A hierarchical distributed MPC based on fuzzy negotiation for multiple agents

Eva Masero\textsuperscript{a,}\textsuperscript{*}, Mario Francisco\textsuperscript{b}, José M. Maestre\textsuperscript{a}, Silvana Revollar\textsuperscript{b}, Pastora Vega\textsuperscript{c}

\textsuperscript{a}Departamento de Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, C/ Camino de los Descubrimientos, s/n., 41092, Sevilla (Spain).
\textsuperscript{b}Departamento de Informática y Automática, Escuela Técnica Superior de Ingeniería Industrial, Universidad de Salamanca, Av. Fernando Ballesteros, Béjar, 37700, Salamanca (Spain).
\textsuperscript{c}Departamento de Informática y Automática, Facultad de Ciencias, Universidad de Salamanca, Plaza de la Merced, s/n, 37008, Salamanca (Spain).

Abstract

This work presents a multi-agent distributed model predictive control 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 the upper control layer, which merge 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 tanks plant via simulation.

Keywords: Distributed model predictive control (DMPC), Pairwise negotiations, Fuzzy logic, Multi-agent systems, Stability.

1. Introduction

Centralized control methods offer, in general, the best achievable 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. To begin with, for complex large-scale processes, it is merely not feasible to compute control actions in this manner due to timing constraints. Additionally, other systems are spread geographically (e.g., traffic, power and water networks, etc.) hence being naturally distributed, 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.

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

*Corresponding author.

Email addresses: evamasero@us.es (Eva Masero), mfs@usal.es (Mario Francisco), pepemaestre@us.es (José M. Maestre), srevolla@usal.es (Silvana Revollar), pvega@usal.es (Pastora Vega)
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 calculate optimal control actions during a given horizon, 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, multiple distributed MPC (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 distributed MPC 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) shows 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 throughout this line of research can be found in the literature, 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 the problem of 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. This fact is formalized by defining fuzzy sets as an extension of the crisp sets that consider a limited number of degrees of membership such as ‘0’ and ‘1’. They represent linguistic labels for relevant characteristics of any process variable, such as ‘good’, ‘bad’, ‘high’, ‘medium’, or ‘low’ (Bai & Wang, 2006). Therefore, fuzzy sets’ boundaries can be ambiguous and are typically presented graphically by their membership functions. The usefulness of considering fuzzy logic in the negotiation framework is the possibility to characterize the suitability of a 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 the traffic lights. Sahebjamnia et al. (2016) propose a fuzzy Q-learning algorithm in the distributed control of chemical plants. Negotiation with fuzzy constraints for 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 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, also 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 proposed 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 more than two subsystems. Thus, the original negotiation scheme is modified to deal with multi-agent systems by adding an extra fuzzy step. It also takes into account additional criteria in the decision-making process to soften the trajectory of the 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. This set is the input for the upper control layer, also fuzzy-based, which merges the resulting proposals to obtain new control sequences for the system. Moreover, stability can be guaranteed for this fast decision-making scheme. Note that mixed Fuzzy-MPC approaches are rare in the literature, except for Francisco et al. (2019), a previous work where fuzzy negotiation has been applied to a four-coupled-tank system, but without stability guarantees and limited to systems with only two agents.

The benefits of this approach 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 been greatly 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); the stabilization of linear systems with delays (El Haoussi et al., 2011), etc.

The organization of this paper is 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 an 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:** The sets of natural and positive real numbers are respectively denoted \( \mathbb{N} \) and \( \mathbb{R}_+ \). The notation \( \mathbb{N}_0 \) indicates a set of non-negative integers, and \( \mathbb{R}^n \) refers to an \( n \)-dimension Euclidean space. The scalar product of vectors \( a, b \in \mathbb{R}^n \) is denoted \( ab^\top \) or \( a \cdot b \). For the sets \( \mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^n \), the Cartesian product is denoted by \( \mathcal{X} \times \mathcal{Y} \equiv \{ (x, y) : x \in \mathcal{X}, y \in \mathcal{Y} \} \). If \( \{ \mathcal{X}_i \}_{i \in \mathcal{N}} \) is a family of sets indexed by \( \mathcal{N} \), then the Cartesian product is defined as \( \bigtimes_{i \in \mathcal{N}} \mathcal{X}_i \equiv \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_N = \{ (x_1, x_2, \ldots, x_N) : x_1 \in \mathcal{X}_1, \ldots, x_N \in \mathcal{X}_N \} \). The set subtraction operation is symbolized by \( \setminus \). For the sets \( \mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^n \), the Minkowski sum is represented by \( \mathcal{X} \oplus \mathcal{Y} \equiv \{ x + y : x \in \mathcal{X}, y \in \mathcal{Y} \} \). The image of a set \( \mathcal{X} \subseteq \mathbb{R}^n \) under a linear mapping \( A : \mathbb{R}^n \rightarrow \mathbb{R}^m \) is represented as \( A\mathcal{X} \equiv \{ Ax : x \in \mathcal{X} \} \). The integral of
the squared error \( \varepsilon \) is defined as \( \text{ISE} = \int_0^{\infty} \varepsilon^2 dt \). An \( N_p \) - long future input sequence is defined as \( U(k) = [u(k), u(k+1), \ldots, u(k+N_p-1)]^T \) and an optimal sequence is denoted using the superscript \( * \), i.e., \( U^*(k) = [u^*(k), u^*(k+1), \ldots, u^*(k+N_p-1)]^T \).

2. Problem Formulation

Let the system be composed of a set \( \mathcal{N} = \{1, 2, \ldots, N\} \) of input-coupled subsystems whose state evolution is given by

\[
x_i(k+1) = A_i x_i(k) + B_i u_i(k) + w_i(k),
\]

where \( k \in \mathbb{N}_p \) denotes the sampling time; \( x_i \in \mathbb{R}^n \) 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 \( \mathcal{X}_i = \{ x_i \in \mathbb{R}^n : A x_i x_i \leq b_{x,i} \} \) and \( \mathcal{U}_i = \{ u_i \in \mathbb{R}^r_i : A_{u,i} u_i \leq b_{u,i} \} \), respectively; and \( A_i \in \mathbb{R}^{n \times n} \) and \( B_i \in \mathbb{R}^{n \times r_i} \) are matrices of proper dimensions of each subsystem. The vector \( w_i \in \mathbb{R}^n \) represents the measurable disturbances resulting from the coupling with other subsystems \( j \) belonging to the set of neighbors \( \mathcal{N}_i = \{ j \in \mathcal{N} \setminus \{i\} : B_{ij} \neq 0 \} \), i.e.,

\[
w_i(k) = \sum_{j \in \mathcal{N}_i} B_{ij} u_j(k),
\]

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

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

\[
x_N(k+1) = A_N x_N(k) + B_N u_N(k),
\]

where \( A_N = [A_{ij}]_{i,j \in \mathcal{N}} \) and \( B_N = [B_{ij}]_{i,j \in \mathcal{N}} \) are respectively the state and input-to-state matrices of the global system, and the state and input vectors are constrained in \( \mathcal{X}_N = \bigtimes_{i \in \mathcal{N}} \mathcal{X}_i \) and \( \mathcal{U}_N = \bigtimes_{i \in \mathcal{N}} \mathcal{U}_i \), respectively. Regarding mutual interaction, the term \( w_N \) is implicitly included in (3), i.e., in the dynamics of the overall system.

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. In particular, the cost function of subsystem \( i \in \mathcal{N} \) is calculated based on the predicted trajectories of its states and inputs along a window of length \( N_p \), the so-called prediction horizon, and is expressed by

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

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

\[
L_i(x_i(k+t), u_i(k+t)) = x_i(k+t)^T Q_i x_i(k+t) + u_i(k+t)^T R_i u_i(k+t),
\]

\[
F_i(x_i(k+N_p)) = x_i(k+N_p)^T P_i x_i(k+N_p),
\]

with \( Q_i \) being a semi-positive definite matrix and \( R_i, P_i \) positive-definite matrices.
3. Proposed Control Architecture

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

The multi-layered fuzzy-based control architecture is introduced 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 evolution. The sequences resulting from the pairwise negotiations are sent to the upper control layer, which merges and fuzzifies them to compute the final control sequence for the whole system.

![Diagram of Proposed Control Architecture](image)

Figure 1: Proposed control architecture.

3.1. Low-level Control Layer

In this layer, a DMPC algorithm for multiple agents is proposed. Specifically, fuzzy-based negotiations are made in pairs taking into account the coupling with their neighboring subsystems, which are assumed to hold their current trajectories. To this end, a shifted sequence of agent \(i\) is used, which is defined by adding \(K_i x_i(k + N_p)\) to the sequence chosen at the previous time step \(U_i(k - 1)\) as follows

\[
U^*_i(k) = \begin{bmatrix}
u_i(k + 1|k - 1) \\
u_i(k + 2|k - 1) \\
\vdots \\
u_i(k + N_p - 1|k - 1) \\
K_i x_i(k + N_p|k - 1)
\end{bmatrix} = \begin{bmatrix}
u_i^*(k) \\
u_i^*(k + 1) \\
\vdots \\
u_i^*(k + N_p - 2) \\
u_i^*(k + N_p - 1)
\end{bmatrix}.
\] (5)

3.1.1. DMPC 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.

1. At sampling time $k$, 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 its neighbor $j$ applies its shifted trajectory $U_j^s(k)$. It is assumed that the rest of the neighboring subsystems $l \in \mathcal{N}_i \setminus \{j\}$ follow their current control trajectories $U_l^i(k)$. Specifically, agent $i$ solves

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

s.t.

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

$$x_i(k) = \tilde{x}_i(k), \quad i \in \mathcal{N},$$

$$x_i(k + t) \in \mathcal{X}_i, \quad t = 0, \ldots, N_p - 1,$$

$$x_i(k + N_p) \in \Omega_i,$$

$$u_i(k + t) \in \mathcal{U}_i, \quad t = 0, \ldots, N_p - 1,$$

$$u_j(k + t) = u_j^s(k + t), \quad t = 0, \ldots, N_p - 1,$$

$$u_l(k + t) = u_l^i(k + t), \quad t = 0, \ldots, N_p - 1,$$

where the 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 its wished neighboring input sequence $U_{j^w_i}(k)$. Here, it is also assumed that subsystems $l$ follow their current trajectories. To this end, agent $i$ solves

$$U_{j^w_i}(k) = \arg\min_{U_{j}(k)} J_i(x_i(k), U_i^*(k), U_j(k), U_l^i(k)),$$  \hfill (7)

s.t.

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

$$x_i(k) = \tilde{x}_i(k), \quad i \in \mathcal{N},$$

$$x_i(k + t) \in \mathcal{X}_i, \quad t = 0, \ldots, N_p - 1,$$

$$x_i(k + N_p) \in \Omega_i,$$

$$u_i(k + t) = u_i^*(k + t), \quad t = 0, \ldots, N_p - 1,$$

$$u_j(k + t) \in \mathcal{U}_j, \quad j \in \mathcal{N}_i, \quad t = 0, \ldots, N_p - 1,$$

$$u_l(k + t) = u_l^i(k + t), \quad t = 0, \ldots, N_p - 1.$$

5. Agent $i$ sends $U_{j^w_i}(k)$ to agent $j$, and receives $U_{i^w_j}(k)$. 
3.1.2. Fuzzy Pairwise Negotiation

Following Stage 1, the idea is to apply fuzzy negotiation to obtain a control solution that decreases the performance index and thus, guarantees stability. To this end, each pair of agents $i, j$ has the sequences of control actions $\{U_i^s(k), U_i^w(k)\}$ and $\{U_j^s(k), U_j^w(k)\}$. 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 shows the fuzzy negotiation scheme (Uddin & Rahman, 1999) that will be implemented inside the control loop.

![Fuzzy Negotiation Scheme](image)

The main three steps of the negotiation process are the following:

1. **Fuzzification**: The fuzzification 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 numerical value for the temperature in a boiler is characterized by the degree of ‘high’ and ‘low’ that it is, or if more linguistic labels are available, by the degree of ‘dangerous’, ‘high’, ‘medium’ and ‘low’ that the temperature is.

Let $x$ be a state, input, or other algebraic variable characterizing the process, such as residence time in a water tank, energy consumption, temperature, etc. 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 $\mu_{T_2}(x)$ are respectively associated with the linguistic labels ‘high’ and ‘low’ as shown in Fig. 3.

$$\mu_{T_1}(x) = \begin{cases} 
1 & \text{for } x < a \\
\frac{b - x}{b - a} & \text{for } a \leq x < b \\
0 & \text{for } b \leq x
\end{cases}, \quad \mu_{T_2}(x) = \begin{cases} 
0 & \text{for } x < a \\
\frac{a - x}{a - b} & \text{for } a \leq x < b \\
1 & \text{for } b \leq x
\end{cases}. \quad (8)$$
Occasionally, the process knowledge allows for the definition of more than two linguistic labels. In this work, the methodology is presented for three linguistic labels, although its extension is straightforward. Let $y$ be another state, input, or algebraic variable of the process. If three linguistic labels are considered, three fuzzy sets $T_3$, $T_4$, and $T_5$ must be defined. The membership functions $\mu_{T_3}(y)$, $\mu_{T_4}(y)$, and $\mu_{T_5}(y)$ are respectively associated with the linguistic labels ‘high’, ‘medium’, and ‘low’, as shown in Fig. 4.

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

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

$$
\mu_{T_5}(y) = \begin{cases} 
0 & \text{for } y < c \\
\frac{c - y}{c - b} & \text{for } c \leq y < b \\
1 & \text{for } b \leq y 
\end{cases}
$$
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 can be written linguistically (Raviraj & Sen, 1997) as follows

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 the fuzzy sets corresponding to some linguistic labels for variables \( L_A, L_B, L_C \), respectively, being \( L_C \) the control action that is evaluated in the negotiation. The use of rules with more than two antecedents or variables is straightforward and 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. There are several defuzzification methods, as shown by Bai & Wang (2006), but in this case, Sugeno-type fuzzy inference has been used (Sugeno, 1985). Each rule weighs the different antecedents depending on the linguistic variables considered as follows

\[
\begin{align*}
  w^x_r &= \begin{cases} 
    w_1 & \text{if } x \text{ is } T_1 \text{ (high)} \\
    w_2 & \text{if } x \text{ is } T_2 \text{ (low)} 
  \end{cases}, \\
  w^y_r &= \begin{cases} 
    w_3 & \text{if } y \text{ is } T_3 \text{ (high)} \\
    w_4 & \text{if } y \text{ is } T_4 \text{ (medium)} \\
    w_5 & \text{if } y \text{ is } T_5 \text{ (low)}
  \end{cases}.
\end{align*}
\]

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

\[
\alpha_{R_r}(U) = w^x_r \cdot \mu_{T_{L_1}}(x) \cdot w^y_r \cdot \mu_{T_{L_2}}(y),
\]

(10)
where \( r \) is the \( r \)-th rule, the variables \( x \) and \( y \) are those associated with the corresponding \( U \), and \( \mu_{T^1_r}(x), \mu_{T^2_r}(y) \) are the fuzzy sets considered in rule \( r \). Hence, the total fitness of the control signal for the set of \( N_r \) rules is given by

\[
T\alpha(U) = \sum_{r=1}^{N_r} \alpha_{R_r}(U). \tag{11}
\]

The last part of the procedure consists of merging all proposed control sequences to obtain the final one. In particular, for the proposed DMPC, outputs \( U^f_i \) and \( U^f_j \) are computed, respectively, as

\[
U^f_i(k) = \frac{U_i(k) \cdot T\alpha(U^p_i(k)) + U^*_i \cdot T\alpha(U^*_i(k)) + U^w_i(k) \cdot T\alpha(U^w_i(k))}{T\alpha(U^p_i(k)) + T\alpha(U^*_i(k)) + T\alpha(U^w_i(k))}, \tag{12}
\]

\[
U^f_j(k) = \frac{U_j(k) \cdot T\alpha(U^p_j(k)) + U^*_j \cdot T\alpha(U^*_j(k)) + U^w_j(k) \cdot T\alpha(U^w_j(k))}{T\alpha(U^p_j(k)) + T\alpha(U^*_j(k)) + T\alpha(U^w_j(k))}. \tag{13}
\]

Bear in mind that control sequences \( U^w_i(k) \) and \( U^w_j(k) \) can be excluded from their fuzzification procedure if they lead their subsystems to unfeasibility. Moreover, note that the feasibility of the final control sequence obtained after the fuzzification process is also ensured because optimization problems (6) and (7) are convex, and (12) and (13) are linear combinations of feasible sequence.

Finally, the fuzzy-pairwise negotiation procedure can be summarized as follows.

\begin{algorithm}
\begin{algorithmic}
\State \textbf{Stage 2: Proposals for fuzzification.}
\begin{enumerate}
\item There are three different possible input trajectories for each agent, namely \( U_i \in \{U^p_i(k), U^w_i(k), U^*_i(k)\} \) and \( U_j \in \{U^p_j(k), U^w_j(k), U^*_j(k)\} \).
\State Since the latter two are computed without taking into account neighbor state constraints, it is checked whether they are feasible to exclude them from the fuzzification in the case they are not. Afterwards, fuzzy negotiation is applied to compute final input sequences \( U^f_i(k) \) and \( U^f_j(k) \).
\item A resulting pairwise fuzzy negotiation sequence is defined based on \( U^f_i(k) \) and \( U^f_j(k) \) assuming that the rest of subsystems follow their pre-defined trajectories.
\item Agents \( i, j \) exchange information with their neighbors regarding their cost function for their fuzzy and stabilizing input trajectories. If it is fulfilled the condition
\[
\sum_{l \in M_i \cup M_j \cup \{i,j\}} J_i(x_l(k), U^f_{ij}) \leq \sum_{l \in M_i \cup M_j \cup \{i,j\}} J_i(x_l(k), U^p_{ij}), \tag{14}
\]

where \( U^p_{ij}(k) = \{U^p_i(k), U^p_j(k), U^p_l(k)\} \) and \( U^w_{ij}(k) = \{U^w_i(k), U^w_j(k), U^w_l(k)\} \), for \( l \in N \setminus \{i, j\} \), then stability is guaranteed and \( U^f_i(k) \) and \( U^f_j(k) \) are sent to the upper control layer. Otherwise, \( U^w_i(k) \) and \( U^w_j(k) \) are sent.
\end{enumerate}
\end{algorithmic}
\end{algorithm}

Finally, note that there may exist up to \( N^2 - N \) negotiation problems running in parallel at each sampling time if all subsystems were coupled with their neighbors. Nevertheless, as shown in Fig. 5, only subsystems with communication links enabled due to their high coupling follow the cooperative scheme proposed.
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 agent \( j \) sends its own trajectory similarly. Once the information is received, agents \( i, j \) compute their optimal sequences \( U_i^s(k), U_j^s(k) \) and their \textit{wished} neighboring input trajectories \( U_i^{w_i}(k) \) and \( U_j^{w_j}(k) \), and share them. Before applying fuzzification, it is necessary to check the feasibility of \( U_i^{w_i}(k) \) and \( U_j^{w_j}(k) \). Unless feasibility, these input sequences will not be included in the set of sequences to fuzzificate. Then, agents communicate their result to each other. In the fourth step, the sequences resulting from the fuzzification \( U_i^f(k), U_j^f(k) \) are shared. Finally, it is necessary to check whether condition (14) is satisfy. Otherwise, the final input trajectories of agents \( i \) and \( j \) will be \( U_i^f(k) = U_i^s(k) \) and \( U_j^f(k) = U_j^s(k) \), respectively. In the fifth step, it is shared the result.
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 an agent $i \in \mathcal{N}$ coupled through inputs with their neighbors and being the control sequences obtained from $M$ pairwise fuzzy negotiations $\{U_i^{f1}(k), U_i^{f2}(k), \ldots, U_i^{fM}(k)\}$, the upper-level control layer is responsible for applying a final fuzzification process with all these options. As a result, a final control sequence $U_i^f(k)$ is obtained and will be finally applied by this agent.

Following the fuzzy inference steps shown in the sub-subsection 3.1.2, the final control sequence for agent $i$ is computed as a weighted average of the control sequences from the pairwise fuzzy negotiations with its neighbors and can be described by the following expression

$$U_i^f(k) = \frac{U_i^{f1}(k) \cdot T\alpha(U_i^{f1}(k)) + U_i^{f2}(k) \cdot T\alpha(U_i^{f2}(k)) + \cdots + U_i^{fM}(k) \cdot T\alpha(U_i^{fM}(k))}{T\alpha(U_i^{f1}(k)) + T\alpha(U_i^{f2}(k)) + \cdots + T\alpha(U_i^{fM}(k))}, \quad (15)$$

where $M$ is the number of negotiations, $T\alpha(u)$ is the total fitness of the control signal for the set of rules used, and $U_i^f(k)$ is the final control sequence of agent $i$. 

Figure 6: Outline of the communication steps involved in the pairwise negotiation between agents $i$ and $j$ for the proposed lower-level control layer.
A control sequence of the overall system $U_N^f(k)$ can be obtained aggregating all the final sequences, i.e., $U_N^f(k) = [U_i^f(k)]_{i \in N}$. Hence, the cost function of the global system is expressed as

$$J_N^f(x_N(k), U_N^f(k)) = \sum_{t=0}^{N_p-1} L_N(x_N(k+t), u_N(k+t)) + F_N(x_N(k+N_p)),$$

where $L_N(\cdot)$ and $F_N(\cdot)$ are respectively the global stage and terminal cost functions defined as

$$L_N(x_N(k+t), u_N(k+t)) = x_N(k+t)^T Q_N x_N(k+t) + u_N(k+t)^T R_N u_N(k+t),$$

$$F_N(x_N(k+N_p)) = x_N(k+N_p)^T P_N x_N(k+N_p),$$

with the weighting matrices $Q_N = [Q_i]_{i \in N}$ and $R_N = [R_i]_{i \in N}$; and the terminal cost matrix $P_N = [P_i]_{i \in N}$. To guarantee stability, the cost-to-go of the overall system must decrease, i.e.,

$$J_N^f(x_N(k+1), U_N^f(k+1)) \leq J_N^f(x_N(k), U_N^f(k)).$$

Otherwise, the final control sequence $U_N^f(k+1)$ will be implemented because it decreases the overall cost, as it is proved in the subsection 4.3.

Algorithm 2 displays a summary of the steps implemented by this layer.

### Algorithm 2

1. From $M$ pairwise negotiations, the upper control layer receives a set of sequences $\{U_i^{f_1}(k), U_i^{f_2}(k), \ldots, U_i^{f_M}(k)\}$ for each agent $i \in N$.
2. The final input sequence $U_i^f$ of each agent $i$ is calculated as a weighted average of sequences from its pairwise negotiations.
3. All final control sequences of agents are aggregated to obtain a global one for the overall system, i.e., $U_N^f(k) = [U_i^f(k)]_{i \in N}$.
4. Then, the overall cost $J_N^f(x_N(k+1), U_N^f(k+1))$ is computed and compared with the cost from the previous time instant $k$. If the cost decreases, $U_N^f(k+1)$ is applied to the system. Otherwise, $U_N^f(k+1)$ is finally implemented.

### 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 convexity of the problem. This strategy is supported by an auxiliary feedback, which provides the basis for the 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 linear matrix inequalities (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 the viewpoint of each agent $i \in N$, 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 described by (1), there is a feedback $K_i$ that ensures that all eigenvalues of $(A_i + B_i K_i)$ are within the unit circle. Likewise, the same holds for the corresponding global feedback $K_N = \text{diag}(K_i)_{i \in \mathbb{N}}$.

Definition 1. The set $\Omega_i$ is a robust positively invariant (RPI) set for subsystem (1) if and only if its evolution satisfies

$$x_i(k) \in \Omega_i \rightarrow x_i(k + 1) \in \Omega_i, \quad \forall w_i(k) \in \mathcal{W}_i, \quad \forall k \in \mathbb{N}_0. \quad (19)$$

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

$$(A_i + B_i K_i) \Omega_i \supseteq \Omega_i, \quad \Omega_i \subseteq \mathcal{X}_i, \quad K_i \Omega_i \subseteq \mathcal{U}_i, \quad (20)$$

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

Taking into account Assumption 2, the RPI set of the overall system can be computed by the Cartesian product of all $\Omega_i$, i.e.,

$$\Omega_N = \times_{i \in \mathbb{N}} \Omega_i. \quad (21)$$

There are several methods to find the 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 MATLAB-based Multi-Parametric Toolbox (MPT) given by Herceg et al. (2013) is employed to compute the maximal RPI set for each subsystem.

Assumption 3. There exists a Lyapunov function $V_N(x_N(k)) = x_N(k)^T P_N x_N(k)$ with $P_N = \text{diag}(P_i)_{i \in \mathbb{N}}$ controlled by $K_N = \text{diag}(K_i)_{i \in \mathbb{N}}$ that provides an upper bound on the cost-to-go of the system, i.e.,

$$V_N(x_N(k)) \geq \sum_{t=0}^{\infty} L_N(x_N(k + t), u_N(k + t)). \quad (22)$$

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) methods can be found in the literature. Here, matrices $K_N$ and $P_N$ for the control of the global system are obtained solving the following LMI such as Magni et al. (2003); Lazar et al. (2009); Darivianakis et al. (2019).

Theorem 1. Let the system be divided into $N$ subsystems with global matrices $A_N = [A_{ij}]_{i,j \in \mathbb{N}}$ and $B_N = [B_{ij}]_{i,j \in \mathbb{N}}$, and stage cost matrices $Q_N = \text{diag}(Q_i)_{i \in \mathbb{N}}$ and $R_N = \text{diag}(R_i)_{i \in \mathbb{N}}$. If there are matrices $H_N = H_N^T = \text{diag}(H_i)_{i \in \mathbb{N}}$, where $H_i \in \mathbb{R}^{n_i \times n_i}$, and $Y_N = \text{diag}(Y_i)_{i \in \mathbb{N}}$, where $Y_i \in \mathbb{R}^{r_i \times n_i}$ in such a way that it holds

$$\begin{bmatrix}
H_N & H_N A_N^T + Y_N^T B_N^T & H_N Q_N^{1/2} & Y_N^T R_N^{1/2} \\
A_N H_N + B_N Y_N & H_N & 0 & 0 \\
Q_N^{1/2} H_N & 0 & I & 0 \\
R_N^{1/2} Y_N & 0 & 0 & I
\end{bmatrix} \succeq 0, \quad (23)$$
then there is a matrix \( P_N = H_N^{-1} \) that satisfies (22), and a feedback control matrix \( K_N = Y_N H_N^{-1} \) that stabilizes the closed-loop system.

**Proof:** Following Maestre et al. (2011b), the condition (23) is derived from the recursive application of the Schur’s complement to

\[(A_N + B_N K_N)^	op P_N (A_N + B_N K_N) - P_N + Q_N + K_N R_N K_N \leq 0. \tag{24}\]

Secondly, it is proved that the closed-loop system is stable with the linear control law \( u_N = K_N x_N \). Pre- and post-multiplying (24) by \( x_N(k) \) and \( x_N(k) \), and then by \(-1\), it can be rewritten as

\[x_N(k)^	op P_N x_N(k) - x_N(k+1)^	op P_N x_N(k+1) \geq L_N(x_N(k)),\tag{25}\]

where

\[
x_N(k+1) = (A_N + B_N K_N)x_N(k),

l_N(x_N(k)) = x_N(k)^	op Q_N x_N(k) + x_N(k)^	op K_N R_N K_N x_N(k).
\]

Hence, \( V_N(x_N(k)) = x_N(k)^	op P_N x_N(k) \) is a Lyapunov function and stability is guaranteed. Moreover, by telescope summation it can be checked that (22) holds.

To design the controller for the overall system, LMI (23) is solved maximizing the trace of \( H_N \), which leads to the minimization of the trace of \( P_N = H_N^{-1} \), and, therefore, minimizing the cost-to-go. The resulting feedback matrix \( K_N = Y_N H_N^{-1} \) is the gain of the controller.

### 4.3. Stability of the MPC Controller

In the previous section, the stability of the system with the corresponding \( K_N \) has been proved. Here, it also proved when the MPC controller is used.

**Theorem 2.** Let a system be composed of \( N \) subsystems, and matrices \( K_N, P_N \) be calculated by (23), the MPC controller can ensure recursive feasibility and closed-loop system stability.

**Proof:** Given a feasible control sequence at time step \( k \)

\[U_N(k) = [u_N(k), u_N(k+1), \ldots, u_N(k+N_p-1)],\tag{26}\]

the constraints \( x_N(k) \in X_N, u_N(k) \in U_N, \) and \( x_N(k+N_p) \in \Omega_N \) are satisfied. Then, a feasible shifted sequence can be formed at time \( k+1 \)

\[U_N^*(k+1) = [u_N^*(k+1), u_N^*(k+2), \ldots, u_N^*(k+N_p)] = [u_N(k+1), u_N(k+2), \ldots, K x_N(k+N_p)]. \tag{27}\]

to guarantee that the terminal state of the controlled system remains in the invariant set \( \Omega_N \). Hence, recursive feasibility can be proved by applying this procedure recursively.

Furthermore, the overall cost decreases from time step \( k \) to \( k+1 \) if it holds that

\[J_N(x_N(k+1), U_N^*(k+1)) \leq J_N(x_N(k), U_N^*(k)).\tag{28}\]
Substituting (16) in (28), and considering definitions (26) and (27), we get
\[
\sum_{t=1}^{N_p} L_N(x_N(k+t), u_N^*(k+t)) + F_N(x_N(k + N_p + 1)) \leq \\
\sum_{t=0}^{N_p-1} L_N(x_N(k+t), u_N(k+t)) + F_N(x_N(k + N_p)).
\]
Removing common terms and rearranging, we have
\[
L_N(x_N(k + N_p), u_N^*(k + N_p)) + F_N(x_N(k + N_p + 1)) - F_N(x_N(k + N_p)) \leq L_N(x_N(k), u_N(k)).
\]
Note that the left side hand of (30) is lower than or equal to zero because it is imposed by design (25), and the right side hand is greater than or equal to zero since the stage cost is positive definite. Thus it is proved that (28) is satisfied using the shifted control sequence, and, consequently, the global cost decreases in time. Finally, note that the control algorithm includes condition (18) to ensure stability, and 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.

5. Case Study

This section shows a description of the eight-coupled tanks plant, which is based on the quadruple tank process (Johansson, 2000) and specifies the non-linear model and the parameters used in simulation. Furthermore, the criteria for the fuzzy negotiation 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: four upper tanks (3, 4, 7 and 8) discharge flows into the four lower ones (1, 2, 5 and 6), and these in turn into the sink tanks, as displayed in Fig. 7. Four pumps control the plant, and there are also six three-way valves \( \gamma_v \), with \( v \in \{1, 2, \ldots, 6\} \) manually operated that divide the input flows into two ways.

The system is divided into \( N = 4 \) subsystems as follows: tanks 1 and 3 belong to subsystem 1; tanks 2 and 4 to 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 (\( h_1, h_2, h_5, h_6 \)) and four inputs (\( q_a, q_b, q_c \) and \( q_d \)).

5.2. Plant and Control Model

The non-linear model of the plant can be obtained on applying mass balances and Bernoulli’s law. The following differential equations describe the system:
where \( h_n \) is the water level of tank \( n \in \{1, 2, \ldots, 8\} \), \( S_n \) is the corresponding cross-section, and \( a_n \) represents the cross-section of the outlet pipe. The parameter \( \gamma_v \in [0, 1] \) with \( v \in \{1, 2, \ldots, 6\} \) refers to the opening of the three-way valves; gravity is denoted by \( g \); and \( q_m \) corresponds to the flow pumped by pump \( m \in \{a, b, c, d\} \).
Table 1: Plant parameters and operating point

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_n$</td>
<td>$13.89 \cdot 10^{-3}$</td>
<td>Cross-section of tank $n \in {1, 2, \ldots, 8}$ (m²)</td>
</tr>
<tr>
<td>$a_n$</td>
<td>$50.265 \cdot 10^{-6}$</td>
<td>Cross-section of outlet pipe $n \in {1, 2, \ldots, 8}$ (m²)</td>
</tr>
<tr>
<td>$g$</td>
<td>9.81</td>
<td>Acceleration of gravity (m/s²)</td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>0.3</td>
<td>Opening parameter of the three-way valve 1</td>
</tr>
<tr>
<td>$\gamma_2$</td>
<td>0.4</td>
<td>Opening parameter of the three-way valve 2</td>
</tr>
<tr>
<td>$\gamma_3$</td>
<td>0.1</td>
<td>Opening parameter of the three-way valve 3</td>
</tr>
<tr>
<td>$\gamma_4$</td>
<td>0.3</td>
<td>Opening parameter of the three-way valve 4</td>
</tr>
<tr>
<td>$\gamma_5$</td>
<td>0.4</td>
<td>Opening parameter of the three-way valve 5</td>
</tr>
<tr>
<td>$\gamma_6$</td>
<td>0.1</td>
<td>Opening parameter of the three-way valve 6</td>
</tr>
<tr>
<td>$h_1^0$</td>
<td>0.10</td>
<td>Steady-state level of the tank 1 (m)</td>
</tr>
<tr>
<td>$h_2^0$</td>
<td>0.15</td>
<td>Steady-state level of the tank 2 (m)</td>
</tr>
<tr>
<td>$h_3^0$</td>
<td>0.07</td>
<td>Steady-state level of the tank 3 (m)</td>
</tr>
<tr>
<td>$h_4^0$</td>
<td>0.03</td>
<td>Steady-state level of the tank 4 (m)</td>
</tr>
<tr>
<td>$h_5^0$</td>
<td>0.10</td>
<td>Steady-state level of the tank 5 (m)</td>
</tr>
<tr>
<td>$h_6^0$</td>
<td>0.15</td>
<td>Steady-state level of the tank 6 (m)</td>
</tr>
<tr>
<td>$h_7^0$</td>
<td>0.025</td>
<td>Steady-state level of the tank 7 (m)</td>
</tr>
<tr>
<td>$h_8^0$</td>
<td>0.10</td>
<td>Steady-state level of the tank 8 (m)</td>
</tr>
<tr>
<td>$q_{a}^0$</td>
<td>0.142</td>
<td>Steady-state of $q_a$ (m³/h)</td>
</tr>
<tr>
<td>$q_{b}^0$</td>
<td>0.421</td>
<td>Steady-state of $q_b$ (m³/h)</td>
</tr>
<tr>
<td>$q_{c}^0$</td>
<td>0.424</td>
<td>Steady-state of $q_c$ (m³/h)</td>
</tr>
<tr>
<td>$q_{d}^0$</td>
<td>0.140</td>
<td>Steady-state of $q_d$ (m³/h)</td>
</tr>
</tbody>
</table>

Given an operating point $h_n^0$ with $n \in \{1, 2, \ldots, 8\}$ and $q_m^0$ with $m \in \{a, b, c, d\}$, a linear state-space control model can be expressed as

$$\bar{x}_N(k + 1) = A_N\bar{x}_N(k) + B_N\bar{u}_N(k),$$  \hspace{1cm} (32)

where the state vector becomes $\bar{x}_N = [h_1(k) - h_1^0, \ldots, h_8(k) - h_8^0]^\top$; the input vector $\bar{u}_N = [q_a(k) - q_{a}^0, \ldots, q_d(k) - q_{d}^0]^\top$; and $A_N, B_N$ are the corresponding matrices of the global system. Similarly, the dynamics of each subsystem $i \in \{1, 2, 3, 4\}$ are given by

$$\bar{x}_i(k + 1) = A_i\bar{x}_i(k) + B_i\bar{u}_i(k) + \bar{w}_i(k),$$
$$\bar{w}_i(k) = \sum_{j \in \mathcal{N}_i} B_{ij}\bar{u}_j(k),$$  \hspace{1cm} (33)

where $\bar{w}_i(k)$ represents the coupling with its neighbours $j \in \mathcal{N}_i$. The operating point and the rest of the plant parameters are displayed in Table 1. As a result, the following
subsystem matrices are obtained

\[
A_1 = \begin{bmatrix} 0.8810 & 0.1325 \\ 0 & 0.8587 \end{bmatrix}, \quad B_{11} = \begin{bmatrix} 0.0282 \\ 0 \end{bmatrix}, \quad B_{12} = \begin{bmatrix} 0.0035 \\ 0.0464 \end{bmatrix}, \quad B_{13} = \begin{bmatrix} 0.0380 \\ 0 \end{bmatrix}, \quad B_{14} = \begin{bmatrix} 0.0010 \\ 0.0089 \end{bmatrix},
\]

\[
A_2 = \begin{bmatrix} 0.9017 & 0.1922 \\ 0 & 0.7973 \end{bmatrix}, \quad B_{21} = \begin{bmatrix} 0.0071 \\ 0.0626 \end{bmatrix}, \quad B_{22} = \begin{bmatrix} 0.0380 \\ 0 \end{bmatrix}, \quad B_{23} = \begin{bmatrix} 0.0010 \\ 0.0089 \end{bmatrix}, \quad B_{24} = \begin{bmatrix} 0.0067 \\ 0.0530 \end{bmatrix},
\]

\[
A_3 = \begin{bmatrix} 0.8810 & 0.2104 \\ 0 & 0.7753 \end{bmatrix}, \quad B_{31} = \begin{bmatrix} 0.0011 \\ 0.0088 \end{bmatrix}, \quad B_{32} = \begin{bmatrix} 0.0282 \\ 0 \end{bmatrix}, \quad B_{33} = \begin{bmatrix} 0.0011 \\ 0.0088 \end{bmatrix}, \quad B_{34} = \begin{bmatrix} 0.0067 \\ 0.0530 \end{bmatrix},
\]

\[
A_4 = \begin{bmatrix} 0.9017 & 0.1126 \\ 0 & 0.8814 \end{bmatrix}, \quad B_{41} = \begin{bmatrix} 0.0035 \\ 0.0564 \end{bmatrix}, \quad B_{42} = \begin{bmatrix} 0.0380 \\ 0 \end{bmatrix}, \quad B_{43} = \begin{bmatrix} 0.0035 \\ 0.0564 \end{bmatrix}, \quad B_{44} = \begin{bmatrix} 0.0067 \\ 0.0530 \end{bmatrix},
\]

\[
A_{ij} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad i \neq j \quad \forall i, j \in \mathcal{N}.
\]

Moreover, the states and inputs are constrained by

\[-h_n^0 < \tilde{x}_n(k) \leq 0.08, \quad -q_m^0 < \tilde{u}_n(k) \leq 0.4, \quad \forall n \in \{1, 2, \ldots, 8\}, \quad \forall m \in \{a, b, c, d\}.
\]

The matrices \(Q_n = \text{diag}(Q_i)_{i \in \mathcal{N}} \in \mathbb{R}^{n \times q_i}\) and \(R_n = \text{diag}(R_i)_{i \in \mathcal{N}} \in \mathbb{R}^{r_i \times r_i}\) are respectively the corresponding positive semi-definite and positive-definite constant weighting matrices set as

\[
Q_i = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \forall i \in \{1, 2, 3, 4\}, \quad R_i^1 = \begin{cases} 0.002 & \text{if } i = 1 \\ 0.20 & \text{if } i \in \{2, 3, 4\} \end{cases}.
\]

Furthermore, the resulting local feedback gain \(K_i\) and weighting matrix of the terminal cost \(P_i\), which have been calculated maximizing the trace of \(H\) through Theorem 1, are

\[
K_1^\top = \begin{bmatrix} -0.0567 \\ 0.0913 \end{bmatrix}, \quad P_1 = \begin{bmatrix} 4.6685 & 2.3288 \\ 2.3288 & 2.4292 \end{bmatrix},
\]

\[
K_2^\top = \begin{bmatrix} -0.2863 \\ -0.2331 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 5.1779 & 3.0510 \\ 3.0510 & 3.0250 \end{bmatrix},
\]

\[
K_3^\top = \begin{bmatrix} -0.0575 \\ -0.0922 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 4.7185 & 2.8848 \\ 2.8848 & 3.0013 \end{bmatrix},
\]

\[
K_4^\top = \begin{bmatrix} -0.2319 \\ -0.1585 \end{bmatrix}, \quad P_4 = \begin{bmatrix} 5.2133 & 2.4217 \\ 2.4217 & 2.3883 \end{bmatrix}.
\]

Recall that to compute the RPI sets of each agent \(i\), a standard approach has been implemented with MPT of MATLAB (Herceg et al., 2013). In Fig. 8, the corresponding invariant sets \(\Omega_i\) of each subsystem \(i \in \mathcal{N}\) are shown.

\[1\text{The weighting parameter } R_1 \text{ is intentionally different to have one controller out of tune and assess its impact on the proposed scheme.}\]
5.3. Fuzzy Negotiation Specifications

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

1. **Fuzzification:** In this work, three performance indicators have been considered for the fuzzy negotiation procedure: residence time ($RT$) for water stored in the lower tanks, current water level ($h$), and pumping energy ($PE$). They represent typical constraints for the plant, and in this case, act as fuzzy criteria similarly as soft constraints.

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

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

     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 the alternatives of constraint satisfaction or violation are defined by the limits presented in Table 2.

   - **Current water level** 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 is considered in the fuzzy negotiation to improve process safety.
Due to the plant setup and the distributed control framework, only the levels of the lower tanks are evaluated, because the upper ones are only influenced by their neighbors’ control action. Three linguistic variables (‘low’, ‘medium’ and ‘high’) are considered for water levels.

- **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.,

\[
PE_m(k) = \frac{0.04}{3600N_p} \sum_{t=1}^{N_p} q_m(k + t). \tag{35}
\]

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

\[
PE(k) = PE_a(k) + PE_b(k) + PE_c(k) + PE_d(k), \tag{36}
\]

and the fuzzy sets for the three criteria are determined by the values of Table 2, representing the limits and vertices of triangles defining the membership functions.

<table>
<thead>
<tr>
<th>Limits of the fuzzy sets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pumping Energy ((PE_m))</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Water level</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Residence time ((RT_n))</strong></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Table 2: Limits of the fuzzy sets.

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

\[
\text{Rule } R_{ri}: \text{ If } RT_i(k + N_p) \text{ is } Tr_1; \ PE_i(k) \text{ is } Tr_2; \text{ and } h_n(k + N_p) \text{ is } Tr_3, \text{ then output } q_m \text{ is } Tr_4,
\]

where \(R_{ri}\) represents the \(r\)-th rule of subsystem \(i \in \{1, 2, 3, 4\}\); \(k\) denotes the sampling time; and \(Tr_1 = \{\text{‘high’, ‘low’}\}\), \(Tr_2 = \{\text{‘high’, ‘medium’, ‘low’}\}\), \(Tr_3 = \{\text{‘high’, ‘medium’, ‘low’}\}\), and \(Tr_4 = \{\text{‘good’, ‘acceptable’, ‘bad’}\}\) are respectively the linguistic variables that characterize the criteria \(RT_i, PE_i, h_n\) and \(q_m\).

Some examples of rules are the following:
3. Defuzzification: The defuzzification provides the fitness of a control sequence for each rule $R_{ri}$. The fitness according to a rule $R_{ri}$ is given by

$$\alpha_{R_{ri}}(q_i) = w_{RT}^{R_{ri}} \cdot \mu_{Tr_1}(RT_i) \cdot w_{PE}^{R_{ri}} \cdot \mu_{Tr_2}(PE_i) \cdot w_h^{R_{ri}} \cdot \mu_{Tr_3}(h_i),$$

where $\mu_{Tr_1}(RT_i)$, $\mu_{Tr_2}(PE_i)$, $\mu_{Tr_3}(h_i)$ are the fuzzy sets for each criterion considered in rule $r$, and $w_{RT}^{R_{ri}}$, $w_{PE}^{R_{ri}}$, $w_h^{R_{ri}}$ are the considered weights that depend on the linguistic variable included in the rule.

In this work, the weights are independent of the rule and the subsystem where they are applied. Their values only depend on the particular fuzzy sets that they are associated with. Then, weights $w^{RT}$, $w^{PE}$, and $w^h$ are expressed as

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

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

Hence, the total fitness of the control signal for the full set of $N_r$ rules is given by

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

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

6. Results

In this section, results of the proposed controller are presented. The sampling time used in simulations is $T_s = 5 \text{ s}$, and the MPC controllers consider a prediction horizon $N_p = 20$. To show the benefits of the proposed DMPC with fuzzy negotiation, a comparison with a centralized MPC and a DMPC using a cooperative game as proposed by Maestre et al. (2011a) has been performed, considering the same tuning parameters for the local controllers. The influence of the fuzzy rules is also evaluated. In Table 3, the weights considered for each performance indicator in the fuzzy rules are shown.

In Table 4, Fig. 9, and Fig. 10, it is displayed that centralized MPC is always the controller with the smallest integral squared error ($ISE$) for tracking in comparison to the other proposed controllers. This fact is obviously caused due to the availability of full information of the plant 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 cooperative game because the fuzzy weights favor small PE, although still higher than the centralized MPC (see Fig. 9 (b) and Fig. 10 (b)). Note that, 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 value of the weights of the considered fuzzy rules, giving the control system designer freedom to promote different behaviors. In this regard, the fuzzy approach is more flexible because it simplifies the tuning of the controller to attain better key performance indicators (KPIs). As for the standard MPC approaches, their 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 effect of the tuning of the fuzzy rules in our approach.

<table>
<thead>
<tr>
<th>Weights</th>
<th>Alternative</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
<th>Case 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Residence time (RT)</td>
<td>Low ($w_1$)</td>
<td>0.01</td>
<td>1</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>High ($w_2$)</td>
<td>1</td>
<td>0.01</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Water Level</td>
<td>Low ($w_3$)</td>
<td>0.01</td>
<td>0.01</td>
<td>1</td>
<td>0.02</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Medium ($w_4$)</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>High ($w_5$)</td>
<td>1</td>
<td>1</td>
<td>0.02</td>
<td>1</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>Pumping Energy (PE)</td>
<td>Low ($w_6$)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.02</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>Medium ($w_7$)</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>High ($w_8$)</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.9</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Table 3: Weights for several criteria.

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

Table 4: Comparison of ISE and PE for the considered control techniques.
Figure 9: Comparison of tanks level evolution (solid line: Centralized MPC, dashed line: Fuzzy DMPC; dotted line: DMPC cooperative game).
In the following set of results, the influence of fuzzy rules is evaluated (Table 5). In Fig. 13, the $RT$ in lower tanks is presented for two cases with different fuzzy-rule weights (Case 1 and Case 2). In Figs. 11 and 12, 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.

In Table 5, some $ISE$ and $PE$ results are shown, and the effect of weights is noticeable.
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. The comparison of cases 3 and 4 (Fig. 14) gives lower levels $h_1$, $h_2$, $h_5$, and $h_6$ in Case 3 than for 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 (see Table 2), this is particularly noticeable in $h_2$, which is maintained around this value.

<table>
<thead>
<tr>
<th></th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
<th>Case 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total $ISE$</td>
<td>0.7456</td>
<td>0.7308</td>
<td>0.7364</td>
<td>0.7456</td>
<td>0.7913</td>
<td>0.7335</td>
</tr>
<tr>
<td>Total $PE$</td>
<td>$1.603 \cdot 10^{-2}$</td>
<td>$1.596 \cdot 10^{-2}$</td>
<td>$1.597 \cdot 10^{-2}$</td>
<td>$1.603 \cdot 10^{-2}$</td>
<td>$1.605 \cdot 10^{-2}$</td>
<td>$1.597 \cdot 10^{-2}$</td>
</tr>
</tbody>
</table>

Table 5: Comparison of $ISE$ and $PE$ for the considered cases.

![Figure 11: Water levels comparison for Case 1 (dashed line) and Case 2 (solid line).](image-url)
Figure 12: Pumping flow rates comparison for Case 1 (dashed line) and Case 2 (solid line).

Figure 13: Residence time comparison for Case 1 (dashed line) and Case 2 (solid line).
Finally, the performed experiments show that in all cases, the cost of centralized MPC is the lowest, 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. 16, the values of the overall Lyapunov function are presented for both Case 5 and Case 6.

![Overall Lyapunov function](image)

Figure 16: Overall Lyapunov function for Case 5 (solid line) and Case 6 (dashed line).

7. Conclusions

A DMPC 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. The methodology can be extended to other complex plants with any number of agents because the lower layer negotiation avoids a combinatorial explosion. Furthermore, the proposed approach provides a more intuitive tuning of the controller to achieve target values regarding the considered performance indicators.

Finally, a linear feedback controller is considered as a backup controller to assure stability in the case that the fuzzy inference increases the cost. In the results obtained, the use of the control action provided by fuzzy inference is mostly used in both layers, showing the usefulness of the methodology.

Acknowledgements

The authors gratefully acknowledge the financial support by the European Research Council [Advanced Research Grant, project number 769051 OCONTSOLAR], the Spanish Ministry of Economy [project numbers DPI2017-86918-R and DPI2015-67341-C02-01] and the Spanish Ministry of Science, Innovation, and Universities [grant number FPU18/04476].
References


