electronics

MDPI

Article

Coalitional Distributed Model Predictive Control Strategy with Switching Topologies for Multi-Agent Systems

Anca Maxim, Ovidiu Pauca and Constantin F. Caruntu *

Department of Automatic Control and Applied Informatics, "Gheorghe Asachi" Technical University of Iasi, 700050 Iasi, Romania; anca.maxim@ac.tuiasi.ro (A.M.)
* Correspondence: caruntuc@ac.tuiasi.ro

Abstract: Controlling multi-agent systems (MASs) has attracted increased interest within the control community. Since the control challenge consists of the fact that each agent has limited local capabilities, our adopted solution is tailored so that a group of such entities works together and shares resources and information to fulfill a given task. In this work, we propose a coalitional control solution using the distributed model predictive control (DMPC) framework, suitable for a multi-agent system. The methodology has a switching mechanism that selects the best communication topology for the overall system. The proposed control algorithm was validated in simulation using a homogeneous vehicle platooning application with longitudinal dynamics. The available communication topologies were specifically tailored taking into account the information flow between adjacent vehicles. The obtained results show that when the platoon's string stability is risked, the algorithm switches between different communication topologies. The resulting coalitions between vehicles ensure an increase in the overall stability of the entire system and prove the efficacy of our proposed methodology.

Keywords: coalitional control; distributed model predictive control; predecessor–follower string stability; switching communication topologies; multi-agent systems

check for updates

Citation: Maxim, A.; Pauca, O.; Caruntu, C.F. Coalitional Distributed Model Predictive Control Strategy with Switching Topologies for Multi-Agent Systems. *Electronics* **2024**, *13*, 792. https://doi.org/10.3390/electronics13040792

Academic Editor: Boris Andrievsky

Received: 29 December 2023
Revised: 7 February 2024
Accepted: 16 February 2024
Published: 18 February 2024

1. Introduction

Nowadays, there is much interest in the control of multi-agent systems (MASs). According to the work in [1], MASs are those systems that are composed of multiple autonomous units (hereafter called agents), each one with limited access to information, that need to communicate and collaborate to achieve a common task. Depending on the communication used to ensure a desired global behavior, MASs can be classified as [2] (i) MASs based on agent-to-agent communication, in which both the local controllers and the communication topology must be accordingly designed, and (ii) MASs based on broadcast, where a global controller analyzes the group performance and broadcasts a signal to all local controllers. According to [3], from the point of view of the problem formulation, MASs can solve various problems such as (i) consensus control, in which all local states must converge to a common value agreed upon among the agents [4], (ii) formation control, where all agents must move in a predefined formation shape [5], and (iii) containment control, in which a subset of agents are considered leaders and dictate the safe regions for the follower agents to move [6].

As stated in the survey work on intelligent control of MASs [7], one key limitation in achieving intelligent control of a MAS is given by the communication and sensing capabilities of each agent, which are usually constrained by the limited range of perception or limited bandwidth within the network resources. Moreover, even with this challenge solved, there are still open issues when dealing with MASs, such as assuring network security concerning cyber attacks, designing fully distributed control frameworks that ensure the global stability of the MAS, testing control strategies on heterogeneous applications, or validating the control strategies on large-scale practical applications. There are manifold control strategies suitable for MASs, such as gain feedback robust control [8], the distributed Kalman filtering algorithm for estimation and tracking applications [9], robust model predictive control (RMPC) [10], and distributed model predictive control (DMPC) [11], among others. In [12], a DMPC algorithm for a linear quadratic consensus problem of MASs is provided. The idea is to optimize at each sampling period both the final consensus state and the input trajectory to achieve the steady-state response in addition to the transient response of the MAS. In [13], an event-triggered synchronous DMPC for MASs is proposed, where the DMPC problem is solved only when the triggering condition is satisfied. In [14], a DMPC algorithm with energy management is compared with a proportional-integral-derivative (PID) controller, using an electric vehicle platoon with longitudinal motion, with leader–predecessor following communication topology. In [15], a DMPC-based string-stable platoon with robustness against communication delays is proposed. A string stability constraint is imposed in the local optimization problem.

A common test case for MASs is a switching topology scenario, since, due to environmental faults, the communication topology established between the agents can change. For example, in a vehicle-to-vehicle communication network established within an MAS, the communication between agents can be disconnected due to communication range or external disturbances [16]. In [17], the consensus problem is solved using a model predictive control (MPC) algorithm, which is designed to ensure a fast convergence to consensus for all the MASs in the presence of input constraints and switching topologies. The algorithm is tested on five MASs with double-integrator dynamics, connected via two communication topologies, which are switched periodically within a sampling period. In [18], a distributed attitude synchronization control of MASs with switching topologies is presented. The algorithm is validated using five agents (rigid bodies) with absolute and relative rotation movements, respectively. The agents are assumed to be connected via two interaction graphs, which are switched periodically, following a predefined switching signal. In [19], distributed reconfiguration strategies for networked MASs with switching topologies in the presence of actuator faults are proposed. Two distributed control laws are tested using a seven-agent system and three periodically switching network graph topologies. In [20], a car-following model useful for a vehicle platooning application with fixed and switching communication topologies is presented. Three communication topologies regarding the information flow in the entire platoon are proposed, which are switched according to a periodic and predefined schedule.

A common characteristic of the above-mentioned works is that the switching mechanism is periodic and predefined. There are several communication topologies designed for the MASs, and, within the simulation scenario, there is a periodic switching signal that selects the activation of a certain topology. Another control approach for MASs is reformulating the problem in the coalitional control framework [21], which is a clustering approach in which the activation of the communication links is penalized to prevent unnecessary information exchange. The idea is to merge local agents into coalitions or clusters, which share relevant information via enabled communication links to solve their common goal. In [22], a coalitional, robust MPC algorithm is given, in which different coalitions between agents are formed using a consensus-based algorithm with potential games. In [23], a coalitional, robust, tube-based MPC algorithm for tracking target sets is proposed. The switching mechanism between different communication topologies is based on the periodic evaluation of an unconstrained performance cost index. In [24], a coalitional, robust, tube-based MPC strategy with plug-and-play capabilities is presented. The coalitions between different agents are formed when the feasibility of the optimization problems is lost due to local disturbances. In [25], a robust, coalitional MPC with a hierarchical architecture is proposed. The transitions between different topologies are predicted by each agent by introducing a new variable in the optimization problem.

In this work, we focus on the formation control of an MAS with agent-to-agent communication. To this end, we formulate a DMPC algorithm in the coalitional control framework, hereafter called (C-DMPC). The proposed C-DMPC algorithm is tested in a vehicle platooning application. The coalitions between different agents are formed when the string stability of the platoon is lost. For example, if a vehicle indexed i is not string stable with respect to vehicle $i-1$ (i.e., its longitudinal error is larger than the longitudinal error of its predecessor), a coalition will form between vehicle i and vehicle $i-1$.

The most recent results by the authors obtained in the coalitional control domain can be classified depending on the methodology used to control the MAS, as follows: (i) coalitional control based on an optimal feedback gain matrix [26], in which each communication topology is described by a different optimal feedback gain matrix, with non-zero elements corresponding to active communication links, and (ii) coalitional control based on a robust DMPC framework [27,28] with a robust, min-max DMPC algorithm, in which the coalitions between agents are formed when the local feasibility with respect to a robust positive invariant terminal set constraint is lost. Furthermore, in [29], a comparative assessment is performed on a vehicle platooning application by evaluating the results achieved using a DMPC strategy and a coalitional control algorithm, formulated using an optimal feedback gain matrix. Thus, the idea is to investigate the performance of the DMPC method, compared with the coalitional control with distributed and decentralized communication topologies, respectively.

With respect to similar works from the literature, the main contributions of this work are summarized as follows:

1. A DMPC-based coalitional control strategy is developed, in which the reconfiguration of the communication network is jointly decided depending on local string stability criteria. This is different from the DMPC algorithm in [30], where the network topology changes by inserting or removing certain agents, or the robust, min-max DMPC algorithm in [27,28], in which the coalitions are formed when the local feasibility of the optimization problem is lost due to the fact that a terminal constraint is not fulfilled.
2. An automatic procedure to switch between different communication topologies is designed, based on a string stability index evaluation. This evaluation is performed periodically, but the period is not fixed and depends on the transient response of each networked sub-system. This means that a coalition between two sub-systems is formed if the string stability criteria between them are not fulfilled (i.e., the output error between two adjacent sub-systems is increasing). An alternative condition for switching between topologies, which is based on a performance index evaluation for each possible topology, performed at periodic time intervals can be found in [23].
3. The DMPC algorithm with the string stability constraint introduced in [29] was extended in a coalitional DMPC framework, using as a switching deciding factor the evaluation of a string stability index, computed outside of the local optimization problems. Thus, in order to decide if a coalition between sub-systems is required, resulting in a change in the communication topology, the string stability index is assessed by each sub-system, and, depending on the result, a coalition is formed (i.e., the communication topology is switched).
4. A comprehensive analysis of the performance achieved with each individual communication topology is performed. Moreover, three simulation scenarios for a homogeneous vehicle platoon are executed to evaluate the selection between different available communication topologies (i.e., corresponding to certain coalitions between sub-systems).

The remainder of this paper has the following structure: Section 2 presents the proposed coalitional distributed model predictive control (C-DMPC) strategy formulation. The methodology is tested in simulation using a vehicle platooning application. The platoon model is provided in Section 3, while the simulation results and discussion are given in Section 4. The conclusions and future work ideas are presented in Section 5.

2. Coalitional Distributed Model Predictive Control (C-DMPC) Strategy

In this section, the proposed C-DMPC method is provided. Starting from the DMPC algorithm description given in [29], a coalitional DMPC strategy is formulated.

2.1. Preliminaries

Consider a multi-agent system composed of N_S sub-systems dynamically coupled through the state and input vectors. The agents controlling each sub-system i, $\forall i \in \mathcal{S} = \{1, \ldots, N_S\}$, are connected with the agents controlling the sub-systems j, $\forall j \in \mathcal{S} - \{i\}$, if the coupling matrices $A_{ij}, B_{ij} \neq 0$.

The discrete-time, state-space model for sub-system $i \in \mathcal{S}$ is the following:

$$\begin{aligned} x_i(k+1) &= A_i x_i(k) + B_{i,i} u_i(k) + \sum_{j \in \mathcal{S}-\{i\}} \Big(A_{ij} x_j(k) + B_{ij} u_j(k) \Big) \\ y_i(k) &= C_i x_i(k) \end{aligned} \tag{1}$$

with k being the discrete time index. The input, output, and state variables are denoted with $u_i \in \mathbb{R}$, $y_i \in \mathbb{R}$, and $x_i \in \mathbb{R}$, respectively. Sub-system i is dynamically coupled with sub-system j through the state $x_j \in \mathbb{R}$ and input $u_j \in \mathbb{R}$ variables. In particular, the first sub-system with the index $i = 1$ is not coupled. Thus, the coupling term $\sum_{j \in \mathcal{S}-\{1\}} A_{1j} x_j(k) + B_{1j} u_j(k) = x_r(k)$ is replaced by the reference imposed for sub-system $i = 1$. Note that matrices A_i, $B_{i,i}$, $A_{i,j}$, $B_{i,j}$, and C_i have adequate dimensions.

For all sub-systems $i \in \mathcal{S}$, linear input and output hard constraints are imposed as:

$$u_i \in \mathbb{U}_i, \; y_i \in \mathbb{Y}_i, \tag{2}$$

where $\mathbb{U}_i$ and $\mathbb{Y}_i$ are defined by sets of linear inequalities.

All sub-systems (1) can exchange information using a communication network described by the graph $(\mathcal{S}, \mathcal{L})$, where $\mathcal{S}$ represents the set of agents and $\mathcal{L}$ represents the set of communication links. Each link $l_{i,j} \in \mathcal{L}$, connects a pair of agents $\{i, j\} \subseteq S, i \neq j$. A network topology is defined as a configuration of communication links between agents. Inside of a network topology, the sub-systems form a coalition C. The members of a coalition C cooperatively compute their future inputs $u_C = (u_i)_{i \in C}$. Hence, a coalition behaves as a single system described by [31]:

$$x_C(k+1) = A_C x_C(k) + B_C u_i(k) + x_{dC}(k), \tag{3}$$

where $x_{dC} = \sum_{j \in \mathcal{S}-\{C\}} \big(A_{Cj} x_j(k) + B_{Cj} u_j(k) \big)$ and $x_C = (x_i)_{i \in C}$ and $u_C = (u_i)_{i \in C}$ denote the aggregate state and input vector composed of the sub-systems $i \in C$ included in the coalition. The corresponding state and input matrices for the coalitions are $B_C = [B_{i,i'}]_{i,i' \in C}$ and $A_C = [A_{i,i'}]_{i,i' \in C}$, respectively. For each coalition C, linear input and output hard constraints are imposed as:

$$u_C \in \prod_{i \in C} \mathbb{U}_i, \; y_C \in \prod_{i \in C} \mathbb{Y}_i, \tag{4}$$

where $\prod_{i \in C} \mathbb{U}_i$ is defined by the Cartesian product between each individual input constraint set, corresponding to each sub-system $i \in C$. The output constraint for the coalition C can be defined similarly.

The interconnected sub-system (1) must respect the string stability property to ensure that a change in the first sub-system's (i.e., the leader) states is attenuated along the upstream direction [32]. In this work, the predecessor–follower string stability condition is defined in terms of output error $e_i(k) = r_i(k) - y_i(k)$, as follows [33]:

Definition 1. *The multi-agent system with N_S sub-systems (1) is predecessor–follower string stable if, for each sub-system $i \in \{2, \dots, N_S\}$, there exists a constant $\alpha_i \in (0, 1)$ such that:*

$$\eta_i = ||e_i^*(l|k)||_\infty - \alpha_i \max_{s \in \{0,..,k\}} ||e_{i-1}^*(l|s)||_\infty \leq 0, \forall k \geq 0, l \in \{1, ..., N\}, \quad (5)$$

where e_i^* is the predicted trajectory error and $\|\cdot\|_\infty$ denotes the infinity-norm.

Notice that the stability condition (5) requires both the history of prediction for e_{i-1}^* and its current optimal predicted trajectory. The information at the current step k cannot be obtained but can be replaced with the prediction from the previous step [32]:

$$\eta_i = ||e_i^*(l|k)||_\infty - \alpha_i \min(\max_{s \in \{0,..,k-1\}} ||e_{i-1}^*(l|s)||_\infty, \max(|e_{i-1}^*(2|k-1)|, |e_{i-1}^*(3|k-1)|)) \leq 0, \quad (6)$$

where $|\cdot|$ denotes the absolute value. The value (6) represents the predecessor–follower string stability condition imposed for each sub-system in the local optimization problem.

Remark 1. *Between two adjacent sub-systems i and $i-1$, which are not in the same coalition, the predecessor–follower string stability constraint is defined as (6). However, if these two sub-systems are members of the same coalition, then the stability constraint becomes:*

$$\eta_i = ||e_i^*(l|k)||_\infty - \alpha_i \min(\max_{s \in \{0,..,k-1\}} ||e_{i-1}^*(l|s)||_\infty, \max(|e_{i-1}^*(2|k)|, |e_{i-1}^*(3|k)|)) \leq 0. \quad (7)$$

Note that, within a coalition, the stability constraint between adjacent sub-systems (7) is computed based on the prediction of e_{i-1} locally computed by sub-system i using the coalition model. This is different than the case in which sub-systems i and $i-1$ do not form a coalition and the information regarding the prediction of e_{i-1} used to compute (6) is received via a vehicle-to-vehicle (V2V) communication network.

Remark 2. *The predecessor–follower string stability condition provided in (5) was firstly introduced in [33]. In this work, we replace the string stability condition (5) with the string stability condition (6), following the interpretation provided in [32], in order to approximate missing current information with past known information. Furthermore, the stability condition (6) was firstly used as a constraint in the DMPC algorithm provided in [29].*

In what follows, a coalitional DMPC strategy to control the multi-agent system composed of N_S sub-systems (1) that ensures the string stability property is proposed.

2.2. C-DMPC—Optimization Problem

First, let us define the optimization problem to be solved over the prediction horizon N by each agent in charge of each sub-system $i \in \{1, \dots, N_S\}$:

Problem 1. *For each sub-system $i \in \{1, \dots, N_S\}$, at each discrete step k, given $x_i(k)$, solve:*

$$\min_{U_i(k)} J_i(x_i(k), U_i(k)), \quad (8)$$

subject to constraints (1), (2), and (6),
where

$$J_i(x_i(k), U_i(k)) = x_i^T(N|k) P_i x_i(N|k) + \sum_{l=0}^{N-1} \left(x_i^T(l|k) Q_i x_i(l|k) + u_i^T(l|k) R_i u_i(l|k) \right). \quad (9)$$

Note that the cost index (9) involves a terminal cost weighted by matrix $P_i \succ 0$ and a stage cost that penalizes the state x_i and the input u_i by the weight matrices $Q_i \succ 0$ and $R_i \succ 0$, respectively.

Second, as previously mentioned, when the string stability constraint (6) is not fulfilled, the agents will choose to merge into a coalition. Inside a coalition, the following optimization problem is solved:

Problem 2. *For each coalition C, at each discrete step k, given* $x_C(k)$*, solve:*

$$\min_{U_C(k)} J_C(x_C(k), U_C(k)) \tag{10}$$

subject to constraints (3), (4), *and* (7),
where

$$\begin{aligned} J_C(x_C(k), U_C(k)) =& x_C^T(N|k) P_C x_C(N|k) + \\ & \sum_{l=0}^{N-1} \left(x_C^T(l|k) Q_C x_C(l|k) + u_C^T(l|k) R_C u_C(l|k) \right), \end{aligned} \tag{11}$$

where $P_C \succ 0$, $Q_C \succ 0$, *and* $R_C \succ 0$ *are the weight matrices for the coalition's C terminal cost, state, and input, respectively.*

Notice that, in a coalition, each member $i \in C$ knows the model of the coalition and has to solve Problem 2 instead of Problem 1.

2.3. C-DMPC—Algorithm

In Algorithm 1, the algorithm for the C-DMPC strategy is provided.

Algorithm 1: Coalitional DMPC algorithm.

for *each discrete time k* **do**
- **if** $mod(k, N_C) == 0$ **then**
  - All sub-systems verify the stability condition (6):
  - **if** $\eta_i > 0$ **then**
    - **if** $i - 1 \in C$ **then**
      - Enter in the existing coalition of sub-system $i-1$;
    - **else**
      - Create a new coalition with sub-system $i-1$;
    - **end**
  - **else**
    - Sub-system i will not form a coalition with sub-system $i-1$;
  - **end**
- **end**
- All sub-systems share their current states within the coalition;
- Sub-system $i-1$ sends to sub-system i the state trajectory prediction, if sub-systems $i-1$ and i do not belong in the same coalition;
- Sub-system i solves Problem 2, if sub-system i belongs in a coalition, otherwise sub-system i solves Problem 1.

end

In the proposed Algorithm 1, the key step performed by each sub-system is to verify if the predecessor–follower string stability condition, i.e., (6) is fulfilled. To minimize the computational burden implied by checking this condition at each sampling period, condition (6) is investigated after every N_C^{th} sampling period. However, the N_C parameter is not static and is cooperatively selected by all agents between two limit values. More details regarding the method used to decide this parameter are provided in Section 3, Algorithm 2. Since each agent checks its own condition, depending on each local η_i value, either a new coalition between two adjacent sub-systems is formed, or a larger coalition is created. Within a coalition, all information is freely shared and Problem 2 is solved. Furthermore, if all local η_i values are negative, no coalition is formed, and each agent solves Problem 1.

Algorithm 2: Condition of determining N_C.

if $|a_i(k)| \leq \epsilon$ **then**
  $N_{Ci} = N_{CMAX}$;
else
  $N_{Ci} = N_{CMIN}$;
end
$N_C = \min(N_{Ci}), i \in \{2, \ldots, N_S\}$.

3. Vehicle Platoon—CASE Study

The proposed C-DMPC algorithm is applied to a vehicle platooning application to control the longitudinal dynamics. Since the platoon is coupled using a predecessor–following communication topology, it means that the information flow is unidirectional between adjacent vehicles.

The leader vehicle is modeled using the following state-space representation:

$$\begin{bmatrix} \dot{v}_1 \\ \dot{e}_{p_1} \\ \dot{a}_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & 0 & \frac{-1}{\tau} \end{bmatrix} \begin{bmatrix} v_1 \\ e_{p_1} \\ a_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} x_r \qquad (12)$$
$$y_1 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} x_1$$

where the state vector variables $x_1 = [v_1 \; e_{p_1} \; a_1]^T$ are the vehicle's velocity, position error, and acceleration, respectively; the control input u_1 is the desired acceleration, while the output y_1 is the position error e_{p_1} of the vehicle, which is computed with respect to an imposed velocity reference trajectory $x_r = r_{v_1}$. The time constant of the first-order vehicle model is denoted with τ.

Starting from the individual vehicle's model, a vehicle platoon is formed, in which the leader vehicle, indexed $i = 1$, follows an imposed velocity reference, whereas a follower vehicle, indexed $i \in \{2, \ldots, N_S\}$, needs to maintain a desired relative distance r_{d_i} with respect to the preceding vehicle indexed with $i - 1$. Let us assume a constant headway time policy imposed for the follower vehicles, defined as follows:

$$r_{d_i} = r + hv_i, \qquad (13)$$

where r is the standstill constant distance between the vehicles, h is the time headway, and v_i is the velocity of vehicle i.

According to the work in [34], the follower vehicle dynamics can be described starting from the computation of the inter-vehicle distance d_i as the difference between the position p_{i-1} of vehicle $i - 1$ and the position of the current vehicle p_i:

$$d_i = p_{i-1} - p_i. \qquad (14)$$

The position error between two consecutive vehicles denoted with e_{p_i} is computed as the difference between the inter-vehicle distance d_i from (14) and the headway time policy r_{d_i} given in (13) as follows:

$$e_{p_i} = d_i - r_{d_i} = p_{i-1} - p_i - r - hv_i. \qquad (15)$$

Using the definition of the position error from (15), the time derivative of the position error $\dot{e}_{p_i}$ is computed:

$$\dot{e}_{p_i} = \dot{p}_{i-1} - \dot{p}_i - \dot{r} - h\dot{v}_i = v_{i-1} - v_i - ha_i = e_{v_i} - ha_i, \tag{16}$$

where the velocity error between two consecutive vehicles e_{v_i} is computed as the difference between the velocity v_{i-1} of vehicle $i-1$ and the velocity of the current vehicle v_i. It results that the time derivative of the velocity error $\dot{e}_{v_i}$ is calculated as the difference between the two consecutive vehicle accelerations:

$$\dot{e}_{v_i} = \dot{v}_{i-1} - \dot{v}_i = a_{i-1} - a_i. \tag{17}$$

For each follower vehicle, $i \in \{2, \ldots, N_S\}$, the following continuous-time, state-space model is used:

$$\begin{aligned} \dot{x}_i &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -h \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & \frac{-1}{\tau} \end{bmatrix} \begin{bmatrix} v_i \\ e_{p_i} \\ e_{v_i} \\ a_i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{\tau} \end{bmatrix} u_i + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{i-1} \\ e_{p_{i-1}} \\ e_{v_{i-1}} \\ a_{i-1} \end{bmatrix} \\ y_i &= \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} x_i \end{aligned} \tag{18}$$

where the state variables $x_i = [v_i \; e_{p_i} \; e_{v_i} \; a_i]^T$, $i \in \{2, \ldots, N_S\}$ are the follower vehicle's velocity, position error, velocity error, and acceleration, respectively; the control input u_i is the desired acceleration, while the output y_i is the position error.

Note that each follower vehicle model (18) is coupled to its predecessor vehicle through the acceleration state variable a_{i-1}. This means that, between adjacent vehicles, the acceleration is unidirectionally transmitted using the communication network.

In this work, for a vehicle platooning application, the parameter N_C, which determines the frequency of the topology switching in Algorithm 1, is variable and depends on the acceleration of vehicles. If the acceleration of the vehicles is close to zero, i.e., the vehicles move with a constant velocity, then the stability condition η_i is rarely checked, and otherwise, it is checked more often.

Remark 3. *In Algorithm 1, when* $mod(k, N_C) == 0$*, all sub-systems will determine their own* N_{Ci} *and share it with all sub-systems to find the new* N_C *according to Algorithm 2.*

4. Results and Discussion

In this section, the simulation results for the proposed C-DMPC strategy obtained for a vehicle platooning application are presented. The multi-agent system is composed of $N_S = 4$ identical vehicles in a predecessor–following platoon formation, consisting of a leader vehicle (labeled in the figures with L) and three follower vehicles (labeled in the figures with F1, F2, and F3, respectively). For all vehicles, the time constant is $\tau = 0.1$, and for the follower vehicles, the time headway is $h = 0.7$.

The discrete-time platoon model is obtained by conversion using the Matlab function `c2dm` with the `zoh` method, starting from the continuous-time models (12) and (18), for the leader and follower vehicles, respectively. The sampling period used is $T_s = 0.1$ s.

The optimization parameters are as follows: the prediction horizon $N = 50$ samples, the weight matrices $P_i = Q_i = [0.15, 0, 0, 0; 0, 0.15, 0, 0; 0, 0, 0.1, 0; 0, 0, 0, 0]$, $i = \{2, \ldots, 4\}$, $P_1 = Q_1 = [0.2, 0, 0; 0, 0.1, 0; 0, 0, 0]$, and $R_i = 0.1$. For the stability condition, the parameter is $\alpha_i = 0.95$, $i = \{1, \ldots, 4\}$. Notice that the numerical values for the optimization parameters were chosen empirically by performing multiple simulation experiments, on a trial-and-error basis, to ensure the best platoon performance. The main focus was to ensure the feasibility of all the optimization problems using a given reference trajectory scenario. In both Problems 1 and 2, the cost functions J_i, $i = \{1, \ldots, 4\}$ and J_C, respectively, have three terms: (i) the terminal cost, which penalizes the states at the end of the prediction horizon with the weight matrix P_i, the stage cost, which penalizes the states predictions over the prediction horizon with the weight matrix Q_i, and the input cost, which penalizes the control effort value with the weight value R_i. Within the optimization problem solved by each sub-system, a trade-off between all these three terms is performed. Moreover, the optimization parameters for both Problems 1 and 2 are identical for all follower vehicles in the platoon $i \in \{2, \ldots, 4\}$, with a slightly larger weight given to states v_i and e_{p_i} when compared to the weight given to state e_{v_i}. For the leader vehicle, the state v_1 is given a larger weight than the state e_{p_i}, since, for the leader, it is more important to maintain the desired velocity than to minimize the position error.

Please note that since the follower vehicles are dynamically coupled through the acceleration states (see model (18)), it results that the cost functions J_i, $i \in \{2, \ldots, 4\}$, which are locally minimized, are implicitly coupled, i.e., to compute the local state prediction for vehicle i, the state value from the preceding vehicle $i-1$ is required.

In Algorithm 2, the value for the parameter is $\epsilon = 0.01$. Moreover, the values available for the N_C parameter used to determine the switching frequency between topologies are $N_{CMIN} = 3$ and $N_{CMAX} = 10$. The inequality input constraints are bounded by the limit values $u_i^{\min} = -2\ \mathrm{m/s^2}$ and $u_i^{\max} = 2\ \mathrm{m/s^2}$. The inequality output constraints are bounded by the limit values $y_i^{\min} = -1$ m and $y_i^{\max} = 1$ m.

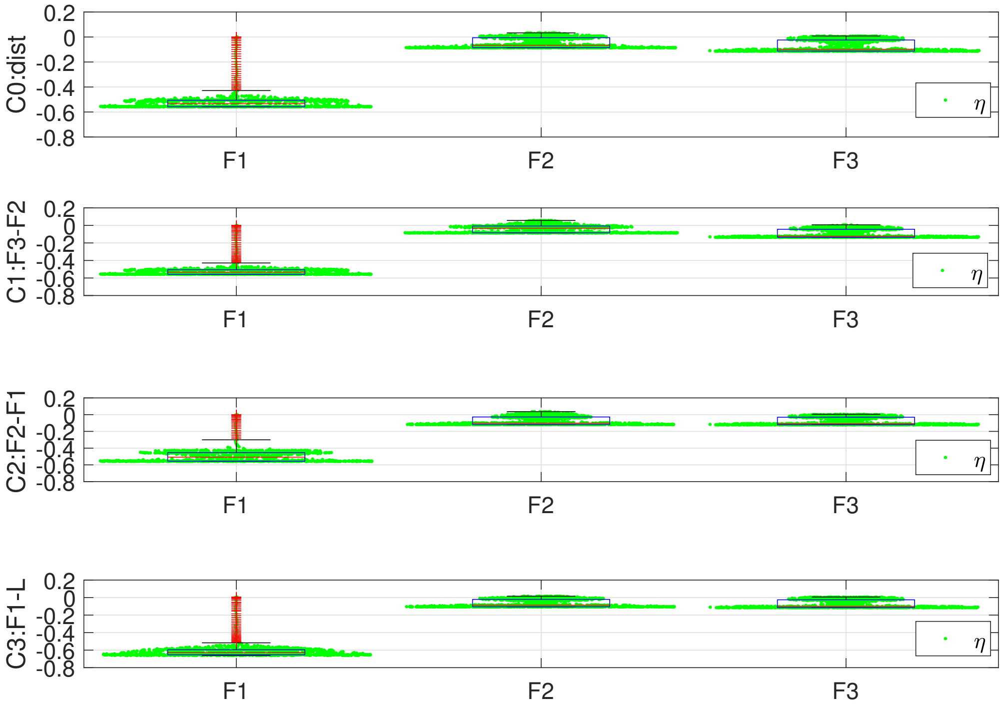
Remark 4. *All the optimization problems were solved using the YALMIP toolbox for Matlab [35], using the optimization solver* `quadprog`*. Within an optimization problem, the iterations ended when the relative dual feasibility stopping condition was less than the optimality tolerance threshold imposed at the value* 1.0×10^{-8}*. Regardless of the communication topology, the optimal solution was found in no more than nine iterations (i.e., C0 required five or six iterations, whereas C7 required seven or nine iterations), in an average computation time of 0.05 s, which is less than the sampling period.*

Since we aim to test a C-DMPC algorithm with switching topologies, let us denote each communication topology with respect to the coalition activated between the platoon vehicles, as follows:

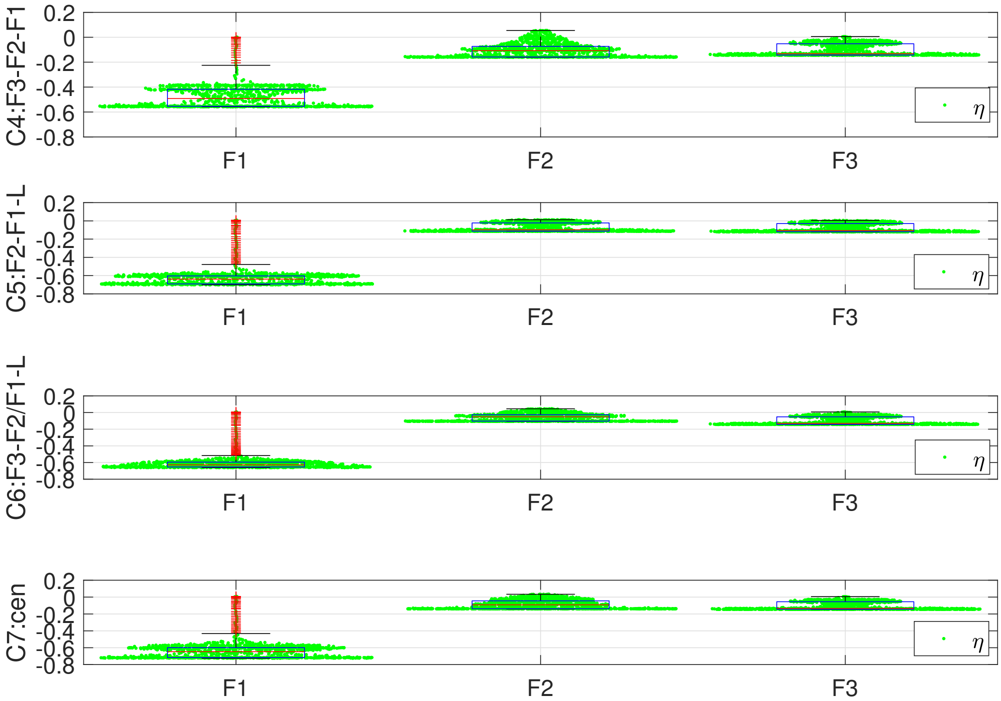
- C0: dist is the default platoon setting, in which each vehicle individually solves a DMPC optimization problem and no coalitions are formed;
- C1: F3-F2 is the coalition between vehicles F3 and F2, whereas vehicles F1 and L are outside the coalition;
- C2: F2-F1 is the coalition between vehicles F2 and F1, whereas vehicles F3 and L are outside the coalition;
- C3: F1-L is the coalition between vehicles F1 and L, whereas vehicles F2 and F3 are outside the coalition;
- C4: F3-F2-F1 is the coalition between vehicles F3, F2, and F1, whereas vehicle L is outside the coalition;
- C5: F2-F1-L is the coalition between vehicles F2, F1, and L, whereas vehicle F3 is outside the coalition;
- C6: F3-F2/F1-L is the simultaneous activation of two separate coalitions, i.e., the coalition between vehicles F3 and F2 and the coalition between vehicles F1 and L;
- C7: cen is centralized coalition, between all platoon vehicles L, F1, F2, and F3.

As previously mentioned in Algorithm 1, the coalitions are formed when the string stability condition is not fulfilled, i.e., the index value η_i is positive. To have an overall performance view, each topology was tested separately in a reference tracking scenario. To this end, a velocity profile for the leader was defined, consisting of segments of linearly spaced velocity values (in ascending or descending order), alternating with segments of constant velocity values.

In Figure 1, the values for the stability condition index η obtained during the simulations are provided. In Figure 1a, the values for coalitions C0, C1, C2, and C3 are given, while Figure 1b depicts the η values for coalitions C5, C6, C7, and C8.



(**a**)



(**b**)

Figure 1. Stability condition index. (**a**) Stability condition index for coalitions C0–C3. (**b**) Stability condition index for coalitions C4–C7.

To have an overall statistic view for each platoon vehicle, the simulation data were visualized in a box plot format using the Matlab function `boxplot`. The bottom and top of each box, which is depicted in black color are the 25th and 75th percentiles of the sample data, and the median of the sample data is depicted with a red line in the middle of the box. The whiskers of each box are the lines extended above and below each box and denote the maximum and the minimum values of the sample data. Data placed beyond the whiskers are considered outliers and are marked with a red cross sign. Each box plot was overlapped with a swarm chart for the numerical values of the data using the Matlab function `swarmchart`, i.e., each vector value is depicted with a green dot marker.

For a better understanding, for each follower vehicle, the mean value of the η index is computed (denoted with η_1, η_2, and η_3, for vehicles F1, F2, and F3, respectively), together with the overall mean value for each topology (denoted with η). For computations, the Matlab function `mean` was used. The results are given in Table 1 and show that the coalitions between adjacent vehicles improve the stability of the platoon, with respect to the default case, i.e., coalition C0, which can be viewed as each vehicle working independently in a distributed manner. See, for example, the numerical values for C1, in which a coalition between vehicles F3 and F2 is formed. Since F3 initialized the coalition, the corresponding η_3 value is improved (since $-0.0925 < -0.0711$). A smaller η index can be viewed as a more stable vehicle because it is further from the stability limit, which is the zero value. As expected, the centralized coalition C7 has the smallest overall η index (is the most stable), whereas the default coalition C0 has the largest value.

Another evaluation criterion for the performance obtained by each topology is a cumulative cost index $J = \sum_{i \in S} J_i$, with $J_i = \sum_{k=0}^{N_{sim}} \left(x_i(k)^T Q_i x_i(k) + u_i(k)^T R_i u_i(k) \right)$, where N_{sim} is the time length of the simulation. The weight matrices Q_i and R_i are the same as those from Problem 1. The values for the cumulative cost index J are also provided in Table 1, last column.

Motivated by these findings, we tested Algorithm 1 in three tests, obtained using the same velocity profile as before (Test 1), and two additional profiles (for Test 2 and Test 3), as depicted in Figure 2.

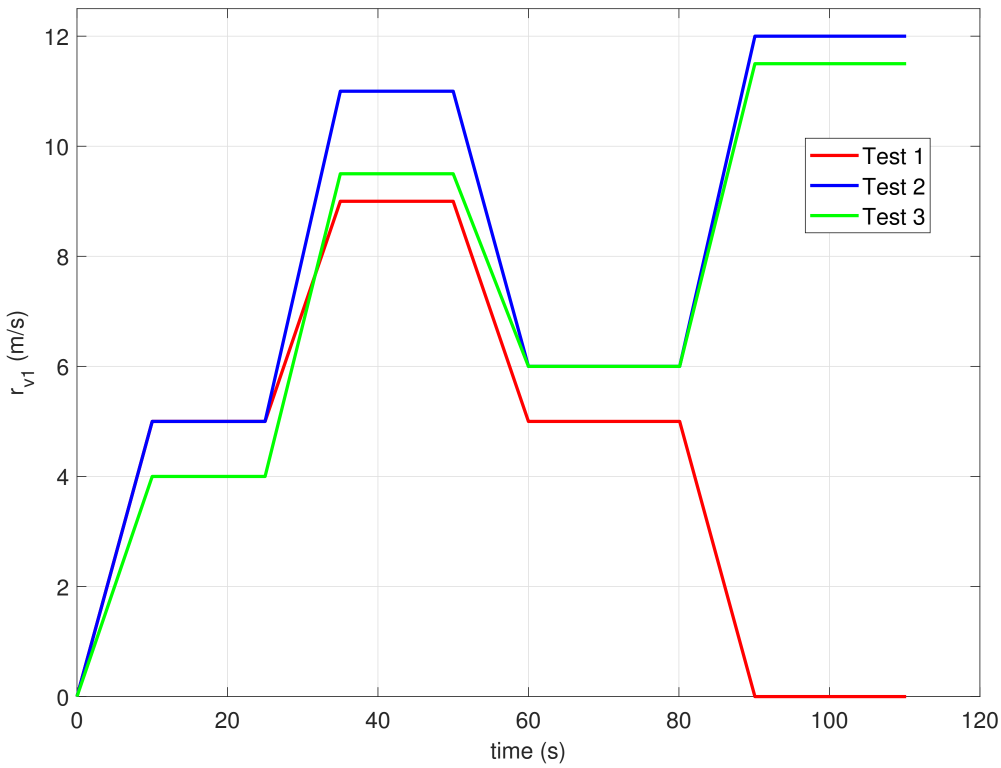


Figure 2. Velocity reference profiles used in Test 1, Test 2, and Test 3.

These tests were used to evaluate the switching mechanism between the available topologies, i.e., which coalitions are selected, and what is the frequency of the selection.

The results obtained in simulation for Test 1, Test 2, and Test 3 are provided in Figures 3–5, respectively. Each figure includes two sub-figures, with indices (a) and (b). Figures 3a–5a show the states and control inputs trajectories obtained in Test 1, Test 2, and Test 3, respectively. In Figures 3b–5b, the stability condition index η obtained for Test 1, Test 2, and Test 3, respectively, are depicted in the upper subplot, while the selected values for the parameter N_C are given in the middle subplot, and the selected coalitions are shown in the lower subplot. Note that when a coalition is formed, the communication topology between vehicles also changes, i.e., new communication links with bidirectional flow are open to ensure full information exchange inside a coalition.

Table 1. Mean value for the stability index η and the cumulative cost performance index J for each coalition.

Topology	η_1	η_2	η_3	η	J
C0	−0.5219	−0.0478	−0.0711	−0.2136	101.6485
C1	−0.5219	−0.0397	−0.0925	−0.2180	101.7256
C2	−0.4948	−0.0739	−0.0787	−0.2158	102.0910
C3	−0.6113	−0.0659	−0.0727	−0.2500	109.8897
C4	−0.4764	−0.1046	−0.1006	−0.2272	102.5453
C5	−0.6313	−0.0713	−0.0771	−0.2599	113.4700
C6	−0.6113	−0.0572	−0.0997	−0.2561	109.9782
C7	−0.6395	−0.0860	−0.1026	−0.2760	115.5049

The simulation results show that, from the eight available coalitions (i.e., C0–C7), in all our tests, only coalitions C4, C2, and C0 are selected by the switching mechanism. However, the switching frequency and the moment of selection between coalitions are different in each test (see, for example, time moment 60 s, when, for Test 1, coalition C0 is active, for Test 2, coalition C2 is active, whereas for Test 3, coalition C0 is active). This is marked in Figure 3b, Figure 4b, or Figure 5b, (3rd subplot) by a bar plot, with the y-axis labeled with the coalition indices.

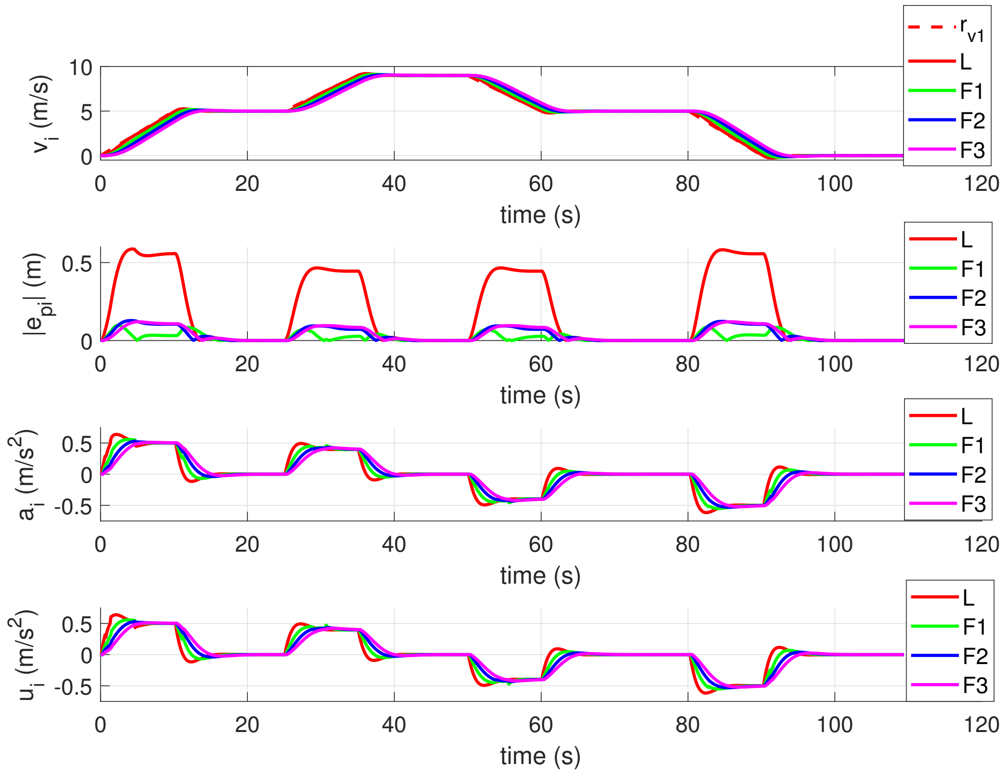
As previously mentioned in Algorithm 2, the frequency of the coalition switching (i.e., the value for the parameter N_C) is related to the acceleration values registered by the platoon vehicles. This is also visible for Test 1 when correlating the acceleration trajectories given in Figure 3a (3rd subplot) and the N_C values shown in Figure 3b (2nd subplot). See, for example, time moment 20 s, when the accelerations are zero, which corresponds to $N_C = 10$. This means that all platoon vehicles agree that there is no need to check the stability more frequently. At this moment, since the platoon is in the steady state regime, there is no need for coalitions (i.e., the default coalition C0 is active).

For all tests, a similar evaluation was performed by computing the mean value for the stability condition index η and the cumulative performance cost J. The numerical values are given in Table 2.

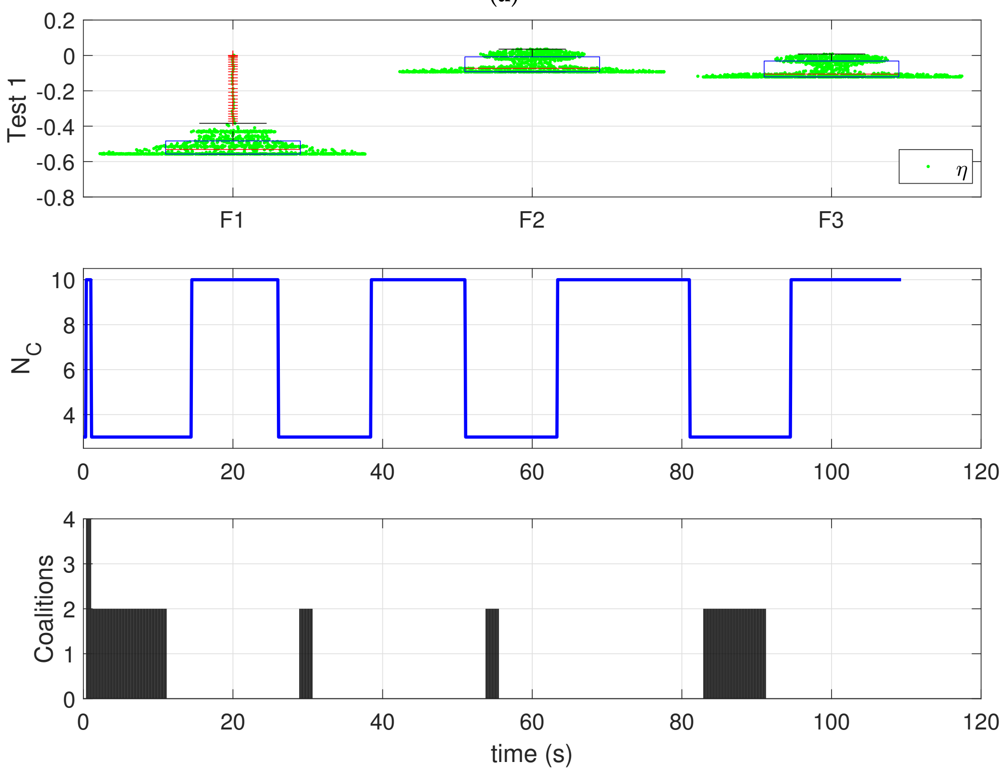
Table 2. Mean value for the stability index η and the cumulative cost performance index J for each test.

Test	η_1	η_2	η_3	η	J
Test 1	−0.5085	−0.0525	−0.0794	−0.2135	102.0183
Test 2	−0.5673	−0.0458	−0.0849	−0.2327	152.2686
Test 3	−0.5139	−0.0438	−0.0787	−0.2121	110.6744

When analyzing the numerical values given in Table 2, it results that Test 2 is 'more stable' since it has the smallest mean value for the index η, from the stability condition index perspective, but it has the largest performance index J. When comparing the values provided in Table 1, obtained for each coalition (while tested independently), and the values computed for Test 1, given in Table 2, it shows that the performance achieved in Test 1 is almost the mean value between the η values obtained for coalitions C0, C2, and C4 (when tested separately). This result is not surprising since, in Test 1, the topology active at each sampling period is freely chosen by the switching mechanism to achieve the best outcome for the entire platoon.

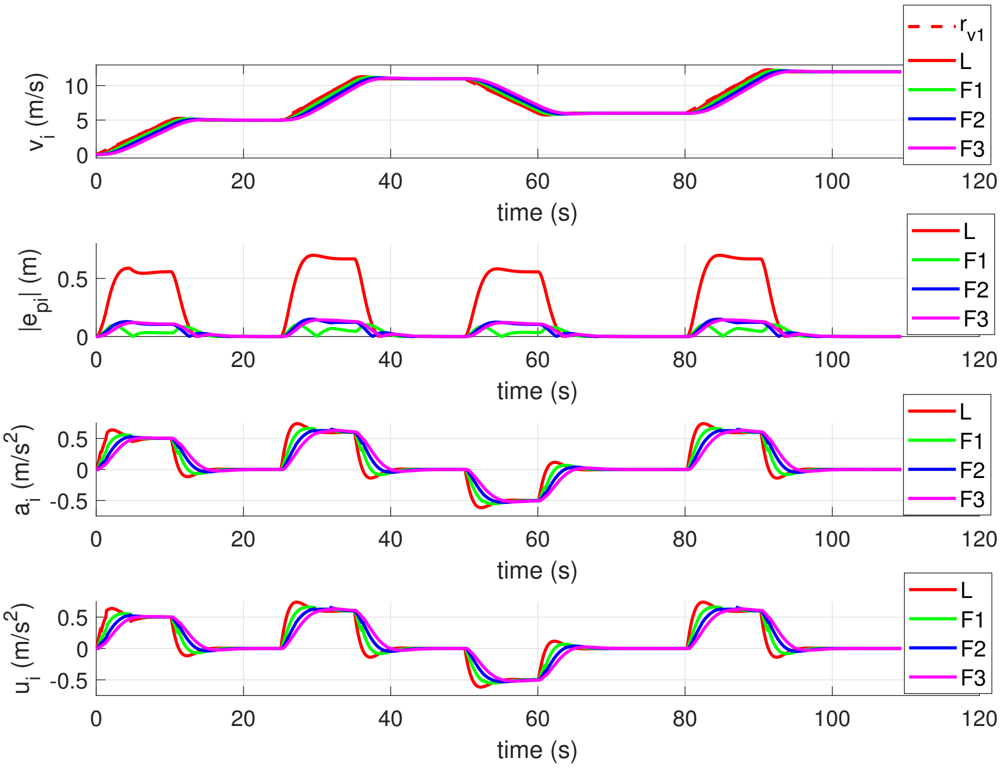


(**a**)

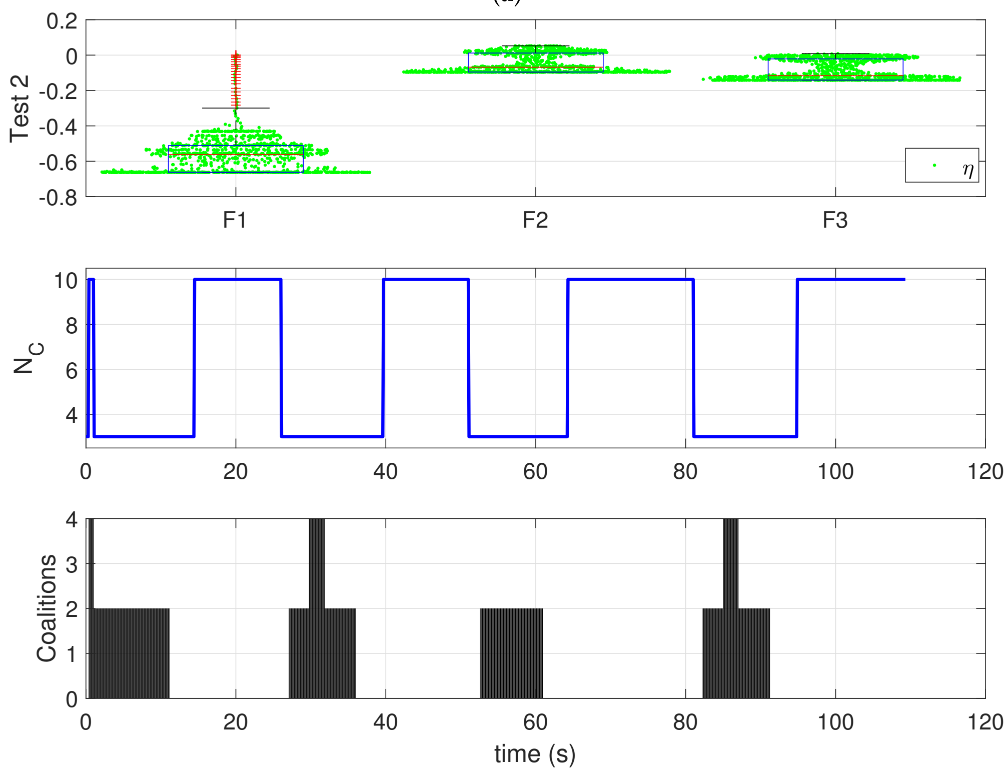


(**b**)

Figure 3. Simulation results for the proposed C-DMPC strategy obtained in Test 1. (**a**) Test 1—velocity trajectories (1st subplot), position error trajectories in absolute value (2nd subplot), acceleration trajectories (3rd subplot), and control input trajectories (4th subplot). (**b**) Test 1—stability condition index (1st subplot), parameter N_C values (2nd subplot), and coalition selection (3rd subplot).

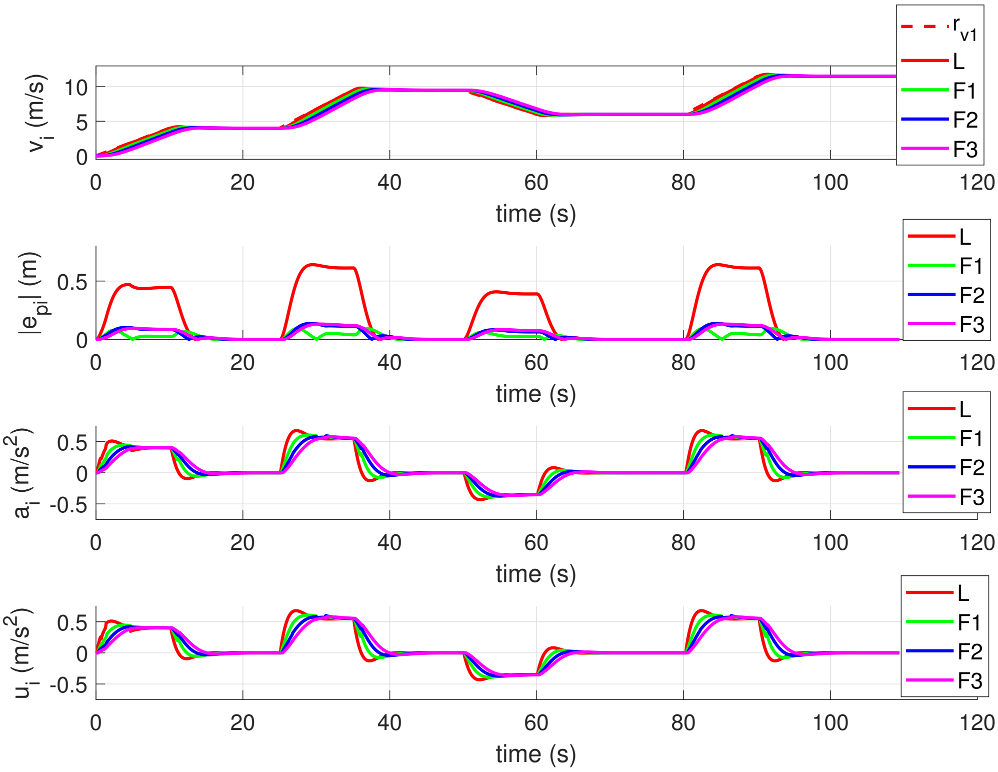


(**a**)

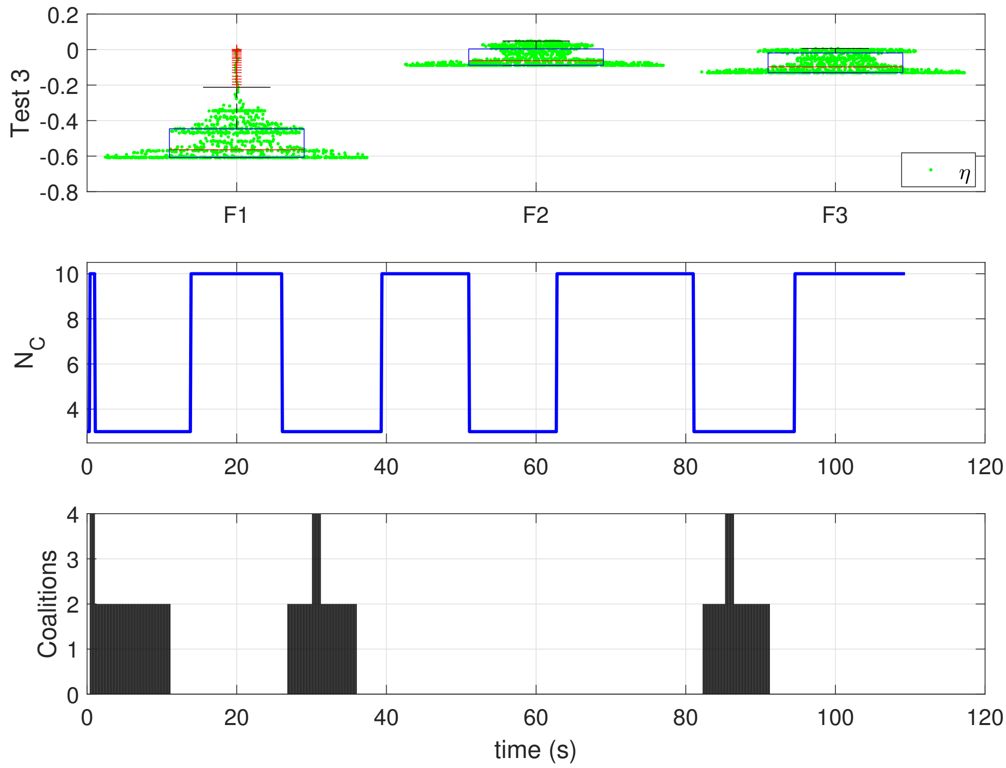


(**b**)

Figure 4. Simulation results for the proposed C-DMPC strategy obtained in Test 2. (**a**) Test 2—velocity trajectories (1st subplot), position error trajectories in absolute value (2nd subplot), acceleration trajectories (3rd subplot), and control input trajectories (4th subplot). (**b**) Test 2—stability condition index (1st subplot), parameter N_C values (2nd subplot), and coalition selection (3rd subplot).



(**a**)



(**b**)

Figure 5. Simulation results for the proposed C-DMPC strategy obtained in Test 3. (**a**) Test 3—velocity trajectories (1st subplot), position error trajectories in absolute value (2nd subplot), acceleration trajectories (3rd subplot), and control input trajectories (4th subplot). (**b**) Test 3—stability condition index (1st subplot), parameter N_C values (2nd subplot), and coalition selection (3rd subplot).

Remark 5. *Although outside the current scope of this work, in the future, a rigorous mathematical stability analysis will be derived, starting from the works in [32,36]. In the latter, the string stability of a vehicle platoon is analyzed using a $\mathcal{L}_2$ norm with respect to the acceleration signals between adjacent vehicles.*

5. Conclusions

In this paper, a coalitional control strategy developed in the DMPC framework, with automatic switching between possible communication topologies, was proposed. The proposed control methodology was tested in simulation using a vehicle platooning application. The performance analysis was performed for both the individual communication topologies and in three separate tests. Each test was designed to evaluate the switching mechanism between the topologies, which periodically selects the best communication topology for the entire platoon. The simulation results show that when given the liberty, the algorithm chooses the best outcome.

Future work will focus on testing the proposed algorithm in a more realistic scenario.

6. Materials and Methods

The simulations from this work were performed using MATLAB R2021a on Windows 10 Pro, 64-bit Operating System with a laptop 12th Gen Intel Core i5-1235U CPU @ 1.30 GHz and 8 GB RAM.

Author Contributions: Conceptualization, A.M., O.P. and C.F.C.; methodology, A.M. and O.P.; software, O.P.; validation, A.M. and O.P.; formal analysis, A.M.; investigation, O.P.; resources, A.M. and O.P.; data curation, O.P.; writing—original draft preparation, A.M. and O.P.; writing—review and editing, A.M., O.P. and C.F.C.; visualization, A.M.; supervision, C.F.C.; project administration, A.M. and C.F.C.; funding acquisition, A.M. 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 the "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.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The work of A.M. was supported by by "Gheorghe Asachi" Technical University of Iasi, through the Project "Performance and excellence in postdoctoral research 2022".

Conflicts of Interest: The authors declare no conflicts 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:

MASs	Multi-agent systems
MPC	Model predictive control
RMPC	Robust model predictive control
DMPC	Distributed model predictive control
C-DMPC	Coalitional distributed model predictive control
V2V	Vehicle-to-vehicle

References

1. Shoham, Y.; Leyton-Brown, K. *Multiagent systems Algorithmic, Game-Theoretic, and Logical Foundations*; Cambridge University Press: Cambridge, UK, 2008.
2. Azuma, S.; Yoshimura, R.; Sugie, T. Broadcast control of multi-agent systems. *Automatica* **2013**, *49*, 2307–2316. [CrossRef]
3. Li, D.; Ge, S.S.; He, W.; Ma, G.; Xie, L. Multilayer formation control of multi-agent systems. *Automatica* **2019**, *109*, 108558. [CrossRef]

4. Zhou, L.; Zheng, Y.; Zhao, Q.; Xiao, F.; Zhang, Y. Game-based coordination control of multi-agent systems. *Syst. Control Lett.* **2022**, *169*, 105376. [CrossRef]
5. Chen, J.; Shi, Z.; Zhong, Y. Robust formation control for uncertain multi-agent systems. *J. Frankl. Inst.* **2019**, *356*, 8237–8254. [CrossRef]
6. Haghshenas, H.; Badamchizadeh, M.A.; Baradarannia, M. Containment control of heterogeneous linear multi-agent systems. *Automatica* **2015**, *54*, 210–216. [CrossRef]
7. Shi, P.; Yan, B. A Survey on Intelligent Control for Multiagent Systems. *IEEE Trans. Syst. Man, Cybern. Syst.* **2021**, *51*, 161–175. [CrossRef]
8. Zhang, Q.; Wang, J.; Yang, Z.; Chen, Z. High gain feedback robust control for flocking of multi-agents system. *Trans. Inst. Meas. Control* **2019**, *41*, 3769–3776. [CrossRef]
9. Talebi, S.P.; Werner, S. Distributed Kalman Filtering and Control Through Embedded Average Consensus Information Fusion. *IEEE Trans. Autom. Control* **2019**, *64*, 4396–4403. [CrossRef]
10. Wang, L.; Li, J.; Liu, X.; Fang, Y.; Peng, C.; Sun, J. Event-Triggered Fault-tolerant Model Predictive Control of Nonlinear Multi-agent System with Time Delay and Parameter Uncertainty. In Proceedings of the 40th Chinese Control Conference, Shanghai, China, 26–28 July 2021; pp. 5350–5355.
11. Ding, B.; Ge, L.; Pan, H.; Wang, P. Distributed MPC for Tracking and Formation of Homogeneous Multi-agent System with Time-Varying Communication Topology. *Asian J. Control* **2016**, *18*, 1030–1041. [CrossRef]
12. Wang, Q.; Duan, Z.; Lv, Y.; Wang, Q.; Chen, G. Linear quadratic optimal consensus of discrete-time multi-agent systems with optimal steady state: A distributed model predictive control approach. *Automatica* **2021**, *127*, 109505. [CrossRef]
13. 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*, 1273–1282. [CrossRef]
14. Pi, D.; Xue, P.; Xie, B.; Wang, H.; Tang, X.; Hu, X. A Platoon Control Method Based on DMPC for Connected Energy-Saving Electric Vehicles. *IEEE Trans. Transp. Electrif.* **2022**, *8*, 3219–3235. [CrossRef]
15. Wang, X.; Hu, M.; Bian, Y.; Wang, X.; Qin, H.; Ding, R. DMPC-based string stable platoon control with robustness against communication delays. *Veh. Commun.* **2023**, *43*, 100655. [CrossRef]
16. Peng, J.; Li, J.; Wang, K.; Xiao, S.; Li, C. Prescribed performance control of nonlinear multi-agent systems under switching topologies. *Syst. Control Lett.* **2023**, *180*, 105609. [CrossRef]
17. Chang, Z.; Zhang, H.T.; Fan, M.C.; Chen, G. Distributed Consensus of Multi-Agent Systems with Input Constraints: A Model Predictive Control Approach. *IEEE Trans. Circuits Syst.-I Regul. Pap.* **2015**, *62*, 825–834. [CrossRef]
18. Thunberg, J.; Song, W.; Montijano, E.; Hong, Y.; Hu, X. Distributed attitude synchronization control of multi-agent systems with switching topologies. *Automatica* **2014**, *50*, 832–840. [CrossRef]
19. Gallehdari, Z.; Meskin, N.; Khorasani, K. Distributed reconfigurable control strategies for switching topology networked multi-agent systems. *ISA Trans.* **2017**, *71*, 51–67. [CrossRef] [PubMed]
20. Li, Y.; Chen, B.; Zhao, H.; Peeta, S.; Hu, S.; Wang, Y.; Zheng, Z. A Car-Following Model for Connected and Automated Vehicles With Heterogeneous Time Delays Under Fixed and Switching Communication Topologies. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 14846–14858. [CrossRef]
21. Sánchez-Amores, A.; Martinez-Piazuelo, J.; Maestre, J.M.; Ocampo-Martinez, C.; Camacho, E.F.; Quijano, N. Coalitional model predictive control of parabolic-trough solar collector fields with population-dynamics assistance. *Appl. Energy* **2023**, *334*, 120740. [CrossRef]
22. Baldivieso-Monasterios, P.; Trodden, P. Coalitional predictive control: Consensus-based coalition forming with robust regulation. *Automatica* **2021**, *125*, 109380. [CrossRef]
23. Chanfreut, P.; Maestre, J.M.; Ferramosca, A.; Muros, F.J.; Camacho, E.F. A Distributed Model Predictive Control for Tracking: A Coalitional Clustering Approach. *IEEE Trans. Autom. Control* **2022**, *67*, 6873–6879. [CrossRef]
24. Masero, E.; Baldivieso-Monasterios, P.R.; Maestre, J.M.; Trodden, P.A. Robust coalitional model predictive control with plug-and-play capabilities. *Automatica* **2023**, *153*, 111053. [CrossRef]
25. Masero, E.; Maestre, J.M.; Francisco, M.; Camacho, E.F. Robust Coalitional Model Predictive Control with Predicted Topology Transitions. *IEEE Trans. Control Netw. Syst.* **2021**, *8*, 1869–1880. [CrossRef]
26. Maxim, A.; Pauca, O.; Amariei, R.G.; Braescu, F.C.; Caruntu, C.F. Coalitional Control Strategy for a Heterogeneous Platoon Application. *Mathematics* **2024**, *12*, 7. [CrossRef]
27. Maxim, A.; Caruntu, C.F. A Coalitional Distributed Model Predictive Control Perspective for a Cyber-Physical Multi-Agent Application. *Sensors* **2021**, *21*, 4041. [CrossRef]
28. Maxim, A.; Caruntu, C.F. Coalitional Distributed Model Predictive Control Strategy for Vehicle Platooning Applications. *Sensors* **2022**, *22*, 997. [CrossRef] [PubMed]
29. Maxim, A.; Pauca, O.; Caruntu, C.F. Assessment of Control Efficiency in a Vehicle Platooning Application. In Proceedings of the European Control Conference, Bucharest, Romania, 13–16 June 2023; pp. 1270–1275.
30. Hou, B.; Li, S.; Zheng, Y. Distributed Model Predictive Control for Reconfigurable Systems With Network Connection. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 907–918. [CrossRef]
31. Maestre, J.M.; Muñoz de la Peña, D.; Jiménez Losada, A.; Algaba, E.; Camacho, E. A coalitional control scheme with applications to cooperative game theory. *Optim. Control Appl. Methods* **2014**, *35*, 592–608. [CrossRef]

32. Kianfar, R.; Falcone, P.; Fredriksson, J. A Distributed Model Predictive Control Approach to Active Steering Control of String Stable Cooperative Vehicle Platoon. *IFAC Proc. Vol.* **2013**, *46*, 750–755. [CrossRef]
33. Dunbar, W.B.; Caveney, D.S. Distributed receding horizon control of vehicle platoons: Stability and string stability. *IEEE Trans. Autom. Control* **2012**, *57*, 620–633. [CrossRef]
34. Zhu, Y.; He, H.; Zhao, D. LMI-Based Synthesis of String-Stable Controller for Cooperative Adaptive Cruise Control. *IEEE Trans. Intell. Transp. Syst.* **2021**, *21*, 4516–4525. [CrossRef]
35. 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.
36. Kianfar, R.; Falcone, P.; Fredriksson, J. A control matching model predictive control approach to string stable vehicle platooning. *Control Eng. Pract.* **2015**, *45*, 163–173. [CrossRef]