification process. Afterward, fuzzy negotiation is applied to compute the final sequence  $U_i^{f_m}(k)$ . Similarly, it is computed  $U_j^{f_m}(k)$ .

2. A resulting pairwise fuzzy negotiation sequence  $U_{ij}^{f_m}(k) = \{U_i^{f_m}(k), U_j^{f_m}(k), U_l^s(k)\}$  is defined based on  $U_i^{f_m}(k)$  and  $U_j^{f_m}(k)$ , assuming that the rest of subsystems  $l \in \mathcal{N} \setminus \{i, j\}$  follows their pre-defined trajectories.
3. Agent  $i$  sends its cost for the fuzzy and stabilizing control inputs to its neighbors, and vice versa. Let us define  $U_{ij}^s = \{U_i^s(k), U_j^s(k), U_l^s(k)\}$ ; if the condition

$$\sum_{l \in \mathcal{M}_i \cup \mathcal{M}_j \cup \{i, j\}} J_l(x_l(k), U_{ij}^{f_m}) \leq \sum_{l \in \mathcal{M}_i \cup \mathcal{M}_j \cup \{i, j\}} J_l(x_l(k), U_{ij}^s) \quad (13)$$

holds, then stability is guaranteed, and thus,  $U_i^{f_m}(k)$  is sent to the upper control layer. Otherwise,  $U_i^s(k)$  is sent.

Note that there can be up to  $N^2 - N$  negotiation problems running in parallel at each time instant  $k$  if all subsystems were coupled with their neighbors. Nevertheless, as shown in Fig. 5, only high-coupled subsystems with enabled communication links (dashed line) follow the cooperative scheme proposed.

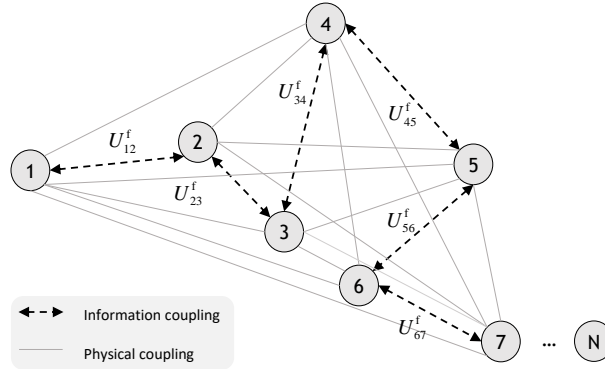


Figure 5: Example of  $N$  agents negotiating in pairs.

### 3.2. Overall low-level control layer scheme

The minimum number of communication steps for this cooperative scheme is five. An outline of this scheme is shown in Fig. 6. First of all, agent  $i \in \mathcal{N}$  sends its shifted input trajectory  $U_i^s(k)$  to agents in  $\mathcal{M}_i$ , and similarly, agent  $j$  sends its trajectory. Once the information is received, agents  $i$  and  $j$  compute, respectively, the optimal sequences  $U_i^*(k), U_j^*(k)$  and the wished neighboring input trajectories  $U_j^{w_i}(k), U_i^{w_j}(k)$ , and exchange them. Before applying fuzzification, it is checked the feasibility of  $U_i^{w_j}(k)$  and  $U_j^{w_i}(k)$ . Unless feasible, these input sequences will not be included in the set of sequences to fuzzification. Then, agents communicate their results to each other. In the fourth step, sequences  $U_i^{f_m}(k), U_j^{f_m}(k)$  resulting from the  $m$ -th fuzzy negotiation are respectively exchanged. Afterward, it is checked whether stability condition (13) is satisfied. Otherwise, the final input

trajectories of agents  $i$  and  $j$  will be  $U_i^{fm}(k) = U_i^s(k)$  and  $U_j^{fm}(k) = U_j^s(k)$ , respectively. Finally, costs associated with these inputs are exchanged in the last step.

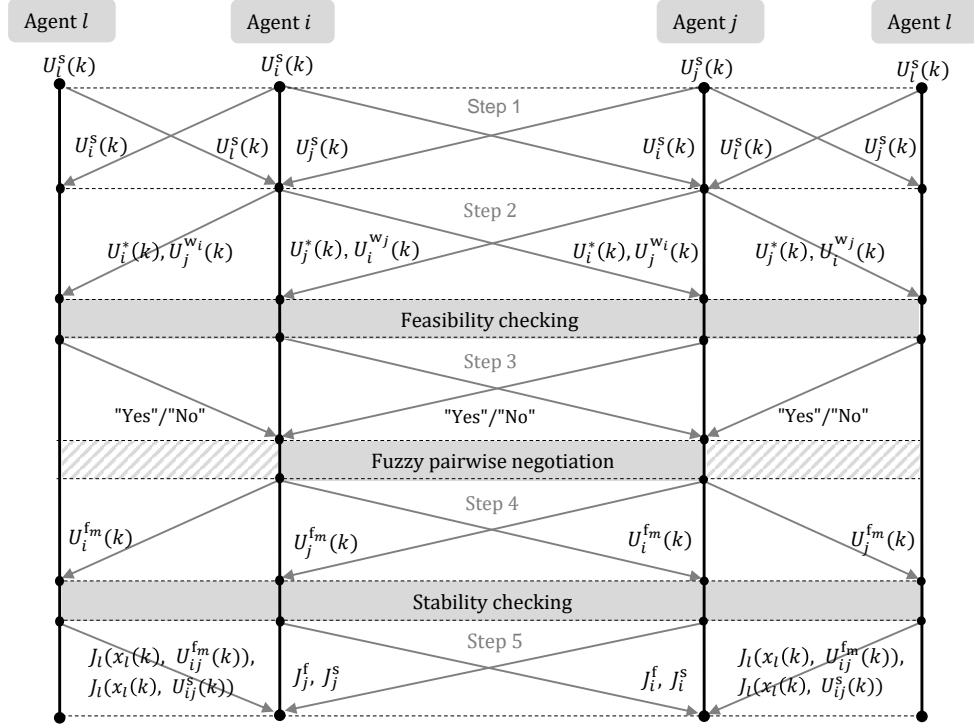


Figure 6: Outline of the communication steps involved in the  $m$ -th pairwise negotiation for the proposed lower control layer.

Once the sequences resulting from the pairwise negotiations are obtained, they are sent to the upper control layer.

### 3.3. Upper-level control layer

Given agent  $i$  coupled through inputs with its neighbors and the set of control sequences  $\{U_i^{f1}(k), U_i^{f2}(k), \dots, U_i^{fM}(k)\}$  obtained from  $M$  pairwise fuzzy negotiations, this supervisory layer is responsible for applying a final fuzzification process with all these options.

The final control sequence  $U_i^f(k)$  is computed as a weighted average of the control sequences from the  $m \in \{1, \dots, M\}$  pairwise fuzzy negotiations with all of its neighbors:

$$U_i^f(k) = \frac{\sum_{m=1}^M U_i^{fm}(k) \cdot T\alpha(U_i^{fm}(k))}{\sum_{m=1}^M T\alpha(U_i^{fm}(k))}, \quad (14)$$

where  $T\alpha(\cdot)$  is the total fitness of the control signal for the set of rules.

For the overall system, the input sequence  $U_{\mathcal{N}}^f(k)$  can be obtained by aggregating all the final sequences, i.e.,  $U_{\mathcal{N}}^f(k) = (U_i^f(k))_{i \in \mathcal{N}}$ . Hence, the global cost function is

$$J_{\mathcal{N}}^f(x_{\mathcal{N}}(k), U_{\mathcal{N}}^f(k)) = \sum_{t=0}^{N_p-1} L_{\mathcal{N}}(x_{\mathcal{N}}(k+t), u_{\mathcal{N}}(k+t)) + F_{\mathcal{N}}(x_{\mathcal{N}}(k+N_p)), \quad (15)$$

where  $L_{\mathcal{N}}(\cdot)$  and  $F_{\mathcal{N}}(\cdot)$  are, respectively, the global stage and terminal cost functions defined as

$$\begin{aligned} L_{\mathcal{N}}(\cdot) &= x_{\mathcal{N}}(k+t)^{\top} Q_{\mathcal{N}} x_{\mathcal{N}}(k+t) + u_{\mathcal{N}}(k+t)^{\top} R_{\mathcal{N}} u_{\mathcal{N}}(k+t), \\ F_{\mathcal{N}}(\cdot) &= x_{\mathcal{N}}(k+N_p)^{\top} P_{\mathcal{N}} x_{\mathcal{N}}(k+N_p), \end{aligned} \quad (16)$$

with the weighting matrices  $Q_{\mathcal{N}} = [Q_i]_{i \in \mathcal{N}}$  and  $R_{\mathcal{N}} = [R_i]_{i \in \mathcal{N}}$ ; and the terminal cost matrix  $P_{\mathcal{N}} = [P_i]_{i \in \mathcal{N}}$ . The overall cost-to-go must decrease over time to guarantee stability, i.e.,

$$J_{\mathcal{N}}^f(x_{\mathcal{N}}(k+1), U_{\mathcal{N}}^f(k+1)) \leq J_{\mathcal{N}}^f(x_{\mathcal{N}}(k), U_{\mathcal{N}}^f(k)). \quad (17)$$

Otherwise, the final control sequence  $U_{\mathcal{N}}^f(k+1) = U_{\mathcal{N}}^s(k+1)$  will be implemented because it decreases the overall cost, as proved later in subsection 4.3.

The steps taken by the upper control layer are summarized in Algorithm 2.

**Algorithm 2:** Supervisory layer

1. From  $M$  pairwise negotiations, the upper control layer receives the set of sequences  $\{U_i^{f_1}(k), U_i^{f_2}(k), \dots, U_i^{f_M}(k)\}$  for each agent  $i$ .
2. Final sequence  $U_i^f$  is calculated as (14) for each  $i \in \mathcal{N}$ , and, afterward, they are aggregated to obtain a global one for the overall system, i.e.,  $U_{\mathcal{N}}^f(k) = (U_i^f(k))_{i \in \mathcal{N}}$ .
3. Finally, the overall cost  $J_{\mathcal{N}}^f(x_{\mathcal{N}}(k+1), U_{\mathcal{N}}^f(k+1))$  is computed and compared with the cost from the previous time instant  $k$ . If the cost decreases,  $U_{\mathcal{N}}^f(k+1)$  is applied to the system. Otherwise,  $U_{\mathcal{N}}^s(k+1)$  is finally implemented.

Note that this layer acts as a supervisory layer, which merges all pairwise negotiations. However, this fact is not contradictory with the proposed architecture because it is only necessary full communication in these steps, relieving communication burden the rest of the time. Additionally, this centralized supervisor is performed with lower computation time than a centralized MPC, which needs to solve an overall optimization problem subject to constraints. See (Marcos et al., 2014; Saad et al., 2018; Velarde et al., 2019) as examples of distributed MPC approaches where a supervisor communicates with local controllers.

#### 4. Stability and Controller Design Procedure

The outcome of the pairwise fuzzy negotiation process is a feasible sequence of inputs for the overall system. Furthermore, the upper layer uses a convex combination of these sequences, which are also feasible due to the problem's convexity. This strategy is supported by auxiliary feedback, which provides the basis for feasibility and stability guarantees of the proposed scheme. In this section, the requirements to hold recursive feasibility and stability are detailed. Additionally, a procedure to design the necessary feedback controllers based on a Linear Matrix Inequality (LMI) is described.

#### 4.1. Stability requirements

A standard approach based on terminal regions/invariant sets  $\Omega_i$  is followed to obtain stability of the closed-loop system. Note that, from each agent's viewpoint, the coupling with its neighbors can be considered as an unknown bounded disturbance set  $\mathcal{W}_i$  when computing its invariant set  $\Omega_i$  to simplify the problem (Maestre et al., 2011a,b).

**Assumption 1.** For each system (1), there is a feedback  $K_i$  that ensures that all eigenvalues of  $(A_i + B_{ii}K_i)$  are within the unit circle. Likewise, the same holds for the corresponding global feedback  $K_{\mathcal{N}} = \text{diag}(K_i)_{i \in \mathcal{N}}$ .

**Definition 1.** The set  $\Omega_i$  is a Robust Positively Invariant (RPI) set for subsystem  $i$  if and only if its evolution satisfies:

$$x_i(k) \in \Omega_i \rightarrow x_i(k+1) \in \Omega_i, \quad \forall w_i(k) \in \mathcal{W}_i, \quad \forall k \in \mathbb{N}_{0+}. \quad (18)$$

**Assumption 2.** There exists an RPI set  $\Omega_i$  that, under the linear control law  $u_i = K_i x_i$ , satisfies

$$(A_i + B_{ii}K_i)\Omega_i \oplus \mathcal{W}_i \subseteq \Omega_i, \quad \Omega_i \subseteq \mathcal{X}_i, \quad K_i\Omega_i \subseteq \mathcal{U}_i, \quad (19)$$

with  $\mathcal{W}_i$  being a convex set that contains the origin in its interior.

Considering Assumption 2, the RPI set of the overall system can be computed as the Cartesian product of all  $\Omega_i$ , i.e.,

$$\Omega_{\mathcal{N}} = \times_{i \in \mathcal{N}} \Omega_i. \quad (20)$$

There are several methods to find a set  $\Omega_i$  that satisfies these constraints (see, e.g., (Kolmanovsky & Gilbert, 1998; Rakovic et al., 2005) for a procedure to find the maximal and the minimal robust positive invariant, respectively). In this work, the Multi-Parametric Toolbox (MPT) of MATLAB (Herceg et al., 2013) is employed to compute each subsystem's maximal RPI set.

**Assumption 3.** There exists a Lyapunov function  $V_{\mathcal{N}}(x_{\mathcal{N}}(k)) = x_{\mathcal{N}}(k)^{\top} P_{\mathcal{N}} x_{\mathcal{N}}(k)$  with  $P_{\mathcal{N}} = \text{diag}(P_i)_{i \in \mathcal{N}}$  controlled by  $K_{\mathcal{N}} = \text{diag}(K_i)_{i \in \mathcal{N}}$  that provides an upper bound on the cost-to-go of the system, i.e.,

$$V_{\mathcal{N}}(x_{\mathcal{N}}(k)) \geq \sum_{t=0}^{\infty} L_{\mathcal{N}}(x_{\mathcal{N}}(k+t), u_{\mathcal{N}}(k+t)). \quad (21)$$

#### 4.2. Controller design

Numerous methods to design controllers such as Internal Model Control (IMC) (Wang et al., 2001; Tan et al., 2003), Gain Scheduling (GS) (Leith & Leithead, 2000), and loop-shaping (McFarlane & Glover, 1992; Zhu et al., 2003) can be found in the literature. Here, matrices  $K_{\mathcal{N}}$  and  $P_{\mathcal{N}}$  for the global system are obtained by solving the following LMI such as in (Magni et al., 2003; Lazar et al., 2009; Darivianakis et al., 2019).

**Theorem 1.** Let the global matrices be  $A_{\mathcal{N}} = [A_{ij}]_{i,j \in \mathcal{N}}$  and  $B_{\mathcal{N}} = [B_{ij}]_{i,j \in \mathcal{N}}$ , and stage cost matrices  $Q_{\mathcal{N}} = \text{diag}(Q_i)_{i \in \mathcal{N}}$  and  $R_{\mathcal{N}} = \text{diag}(R_i)_{i \in \mathcal{N}}$ . If there are matrices  $H_{\mathcal{N}} = H_{\mathcal{N}}^{\top} = \text{diag}(H_i)_{i \in \mathcal{N}}$ , where  $H_i \in \mathbb{R}^{q_i \times q_i}$ , and  $Y_{\mathcal{N}} = \text{diag}(Y_i)_{i \in \mathcal{N}}$ , where  $Y_i \in \mathbb{R}^{r_i \times q_i}$  in such a way that it is held:

$$\begin{bmatrix} H_{\mathcal{N}} & H_{\mathcal{N}}A_{\mathcal{N}}^{\top} + Y_{\mathcal{N}}^{\top}B_{\mathcal{N}}^{\top} & H_{\mathcal{N}}Q_{\mathcal{N}}^{1/2} & Y_{\mathcal{N}}^{\top}R_{\mathcal{N}}^{1/2} \\ A_{\mathcal{N}}H_{\mathcal{N}} + B_{\mathcal{N}}Y_{\mathcal{N}} & H_{\mathcal{N}} & 0 & 0 \\ Q_{\mathcal{N}}^{1/2}H_{\mathcal{N}} & 0 & I & 0 \\ R_{\mathcal{N}}^{1/2}Y_{\mathcal{N}} & 0 & 0 & I \end{bmatrix} \geq 0, \quad (22)$$

then there exists a matrix  $P_{\mathcal{N}} = H_{\mathcal{N}}^{-1}$  that satisfies (21), and a feedback control matrix  $K_{\mathcal{N}} = Y_{\mathcal{N}}H_{\mathcal{N}}^{-1}$  that stabilizes the closed-loop system.

*Proof.* Following Maestre et al. (2011b), the condition (22) is derived from the recursive application of the Schur's complement to

$$(A_{\mathcal{N}} + B_{\mathcal{N}}K_{\mathcal{N}})^{\top}P_{\mathcal{N}}(A_{\mathcal{N}} + B_{\mathcal{N}}K_{\mathcal{N}}) - P_{\mathcal{N}} + Q_{\mathcal{N}} + K_{\mathcal{N}}^{\top}R_{\mathcal{N}}K_{\mathcal{N}} \leq 0. \quad (23)$$

Secondly, it is proved that the closed-loop system is stable with the linear control law  $u_{\mathcal{N}} = K_{\mathcal{N}}x_{\mathcal{N}}$ . Pre- and post-multiplying (23) by  $x_{\mathcal{N}}(k)^{\top}$  and  $x_{\mathcal{N}}(k)$ , and then by  $-1$ , it can be rewritten as

$$x_{\mathcal{N}}(k)^{\top}P_{\mathcal{N}}x_{\mathcal{N}}(k) - x_{\mathcal{N}}(k+1)^{\top}P_{\mathcal{N}}x_{\mathcal{N}}(k+1) \geq L_{\mathcal{N}}(x_{\mathcal{N}}(k)), \quad (24)$$

where

$$\begin{aligned} x_{\mathcal{N}}(k+1) &= (A_{\mathcal{N}} + B_{\mathcal{N}}K_{\mathcal{N}})x_{\mathcal{N}}(k), \\ l_{\mathcal{N}}(x_{\mathcal{N}}(k)) &= x_{\mathcal{N}}(k)^{\top}Q_{\mathcal{N}}x_{\mathcal{N}}(k) + x_{\mathcal{N}}(k)^{\top}K_{\mathcal{N}}^{\top}R_{\mathcal{N}}K_{\mathcal{N}}x_{\mathcal{N}}(k). \end{aligned}$$

Hence,  $V_{\mathcal{N}}(x_{\mathcal{N}}(k)) = x_{\mathcal{N}}(k)^{\top}P_{\mathcal{N}}x_{\mathcal{N}}(k)$  is a Lyapunov function and stability is guaranteed. Moreover, it can be checked that (21) holds by telescope summation.  $\square$

The LMI (22) is solved maximizing the trace of  $H_{\mathcal{N}}$  to design the controller for the overall system. This fact leads to the minimization of the trace of  $P_{\mathcal{N}} = H_{\mathcal{N}}^{-1}$ , and, therefore, minimizing the cost-to-go. The resulting feedback matrix  $K_{\mathcal{N}} = Y_{\mathcal{N}}H_{\mathcal{N}}^{-1}$  is the gain of the controller.

### 4.3. Stability of the MPC controller

In the previous section, the system's stability with the corresponding  $K_{\mathcal{N}}$  has been proved. Here, it is also proved stability guarantees when the MPC controller is used.

**Assumption 4.** There is a feasible overall control sequence at time instant  $k = 0$ .

**Theorem 2.** Given matrices  $K_{\mathcal{N}}, P_{\mathcal{N}}$  computed by maximizing the trace of  $H_{\mathcal{N}}$  subject to LMI (22), and let Assumption 4 hold, the MPC controller can ensure recursive feasibility and closed-loop system stability.



*Proof.* Given a feasible control sequence at time step  $k$ :

$$U_{\mathcal{N}}(k) = [u_{\mathcal{N}}(k), u_{\mathcal{N}}(k+1), \dots, u_{\mathcal{N}}(k+N_p-1)], \quad (25)$$

the constraints  $x_{\mathcal{N}}(k) \in \mathcal{X}_{\mathcal{N}}$ ,  $u_{\mathcal{N}}(k) \in \mathcal{U}_{\mathcal{N}}$ , and  $x_{\mathcal{N}}(k+N_p) \in \Omega_{\mathcal{N}}$  are satisfied. A feasible shifted sequence can then be formed with the tail extension of (25) at time instant  $k+1$ :

$$\begin{aligned} U_{\mathcal{N}}^s(k+1) &= [u_{\mathcal{N}}^s(k+1), u_{\mathcal{N}}^s(k+2), \dots, u_{\mathcal{N}}^s(k+N_p)] \\ &= [u_{\mathcal{N}}(k+1), u_{\mathcal{N}}(k+2), \dots, Kx_{\mathcal{N}}(k+N_p)]. \end{aligned} \quad (26)$$

Since  $x_{\mathcal{N}}(k+N_p) \in \Omega_{\mathcal{N}}$ , the following equation holds at  $k+1$ :

$$(A_{\mathcal{N}} + B_{\mathcal{N}}K_{\mathcal{N}})x_{\mathcal{N}}(k+N_p) \in \Omega_{\mathcal{N}}.$$

Hence, under Assumption 4, and applying this procedure recursively, feasibility is held for all  $k$ .

Furthermore, it is proved that the global cost decreases in time when using  $U_{\mathcal{N}}^s$  as a backup control sequence. To this end, the overall cost decreases from time step  $k$  to  $k+1$  if it holds that

$$J_{\mathcal{N}}(x_{\mathcal{N}}(k+1), U_{\mathcal{N}}^s(k+1)) \leq J_{\mathcal{N}}(x_{\mathcal{N}}(k), U_{\mathcal{N}}(k)). \quad (27)$$

Substituting (15) in (27), and considering definitions (25) and (26), we get

$$\begin{aligned} \sum_{t=1}^{N_p} L_{\mathcal{N}}(x_{\mathcal{N}}(k+t), u_{\mathcal{N}}^s(k+t)) + F_{\mathcal{N}}(x_{\mathcal{N}}(k+N_p+1)) &\leq \\ \sum_{t=0}^{N_p-1} L_{\mathcal{N}}(x_{\mathcal{N}}(k+t), u_{\mathcal{N}}(k+t)) + F_{\mathcal{N}}(x_{\mathcal{N}}(k+N_p)). & \end{aligned} \quad (28)$$

Removing terms in common and rearranging, we have

$$\begin{aligned} L_{\mathcal{N}}(x_{\mathcal{N}}(k+N_p), u_{\mathcal{N}}^s(k+N_p)) + F_{\mathcal{N}}(x_{\mathcal{N}}(k+N_p+1)) - F_{\mathcal{N}}(x_{\mathcal{N}}(k+N_p)) \\ \leq L_{\mathcal{N}}(x_{\mathcal{N}}(k), u_{\mathcal{N}}(k)) \end{aligned} \quad (29)$$

Note that the left side hand of (29) is lower than or equal to zero because it is imposed by design (24), and the right side hand is greater than or equal to zero since the stage cost is positive definite. Thus it is proved that (27) is satisfied using the shifted control sequence, and, consequently, the global cost decreases in time. Finally, note that the control algorithm includes condition (17) to ensure stability. If it is not satisfied with the *fuzzified* control action obtained in the negotiation, the shifted control signal is implemented, which has been proved to decrease the overall cost. Hence, the feedback controller can be seen as a backup control strategy to provide the proposed scheme with feasibility and stability guarantees.  $\square$

In a nutshell, stability holds in a straight forward manner when  $U_{\mathcal{N}}^s$  is applied, which is formed using a linear feedback conveniently designed using an LMI-based procedure. At each time instant, a convex combination of feasible sequences is also formed by using

fuzzy inference and then its performance is compared with that of the stabilizing sequence  $U_{\mathcal{N}}^s$ . If the fuzzy sequence  $U_{\mathcal{N}}^f$  results in a decrease of the cost of the system, then it is adopted and implemented; otherwise  $U_{\mathcal{N}}^s$  is kept and applied. In either case, there is a feasible sequence available at the next time step that can be shifted to form the new  $U_{\mathcal{N}}^s$  guaranteeing both feasibility and stability. Hence, the feedback provides us with a back-up strategy that guarantees stability while the fuzzy proposal created at each time step can introduce improvements in the control sequence that optimize the system evolution according to the fuzzy tuning parameters

## 5. Case Study

This section describes the eight-coupled tanks plant based on the quadruple tank process (Johansson, 2000) and specifies the non-linear model and the parameters used in the simulations. Furthermore, the fuzzy negotiation criteria are detailed, bearing in mind the goal and constraints of the problem.

### 5.1. Plant description

The eight-coupled tanks plant is composed of eight interconnected tanks as depicted in Fig. 7. There are four upper tanks (3, 4, 7, and 8) that discharge flows into the lower ones (1, 2, 5, and 6), and these, in turn, into sinking tanks. Four pumps control the plant, and there are also six three-way valves  $\gamma_v$ , with  $v \in \{1, 2, \dots, 6\}$  manually operated that divide the input flows into two ways.

The system is divided into  $N = 4$  subsystems: tanks 1 and 3 belong to subsystem 1; tanks 2 and 4 form subsystem 2; tanks 5 and 7 take part in subsystem 3; and the remaining tanks form subsystem 4. The control goal is to reach some target levels considering the operational cost and satisfying operational constraints. Hence, it is a multi-variable control problem with four outputs ( $h_1, h_2, h_5$ , and  $h_6$ ) and four inputs ( $q_a, q_b, q_c$ , and  $q_d$ ).

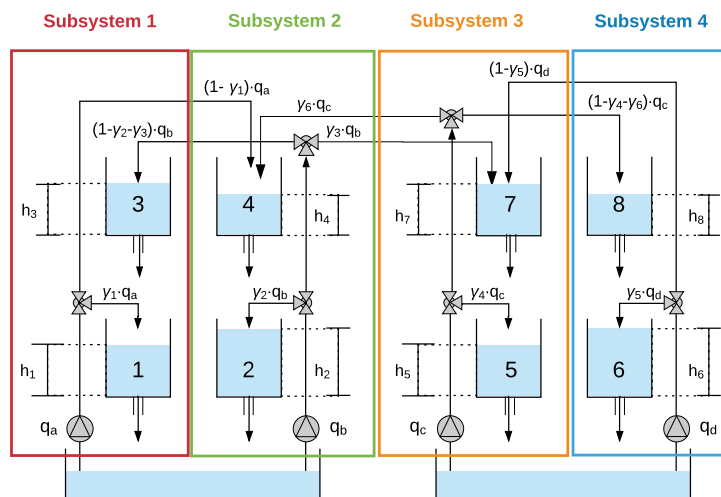


Figure 7: Schematic diagram of the eight-coupled tanks plant with the proposed subsystems.

## 5.2. Plant and control model

The non-linear model of the plant can be obtained by applying mass balances and Bernoulli's law. The following differential equations describe the system:

$$\begin{aligned}
S_1 \frac{dh_1}{dt} &= a_3 \sqrt{2gh_3} - a_1 \sqrt{2gh_1} + \gamma_1 \frac{q_a}{3600}, \\
S_2 \frac{dh_2}{dt} &= a_4 \sqrt{2gh_4} - a_2 \sqrt{2gh_2} + \gamma_2 \frac{q_b}{3600}, \\
S_3 \frac{dh_3}{dt} &= -a_3 \sqrt{2gh_3} + (1 - \gamma_2 - \gamma_3) \frac{q_b}{3600}, \\
S_4 \frac{dh_4}{dt} &= -a_4 \sqrt{2gh_4} + (1 - \gamma_1) \frac{q_a}{3600} + \gamma_6 \frac{q_c}{3600}, \\
S_5 \frac{dh_5}{dt} &= a_7 \sqrt{2gh_7} - a_5 \sqrt{2gh_5} + \gamma_4 \frac{q_c}{3600}, \\
S_6 \frac{dh_6}{dt} &= a_8 \sqrt{2gh_8} - a_6 \sqrt{2gh_6} + \gamma_5 \frac{q_d}{3600}, \\
S_7 \frac{dh_7}{dt} &= -a_7 \sqrt{2gh_7} + (1 - \gamma_5) \frac{q_d}{3600} + \gamma_3 \frac{q_b}{3600}, \\
S_8 \frac{dh_8}{dt} &= -a_8 \sqrt{2gh_8} + (1 - \gamma_4 - \gamma_6) \frac{q_c}{3600},
\end{aligned} \tag{30}$$

where  $h_n$  is the water level of tank  $n \in \{1, 2, \dots, 8\}$ ,  $S_n = 13.89 \cdot 10^{-3} \text{ m}^2$  is the corresponding cross-section (equal for all tanks), and  $a_n = 50.265 \cdot 10^{-6} \text{ m}^2$  is the cross-section of all outlet pipes. The parameter  $\gamma_v \in [0, 1]$  with  $v \in \{1, 2, \dots, 6\}$  refers to the opening of the six three-way valves ( $\gamma_1, \gamma_4=0.3$ ,  $\gamma_2, \gamma_5=0.4$ , and  $\gamma_3, \gamma_6=0.1$ );  $g = 9.8 \text{ m/s}^2$  is the gravity constant; and  $q_m$  corresponds to the flow given by pump  $m \in \{a, b, c, d\}$ .

Let us define the operating point of tank level  $h_n^0$  for  $n \in \{1, 2, \dots, 8\}$  as  $h_1^0 = 0.10$ ,  $h_2^0 = 0.15$ ,  $h_3^0 = 0.07$ ,  $h_4^0 = 0.03$ ,  $h_5^0 = 0.10$ ,  $h_6^0 = 0.15$ ,  $h_7^0 = 0.025$ , and  $h_8^0 = 0.10$  (units: meters), and the operating point of pumps' flow rate  $q_m^0$  for  $m \in \{a, b, c, d\}$  as  $q_a^0 = 0.142$ ,  $q_b^0 = 0.421$ ,  $q_c^0 = 0.424$ , and  $q_d^0 = 0.140$  (units:  $\text{m}^3/\text{h}$ ). The linear discrete-time state-space model is

$$\bar{x}_{\mathcal{N}}(k+1) = A_{\mathcal{N}} \bar{x}_{\mathcal{N}}(k) + B_{\mathcal{N}} \bar{u}_{\mathcal{N}}(k), \tag{31}$$

where the state vector becomes  $\bar{x}_{\mathcal{N}} = [h_1(k) - h_1^0, \dots, h_8(k) - h_8^0]^{\top}$ ; the input vector  $\bar{u}_{\mathcal{N}} = [q_a(k) - q_a^0, \dots, q_d(k) - q_d^0]^{\top}$ ; and  $A_{\mathcal{N}}$ ,  $B_{\mathcal{N}}$  are the corresponding matrices of the global system. Similarly, the dynamics of each subsystem  $i \in \{1, 2, 3, 4\}$  are

$$\begin{aligned}
\bar{x}_i(k+1) &= A_i \bar{x}_i(k) + B_{ii} \bar{u}_i(k) + \bar{w}_i(k), \\
\bar{w}_i(k) &= \sum_{j \in \mathcal{N}_i} B_{ij} \bar{u}_j(k),
\end{aligned} \tag{32}$$

where  $\bar{w}_i(k)$  represents the coupling with its neighbours  $j \in \mathcal{N}_i$ , and the corresponding matrices of subsystems are:

$$\begin{aligned}
A_1 &= \begin{bmatrix} 0.8810 & 0.1325 \\ 0 & 0.8587 \end{bmatrix}, B_{11} = \begin{bmatrix} 0.0282 \\ 0 \end{bmatrix}, B_{12} = \begin{bmatrix} 0.0035 \\ 0.0464 \end{bmatrix}, \\
A_2 &= \begin{bmatrix} 0.9017 & 0.1922 \\ 0 & 0.7973 \end{bmatrix}, B_{21} = \begin{bmatrix} 0.0071 \\ 0.0626 \end{bmatrix}, B_{22} = \begin{bmatrix} 0.0380 \\ 0 \end{bmatrix}, B_{23} = \begin{bmatrix} 0.0010 \\ 0.0089 \end{bmatrix}, \\
A_3 &= \begin{bmatrix} 0.8810 & 0.2104 \\ 0 & 0.7753 \end{bmatrix}, B_{32} = \begin{bmatrix} 0.0011 \\ 0.0088 \end{bmatrix}, B_{33} = \begin{bmatrix} 0.0282 \\ 0 \end{bmatrix}, B_{34} = \begin{bmatrix} 0.0067 \\ 0.0530 \end{bmatrix}, \\
A_4 &= \begin{bmatrix} 0.9017 & 0.1126 \\ 0 & 0.8814 \end{bmatrix}, B_{43} = \begin{bmatrix} 0.0035 \\ 0.0564 \end{bmatrix}, B_{44} = \begin{bmatrix} 0.0380 \\ 0 \end{bmatrix}, \\
A_{ij} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad i \neq j \quad \forall i, j \in \mathcal{N}.
\end{aligned}$$

Moreover, states and inputs are constrained by

$$\begin{aligned}
-h_n^0 < \bar{x}_n(k) \leq 0.08, \quad -q_m^0 < \bar{u}_m(k) \leq 0.4, \\
\forall n \in \{1, 2, \dots, 8\}, \quad \forall m \in \{a, b, c, d\}.
\end{aligned}$$

Matrices  $Q_{\mathcal{N}} = \text{diag}(Q_i)_{i \in \mathcal{N}}$  and  $R_{\mathcal{N}} = \text{diag}(R_i)_{i \in \mathcal{N}}$  are, respectively, the corresponding constant weighting matrices with:

$$\begin{aligned}
Q_i &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \forall i \in \{1, 2, 3, 4\}, \\
R_1 &= \begin{bmatrix} 0.002 & 0 \\ 0 & 0.2 \end{bmatrix}, {}^1 R_2, R_3 = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}, R_4 = \begin{bmatrix} 0.02 & 0 \\ 0 & 0.2 \end{bmatrix}.
\end{aligned}$$

Furthermore, local feedback gain  $K_i$ , and weighting matrix of the terminal cost  $P_i$ , which

---

<sup>1</sup>The weighting parameter  $R_{11} = 0.002$  is intentionally different to have one controller out of tune and assess its impact on the proposed scheme.

have been calculated maximizing the trace of  $H_{\mathcal{N}}$  with Theorem 1, are

$$\begin{aligned}
K_1^\top &= \begin{bmatrix} -0.0567 \\ 0.0913 \end{bmatrix}, & P_1 &= \begin{bmatrix} 4.6685 & 2.3288 \\ 2.3288 & 2.4292 \end{bmatrix}, \\
K_2^\top &= \begin{bmatrix} -0.2863 \\ -0.2331 \end{bmatrix}, & P_2 &= \begin{bmatrix} 5.1779 & 3.0510 \\ 3.0510 & 3.0250 \end{bmatrix}, \\
K_3^\top &= \begin{bmatrix} -0.0575 \\ -0.0922 \end{bmatrix}, & P_3 &= \begin{bmatrix} 4.7185 & 2.8848 \\ 2.8848 & 3.0013 \end{bmatrix}, \\
K_4^\top &= \begin{bmatrix} -0.2319 \\ -0.1585 \end{bmatrix}, & P_4 &= \begin{bmatrix} 5.2133 & 2.4217 \\ 2.4217 & 2.3883 \end{bmatrix}.
\end{aligned}$$

A standard approach has been implemented with MPT of MATLAB (Herceg et al., 2013) to compute each agent's RPI set. The corresponding invariant sets  $\Omega_i$  of all subsystems are depicted in Fig. 8.

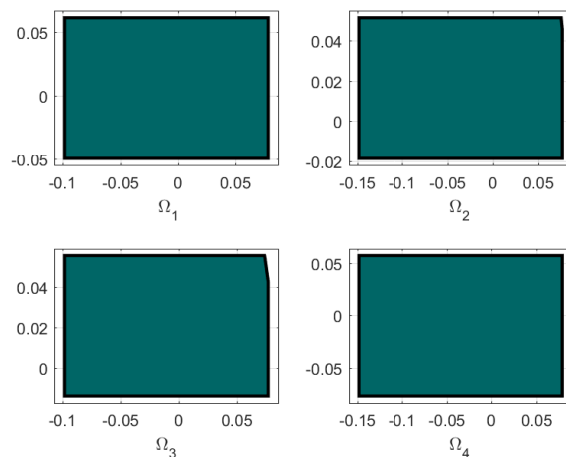


Figure 8: Invariant sets of the four subsystems.

### 5.3. Fuzzy negotiation specifications

This section details the fuzzy negotiation criteria in the eight-coupled tanks system, taking into account the process constraints and operational costs.

- 1) **Fuzzification:** Three performance indicators have been considered for the fuzzy negotiation procedure: the residence time for water stored in the lower tanks, the current water level, and the pumping energy. They represent typical constraints for the plant, and, in this case, act as fuzzy criteria similarly to soft constraints.

- i. Residence Time ( $RT$ ) is the time that water spends at each tank. At each time instant  $k$ , the  $RT$  for tank  $n \in \{1, 2, 5, 6\}$  is evaluated with level  $h_n$  at the end of the prediction horizon  $N_p$  as

$$RT_n(k) = \frac{S_n}{a_n \sqrt{2gh_n(k + N_p)}}. \quad (33)$$

This criterion is particularly interesting if some chemical reactions take place in the tanks. Thus, a minimum residence time has to be imposed at each tank to fulfill the requirements for reactions. To this end, two fuzzy sets representing constraint satisfaction and violation are defined by the limits presented in Table 1.

- ii. Current water level ( $h$ ) is essential to prevent overflow or excess of product storage. Although a hard constraint is also added to the DMPC to avoid spilling, the water height  $h_n$  is considered in the fuzzy negotiation to improve process safety. Due to the plant setup and the distributed control framework, only lower-tank levels ( $h_1, h_2, h_5$ , and  $h_6$ ) are evaluated because the upper ones are only influenced by their neighbors' control actions. Three linguistic variables ('low', 'medium', and 'high') are considered for water levels.
- iii. Pumping Energy ( $PE$ ) is defined for each pump  $m \in \{a, b, c, d\}$  as the average  $PE$  over the prediction horizon, and it is assumed to be proportional to the water flows provided by pumps, i.e.,

$$PE_m(k) = \frac{0.04}{3600N_p} \sum_{t=1}^{N_p} q_m(k + t). \quad (34)$$

Note that the definition includes absolute flow values instead of flow increments. Finally, the total pumping energy is computed as

$$PE(k) = PE_a(k) + PE_b(k) + PE_c(k) + PE_d(k), \quad (35)$$

and the fuzzy sets for the three criteria are determined by the values of Table 1, representing the limits and vertices of triangles that define the membership functions.

Table 1: Summary of fuzzy sets limits for the three performance indicators.

<b>Performance indicators</b>			
<b>Limits</b>	$PE_m$ (kWh)	$h_n$ (m)	$RT_n$ (s)
<b>a</b>	$1 \cdot 10^{-5}/N_p$	0.15	26
<b>b</b>	$5 \cdot 10^{-5}/N_p$	0.20	32
<b>c</b>	$10 \cdot 10^{-5}/N_p$	0.25	—

- 2) **Rule evaluation:** A set of fuzzy rules has been defined according to the process. In this plant, only three negotiation problems between subsystems  $\{1, 2\}$ ,  $\{2, 3\}$ , and  $\{3, 4\}$  are performed in the lower layer due to their physical distribution. A generic rule to assign the fitness for control signal  $q_m$  can be expressed as

$$R_{ri}: \text{ If } RT_n(k + N_p) \text{ is } Tr_1, PE_m(k) \text{ is } Tr_2, \text{ and } h_n(k + N_p) \text{ is } Tr_3, \text{ then output } q_m \text{ is } Tr_4,$$

where  $R_{ri}$  represents the  $r$ -th rule of subsystem  $i \in \{1, 2, 3, 4\}$ ,  $Tr_1 = \{\text{'high'}, \text{'low'}\}$ ,  $Tr_2 = \{\text{'high'}, \text{'medium'}, \text{'low'}\}$ ,  $Tr_3 = \{\text{'high'}, \text{'medium'}, \text{'low'}\}$ , and  $Tr_4 = \{\text{'good'}, \text{'acceptable'}, \text{'bad'}\}$  are the linguistic variables that characterize the criteria  $RT_n$ ,  $PE_m$ ,  $h_n$ , and  $q_m$ , respectively. Here are some examples:

$$R_{1i}: \text{ If } (RT_n \text{ is 'high'}), (PE_m \text{ is 'low'}), \text{ and } (h_n \text{ is 'medium'}), \text{ then } (q_m \text{ is 'good'}),$$

$$R_{2i}: \text{ If } (RT_n \text{ is 'low'}), (PE_m \text{ is 'high'}), \text{ and } (h_n \text{ is 'high'}), \text{ then } (q_m \text{ is 'bad'}).$$

Since the number of antecedents in each rule is 3, a full set of rules comprises  $N_r = 18$  rules.

- 3) **Defuzzification:** It provides the fitness of a control sequence for each rule  $R_{ri}$ , which is given by

$$\alpha_{R_{ri}}(q_m) = w_r^{RT_n} \cdot \mu_{Tr_1}(RT_n) \cdot w_r^{PE_m} \cdot \mu_{Tr_2}(PE_m) \cdot w_r^{h_n} \cdot \mu_{Tr_3}(h_n), \quad (36)$$

where  $\mu_{Tr_1}(RT_n)$ ,  $\mu_{Tr_2}(PE_m)$ , and  $\mu_{Tr_3}(h_n)$  are the fuzzy sets for each criterion considered in rule  $r$ ; and  $w_r^{RT_n}$ ,  $w_r^{PE_m}$ , and  $w_r^{h_n}$  are the considered weights that depend on the linguistic variables included in the rule. In this work, the weights, whose values only depend on their associated fuzzy sets, are independent of the rule and the subsystem where they are applied. Specifically, weights  $w^{RT}$ ,  $w^{PE}$ , and  $w^h$  are

$$w^{RT} = \begin{cases} w_1 & \text{if } RT \text{ is } T_1 \text{ (high)} \\ w_2 & \text{if } RT \text{ is } T_2 \text{ (low)} \end{cases}, \quad w^{PE} = \begin{cases} w_3 & \text{if } PE \text{ is } T_3 \text{ (high)} \\ w_4 & \text{if } PE \text{ is } T_4 \text{ (medium)} \\ w_5 & \text{if } PE \text{ is } T_5 \text{ (low)} \end{cases},$$

$$w^h = \begin{cases} w_6 & \text{if } h \text{ is } T_6 \text{ (high)} \\ w_7 & \text{if } h \text{ is } T_7 \text{ (medium)} \\ w_8 & \text{if } h \text{ is } T_8 \text{ (low)} \end{cases}.$$

Finally, the total fitness of the control signal  $q_m$  for the full set of  $N_r$  rules is

$$T\alpha(q_m) = \sum_{r=1}^{N_r} \alpha_{R_{ri}}(q_m). \quad (37)$$

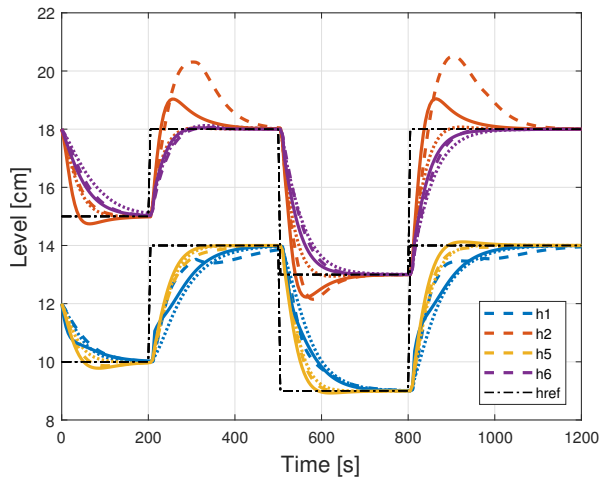
The specific weights of each linguistic label are defined in the results section.

## 6. Results

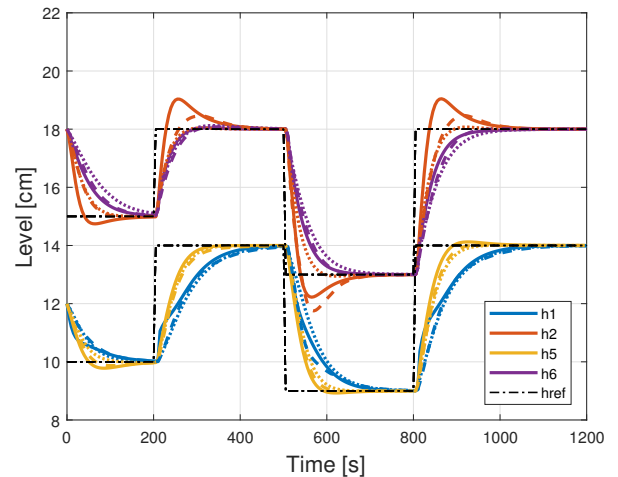
Results of the proposed DMPC with fuzzy negotiation are here presented. The sampling time used in simulations is  $T_s = 5$  s, and the MPC controllers consider a prediction horizon  $N_p = 20$ . To show the benefits of the proposed Fuzzy DMPC, it is compared with the centralized MPC and DMPC using a cooperative game (Maestre et al., 2011a), considering the same tuning parameters for local controllers. The influence of the fuzzy rules is also evaluated. The weights considered for each performance indicator in the fuzzy rules are shown in Table 2.

Table 2: Weights of the three performance indicators for several criteria.

		Weights					
	Alternative	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
$RT_n$	Low ( $w_1$ )	0.01	1	0.01	0.01	0.01	0.01
	High ( $w_2$ )	1	0.01	1	1	1	1
$h_m$	Low ( $w_3$ )	0.01	0.01	1	0.02	1	1
	Medium ( $w_4$ )	0.5	0.5	0.5	0.5	0.5	0.5
	High ( $w_5$ )	1	1	0.02	1	0.02	0.02
$PE_m$	Low ( $w_6$ )	1	1	1	1	0.02	0.9
	Medium ( $w_7$ )	0.5	0.5	0.5	0.5	0.5	0.5
	High ( $w_8$ )	0.01	0.01	0.01	0.01	0.9	0.02



(a) Fuzzy DMPC (Case 5).



(b) Fuzzy DMPC (Case 6).

Figure 9: Comparison of tanks level evolution (solid line: Centralized MPC, dashed line: Fuzzy DMPC, and dotted line: DMPC cooperative game).



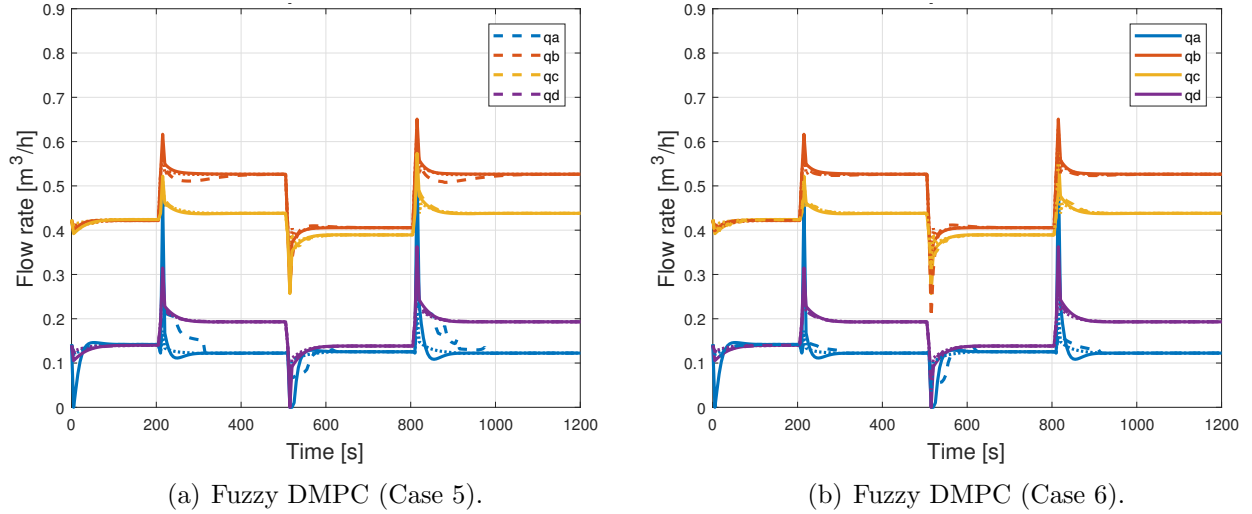


Figure 10: Comparison of pumps flow rates (solid line: Centralized MPC, dashed line: Fuzzy DMPC, and dotted line: DMPC cooperative game).

Table 3: Comparison of  $ISE$  and  $PE$  for the considered control techniques.

	Control techniques			
	Fuzzy DMPC (Case 5)	Fuzzy DMPC (Case 6)	DMPC Coop. Game	Centralized MPC
<b>Total <math>ISE</math></b>	0.7913	0.7335	0.7867	0.6171
<b>Total <math>PE</math></b>	$1.605 \cdot 10^{-2}$	$1.597 \cdot 10^{-2}$	$1.600 \cdot 10^{-2}$	$1.601 \cdot 10^{-2}$

As displayed in Table 3, Fig. 9, and Fig. 10, centralized MPC is always the controller with the smallest Integral Squared Error ( $ISE$ ) for tracking due to the availability of full plant information for prediction. On the other hand, fuzzy DMPC (Case 5) presents a higher  $ISE$  than DMPC with the cooperative game because the fuzzy rules favor high pumping energies (see Fig. 9 (a) and Fig. 10 (a)). On the contrary, fuzzy DMPC (Case 6) presents smaller  $ISE$  and  $PE$  than DMPC with the cooperative game because the fuzzy weights favor small  $PE$ , but still higher than centralized MPC (see Fig. 9 (b) and Fig. 10 (b)). Although the pumping energy is considered in absolute value in these simulations, increments could also be included as a performance indicator for the fuzzy negotiation. The conclusion is that fuzzy DMPC performance depends strongly on the weights of the considered fuzzy rules, giving the control system designer freedom to promote different behaviors. In this regard, the flexibility of the fuzzy approach simplifies the controller's tuning to attain better Key Performance Indicators (KPIs). As for the standard MPC approaches, tuning parameters (typically  $Q_i$ ,  $R_i$ , and  $N_p$ ) are indirectly related to these KPIs, and hence, complicating the controller setup. In the rest of this section, the standard MPC controllers will remain fixed to assess the tuning of the fuzzy rules in our approach.

Table 4: Comparison of  $ISE$  and  $PE$  for the considered cases.

	Weights					
	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
<b>Total <math>ISE</math></b>	0.7456	0.7308	0.7364	0.7456	0.7913	0.7335
<b>Total <math>PE</math></b>	$1.603 \cdot 10^{-2}$	$1.596 \cdot 10^{-2}$	$1.597 \cdot 10^{-2}$	$1.603 \cdot 10^{-2}$	$1.605 \cdot 10^{-2}$	$1.597 \cdot 10^{-2}$

In the following set of results, the influence of fuzzy rules is evaluated (Table 4). In Fig. 11, the  $RT$  in lower tanks are presented for two cases (Case 1 and Case 2) with different fuzzy-rule weights. In Figs. 12 and 13, the tank levels and pump flow rates are presented for this comparison. Case 1 gives more importance to high  $RT$  because  $w_2$  is much larger than  $w_1$ , resulting in higher values, particularly in the transients due to step reference changes.

Table 4 displays  $ISE$  and  $PE$  results and the noticeable effect of weights. For example, Case 6 produces lower  $PE$  than Case 5 because the weight of the fuzzy set corresponding to low pumping energies is higher than in Case 5. Regarding Fig. 14 (a), it is obtained lower levels of  $h_1, h_2, h_5, h_6$  in Case 3 than in Case 4 because the weight for the fuzzy set representing low levels is the highest. Since the bounds of the fuzzy sets for the water level are around 0.20 m (see Table 1), this is particularly noticeable in  $h_2$ , which is maintained around this value.

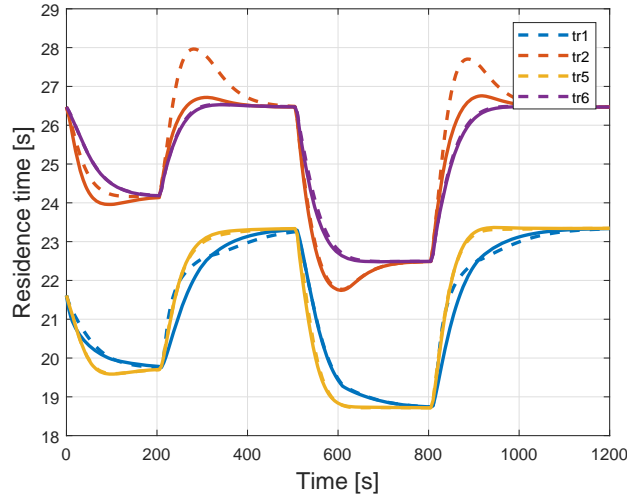


Figure 11: Residence time comparison for Case 1 (dashed line) and Case 2 (solid line).

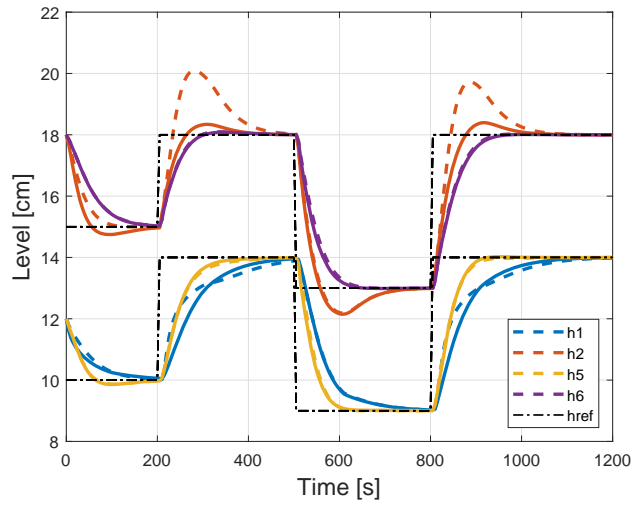


Figure 12: Water levels comparison for Case 1 (dashed line) and Case 2 (solid line).

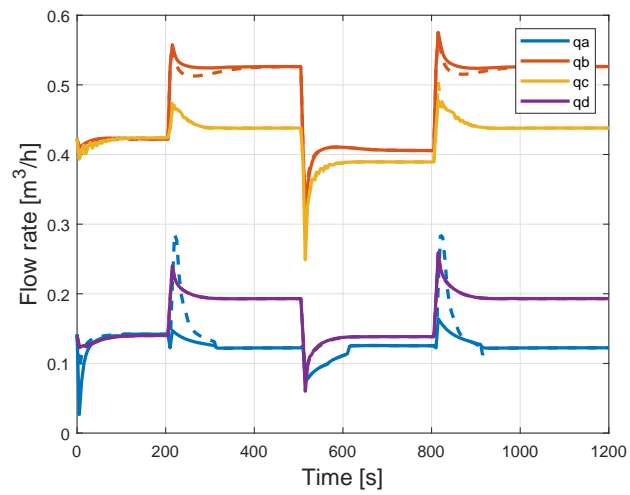


Figure 13: Flow rates comparison for Case 1 (dashed line) and Case 2 (solid line).

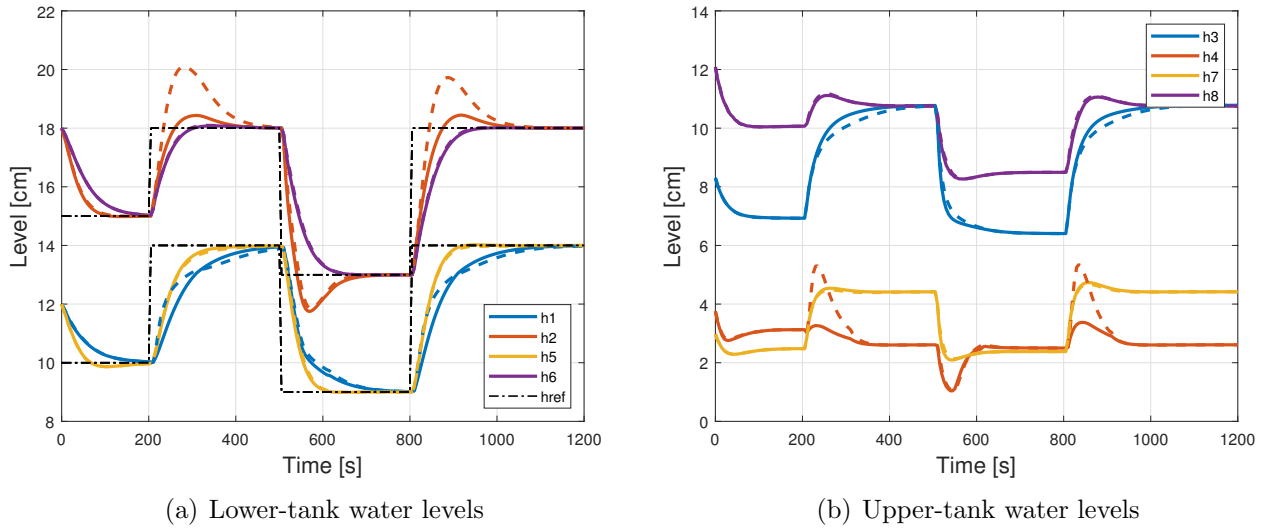


Figure 14: Results comparison for Case 3 (solid line) and Case 4 (dashed line).

Finally, the performed experiments show that the cost of centralized MPC is the lowest in all cases, acting as a lower bound of DMPC approaches. Moreover, the evolution of the system cost decreases as water levels are stabilized towards their set-points. It is caused by the inclusion of stability conditions in the fuzzy negotiation. In Fig. 15, the values of the overall Lyapunov function are presented for Case 5 and Case 6.

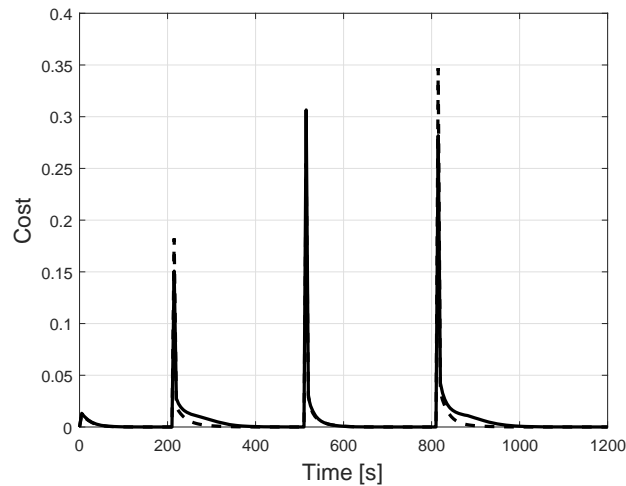


Figure 15: Overall Lyapunov function for Case 5 (solid line) and Case 6 (dashed line).

## 7. Conclusions

A distributed MPC with fuzzy cooperative negotiations has been developed based on a two-layer control architecture. The results achieved are satisfactory, obtaining reference tracking with similar performance to centralized MPC and outperforming DMPC without

fuzzy negotiation. Fuzzy negotiation presents a remarkable advantage over other methods since the computation of the final input do not require many communication steps, but just merging all the inputs proposed by the involved agents. Moreover, it also preserves privacy of the internal state and model of the agents, which is also an interesting property for some systems. The proposed method can then be extended to other complex plants with many agents because the lower layer negotiation avoids a combinatorial explosion. Furthermore, it is provided a more intuitive controller's tuning to achieve target values regarding the considered performance indicators.

Finally, a linear feedback controller is considered as backup controller to ensure stability when the fuzzy inference increases the cost. Moreover, the performed simulations show that the control action provided by fuzzy inference is mostly used in both layers, showing the usefulness of the methodology.

Future work will deal with the application of the assessment of this method for nonlinear systems, for few schemes are available in the literature for such problem setting.

### Author contributions

E. Masero: Investigation, Writing– Original draft preparation, Visualization, Writing– Reviewing & Editing, Funding acquisition; M. Francisco: Methodology, Investigation, Data curation, Software, Validation, Writing– Reviewing & Editing, Project administration; J. M. Maestre: Methodology, Supervision, Formal analysis, Writing– Reviewing & Editing, Funding acquisition; S. Revollar: Conceptualization, Methodology, Formal analysis, Writing– Reviewing & Editing; P. Vega: Conceptualization, Supervision, Formal analysis, Resources, Funding acquisition.

### Acknowledgements

The authors gratefully acknowledge the financial support by the European Research Council [Advanced Research Grant, project number 789051 OCONTSOLAR], the Spanish Ministry of Economy [project numbers DPI2017-86918-R and PID2019-105434RB-C31], the Samuel Solórzano Foundation [project FS/20-2019], and the Spanish Ministry of Science, Innovation, and Universities [grant number FPU18/04476].

### References

- Alvarado, I., Limón, D., Muñoz de la Peña, D., Maestre, J. M., Ridao, M. A., Scheu, H., Marquardt, W., Negenborn, R. R., De Schutter, B., Valencia, F. et al. (2011). A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark. *Journal of Process Control*, *21*, 800–815.
- Bai, Y., & Wang, D. (2006). Fundamentals of fuzzy logic control—fuzzy sets, fuzzy rules and defuzzifications. In *Advanced Fuzzy Logic Technologies in Industrial Applications* (pp. 17–36). Springer.
- Berglind, J. B., Gommans, T., & Heemels, W. (2012). Self-triggered mpc for constrained linear systems and quadratic costs. *IFAC Proceedings Volumes*, *45*, 342–348.
- Darivianakis, G., Eichler, A., & Lygeros, J. (2019). Distributed model predictive control for linear systems with adaptive terminal sets. *IEEE Transactions on Automatic Control*, .

- Doan, M. D., Keviczky, T., & De Schutter, B. (2014). A hierarchical MPC approach with guaranteed feasibility for dynamically coupled linear systems. In *Distributed Model Predictive Control Made Easy* (pp. 393–406). Springer.
- El Haoussi, F., Tissir, E. H., Tadeo, F., & Hmamed, A. (2011). Delay-dependent stabilisation of systems with time-delayed state and control: Application to a quadruple-tank process. *International Journal of Systems Science*, *42*, 41–49.
- Francisco, M., Mezquita, Y., Revollar, S., Vega, P., & De Paz, J. F. (2019). Multi-agent distributed model predictive control with fuzzy negotiation. *Expert Systems with Applications*, *129*, 68–83.
- Herceg, M., Kvasnica, M., Jones, C. N., & Morari, M. (2013). Multi-parametric toolbox 3.0. In *Proceedings of the 12th European Control Conference (ECC)* (pp. 502–510). IEEE.
- Hsu, C., Kao, B., Li, L., Lai, K. R. et al. (2016). An agent-based fuzzy constraint-directed negotiation model for solving supply chain planning and scheduling problems. *Applied Soft Computing*, *48*, 703–715.
- Johansson, K. H. (2000). The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Transactions on Control Systems Technology*, *8*, 456–465.
- Kolmanovsky, I., & Gilbert, E. G. (1998). Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering*, *4*, 317–367.
- Kosonen, I. (2003). Multi-agent fuzzy signal control based on real-time simulation. *Transportation Research Part C: Emerging Technologies*, *11*, 389–403.
- Kowalczyk, R. (2002). Fuzzy e-negotiation agents. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, *6*, 337–347.
- Lazar, M., Heemels, W., Muñoz de la Peña, D., & Alamo, T. (2009). Further results on “robust MPC using linear matrix inequalities”. In *Collection of Nonlinear model predictive control* (pp. 89–98). Springer.
- Leith, D. J., & Leithead, W. E. (2000). Survey of gain-scheduling analysis and design. *International Journal of Control*, *73*, 1001–1025.
- Limón, D., Alvarado, I., Alamo, T., & Camacho, E. F. (2010). Robust tube-based MPC for tracking of constrained linear systems with additive disturbances. *Journal of Process Control*, *20*, 248–260.
- Maestre, J. M., Muñoz de la Peña, D., & Camacho, E. F. (2011a). Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods*, *32*, 153–176.
- Maestre, J. M., Muñoz de la Peña, D., & Camacho, E. F. (2009). A distributed MPC scheme with low communication requirements. In *Proceedings of the IEEE American Control Conference* (pp. 2797–2802).
- Maestre, J. M., Muñoz de la Peña, D., Camacho, E. F., & Alamo, T. (2011b). Distributed model predictive control based on agent negotiation. *Journal of Process Control*, *21*, 685–697.
- Maestre, J. M., Ridao, M. A., Kozma, A., Savorgnan, C., Diehl, M., Doan, M. D., Sadowska, A., Keviczky, T., De Schutter, B., Scheu, H. et al. (2015). A comparison of distributed MPC schemes on a hydro-power plant benchmark. *Optimal Control Applications and Methods*, *36*, 306–332.
- Magni, L., De Nicolao, G., Scattolini, R., & Allgöwer, F. (2003). Robust model predictive control for nonlinear discrete-time systems. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, *13*, 229–246.
- Marcos, N. I., Forbes, J. F., & Guay, M. (2014). Prediction-driven coordination of distributed MPC controllers for linear unconstrained dynamic systems. *International Journal of Control*, *87*, 1496–1512.
- McFarlane, D., & Glover, K. (1992). A loop-shaping design procedure using  $\mathcal{H}_\infty$  synthesis. *IEEE Transactions on Automatic Control*, *37*, 759–769.
- Mi, X., Zou, Y., Li, S., & Karimi, H. R. (2019). Self-triggered DMPC design for cooperative multiagent systems. *IEEE Transactions on Industrial Electronics*, *67*, 512–520.
- Negenborn, R. R., & Maestre, J. M. (2014). Distributed model predictive control: An overview and roadmap of future research opportunities. *IEEE Control Systems Magazine*, *34*, 87–97.
- Rakovic, S., Kerrigan, E., Kouramas, K., & Mayne, D. (2005). Robust MPC for nonlinear discrete-time systems. *IEEE Transactions on Automatic Control*, *50*, 406–410.
- Raviraj, V., & Sen, P. C. (1997). Comparative study of proportional-integral, sliding mode, and fuzzy logic controllers for power converters. *IEEE Transactions on Industry Applications*, *33*, 518–524.
- Rossiter, J. A. (2003). *Model-based predictive control: a practical approach*. CRC press.

- Saad, A., Youssef, T., Elsayed, A. T., Amin, A., Abdalla, O. H., & Mohammed, O. (2018). Data-centric hierarchical distributed model predictive control for smart grid energy management. *IEEE Transactions on Industrial Informatics*, *15*, 4086–4098.
- Sahebjamnia, N., Tavakkoli-Moghaddam, R., & Ghorbani, N. (2016). Designing a fuzzy Q-learning multi-agent quality control system for a continuous chemical production line – A case study. *Computers & Industrial Engineering*, *93*, 215–226.
- Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control – A review. *Journal of Process Control*, *19*, 723–731.
- Shneiderman, D., & Palmor, Z. J. (2010). Properties and control of the quadruple-tank process with multivariable dead-times. *Journal of Process Control*, *20*, 18–28.
- Sugeno, M. (1985). *Industrial applications of fuzzy control*. Elsevier Science Inc.
- Tan, W., Marquez, H. J., & Chen, T. (2003). IMC design for unstable processes with time delays. *Journal of Process Control*, *13*, 203–213.
- Thibodeau, B., Ren, Q., Bai, L., Biswas, S., Ferrese, F., & Dong, Q. (2013). Distributed fuzzy logic price negotiation in market based multi-agent control. In *Proceedings of the 6th International Symposium on Resilient Control Systems (ISRC)* (pp. 25–30). IEEE.
- Uddin, M. N., & Rahman, M. A. (1999). Fuzzy logic based speed control of an IPM synchronous motor drive. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering* (pp. 1259–1264). IEEE volume 3.
- Velarde, P., Tian, X., Sadowska, A. D., & Maestre, J. M. (2019). Scenario-based hierarchical and distributed MPC for water resources management with dynamical uncertainty. *Water Resources Management*, *33*, 677–696.
- Wang, Q., Hang, C. C., & Yang, X. (2001). Single-loop controller design via IMC principles. *Automatica*, *37*, 2041–2048.
- Xing, G., Lu, C., Zhang, Y., Huang, Q., & Pless, R. (2007). Minimum power configuration for wireless communication in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, *3*, 11.
- Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-3*, 28–44.
- Zafra-Cabeza, A., Maestre, J. M., Ridao, M. A., Camacho, E. F., & Sánchez, L. (2011). A hierarchical distributed model predictive control approach to irrigation canals: A risk mitigation perspective. *Journal of Process Control*, *21*, 787–799.
- Zhu, C., Khammash, M., Vittal, V., & Qiu, W. (2003). Robust power system stabilizer design using  $\mathcal{H}_\infty$  loop shaping approach. *IEEE Transactions on Power Systems*, *18*, 810–818.