

Article



A Coalitional Distributed Model Predictive Control Perspective for a Cyber-Physical Multi-Agent Application

Anca Maxim 🗈 and Constantin-Florin Caruntu *🕩

Department of Automatic Control and Applied Informatics, "Gheorghe Asachi" Technical University of Iasi, 700050 Iasi, Romania; anca.maxim@ac.tuiasi.ro

* Correspondence: caruntuc@ac.tuiasi.ro

Abstract: Following the current technological development and informational advancement, more and more physical systems have become interconnected and linked via communication networks. The objective of this work is the development of a Coalitional Distributed Model Predictive Control (C-DMPC) strategy suitable for controlling cyber-physical, multi-agent systems. The motivation behind this endeavour is to design a novel algorithm with a flexible control architecture by combining the advantages of classical DMPC with Coalitional MPC. The simulation results were achieved using a test scenario composed of four dynamically coupled sub-systems, connected through an unidirectional communication topology. The obtained results illustrate that, when the feasibility of the local optimization problem is lost, forming a coalition between neighbouring agents solves this shortcoming and maintains the functionality of the entire system. These findings successfully prove the efficiency and performance of the proposed coalitional DMPC method.

Keywords: coalitional model predictive control; distributed model-predictive control; multi-agent systems; closed-loop stability

1. Introduction

Presently, manifold systems are modular, interconnected and have a cyber-physical setup, meaning they can be viewed as coupled physical sub-systems, which are connected via communication networks [1–5]. For such processes, Distributed Model Predictive Control (DMPC) is a reliable control solution that uses local controllers that compute the control action using both (i) the local information derived from specific sensors and (ii) coupling data received/transmitted using the communication network [6].

As recent studies attest, the DMPC strategy was successfully applied on multi-agent systems in varying applications, such as formation control of autonomous surface and aerial vehicles [7], leader–follower platoons [8,9], traffic signal control [10], temperature regulation systems [11], battery energy storage systems [12] and microgrids [13,14]. In [15], a DMPC strategy for multi-agent systems based on error upper bounds is provided. This criterion is used in a min–max optimization of the cost function to minimize the communication between neighbouring agents. An event-triggered synchronous DMPC for multi-agent systems, coupled through a cost function. An event-triggered mechanism designed using the forward difference of the cost function is deployed to activate the local optimization problem at each sampling time; otherwise the agents use the solutions computed in the previous sampling period. In [17], a DMPC to reach consensus for time-varying, multi-agent systems is proposed. The consensus DMPC algorithm is designed for heterogeneous, time-varying decoupled sub-systems, connected uni-directionally with a coupled cost function.

In all the research mentioned above, regardless of the application or the methodology details, one key feature is noticeable, namely that the architecture of both sub-systems and agents (i.e., local controllers) is fixed. The latter is predefined in the initialization phase of



Citation: Maxim, A.; Caruntu, C.F. A Coalitional Distributed Model Predictive Control Perspective for a Cyber-Physical Multi-Agent Application. *Sensors* **2021**, *21*, 4041. https://doi.org/10.3390/s21124041

Academic Editor: Reinaldo Martinez Palhares

Received: 6 May 2021 Accepted: 9 June 2021 Published: 11 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). the control design, based on the sensors placements and interconnection between the local sub-systems. Therefore, the configuration of the DMPC neighbourhoods (i.e., groups of local sub-systems that are interconnected either dynamically or through cost functions or constraints) is established and predefined [18].

To overcome this shortcoming, a new approach emerged from cooperative game theory framework named Coalitional Control was introduced, with the following characteristics [19,20]: (i) the topology of the communication links between agents is flexible (i.e., links can be enabled or disabled when necessary), (ii) the control strategy encourages the agents to group in cooperative clusters called coalitions (to reduce the communication burden), and (iii) the communication links between agents that ensure their cooperation are weighted and introduce supplementary costs in the cost functions when activated.

Using this foundation, a Coalitional Model Predictive Control (C-MPC) strategy was developed and applied on different applications such as cellular networks in [21,22] or an eight-coupled tank process [23]. Thus, the agents charged with controlling the local sub-systems can form coalitions depending on the activation of the communication links between them. Several topologies can be derived, starting from the default one, which is the decentralized MPC strategy (i.e., with no communication between agents), to the most complex one. described as centralized MPC (i.e., in which all the communication links are active). In between, there are the coalitions between several agents in a neighbourhood, while the remaining ones work independently. In [24], details regarding the feasible regions for tube-based MPC controllers are discussed. The coalitions are associated with different partitions of a large-scale system (i.e., several sub-systems can be joined in a single entity), and their feasible region is analysed.

The main contribution of this work is the development of a novel perspective of the DMPC algorithm, called Coalitional DMPC (C-DMPC), which combines both the advantages and features of classical DMPC strategy with the characteristics of Coalitional MPC. Hence, the envisioned solution is to dynamically reshape the controller network by merging some of the agents within a neighbourhood into coalitions when needed. The advantages of this approach are shown when, due to various reasons (e.g., reference changes, unknown disturbances, etc.), one or more local optimal solutions become infeasible. In this case, to maintain the feasibility and functionality of the interconnected cyber-physical multi-agent system, the agents will decide to form a coalition. This means that inside a coalition, the sub-systems become a single entity and the controllers aggregate and solve a cooperative optimization problem (i.e., a global cost function is minimized) [25].

The main difference of our approach with respect to the cited coalitional literature is that the default topology is a non-cooperative DMPC (i.e., each agent minimizes a local cost function, using received information from its neighbours) [26]. This means that when a coalition occurs, the remaining agents outside the coalitions are not independent but retain their previous status and solve a non-cooperative optimization problem. Thus, depending on the topology, it is possible that the coalition must exchange information with its neighbours (if not all the agents inside a neighbourhood are merged into the coalition). Since all the agents start as non-cooperative players, they use the communication network to share relevant data, according to their coupling within neighbourhoods, and all pertinent communication links are activated and not weighted.

Another key difference is our proposed merging procedure, which selects the agents that will form a coalition. This is done at each agent level, without using a hierarchical supervisory layer. Moreover, when the local optimization problem becomes unfeasible (due to the coupling information), the coalition is activated. Furthermore, two simplified versions of this method with different agent merging procedures were published in [27,28]. In [27], each agent considers that the coupling information received from the neighboring agents is an uncertainty in the local nominal model. When a predefined threshold value for the local uncertainty level is crossed, than a coalition between the agents is formed. The further development of this idea is given in [28], in which the coalition between the agents

is formed, when the local optimization problems become infeasible due to the received uncertainty level.

With respect to our previous papers, the method proposed in the current work has significant improvements, such as the following: (i) the network topology is tailored for in-chain coupled sub-systems, with unidirectional communication links; (ii) a more realistic academic example is used for simulation tests, with four heterogeneous sub-systems dynamically coupled through the inputs; (iii) each sub-system model is augmented with an additional state defined as the integral of control error to ensure a non-zero reference tracking; and (iv) multiple coalitions between agents can be simultaneously active at each sampling time.

The remaining of this paper is structured as follows: Section 2 presents the problem formulation and details the proposed method, whereas the simulation configuration, results, and discussions are provided in Section 3. The conclusions of this work and future work plans are addressed in Section 4.

2. Problem Formulation

A cyber-physical multi-agent system (CP-MAS), as depicted in Figure 1, is composed of *N* interconnected cyber-physical sub-systems (CPsS). Each CPsS is defined by the pair $(S_i, A_i), \forall i \in \mathcal{N}$, where \mathcal{N} denotes the set $\{1, \ldots, N\} \subseteq \mathbb{N}$, with $N \in \mathbb{N}$ the number of subsystems and \mathbb{N} the set of natural numbers. The physical part of the CPsS is denoted with S_i , whereas the cyber part of the CPsS is denoted with A_i and represents the corresponding local controller or agent. All the interconnected sub-systems S_i form the physical layer (depicted with grey colour), while the cyber layer (depicted with blue colour) is composed of all the agents and the communication networks.





Let each sub-system S_i be defined by the following model:

$$\begin{aligned} x_{p}^{i}(k+1) &= A_{p}^{i,i}x_{p}^{i}(k) + B_{p}^{i,i}u_{p}^{i}(k) + w_{p}^{i}(k) \\ w_{p}^{i}(k) &= B_{p}^{i,i-1}u_{p}^{i-1}(k) \\ y_{n}^{i}(k) &= C_{n}^{i}x_{p}^{i}(k), \, \forall i \in \mathcal{N} \end{aligned}$$
(1)

with the notations for the state $x_p^i \in \mathbb{R}^{n_i}$, input $u_p^i \in \mathbb{R}^{m_i}$, input uncertainty $w_p^i \in \mathbb{R}^{p_i}$ and output $y_p^i \in \mathbb{R}^{q_i}$. $A_p^{i,i}$, $B_p^{i,i-1}$ and C_p^i are matrices with adequate dimensions. n_i , m_i , p_i and q_i are the number of states, inputs, input uncertainties and outputs, respectively. \mathbb{R} denotes the set of real numbers. Note that, $u_p^{i-1} \in \mathbb{R}^{m_i}$ denotes the input signal received from the predecessor sub-system with index i - 1.

Note that (1) defines a model in which the input-coupling information w_p^i is considered an uncertainty in the nominal model. Moreover, all the sub-systems S_i , $\forall i \in \mathcal{N}$, are in a

chain architecture, and for sub-system indexed *i*, the information is received through an unidirectional link from its predecessor and neighbour, defined as the sub-system with index i - 1.

To ensure that the reference tracking control problem has a zero error in stationary regime, the state vector x_p^i , $\forall i \in \mathcal{N}$, from (1) is extended with an additional state \overline{x}_p^i defined as integral of the control error, using the following definition [29,30]:

$$\overline{x}_{p}^{i}(k+1) = \overline{x}_{p}^{i}(k) - C_{p}^{i}x_{p}^{i}(k) + r_{i}(k),$$
(2)

obtaining

$$\frac{x_{p}^{i}(k+1)}{\overline{x}_{p}^{i}(k+1)} = \underbrace{\left[\begin{array}{c}A_{p}^{i,i} & O\\-C_{p}^{i} & I\end{array}\right]}_{A_{i,i}} \underbrace{\left[\begin{array}{c}x_{p}^{i}(k)\\\overline{x}_{p}^{i}(k)\end{array}\right]}_{x_{i}(k)} + \underbrace{\left[\begin{array}{c}B_{p}^{i,i}\\O\end{array}\right]}_{B_{i,i}}u_{i}(k) \\ + \underbrace{\left[\begin{array}{c}B_{p}^{i,i-1}\\O\end{array}\right]}_{B_{i,i-1}}u_{i-1}(k) + \underbrace{\left[\begin{array}{c}O\\I\end{array}\right]}_{R_{sp_{i}}}r_{i}(k) \\ y_{i}(k) = \underbrace{\left[\begin{array}{c}C_{p}^{i} & O\end{array}\right]}_{C_{i}}\underbrace{\left[\begin{array}{c}x_{p}^{i}(k)\\\overline{x}_{p}^{i}(k)\end{array}\right]}_{x_{i}(k)} \\ \end{array} \right]$$
(3)

where $r_i(k)$ is the imposed reference value at time k. $x_i(k)$, $u_i(k)$ and $y_i(k)$ are the extended state, input and output vectors, respectively. Note that the input uncertainty $w_i(k)$ is defined based on the input vector received from the predecessor $u_{i-1}(k)$. I and O are the identity and zero matrix, respectively, each with appropriate dimensions.

Hereafter, each sub-system S_i , $\forall i \in \mathcal{N}$, will be represented by the compact extended model:

$$\begin{aligned} x_i(k+1) &= A_{i,i}x_i(k) + B_{i,i}u_i(k) + w_i(k) + R_{{\rm sp}_i}r_i(k) \\ w_i(k) &= B_{i,i-1}u_{i-1}(k) \\ y_i(k) &= C_ix_i(k) \end{aligned}$$
 (4)

where $A_{i,i}$, $B_{i,i-1}$, C_i and R_{sp_i} are matrices with adequate dimensions.

Consider linear inequality constraints for the outputs, inputs and uncertainties defined with:

$$y_i \in \mathcal{Y}_i, \ u_i \in \mathcal{U}_i, \ w_i \in \mathcal{W}_i, \ \forall i \in \mathcal{N}$$
 (5)

where \mathcal{Y}_i , \mathcal{U}_i and \mathcal{W}_i are sets defined by linear inequalities.

At every sampling time, each agent A_i , $\forall i \in \mathcal{N}$, solves a min–max optimization problem, which aims to obtain the minimum optimal input with respect to the maximum level of uncertainty received from its neighbour.

$$J_{i}(x_{i}^{0}) = \min_{\substack{u_{i}^{0},...,u_{i}^{N_{p}-1} \\ \hat{u}_{i}^{\max}}} \max_{i}^{N_{p}-1} w_{i}^{0},...,w_{i}^{N_{p}-1}} \left\| R_{u_{i}}u_{i}^{l} \right\|_{1} + \left\| R_{u_{i}}u_{i}^{l} \right\|_{1} + \left\| r_{i}^{N_{p}} - y_{i}^{N_{p}} \right\|_{1}}$$
s. t. (4)
$$y_{i}^{l} \in \mathcal{Y}_{i}, \ l = 1,...,N_{p} - 1$$

$$y_{i}^{N_{p}} \in \Omega_{i}$$

$$u_{i}^{l} \leq \hat{u}_{i}^{\max} \leq u_{i}^{\max}, \ l = 0,...,N_{p} - 1$$

$$w_{i}^{l} \leq w_{i}^{\max}$$
(6)

where $y_i^l = y_i(k + l|k)$ denotes the output predictions for sub-system S_i at time k + l, computed at time step k; this is calculated recursively starting from the initial state $x_i^0 = x_i(k)$ measured at time k, using the model (4); the input sequence $u_i^l = \left\{u_i^0, \ldots, u_i^{N_p-1}\right\}$ computed over the prediction horizon N_p ; and the uncertainty sequence $\mathcal{W}_i = B_{i,i-1}\mathcal{U}_{i-1}$ received from the neighbour (where \mathcal{W}_i is the uncertainty polytope and $\mathcal{U}_{i-1} = \left\{u \in \mathbb{R}^{m_i} : Au \leq b$ is a H-polytope); r_i^l is the value at time k + l for the output reference trajectory; $r_i^{N_p}$ and $y_i^{N_p}$ are the values for the reference and the output trajectories, at the end of the prediction horizon $k + N_p$, respectively; u_i^{\max} , w_i^{\max} are the maximum limits for the input and the uncertainty sequences, respectively; $\|.\|_1$ denotes the 1-norm; $R_{u_i} \in \mathbb{R}^{m_i \times m_i}$ and $R_{w_i} \in \mathbb{R}^{m_i \times m_i}$ are the weight matrices for the input and self-imposed input limit \hat{u}_i^{\max} . The latter is an additional optimization parameter introduced in the local cost function, and its value is communicated at each sampling period to the neighbour. This will guarantee that the uncertainty level received from the neighbour is smaller than this value, without actually transmitting the entire input sequence. The set Ω_i is a robust positive invariant set used to ensure the closed-loop stability of the algorithm by means of the terminal invariant set.

Remark 1. The uncertainty in each sub-system model refers to the coupling information that must be received from the neighbouring sub-system. Please note that the local optimization problem (6) minimizes the control input, for the worst-case scenario related to uncertainty level received from the predecessor agent. This means that, although unknown, this uncertainty must be bounded to a known value, which is shared between consecutive sub-system. Moreover, this ensures that each local sub-system is prepared for the disturbance signal, which is received via the coupling links.

Next, some details regarding the computation of the invariant set Ω_i , followed by the proposed coalitional DMPC method are given.

2.1. Robust Positive Invariant Set Computation

In this sub-section, the details regarding the computation of the robust positive invariant set Ω_i , $\forall i \in \mathcal{N}$, which acts as a constraint region for the terminal state $y_i^{N_p} \in \Omega_i$ are presented. To this end, the procedure firstly introduced in [28] is briefly summarized below, tailored for the extended sub-system model.

For each sub-system S_i , $\forall i \in \mathcal{N}$, with the model defined in (4) and subject to constraints (5), only the nominal model (i.e., w_i and r_i are zero) is considered. Let us compute a local linear feedback $u_i = K_i x_i$, which ensures that the closed loop eigenvalues are in the unit circle. One suggestion to compute the state feedback matrix K_i is to apply classical state-space feedback control designed for the nominal model using Ackermann's formula (i.e., solving a pole allocation problem) [31], or to calculate it through the minimization of a linear-quadratic cost function, by solving a discrete-time Riccatti Equation [32]. The set Ω_i is robust positive invariant for the nominal model from (4), if the following assumption holds [28,33]:

$$x_{i} \in \Omega_{i} \to (A_{i,i} + B_{i,i}K_{i})x_{i} + w_{i} \in \Omega_{i},$$

$$K_{i}x_{i} \in \mathcal{U}_{i}, C_{i}\Omega_{i} \subseteq \mathcal{Y}_{i}, \forall w_{i} \in \mathcal{W}_{i}$$
(7)

It is worth mentioning the following observations regarding the use of the invariant set in the C-DMPC context:

- the default working framework is non-cooperative DMPC, which implies that each agent A_i, ∀i ∈ N, from the multi-agent application communicates with its neighbour, in order to compute the local solution;
- each sub-system model S_i , $\forall i \in N$, is subject to input uncertainties received from the sub-system to whom it is connected (in our case its predecessor);
- to provide a simplified algorithm with minimal communication load in the network, only the self-imposed upper bound for the local input trajectory is broadcast in the network (i.e., the optimization variable \hat{u}^{max} introduced in (6));
- a table with different predefined robust positive invariant sets Ω_i is computed using the constraints limits from (5), in which each element is a particular combination of the variable bounds (see Algorithm 1);
- at each sampling period, after the uncertainty upper bound is received from the neighbour, each agent A_i uses this information to compute the uncertainty polytope. Next, from the predefined terminal sets table, a set Ω_i is searched for, which includes the received uncertainty polytope (i.e., which will ensure a local feasible solution in the terminal state framework).

Further on, the pseudo-code algorithm used to compute the invariant set table is provided (where for simplicity the sub-system indices are omitted):

Thus, each agent A_i , $\forall i \in \mathcal{N}$, uses Algorithm 1 in the initialization phase of the proposed method to compute a table of invariant sets Ω_i , for different input and uncertainty parametrizations (i.e., distinct combinations for the two parameters α and β). Note that the first set Ω_i from the table corresponds to the largest value for the input constraint, denoted u^{max} , whereas the uncertainty has the smallest value. The latter is gradually increased with a step size denoted $step_{\beta}$, until it reaches its maximum admissible value w^{max} . In doing so, the size of the invariant set slowly reduces, as the input constraint limit value decreases with a step size denoted $step_{\alpha}$ and the uncertainty level rises.

In practice, a good start for u^{max} and w^{max} bounds are the values for the imposed constraints (5). The values of the step size $step_{\alpha}$, $step_{\beta}$ should be selected such that the table size remains reasonable, with various invariant sets. Moreover, the limits in the state constraints are considered fixed, according to the sub-systems dynamics and used to compute every set Ω_i from the table.

Algorithm 1

For $\alpha = u^{max} : -step_{\alpha} : 0.1$

For $\beta = 0.1 : step_{\beta} : w^{max}$

1. Compute the inequality constraints:

 $A_u u \leq \alpha b_u; \quad A_w w \leq \beta b_w; \quad A_x x \leq b_x$

2. Compute the robust positive-invariant set:

 $\Omega(A, B, K, A_u, b_u, A_w, b_w, A_x, b_x)$

3. Save the information α , β , Ω

end end

2.2. Coalitional Distributed Model Predictive Control (C-Dmpc) Methodology

As previously mentioned, what differentiates our proposed coalitional algorithm from the existing works is the flexible framework set for the cyber-physical multi-agent system with a chain architecture. Hence, at each step time, the agents architecture starts as non-cooperative DMPC and will switch to coalitional DMPC (C-DMPC)—when the local feasibility of the interconnected agents is lost. In the C-DMPC framework, the coalition procedure is initialized without a hierarchical level by the local agents with infeasible problems, because due to the coupling links between sub-systems, if not solved, this problem will propagate among neighbouring sub-systems. Using the communication links, these agents share their optimization status with their neighbour, and after that, one of them is randomly selected to start a coalition. Once the coalition procedure is activated, the agents framework changes.

To simplify the design and computational costs, the size of the coalition is increased gradually, if needed. That is, if a coalition of two agents, coupled with the remaining agents from the network, still does not provide feasible solutions for all involved actors, then more work needs to be done. The idea is to first activate all coalitions of two agents, if needed, then the coalitions of three agents, and so on, until in the end, in the extreme case, all the agents are involved in a single coalition. Note that this last case is equivalent to solving a centralized problem for the multi-agent system and will be used in the last resort, if nothing else solved the infeasibility problems that started the coalitional procedure. The reason for this is related with the coalition dynamics (i.e., when two or more agents form a coalition, their respective sub-system models are aggregated and become a single entity). Thus, the number of the optimization variables in a coalition increases with its size, and the local non-cooperative optimization problem becomes a cooperative one inside the coalition. The extreme case of a 'grand' coalition between all agents will aggregate all the sub-systems in a single entity (from the control point of view).

2.2.1. Coalition Dynamics

As described before, our C-DMPC algorithm is tailored specifically for cyber-physical multi-agent systems, linked in a unidirectional communication topology. Thus, the coupling information, which is treated as an uncertainty in the local nominal model of each sub-system S_i , is received from its predecessor sub-system S_{i-1} . To minimize the communication burden between consecutive agents, only the self imposed optimization variable \hat{u}^{max} introduced in (6) is broadcast. This value is firstly used to search for an invariant set inside the predefined table, and secondly acts as the uncertainty limit constraint in the local optimization problem. Using this information, the local optimization problem is then solved, and if the solution is infeasible, then the coalition procedure must be started.

Inside a coalition between different consecutive agents, the aim is to solve a cooperative optimization problem; thus the uncertainty variable becomes fully known. Each agent A_i , $\forall i \in \mathcal{N}$, can form a coalition only with its predecessor, i.e., agent A_{i-1} , due to the particular dynamical coupling between their corresponding sub-systems (i.e., linked in a chain). When this occurs, the agents involved will form a compact set denoted generically C. To simplify the notations, the coalition is described without sub-script indices with the following model:

$$x_{\mathcal{C}}(k+1) = A_{\mathcal{C}}x_{\mathcal{C}}(k) + B_{\mathcal{C}}u_{\mathcal{C}}(k) + w_{\mathcal{C}}(k)$$

$$w_{\mathcal{C}}(k) = \sum_{j \in \mathcal{N}_{\mathcal{C}}} B_{\mathcal{C}}^{j}u_{j}(k)$$

$$y_{\mathcal{C}}(k) = C_{\mathcal{C}}x_{\mathcal{C}}(k)$$
(8)

where x_C is the state vector of the coalition, u_C is the coalition's input vector, w_C is the uncertainty vector of the coalition and y_C is output vector for the coalition. All these vectors are composed by aggregating the local vectors corresponding to each sub-system involved

in the coalition (e.g., $x_{\mathcal{C}} = [x_i]_{i \in \mathcal{C}}$). Moreover, the matrices $A_{\mathcal{C}}$, $B_{\mathcal{C}}$, $B_{\mathcal{C}}^j$ and $C_{\mathcal{C}}$ are computed according to the aggregation.

The set $\mathcal{N}_{\mathcal{C}}$ denotes the coalition's \mathcal{C} neighbour, defined as the predecessor sub-system for the sub-systems inside the coalition (e.g., if Agent 2 and 3 form a coalition, then $\mathcal{N}_{\mathcal{C}} = \{1\}$, because sub-system 2 is coupled to sub-system 1; thus the coalition in which Agent 2 is involved must receive relevant information from Agent 1, which is outside the coalition and solves a non-cooperative DMPC problem). Moreover, following this reasoning, a coalition involving Agent 1 does not have neighbours (i.e., $\mathcal{N}_{\mathcal{C}} = \emptyset$, because Agent 1 does not have predecessors).

2.2.2. Coalition Problem Definition

^

In this section, some details regarding the construction of the constraints sets imposed for the coalition and the optimization problem solved by the coalition are presented.

Hence, the constraint sets for the coalition C are computed as the union of the constraints sets (5) corresponding to each agent A_i , $i \in C$:

$$y_{\mathcal{C}} \in \mathcal{Y}_{\mathcal{C}} = \prod_{i \in \mathcal{C}} \mathcal{Y}_i, \ u_{\mathcal{C}} \in \mathcal{U}_{\mathcal{C}} = \prod_{i \in \mathcal{C}} \mathcal{U}_i, \ w_{\mathcal{C}} \in \mathcal{W}_{\mathcal{C}} = \prod_{i \in \mathcal{C}} \mathcal{W}_i,$$
(9)

and the min-max optimization problem solved by the coalition is:

$$J_{\mathcal{C}}(x_{\mathcal{C}}^{0}) = \min_{\substack{u_{\mathcal{C}}^{0},...,u_{\mathcal{C}}^{N-1}w_{\mathcal{C}}^{0},...,w_{\mathcal{C}}^{N-1} \\ \hat{u}_{\mathcal{C}}^{\max}}} \max_{\substack{u_{\mathcal{C}}^{0},...,u_{\mathcal{C}}^{N} \\ \|R_{w_{\mathcal{C}}}\hat{u}_{\mathcal{C}}^{\max}\|_{1}^{2} + \sum_{l=0}^{N_{p}-1} \|r_{\mathcal{C}}^{l} - y_{\mathcal{C}}^{l}\|_{1}^{2} + \|R_{u_{\mathcal{C}}}u_{\mathcal{C}}^{l}\|_{1}^{2} + \|r_{\mathcal{C}}^{N_{p}} - y_{\mathcal{C}}^{N_{p}}\|_{1}}}$$
s. t. (8)
$$y_{\mathcal{C}}^{l} \in \mathcal{Y}_{\mathcal{C}}, l = 1, \dots, N_{p} - 1$$

$$y_{\mathcal{C}}^{N_{p}} \in \Omega_{\mathcal{C}}$$

$$u_{\mathcal{C}}^{l} \leq \hat{u}_{\mathcal{C}}^{\max} \leq u_{\mathcal{C}}^{\max}$$

$$w_{\mathcal{C}}^{l} \leq w_{\mathcal{C}}^{\max}$$
(10)

The weighting matrices R_{u_c} and R_{w_c} are block diagonal, Ω_c is the aggregated terminal set and r_c^l and x_c^0 are aggregated vectors containing the corresponding imposed references and initial state values, respectively. $r_c^{N_p}$ and $y_c^{N_p}$ are aggregated vectors containing the corresponding imposed references and output predictions values at time $k + N_p$, respectively. u_c^l , w_c^l and y_c^l are are aggregated vectors containing the corresponding input, uncertainty and output sequences, respectively. \hat{u}_c^{max} is an aggregated vector containing the corresponding self-imposed input limits. u_c^{max} and w_c^{max} are aggregated vectors containing the corresponding input and uncertainty limits, respectively.

2.2.3. C-Dmpc Algorithm

To summarize the C-DMPC methodology, the following pseudo-code is provided: With regard to Algorithm 2, the following observations are in order:

- the default uncertainty value used in Step 1 is selected to ensure that optimization
 problems from Step 3 are feasible, thus ensuring that the proposed methodology is
 recursively stable (i.e., the terminal set for the coalition is obtained by aggregating the
 terminal sets of the involved individual agents).
- if the condition from Step 6 is satisfied, then at that sampling period, the working framework is non-cooperative DMPC; otherwise the framework changes to coalitional DMPC (since at least one coalition is activated).

- the priority value, which is used as a condition term to initialize a coalition, is defined by each agent as a random sub-unitary number. In this manner, there is no use of a hierarchical control level to assign these priorities.
- in the extreme, all the agents can be combined in a coalition (C = N), which corresponds to a centralized MPC working framework.
- one or more coalitions can be active simultaneously and are dissolved at the end of each sampling period.

Remark 2 ([28]). In Algorithm 2, the coalitional control problem is feasible (i.e., Step 6. (c). ii.), because $W_C \subseteq \prod_{i \in C} W_i$, $U_C = \prod_{i \in C} U_i$ and $\Omega_C = \prod_{i \in C} \Omega_i$. The stability of the coalition is ensured by the terminal constraint set of the coalition, which is calculated as the Minkowski sum of the terminal sets polytopes defined for each individual agent from the coalition. The coalitional algorithm is recursive-feasible, contingent on Step 3, for which all the optimization problems are feasible, i.e., for which systems can work in a decentralized fashion.

Next, the C-DMPC methodology is validated in simulation, and the results are provided in Section 3.

3. Illustrative Example

In this section, the simulation results and discussion for the C-DMPC method are presented. The proposed simulation scenario for the cyber-physical multi-agent system described in Section 2, Figure 1, has the following characteristics:

• Four heterogeneous discrete-time sub-systems S_i , $\forall i \in \{1, ..., 4\}$, coupled in a chain architecture were defined using (1), with the following numerical matrices:

$$S_{1}: A_{p}^{1,1} = \begin{bmatrix} 0.7913 & 0.2020 \\ 0.1010 & 0.8417 \end{bmatrix} \quad B_{p}^{1,1} = \begin{bmatrix} 0.0271 \\ 0.2291 \end{bmatrix}$$
$$B_{p}^{1,0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad C_{p}^{1} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$
(11)

$$S_{2}: A_{p}^{2,2} = \begin{bmatrix} 0.7936 & 0.1996 \\ 0.1198 & 0.8236 \end{bmatrix} B_{p}^{2,2} = \begin{bmatrix} 0.0269 \\ 0.2265 \end{bmatrix}$$
$$B_{p}^{2,1} = \begin{bmatrix} 0.0004 \\ 0.0034 \end{bmatrix} C_{p}^{2} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$
(12)

$$S_{3}: A_{p}^{3,3} = \begin{bmatrix} 0.7888 & 0.2043 \\ 0.0817 & 0.8604 \end{bmatrix} B_{p}^{3,3} = \begin{bmatrix} 0.0273 \\ 0.2316 \end{bmatrix}$$
$$B_{p}^{3,2} = \begin{bmatrix} 0.0004 \\ 0.0034 \end{bmatrix} C_{p}^{3} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$
(13)

$$S_{4}: A_{p}^{4,4} = \begin{bmatrix} 0.7912 & 0.1994 \\ 0.0997 & 0.8211 \end{bmatrix} \quad B_{p}^{4,4} = \begin{bmatrix} 0.0269 \\ 0.2263 \end{bmatrix}$$
$$B_{p}^{4,3} = \begin{bmatrix} 0.0004 \\ 0.0034 \end{bmatrix} \quad C_{p}^{4} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$
(14)

• The limit constraints for the inputs, disturbances and outputs are the following:

$$\begin{cases} u_i^{\min} = -5 \\ u_i^{\max} = 5 \end{cases} \begin{cases} w_i^{\min} = -1 \\ w_i^{\max} = 1 \end{cases} \begin{cases} y_i^{\min} = -8 \\ y_i^{\max} = 8 \end{cases} \quad \forall i \in \{1, \dots, 4\}$$
 (15)

• For all sub-systems S_i , $\forall i \in \{1, ..., 4\}$, the following optimization parameters are used: the prediction horizon $N_p = 5$, the input weights $R_{u_i} = 0.1$ and $R_{w_i} = 0.01$.

Remark 3. The optimization parameters were carefully selected after a thorough analysis from the point of view of achieved performances. Several tests were performed, with different values for the weights and the prediction horizon. The chosen values ensured the best performances.

• The feedback laws were computed using classical state-feedback control based on the Ackermann's formula [31], applied for the extended model (4), obtaining:

$$K_{1} = \begin{bmatrix} 0.5494 & -2.6061 & 0.7488 \end{bmatrix}, \\ K_{2} = \begin{bmatrix} 0.3473 & -2.5199 & 0.7047 \end{bmatrix}, \\ K_{3} = \begin{bmatrix} 0.6249 & -2.6565 & 0.7401 \end{bmatrix}, \\ K_{4} = \begin{bmatrix} 0.4355 & -2.5107 & 0.7051 \end{bmatrix}.$$
(16)

Remark 4. The Ackermann's formula [31] was used to achieve specific closed-loop transient performances, chosen as an overshoot value of 5% and settling time of 5 time units, for subsystems S_1 and S_3 , and an overshoot value of 4%, and the same settling time, corresponding to sub-systems S_2 and S_4 . These performance values, were accordingly selected based on each sub-system dynamics.

• The reference tracking scenario was constructed for 12 time samples, using a sampling period $T_s = 0.25s$, with the following imposed references:

$r_1 = [0.2]$	0.2	0.2	0.2	0.2	0.2	0.5	0.5	0.5	0.5	0.5	0.5],	
$r_2 = [0.2]$	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2],	(17)
$r_3 = [0.2]$	0.2	0.2	0.2	0.5	0.5	0.5	0.5	0.2	0.2	0.2	0.2],	(17)
$r_4 = [0.2]$	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2].	

Since our proposed scenario has four sub-systems in a chain architecture with unidirectional communication links between the agents, there are eight possible frameworks including coalitions of two, three or four agents defined as follows:

- 1. default case—no coalitions between A_1 , A_2 , A_3 , A_4 ;
- 2. coalition C_{12} between A_1 and A_2 , while A_3 , A_4 remain outside the coalition but interconnected;
- 3. coalition C_{123} between A_1 , A_2 , and A_3 , while A_4 remains outside the coalition but interconnected;
- 4. twosimultaneous active coalitions C_{12} and C_{34} between A_1 and A_2 and A_3 and A_4 , respectively, which are interconnected;
- 5. coalition C_{23} between A_2 and A_3 , while A_1 , A_4 remain outside the coalition but interconnected;
- 6. coalition C_{234} between A_2 , A_3 and A_4 , while A_1 remains outside the coalition but interconnected;
- 7. coalition C_{34} between A_3 and A_4 , while A_1 , A_2 remain outside the coalition but interconnected;
- 8. extreme case: coalition C_{1234} between all agents A_1 , A_2 , A_3 , A_4 .

Algorithm 2

Initialization: For each agent A_i , $\forall i \in \mathcal{N}$, compute a table \mathcal{T}_i , with potential terminal sets Ω_i .

At each sampling time *k*, each agent A_i , $\forall i \in N$, receives the local state value and performs the following steps: 1. Computes the uncertainty polytope using default limit values for the constraints:

$$\mathcal{W}_i = B_{i,i-1} \mathcal{U}_{i-1}^0 \mathcal{U}_{i-1} = [u_{i-1}^{\max,0}; -u_{i-1}^{\max,0}].$$

- 2. Searches in the predefined table \mathcal{T}_i for a terminal set Ω_i^0 that includes the default uncertainty $\mathcal{W}_i \subseteq \Omega_i^0$.
- 3. Solves the local optimization problem (6) and obtains the optimal values $U_i^{*,0}$, $\hat{u}_i^{\max,0}$ using the default values $\Omega_i = \Omega_i^0$ for the terminal set and the uncertainty constraint limit ($w_i^{\max} = u_{i-1}^{\max,0}$).
- 4. Broadcasts to its successor the local optimal value $\hat{u}_i^{\max,0}$ and receives the corresponding value $\hat{u}_{i-1}^{\max,0}$ from its predecessor.
- 5. Repeats Steps 1–3 using the uncertainty constraint value received in Step 4.
- 6. Checks the feasibility of the local optimization problem:
- *If* the optimization problem from Step 5 is feasible:
- *then*: Coalitions between agents are not necessary. Each local agent A_i sends to its sub-system S_i , the first value from the optimal trajectory U_i^* ;
- *else*: Coalitions between agents are necessary. In this case, in order to be included in a coalition, each agent A_i , $\forall i \in N$, performs the following steps:
 - a. Receives, from its predecessor, a coalitional report containing the following information: the feasibility status (for the local optimization problem solved at Step 5) and priority value relating to all the predecessor agents from the chain architecture.
 - b. Sends to its successor, the updated coalitional report (i.e., all the relevant information received, together with its own local feasibility and priority data).
 - c. Initializes a coalition only if its local priority is the highest from the report. Within a coalition between two agents, the following steps are performed:
 - i. the coalition model is defined as (8);
 - ii. the optimization problem (10) subject to (9) is solved.
 - iii. the relevant information is broadcast to the coalition's neighbour.
 - iv. a feasibility check for all the optimization problems is done.
 - If the all the optimization problems are feasible:
 - *then*: The existing coalition was successful and can be dissolved after every sub-system S_i receives the first value from the optimal trajectory U_i^* ;
 - *else*: The existing coalition was not successful. Another agent must be included in the existing coalition (if the coalition's status is infeasible), or another coalition can be activated (if more agents outside the existing coalition have infeasible problems). At this stage, Step (c) is repeated as necessary.
- 7. End algorithm.

Remark 5. Please note that our proposed Coalitional DMPC algorithm is tailored specifically for cyber-physical multi-agent systems. The key feature is its capability to switch between control architectures, whenever the feasibility of the multi-agent system is lost, due to uncertainties in the local sub-systems. One example of such multi-agent system is a vehicle platoon. In this case, it is clear that classical centralized MPC is not suitable for controlling this application. Moreover, decentralized MPC, in which the couplings between sub-systems are ignored, can render instability within the platoon. One compromise solution is distributed MPC, in which the interactions are taken into account when computing the local solutions. However, if the distributed MPC (i.e., in non-cooperative framework) fails at this task, then our proposed coalitional DMPC provides a backup plan, namely to merge different sub-systems into coalitions. Inside a coalition, all the information is known; thus, only the coupling signals with sub-systems outside the coalition must be accounted for.

The invariant sets obtained for sub-system S_1 using Algorithm 1 presented in Section 2.1 are depicted in Figure 2. For the computation, the following numerical values were used:

 $u^{max} = 5$, $w^{max} = 5$, $step_{\alpha} = step_{\beta} = 0.5$. As expected, the larger invariant set (depicted with red colour) was obtained for $\alpha = u^{max} = 5$ and $\beta = 0.1$. Moreover, as the constraint limits become smaller, the set Ω_i decreases in dimension and is included in the larger red set (Figure 3—the sets plotted with green, blue, magenta and black colours, respectively). Since, the state variable for the extended model (4) has three values, the computed invariant sets are three dimensional and can be plotted as convex hulls (ref. Figures 2 and 3). This graphical representation of the invariant sets, which are predefined options for the terminal set constraint (6), are also useful when defining the reference target for the multi-agent system. Thus, one must take into account that the imposed trajectory for each sub-system S_i , $\forall i \in \{1, ..., 4\}$ should be placed in the interior of the invariant set.



Figure 2. Depiction of the predefined invariant sets corresponding to sub-system *S*1, computed for uncertainty constraint limit value $\beta = 0.1$.



Figure 3. Detail regarding the depiction of the predefined invariant sets corresponding to sub-system *S*1, computed for uncertainty constraint limit value $\beta = 0.1$.

Remark 6. It is worth mentioning that the step values $step_{\alpha} = step_{\beta} = 0.5$ were selected taking into account the numerical values of the input and uncertainty constraints to ensure a sufficient number of invariant sets computed. If a smaller value, e.g., $step_{\alpha} = step_{\beta} = 0.1$ is chosen, the result would be an increased size for the table containing the invariant sets. However, as depicted in Figure 2, these values also parametrize the dimensions of the invariant set polytopes. Thus, although we would have more available sets, their dimensions would be too similar, to justify the involved computational costs.

The reference tracking results and the formation of the coalitions during the simulation are presented in Figure 4.

As depicted in Figure 4, lower subplot, during the first seven time steps, the simulation runs in the default scenario, in which all the agents solve a non-cooperative DMPC algorithm without being involved in a coalition. This is marked with blue circles, at each time step, for each agent A_i , $\forall i \in \{1, \dots, 4\}$. At time step 8, due to the setpoint change of 0.3 units in r_3 for sub-system S_3 and the corresponding increase in the control effort u_3 , the local feasibility for sub-system S_4 is lost. Hence, the coalition C_{34} is activated, which is plotted with a red star marker for A_3 and A_4 . At the next time step, coalition C_{234} between agents A_2 , A_3 and A_4 is active and is coupled with the remaining agent A_1 , because sub-system S_2 is dynamically coupled through the input with S_1 and their corresponding agents share information. At time step 10, coalition C_{34} is activated, and for the remaining two time steps of the simulation, coalition C_{234} is active. Moreover, the reference tracking results show that all the imposed set-points are successfully reached in one sampling time, with zero offset error. This occurs for the first seven time steps, in which all the agents work outside a coalition, and also for the remaining simulation time, when coalitions of two or three agents are necessary to maintain the feasibility of the CP-MAS. The results clearly prove the efficiency of our proposed C-DMPC method in a reference tracking scenario.



Figure 4. Reference tracking results (subplots 1–4), control efforts (subplots 5–8) and the corresponding coalitions formation (subplot 9) for a cyber-physical multi-agent system composed of 4 interconnected sub-systems.

4. Conclusions

In this work, a coalitional distributed model predictive (C-DMPC) methodology suitable for input coupled cyber-physical multi-agent systems was proposed. The algorithm was tailored for an in-chain system architecture with unidirectional communication links (i.e., the coupling information viewed as an uncertainty in the local nominal model was broadcast from a predecessor sub-system to a successor). The methodology was validated in simulation, using an academic cyber-physical multi-agent system as a proof of concept for the proposed algorithm. The simulation results show that if the uncertainty level received by the local agent is manageable, a non-cooperative DMPC algorithm could be locally solved. However, when the local feasibility of the optimization problem was lost, then forming coalitions between agents showed satisfactory performance and the usefulness of the C-DMPC algorithm was proven.

Future work will test the efficiency of the proposed algorithm on a vehicle platooning application.

5. Materials and Methods

The simulations from this work were performed using MATLAB R2020b on Windows 10, 64-bit Operating System with a laptop Intel Core i7-9850H CPU @ 2.60 GHz and 16 GB RAM.

The optimizations were implemented using the YALMIP toolbox [34].

Author Contributions: Conceptualization, A.M. and C.-F.C.; methodology, A.M.; software, A.M.; validation, A.M. and C.-F.C.; formal analysis, A.M.; investigation, A.M.; resources, A.M.; data curation, A.M.; writing—original draft preparation, A.M.; writing—review and editing, A.M. and C.-F.C.; visualization, A.M.; supervision, C.-F.C.; project administration, A.M. and C.-F.C.; funding acquisition, A.M. and C.-F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by two grants of the Ministry of Research, Innovation and Digitization, CNCS/CCCDI – UEFISCDI, project numbers PN-III-P1-1.1-PD-2019-0757 and 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. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

MPC	Model Predictive Control
DMPC	Distributed Model Predictive Control
C-MPC	Coalitional Model Predictive Control
C-DMPC	Coalitional Distributed Model Predictive Control
CP-MAS	Cyber-Physical Multi-Agent System
CPsS	Cyber-Physical sub-system

References

- 1. Gomes, L.; Almeida, C.; Vale, Z. Recommendation of Workplaces in a Coworking Building: A Cyber-Physical Approach Supported by a Context-Aware Multi-Agent System. *Sensors* 2020, *20*, 3597. [CrossRef] [PubMed]
- Dumitrache, I.; Caramihai, S.I.; Moinescu, M.A.; Sacala, I.S.; Vladareanu, L.; Repta, D. A Perceptive Interface for Intelligent Cyber Enterprises. Sensors 2019, 19, 4422. [CrossRef] [PubMed]
- 3. Ierardi, C.; Orihuela, L.; Jurado, I. Distributed Estimation Techniques for Cyber-Physical Systems: A Systematic Review. *Sensors* **2019**, *19*, 4720. [CrossRef]

- 4. Dumitrache, I.; Sacala, I.S.; Moinescu, M.A.; Caramihai, S.I. A Conceptual Framework for Modeling and Design of Cyber-Physical Systems. *Stud. Inform. Control* 2017, *26*, 325–334. [CrossRef]
- 5. Lee, E.A. The Past, Present and Future of Cyber-Physical Systems: A Focus on Models. Sensors 2015, 15, 4837–4869. [CrossRef]
- 6. 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]
- Huang, D.; Li, H.; Li, X. Formation of Generic UAVs-USVs System Under Distributed Model Predictive Control Scheme. *IEEE Trans. Circ. Syst. Express Briefs* 2020, 67, 3123–3127. [CrossRef]
- 8. Franzè, G.; Casavola, A.; Famularo, D. Distributed Receding Horizon Control of Constrained Networked Leader–Follower Formations Subject to Packet Dropouts. *IEEE Trans. Control Syst. Technol.* **2017**, *25*, 1798–1809. [CrossRef]
- Maxim, A.; Caruntu, C.F.; Lazar, C. Cruise and headway control for vehicle platooning using a distributed model predictive control algorithm. In Proceedings of the 21st International Conference on System Theory, Control and Computing, Sinaia, Romania, 19–21 October 2017; pp. 146–151.
- Sasaki, S.; Namerikawa, T. Traffic Signal Control Considering Switching Timing via Distributed Model Predictive Control. In Proceedings of the 19th International Conference on Control, Automation and Systems, Jeju, Korea, 15–18 October 2019; pp. 286–291.
- 11. Zheng, Y.; Wei, Y.; Li, S. Coupling Degree Clustering-Based Distributed Model Predictive Control Network Design. *IEEE Trans. Autom. Sci. Eng.* **2018**, *15*, 1749–1758. [CrossRef]
- 12. Meng, K.; Dong, Y.; Xu, Z.; Weller, S.R. Cooperation-Driven Distributed Model Predictive Control for Energy Storage Systems. *IEEE Trans. Smart Grid* 2015, *6*, 2583–2585. [CrossRef]
- 13. 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]
- 14. Lou, G.; Gu, W.; Sheng, W.; Song, X.; Gao, F. Distributed Model Predictive Secondary Voltage Control of Islanded Microgrids With Feedback Linearization. *IEEE Access* 2018, *6*, 50169–50178. [CrossRef]
- 15. Zhu, J.; Xue, B. Distributed Predictive Control of Multi-Agent Systems Based on Error Upper Bound Approach. *IEEE Access* 2021, 9, 11470–11478. [CrossRef]
- 16. Tang, X.; Li, M.; Wei, S.; Ding, B. Event-triggered Synchronous Distributed Model Predictive Control for Multi-agent Systems. *Int. J. Control. Autom. Syst.* **2021**, *19*, 1–10. [CrossRef]
- 17. Li, H.; Li, X. Distributed Model Predictive Consensus of Heterogeneous Time-Varying Multi-Agent Systems: With and Without Self-Triggered Mechanism. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *67*, 5358–5368. [CrossRef]
- 18. Maestre, J.M.; Negenborn, R.R. Distributed Model Predictive Control Made Easy; Springer: Berlin/Heidelberg, Germany, 2014.
- 19. Fele, F.; Maestre, J.M.; Camacho, E.F. Coalitional control: Cooperative game theory and control. IEEE Control Syst. 2017, 37, 53–69.
- Fele, F.; Debada, E.; Maestre, J.M.; Camacho, E.F. Coalitional Control of Self-Organizing Agents. *IEEE Trans. Autom. Control.* 2018, 63, 2883–2897. [CrossRef]
- Masero, E.; Fletcher, L.A.; Maestre, J.M. A Coalitional Model Predictive Control for the Energy Efficiency of Next-Generation Cellular Networks. *Energies* 2020, 13, 6546. [CrossRef]
- 22. Masero, E.; Fletcher, L.A.; Maestre, J.M. A Coalitional Model Predictive Control Approach for Heterogeneous Cellular Networks. In Proceedings of the 2020 European Control Conference, Saint Petersburg, Russia, 12–15 May 2020; pp. 448–453.
- 23. Masero, E.; Maestre, J.M.; Francisco, M.; Camacho, E.F. Coalitional MPC with predicted topology transitions. *IFAC PapersOnLine* **2020**, *52*, 3342–3347. [CrossRef]
- 24. Baldavieso Monasterios, P.R.; Trodden, P.A.; Cannon, M. On feasible sets for coalitional MPC. In Proceedings of the 58th IEEE Conference on Decision and Control, Nice, France, 11–13 December 2019; pp. 4668–4673.
- 25. Maestre, J.M.; Muñoz de la Peña, D.; Camacho, E.F. Distributed model predictive control based on a cooperative game. *Optim. Control. Appl. Methods* **2011**, *32*, 153–176. [CrossRef]
- 26. Scattolini, R. Architectures for distributed and hierarchical Model Predictive Control—A review. J. Process. Control 2009, 19, 723–731. [CrossRef]
- 27. Maxim, A.; Maestre, J.M.; Caruntu, C.F.; Lazar, C. Min-max coalitional model predictive control algorithm. In Proceedings of the 22nd International Conference on Control Systems and Computer Science, Bucharest, Romania, 28–30 May 2019.
- Maxim, A.; Maestre, J.M.; Caruntu, C.F.; Lazar, C. Robust coalitional distributed model predictive control algorithm with stability via terminal constraint. In Proceedings of the IEEE Conference on Control Technology and Applications, Copenhagen, Denmark, 21–24 August 2018; pp. 964–969.
- Maxim, A.; Pauca, O.; Caruntu, C.; Lazar, C. Distributed Model Predictive Control Algorithm with Time-Varying Communication Delays for a CACC Vehicle Platoon. In Proceedings of the 24th International Conference on System Theory, Control and Computing, Sinaia, Romania, 8–10 October 2020.
- Maxim, A.; Lazar, C.; Caruntu, C. 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.
- 31. Ogata, K. Modern Engineering Control, 5th ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2010.

- 32. Kwakernaak, H.; Sivan, R. Linear Optimal Control Systems; John Wiley & Sons, Inc.: New York, NY, USA, 1972.
- 33. Maestre, J.M.; Muñoz de la Peña, D.; Camacho, E.F.; Alamo, T. Distributed model predictive control based on agent negotiation. *J. Process. Control* 2011, 21, 685–697. [CrossRef]
- 34. Löfberg, J. YALMIP: A Toolbox for Modeling and Optimization in MATLAB. In Proceedings of the IEEE International Symposium on Computer Aided Control Systems Design, New Orleans, LA, USA, 2–4 September 2004; pp. 284–289.