actuators

MDPI

Article

Distributed Model Predictive Control and Coalitional Control Strategies—Comparative Performance Analysis Using an Eight-Tank Process Case Study

Anca Maxim , Ovidiu Pauca and Constantin-Florin Caruntu *

Department of Automatic Control and Applied Informatics, "Gheorghe Asachi" Techical University of Iasi, 700050 Iasi, Romania

* Correspondence: caruntuc@ac.tuiasi.ro

Abstract: Complex systems composed of multiple interconnected sub-systems need to be controlled with specialized control algorithms. In this paper, two classes of control algorithms suitable for such processes are presented. Firstly, two distributed model predictive control (DMPC) strategies with different formulations are described. Afterward, a coalitional control (CC) strategy is proposed, with two different communication topologies, i.e., a default decentralized topology and a distributed topology. All algorithms were tested on the same simulation setup consisting of eight water tanks. The simulation results show that the coalitional control methodology has a similar performance to the distributed algorithms. Moreover, due to its simplified formulation, the former can be easily tested on embedded systems with limited computation storage.

Keywords: distributed model predictive control; coalitional control; networked systems

check for updates

Citation: Maxim, A.; Pauca, O.; Caruntu, C.-F. Distributed Model Predictive Control and Coalitional Control Strategies—Comparative Performance Analysis Using an Eight-Tank Process Case Study. *Actuators* **2023**, *12*, 281. https://doi.org/10.3390/act12070281

Academic Editor: Eihab M. Abdel-Rahman

Received: 24 May 2023
Revised: 23 June 2023
Accepted: 7 July 2023
Published: 10 July 2023

1. Introduction

Distributed model predictive control (DMPC) is a preferred control strategy when dealing with complex systems. Such processes are composed of multiple sub-systems, more often completely or partially interconnected, either physically or through common shared resources or goals [1]. To control such systems, centralized control is not a reliable strategy, due to the sheer size of the computational burden, for solving a unique optimization problem [2]. Decentralized control can be applied only in the particular case of a weak interconnection between sub-systems since, from the control point of view, all of them are independently treated, deliberately ignoring the interdependent connections [3]. Thus, distributed control is a control strategy of compromise between the aforementioned ones, by independently controlling the sub-systems while also taking into account the links between them. The DMPC methodology was developed within the mature model predictive control (MPC) research field [4], in which each sub-system solves a coupled MPC optimization problem, considering both local and inter-shared information.

The subject is ongoing and in fast development, evidenced by extensive research in the DMPC field. During the last decade (i.e., publication years 2013–2023), in the Web Of Science Core Collection, around 1000 DMPC-related papers were published, with more than 500 articles published in prestigious journals such as Annual Reviews in Control, Automatica, IEEE Transactions on Control Systems Technology, Systems & Control Letters and IEEE Control Systems Magazine, among others.

The DMPC strategy was successfully applied in various domains, such as micro-grids [5,6], smart grids [7,8], traffic control [9–14], vehicle platooning [15–18] wind farms [19–21], wastewater treatment plants [22,23], chemical processes [24,25] or network systems [26], just to name a few. In [27], a robust DMPC algorithm for energy management optimization in a multi-microgrid system was presented. The stability of an independent microgrid with respect to the uncertainties introduced by the renewable energy sources was ensured using the advantages of robust MPC optimization. Moreover, a robust DMPC strategy was used to dynamically develop an energy schedule for the multi-microgrid system, using the advantage of power transactions between independent units. In [28], a DMPC approach for the online scheduling involved in the coordination problem between demand response and alternating current optimal power flow in a smart grid was proposed. In [29], a DMPC strategy for high-speed train traffic control was developed. To ensure smaller traveling distances between each train, a virtual coupling was considered, and proofs for feasibility and terminal invariant set constraint stability were provided. In [30], a DMPC approach for a vehicle platoon with two string stability criteria based on $l_\infty-$ and l_2- norms was investigated. In [31], an economic DMPC strategy for a large-scale wind farm was introduced. For each wind turbine, a local Nash optimal solution was reached using an iterative algorithm while also ensuring the dynamic global economic target for the overall wind farm. In [22], an economic DMPC method for a wastewater treatment plant was presented. Two design approaches for the economic DMPC were proposed, with the difference consisting in the model used in the local controller. In one case, for each subsystem, the centralized plant model was used in the optimization problem, whereas the other approach used the corresponding local model defined for each subsystem in the local controller. The simulations performed in various weather conditions showed that the first approach outperforms the second one in terms of control performance. In [32], an explicit DMPC design for chemical processes was introduced. The strategy was used to handle the constraints in a matrix form, while dividing them into two sets. When compared with a classical DMPC, the simulation results obtained for a coke oven pressure control system showed the efficiency of the explicit DMPC formulation. In [33], a robust DMPC for networked control systems with uncertainties and time delays was presented. By decomposing the network optimization problem in multiple optimization sub-problems, each one described using an upper bound robust objective, the computational complexity of the algorithm was decreased.

A comprehensive recent review work on DMPC strategies classified depending on their robustness in the presence of system faults (in sensors and actuators), external cyber-attacks on the communication network or internal attacks from malignant agents inside the network that share false information is given in [34].

A highly cited review work in the DMPC field, which early on envisioned the future research trends for the next decade, is provided in [35]. Furthermore, the DMPC algorithms are classified depending on the optimization problem to be solved as:

- Non-cooperative DMPC—if each agent (or controller) solves a local cost function using both local information from its sub-system and information received from the interconnected sub-systems;
- Cooperative DMPC—if each agent solves a global cost function, taking into account both local information and information received from the entire system.

  Depending on the communication protocols established between different agents, the cooperative architectures are further classified as:

  - Iterative DMPC—if each agent exchanges information with other agents multiple times within a sampling period; to this end, the communication flow is bidirectional.
  - Non-iterative or sequential DMPC—if each agent exchanges information with other agents only once during a sampling period; in this case, the communication flow is unidirectional.

Moreover, based on the topology of the communication network, DMPC methods are categorized as [36]:

- Fully connected DMPC—if each agent is connected with all other agents from the network;
- Partially connected DMPC—if each agent is connected with only a group of agents within the network, called neighbours.

All the above mentioned DMPC strategies have one common denominator, namely that the communication and controller topologies are fixed, i.e., once established in the beginning, they do not change during operation. However, this characteristic is rather restrictive, and thus another methodology was introduced, called coalitional control (CC), using the principle of flexible architecture [37]. In this methodology, rather than having to choose between a fully connected or a partially connected communication network, the idea is that, during operation, in a fully connected topology, certain communication links can be disabled (if they are not necessary), thus obtaining a partially connected topology. A group of agents partially connected (through communication link activation) is called a coalition (or cooperative group), and, within the coalition, a cooperative optimization problem is solved. When the links are disabled, the coalition is dissolved, and the agents solve a non-cooperative optimization problem [38,39].

In this work, we extended the comparative performance analysis provided in [40] for two DMPC methodologies to also include a coalitional control algorithm. The contributions of this work are the following:

- A comprehensive performance analysis was performed for two non-cooperative DMPC algorithms (one formulated using a state-space model, and another formulated using an input–output model) and a CC method, described using a state-space model.
- All three algorithms were tested in simulation on the same process, i.e., the eight-tank process introduced in [40].
- The CC algorithm was based on a matrix gain feedback controller, computed by solving a gradient-based optimization problem. The basic principle of computing the gains was firstly presented in [41].

With respect to our previous works, the following novelties are listed:

- The eight-tank process model introduced in [40] was extended with the nonlinear mathematical description based on Bernoulli's law and the mass balances.
- The DMPC strategies given in [40] are presented in an extended version.
- The gradient-based methodology for computing the gain feedback matrix in the coalitional control framework provided in [41] was reformulated to achieve comparative results with respect to the DMPC strategies. To this end, the feedback gain matrices used in the coalitional control methodology were computed solving a cost function, which minimizes the error between the coalitional state trajectories, with respect to a set of DMPC state trajectories. Moreover, a closed-loop stability constraint was also introduced.
- Two communication topologies were designed for the CC algorithm (with different sets of feedback matrices optimally computed), i.e., a default decentralized communication topology without communication between sub-systems, and a distributed topology with communication links between sub-systems.
- A procedure that automatically switches between the distributed and decentralized communication topologies designed for the coalitional control methodology is introduced.

The remainder of this paper is structured as follows: Section 2 introduces the state-space DMPC algorithm, Section 3 describes the input–output DMPC algorithm, and, in Section 4, the CC algorithm is provided. The process model description, followed by the simulation results and discussion, is given in Section 5. The conclusions and future work ideas are presented in Section 6.

2. DMPC Algorithm with State-Space Model (DMPC$_{\mathrm{SS}}$)

In this section, a non-cooperative DMPC algorithm with velocity-form formulation, designed for a system composed of N sub-systems, is presented. This algorithm was firstly introduced in [42] for a two-agent system and then extended to N sub-systems in [40].

2.1. Problem Formulation

Let us introduce a class of linear-time-invariant (LTI) systems consisting of N sub-systems, interconnected through inputs signals. Each sub-system i, $\forall i \in \mathcal{N}$, with $\mathcal{N}$ the set $\{1, \dots, N\} \subseteq \mathbb{N}$, has the following dynamics:

$$x_{p_i}(k+1) = A_{p_i} x_{p_i}(k) + B_{p_{ii}} u_i(k) + \sum_{j \in \mathcal{N}_i} B_{p_{ij}} u_j(k) \tag{1}$$

$$y_i(k) = C_{p_i} x_{p_i}(k),\ \forall i \in \mathcal{N} \tag{2}$$

with $x_{p_i} \in \mathbb{R}^{n_x}$, $u_i \in \mathbb{R}^{n_u}$, $u_j \in \mathbb{R}^{n_u}$ and $y_i \in \mathbb{R}^{n_y}$ the state, input, coupling inputs and output vectors for the process, respectively; k is the discrete-time index; $A_{p_i}, B_{p_{ii}}, B_{p_{ij}}$ and C_{p_i} are matrices with adequate dimensions. All the sub-systems coupled with sub-system i are included in the set $\mathcal{N}_i = \{j \in \mathcal{N} : B_{p_{ij}} \neq 0\}$. Within this neighbourhood set, between sub-systems i and j, relevant information pertaining to the input vectors is exchanged.

Both input and output vectors are constrained as:

$$u_i \in \mathcal{U}_i,\ y_i \in \mathcal{Y}_i,\ \forall i \in \mathcal{N} \tag{3}$$

where $\mathcal{U}_i$ and $\mathcal{Y}_i$ denote sets of linear inequalities.

As previously mentioned, the proposed DMPC strategy has velocity-form formulation to ensure the presence of an integral action in the control loop. This is achieved using the difference operation on both sides of (1), obtaining:

$$\begin{aligned} \underbrace{x_{p_i}(k+1) - x_{p_i}(k)}_{\Delta x_{p_i}(k+1)} &= A_{p_i} \underbrace{\left(x_{p_i}(k) - x_{p_i}(k-1)\right)}_{\Delta x_{p_i}(k)} + B_{p_{ii}} \underbrace{\left(u_i(k) - u_i(k-1)\right)}_{\Delta u_i(k)} + \\ &+ \sum_{j \in \mathcal{N}_i} B_{p_{ij}} \underbrace{\left(u_j(k) - u_j(k-1)\right)}_{\Delta u_j(k)},\ \forall i \in \mathcal{N} \end{aligned} \tag{4}$$

with the compact form as:

$$\Delta x_{p_i}(k+1) = A_{p_i} \Delta x_{p_i}(k) + B_{p_{ii}} \Delta u_i(k) + \sum_{j \in \mathcal{N}_i} B_{p_{ij}} \Delta u_j(k),\ \forall i \in \mathcal{N} \tag{5}$$

Using the same operation on (2), and substituting (5), we obtain:

$$\begin{aligned} \underbrace{y_i(k+1) - y_i(k)}_{\Delta y_i(k+1)} &= C_{p_i} \Delta x_{p_i}(k+1) \\ &= C_{p_i} \left(A_{p_i} \Delta x_{p_i}(k) + B_{p_{ii}} \Delta u_i(k) + \sum_{j \in \mathcal{N}_i} B_{p_{ij}} \Delta u_j(k) \right),\ \forall i \in \mathcal{N} \end{aligned} \tag{6}$$

The new state variable is selected as $x_i(k) = \left[\Delta x_{p_i}(k)^{\mathrm{T}}\ y_i(k)\right]^{\mathrm{T}}$, obtaining the velocity-form model:

$$\begin{aligned} \underbrace{\begin{bmatrix} \Delta x_{p_i}(k+1) \\ y_i(k+1) \end{bmatrix}}_{x_i(k+1)} &= \underbrace{\begin{bmatrix} A_{p_i} & O \\ C_{p_i} A_{p_i} & I \end{bmatrix}}_{A_i} \underbrace{\begin{bmatrix} \Delta x_{p_i}(k) \\ y_i(k) \end{bmatrix}}_{x_i(k)} \\ &+ \underbrace{\begin{bmatrix} B_{p_{ii}} \\ C_{p_i} B_{p_{ii}} \end{bmatrix}}_{B_{ii}} \Delta u_i(k) + \sum_{j \in \mathcal{N}_i} \underbrace{\begin{bmatrix} B_{p_{ij}} \\ C_{p_i} B_{p_{ij}} \end{bmatrix}}_{B_{ij}} \Delta u_j(k) \\ y_i(k) &= \underbrace{\begin{bmatrix} O & I \end{bmatrix}}_{C_i} \begin{bmatrix} \Delta x_{p_i}(k) \\ y_i(k) \end{bmatrix},\ \forall i \in \mathcal{N} \end{aligned} \tag{7}$$

where I and O are the identity and zero matrix, respectively, with adequate dimensions.

In a compact form, model (7) can be written as:

$$\begin{cases} x_i(k+1) = A_i x_i(k) + B_{ii}\Delta u_i(k) + \sum_{j\in\mathcal{N}_i} B_{ij}\Delta u_j(k) \\ y_i(k) = C_i x_i(k),\ \forall i \in \mathcal{N} \end{cases} \tag{8}$$

where $\Delta u_i(k)$ and $\Delta u_j(k)$, $\forall i \in \mathcal{N}$, $\forall j \in \mathcal{N}_i$, are the inputs in velocity form.

2.2. Optimization Problem

Each agent $\forall i \in \mathcal{N}$ solves the following cost function J_i:

$$J_i(x_i(k), \Delta U_i(k), \{\Delta U_j(k)\}_{j\in\mathcal{N}_i}) = \left(R_{sp_i} - Y_i\right)^{\mathrm{T}}\left(R_{sp_i} - Y_i\right) + \Delta U_i(k)^{\mathrm{T}} R_i \Delta U_i(k) \tag{9}$$

The optimal input sequence

$$\Delta U_i^*(k) = [\Delta u_i^*(k|k) \ \ \ldots \ \Delta u_i^*(k+N_c-1|k)]^{\mathrm{T}}$$

is computed minimizing (9), defined based on the output predictor:

$$Y_i = \left[y_i(k+1|k) \ \ldots \ y_i(k+N_p|k)\right]^{\mathrm{T}},\ \forall i \in \mathcal{N}$$

where N_p is the prediction horizon and $N_c \leq N_p$ is the control horizon. $R_{sp_i} \in \mathbb{R}^{N_p}$ is the predicted reference trajectory, imposed constant over the prediction window, equal to the imposed setpoint at sampling time k. $R_i = \alpha_i I_{N_c}$, $\alpha_i \geq 0$ is the input weight matrix.

The output predictor Y_i is interactively calculated from (8), obtaining the following compact form:

$$Y_i = \tilde{A}_i x_i(k) + \tilde{B}_{ii}\Delta U_i(k) + \sum_{j\in\mathcal{N}_i} \tilde{B}_{ij}\Delta U_j(k) \tag{10}$$

in terms of the current state $x_i(k)$ (and, implicitly, the measured process state $x_{p_i}(k)$), and the input trajectories $\Delta U_i(k)$, $\forall i \in \mathcal{N}$, and $\{\Delta U_j(k)\}_{j\in\mathcal{N}_i}$. $\tilde{A}_i$, $\tilde{B}_{ii}$ and $\tilde{B}_{ij}$ are the predictor matrices.

Explicitly, the cost function to be minimized by each agent $\forall i \in \mathcal{N}$ is:

$$\begin{aligned} &J_i(x_i(k), \Delta U_i, \{\Delta U_j(k)\}_{j\in\mathcal{N}_i}) = \\ &(R_{sp_i} - \tilde{A}_i x_i(k))^{\mathrm{T}}(R_{sp_i} - \tilde{A}_i x_i(k)) + 2\Delta U_i^T \tilde{B}_{ii}^{\mathrm{T}} \sum_{j\in\mathcal{N}_i} \tilde{B}_{ij}\Delta U_j - 2\Delta U_i^{\mathrm{T}} \tilde{B}_{ii}^{\mathrm{T}} [R_{sp_i} - \tilde{A}_i x_i(k)] \\ &\quad - 2\sum_{j\in\mathcal{N}_i} \Delta U_j^{\mathrm{T}} \tilde{B}_{ij}^{\mathrm{T}} [R_{sp_i} - \tilde{A}_i x_i(k)] + 2\Delta U_i^{\mathrm{T}} (\tilde{B}_{ii}^{\mathrm{T}} \tilde{B}_{ii} + R_i)\Delta U_i + \sum_{j\in\mathcal{N}_i} \Delta U_j^{\mathrm{T}} (\tilde{B}_{ij}^{\mathrm{T}} \tilde{B}_{ij})\Delta U_j \end{aligned} \tag{11}$$

obtained by the substitution of (10) in (9). Note that, in (11), the unknown variable is $\Delta U_i(k)$, $\forall i \in \mathcal{N}$, while we consider that $\{\Delta U_j(k)\}_{j\in\mathcal{N}_i}$ is available inside the neighbourhood.

The optimal solution is obtained minimizing (11) subject to (3).

3. DMPC Algorithm with Input–Output Model (DMPC$_{\mathrm{IO}}$)

In this section, a non-cooperative DMPC with an input–output model, designed for a system composed of N sub-systems, is presented. The algorithm was firstly tested on a three-agent system in [43], and extended to N sub-systems in [40].

3.1. Problem Formulation

Let us introduce an LTI system, similar to the one given in Section 2.1, where each sub-system i has the following dynamics:

$$y_i(k) = G_{ii}(q^{-1})u_i(k) + \sum_{j\in\mathcal{N}_i} G_{ij}(q^{-1})u_j(k) + w_i(k) \tag{12}$$

with $u_i \in \mathbb{R}^{n_u}$, $y_i \in \mathbb{R}^{n_y}$ and $w_i \in \mathbb{R}^{n_w}$ the input, output and disturbance vectors, respectively; q^{-1} is the backward shift operator; k denotes the discrete-time index; $G_{ii}(q^{-1})$ and $G_{ij}(q^{-1})$ are discrete-time transfer functions with monic denominators.

All the sub-systems coupled with sub-system i are included in the set $\mathcal{N}_i = \{j \in \mathcal{N} : G_{ij}(q^{-1}) \neq 0\}$. The disturbance term w_i, $\forall i \in \mathcal{N}$ is considered as a white noise signal filtered with an appropriate model [44]. To introduce an integral action in the control loop, the disturbance model was chosen as an integrator:

$$w_i(k) = \frac{C_i(q^{-1})}{D_i(q^{-1})} e_i(k) = \frac{1}{1-q^{-1}} e_i(k) \tag{13}$$

where e_i, $\forall i \in \mathcal{N}$ is a white noise signal.

The input and output vectors are constrained as (3).

3.2. Optimization Problem

Each agent $\forall i \in \mathcal{N}$ solves the following cost function J_i:

$$\begin{aligned} J_i(Y_i(k), U_i(k), \{U_j(k)\}_{j\in\mathcal{N}_i}) = {} & (R_{sp_i}(k) - Y_i(k))^{\mathrm{T}}(R_{sp_i}(k) - Y_i(k)) \\ & + \Delta U_i(k)^{\mathrm{T}} R_i \Delta U_i(k) \end{aligned} \tag{14}$$

where $Y_i(k) = \left[y_i(k+1|k) \ \ldots y_i(k+N_p|k)\right]^{\mathrm{T}}$ is the output predictor; the input sequence $\Delta U_i(k) = [\Delta u_i(k|k) \ \ldots \Delta u_i(k+N_c-1|k)]^{\mathrm{T}}$ is defined as the control increment over the control horizon $N_c \leq N_p$; $R_{sp_i}(k) \in \mathbb{R}^{N_p}$ is the reference trajectory imposed constant over the prediction horizon and equal with the set-point at the current time instant k; $R_i = \alpha_i I_{N_c}$ is the input weight.

The input–output MPC formulation provided in [45], which is the basis for the DMPC implementation, computes the output predictor by aggregating past and future effects:

$$Y_i(k) = \bar{Y}_i(k) + Y_i^{\mathrm{opt}}(k), \tag{15}$$

where $Y_i^{\mathrm{opt}}(k)$ formulated in (16) represents the future actions, while $\bar{Y}_i(k) = X_i(k) + W_i(k)$ represents the past actions $X_i(k)$ and the disturbance prediction $W_i(k)$.

In compact matrix form, $Y_i^{\mathrm{opt}}(k)$ is calculated as:

$$Y_i^{\mathrm{opt}}(k) = \tilde{G}_{ii} U_i(k) + \textstyle\sum_{j\in\mathcal{N}_i} \tilde{G}_{ij} U_j(k), \forall i \in \mathcal{N} \tag{16}$$

with

$$\tilde{G}_{ii} = \begin{bmatrix} h_1^{ii} & 0 & \ldots & g_{1-N_c+1}^{ii} \\ h_2^{ii} & h_1^{ii} & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots \\ h_{N_p}^{ii} & h_{N_p-1}^{ii} & \ldots & g_{N_p-N_c+1}^{ii} \end{bmatrix} \tilde{G}_{ij} = \begin{bmatrix} h_1^{ij} & 0 & \ldots & g_{1-N_c+1}^{ij} \\ h_2^{ij} & h_1^{ij} & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots \\ h_{N_p}^{ij} & h_{N_p-1}^{ij} & \ldots & g_{N_p-N_c+1}^{ij} \end{bmatrix} \tag{17}$$

where $\{h_1^{ij} \ h_2^{ij} \ h_3^{ij} \ldots\}$ are the impulse responses from input j, $\forall j \in \mathcal{N}_i$, to output i, and $g_{N_p-N_c+1}^{ij}$ is the corresponding step response.

Explicitly, the cost function to be minimized by each agent $i, \forall i \in \mathcal{N}$ is:

$$
\begin{aligned}
&J_i(Y_i, U_i, \{U_j\}_{j\in\mathcal{N}_i}) \\
&\quad = ((R_{sp_i} - \bar{Y}_i - \tilde{G}_{ii}U_i - \sum_{j\in\mathcal{N}_i} \tilde{G}_{ij}U_j)^{\mathrm{T}}(R_{sp_i} - \bar{Y}_i - \tilde{G}_{ii}U_i - \sum_{j\in\mathcal{N}_i} \tilde{G}_{ij}U_j) \\
&\qquad + (\bar{A}_iU_i + \bar{b}_i)^{\mathrm{T}}R_i(\bar{A}_iU_i + \bar{b}_i)) \\
&\quad = (U_i^{\mathrm{T}}(\tilde{G}_{ii}^{\mathrm{T}}\tilde{G}_{ii} + \bar{A}_i^{\mathrm{T}}R_i\bar{A}_i)U_i - 2[\tilde{G}_{ii}^{\mathrm{T}}(R_{sp_i} - \bar{Y}_i - \sum_{j\in\mathcal{N}_i} \tilde{G}_{ij}U_j) + \bar{A}_i^{\mathrm{T}}R_i\bar{b}_i]^{\mathrm{T}}U_i \\
&\qquad + (R_{sp_i} - \bar{Y}_i - \sum_{j\in\mathcal{N}_i} \tilde{G}_{ij}U_j)^{\mathrm{T}}(R_{sp_i} - \bar{Y}_i - \sum_{j\in\mathcal{N}_i} \tilde{G}_{ij}U_j) + \bar{b}_i^{\mathrm{T}}R_i\bar{b}_i)
\end{aligned}
\tag{18}
$$

where the incremental variable $\Delta U_i(k)$ is written in matrix form $\Delta U_i = \bar{A}_iU_i + \bar{b}_i$. Matrix $\bar{A}_i$ and vector $\bar{b}_i$ are recursively computed from the formula $\Delta u_i(k|k) = u_i(k|k) - u_i(k-1)$, with $u_i(k-1)$ being the actual input sent to the sub-system at the previous sampling instant.

Note that, in (14), the unknown variable is $U_i(k)$, $\forall i \in \mathcal{N}$, while we consider that $\{U_j(k)\}_{j\in\mathcal{N}_i}$ is available inside the neighbourhood.

The optimal solution $U_i(k)^*$ is obtained minimizing (18) subject to (3).

4. Coalitional Control with Gain Feedback Control (CC)

In this section, a coalitional control algorithm with gain feedback matrix formulation based on a state-space model is presented. The algorithm was firstly introduced in [41]. As previously mentioned, the idea behind the coalitional control is to ensure a degree of flexibility in the control architecture. This is obtained by enabling or disabling certain communication links between different agents, thus obtaining different communication topologies [41].

4.1. Problem Formulation

Consider the LTI system introduced in Section 2.1, where each sub-system i has the dynamics (1) and (2) and the constraints (3).

In the proposed CC strategy, to ensure the presence of an integral action in the control loop, an additional state was introduced. This state was defined as an integral of the control error, denoted $\bar{x}_{p_i}$, and defined as $\bar{x}_{p_i}(k+1) = \bar{x}_{p_i}(k) + r_i(k) - C_{p_i}x_{p_i}(k)$. This additional state was used to extend the state vector, obtaining an extended model:

$$
\begin{aligned}
\underbrace{\begin{bmatrix} x_{p_i}(k+1) \\ \bar{x}_{p_i}(k+1) \end{bmatrix}}_{x_i(k+1)} &= \underbrace{\begin{bmatrix} A_{p_i} & O \\ -C_{p_i} & I \end{bmatrix}}_{A_i} \underbrace{\begin{bmatrix} x_{p_i}(k) \\ \bar{x}_{p_i}(k) \end{bmatrix}}_{x_i(k)} + \underbrace{\begin{bmatrix} O \\ I \end{bmatrix}}_{R_{sp_i}} r_i(k) \\
&\quad + \underbrace{\begin{bmatrix} B_{p_{ii}} \\ O \end{bmatrix}}_{B_{ii}} u_i(k) \quad + \textstyle\sum_{j\in\mathcal{N}_i} \underbrace{\begin{bmatrix} B_{p_{ij}} \\ O \end{bmatrix}}_{B_{ij}} u_j(k)
\end{aligned}
\tag{19}
$$

$$
y_i(k) = \underbrace{\begin{bmatrix} C_{p_i} & O \end{bmatrix}}_{C_i} \begin{bmatrix} x_{p_i}(k) \\ \bar{x}_{p_i}(k) \end{bmatrix}, \forall i \in \mathcal{N}
\tag{20}
$$

where I and O are the identity and zero matrix, respectively, with adequate dimensions.

In a compact form, model (19) and (20) can be written as:

$$
\begin{cases}
x_i(k+1) = A_ix_i(k) + B_{sp_i}r_i(k) + B_{ii}u_i(k) + \sum_{j\in\mathcal{N}_i} B_{ij}u_j(k) \\
y_i(k) = C_ix_i(k), \; \forall i \in \mathcal{N}
\end{cases}
\tag{21}
$$

where $u_i(k)$ and $u_j(k)$, $\forall i \in \mathcal{N}$, $\forall j \in \mathcal{N}_i$, are the input and the coupling input, respectively.

4.2. Optimization Problem

In the proposed coalitional control strategy, each agent $\forall i \in \mathcal{N}$ is controlled using a state feedback gain matrix. Within the methodology, a given communication topology will have a particular form for the corresponding overall gain matrix (comprising all individual feedback matrices, correlated to each sub-system). As such, in the initialization phase of the methodology, one must decide the communication topologies that will be employed in the coalitional control. The difference between different topologies is the uni-directional communication links that are enabled, thus resulting in different overall gain feedback matrices.

Hereafter, we will formulate the following communication topologies:

1. A decentralized topology, where the control action of the sub-systems is computed without external information; thus, all the communication links are disabled;
2. A distributed topology, where the control action of the sub-systems is computed using relevant external information from the neighbours. This means that the communication links between neighbours are enabled.

In all tests, for each sub-system, the control action is obtained using the gain feedback matrix formulation obtained as an optimal solution that minimizes the difference between the DMPC algorithm and the feedback gain matrix solution.

Each feedback gain matrix K, corresponding to each communication topology, is computed by solving the following cost function using gradient optimization:

$$J(K) = \sum_{x_i^{DMPC} \in X_{DMPC}} \sum_{i=1}^{\mathcal{N}} J_{x_i^{DMPC}}(K) \tag{22}$$

with

$$J_{x_i^{DMPC}}(K) = \sum_{j=1}^{M} \|x_i(j) - x_i^{DMPC}(j)\|_2^2, \tag{23}$$

s.t. (21), (3),

$$\max(|\text{eig}(A_i + B_{ii}K_{i,i})|) < 1 \tag{24}$$

$$\text{with } u_i(k) = K_{i,i}x_i(k). \tag{25}$$

where X_{DMPC} is a set of state trajectories denoted x_i^{DMPC}, $\forall i \in \mathcal{N}$, obtained from the DMPC$_{SS}$ algorithm, simulated for M time samples.

The overall gain feedback matrix K is the optimal solution of problem (22).

Within the optimization, to compute the matrix K, a cost index is defined as the error between the state trajectory x_i^{DMPC} chosen as an imposed reference for the state trajectories x_i obtained using the control law (25) corresponding to the decentralized communication topology. In this manner, we ensure that the closed-loop dynamics obtained using the coalitional control strategy are similar to the closed-loop dynamics from DMPC$_{SS}$ (i.e., we consider the response generated by the DMPC$_{SS}$ strategy to be the desired response for our coalitional control method). Moreover, note that constraint (24) ensures that all eigenvalues (computed with Matlab function `eig.m`) of the closed-loop system are within the unit circle, i.e., the closed-loop stability is satisfied, with the control law based on the feedback gain matrix $K_{i,i}$, $\forall i \in \mathcal{N}$.

The set X_{DMPC} contains manifold state trajectories obtained by testing the process in multiple operating points feasible for the process functionality (i.e., respecting the imposed hard constraints (3)). Using this set ensures that no bias from a particular simulation case influences the computation of the optimal overall gain matrix K.

Since we wished to compare the distributed results obtained with the DMPC$_{SS}$ strategy with the coalitional ones, a distributed communication topology was defined taking into account the physical coupling between sub-systems. It resulted in an optimal feedback matrix K, which has elements $K_{i,i}$, $\forall i \in \mathcal{N}$, on the main diagonal, corresponding to each sub-system and elements off-diagonal $K_{i,j}$, $\forall i,j \in \mathcal{N}$, $\forall j \in \mathcal{N}_i$, corresponding to the communication links enabled between neighbours.

The overall gain matrix K for the distributed topology was computed by minimizing the same cost function (22), where (25) was rewritten as $u_i(k) = K_{i,i}x_i(k) + K_{i,j}x_j(k)$, and (24) was rewritten as $\max(|\mathrm{eig}(A_i + B_{ii}K_{i,i} + B_{ij}K_{i,j})|) < 1$, so that the interaction between neighbours is considered.

Note that, for the proposed coalitional control strategy, we designed two communication topologies. From the coalitional point of view, these two case studies can be regarded as: (i) the default test without coalitions, where the sub-systems do not exchange information, and the overall gain matrix is diagonal, and (ii) the test with uni-directional coalitions only between each two neighbours, which are coupled directly through inputs. In this case, the overall gain matrix has only one non-zero element on each row, placed off-diagonal.

As previously mentioned, the main advantage of the proposed coalitional control methodology is to minimize the communication burden of the algorithm. This is managed by opening additional communication links only when needed. In this framework, a coalitional control strategy with switching communication topologies was designed, in which the sub-systems can work either in a decentralized or in a distributed manner.

An important aspect of the coalitional control test is the criteria that switching between the two topologies are based on. In our case, we decided on a time-based framework in which, during the simulation, at each T sample times, each communication topology was re-evaluated (i.e., a cost index was computed). The evaluation was performed for the next T samples horizon, starting from the current initial conditions (i.e., similar with the receding horizon principle in DMPC). The topology that has the 'future' smallest cumulative cost was used for the next T sample times.

Let us denote with $J_{\text{dist}}(K)$ the cumulative cost for the distributed communication topology, computed as follows:

$$J_{\text{dist}}(K) = \sum_{i=1}^{\mathcal{N}} J_{x_i}(K_i) \tag{26}$$

with

$$J_{x_i}(K_i) = \sum_{j=1}^{T} \|r_i(k+j) - C_i x_i(k+j)\|_2^2 + \beta\|u_i(k+j)\|_2^2 + \gamma|K_i| \tag{27}$$

s.t. (21), (3),

$$\text{with } u_i(k) = K_{i,i}x_i(k) + K_{i,j}x_j(k) \tag{28}$$

where $|K_i|$ denotes the number of off-diagonal, non-zero elements from gain matrix K_i corresponding to sub-system i. The weight γ is selected by the user, and influences the importance given to the communication cost involved within a given topology (i.e., to provide a balance between performance and the number of enabled communication links).

In an analogous manner, the cumulative cost for the decentralized communication topology $J_{\text{dec}}(K)$ can be computed using (26) by replacing (28) with (25) and selecting $\gamma = 0$, since no communication links are opened.

5. Numerical Analysis on an Eight-Tank Process

The proposed control strategies (i.e., DMPC$_{\text{SS}}$, DMPC$_{\text{IO}}$ and CC) were tested in simulation on a process consisting of eight interconnected water tanks.

5.1. Process Description

Let us introduce a benchmark process that can be decomposed into four input-coupled sub-systems. Namely, two quadruple-tank processes, described in [46] (consisting of two sub-systems each) were connected in a circular architecture (i.e., sub-system 1 coupled with sub-system 4, which is coupled with sub-system 3, which is coupled with sub-system 2, which is coupled with sub-system 1), obtaining an eight-tank process, introduced in [40]. In Figure 1 (from [40]), the schematic diagram of the eight-tank process is provided. For this process, the idea is to control the water level in the lower tanks (L_2, L_4, L_6, L_8) by manipulating the corresponding water flows (i.e., implicitly, by changing the voltages of the four pumps V_{p1}, V_{p2}, V_{p3}, V_{p4}). Note that, the sub-systems are coupled through the inputs (marked in Figure 1 with dashed coloured lines). Thus, a percentage of the water flow provided by pump V_{p1} from sub-system 1, influences the water level L_4 from sub-system 2 (see the water flow marked with red dashed arrow).

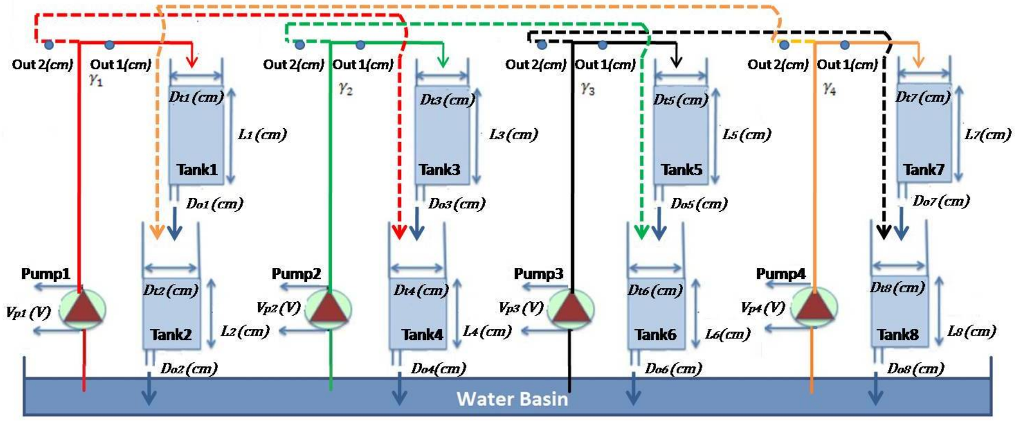

Figure 1. Schematic diagram of the eight-tank process [40].

The nonlinear mathematical model corresponding to sub-system 1 (ensemble of two water tanks, denoted Tank 1 (upper level) and Tank 2 (lower level)) is described using the Bernoulli's law and the mass balances, obtaining:

$$\frac{dL_2}{dt} = \underbrace{\frac{(1-\gamma_4)k_p}{A_{t2}}}_{a_4} V_{p4} - \underbrace{\frac{A_{o2}}{A_{t2}}}_{D_2} \sqrt{2gL_2} + \underbrace{\frac{A_{o1}}{A_{t2}}}_{D_1} \sqrt{2gL_1} \tag{29}$$

$$\frac{dL_1}{dt} = \underbrace{\frac{\gamma_1 k_p}{A_{t1}}}_{b_1} V_{p1} - \underbrace{\frac{A_{o1}}{A_{t1}}}_{D_1} \sqrt{2gL_1} \tag{30}$$

where $g = 981\ \text{cm/s}^2$ is the gravitational constant on Earth, and $A_{oi} = \pi\frac{D_{oi}^2}{4}\ \text{cm}^2$ and $A_{ti} = \pi\frac{D_{ti}^2}{4}\ \text{cm}^2$ are the cross-section of the outflow orifice and the cross-section of Tank i, $i = \{1,2\}$, respectively. The voltage applied to Pump i, $i = \{1,4\}$, is V_{pi} and the corresponding flow is $k_p V_{pi}$. The parameters $\gamma_i \in (0,1)$, $i = \{1,4\}$ represent the percentages of the flow from Pump i through inlets Out 1 and Out 2, respectively, and are defined as:

$$\gamma_1 = \frac{A_{i1}}{(A_{i1}+A_{i2})},\ \gamma_4 = \frac{A_{i7}}{(A_{i7}+A_{i8})} \tag{31}$$

where $A_{i1} = A_{i7} = \frac{\pi\text{Out1}^2}{4}\ \text{cm}^2$ and $A_{i2} = A_{i8} = \frac{\pi\text{Out2}^2}{4}\ \text{cm}^2$ are the upper and lower tanks inlet areas. The numerical values for the set-up parameters are derived from the user manual for the quadruple tank process provided by Quanser and are given in Table 1. Note that sub-system 1 defined with (29) and (30) is coupled with sub-system 4 through input Pump 4, since the water level L_2 depends on the flow $k_p V_{p4}$, which is the control input in sub-system 4. The water level L_1 for the upper tank Tank 1 depends on the flow provided by Pump 1, e.g., $k_p V_{p1}$ (see Figure 1).

Following this reasoning and the schematic diagram of the process, which indicates the interconnection between sub-systems, the remaining models for sub-systems 2, 3 and 4 can be easily derived.

The nonlinear sub-system's model was linearized in Taylor expansion in the desired equilibrium value for the lower tank level (i.e., $L_{20} = 10$ cm). Same equilibrium point values were used for sub-systems 2, 3 and 4.

The process states were chosen as deviations from the equilibrium point $x_i := L_i - L_{i0}$, $i = \{1, \ldots, 8\}$, (i.e., the upper tanks equilibrium points were chosen as: $L_{10} = 3.69$ cm, $L_{30} = 6.76$ cm, $L_{50} = 2.89$ cm and $L_{70} = 4.86$ cm). The inputs variables were defined also as deviations $u_i := V_{pi} - V_{pi0}$, $i = \{1, \ldots, 4\}$, (i.e., with the equilibrium values $V_{p10} = 3.73$ V, $V_{p20} = 9.71$ V, $V_{p30} = 6.35$ V and $V_{p40} = 8.24$ V).

Table 1. Eight-tank process from Quanser model parameters.

Variable	Value	Unit	Description
Out 1	0.635	cm	"Out 1" Orifice diameter
Out 2	0.476	cm	"Out 2" Orifice diameter
D_{ti}	4.445	cm	Inner diameter Tank i, $i \in \{1, \ldots, 8\}$
D_{oi}	0.476	cm	Outlet diameter Tank i, $i \in \{1, \ldots, 8\}$
γ_i	0.6402	-	Flow ratio parameter for Pump i, $i \in \{1, \ldots, 4\}$
A_{i1}, A_{i3}, A_{i5}, A_{i7}	0.316	cm^2	Inlet area Tank i, $i \in \{1, 3, 5, 7\}$
A_{i2}, A_{i4}, A_{i6}, A_{i8}	0.178	cm^2	Inlet area Tank i, $i \in \{2, 4, 6, 8\}$
A_{ti}	15.517	cm^2	Inside cross-section area Tank i, $i \in \{1, \ldots, 8\}$
A_{oi}	0.178	cm^2	Outlet area Tank i, $i \in \{1, \ldots, 8\}$
k_p	3.3	cm^3/s/V	Pump flow constant
g	981	cm/s^2	Gravitational constant on Earth

Further on, after the linerization procedure, we obtained the following overall linearized state-space model for the eight-tank process:

$$\dot{x} = \underbrace{\begin{bmatrix} -\eta_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \eta_1 & -\eta_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\eta_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \eta_3 & -\eta_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\eta_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \eta_5 & -\eta_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\eta_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \eta_7 & -\eta_8 \end{bmatrix}}_{\bar{A}_c} x + \underbrace{\begin{bmatrix} b_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_4 \\ 0 & b_2 & 0 & 0 \\ a_1 & 0 & 0 & 0 \\ 0 & 0 & b_3 & 0 \\ 0 & a_2 & 0 & 0 \\ 0 & 0 & 0 & b_4 \\ 0 & 0 & a_3 & 0 \end{bmatrix}}_{\bar{B}_c} u, \tag{32}$$

$$y = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\bar{C}_c} x$$

where $x = [x_1 \ldots x_8]^{\mathrm{T}}$ is the state vector, $u = [u_1 \ldots u_4]^{\mathrm{T}}$ is the input vector and $y = [y_1 \ldots y_4]^{\mathrm{T}}$ is the output vector. The parameters $\eta_i = \frac{D_i \sqrt{2g}}{2\sqrt{L_{i0}}}$, $i \in \{1 \ldots 8\}$ were computed with partial derivatives.

By replacing all the numerical values provided in Table 1, we obtained the following system matrices:

$$
\begin{aligned}
\bar{A}_c &= \begin{bmatrix} -0.13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.13 & -0.08 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.09 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.09 & -0.08 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.14 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.14 & -0.08 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.11 & -0.08 \end{bmatrix} \\
\bar{B}_c &= \begin{bmatrix} 0.13 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.07 \\ 0 & 0.13 & 0 & 0 \\ 0.07 & 0 & 0 & 0 \\ 0 & 0 & 0.13 & 0 \\ 0 & 0.07 & 0 & 0 \\ 0 & 0 & 0 & 0.13 \\ 0 & 0 & 0.07 & 0 \end{bmatrix}
\end{aligned} \tag{33}
$$

The overall state-space continuous time model (32) was discretized with the sampling period $T_s = 1$ s using the MATLAB function `c2d.m`, and the discretization method zero-order-hold, obtaining:

$$
\begin{aligned}
x_d(k+1) &= \bar{A}_d x_d(k) + \bar{B}_d u_d(k) \\
y_d(k) &= \bar{C}_d x_d(k)
\end{aligned} \tag{34}
$$

where $\bar{A}_d$, $\bar{B}_d$ and $\bar{C}_d$ are the discrete-time counterparts for the continuous-time system matrices from (32).

Next, the system was decomposed into four input-coupled sub-systems, hereafter denoted by S_i, $i \in \{1, \ldots, 4\}$, with the following components:

$$
\begin{aligned}
S_1 : \begin{cases} x_{S_1} = [x_{d1}\ x_{d2}]^T \\ u_{S_1} = u_1 \\ \mathcal{N}_{S_1} = \{4\} \\ y_{S_1} = x_{d2} \end{cases} \quad & S_2 : \begin{cases} x_{S_2} = [x_{d3}\ x_{d4}]^T \\ u_{S_2} = u_2 \\ \mathcal{N}_{S_2} = \{1\} \\ y_{S_2} = x_{d4} \end{cases} \\
S_3 : \begin{cases} x_{S_3} = [x_{d5}\ x_{d6}]^T \\ u_{S_3} = u_3 \\ \mathcal{N}_{S_3} = \{2\} \\ y_{S_3} = x_{d6} \end{cases} \quad & S_4 : \begin{cases} x_{S_4} = [x_{d7}\ x_{d8}]^T \\ u_{S_4} = u_4 \\ \mathcal{N}_{S_4} = \{3\} \\ y_{S_4} = x_{d8} \end{cases}
\end{aligned} \tag{35}
$$

where x_{S_1}, u_{S_1}, $\mathcal{N}_{S_1}$ and y_{S_1} are the states, input, neighbourhood set and output for S_1, respectively. Similar definitions correspond to sub-systems S_2, S_3 and S_4.

With the state, input and output partitions given in (35), the discrete-time matrices of sub-systems S_i, $i \in \{1, \ldots, 4\}$, are the following:

$$
\begin{aligned}
S_1 &: \left\{ \bar{A}_{d1} = \begin{bmatrix} 0.8761 & 0 \\ 0.1189 & 0.9227 \end{bmatrix} \bar{B}_{d11} = \begin{bmatrix} 0.1275 \\ 0.0084 \end{bmatrix} \bar{B}_{d14} = \begin{bmatrix} 0 \\ 0.0735 \end{bmatrix} \bar{C}_{d1} = \begin{bmatrix} 0 & 1 \end{bmatrix} \right. \\
S_2 &: \left\{ \bar{A}_{d2} = \begin{bmatrix} 0.9069 & 0 \\ 0.0894 & 0.9227 \end{bmatrix} \bar{B}_{d22} = \begin{bmatrix} 0.1297 \\ 0.0063 \end{bmatrix} \bar{B}_{d21} = \begin{bmatrix} 0 \\ 0.0735 \end{bmatrix} \bar{C}_{d2} = \begin{bmatrix} 0 & 1 \end{bmatrix} \right. \\
S_3 &: \left\{ \bar{A}_{d3} = \begin{bmatrix} 0.8612 & 0 \\ 0.1333 & 0.9227 \end{bmatrix} \bar{B}_{d33} = \begin{bmatrix} 0.1265 \\ 0.0094 \end{bmatrix} \bar{B}_{d32} = \begin{bmatrix} 0 \\ 0.0735 \end{bmatrix} \bar{C}_{d3} = \begin{bmatrix} 0 & 1 \end{bmatrix} \right. \\
S_4 &: \left\{ \bar{A}_{d4} = \begin{bmatrix} 0.8912 & 0 \\ 0.1045 & 0.9227 \end{bmatrix} \bar{B}_{d44} = \begin{bmatrix} 0.1286 \\ 0.0074 \end{bmatrix} \bar{B}_{d43} = \begin{bmatrix} 0 \\ 0.0735 \end{bmatrix} \bar{C}_{d4} = \begin{bmatrix} 0 & 1 \end{bmatrix} \right.
\end{aligned} \tag{36}
$$

Each sub-system S_i, $i \in \{1, \ldots, 4\}$, with the state-space model matrices given in (36), was converted to a minimal realization of its corresponding transfer function form using the MATLAB functions `ss2tf.m` and `minreal.m`, obtaining:

$$
\begin{aligned}
S_1 : \bar{G}_{d11} &= \frac{0.00839q^{-1} + 0.007816q^{-2}}{1 - 1.799q^{-1} + 0.8084q^{-2}} & \bar{G}_{d14} &= \frac{0.07351q^{-1}}{1 - 0.9227q^{-1}} \\
S_2 : \bar{G}_{d22} &= \frac{0.006274q^{-1} + 0.005912q^{-2}}{1 - 1.83q^{-1} + 0.8368q^{-2}} & \bar{G}_{d21} &= \frac{0.07351q^{-1}}{1 - 0.9227q^{-1}} \\
S_3 : \bar{G}_{d33} &= \frac{0.009428q^{-1} + 0.008733q^{-2}}{1 - 1.784q^{-1} + 0.7946q^{-2}} & \bar{G}_{d32} &= \frac{0.07351q^{-1}}{1 - 0.9227q^{-1}} \\
S_4 : \bar{G}_{d44} &= \frac{0.007351q^{-1} + 0.006887q^{-2}}{1 - 1.814q^{-1} + 0.8223q^{-1}} & \bar{G}_{d43} &= \frac{0.07351q^{-1}}{1 - 0.9227q^{-1}}
\end{aligned}
\tag{37}
$$

Since DMPC$_{\text{SS}}$ has a velocity-form formulation, each sub-system S_i, $i \in \{1, \ldots, 4\}$, with the state-space model matrices given in (36), was converted to the augmented state-space model (8). Moreover, since the CC algorithm has an extended model with an integrator, each sub-system S_i, $i \in \{1, \ldots, 4\}$, with the state-space model matrices given in (36). was converted to the extended state-space model (21).

5.2. Simulation Results

The proposed DMPC and CC strategies have the following optimization parameters and constraint limits:

- The sampling period $T_s = 1$ s, the prediction horizon N_p = 30 samples and the control horizon N_c = 30 samples;
- The input weight matrices $R_i = \alpha I_{N_c}$, with $\alpha = 10$, $\forall i \in \{1, \ldots, 4\}$.
- The input weight $\beta = 0.01$, the communication cost $\gamma = 0.01$ and the horizon $T = 20$ samples.
- The input constraints are $0 \text{ V} \leq u_i \leq 22 \text{ V}$, $\forall i \in \{1, \ldots, 4\}$;
- The output constraints are $0 \text{ cm} \leq y_i \leq 25 \text{ cm}$, $\forall i \in \{1, \ldots, 4\}$.

All proposed methodologies were compared in a setpoint tracking test, performed on the eight-tank process described in Section 5.1. The test had a length of $M = 1000$ s and was designed as a series of step changes as follows:

- During the first 200 s, all references r_i for all sub-systems S_i, $i \in 1, \ldots, 4$ are equal to 5 cm.
- At time 201 s, the references values are: $r_1 = 8$ cm, $r_2 = 10$ cm, $r_3 = 12$ cm and $r_4 = 15$ cm.
- At time 401 s, the references values are: $r_1 = 15$ cm, $r_2 = 12$ cm, $r_3 = 10$ cm and $r_4 = 15$ cm.
- At time 601 s, the references values are: $r_1 = 10$ cm, $r_2 = 15$ cm, $r_3 = 15$ cm and $r_4 = 12$ cm.
- At time 801 s, the references values are: $r_1 = 10$ cm, $r_2 = 20$ cm, $r_3 = 15$ cm and $r_4 = 15$ cm.

Remark 1. *For the DMPC strategies, the numerical values for the optimization parameters were empirically chosen, after several numerical simulations, taking into account various factors such as: the open-loop dynamics of the process, the compromise between a good closed-loop performance and small control effort, etc.*

The prediction horizon N_p was selected as large enough such that the prediction will cover part of the transient response of the open-loop sub-system. However, a larger prediction horizon will result in a slower closed-loop response, with the benefit of a smaller control effort.

The input weight matrix R_i was chosen as a compromise between a good tracking error and smaller control effort. A smaller value will put more emphasis on the minimization of the tracking error at the detriment of the value of the control effort. Taking into account that the used process is hard-constrained in the input values, it makes more sense to influence the optimization toward the minimization of the input, and the second priority is given to the tracking error.

Remark 2. *For the CC strategy with switching topologies, the values for the parameters from the cumulative cost* (26) *used for the evaluation of the topologies were also empirically chosen, after several tests.*

Similar to the prediction horizon parameter from the DMPC, the value of the horizon T was selected as large enough to cover part of the transient response of the open-loop system. A larger value for the horizon T will influence the switching rate between topologies.

The weight γ was selected taking into account that the decentralized topology has $\gamma = 0$ (i.e., no links enabled). This results in a non-zero, positive value influencing the evaluation result with respect to the cumulative cost corresponding to the distributed topology. A larger value can excessively penalize the communication, forcing only the activation of the decentralized topology.

The comparative simulation results for the DMPC_{SS} and DMPC_{IO} strategies are given in Figures 2 and 3, depicting the outputs and inputs, respectively. As expected, despite the fact that these two DMPC algorithms have different implementations, using the same optimization parameters and in identical simulation conditions, we obtained quasi-indistinguishable transient performances. This is because the distributed methodologies are similar, exchanging the optimal input between coupled sub-systems.

Next, the decentralized $\text{CC}_{\text{K dec}}$ and the distributed $\text{CC}_{\text{K dist}}$ communication topologies designed for the coalitional control strategy were comparatively tested in the same simulation scenario. The results obtained are given in Figures 4 and 5, depicting the outputs and inputs, respectively. As previously mentioned, within the decentralized formulation, there are no communication links enabled between coupled sub-systems.

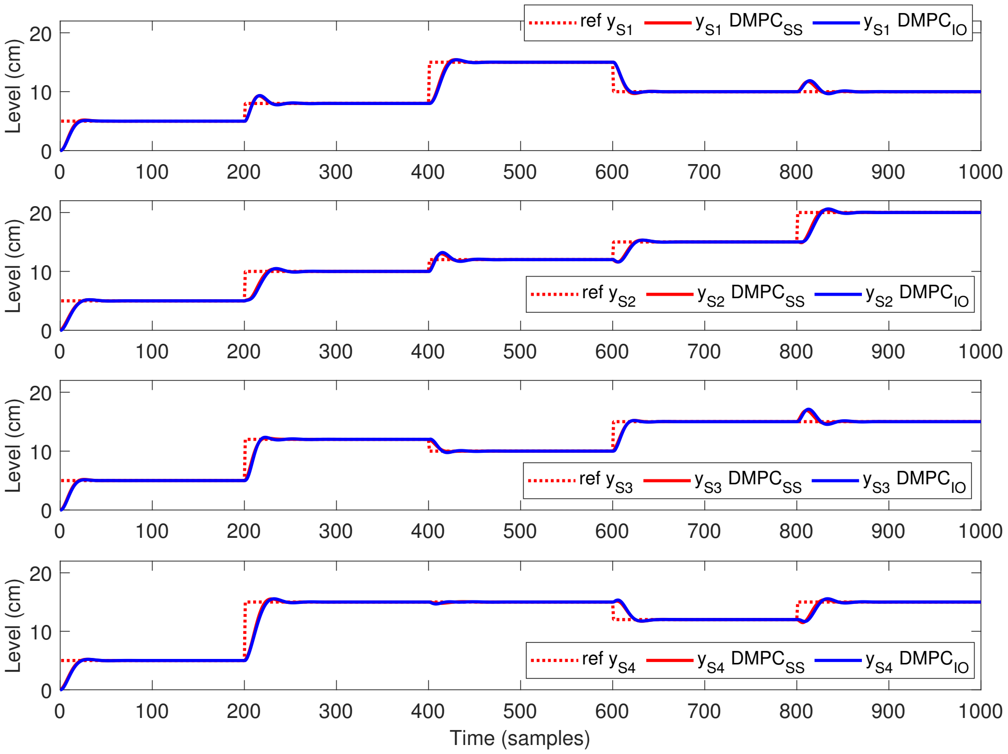

Figure 2. Comparative simulation results for DMPC_{SS} (red lines) and DMPC_{IO} (blue lines) strategies—outputs for all sub-systems.

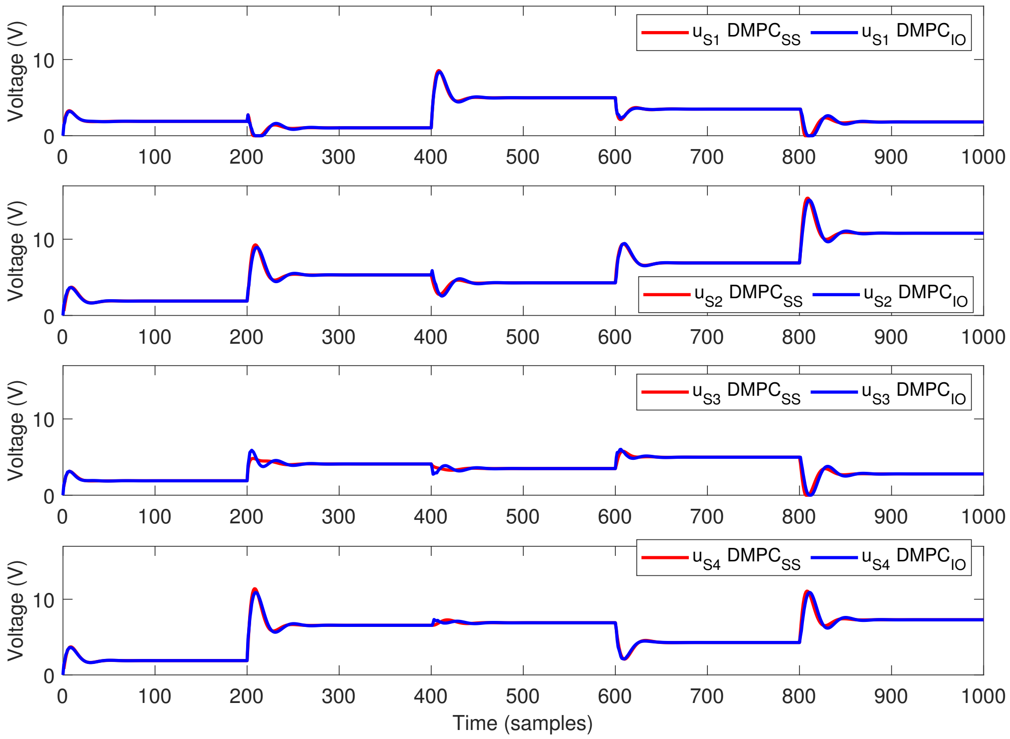

Figure 3. Comparative simulation results for $DMPC_{SS}$ (red lines) and $DMPC_{IO}$ (blue lines) strategies—inputs for all sub-systems.

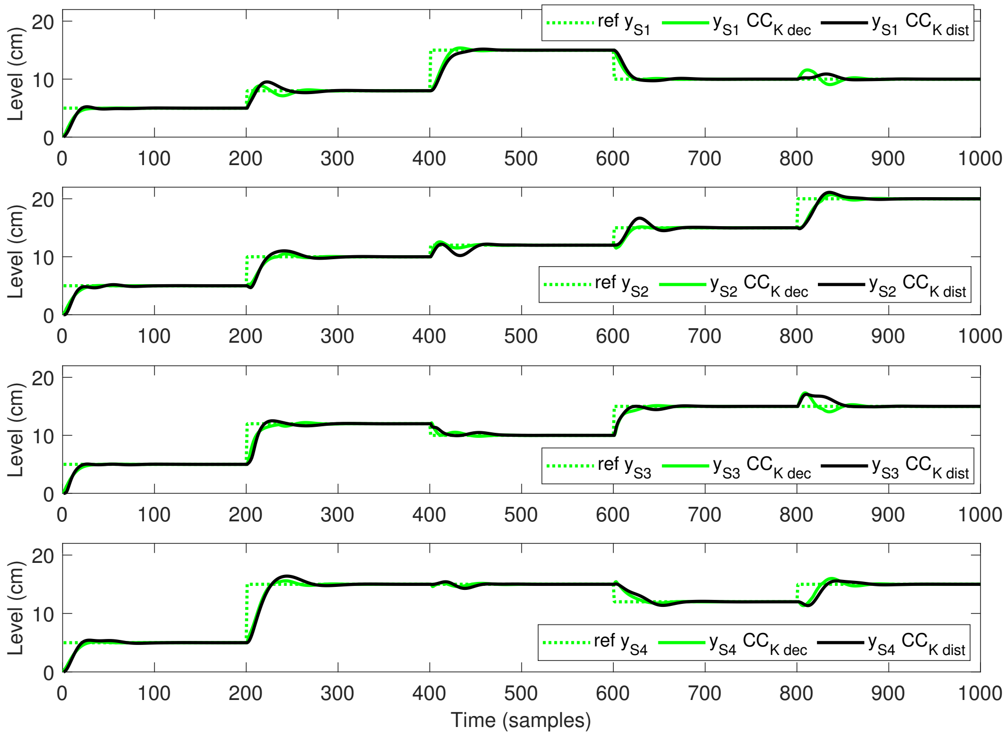

Figure 4. Comparative simulation results for $CC_{K\,dec}$ (green lines) and $CC_{K\,dist}$ (black lines) strategies—outputs for all sub-systems.

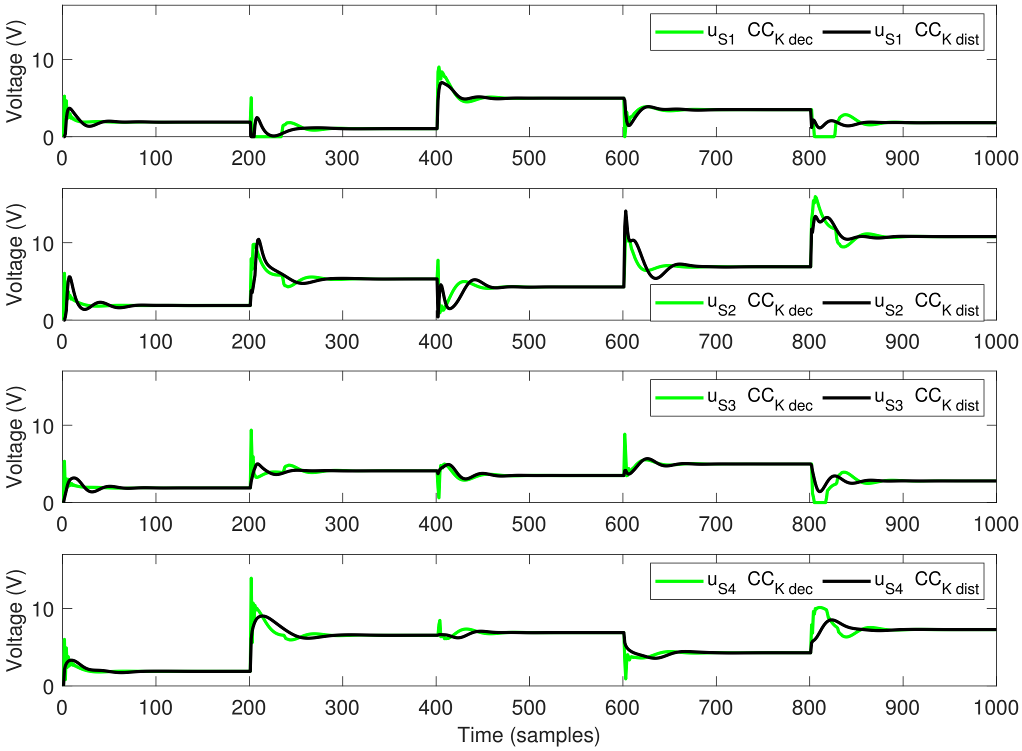

Figure 5. Comparative simulation results for $CC_{K\,dec}$ (green lines) and $CC_{K\,dist}$ (black lines) strategies—inputs for all sub-systems.

For this reason, one can see that the control effort is more aggressive during the transient time when compared with the distributed topology (see Figure 5, at time 600 samples). Because, in the latter, there are communication links opened between coupled sub-systems, it results in a smoother output response.

Moreover, the strength of the proposed coalitional control methodology is the dynamical configuration of the communication topology. Thus, the next step in our analysis was to test the efficiency of the algorithm by automatically switching between the decentralized and distributed communication topologies.

The obtained results are presented in Figures 6 and 7, depicting the outputs and inputs, respectively. In Figure 8, the switching times between the two topologies are presented. It is interesting to notice in this figure that the distributed topology is activated when the need for coupling information is more stringent to ensure a better response. Thus, between time 0 samples and time 390 samples, the topology is decentralized. When the simulation conditions are more challenging (see Figure 6, in the interval 390–600 samples and 790–1000 samples), the communication topology switches to distributed and shares information between sub-systems. This is partially due to the fact that sub-systems S_2 and S_3 are coupled and have opposite setpoint changes.

Another remark is the fact that, for this setup, if a decrease in the water level in a tank is desired, this results in a decrease in the water flow, and implicitly a lower pump voltage. However, if the coupling sub-system has a significant water level increase, due to the physical coupling between sub-systems, this can evolve to a pump saturation on the lower limit of 0 volts (see Figure 7 at time 200 samples for sub-system S_1).

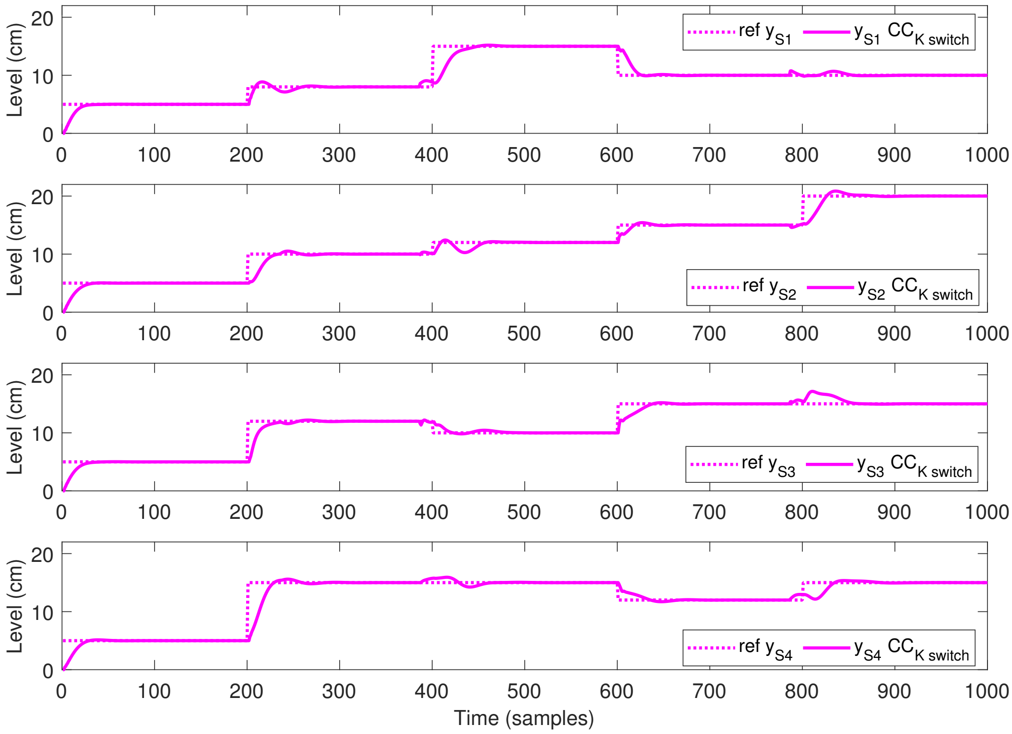

Figure 6. Simulation results for $CC_{K\ switch}$ strategy—outputs for all sub-systems.

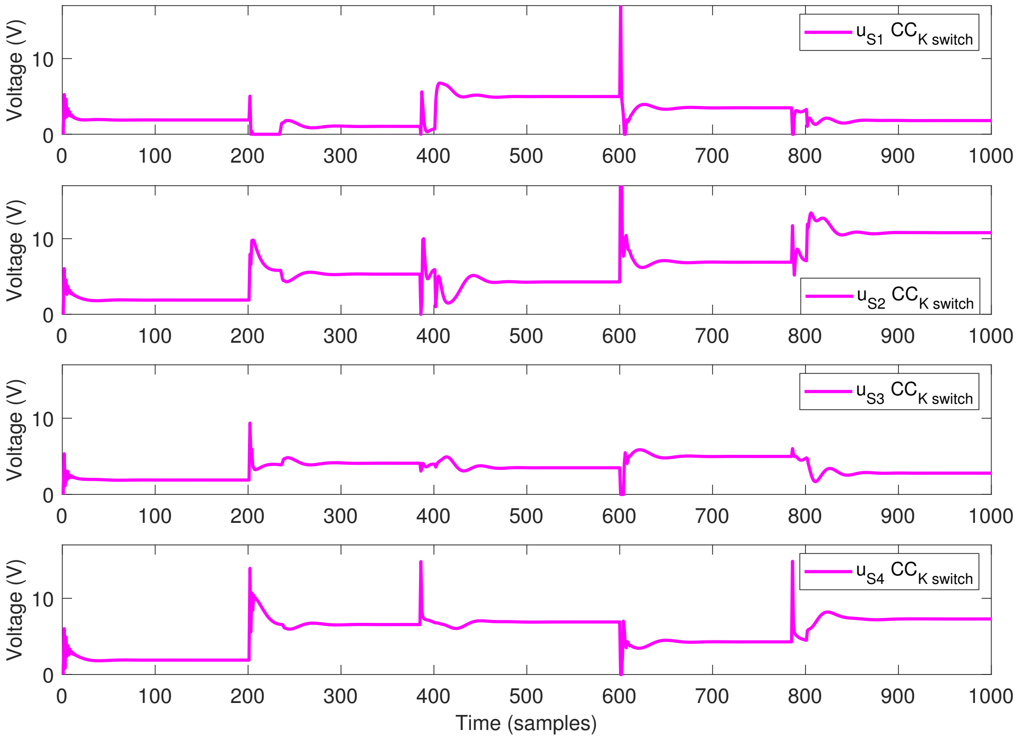

Figure 7. Simulation results for $CC_{K\ switch}$ strategy—inputs for all sub-systems.

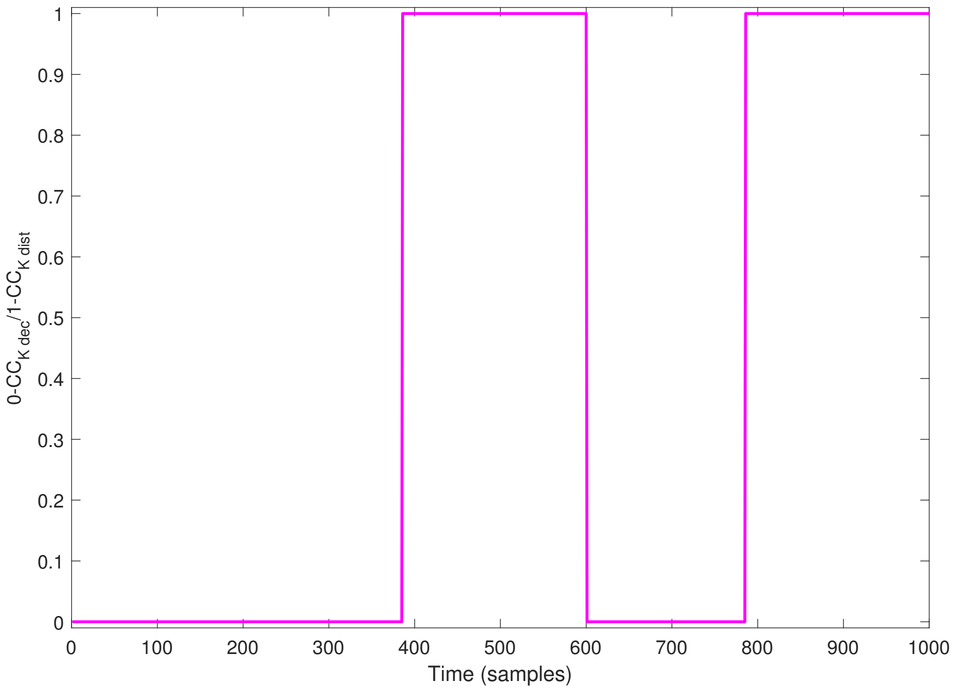

Figure 8. Switching dynamics for $\text{CC}_{\text{K switch}}$ strategy—1 corresponds to $\text{CC}_{\text{K dist}}$, whereas 0 corresponds to $\text{CC}_{\text{K dec}}$.

5.3. Discussion

The performance of the proposed strategies was analyzed with respect to the following performance index:

$$J_{\text{cost}} = \frac{1}{M} \sum_{k=1}^{M} \sum_{i=1}^{4} \left(r_i(k) - y_i(k) \right)^2 + \beta u_i(k)^2 \Big) \tag{38}$$

where M is the length of the simulation time and $y_i(k)$, $r_i(k)$ and $u_i(k)$ are the measured output, the imposed reference and the computed input of sub-system S_i, $\forall i \in \{1, \ldots, 4\}$, at sample time k. As the numerical values given in Table 2 show, the DMPC_{SS} has a slightly smaller cost index than the DMPC_{IO}. When comparing the coalitional strategies using the same criteria, as expected, it results in the coalitional control with the switching communication topology $\text{CC}_{\text{K switch}}$ outperforming the other two CC strategies, with the smallest J_{cost}.

Table 2. Comparative analysis for DMPC_{SS}, DMPC_{IO}, $\text{CC}_{\text{K dist}}$, $\text{CC}_{\text{K dec}}$ and $\text{CC}_{\text{K switch}}$ algorithms based on performance index J_{cost}, overshoot (σ) and settling time (tt).

Algorithm	J_{cost}	σ (%)	tt (s)
DMPC_{SS}	4.6103	3.9102	33
DMPC_{IO}	5.0120	2.3250	31
$\text{CC}_{\text{K dec}}$	4.4757	0	29
$\text{CC}_{\text{K dist}}$	5.4070	4.6806	54
$\text{CC}_{\text{K switch}}$	4.4682	0	30

What is noteworthy is the fact that, from this cost analysis, it results in the coalitional control methods with the gain feedback formulations having similar performances to the DMPC strategies. This outcome was expected since the CC algorithms were designed using as the results obtained with the DMPC_{SS} method as a reference.

In terms of transient response performances (i.e., overshoot and settling time), for simplicity, only sub-system S_1 was analyzed, at the beginning of the experiment (first

100 samples). The results are also given in Table 2, and confirm that the DMPC strategies have comparable results with the coalitional control. The latter algorithm, based on gain feedback matrix control, provides an alternative control strategy to the optimization-based distributed model predictive control methods, and can be easily implemented on embedded systems due to its simpler formulation.

The time resource required for the local controller to compute the solution at each sampling time is: DMPC_{SS} 6.75×10^{-3} s, DMPC_{IO} 5.75×10^{-3} s, $\text{CC}_{\text{K dec}}$ 6.3609×10^{-8} s, $\text{CC}_{\text{K dist}}$ 6.4234×10^{-8} s and $\text{CC}_{\text{K switch}}$ 6.8371×10^{-8} s. These numerical values show that the CC strategy is more time-efficient that the DMPC methods.

Note that the numerical value of the J_{cost} for $\text{CC}_{\text{K switch}}$ given in Table 2 depends on the simulation test (i.e., the switching dynamics from Figure 8). Another simulation test, with other references, can give different results. The overall index value will be influenced by which topology is 'dominant' in the switching test depending on the corresponding simulation scenario.

To this end, an additional analysis was performed to evaluate the performance cost for multiple tracking scenarios. Hence, a set of 50 references was generated with the following characteristics:

- Length of the simulation time $M = 500$.
- The input weight $\beta = 0.001$.
- During the first 100 s, reference $r_1 = 10$ cm, at time 101 s, r_1 has a step change to a randomly generated value between 5 and 15 cm.
- During the first 200 s, reference $r_2 = 10$ cm, at time 201 s, r_2 has a step change to a randomly generated value between 5 and 15 cm.
- During the first 300 s, reference $r_3 = 10$ cm, at time 301 s, r_3 has a step change to a randomly generated value between 5 and 15 cm.
- During the first 400 s, reference $r_4 = 10$ cm, at time 401 s, r_4 has a step change to a randomly generated value between 5 and 15 cm.

For clarity, only the first 4 out of 50 references are depicted in Figure 9. For all 50 references, the J_{cost} was computed and is provided in Table 3. The results show that there are situations (see ref_1 and ref_{11}) in which the switching dynamics for $\text{CC}_{\text{K switch}}$ selects only one strategy for the entire simulation. In this case, for that reference, there are two equal values for J_{cost}. For each algorithm, the mean of J_{cost} values from Table 3 is 7.32 for DMPC_{SS}, 4.08 for DMPC_{IO}, 6.96 for $\text{CC}_{\text{K dec}}$, 8.39 for $\text{CC}_{\text{K dist}}$ and 7.02 for $\text{CC}_{\text{K switch}}$. These mean values reinforce the initial findings, i.e., that the coalitional control strategy has a similar performance to DMPC_{SS}.

Another analysis was performed to investigate the influence of the horizon T value within the switching algorithm. Using the same reference scenarios provided in Table 3, the algorithm $\text{CC}_{\text{K switch}}$ was tested for $T = 40$ and $T = 70$. For simplicity, only the mean of J_{cost} values are provided. Thus, algorithm $\text{CC}_{\text{K switch}}$ has an average J_{cost} of 6.98 and 6.97 for $T = 40$ and $T = 70$, respectively. This small difference when compared with the average cost of 7.02 corresponding to $T = 20$ implies that there is no gain in using larger horizon values when evaluating the topologies.

With respect to satisfying the imposed hard input and output constraints, only the lower limit of the input constraint was reached and respected, whereas the upper limits were never touched.

In the coalitional control strategy, when computing the optimal K for each topology (distributed and decentralized), a closed loop stability constraint (24) was imposed within the problem. After the computation of matrix K, for each topology, the stability constraint value denoted ρ was computed. Thus, the closed-loop stability of the coalition control strategy was assessed numerically for both communication topologies, obtaining two values within the unit circle, i.e., $\rho = 0.9506$ for $\text{CC}_{\text{K dec}}$ and $\rho = 0.9596$ for $\text{CC}_{\text{K dist}}$.

Remark 3. *Both DMPC and CC algorithms were tested using an academic simulation benchmark. The simulations were performed using MATLAB R2021a on Windows 10, 64-bit Operating System* *with a laptop Intel Core i5-9850H CPU @ 2.60 GHz460 and 8 GB RAM. Thus, the DMPC algorithms were not yet optimized to be executed on embedded devices and to be tested in a real-time setup, but this is a subject of future work. However, the simplicity of the coalitional control formulation, as well as its reduced computation burden, makes it suitable for controlling various coupled sub-systems, using embedded devices with limited storage and computation capabilities. This endeavor is subject to ongoing work.*

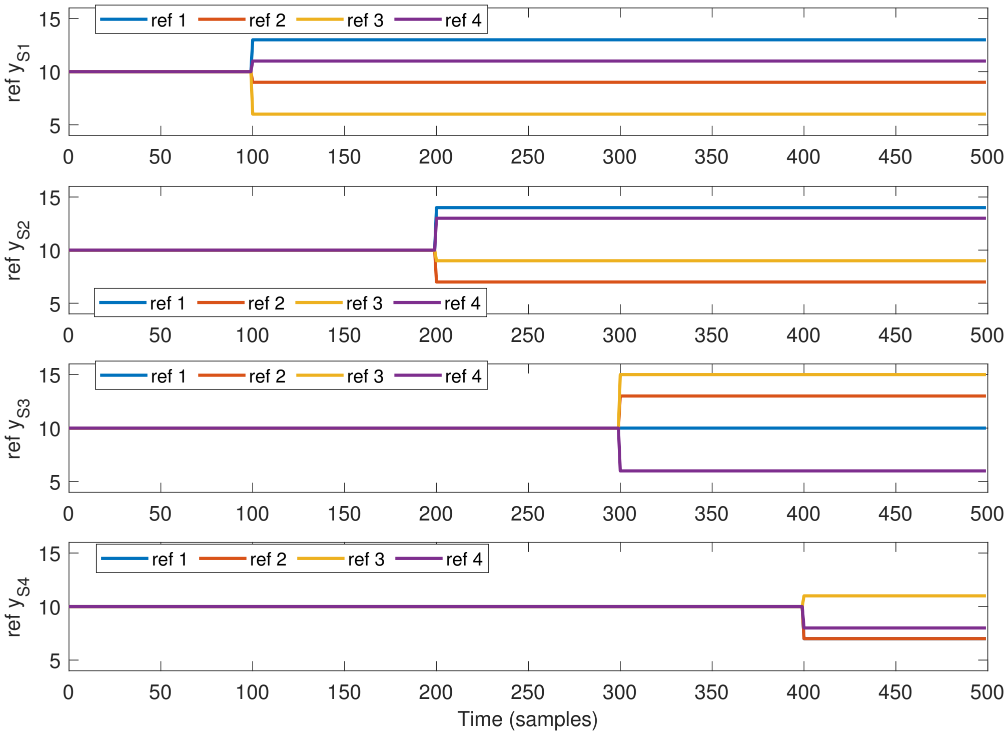

Figure 9. First 4 out of 50 reference sets scenarios used for the performance analysis provided in Table 3.

Table 3. Comparative analysis for DMPC$_{SS}$, DMPC$_{IO}$, CC$_{K\ dist}$, CC$_{K\ dec}$ and CC$_{K\ switch}$ algorithms based on performance index J_{cost} for 50 reference tracking scenarios.

Algorithm	**ref$_1$**	**ref$_2$**	**ref$_3$**	**ref$_4$**	**ref$_5$**	**ref$_6$**	**ref$_7$**	**ref$_8$**	**ref$_9$**	**ref$_{10}$**
DMPC$_{SS}$	7.06	6.9232	7.234	6.9941	7.4663	7.1101	6.2858	7.5058	7.1761	7.2923
DMPC$_{IO}$	3.8399	3.7616	3.9228	3.7897	4.0333	3.8603	3.4247	4.5454	3.9086	3.9745
CC$_{K\ dec}$	6.6852	6.5503	6.9014	6.6434	7.1358	6.7618	5.8971	7.1275	6.7765	6.957
CC$_{K\ dist}$	8.0269	7.9341	8.3214	7.9283	8.6276	8.1404	7.1244	8.4239	8.1521	8.3999
CC$_{K\ switch}$	6.6852	6.6496	6.9014	6.6434	7.2027	6.708	5.8971	7.1338	6.9815	7.0748
Algorithm	**ref$_{11}$**	**ref$_{12}$**	**ref$_{13}$**	**ref$_{14}$**	**ref$_{15}$**	**ref$_{16}$**	**ref$_{17}$**	**ref$_{18}$**	**ref$_{19}$**	**ref$_{20}$**
DMPC$_{SS}$	7.1924	7.8314	7.9128	6.8795	7.1698	7.5848	7.3844	6.8498	6.684	7.3929
DMPC$_{IO}$	3.906	4.6893	4.278	3.7476	3.8727	4.1029	4.0108	3.7335	3.6424	4.0175
CC$_{K\ dec}$	6.8137	7.4959	7.5955	6.529	6.8305	7.2667	7.0623	6.4549	6.3308	7.0272
CC$_{K\ dist}$	8.1562	10.0325	9.0328	7.8895	8.2037	8.7419	8.5967	7.7314	7.6851	8.5349
CC$_{K\ switch}$	6.8137	7.3403	7.5955	6.7604	6.8305	7.3402	7.0623	6.4549	6.3626	7.0342
Algorithm	**ref$_{21}$**	**ref$_{22}$**	**ref$_{23}$**	**ref$_{24}$**	**ref$_{25}$**	**ref$_{26}$**	**ref$_{27}$**	**ref$_{28}$**	**ref$_{29}$**	**ref$_{30}$**
DMPC$_{SS}$	6.6234	6.526	7.534	8.3664	8.869	6.8697	6.9339	7.4802	6.8035	6.4023
DMPC$_{IO}$	3.6011	3.5644	4.2458	4.5478	6.5396	3.7348	3.7476	4.0698	3.7181	3.4967
CC$_{K\ dec}$	6.2535	6.1489	7.1624	8.0334	8.5175	6.5079	6.588	7.1204	6.4382	6.016
CC$_{K\ dist}$	7.4819	7.4183	8.8481	9.691	10.2905	7.8125	7.9544	8.5508	7.7259	7.2463
CC$_{K\ switch}$	6.2535	6.1489	7.4142	8.434	8.5175	6.5079	6.588	7.1204	6.4382	6.016

Table 3. *Cont.*

Algorithm	ref_{31}	ref_{32}	ref_{33}	ref_{34}	ref_{35}	ref_{36}	ref_{37}	ref_{38}	ref_{39}	ref_{40}
DMPC_{SS}	10.0696	7.6568	6.1639	6.9149	7.0216	6.7982	8.2884	7.6686	8.1218	6.5389
DMPC_{IO}	7.3631	4.1428	3.3615	3.7589	3.8114	3.708	5.2155	4.1158	4.3801	3.5525
$\text{CC}_{\text{K dec}}$	9.7118	7.2916	5.7669	6.5551	6.6673	6.4463	7.9295	7.3523	7.8033	6.1602
$\text{CC}_{\text{K dist}}$	11.4071	8.7722	6.9522	7.8125	8.0195	7.795	9.2517	8.8459	9.3484	7.4062
$\text{CC}_{\text{K switch}}$	9.9748	7.2916	5.7669	6.6249	6.7133	6.4463	7.9884	7.6813	7.8033	6.2888

Algorithm	ref_{41}	ref_{42}	ref_{43}	ref_{44}	ref_{45}	ref_{46}	ref_{47}	ref_{48}	ref_{49}	ref_{50}
DMPC_{SS}	7.9808	7.4112	7.6753	7.3881	8.3035	6.7645	7.0579	7.3141	7.4849	7.1852
DMPC_{IO}	4.2934	4.347	4.1347	4.0217	4.5533	3.6815	3.8437	3.9524	4.035	3.8977
$\text{CC}_{\text{K dec}}$	7.6536	7.0194	7.3547	7.0098	7.9717	6.4051	6.6749	6.9475	7.1379	6.7994
$\text{CC}_{\text{K dist}}$	9.0457	8.8949	8.8047	8.3742	9.8268	7.7782	7.9759	8.2984	8.4842	8.1058
$\text{CC}_{\text{K switch}}$	7.494	7.7026	7.3214	7.0098	7.8147	6.5034	6.6749	7.1071	7.2662	6.7994

6. Conclusions

In this paper, a comparative performance analysis for two classes of control strategies was performed. When testing the DMPC and coalitional control strategies in a simulation setup, chosen as an eight-tank process with interconnected sub-systems, the results reveal that the coalitional methodology, based on feedback gain matrix control, is a suitable replacement for the optimization-based DMPC algorithms. Since the DMPC algorithm is based on online optimization and requires specialized optimization software, it is not trivial to use it on embedded systems, with limited capabilities. This was the motivation behind introducing the CC methodology, which has a simpler formulation based on a matrix gain feedback controller, and, once computed offline, can be easily employed on embedded systems. These findings are encouraging, and future work will test the proposed coalitional control strategy in a challenging, real-time experimental setup.

Author Contributions: Conceptualization, A.M. and C.-F.C.; methodology, A.M., O.P. and C.-F.C.; software, A.M. and O.P.; validation, A.M. and O.P.; writing—original draft preparation, A.M.; supervision, C.-F.C. All authors have read and agreed to the published version of the manuscript.

Funding: The work of A.M. and O.P. was supported by “Institutional development through increasing the innovation, development and research performance of TUIASI—COMPETE 2.0”, project funded by contract no. 27PFE /2021, financed by the Romanian government. The work of A.M. was also supported by “Gheorghe Asachi” Technical University of Iasi (TUIASI) through the Project “Performance and excellence in postdoctoral research 2022”. The work of C.F.C. was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS/CCCDI-UEFISCDI, project number PN-III-P1-1.1-TE-373 2019-1123, within PNCDI III.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MPC	Model Predictive Control
DMPC	Distributed Model Predictive Control
DMPC_{SS}	DMPC with state-space model
DMPC_{IO}	DMPC with input–output model
CC	Coalitional Control
$\text{CC}_{\text{K dec}}$	CC with decentralized communication topology
$\text{CC}_{\text{K dist}}$	CC with distributed communication topology
$\text{CC}_{\text{K switch}}$	CC with switching communication topology

References

1. Maestre, J.M.; Negenborn, R.R. *Distributed Model Predictive Control Made Easy*; Springer Science+Business Media: Dordrecht, The Netherlands, 2014.
2. Espin-Sarzosa, D.; Palma-Behnke, R.; Nuñez-Mata, O. Energy Management Systems for Microgrids: Main Existing Trends in Centralized Control Architectures. *Energies* **2020**, *13*, 547. [CrossRef]
3. Sun, X.; Yin, Y. Decentralized game-theoretical approaches for behaviorally-stable and efficient vehicle platooning. *Transp. Res. Part B -Methodol.* **2021**, *153*, 45–69. [CrossRef]
4. Camacho, E.F.; Bordons, C. *Model Predictive Control*; Springer: Berlin, Germany, 1999.
5. Lou, G.; Gu, W.; Xu, Y.; Cheng, M.; Liu, W. Distributed MPC-based secondary voltage control scheme for autonomous drop-control microgrids. *IEEE Trans. Sustain. Energy* **2017**, *8*, 792–804. [CrossRef]
6. Cortés, A.; Martínez, S. On distributed reactive power and storage control on microgrids. *Int. J. Robust Nonlinear Control* **2016**, *16*, 3150–3169. [CrossRef]
7. Liu, M.; Shi, Y.; Liu, X. Distributed MPC of aggregated heterogeneous thermostatically controlled loads in smart grids. *Int. Trans. Ind. Electron.* **2016**, *63*, 1120–1129. [CrossRef]
8. del Real, A.J.; Arce, A.; Bordons, C. Combined environmental and economic dispatch of smart grids using distributed model predictive control. *Electr. Power Energy Syst.* **2014**, *54*, 65–76. [CrossRef]
9. Pham, V.; Ahn, H. Distributed Stochastic MPC Traffic Signal Control for Urban Networks. *IEEE Trans. Intell. Transp. Syst.* 2023, *early access*. [CrossRef]
10. Liu, P.; Ozguner, U.; Zhang, Y. Distributed MPC for cooperative highway driving and energy-economy validation via microscopic simulations. *Transp. Res. Part C Emerg. Technol.* **2017**, *77*, 80–95. [CrossRef]
11. Yan, X.; Cai, B.; Ning, B.; ShangGuan, W. Online distributed cooperative model predictive control of energy-saving trajectory planning for multiple high speed train movements. *Transp. Res. Part C Emerg. Technol.* **2016**, *69*, 60–78. [CrossRef]
12. Kersbergen, B.; van den Boom, T.; de Schutter, B. Distributed model predictive control for railway traffic management. *Transp. Res. Part C Emerg. Technol.* **2016**, *68*, 462–489. [CrossRef]
13. Ferarra, A.; Nai Oleari, A.; Sacone, S.; Siri, S. Freeway as system of systems: A distributed model predictive control scheme. *IEEE Syst. J.* **2015**, *9*, 462–489. [CrossRef]
14. Ye, B.L.; Wu, W.; Mao, W. Distributed model predictive control method for optimal coordination of signal splits in urban traffic networks. *Asian J. Control* **2015**, *17*, 775–790. [CrossRef]
15. Li, H.; Zhang, T.; Zheng, S.; Sun, C. Distributed MPC for Multi-Vehicle Cooperative Control Considering the Surrounding Vehicle Personality. *IEEE Trans. Intell. Transp. Syst.* 2023, *early access*. [CrossRef]
16. Pauca, O.; Maxim, A.; Caruntu, C.F. DMPC-based Data-packet Dropout Compensation in Vehicle Platooning Applications using V2V Communications. In Proceedings of the 2021 European Control Conference, Rotterdam, The Netherlands, 29 June–2 July 2021; pp. 2639–2644.
17. Maxim, A.; Lazar, C.; Caruntu, C.F. Distributed Model Predictive Control Algorithm with Communication Delays for a Cooperative Adaptive Cruise Control Vehicle Platoon. In Proceedings of the 28th Mediterranean Conference on Control and Automation, Saint-Raphaël, France, 15–18 September 2020; pp. 909–914.
18. Caruntu, C.F.; Braescu, C.; Maxim, A.; Rafaila, R.C.; Tiganasu, A. Distributed model predictive control for vehicle platooning: A brief survey. In Proceedings of the 20th International Conference on System Theory, Control and Computing, Sinaia, Romania, 13–15 October 2016; pp. 644–650.
19. Liu, X.; Zhang, Y.; Lee, K.Y. Coordinated distributed MPC for load frequency control of power system with wind farms. *IEEE Trans. Ind. Electron.* **2017**, *64*, 5140–5150. [CrossRef]
20. Spudić, V.; Conte, C.; Baotić, M.; Morari, M. Cooperative distributed model predictive control for wind farms. *Optim. Control Appl. Methods* **2015**, *36*, 333–352. [CrossRef]
21. Zhao, H.; Wu, Q.; Guo, Q.; Sun, H.; Xue, Y. Distributed model predictive control of a wind farm for optimal active power control: Part II: Implementation with clustering based piece-wise affine wind turbine model. *IEEE Trans. Sustain. Energy* **2015**, *6*, 840–849. [CrossRef]
22. Zhang, A.; Yin, X.; Liu, S.; Zeng, J.; Liu, J. Distributed economic model predictive control of wastewater treatment plants. *Chem. Eng. Res. Des.* **2019**, *141*, 144–155. [CrossRef]
23. Foscoliano, C.; Del Vigo, S.; Mulas, M.; Tronci, S. Improving the wastewater treatment plant performance through model predictive control strategies. In Proceedings of the 26th European Symposium on Computer Aided Process Engineering, Portoroz, Slovenia, 12–15 June 2016; pp. 1863–1868.
24. Albalawi, F.; Durand, H.; Christofides, P.D. Distributed Economic Model Predictive Control with Safeness-Index Based Constraints of a Nonlinear Chemical Process. In Proceedings of the 2018 Annual American Control Conference, Milwaukee, WI, USA, 27–29 June 2018; pp. 2078–2083.
25. Zhang, S.; Zhao, D.; Spurgeon, S.K.; Yan, X. Distributed Model Predictive Control for the Atmospheric and Vacuum Distillation Towers in a Petroleum Refining Process. In Proceedings of the 11th UKACC International Conference on Control, Belfast, North Ireland, 31 August–2 September 2016.
26. Ocampo-Martinez, C.; Puig, V.; Cembrano, G.; Quevedo, J. Application of predictive control strategies to the management of complex networks in the urban water cycle. *IEEE Control Syst.* **2013**, *33*, 15–41.

27. Zhao, Z.; Guo, J.; Luo, X.; Lai, C.S.; Yang, P.; Lai, L.; Li, P.; Guerrero, J.; Shahidehpour, M. Distributed Robust Model Predictive Control-Based Energy Management Strategy for Islanded Multi-Microgrids Considering Uncertainty. *IEEE Trans. Smart Grid* **2022**, *13*, 2107–2120. [CrossRef]
28. Shi, Y.; Tuan, H.D.; Savkin, A.V.; Lin, C.T.; Zhu, J.G.; Poor, H.V. Distributed model predictive control for joint coordination of demand response and optimal power flow with renewables in smart grid. *Appl. Energy* **2021**, *209*, 116701. [CrossRef]
29. Liu, Y.; Liu, R.; Wei, C.; Xun, J.; Tang, T. Distributed Model Predictive Control Strategy for Constrained High-Speed Virtually Coupled Train Set. *IEEE Trans. Veh. Technol.* **2022**, *71*, 171–183. [CrossRef]
30. Zhou, Y.; Wang, M.; Ahn, S. Distributed model predictive control approach for cooperative car-following with guaranteed local and string stability. *Transp. Res. Part B* **2019**, *128*, 69–86. [CrossRef]
31. Kong, X.; Ma, L.; Wang, C.; Guo, S.; Abdelbaky, M.A.; Liu, X.; Lee, K.Y. Large-scale wind farm control using distributed economic model predictive scheme. *Renew. Energy* **2022**, *181*, 581–591. [CrossRef]
32. Teng, Y.; Bai, J.; Wu, F.; Zou, H. Explicit distributed model predictive control design for chemical processes under constraints and uncertainty. *Can. J. Chem. Eng.* 2023, *early access*. [CrossRef]
33. Zhang, L.; Wang, J.; Liu, Z.; Li, K. Distributed MPC for tracking based on reference trajectories. In Proceedings of the 33rd Chinese Control Conference, Nanjing, China, 28–30 July, 2014; pp. 7778–7783.
34. Arauz, T.; Chanfreut, P.; Maestre, J.M. Cyber-security in networked and distributed model predictive control. *Annu. Rev. Control* **2022**, *53*, 338–355. [CrossRef]
35. Christofides, P.D.; Scattolini, R.; Muñoz de la Peña, D.; Liu, J. Distributed model predictive control: A tutorial review and future research directions. *Comput. Chem. Eng.* **2013**, *51*, 21–41. [CrossRef]
36. Scattolini, R. Architectures for distributed and hierarchical Model Predictive Control—A review. *J. Process Control* **2009**, *19*, 723–731. [CrossRef]
37. Fele, F.; Maestre, J.M.; Camacho, E.F. Coalitional control: Cooperative game theory and control. *IEEE Control Syst.* **2017**, *37*, 53–69.
38. Chanfreut, P.; Maestre, J.M.; Camacho, E.F. A survey on clustering methods for distributed and networked control systems. *Annu. Rev. Control* **2021**, *52*, 75–90. [CrossRef]
39. Maxim, A.; Caruntu, C.F. A Coalitional Distributed Model Predictive Control Perspective for a Cyber-Physical Multi-Agent Application. *Sensors* **2021**, *21*, 4041. [CrossRef]
40. Maxim, A.; Caruntu, C.F.; Lazar, C.; De Keyser, R.; Ionescu, C.M. Comparative Analysis of Distributed Model Predictive Control Strategies. In Proceedings of the 23rd International Conference on System Theory, Control and Computing, Sinaia, Romania, 9–11 October 2019; pp. 468–473.
41. Maxim, A.; Pauca, O.; Maestre, J.M.; Caruntu, C.F. Assessment of computation methods for coalitional feedback controllers. In Proceedings of the 2022 European Control Conference, London, UK, 12–15 July 2022; pp. 1448–1453.
42. Maxim, A.; Ionescu, C.M.; Caruntu, C.F.; Lazar, C.; De Keyser, R. Reference Tracking using a Non-Cooperative Distributed Model Predictive Control Algorithm. In Proceedings of the 11th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems, Trondheim, Norway, 6–8 June 2016; pp. 1079–1084.
43. Maxim, A.; Copot, D.; De Keyser, R.; Ionescu, C.M. An industrially relevant formulation of a distributed model predictive control algorithm based on minimal process information. *J. Process Control* **2018**, *68*, 240–253. [CrossRef]
44. De Keyser, R.; Ionescu, C.M. The disturbance model in model based predictive control. In Proceedings of the 2003 IEEE Conference on Control Applications, Istanbul, Turkey, 25–25 June 2003; pp. 446–451.
45. De Keyser, R. Model Based Predictive Control for Linear Systems. In *UNESCO Encyclopaedia of Life Support Systems, Control Systems, Robotics and Automation—Vol. XI, Article Contribution 6.43.16.1*; Eolss Publishers Co. Ltd.: Oxford, UK, 2003. Available online: http://www.eolss.net/sample-chapters/c18/e6-43-16-01.pdf (accessed on 1 January 2023).
46. Maxim, A.; Ionescu, C.M.; Copot, C.; De Keyser, R.; Lazar, C. Multivariable model-based control strategies for level control in a quadruple tank process. In Proceedings of the 17th International Conference on System Theory, Sinaia, Romania, 11–13 October 2013; pp. 343–348.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.