

Distributed Generalized Predictive Control with hierarchical structure based on an adaptive mechanism layer

Mohsen Erfani Hajipour

Ferdowsi University of Mashhad

Heydar Toossian Shandiz

htoosian@um.ac.ir

Ferdowsi University of Mashhad

Research Article

Keywords: Hierarchical Distributed Model Predictive Control, Generalized Predictive Control, Decision Space, Adaptive Mechanism Layer, Dynamic Game Theory, Quadruple Tank System

Posted Date: October 15th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11110627/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Distributed Generalized Predictive Control with hierarchical structure based on an adaptive mechanism layer

Mohsen Erfani Haji Pour¹, Heydar Toossian Shandiz^{1*}

Ferdowsi University of Mashhad, Engineering Faculty, Mashhad, Iran.

Abstract:

In this paper, a novel hierarchical distributed model predictive control (HDMPC) framework is proposed to improve the control performance of systems with input coupling under varying operating conditions. The proposed architecture consists of two layers. In the first layer, a distributed generalized predictive control (DGPC) scheme is developed based on the controller autoregressive integrated moving average (CARIMA) model to explicitly model and compensate for the effects of interactions arising from coupling among subsystems. This layer provides a decision space for each subsystem, including three strategies: conservative, balanced, and aggressive, by considering different values of the control effort penalty coefficient. In the second layer, non-cooperative dynamic game theory is used so that each subsystem, at each time step, chooses the most appropriate control strategy from the options in the decision space in a rational and adaptive manner according to the operating conditions and by analyzing the decisions of the other subsystems. This adaptive mechanism (AM) enables the system to dynamically respond to environmental and operational conditions and achieve a stable equilibrium between performance accuracy and control effort. The efficiency of the proposed framework is evaluated on a Quadruple-Tank benchmark system. Simulation results show that the proposed approach provides superior or equivalent performance in terms of control quality compared to the reference approaches. Additionally, the results indicate that the proposed method has good robustness and

adaptability to parametric uncertainties, including variations in the valve flow distribution parameters.

Keywords: Hierarchical Distributed Model Predictive Control, Generalized Predictive Control, Decision Space, Adaptive Mechanism Layer, Dynamic Game Theory, Quadruple Tank System.

1. Introduction

Centralized controllers have long been a common and effective approach to controlling various systems, remaining reliable in many industrial systems due to their simple implementation and ability to manage interactions arising from subsystem coupling. However, in complex systems with multiple interconnected subsystems, a centralized architecture can be difficult to implement given the model's large scale, subsystems' geographical dispersion, and the vulnerability of a single point of failure. Consequently, distributed control has increasingly been proposed as an alternative or complement to centralized control, enhancing management and coordination by distributing control tasks among local controllers that decide based on their own models while exchanging limited information with one another [1].

Among the control strategies available within both centralized and distributed architectures, model predictive control (MPC) is widely regarded as one of the most powerful control frameworks. By determining control actions over a prediction horizon based on system dynamics, MPC enables explicit handling of input and output constraints, model uncertainties, modeling errors, and external disturbances [2-5]. To improve MPC performance in large-scale systems, distributed model predictive control (DMPC) architectures have been developed, in which control decisions are made at the subsystem level, improving flexibility, scalability, and reliability. Numerous DMPC approaches have since been proposed for a wide range of objectives and applications. The

following section reviews representative DMPC approaches to identify research gaps and motivate the method proposed in this paper.

Several review studies have systematically examined the structural characteristics of DMPCs. References [6] and [7] review DMPC architectures as cooperative, iterative, and hierarchical structures, analyzing their stability, computational complexity, and communication requirements, and identifying challenges such as network reconfiguration and distributed state estimation. Reference [8] similarly reviews DMPC methods and proposes an analytical framework for evaluating stability guarantees, disturbance robustness, and communication complexity. Reference [9] compares several DMPC algorithms for the Quadruple-Tank system in terms of settling time and communication requirements. Together, these studies offer a structured overview of DMPC methodologies. A substantial body of research has focused on DMPC algorithms under practical conditions and for specific applications. Reference [10] proposes a DMPC method under data-rate constraints and dynamic quantization, relevant to systems with limited communication resources. Reference [11] introduces a separable optimization algorithm that facilitates local computation and enhances robustness to communication delays without requiring a central coordinator. Reference [12] presents a DMPC framework for dynamic consensus in multiagent systems that uses the exponentially weighted Laguerre function to reduce computational complexity. Furthermore, an embedded integrator model removes the need for a separate disturbance model, improves closed-loop stability, and yields superior disturbance rejection, validated on a cluster of ground-based vehicles. Reference [13] investigates cybersecurity vulnerabilities in DMPC systems, modeling spoofing and disruption attacks with numerical case studies. Reference [14] develops a tube-based DMPC strategy for wheeled mobile robots within a leader-follower architecture. Reference [15] combines parametric system identification with backpropagation neural networks to learn the

system model for DMPC design. Although this method is computationally demanding, it suits systems lacking a detailed model. Reference [16] proposes a self-triggered DMPC mechanism that simultaneously optimizes triggering instants and control inputs, significantly reducing communication. Despite the important applications and distinctive features of these approaches, their applicability is often constrained by their underlying assumptions, limiting their effectiveness beyond the scenarios for which they were developed. Another prominent direction has focused on HDMPC frameworks that manage subsystems through layers of supervisory, negotiation, or decision-making. Reference [1] proposes an HDMPC framework based on fuzzy negotiation. In this approach, local predictive controllers are coordinated through an upper-level fuzzy inference mechanism, in which subsystem interactions are managed using predefined fuzzy rules. This coordination layer provides a negotiation mechanism that adapts control actions based on interactions among subsystems and the overall operating state of the process. The reported results demonstrate tracking performance comparable to that of centralized MPC while improving coordination efficiency and reducing communication requirements. However, its effectiveness depends heavily on the offline design of fuzzy rules and membership functions, and satisfactory performance under varying conditions may require extensive tuning. Reference [17] instead uses a reinforcement learning trained deep neural network as the negotiation agent, eliminating the need for explicit process models during negotiation and enabling optimization across a wide range of performance indices and constraints. Nevertheless, the approach relies on training process and careful reward function design. Game-theoretic concepts have also been incorporated into DMPC to address subsystem interactions. Reference [18] develops a robust DMPC method for polytopic uncertainty using a distributed Kalman filter, while Reference [19] proposes a Laguerre-function-based DMPC strategy reducing optimization problem dimensionality; both approaches use a Nash

equilibrium-based coordination mechanism in which subsystems share their local optimal solutions with their neighbors at each sampling instant and re-solve their optimization problem, which includes the effect of interactions, until the solutions converge. This approach generally outperforms decentralized approaches that ignore interactions. However, convergence within the sampling period cannot always be guaranteed, limiting applicability in fast dynamic systems. Reference [20] presents a cooperative game-based DMPC approach in which local controllers coordinate through limited information exchange. The game process enables agents to jointly minimize a global cost while maintaining closed-loop stability. The method reduces computational complexity compared with centralized MPC. However, its coordination strategy becomes increasingly complex as the number of interacting agents grows.

Despite progress in developing DMPC architectures, two fundamental challenges remain in improving the efficiency, robustness, and applicability of this control paradigm under varying operating conditions. These two key issues have not been systematically and jointly addressed in the existing literature. The first challenge lies in accurately modeling subsystems and designing predictive controllers that mitigate uncertainties and disturbances while explicitly accounting for interactions from subsystem coupling such as input-type interactions. Neglecting or oversimplifying these interactions can substantially alter dynamic behavior, degrade performance, and even destabilize the system, particularly when their magnitude or structure varies over time; explicitly modeling them is therefore essential for effective DMPC design. The second challenge concerns the intelligent and optimal use of control parameters, particularly the control effort penalty coefficient, incorporated into the MPC objective function to limit effort and prevent impractical signals. This coefficient directly shapes response speed, tracking performance, and the intensity of subsystem interactions, producing a classical trade-off. Increasing it smooths control

signals and reduces control effort but slows response and may raise steady-state error, while decreasing it improves tracking but risks excessive control effort and oscillations. This trade-off becomes even more challenging in distributed systems with input-type interactions because the control input is not merely a local variable; it is one of the primary channels through which interactions are transmitted among subsystems, thereby significantly influencing overall system performance. Despite these sensitivities, in most conventional designs, this coefficient is treated as a specific, predetermined value. This simplifies design but lacks flexibility for real systems with varying conditions. A specific value may be overly conservative in some conditions and degrade response quality. In contrast, under other conditions, it may generate extreme control signals that induce instability and disrupt other subsystems. Desired performance across the operating range therefore cannot be guaranteed by a specific coefficient. Therefore, the primary challenge is not merely tuning a control parameter's numerical value, but developing a mechanism that allows subsystems to adapt dynamically to conditions. Such a mechanism can define a limited set of control options, each corresponding to a distinct level of control effort and dynamic response. Consequently, the system can make an informed and coordinated selection among these predefined options at each time step. Adaptability is thus achieved not through continuous parameter variation but through selecting control sequences from a predefined decision space. This approach enables the system to overcome traditional trade-offs while simultaneously achieving high tracking accuracy and smooth control actions with minimal cost.

Addressing these two challenges simultaneously provides a pathway for improving the DMPC structure and enhancing its adaptability to varying operating conditions. Accordingly, this paper proposes a novel HDMPC framework, comprising a two-layer architecture, that simultaneously addresses subsystem interactions and the need to adapt control behavior to changing operating

conditions. In the first layer, DGPC controllers are designed to generate a decision space using several control sequences with different control effort penalty coefficients and different levels of conservatism and aggressiveness. These sequences are generated by considering input and output constraints, as well as input interactions. generalized predictive controller (GPC) is selected for this layer for its ability to reduce steady-state errors, explicitly handle input and output constraints, and provide an analytical structure suited for distributed architectures [21,22]. Since the proposed controller's performance depends on model accuracy, the CARIMA model employed in GPC is developed in this study by treating input-type interactions as measurable disturbances. This modeling enables accurate prediction at the local level. In the second layer, an AM is developed based on dynamic non-cooperative game theory, considering each subsystem as a player. By exploiting the decision space generated in the first layer, this mechanism enables the rational, coordinated selection of an appropriate control strategy for each subsystem, while accounting for the behavior and decisions of other subsystems at each time step. The non-cooperative nature of the game indicates that each subsystem pursues its own local objective and coordinates with other subsystems indirectly through the game structure, without forming explicit coalitions. The dynamic nature of the game implies that the decision-making process is carried out sequentially by the players to establish a stable equilibrium among the subsystems. In this context, equilibrium refers to a condition in which no subsystem can improve its control performance by unilaterally changing its selected option, assuming rational responses from the other subsystems. This equilibrium represents a stable selection of control strategies that promotes coordinated subsystem behavior over time [23]. While most previous studies on DMPC have primarily employed game theory in the form of static or cooperative games [19,20], the application of dynamic non-cooperative games within a hierarchical structure, based on decision spaces generated by local

controllers, remains underexplored in the literature. The use of dynamic game structures and the solution of the game through a backward induction approach enables the management of computational complexity, even in systems comprising a large number of subsystems. This capability is difficult to achieve in static games due to the combinatorial growth of the strategy space.

This research therefore aims to design and evaluate an HDMPC method in which input-type interactions are managed through an AM, improving system performance under varying operating conditions. The effectiveness of the proposed approach is validated through simulations of a Quadruple-Tank system. This benchmark system is widely recognized as a standard platform for evaluating and comparing advanced control techniques. Previous studies employing this system have investigated various approaches, including nonlinear MPC with an extended Kalman filter [24], linear MPC based on Takagi-Sugeno fuzzy models and neural networks [25], DMPC approaches [9,26], decentralized robust MPC [27], robust state-feedback control [28], and stabilization of delayed systems [29].

The remainder of this paper is organized as follows. The second section presents the subsystem modeling using the CARIMA model. The third section describes the proposed control structure, including the DGPC controllers and the AM. The fourth section is devoted to the simulation and performance evaluation of the proposed method for the control of the Quadruple-Tank process. Finally, the fifth section presents the conclusions and suggestions for future research.

2. System Modeling

One of the key steps in the design of distributed controllers is the accurate modeling of subsystems. Among the available modeling approaches, the Controller Auto-Regressive Moving-Average (CARMA) model is one of the most widely used methods for representing linearized systems. In this approach, the nonlinear system is linearized about a specific operating point and is expressed as (1).

$$\begin{aligned}
A(z^{-1})y(k) &= z^{-d}B(z^{-1})u(k-1) + C(z^{-1})e(k) \\
A(z^{-1}) &= 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{n_a}z^{-n_a} \\
B(z^{-1}) &= b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_{n_b}z^{-n_b} \\
C(z^{-1}) &= 1 + c_1z^{-1} + c_2z^{-2} + \dots + c_{n_c}z^{-n_c}
\end{aligned} \tag{1}$$

In these equations, $y(k)$ and $u(k)$ denote the system output and input, respectively; d represents the time delay; $e(k)$ is a zero-mean white noise; and $A(z^{-1})$, $B(z^{-1})$, and $C(z^{-1})$ are polynomials in the backward shift operator [30].

One of the fundamental challenges in the design of distributed controllers is accounting for the mutual influences and interactions among subsystems [31]. This consideration is particularly important when individual subsystems must interact and coordinate to achieve the overall system objective. Accordingly, the CARMA model for each subsystem is formulated as (2), where interactions among subsystems are modeled as measurable disturbances via additional terms in the model equations.

$$\begin{aligned}
A_x(z^{-1})y_x(k) &= z^{-d}B_x(z^{-1})u_x(k-1) + W_x(k) + C_x(z^{-1})e_x(k) \\
W_x(k) &= \sum_{y=1}^M z^{-d}B_y(z^{-1})u_y(k-1)
\end{aligned} \tag{2}$$

In these equations, x denotes the current subsystem, $W_x(k)$ represents the effects of the subsystems interacting with subsystem x , and M denotes the total number of subsystems. In this context, if subsystems x and y interact with each other, then $B_y(z^{-1}) \neq 0$; otherwise, $B_y(z^{-1}) = 0$.

Furthermore, $W_x(k) \in \mathcal{W}_x$, where $\mathcal{W}_x$ is a convex set containing the origin. Specifically, when the inputs of the interacting subsystems are subject to constraints, $\mathcal{W}_x$ defines the feasible set of $W_x(k)$ that is consistent with those input constraints.

In many industrial processes, the system disturbances are not static, and their mean and variance may change over time. The CARMA model is only suitable for systems affected by static disturbances and may exhibit degraded performance when disturbances contain long-term variations. To address this limitation, an integral operator is incorporated into the model, enabling it to more effectively identify and compensate for such disturbances and acting as a high-pass filter [32]. Consequently, the CARIMA model for each subsystem is defined as (3).

$$\begin{aligned} A_x(z^{-1})y_x(k) &= z^{-d}B_x(z^{-1})u_x(k-1) + W_x(k) + C_x(z^{-1})\frac{e_x(k)}{\Delta} \\ W_x(k) &= \sum_{y=1}^M z^{-d}B_y(z^{-1})u_y(k-1) \end{aligned} \quad (3)$$

Where $\Delta = 1 - z^{-1}$ denotes the integral operator. In many practical applications, the assumption $C_x(z^{-1}) = 1$ is adopted to simplify the system identification process and to facilitate the estimation of the parameters $A_x(z^{-1})$, $B_x(z^{-1})$ and $B_y(z^{-1})$.

3. Proposed Control Architecture

In this section, a hierarchical distributed control strategy is proposed. Based on the subsystem model and the analytical framework developed in the previous section, GPC is adopted as the main framework for controller design. The proposed control architecture consists of a two-layer structure. In the first layer, DGPC controllers establish an appropriate decision space for subsystems by using control effort penalty coefficients. In the second layer, using this decision space and dynamic game theory, an AM is implemented to ensure optimal system performance under various operating conditions.

Figure 1 illustrates the block diagram of the control system for subsystem x.

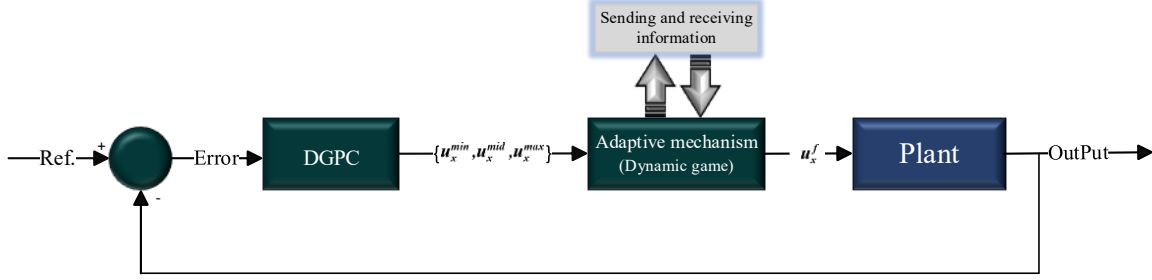


Figure 1. Block diagram of the control system of subsystem x

Remark 1: To simplify the presentation and facilitate understanding of the proposed method, the controller design is considered for a system consisting of two subsystems in this paper. It should be noted that the presented formulation can be readily extended to systems with an arbitrary number of subsystems.

3.1. First Layer of Control

This section presents the structure of the first control layer. This layer provides the necessary decision-making space for the AM layer by implementing DGPC controllers.

3.1.1. DGPC Controller

Consider the CARIMA model of subsystem x, as given by (4).

$$A_x(z^{-1})y_x(k) = B_x(z^{-1})u_x(k-d-1) + B_y(z^{-1})u_y(k-d-1) + \frac{e_x(k)}{\Delta} \quad (4)$$

To design a predictive controller, the predictive model must first be obtained. In this regard, the predictive model can be derived using the Diophantine equation. The Diophantine equation is defined as (5).

$$1 = E_{xj}(z^{-1})\tilde{A}_x(z^{-1}) + z^{-j}F_{xj}(z^{-1}) \quad (5)$$

Where $\tilde{A}_x(z^{-1}) = \Delta A_x(z^{-1})$. By multiplying both sides of (4) by $\Delta E_{xj}(z^{-1})z^j$ and applying the Diophantine equation, (6) is obtained.

$$\begin{aligned} E_{xj}(z^{-1})\tilde{A}_x(z^{-1})y_x(k+j) &= E_{xj}(z^{-1})B_x(z^{-1})\Delta u_x(k+j-d-1) \\ &\quad + E_{xj}(z^{-1})B_y(z^{-1})\Delta u_y(k+j-d-1) + E_{xj}(z^{-1})e_x(k+j) \end{aligned} \quad (6)$$

According to the Diophantine equation, equation (7) can be expressed as follows:

$$\begin{aligned} y_x(k+j) &= F_{xj}(z^{-1})y_x(k) + E_{xj}(z^{-1})B_x(z^{-1})\Delta u_x(k+j-d-1) \\ &\quad + E_{xj}(z^{-1})B_y(z^{-1})\Delta u_y(k+j-d-1) + E_{xj}(z^{-1})e_x(k+j) \end{aligned} \quad (7)$$

Consequently, the best prediction for $y_x(k+j)$ is obtained using (8) and (9).

$$\begin{aligned} \hat{y}_x(k+j) &= E\{y_x(k+j)\} = E\{F_{xj}(z^{-1})y_x(k) + E_{xj}(z^{-1})B_x(z^{-1})\Delta u_x(k+j-d-1) \\ &\quad + E_{xj}(z^{-1})B_y(z^{-1})\Delta u_y(k+j-d-1)\} + E\{E_{xj}(z^{-1})e_x(k+j)\} \end{aligned} \quad (8)$$

In this equation, $E\{E_{xj}(z^{-1})e_x(k+j)\} = 0$. Consequently,

$$\begin{aligned} \hat{y}_x(k+j) &= F_{xj}(z^{-1})y_x(k) + E_{xj}(z^{-1})B_x(z^{-1})\Delta u_x(k+j-d-1) \\ &\quad + E_{xj}(z^{-1})B_y(z^{-1})\Delta u_y(k+j-d-1) \end{aligned} \quad (9)$$

By rewriting (9), (10) is obtained.

$$\hat{y}_x(k+j|k) = G_{xj}(z^{-1})\Delta u_x(k+j-d-1) + G_{yj}(z^{-1})\Delta u_y(k+j-d-1) + F_{xj}(z^{-1})y_x(k) \quad (10)$$

Where $G_{xj}(z^{-1})$ and $G_{yj}(z^{-1})$ are defined in (11).

$$G_{sj}(z^{-1}) = E_{xj}(z^{-1})B_s(z^{-1}) = g_{s0} + g_{s1}z^{-1} + \dots + g_{sj}z^{-j}, \quad \forall s = x, y \quad (11)$$

Using Section 4.2 of reference [32] and the definition $j = d + 1, d + 2, \dots, d + N$, the prediction set for the prediction horizon N is generally defined as in (12).

$$\begin{aligned} \mathbf{y}_x &= \mathbf{G}_x \mathbf{u}_x + \mathbf{f}'_x \\ \mathbf{f}'_x &= \mathbf{G}_y \mathbf{u}_y + \mathbf{f}_x \end{aligned} \quad (12)$$

Where $\mathbf{y}_x = [\hat{y}_x(k + d + 1)|k], \dots, \hat{y}_x(k + d + N)|k]^T$, $\mathbf{u}_x = [\Delta u_x(k), \dots, \Delta u_x(k + N - 1)|k]^T$, $\mathbf{u}_y = [\Delta u_y(k), \dots, \Delta u_y(k + N - 1)|k]^T$ and $\mathbf{f}'_x$ is related to past values, or in other words, the free response and $\mathbf{f}_x$ is defined as in (13).

$$\mathbf{f}_x = \mathbf{G}'_x(z^{-1})\Delta u_x(k-1) + \mathbf{G}'_y(z^{-1})\Delta u_y(k-1) + \mathbf{F}_x(z^{-1})y_x(k) \quad (13)$$

The complete forms of the matrices in (12) and (13) are provided in Appendix A.

In the next step, the objective function is introduced. For each subsystem, the objective function is generally defined as in (14).

$$J_x = \sum_{i=n_{e_1}}^{n_{e_2}} (\hat{y}_x(k+i) - y_{rx}(k+i))^2 + \sum_{j=1}^{n_u} \lambda_{u_{xj}} \Delta u_x^2(k+j-1) \quad (14)$$

Where n_u is the control horizon (in this study, $n_u = N$), $n_{e_1} = d + 1$ and $n_{e_2} = d + N$ represent the lower and upper bounds of the prediction horizon.

This objective function considers the quadratic deviation between the reference signal and the predicted output signal over the prediction horizon and penalizes control effort quadratically. With information about the reference signal, the control error can be predicted and minimized [32,33]. Therefore, the controlled output signal is driven to its reference value sooner than when no prediction is made. This relationship can be rewritten as (15).

$$\begin{aligned} J_x &= (\mathbf{G}_x \mathbf{u}_x + \mathbf{f}'_x - \mathbf{y}_{rx})^T (\mathbf{G}_x \mathbf{u}_x + \mathbf{f}'_x - \mathbf{y}_{rx}) + \lambda_{u_x} \mathbf{u}_x^T \mathbf{u}_x \\ \mathbf{y}_{rx} &= [\mathbf{y}_{rx}(k+d+1) \quad \dots \quad \mathbf{y}_{rx}(k+d+N)]^T \end{aligned} \quad (15)$$

Where, the control effort penalty coefficients, $\lambda_{u_{xj}}$ are assumed to be constant over the entire prediction horizon.

Remark 2: The objective function J_x includes the terms of control effort of other subsystems at the current and future moments. Therefore, to minimize J_x , each subsystem uses the shifted sequence of its interacting subsystems. In other words, the subsystem assumes that other subsystems will continue their previous control sequence [26]. Consequently, in $\mathbf{f}'_x$, the shifted sequence $\mathbf{u}_y^s$ replaces the sequence $\mathbf{u}_y$.

$$\mathbf{u}_y^s = \begin{bmatrix} \Delta u_y^s(k) \\ \Delta u_y^s(k+1) \\ \vdots \\ \Delta u_y^s(k+N-1) \end{bmatrix} = \begin{bmatrix} \Delta u_y(k+1|k-1) \\ \Delta u_y(k+2|k-1) \\ \vdots \\ \Delta u_y(k+N-1|k-1) \\ \Delta u_y(k+N-1|k-1) \end{bmatrix} \quad (16)$$

Therefore, by minimizing the function J_x , while taking into account the system constraints, the optimal control effort for each subsystem is obtained as follows:

$$\mathbf{u}_x^* = \arg \min_{\mathbf{u}_x \in \mathcal{U}_x} J_x \quad (17)$$

where $\mathcal{U}_x$ is the admissible set defined by the physical and operational constraints of the system.

After determining the optimal control sequence, the first element of the sequence, $\Delta u_x^*(k)$, is used to calculate the control signal at time k , which is applied to the system as shown in (18).

$$u_x(k) = u_x(k+1) + \Delta u_x^*(k) \quad (18)$$

3.1.2. Determination of the Decision Space

As evident from the objective function, the second term is introduced to prevent excessive control effort and to avoid generating impractical control inputs, such as impulse signals. This term includes the control effort penalty coefficient, λ_x . In addition to its primary role, this parameter can significantly influence the system's performance and even its stability [21,34]. Increasing the value of λ_x imposes a greater penalty on control effort, resulting in a smoother and less aggressive control signal. However, this may slow the system response and increase the steady-state error. Conversely, reducing λ_x provides the controller with greater freedom to apply control effort. This can improve reference tracking performance but may also increase the likelihood of oscillations in both the control signal and the system output. In the proposed DGPC framework, where strong input-type interactions exist among subsystems, the importance of λ_x becomes even more pronounced. This is because variations in the control signal directly affect interactions among subsystems, influencing not only each subsystem's performance but also the overall behavior and coordination of the interconnected system.

Given these considerations, determining a specific value of λ_x that guarantees satisfactory performance across all operating conditions, including large tracking errors and steady-state operation, is challenging. In many industrial applications, there exists an admissible range of λ_x values within which the system remains stable while achieving the desired performance. This range is typically established by experts based on prior knowledge of the system, supplemented by experimental observations and simulation studies.

Within this permissible range, the system can exhibit three distinct types of control behavior:

1. **Aggressive behavior**, associated with low values of λ_x , which leads to the application of larger control efforts.
2. **Conservative behavior**, characterized by high values of λ_x , in which the control effort is lower.
3. **Balanced behavior**, located in the middle of the range, exhibiting characteristics between the two previous states.

Accordingly, the decision space can be defined by discrete values of λ_x within the expert-declared admissible range. Each value represents a specific, identifiable control strategy that can be employed based on the system's current operating conditions, such as periods of large tracking error or steady-state operation. Therefore, for each subsystem, the admissible range of the control effort penalty coefficient is defined as in (19).

$$\lambda_x \in [\lambda_{x\min}, \lambda_{x\max}] \quad (19)$$

Within this admissible range, three discrete values are chosen as the decision space, which is given in (20).

$$\Lambda_x \in \{\lambda_{x\min}, \lambda_{x\text{mid}}, \lambda_{x\max}\} \quad (20)$$

In this decision space, $\lambda_{x\min}$ represents the aggressive mode, $\lambda_{x\text{mid}}$ represents the balanced mode, and $\lambda_{x\max}$ represents the conservative mode. Based on the defined decision space, each subsystem computes the control sequences corresponding to all available decision options. This computation is performed under the assumption that the other subsystems maintain their shifted control sequences, $\mathbf{u}_y^s$, at the current time step, as described in Section 3.1.1. Consequently,

$$\lambda_{x\min} \rightarrow \mathbf{u}_x^{\min}, \lambda_{xmid} \rightarrow \mathbf{u}_x^{\text{mid}}, \lambda_{x\max} \rightarrow \mathbf{u}_x^{\max} \quad (21)$$

The first layer of control for each subsystem is summarized in Algorithm 1 (Step 1).

Algorithm 1 (Step 1): The first layer of control

1. Initialization:

- 1.1. The prediction horizon N and the initial conditions of the system are set.
- 1.2. The CARIMA model (A_x , B_x , B_y) is determined.
- 1.3. The prediction matrices are calculated using Diophantine equations (G_x , G_y and etc.) through equations (4) to (13).
- 1.4. The permissible interval (19) and decision space are determined (20).

2. At each time step k:

- 2.1. The free response $\mathbf{f}'_x$ is calculated (equation (12) and (13)).
- 2.2. The cost function J_x is minimized (equation (17)) for each value of λ_x in the decision space, and the control sequences corresponding to each value of λ_x are calculated:

$$\lambda_{x\min} \rightarrow \mathbf{u}_x^{\min}, \lambda_{xmid} \rightarrow \mathbf{u}_x^{\text{mid}}, \lambda_{x\max} \rightarrow \mathbf{u}_x^{\max}$$

3.2. Adaptive Mechanism Layer

As discussed previously, the parameter λ_x not only directly influences the performance of its corresponding subsystem but also significantly affects the performance of the subsystems interacting with it. Therefore, an adaptive and rational decision-making mechanism can be developed in which each subsystem selects a decision at every sampling instant while considering the behavior of the other related subsystems.

In this regard, the use of game theory, especially dynamic non-cooperative games, is considered a suitable approach. Within this framework, each subsystem can make decisions based on an analysis of the impact of its choice and those of other subsystems. In other words, a form of non-cooperative interaction emerges between the subsystems, in which each subsystem tries to make a rational yet adaptive decision.

3.2.1. How to Run a Dynamic Game

To run the game, a suitable cost function must first be defined. This function should be designed based on key performance criteria such as the reduction of tracking error and unnecessary control effort. Additionally, the interaction between subsystems must be considered, meaning that the control sequences resulting from the decisions of the interacting subsystems play an effective role in the calculation of the cost function. Accordingly, the proposed cost function is formulated as (22).

$$L_x(\mathbf{u}_x, \mathbf{u}_y) = (\mathbf{G}_x \mathbf{u}_x + \mathbf{f}'_x - \mathbf{y}_{rx})^T (\mathbf{G}_x \mathbf{u}_x + \mathbf{f}'_x - \mathbf{y}_{rx}) + \alpha \mathbf{u}_x^T \mathbf{u}_x \quad (22)$$

$$\mathbf{f}'_x = \mathbf{G}_y \mathbf{u}_y + \mathbf{f}_x$$

This function is employed to determine the decision criterion for each subsystem. The first term represents the cost associated with the tracking error between the predicted subsystem output and the reference trajectory over the prediction horizon at each time step. This term depends not only on the decision of the subsystem itself but also on the decisions of the subsystems interacting with it. The second term represents the cost associated with the control effort generated by the subsystem decision. The parameter $\alpha \geq 1$ is introduced to adjust the relative weighting between these two terms and to increase the sensitivity of the game to control effort when the tracking error is small. Clearly, a lower value of this cost function indicates a more desirable decision.

This function is calculated at each time step for all possible decision states of each subsystem. For example, for a system consisting of two subsystems, the following is obtained:

$$(L_x(\mathbf{u}_x^A, \mathbf{u}_y^B), L_y(\mathbf{u}_x^A, \mathbf{u}_y^B)), \forall A, B \in \{\min, \text{mid}, \max\} \quad (23)$$

To run a dynamic game, one of the subsystems is first selected as the game initiator. The game is then solved using the backward induction method. In this method, the subsystem at the end of the

decision-making chain (i.e., the last decision-maker) first calculates its best response (the optimal control sequence) for each possible state of the decisions of the other subsystems. Next, the preceding decision-maker, considering the optimal responses of the subsequent subsystem, selects its best strategy. This recursive process continues toward the game initiator. Finally, the optimal decision for each subsystem is determined at each time step [23,35]. The game-solving relationship is also expressed in (24).

$$\begin{aligned}
& \forall \mathbf{u}_x^A \in \{\mathbf{u}_x^{min}, \mathbf{u}_x^{mid}, \mathbf{u}_x^{max}\} \\
& \text{BR}_y(\mathbf{u}_x^A) = \arg \min_{\mathbf{u}_y^B \in \{\mathbf{u}_y^{min}, \mathbf{u}_y^{mid}, \mathbf{u}_y^{max}\}} L_y(\mathbf{u}_x^A, \mathbf{u}_y^B) \\
& \Rightarrow \mathbf{u}_x^{A*} = \arg \min_{\mathbf{u}_x^A \in \{\mathbf{u}_x^{min}, \mathbf{u}_x^{mid}, \mathbf{u}_x^{max}\}} L_x(\mathbf{u}_x^A, \text{BR}_y(\mathbf{u}_x^A)) \\
& \Rightarrow \mathbf{u}_y^{B*} = \text{BR}_y(\mathbf{u}_x^{A*})
\end{aligned} \tag{24}$$

Figure 2 illustrates an example of the dynamic game-solving process.

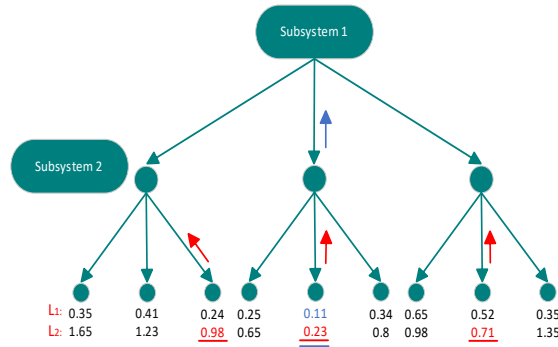


Figure 2. Example of solving a dynamic game

This procedure enables each subsystem to make a rational and adaptive decision while accounting for own decisions and those of other subsystems. Consequently, a dynamic equilibrium is established, through which the overall performance of the interconnected system is optimized.

3.2.2. Applying the Final Decision of the AM Layer

The final decisions determined by the subsystems are applied to the system as the final control effort sequences at each time step. The first element of each sequence is then used to calculate the control signal at time k , according to (18).

$$\mathbf{u}_x^f \rightarrow \Delta u_x(k), \mathbf{u}_y^f \rightarrow \Delta u_y(k) \quad (25)$$

In this Equation, $u_x^f = u_x^{A*}$ and $u_y^f = u_y^{B*}$ denote the final sequences of the control effort.

Remark 3: In some cases, the solution of the game may not yield a unique equilibrium, and several equilibrium points may exist. In such a situation, it is suggested that the decision corresponding to the balanced behavior $((u_x^{mid}, u_y^{mid}))$ be chosen as the final decision.

The AM layer is represented by Algorithm 1 (Step 2).

Algorithm 1 (Step 2): The AM layer

At each time step k :

1. The cost functions for each decision pair for each subsystem are calculated using Equations (22) and (23).
 2. The dynamic game is run (Equation (24)):
 - 2.1. If the game has a unique equilibrium, the game outcome is adopted and the final control effort sequences are applied: $\mathbf{u}_x^f = \mathbf{u}_x^{A*}$, $\mathbf{u}_y^f = \mathbf{u}_y^{B*}$
 - 2.2. If the game does not have a unique equilibrium, the balanced behavior is adopted as the final control effort sequences: $\mathbf{u}_x^f = \mathbf{u}_x^{mid}$, $\mathbf{u}_y^f = \mathbf{u}_y^{mid}$
 3. The first element of the final control effort sequences is applied, and the control signal at time k is calculated using Equation (18).
 4. The Shifted sequences $\mathbf{u}_x^s$ and $\mathbf{u}_y^s$ are calculated.
-

Remark 4: Regarding the scalability of the proposed framework, as the number of subsystems increases, the dynamic game formulation can be directly constructed based on the interaction

structure and topology among subsystems. In this way, redundant decision branches can be eliminated, thereby reducing the computational complexity associated with large-scale systems.

4. Simulation and Results

In this section, the performance of the proposed control method for the Quadruple-Tank system is evaluated. First, the system is introduced and described. Subsequently, simulation results obtained are presented and analyzed.

4.1. Description of the Quadruple-Tank System

The Quadruple-Tank system consists of four interconnected tanks and two pumps, as illustrated in Figure 3. The inputs are the flow rates of the two pumps, while the outputs are the water levels in the two lower tanks. Owing to its physical configuration, the system can be naturally decomposed into two interacting subsystems. The first subsystem comprises tanks 1 and 3, whereas the second subsystem consists of tanks 2 and 4. The continuous-time linearized model of this system around the operating point is given by (26).

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \begin{bmatrix} -\frac{1}{T_1} & 0 & \frac{A_3}{A_1 T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{A_4}{A_2 T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} \frac{\gamma_1}{A_1} & 0 \\ 0 & \frac{\gamma_2}{A_2} \\ 0 & \frac{(1-\gamma_2)}{A_3} \\ \frac{(1-\gamma_1)}{A_4} & 0 \end{bmatrix} \mathbf{u}(t) \\ \mathbf{y}(t) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}(t) \end{aligned} \tag{26}$$

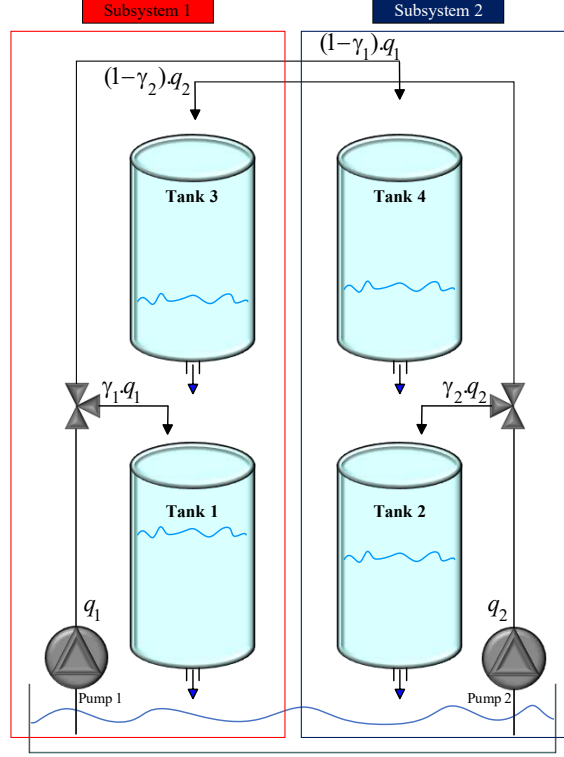


Figure 3. Quadruple-Tank system

Where $\mathbf{x}(t) = [h_1(t) - h_1^0, \dots, h_4(t) - h_4^0]^T$ and $\mathbf{u}(t) = [q_1(t) - q_1^0, q_2(t) - q_2^0]^T$ denote the state and input vectors. $T_i = \frac{A_i}{a_i} \sqrt{\frac{2h_i^0}{g}}$ are the time constants of the system [36]. A_i and a_i represent the cross-sectional area of tank i and its drain pipe, respectively.

In this study $A_i = 32 \text{ cm}^2$ and $a_i = 0.071 \text{ cm}^2$. $g = 981 \text{ cm/s}^2$ is the gravity constant. h_i is the height of tank i (cm), and q_1 and q_2 are the flow of each pump (cm^3 / s). The parameters γ_1 and γ_2 determine the distribution of the pump flows through the control valves and are specified prior to the experiment. In this study $\gamma_1 = 0.7$ and $\gamma_2 = 0.6$. In this study, the Quadruple-Tank system is subject to physical constraints on both the tank water levels and the pump flow rates. These constraints are defined as (27).

$$0 \leq h_i \leq 20, \quad 0 \leq q_i \leq 2q_i^0 \quad (27)$$

The operating-point values used in this study are summarized in Table 1.

Table 1. Operating points

h_1^0, h_2^0	[cm]	(12.25, 8.23)
h_3^0, h_4^0	[cm]	(1.63, 0.91)
q_1^0, q_2^0	[cm ³ /s]	(9.98, 10.05)

Using the continuous-time state-space model presented in (26), the system is discretized using the zero-order hold (ZOH) method with a sampling time of $T_s = 2$ s. The resulting discrete transfer function is given in (28).

$$G(z^{-1}) = \begin{bmatrix} \frac{0.04314z^{-1}}{1-0.9723z^{-1}} & \frac{0.0009281z^{-1} + 0.0008962z^{-2}}{1-1.898z^{-1} + 0.9004z^{-2}} \\ \frac{0.0009247z^{-1} + 0.0008832z^{-2}}{1-1.868z^{-1} + 0.8715z^{-2}} & \frac{0.03687z^{-1}}{1-0.9663z^{-1}} \end{bmatrix} \quad (28)$$

To derive the CARMA model of each subsystem, the discrete transfer function obtained in (28) is utilized. Specifically, each row of the transfer function matrix represents the relationship between the subsystem inputs and its corresponding output. Accordingly, the subsystem models are formulated as (29).

$$\begin{aligned} G_1(z^{-1}) &= \begin{bmatrix} \frac{0.04314z^{-1}}{1-0.9723z^{-1}} & \frac{0.0009281z^{-1} + 0.0008962z^{-2}}{1-1.898z^{-1} + 0.9004z^{-2}} \end{bmatrix} \\ G_2(z^{-1}) &= \begin{bmatrix} \frac{0.0009247z^{-1} + 0.0008832z^{-2}}{1-1.868z^{-1} + 0.8715z^{-2}} & \frac{0.03687z^{-1}}{1-0.9663z^{-1}} \end{bmatrix} \end{aligned} \quad (29)$$

To derive the CARMA representation, the Left Matrix Fraction Description (LMFD) method is applied to each subsystem [30]. In this approach, the denominators of the transfer function elements within each row are unified by determining their least common denominator. The

resulting numerator and denominator polynomials are then used to construct the corresponding CARMA model. The equations in (30) represent this procedure.

$$\begin{aligned} G_i(z^{-1}) &= A_i^{-1}(z^{-1})B_i(z^{-1})z^{-1}, \forall i = 1, 2 \\ B_i(z^{-1}) &= [B_{i1}(z^{-1}) \quad B_{i2}(z^{-1})] \end{aligned} \quad (30)$$

Consequently, the CARMA model polynomials for the two subsystems of the Quadruple-Tank system are given by (31) and (32).

$$\begin{aligned} A_1(z^{-1}) &= [1 - 1.898z^{-1} + 0.9004z^{-2}] \\ B_{11}(z^{-1}) &= [0.04314 - 0.0399z^{-1}] \\ B_{12}(z^{-1}) &= [0.0009281 + 0.0008962z^{-1}] \end{aligned} \quad (31)$$

$$\begin{aligned} A_2(z^{-1}) &= [1 - 1.868z^{-1} + 0.8715z^{-2}] \\ B_{21}(z^{-1}) &= [0.0009247 + 0.0008832z^{-1}] \\ B_{22}(z^{-1}) &= [0.03687 - 0.03325z^{-1}] \end{aligned} \quad (32)$$

4.2. Assessment Metrics

To evaluate the performance of the proposed control system, two performance indices are considered:

1. Mean Squared Error (MSE):

The MSE is used as a measure of the accuracy of tracking the reference signal. This measure is expressed by Equation (33).

$$MSE_x = \frac{1}{2} \sum_{i=1}^n (\hat{y}_{xi} - y_{rxi}) \quad (33)$$

2. Integral of Squared Control Effort (ISCE)

The ISCE is used to measure the control effort and energy consumption in the control process [26, 37]. This measure is expressed by Equation (34).

$$ISCE_x = \int_0^{T_f} \Delta u_x^2 dt \quad (34)$$

4.3. Control System Simulation and Results

To provide a comprehensive assessment, the proposed controller is compared with three control strategies. The first method is the DGPC method without AM layer. This comparison is performed in the form of two configurations of aggressive tuning (low λ_x) and conservative tuning (high λ_x), which allows for a direct evaluation of the proposed AM contribution. The second method is a centralized GPC controller, which serves as a reference corresponding to a fully centralized control architecture with complete access to all system information. This comparison assesses how closely the proposed distributed strategy approaches the performance of an ideal centralized controller while retaining the inherent advantages of a distributed implementation. The third method is a fuzzy-based HDMPC (FDMPC), implemented according to the hierarchical coordination framework presented in [1]. In this approach, interactions among subsystems are managed through an upper-level fuzzy negotiation mechanism. This benchmark was selected because it addresses the same fundamental objectives as the proposed method, namely the management of subsystem interactions and the adaptation of control actions to varying operating conditions within a hierarchical distributed control framework. Although the proposed approach employs a game-theoretic AM mechanism rather than fuzzy negotiation, both methods pursue similar control objectives. Therefore, the FDMPC strategy provides an appropriate benchmark for evaluating the effectiveness of the proposed control architecture. for implementation on the considered Quadruple-Tank system, the fuzzy rule base was adopted from [37], which was developed within the same hierarchical fuzzy-based distributed predictive control framework as that presented in [1]. The selected fuzzy rules are formulated based on the control effort component, making them particularly suitable for comparison with the proposed method.

Remark 5: The abbreviations used in the following figures and tables are defined as follows. $DGPC_{C1}$ denotes the DGPC controller without the AM layer operating under an aggressive tuning configuration, whereas $DGPC_{C2}$ denotes the same controller operating under a conservative tuning configuration. CGPC denotes the centralized GPC controller, FDMPC denotes the fuzzy-based HDMPC adopted from [1], and $DGPC_{GT}$ denotes the proposed control method.

The main simulation parameters used throughout this study are as follows: The Final simulation time is $T_f = 400s$ the sampling time is $T_s = 2s$, and the prediction horizon is $N = 10$. The dynamic game weighting factor is $\alpha = 1.75$. For both subsystems, the control effort penalty coefficient range is $\lambda_i \in [1,7]$, and the corresponding decision space is defined as $\Lambda_i \in \{\lambda_{i\min} = 1, \lambda_{i\mid} = 4, \lambda_{i\max} = 7\}$. The CGPC controller employs a fixed penalty coefficient of $\lambda = 1$. The DGPC controllers without the AM layer use $\lambda_1 = 1, \lambda_2 = 1$ for $DGPC_{C1}$ and $\lambda_1 = 4, \lambda_2 = 4$ for $DGPC_{C2}$. The FDMPC controller design parameters and matrices used in this study are provided in Appendix B.

The simulation study is divided into two case studies. Case 1 evaluates the reference tracking performance under nominal operating conditions, whereas Case 2 investigates the disturbance rejection capability of the control strategies considered. In Case 2, a disturbance was introduced to emulate an unmeasured inflow entering the upper tanks of the Quadruple-Tank system. The disturbance was applied over the sampling interval $k = 100$ to $k = 130$, producing an additive bias in the dynamics of tanks 3 and 4. From a physical perspective, this disturbance can be interpreted as an additional constant inflow that increases the water level in the upper tanks by approximately 0.3 cm per sampling period. This disturbance scenario was selected to evaluate the ability of the controllers to reject persistent process disturbances and to restore the system to the desired operating conditions while maintaining satisfactory control performance.

Remark 6: The physical constraints of the Quadruple-Tank system, including the admissible tank levels and pump flow rates, were considered in all simulation scenarios. The simulation results confirm that all investigated control strategies satisfied these constraints throughout the simulation horizon. Consequently, the reported reference tracking and disturbance rejection performances were achieved while ensuring feasible and physically admissible system operation.

The simulation results are examined below. Figure 4 presents the reference tracking responses of the considered control strategies for the water levels of tanks 1 and 2 in Case 1. The results indicate that the tracking performance of the DGPC controller without the AM layer was strongly influenced by the selected control effort penalty coefficient. Specifically, the aggressive configuration (DGPC_{C1}) achieved a faster transient response and higher tracking accuracy, whereas the conservative configuration (DGPC_{C2}) exhibited a slower response accompanied by a larger overshoot. The comparison between CGPC and distributed control strategies indicates that the distributed architecture achieved performance comparable to that of the centralized controller, despite relying solely on local subsystem models. Furthermore, the proposed controller (DGPC_{GT}) produced a response comparable to those of DGPC_{C1} and CGPC, indicating that the game-based AM layer was capable of selecting appropriate control actions. Compared with the FDMPC method, the proposed controller exhibited a slightly larger overshoot following the reference change; however, its overall tracking accuracy remained comparable. These results demonstrate that the proposed game-based AM layer effectively managed subsystem interactions while maintaining satisfactory closed-loop performance. Figure 5 illustrates the dynamic responses of the upper tanks (Tanks 3 and 4) in Case 1. Although these states were not directly controlled to follow reference trajectories, their behavior significantly influences the overall dynamics and interaction patterns of the Quadruple-Tank process. As shown in the figure, all control strategies

maintained the upper-tank water levels within admissible operating limits and exhibited stable internal dynamics.

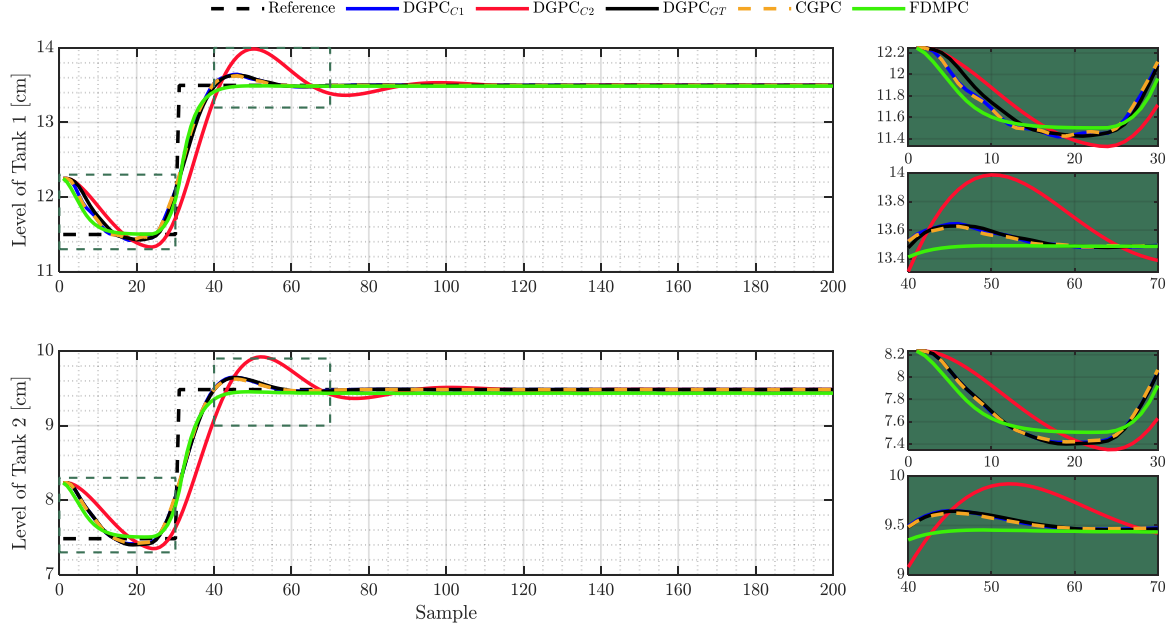


Figure 4. Level of tanks 1 and 2 (Case1)

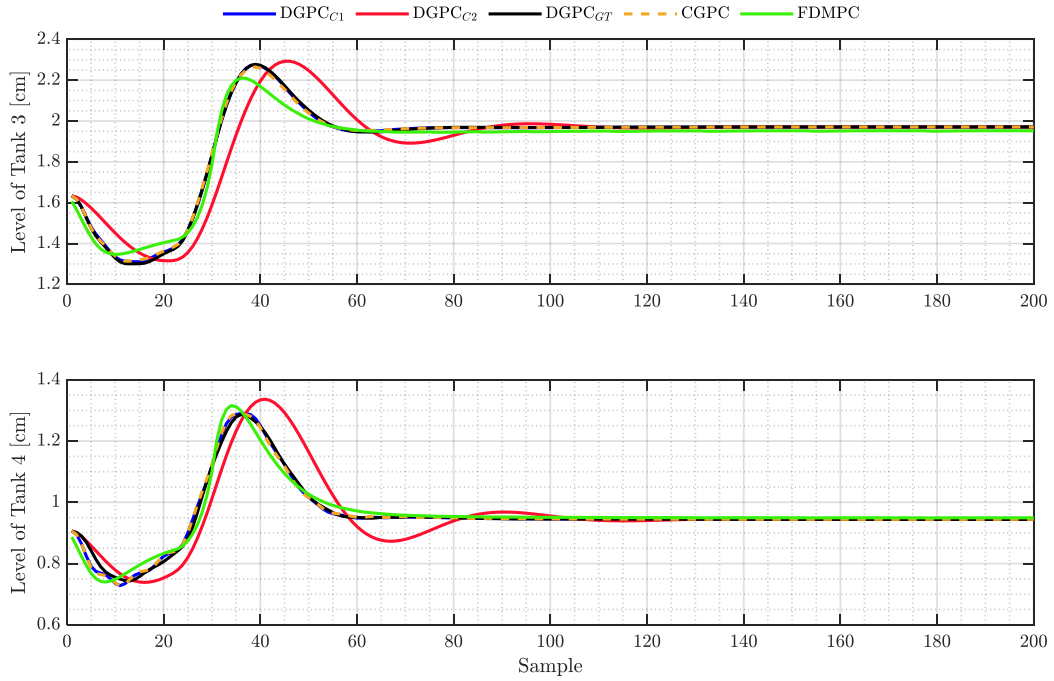


Figure 5. Level of tanks 3 and 4 (Case1)

Figures 6 and 7 present the control inputs (pump flow rates) and the control efforts in Case 1. As expected, the control actions of the DGPC controller without the AM layer were strongly influenced by the selected control effort penalty coefficient. The aggressive configuration (DGPC_{C1}) produced larger control variations and more oscillatory control signals to achieve better tracking performance. In contrast, the conservative configuration (DGPC_{C2}) generated smoother control actions at the expense of a slower transient response. A comparable level of control activity was also observed for CGPC. The FDMPC method generated relatively smooth control signals throughout most of the time. However, pronounced peaks appeared around the reference changes. In comparison, the proposed controller (DGPC_{GT}) maintained control variations at levels comparable to, or lower than, those of DGPC_{C1} and CGPC, while avoiding the excessive conservatism associated with DGPC_{C2}. These results demonstrate that the proposed game-based AM layer achieved a favorable equilibrium between control effort and closed-loop performance by adaptively adjusting the control strategy according to the operating conditions.

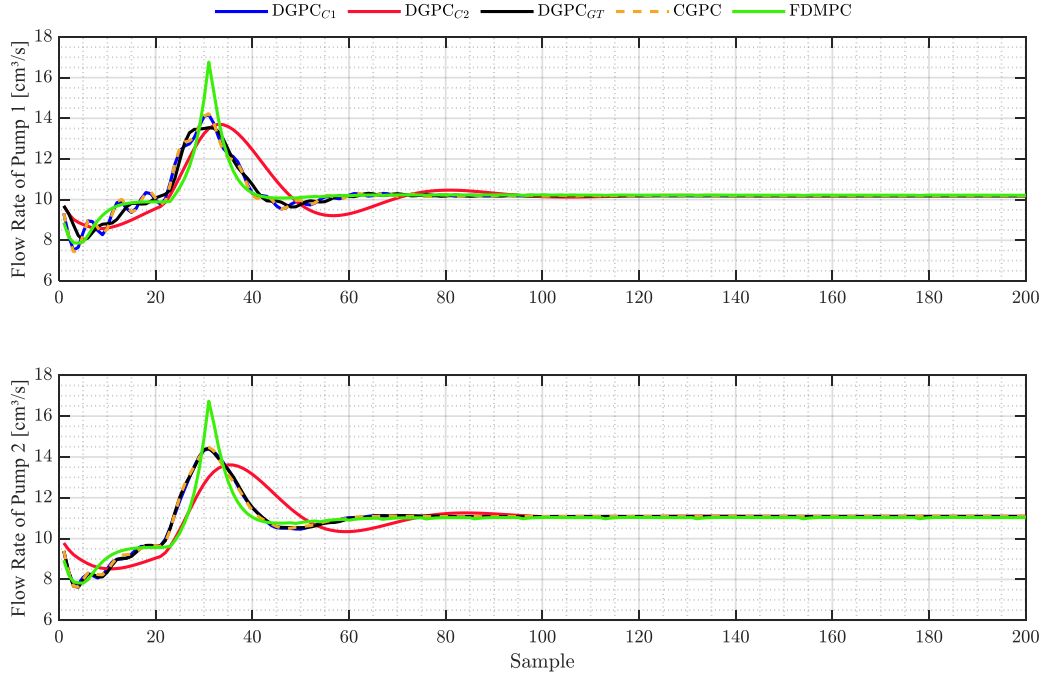


Figure 6. Flow Rate of Pumps (Case1)

Table 2 summarizes the MSE values obtained for the considered control strategies in Case 1. Consistent with the reference tracking responses shown in Figure 4, the aggressive distributed configuration (DGPC_{C1}) achieved the lowest MSE among the DGPC-based methods for both subsystems. CGPC and FDMPC also produced slightly lower MSE values than the proposed method. Nevertheless, DGPC_{GT} achieved tracking accuracy comparable to that of the best-performing strategies while significantly outperforming the conservative configuration (DGPC_{C2}). Table 3 presents the ISCE values for all control strategies. As expected, DGPC_{C2} yielded the lowest control effort in both subsystems, reflecting its conservative tuning. However, this reduction in control activity was accompanied by the largest tracking errors, as shown in Table 2.

Table 2. MSE for subsystems (Case1)

Subsystem	CGPC	DGPC _{C2}	DGPC _{C1}	DGPC _{GT}	FDMPC
1	0.0384	0.0889	0.0398	0.0448	0.0323
2	0.0418	0.1042	0.0432	0.0438	0.0392
Total	0.0802	0.1931	0.0830	0.0886	0.0715

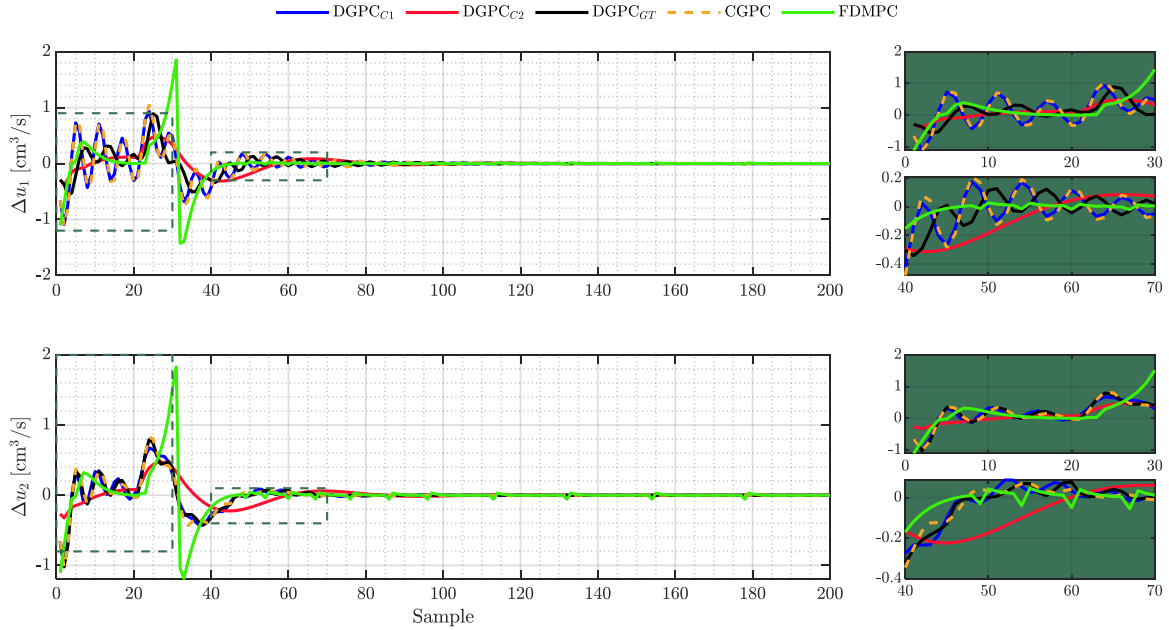


Figure 7. Control efforts (Case1)

In contrast, $DGPC_{GT}$ achieved a substantial reduction in control effort compared with $DGPC_{C1}$, CGPC, and FDMPC. Specifically, for Subsystem 1, the ISCE value of $DGPC_{GT}$ was approximately 49%, 56%, and 71% lower than those of $DGPC_{C1}$, CGPC, and FDMPC, respectively. A similar trend was observed for Subsystem 2, where the proposed method required less control effort than both CGPC and FDMPC.

Table 3. ISCE for subsystems (Case1)

Subsystem	CGPC	$DGPC_{C2}$	$DGPC_{C1}$	$DGPC_{GT}$	FDMPC
1	22.71	6.45	19.44	9.91	33.89
2	13.28	5.36	12.38	12.89	30.98
Total	35.99	11.81	31.82	22.80	64.87

Figure 8 illustrates the final decisions adopted by the subsystems within the AM layer in Case 1. At each sampling instant, each subsystem selected one of the candidate control sequences associated with different control effort penalty coefficients according to the current operating conditions and the interactions among the subsystems. Consequently, the adaptive layer continuously adjusted the control strategy instead of relying on a single fixed controller configuration. As shown in the figure, different candidate control sequences were selected during the simulation, indicating that the preferred control action varied with the operating conditions. This adaptive selection mechanism enabled the controller to balance tracking performance and control effort while accounting for subsystem interactions. These observations are consistent with the results presented in Figures 4-7 and Tables 2-3.

Figure 9 presents the reference tracking responses of tanks 1 and 2 in Case 2, in which an external disturbance was applied to the process. As shown, $DGPC_{C1}$, CGPC, and the proposed $DGPC_{GT}$ controller exhibited the smallest deviations during the disturbance interval and recovered more rapidly to the desired operating conditions after the disturbance was removed. In contrast, $DGPC_{C2}$

has significantly larger deviations, indicating that its conservative tuning reduced its disturbance rejection capability. The FDMPC method also exhibited larger deviations during the disturbance interval than $DGPC_{GT}$.

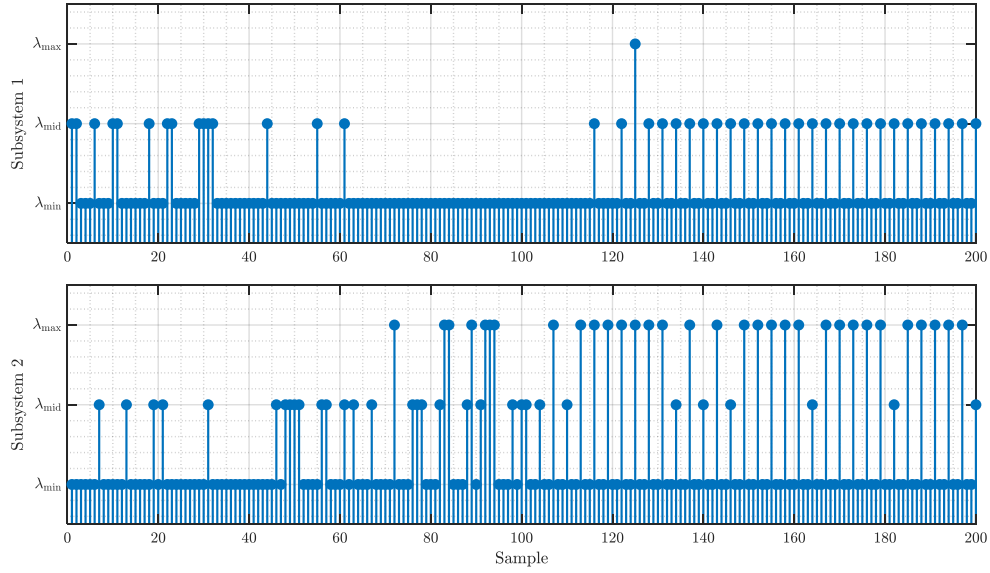


Figure 8. Final decisions adopted by the subsystems (Case1)

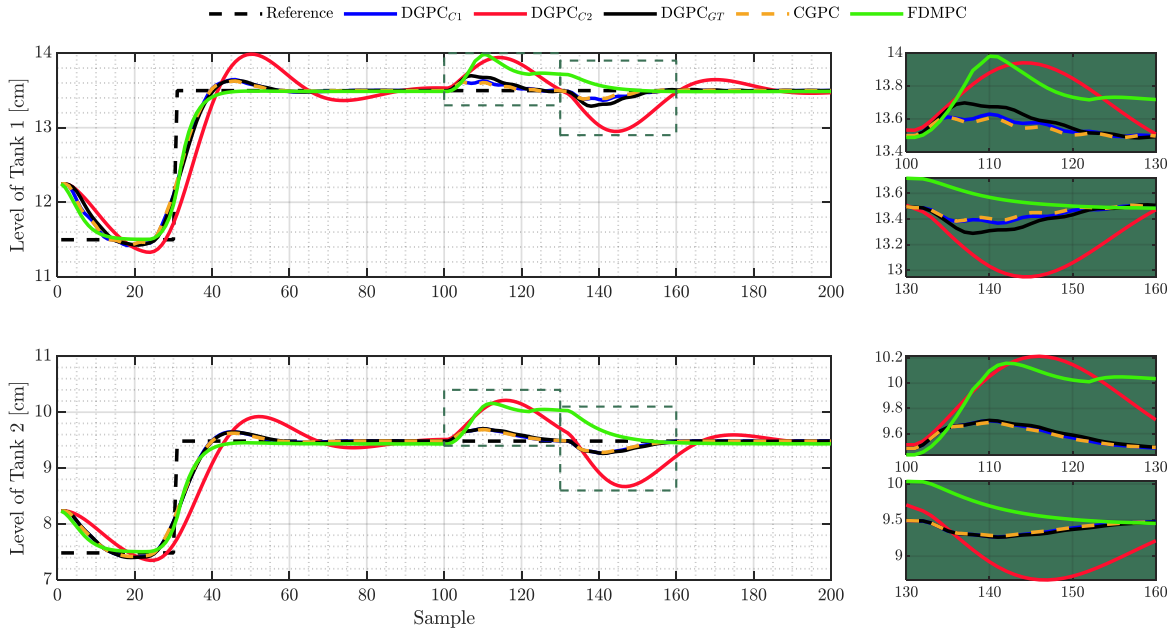


Figure 9. Level of tanks 1 and 2 (Case2)

Figure 10 illustrates the dynamic responses of the upper tanks (Tanks 3 and 4) in Case 2. Because the disturbance was introduced as an additional inflow to the upper tanks, its effect appeared directly as an increase in their water levels during the disturbance interval. Consequently, all control strategies exhibited a similar increase in the water levels of tanks 3 and 4 between sampling instants $k = 100$ and $k = 130$. After the disturbance was removed, the water levels gradually returned to their nominal operating conditions, indicating that all controllers maintained stable internal dynamics despite the external perturbation.

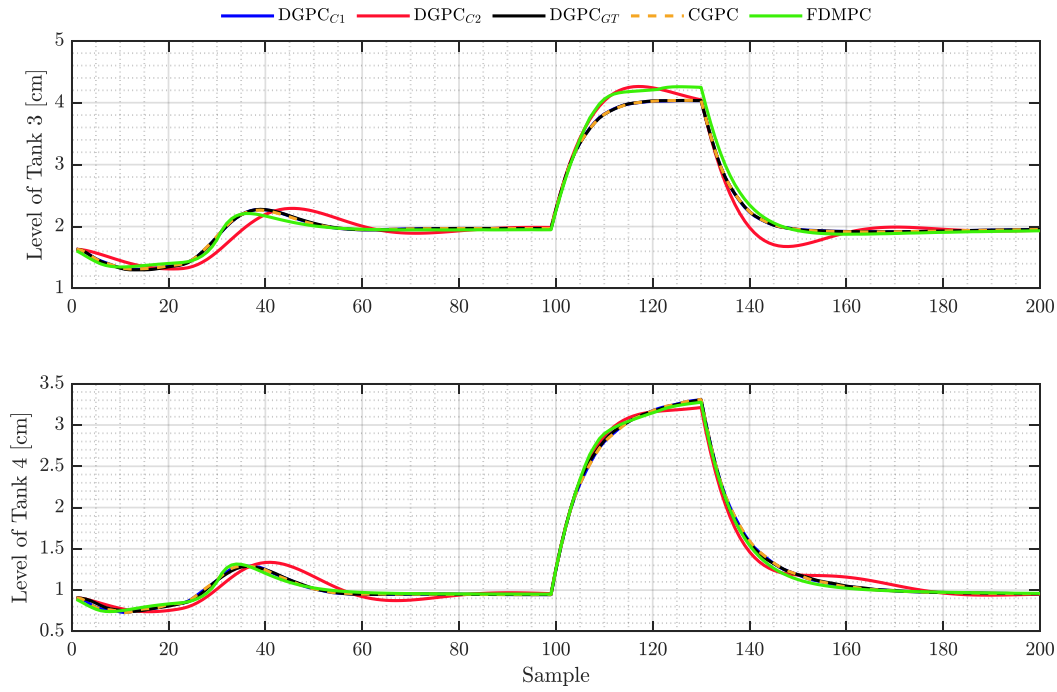


Figure 10. Level of tanks 3 and 4 (Case2)

Figures 11 and 12 present the control inputs (pump flow rates) and the corresponding control efforts of the considered controllers in Case 2. The disturbance introduced between sampling instants $k = 100$ and $k = 130$ was clearly reflected in the control actions of all methods. Similar to Case 1, DGPC_{C1} and CGPC exhibited more aggressive control behavior, whereas DGPC_{C2} produced smoother control signals. The FDMPC method generated significant oscillations and peaks during

the disturbance interval. The proposed $DGPC_{GT}$ controller adapted its control behavior through the AM layer in response to the disturbance. Consequently, more aggressive candidate control sequences were selected during the disturbance interval, resulting in tracking performance and control actions comparable to those of $DGPC_{C1}$ and CGPC. This adaptive behavior contributed to the improved disturbance rejection performance observed in Figure 9.

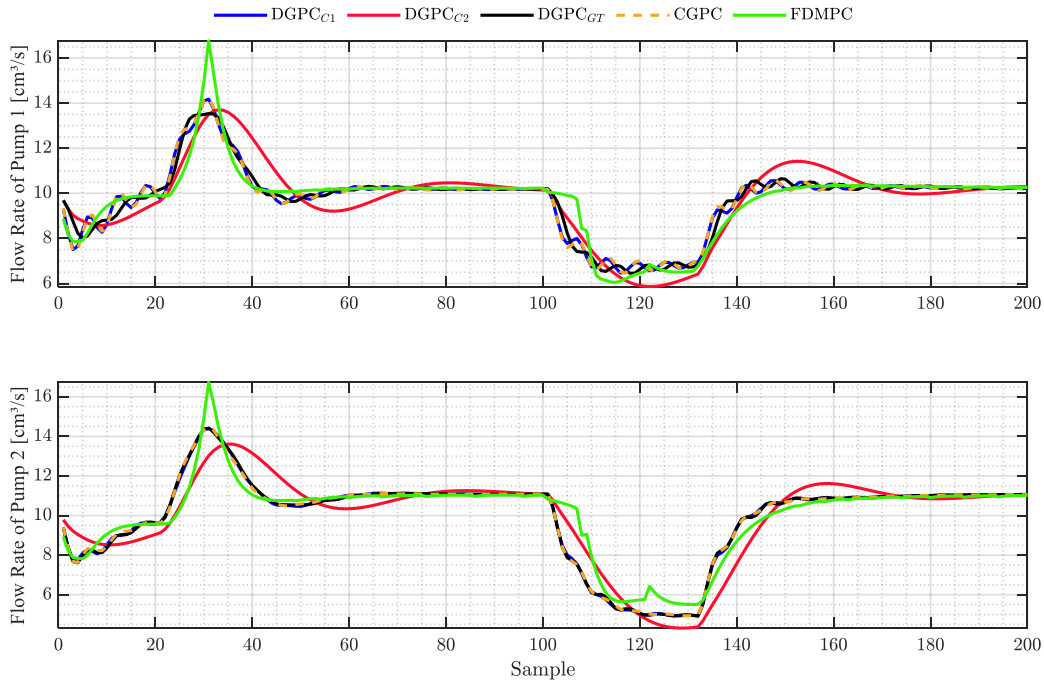


Figure 11. Flow Rate of Pumps (Case2)

Tables 4 and 5 quantitatively confirm the observations presented in Figures 9-12. While CGPC and $DGPC_{C1}$ achieved the lowest MSE values because of their more aggressive control actions, the proposed $DGPC_{GT}$ controller provided comparable tracking performance while outperforming the FDMPC method in terms of the total MSE. In addition, although $DGPC_{C2}$ has the lowest control effort, it exhibited the weakest disturbance rejection performance. In contrast, the proposed controller maintained a substantially lower control effort than $DGPC_{C1}$, CGPC, and FDMPC,

demonstrating that the AM layer improved disturbance rejection without relying on a permanently aggressive control strategy.

Table 4. MSE for subsystems (Case2)

Subsystem	CGPC	DGPC _{C2}	DGPC _{C1}	DGPC _{GT}	FDMPC
1	0.0391	0.1231	0.0412	0.0481	0.0461
2	0.0462	0.1931	0.0483	0.0494	0.0873
Total	0.0853	0.3162	0.0895	0.0975	0.1334

Table 5. ISCE for subsystems (Case2)

Subsystem	CGPC	DGPC _{C2}	DGPC _{C1}	DGPC _{GT}	FDMPC
1	32.02	12.01	29.94	18.09	43.92
2	27.52	14.67	26.63	27.76	44.47
Total	59.54	26.68	56.57	45.85	88.39

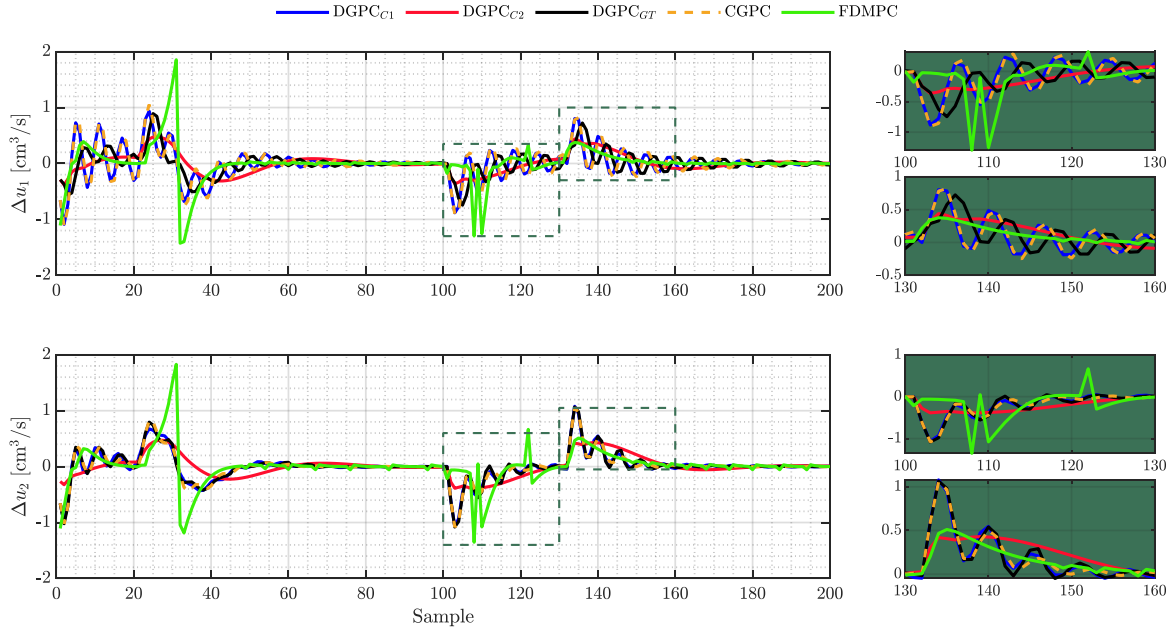


Figure 12. Control efforts (Case2)

Figure 13 illustrates the final decisions adopted by the subsystems within the AM layer in Case 2.

Compared with Case 1, a noticeable increase in the selection of more aggressive candidate control

sequences was observed during the disturbance interval and the subsequent recovery period. This behavior indicates that the proposed decision-making mechanism adapted the controller configuration according to the system operating conditions and favored stronger corrective actions when disturbance rejection became more critical.

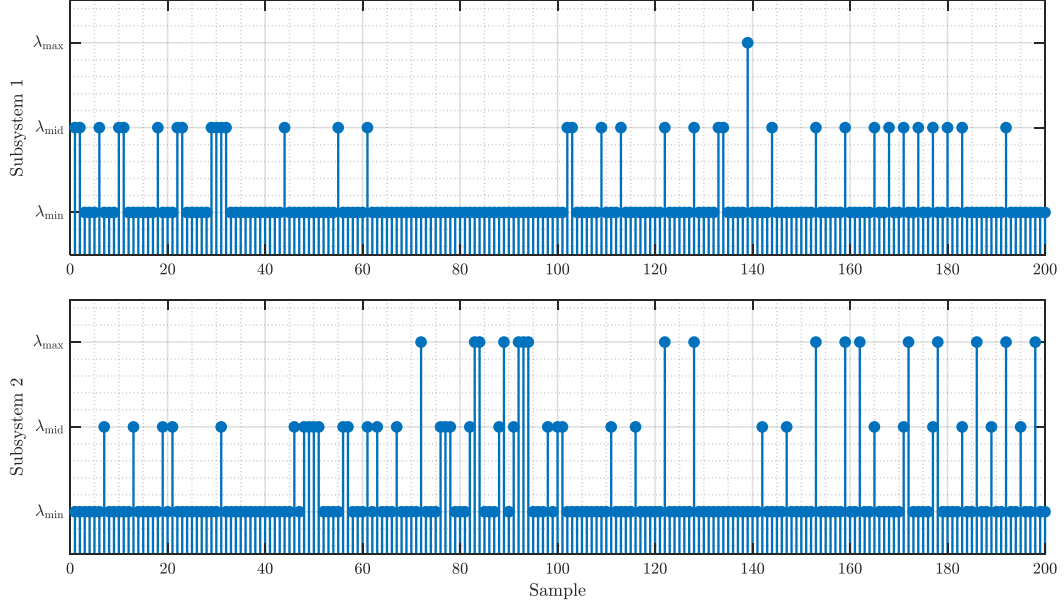


Figure 13. Final decisions adopted by the subsystems (Case2)

Table 6 shows the number of selections for each decision-making mode in both simulation cases. The results quantitatively support the behavior observed in Figure 13, confirming that the AM layer adjusted its decisions in response to changes in the operating conditions and disturbance effects.

Table 6. Number of selections for each decision-making mode

Case	Subs.	Decision-making modes		
		$\lambda_{\min}$	λ_{mid}	$\lambda_{\max}$
1	1	157	42	1
	2	136	32	32
2	1	165	34	1
	2	149	34	17

Figure 14 presents the total system cost associated with the proposed controller in Cases 1 and 2. As shown, the total cost increased during periods of significant system changes, including reference transitions and disturbance events. In both cases, the cost decreased rapidly after the transient response and approached zero under steady-state operating conditions.

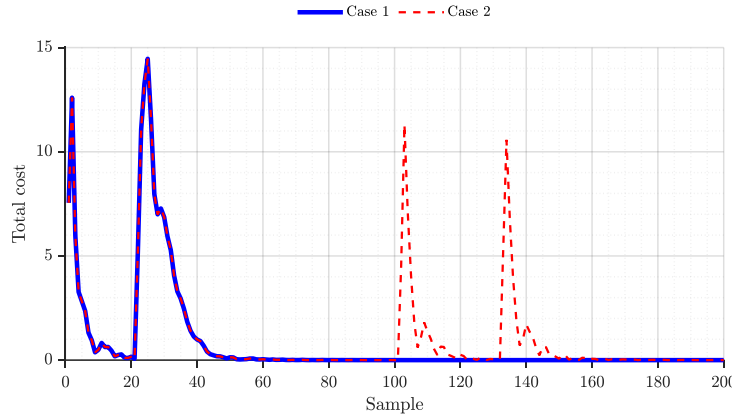


Figure 14. Total system cost function

4.4. Robust Analysis

To evaluate the robustness of the proposed control framework, a parametric uncertainty analysis was conducted by varying the valve flow distribution parameters, γ_1 and γ_2 , by $\pm 25\%$ around their nominal values. These parameters determine how the pump flows are distributed among the interconnected tanks and therefore directly influence the interaction dynamics of the system. In practical industrial environments, calibration inaccuracies and unwanted variations may cause these parameters to deviate from their nominal values. Consequently, they provide an appropriate basis for assessing the robustness of the proposed controller.

Figure 15 shows the levels in tanks 1 and 2 for different values of the valve flow distribution parameters. As shown, the introduced parameter uncertainty affected the transient characteristics of the responses, leading to slight variations in the response trajectories and overshoot levels.

Nevertheless, all output responses remained bounded and converged to the desired reference values. These results demonstrate that the proposed controller preserved satisfactory closed-loop tracking performance in the presence of significant parameter uncertainty and exhibited robustness to variations in the subsystem interaction dynamics.

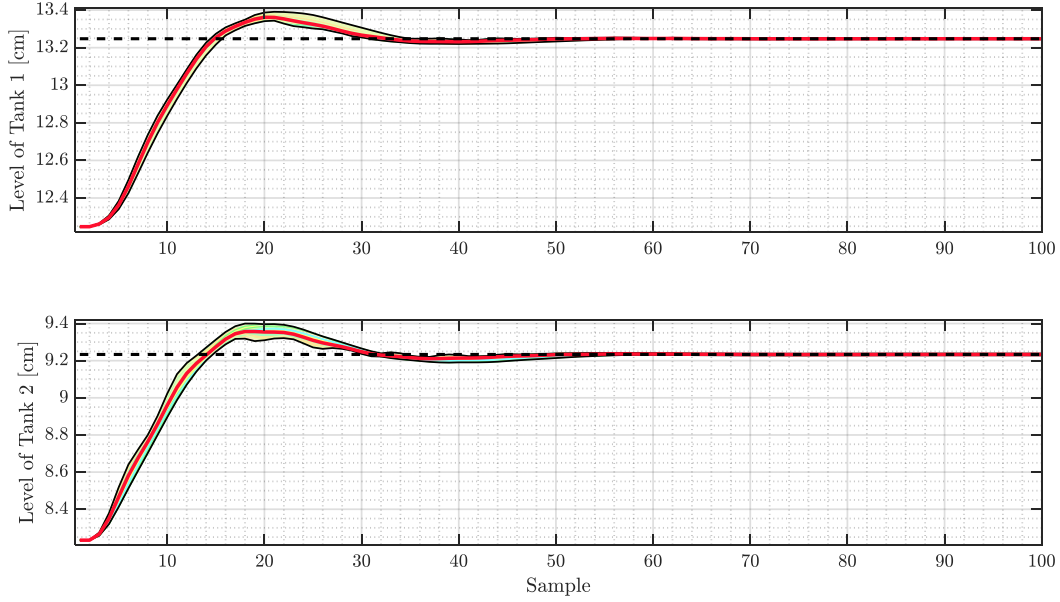


Figure 15. Level of tanks 1 and 2 under $\pm 25\%$ uncertainty in γ_1 and γ_2

Figure 16 presents the corresponding control signals generated under uncertainty conditions. Although the parameter variations alter the required control actions, all control signals remain bounded and within an acceptable operating range throughout the simulation horizon. No excessive oscillations or unstable control behavior are observed. The AM automatically selects the most appropriate candidate control sequences according to the modified system dynamics, enabling the controller to compensate for uncertainty while maintaining acceptable tracking performance. Overall, the results obtained in Figs. 15 and 16 demonstrate the robustness of the proposed control architecture against uncertainties in critical process parameters and confirm its ability to maintain bounded closed-loop behavior under parameter variations.

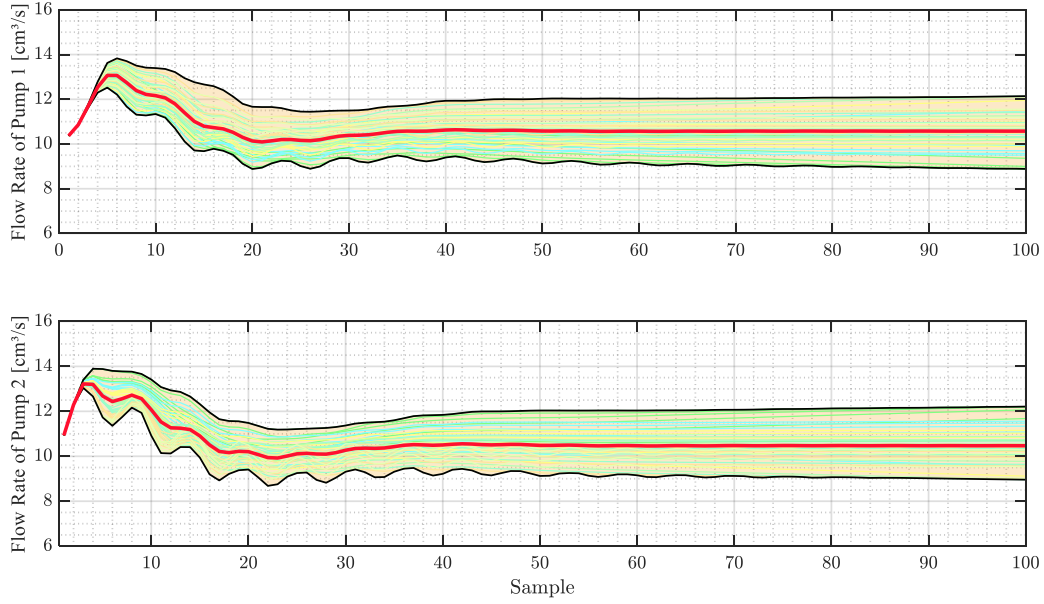


Figure 16. Flow Rate of Pumps under $\pm 25\%$ uncertainty in γ_1 and γ_2

5. Conclusion

In this paper, a novel HDMPC structure is presented. The proposed structure consists of two layers. The first layer is based on GPC, and the second layer is designed based on dynamic non-cooperative game theory. In the first layer, local GPC controllers were developed based on the CARIMA model, so that the mutual effects and interactions between subsystems are explicitly considered in the prediction process. Then, using the designed local controllers and different levels of control effort penalty coefficient, a set of control options appropriate to different operating conditions was generated and defined in the form of a decision space for each subsystem. In the second layer, an AM based on a dynamic non-cooperative game was developed so that each subsystem can choose the most appropriate control option at any time, considering the current state of the system, its performance quality, and the behavior of other related subsystems. This mechanism enables adaptive, and coordinated decisions to be made according to operational conditions. The efficiency of the proposed method was evaluated on a Quadruple-Tank system and

compared with CGPC controllers, DGPC without AM, and fuzzy logic-based HDMPC. Simulation results showed that although CGPC and DGPC controllers with low penalty coefficients provide higher tracking accuracy, this improvement is accompanied by an increase in control effort. In contrast, the proposed structure, using a game-based AM, was able to select the appropriate control strategy at any time and establish an effective equilibrium between tracking accuracy and control effort. Specifically, under nominal conditions, the proposed method has achieved a significant reduction in the ISCE index compared to the CGPC, $DGPC_{C1}$, and FDMPC controllers, despite achieving similar MSE. Also, in the presence of process disturbances, the AM, by automatically selecting more aggressive strategies, limited output deviations and improved system recovery speed, resulting in performance close to that of the centralized controller and better than the FDMPC method. In addition, the results of the robust analysis of changes in the flow distribution parameters of the valves showed that the stability of the responses was maintained and no significant steady-state error was created in the system. Overall, the results indicate that the proposed framework is an effective option for controlling coupled multivariable systems. Future research directions include the development of the method for large-scale systems, consideration of communication constraints, and its laboratory implementation.

Appendix

Appendix A. Vectors and Matrices Used in the Proposed Controller Formulation

This appendix provides the vectors and matrices used in the formulation of the proposed controller, as given in equations (12) and (13).

$$\mathbf{F}_x(z^{-1}) = \begin{bmatrix} F_{x(d+1)}(z^{-1}) \\ F_{x(d+2)}(z^{-1}) \\ \vdots \\ F_{x(d+N)}(z^{-1}) \end{bmatrix} \quad (\text{A.1})$$

$$\mathbf{G}_x = \begin{bmatrix} g_{x0} & 0 & \cdots & 0 \\ g_{x1} & g_{x0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{xN-1} & \cdots & \cdots & g_{x0} \end{bmatrix} \quad (\text{A.2})$$

$$\mathbf{G}_y = \begin{bmatrix} g_{y0} & 0 & \cdots & 0 \\ g_{y1} & g_{y0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{yN-1} & \cdots & \cdots & g_{y0} \end{bmatrix} \quad (\text{A.3})$$

$$\mathbf{G}'_x(z^{-1}) = \begin{bmatrix} (G_{x(d+1)}(z^{-1}) - g_{x0})z \\ (G_{x(d+2)}(z^{-1}) - g_{x0} - g_{x1}z^{-1})z^2 \\ \vdots \\ (G_{x(d+N)}(z^{-1}) - g_{x0} - g_{x1}z^{-1} - g_{xN-1}z^{-(N-1)})z^n \end{bmatrix} \quad (\text{A.4})$$

$$\mathbf{G}'_y(z^{-1}) = \begin{bmatrix} (G_{y(d+1)}(z^{-1}) - g_{y0})z \\ (G_{y(d+2)}(z^{-1}) - g_{y0} - g_{y1}z^{-1})z^2 \\ \vdots \\ (G_{y(d+N)}(z^{-1}) - g_{y0} - g_{y1}z^{-1} - g_{yN-1}z^{-(N-1)})z^n \end{bmatrix} \quad (\text{A.5})$$

Appendix B. FDMPC controller design parameters and matrices

To facilitate the reproducibility of the FDMPC controller implementation used for comparison, the design parameters and matrices are summarized in Table B.1.

Table B.1. FDMPC controller design parameters and matrices

Fuzzy membership function parameters	$a_x = 0, c_x = 0.5, b_x = 1$
Fuzzy rules weights	Low (w_1) = 0.9 Medium (w_2) = 0.6 High (w_3) = 0.3
The local feedback gains	$K_1 = [-1.7636 \quad -1.2434]$ $K_2 = [-0.3548 \quad -0.3120]$
Weighting matrix of the terminal cost	$P_1 = \begin{bmatrix} 37.3595 & 16.9026 \\ 16.9026 & 23.8592 \end{bmatrix}$ $P_2 = \begin{bmatrix} 1.8847 & -0.0534 \\ -0.0524 & 0.9826 \end{bmatrix}$

References

- [1] Masero E, Francisco M, Maestre JM, Revollar S, Vega P (2021) Hierarchical distributed model predictive control based on fuzzy negotiation. *Expert Syst Appl* 176:114–836. <https://doi.org/10.1016/j.eswa.2021.114836>
- [2] Ofoegbu E, Adeleke K (2026) Optimising complex systems: model predictive control in engineering applications. *J Eng Appl Sci* 73:108. <https://doi.org/10.1186/s44147-026-00969-w>
- [3] Ding Y, Wang L, Li Y, Li D (2018) Model predictive control and its application in agriculture: A review. *Comput Electron Agric* 151:104–117. <https://doi.org/10.1016/j.compag.2018.06.004>
- [4] Song Y, Zhang B, Wen C, Wang D, Wei G (2025) Model predictive control for complicated dynamic systems: a survey. *Int J Syst Sci* 56:2168–2193. <https://doi.org/10.1080/00207721.2024.2439473>
- [5] Schwenzer M, Ay M, Bergs T, Abel D (2021) Review on model predictive control: an engineering perspective. *Int J Adv Manuf Technol* 117:1327–1349. <https://doi.org/10.1007/s00170-021-07682-3>
- [6] Scattolini R (2009) Architectures for distributed and hierarchical Model Predictive Control – A review. *J Process Control* 19:723–731. <https://doi.org/10.1016/j.jprocont.2009.02.003>
- [7] Christofides PD, Scattolini R, Muñoz de la Peña D, Liu J (2013) Distributed model predictive control: A tutorial review and future research directions. *Comput Chem Eng* 51:21–41. <https://doi.org/10.1016/j.compchemeng.2012.05.011>

- [8] Negenborn RR, Maestre JM (2014) Distributed model predictive control: An overview and roadmap of future research opportunities. *IEEE Control Syst* 34:87–97. <https://doi.org/10.1109/mcs.2014.2320397>.
- [9] Alvarado I, Limon D, Muñoz de la Peña D et al (2011) A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark. *J Process Control* 21:800–815. <https://doi.org/10.1016/j.jprocont.2011.03.003>
- [10] Yang Y, Wang Y, Manzie C, Pu Y (2025) Real-Time distributed model predictive control with limited communication data rates. *IEEE Trans Automat Contr* 70:4928–4935. <https://doi.org/10.1109/tac.2025.3545220>
- [11] Shorinwa O, Schwager M (2023) Distributed model predictive control via separable optimization in multiagent networks. *IEEE Trans Automat Contr* 69:230–245. <https://doi.org/10.1109/tac.2023.3269338>
- [12] Resmi R, Mija S, Jacob J (2023) Distributed model-predictive control using exponentially weighted laguerre functions and periodic time triggering for dynamic consensus of linear multiagent systems. *IEEE Trans Industr Inform* 19:9867–9876. <https://doi.org/10.1109/tii.2022.3233661>
- [13] Arauz T, Chanfreut P, Maestre JM (2022) Cyber-security in networked and distributed model predictive control. *Annu Rev Control* 53:338–355. <https://doi.org/10.1016/j.arcontrol.2021.10.005>
- [14] Nguyen TL, Mai XS, Dao PN (2025) A robust distributed model predictive control strategy for Leader–Follower formation control of multiple perturbed wheeled mobile robotics. *Eur J Control* 81:101160. <https://doi.org/10.1016/j.ejcon.2024.101160>
- [15] Li Y, Zhang H, Wang Z, Huang C, Yan H (2025) Data-based distributed model predictive control for large-scale systems. *Nonlinear Dyn* 113:3965–3980. <https://doi.org/10.1007/s11071-024-10340-4>
- [16] Mi X, Zou Y, Li S, Karimi HR (2019) Self-Triggered DMPC design for cooperative multiagent systems. *IEEE Trans Ind Electron* 67:512–520. <https://doi.org/10.1109/tie.2019.2896098>
- [17] Aponte-Rengifo O, Vega P, Francisco M (2023) Deep reinforcement learning agent for negotiation in multi-agent cooperative distributed predictive control. *Applied Sciences* 13:2432. <https://doi.org/10.3390/app13042432>
- [18] Shalmani RA, Rahmani M, Bigdeli N (2020) Nash-based robust distributed model predictive control for large-scale systems. *J Process Control* 88:43–53. <https://doi.org/10.1016/j.jprocont.2020.02.005>
- [19] Hu Y, Sun J, Peng W, Zhang D (2021) Nash equilibrium-based distributed predictive control strategy for thickness and tension control on tandem cold rolling system. *J Process Control* 97:92–102. <https://doi.org/10.1016/j.jprocont.2020.11.014>

- [20] Maestre JM, Muñoz de la Peña D, Camacho EF (2011) Distributed model predictive control based on a cooperative game. *Optim Control Appl Methods* 32:153–176. <https://doi.org/10.1002/oca.940>
- [21] Alkorta P, Cortajarena JA, Barambones O, Maseda FJ (2020) Effective generalized predictive control of induction motor. *ISA Trans* 103:295–305. <https://doi.org/10.1016/j.isatra.2020.04.008>
- [22] Qu J, Xia Y, Xue S, Wang H, Ma Y (2020) Novel generalized predictive control for photoelectric tracking system based on improved objective function and predictive value correction. *IEEE Access* 8:18893–18901. <https://doi.org/10.1109/access.2020.2966802>
- [23] Morris P (2012) *Introduction to game theory*. Springer Science & Business Media
- [24] Oyehan IA, Osunleke AS, Ajani OO (2023) Closed-Loop stability of a non-minimum phase quadruple tank system using a nonlinear model predictive controller with EKF. *ChemEngineering* 7:74. <https://doi.org/10.3390/chemengineering7040074>
- [25] Bland PC, Han X, Mourtaji I (2025) Linear model predictive control using Takagi–Sugeno fuzzy model and vanilla autoencoder neural networks. *Int J Dynam Control* 13:238. <https://doi.org/10.1007/s40435-025-01759-7>
- [26] Francisco M, Mezquita Y, Revollar S, Vega P, De Paz J F (2019) Multi-agent distributed model predictive control with fuzzy negotiation. *Expert Syst Appl* 129:68–83. <https://doi.org/10.1016/j.eswa.2019.03.056>
- [27] Sorcia-Vázquez FDJ, García-Beltrán CD, Valencia-Palomo G et al (2020) Decentralized robust tube-based model predictive control: Application to a four-tank -system. *Rev Mex Ing Quim* 19:1135–1151. <https://doi.org/10.24275/rmiq/sim778>
- [28] Zhang H, Prempain E (2026) A robust state-feedback control approach for a class of uncertain non-linear systems. *Int J Control* 99:2295–2307. <https://doi.org/10.1080/00207179.2026.2663531>
- [29] El Haoussi F, Tissir EH, Tadeo F, Hmamed A (2011) Delay-dependent stabilisation of systems with time-delayed state and control: application to a quadruple-tank process. *Int J Syst Sci* 42:41–49. <https://doi.org/10.1080/00207720902990054>
- [30] Camacho EF, Bordons C (1999) Commercial model predictive control schemes. *Advanced Textbooks in Control and Signal Processing* 33–50. https://doi.org/10.1007/978-1-4471-3398-8_3
- [31] Saini RST, Pappas I, Avraamidou S, Ganesh HS (2023) Noncooperative distributed model predictive control: A multiparametric programming approach. *Ind Eng Chem Res* 62:1044–1056. <https://doi.org/10.1021/acs.iecr.2c03057>
- [32] Camacho E F, Bordons C, Maestre J M (2026) *Model predictive control*. 3rd ed. Springer.

- [33] Haber R, Bars R, Schmitz U (2012) Predictive control in process engineering: From the basics to the applications. John Wiley & Sons
- [34] Kansha Y, Chiu M-S (2009) Adaptive generalized predictive control based on JITL technique. *J Process Control* 19:1067–1072. <https://doi.org/10.1016/j.jprocont.2009.04.002>
- [35] Ben Sassi D, Chaieb M (2024) A game theory competitive intelligence solution stimulated from a Stackelberg game: A three players scenario. *Eng Appl Artif Intell* 135:108825. <https://doi.org/10.1016/j.engappai.2024.108825>
- [36] Johansson KH (2002) The quadruple-tank process: a multivariable laboratory process with an adjustable zero. *IEEE Trans Contr Syst Technol* 8:456–465. <https://doi.org/10.1109/87.845876>
- [37] Francisco M, Masero E, Morales-Rodelo K, Maestre J, Vega P (2023) Offset-free distributed predictive control based on fuzzy logic: Application to a real four-tank plant. *IFAC Pap OnLine* 56:3247–3252. <https://doi.org/10.1016/j.ifacol.2023.10.1464>