Delft University of Technology

Delft Center for Systems and Control

Technical report 15-040

Time-varying scheme for noncentralized model predictive control of large-scale systems*

A. Núñez, C. Ocampo-Martinez, J.M. Maestre, and B. De Schutter

If you want to cite this report, please use the following reference instead:
A. Núñez, C. Ocampo-Martinez, J.M. Maestre, and B. De Schutter, "Time-varying scheme for noncentralized model predictive control of large-scale systems," *Mathematical Problems in Engineering*, vol. 2015, 17 pp., 2015. Article ID 560702. doi:10.1155/2015/560702

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

Time-Varying Scheme for NonCentralized Model Predictive Control of Large-Scale Systems

A. Núñez[a], C. Ocampo-Martinez[b,*], J.M. Maestre[c], B. De Schutter[d]

[a]*Section of Railway Engineering, Delft University of Technology*
Stevinweg 1, 2628 CN, Delft, The Netherlands
[b]*Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Universitat Politècnica de Catalunya (UPC)*
Carrer Llorens i Artigas, 4-6, 08028 Barcelona, Spain
[c]*Departamento de Ingeniería de Sistemas y Automática, University of Seville*
Camino de los Descubrimientos s/n, 41092 Seville, Spain
[d]*Delft Center for Systems and Control, Delft University of Technology*
Mekelweg 2, 2628 CD, Delft, The Netherlands

Abstract

The Non-Centralized Model Predictive Control (NC-MPC) framework in this paper refers to any distributed, hierarchical, or decentralized model predictive controller (or a combination of them) the structure of which can change over time and the control actions of which are not obtained based on a centralized computation. Within this framework, we propose suitable on-line methods to decide which information is shared and how this information is used between the different local predictive controllers operating in a decentralized, distributed, and/or hierarchical way. Evaluating all the possible structures of the NC-MPC controller leads to a combinatorial optimization problem. Therefore, we also propose heuristic reduction methods, to keep tractable the number of NC-MPC problems to be solved. To show the benefits of the proposed framework, a case study of a set of coupled water tanks is presented.

Keywords: Model predictive control, large-scale systems, non-centralized control, time-varying control topologies

1. INTRODUCTION

During the last decades, there has been a notable increment in the size of the problems dealt by control engineers. Large-scale applications such as irrigation canals [1], transportation networks [2], urban water systems [3], or supply chains [4], among many others, are now within the scope of control theory due to the proliferation of non-centralized control techniques (see, e.g., the surveys [5, 6]). The basic idea behind these control schemes is the well-known *divide and conquer* principle. In this way, the control problem of a large-scale monolithic system is partitioned into several smaller control problems that are assigned to a set of local controllers or *agents*. A similar approach can be used to deal with the overall control problem that results from the interaction of several coupled independent dynamical systems that pursue different goals.

In the literature, most non-centralized schemes focus on the following scenarios: 1) the overall system is partitioned in such a way that the coupling between subsystems is weak and can be ignored, i.e., the agents work in a decentralized fashion, and 2) the coupling between the different subsystems demands coordination between the local controllers and, for this reason, a communication mechanism between the agents has to be provided. In the latter scenario, we say that the agents work in a distributed or in a hierarchical fashion. In general, distributed control schemes outperform the decentralized ones but at the price of a higher complexity from both a communication burden viewpoint and an algorithmic viewpoint. More recently, the evolution of the field has led to the development of control schemes in which the local controllers adopt a decentralized attitude when the coupling between the control tasks is low and a distributed approach when it is high. In other words, the coordination and communication structure are adapted to the coupling between the control tasks. As a result of this, the local controllers are separated dynamically into cooperative groups or *coalitions*. For example, in [7], the set of active constraints is used to modify the sets of cooperating agents; in [8, 9], the coupling structure of the plant is exploited to divide it into hierarchically coupled clusters; in [10, 11], the coalitional model predictive control (MPC) framework is used, where only the couplings with an important contribution to the overall system performance are considered. Finally, the aggregation of control nodes and the inclusion of constraints regarding the division of the benefits and costs derived from the cooperation is studied in [12].

In this work, we focus on a novel type of control schemes with time-varying communication topology, which presents several open research issues. In the first place, it is clear that in a large-scale application the control scheme cannot switch between all the possible network topologies [13, 14]. In fact, the problems derived from the resulting combinatorial explosion in

*Corresponding author. Tel: +34 93 401 5752; Fax: +34 93 401 5750.
Email address: cocampo@iri.upc.edu (C. Ocampo-Martinez)

this context are pointed out in several of the aforementioned works, e.g., [10, 11]. How to decide on the most appropriate topology at a given time step is a difficult problem similar to that of system partitioning, for which there are relatively few results available in the literature (see, e.g., [15, 16, 17, 18] and the references therein).

Another open issue is the optimal way to define hierarchies between local controllers [19]. Most distributed control schemes are simply based on peer-to-peer coordination, but there are also other alternatives: e.g., there are schemes that implement a master-slave hierarchy in which the agents have to wait for their turn before calculating and implementing their control actions [5]. How to determine dynamically the best hierarchical relationships between the controllers is another open problem.

This work proposes a non-centralized MPC (NC-MPC) framework in which the overall system partition and the hierarchy relationship between the corresponding subsystems vary dynamically over time. The task of the NC-MPC controller is to identify the relevant regions (partitioning) and to assign to them more importance by changing the control structure. To achieve this, the amount of information exchanged between the controllers can be increased or the hierarchical level of those crucial regions/subsystems can be augmented. In particular, several possible control structures for the communication between subsystems are considered and the hierarchical control system implements the one that provides the best performance according to a set of given objectives. In this way, the control structure gains flexibility to increase its adaptability to the evolution of the system conditions and external variables. Specifically, in this paper we focus on large-scale systems in which there is a *flow* between or through the constitutive elements of the system. Water, traffic, electricity, logistic, and data networks are practical examples of this type of systems. In this context, flow is understood in the sense of movement of raw material/particles/matter related to the use or function of the system. For instance, in water networks, flow would correspond to the movement of water from point A to B; in transportation systems, it would correspond to the movement of cars/trains/bikes within the network; in data networks, it would be related to the data packets moving within a given network.

The remainder of the paper is organized as follows. In Section 2, the control-oriented framework and a proposed partitioning method are presented. Section 3 presents the non-centralized model predictive control (NC-MPC) framework. Section 4 details the proposed rules to define the changes in the structure of the NC-MPC controller. Section 5 presents numerical results using an interconnected water tank system benchmark. Finally, the main conclusions of the paper and relevant lines for future research are given in Section 6.

2. System Modeling

Given the complex nature of large-scale network systems (LSNS), from a control viewpoint it is preferable to work with control-oriented models [20, 21] that are accurate enough to capture the relevant dynamics but yet simple enough to reduce both complexity and computation burden [22].

2.1. Control-oriented Modeling Framework

In flow networks, an LSNS may be represented by a directed graph $G(\mathcal{V}, \mathcal{E})$, where nodes in $\mathcal{V}$ are compositional elements that characterize an attribute of the system [21]. This set is composed of n_x storage elements, n_u flow handling elements, n_d sinks, and n_q intersection nodes [20]. Likewise, the edge (a, b) in the set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ models that the element b is physically connected with the element a (so there are variables from b that have an influence over a).

Considering the volume as the state variable, the flow through handling elements as the controlled inputs, and flows to sinks as system disturbances, an LSNS may be generally described in a state-space form by the following linear discrete-time dynamic model:

$$x(k+1) = Ax(k) + B\,u(k) + F\,d(k), \tag{1a}$$

$$\mathbf{0} = E_u\,u(k) + E_d\,d(k), \tag{1b}$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, and $d \in \mathbb{R}^{n_d}$ correspond to the states vector, the controlled input vector, and measured disturbances vector, respectively. Moreover, $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, and $F \in \mathbb{R}^{n_x \times n_d}$ are state-space system matrices for balances in storage elements, and $E_u \in \mathbb{R}^{n_q \times n_u}$, $E_d \in \mathbb{R}^{n_q \times n_d}$ are matrices for static balances in nodes. Notice that there is no x term in (1b) since it is supposed that all storage element outflows are controlled. Besides, $\mathbf{0} \in \mathbb{R}^{n_q}$ is a zero vector. All vectors and matrices are dictated by the network topology. In general, states and control inputs are subject to constraints of the form

$$x(k) \in \mathcal{X}, \quad \forall k, \tag{2a}$$

$$u(k) \in \mathcal{U}, \quad \forall k, \tag{2b}$$

where $\mathcal{X} \subset \mathbb{R}^{n_x}$ and $\mathcal{U} \subset \mathbb{R}^{n_u}$ are the resulting hyperboxes of the corresponding element constraints.

2.2. Model Decomposition

Considering the control-oriented model (1), when a particular partitioning methodology is applied, the resulting subsystems may be connected by *topological relations* and/or *information relations*. The former are related to the nature of the variables that different subsystems may share: states and/or control inputs. The latter are related to the information that the controllers of the corresponding subsystems might exchange.

The overall system (1) is assumed to be decomposed in a set $\mathcal{N} = \{S_1, \ldots, S_M\}$ of non-overlapping subsystems, which are output-decentralized and input-coupled. The model of the i-th subsystem S_i, for $i \in \{1, \ldots, M\}$, is stated as follows[1]:

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + \psi_i(k) + F_i d_i(k) \tag{3a}$$

$$\mathbf{0} = E_{1,i} u_i(k) + E_{2,i} u_{\mathcal{H}_i,i}(k) + E_{3,i} u_{\mathcal{M}_i,i}(k) + E_{4,i} d_i(k), \tag{3b}$$

[1]Considering the partitioning approach in [15], we assume that constraints including the state of subsystems are not coupled. The only cross-influence between subsystems is given by the established shared input variables.

with

$$\psi_i(k) \triangleq B_{1,i}\, u_{\mathcal{H}_i,i}(k) + B_{2,i}\, u_{i,\mathcal{M}_i}(k), \tag{4}$$

where $x_i \in \mathbb{R}^{n_{x_i}}$ is the local state vector; $d_i \in \mathbb{R}^{n_{d_i}}$ is the local measurable disturbances vector; $u_i \in \mathbb{R}^{n_{u_i}}$ stands for the input vector that only affects the local dynamics; $u_{\mathcal{H}_i,i} \in \mathbb{R}^{|\mathcal{H}_i|}$ is the input vector decided by the i-th subsystem that affects both the local dynamics and the dynamics of the aggregated set $\mathcal{H}_i \subset \mathcal{N}$ of neighboring subsystems; and the set $\mathcal{M}_i \subset \mathcal{N}$ aggregates the neighboring subsystems whose inputs $u_{i,\mathcal{M}_i} \in \mathbb{R}^{|\mathcal{M}_i|}$ affect the i-th subsystem. The dimensions of the matrices in (3) and (4) are stated in Table 1.

Table 1: Dimension of matrices in (3) and (4)

Matrix	Dimension
A_i	$n_{x_i} \times n_{x_i}$
B_i	$n_{x_i} \times n_{u_i}$
F_i	$n_{x_i} \times n_{d_i}$
$E_{1,i}$	$n_{q_i} \times n_{u_i}$
$E_{2,i}$	$n_{q_i} \times \lvert\mathcal{H}_i\rvert$
$E_{3,i}$	$n_{q_i} \times \lvert\mathcal{M}_i\rvert$
$E_{4,i}$	$n_{q_i} \times n_{d_i}$
$B_{1,i}$	$n_{x_i} \times \lvert\mathcal{H}_i\rvert$
$B_{2,i}$	$n_{x_i} \times \lvert\mathcal{M}_i\rvert$

In the same way, constraints (2) are partitioned for each i-th subsystem as

$$x_i(k) \in \mathcal{X}_i, \quad \forall k, \tag{5a}$$

$$u_i(k) \in \mathcal{U}_i, \quad \forall k, \tag{5b}$$

where $\mathcal{X} = \bigtimes_{i=1}^{M} \mathcal{X}_i$ and $\mathcal{U} = \bigtimes_{i=1}^{M} \mathcal{U}_i$ (Cartesian product).

3. Non-centralized Model Predictive Control (NC-MPC)

From the LSNS model (1) at time step k, we consider the following sequences of states, controlled inputs, and disturbances over a fixed-time prediction horizon N_p:

$$\mathbf{x}(k) = [x^T(k+1|k), \ldots, x^T(k+N_\mathrm{p}|k)]^T, \tag{6a}$$

$$\mathbf{u}(k) = [u^T(k|k), \ldots, u^T(k+N_\mathrm{p}-1|k)]^T, \tag{6b}$$

$$\mathbf{d}(k) = [d^T(k|k), \ldots, d^T(k+N_\mathrm{p}-1|k)]^T, \tag{6c}$$

with $u(k+\ell) = u(k+N_\mathrm{u}-1)$, for $\ell = N_\mathrm{u}, ..., N_\mathrm{p}-1$, and N_u the control horizon. These sequences depend on the initial state vector $x(k) = x_k$. The sequence $\mathbf{d}(k)$ can be defined according to the case and the nature of the system disturbances. Hence, $\mathbf{d}(k)$ may be considered as a constant value over N_p or can be computed using a forecasting algorithm. Now we state the overall control problem:

Problem 3.1 (Centralized MPC). *Design an MPC controller that solves the open-loop optimization problem*

$$\min_{\mathbf{u}(k)} J(\mathbf{u}(k), x_k, \mathbf{d}(k)) \triangleq \sum_{m=1}^{|\mathcal{O}|} \gamma_m J_m(\mathbf{u}(k), x_k, \mathbf{d}(k)), \tag{7a}$$

subject to system model (1), *system constraints* (2) *over* N_p, *the initial condition* $x(k) = x_k$ *and a set of* n_c *operational constraints given by management policies of the system and collected in the expression*

$$G_1\mathbf{x}(k) + G_2\mathbf{u}(k) + G_3\mathbf{d}(k) \leq g, \tag{7b}$$

where $J(\cdot) : \mathbb{R}^{(n_\mathrm{u}+n_\mathrm{d})N_\mathrm{p}+n_\mathrm{x}} \rightarrow \mathbb{R}$ *in* (7a) *is the cost function collecting all control objectives with index set* $\mathcal{O}$ *and* γ_m *are positive scalar weights to prioritize the m-th control objective. Moreover,* $G_1 \in \mathbb{R}^{n_\mathrm{c}\times n_\mathrm{x}N_\mathrm{p}}$, $G_2 \in \mathbb{R}^{n_\mathrm{c}\times n_\mathrm{u}N_\mathrm{p}}$, $G_3 \in \mathbb{R}^{n_\mathrm{c}\times n_\mathrm{d}N_\mathrm{p}}$, *and* $g \in \mathbb{R}^{n_\mathrm{c}}$. *Assuming that the optimization problem* (7) *is feasible, then there is an optimal solution given by the sequence of control inputs* $\mathbf{u}^*(k)$ *and then the receding horizon procedure sets*

$$u_{\mathrm{MPC}}(x_k) \triangleq u^*(k|k), \tag{8}$$

and disregards the computed inputs from $k+1$ *to* $k+N_\mathrm{p}-1$, *with the whole process repeated at the next time step* $k+1$.

Expression (8) is known in the MPC literature as the *MPC law* [23]. Typically, the minimization in (7a) is implemented in a centralized way. For large-scale systems, centralized MPC may become impractical because of the large number of variables and large amounts of information exchange, which in turn might imply a huge computational burden. Therefore, NC-MPC schemes are proposed to deal with large-scale MPC problems given their capabilities to divide a complex problem into several less-complex sub-problems.

3.1. Non-Centralized Predictive Control Approach

To overcome the computational problems associated with the implementation of the centralized MPC schemes, NC-MPC arises to deal with large-scale systems [5, 24]. This strategy relies on designing less complex MPC controllers, in order to have a more tractable and less computationally demanding control structure. Features like sparsity of the state equations, distance between actuators, and communication issues are typically used to merge local states and inputs and to define the resulting subsystems. The way the original problem is decomposed determines the design of the local MPC controller $C_i \in \mathcal{C}$, with $i \in \{1, \ldots, M\}$, for the subsystem $\mathcal{S}_i$. The set $\mathcal{C}$ includes the local MPC controllers of all the LSNS subsystems. The drawback of NC-MPC with respect to a centralized MPC is the potential occurrence of suboptimalities arising from the way the system is decomposed and from the greater algorithmic complexity.

In (1) only input coupling is considered [15]. We also assume the possibility of defining local operational constraints; so the rules for the overall system (7b) can be decoupled without affecting the performance of the controller. We assume the cost function (7a) can be split such that each subsystem $\mathcal{S}_i$ considers the local cost function

$$J_i(\mathbf{u}_i(k), x_{k,i}, \mathbf{d}_i(k)) = \sum_{m=1}^{|\mathcal{O}|} \gamma_{m,i} J_{m,i}(\mathbf{u}_i(k), x_{k,i}, \mathbf{d}_i(k)), \tag{9}$$

and

$$\mathbf{x}_i(k) = [x_i^T(k+1|k), \ldots, x_i^T(k+N_\mathrm{p}|k)]^T, \tag{10a}$$

$$\mathbf{u}_i(k) = [u_i^T(k|k), \ldots, u_i^T(k+N_\mathrm{p}-1|k)]^T, \tag{10b}$$

$$\mathbf{d}_i(k) = [d_i^T(k|k), \ldots, d_i^T(k+N_\mathrm{p}-1|k)]^T. \tag{10c}$$

Notice that for the m-th objective, the weights $\gamma_{m,i}$ and $\gamma_{m,j}$ for subsystems S_i and S_j may be different, which implies different prioritization of control objectives, also to compensate for possible couplings through the objective function[2]. This fact would introduce some extra performance suboptimality in case a proper estimation of those couplings is not available. It is assumed that in case of availability of a communication channel, local MPC controllers can coordinate or cooperate with each other to calculate their best control sequences that increase the overall performance, considering the effects of other MPC controllers, and to decide their control actions with this information. From Problem 3.1, Problem 3.2 arises naturally.

Problem 3.2 (Non-Centralized MPC). *Design a local MPC controller C_i that solves the open-loop optimization problem*

$$\min_{\mathbf{u}_i(k)} J_i(\mathbf{u}_i(k), x_{k,i}, \mathbf{d}_i(k)) \triangleq \sum_{m=1}^{|O|} \gamma_{m,i} J_{m,i}(\mathbf{u}_i(k), x_{k,i}, \mathbf{d}_i(k)), \tag{11a}$$

subject to system local model (3)*, system local constraints* (5) *over N_p, initial condition $x_i(k) = x_{k,i}$, and a set of n_{c_i} operational constraints given by management policies of the system and collected in the form*

$$G_{1,i}\mathbf{x}_i(k) + G_{2,i}\mathbf{u}_i(k) + G_{3,i}\mathbf{d}_i(k) \leq g_i, \tag{11b}$$

with all matrices having suitable dimensions according to the length of the state, controlled input and disturbance vectors related to the subsystem S_i. Assuming that the optimization problem (11) *is feasible, then there is an optimal solution given by the sequence of control inputs $\mathbf{u}_i^*(k)$, and then the receding horizon procedure sets*

$$u_{\mathrm{MPC},i}(x_{k,i}) \triangleq u_i^*(k|k), \tag{12}$$

repeating the whole process at the next time step $k+1$.

The control input vector in (3) depends on the availability of the neighboring controllers to communicate their information. In particular, in this paper we consider three cases for the relationships between two local controllers: 1) C_i decides not to share the inputs with C_j at all, 2) C_i shares the control sequence decided in the previous time step $\mathbf{u}_{j,i}(k-1)$, and 3) C_i shares its current decision $\mathbf{u}_{j,i}(k)$. The option to communicate information (or not) will define a dynamic topology for the communications of the overall system. Next in Section 3.2, the possible relationships between controllers are described.

[2] Moreover, $\gamma_{m,i}$ and $\gamma_{m,j}$ could even be time-varying. In this work, for simplicity, we assume they are constant.

3.2. Relationships between controllers

The control input vector of the local model (3) is defined as

$$\tilde{u}_i(k) \triangleq \begin{bmatrix} u_i(k) \\ u_{\mathcal{H}_i,i}(k) \\ u_{i,\mathcal{M}_i}(k) \end{bmatrix}.$$

Note that not all these inputs are computed by controller C_i. In particular, $u_{j,i}$, $j \in \mathcal{H}_i$, are computed by C_i while $u_{i,j}$, $j \in \mathcal{M}_i$, are decided by the controller C_j. In general, $\mathbf{u}_{j,i}(k)$ and $\mathbf{u}_{i,j}(k)$ depend on the type and amount of information exchanged between controllers C_i and C_j. The following cases can be considered for $\mathbf{u}_{j,i}(k)$ (computed by C_i, affecting C_j):

- If C_i is at a higher level of the hierarchy than C_j, C_i will compute first $\mathbf{u}_{j,i}(k)$ and then it will share this value with C_j.
- If C_i is at the same level of the hierarchy than C_j, we have the following cases:
  - in a distributed MPC scheme, $\mathbf{u}_{j,i}(k)$ obtained by C_i will be jointly calculated with C_j. We will say in this case that subsystems S_i and S_j are working within a *coalition*.
  - in a decentralized fashion with information broadcast, the value of $\mathbf{u}_{j,i}(k-1)$ will be transmitted;
  - if there is no communication, a nominal value is used.
- If C_i is at a lower level of the hierarchy than C_j, we have the following cases for the controllers:
  - if there is communication, the value of $\mathbf{u}_{j,i}(k-1)$ will be known.
  - if there is no communication, a nominal value is used.

In the next section, integer variables $\delta_{j,i}(k)$ are used to capture the option of controllers to share information and to define the topology for the communication between controllers. For a given value of each $\delta_{j,i}(k)$, the subsystems will be organized in L levels of hierarchy, where there are P_q subsystems at each level, for $q \in \{1, \ldots, L\}$. Therefore, each subsystem in the q-th level is denoted as $S_{r,q}$, with $r \in \{1, \ldots, P_q\}$ and $\sum_q P_q = M$.

4. Switching Mechanism for Communication

In this section, the switching mechanism problem for communication between local controllers is described. The idea is to control the large-scale system by clustering dynamically the local MPC controllers. To this end, a supervisory controller decides how the information flows into the NC-MPC controller.

The optimization variable for the supervisory controller is the NC-MPC structure that the system will operate under. This means that we require that the controllers can adjust their operation based on the instructions from the supervisory controller about the structural configuration that they will have to follow. The communication between controllers and supervisory controller can be either fully centralized or it can include some degrees of decentralized decisions [25, 26, 27, 28]. In this paper, a hierarchical methodology is used, where the supervisory controller decides the best structure (NC-MPC topology given by the integer variables $\delta_{i,j}(k)$), while the local controllers will optimize their control and state sequences. This keeps the calculation as much non-centralized as possible, for each of the possible scenarios of NC-MPC structures suggested by the supervisory controller. The communication from each controller C_i to the supervisory controller includes the initial states $x_{k,i}$ and the set of control sequences $\mathbf{u}_i(k)$ for each of the possible configurations. Then, the supervisory controller evaluates, among all the received solutions, which one will be the best configuration for the local systems according to the global objective function that includes both performance and communication effort. Note that complexity of this calculation corresponds to the number of function evaluations among the total number N_c of possible structures, and the optimization to decide the next structure and the model of the overall system that the supervisory controller has available. Likewise, the controllers of the lower control layer that communicate following the supervisory controller instructions may use different communication burden depending on the particular scheme implemented to this purpose. This is beyond of the scope of this paper but some works have quantified the amount of communication required for distributed MPC schemes, e.g. [29, 30].

4.1. Information Topology

Consider the interactions between two subsystems S_i and S_j. In general, the control action sequences decided by the local MPC controller C_i are $\mathbf{u}_i(k)$ and $\mathbf{u}_{j,i}(k)$, and for C_j the sequences are $\mathbf{u}_j(k)$ and $\mathbf{u}_{i,j}(k)$. The control actions that are decided by the controller i and affecting the subsystem j are $\mathbf{u}_{j,i}(k)$, and analogous for C_j and $\mathbf{u}_{i,j}(k)$.

Let $\delta_{j,i}(k) = \{0, 1, 2\}$ represent the availability of C_i to communicate $\mathbf{u}_{j,i}(k)$ to C_j at time step k. In particular, $\delta_{j,i}(k) = 0$ if C_i does not share $\mathbf{u}_{j,i}(k)$ with C_j at all; $\delta_{j,i}(k) = 1$ if C_i shares the control sequence decided in the previous time step $\mathbf{u}_{j,i}(k-1)$, and $\delta_{j,i}(k) = 2$ if C_i shares its current decision $\mathbf{u}_{j,i}(k)$. These options lead to nine different cases for the way the controllers C_i and C_j can share their relevant information, as shown in Figure 1:

- In NC-MPC$_1$, $\delta_{i,j}(k) = \delta_{j,i}(k) = 2$. This case the local MPC controllers C_i and C_j, based for example on a consensus algorithm or any other distributed MPC approach, will decide their control actions jointly during the sampling time (coalition between subsystems S_i and S_j).
- In NC-MPC$_2$, $\delta_{i,j}(k) = 2$ and $\delta_{j,i}(k) = 1$. This case is a full-communication case as C_j and C_i communicate $\mathbf{u}_{i,j}(k)$ and $\mathbf{u}_{j,i}(k-1)$ respectively. Controller C_i knows that controller C_j will share information, and the optimization procedure of C_j will hierarchically communicate its resulting optimal variables. This suggests a hierarchical structure, where C_j is the master and C_i the slave at time step k. This is analogous for the case NC-MPC$_4$, with $\delta_{i,j}(k) = 1$ and $\delta_{j,i}(k) = 2$.
- In NC-MPC$_3$, $\delta_{i,j}(k) = 2$ and $\delta_{j,i}(k) = 0$. This case is a hierarchical case, where the information $\mathbf{u}_{i,j}(k)$ flows from C_j to C_i in a hierarchical way, but the controller C_i does not communicate its control actions. In this case, the controller C_j will include the effect of C_i using nominal values. There are different ways to incorporate the nominal values: using an optimized single static value, using a look-up table with a set of static variables suitable for different conditions, or via a dynamic model capable to estimate the unavailable information. This is analogous for the case NC-MPC$_5$ with $\delta_{i,j}(k) = 0$ and $\delta_{j,i}(k) = 2$.
- In NC-MPC$_6$, $\delta_{i,j}(k) = 0$ and $\delta_{j,i}(k) = 0$. The case is a decentralized one, where the effect of $\mathbf{u}_{i,j}(k)$ in the MPC controller C_i and the effect of $\mathbf{u}_{j,i}(k)$ in C_j are included in the optimization procedure by using nominal values, independently of the current or previous decision taken by those controllers.
- The cases NC-MPC$_7$, NC-MPC$_8$ and NC-MPC$_9$, are all decentralized. In case NC-MPC$_7$, with $\delta_{i,j}(k) = 1$ and $\delta_{j,i}(k) = 0$, only C_j communicates and it stores/transmits $\mathbf{u}_{i,j}(k-1)$. In case NC-MPC$_8$, with $\delta_{i,j}(k) = 0$ and $\delta_{j,i}(k) = 1$, only C_i communicates and it stores/transmits $\mathbf{u}_{j,i}(k-1)$. In case NC-MPC$_9$, with $\delta_{i,j}(k) = 1$ and $\delta_{j,i}(k) = 1$, both C_i and C_j communicate their whole control sequences. In the case when the control actions are not communicated, the controllers will consider the effect of the other controller using nominal values.

The number of possible communication topologies grows exponentially with the number of control actions involved in the control problem. In particular, if there are N_l control variables, 3^{N_l} different NC-MPC control topologies can be considered. Nevertheless, this number can be reduced because some of them may not make sense for a particular problem. For this reason, it is acceptable to assume that a set of meaningful possible control topologies is selected a priori. Given the large-scale nature of the considered problems, we assume that an offline component will limit the number of topologies. However, this paper mainly focuses on the management of the local controllers.

4.2. Optimization Methods for Switching Procedures

The supervisory controller solves an optimization problem by comparing and selecting the best NC-MPC structure at the moment of the switching. Each possible NC-MPC structure is determined by the variables $\delta_{i,j}(k)$. The supervisory controller evaluates the following global objective function that includes both performance and communication effort:

$$J(\mathbf{u}(k), x_k, \mathbf{d}(k)) = \sum_{m=1}^{N_c} J_m(\mathbf{u}(k), x_k, \mathbf{d}(k)) + \Lambda_{\mathrm{NC-MPC_c}}(k), \quad (13)$$

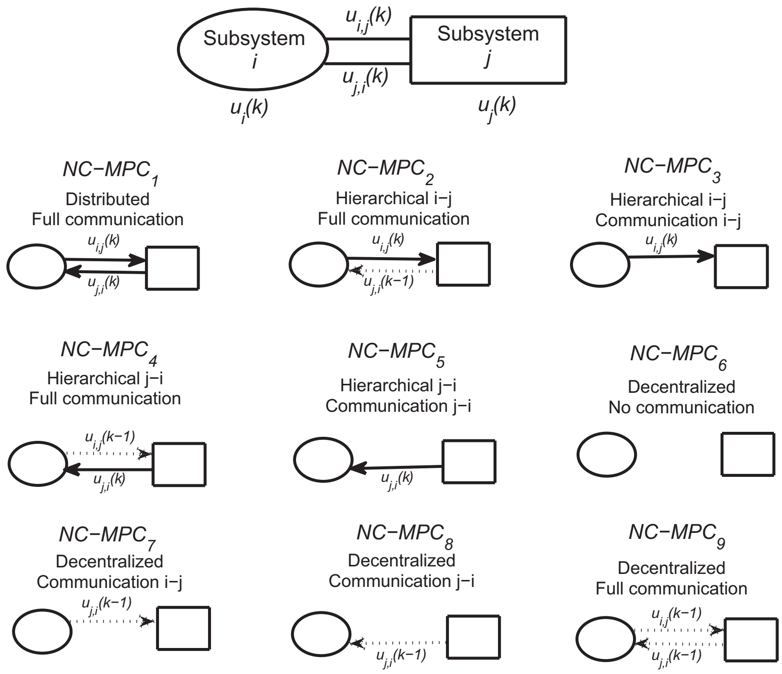


Figure 1: Possible control schemes for two subsystems. Full arrows indicate the control sequence at time step k is available, dotted arrows when the control sequence of time step $k-1$ is available, and no arrows when there is no flow of information.

where the first term represents the performance term, and the function $\Lambda_{\mathrm{NC-MPC_c}}$ penalizes the communication efforts of the $\mathrm{NC-MPC_c}$ topology. The computations of the control actions are done in a non-centralized manner, the function of the supervisory controller being to evaluate N_c times the objective function by using the information coming from the local controllers, and then to find the best NC-MPC structure. To facilitate the understanding of the control structure, Figure 2 presents a scheme of the global algorithm, which includes both offline and online components. The offline component consists of the heuristic approaches that allow to reduce the number of topologies to be evaluated by the supervisory controller. This reduction allows to keep a suitable balance between the global optimal solution (by evaluating the theoretical maximum number of possible topologies) and a reasonable computation time (depending on the application and its time constants). The off-line component together with proper solvers for the online part are crucial to keep the strategy tractable.

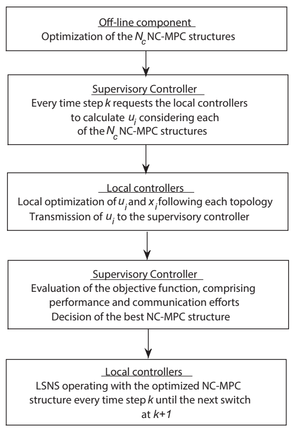
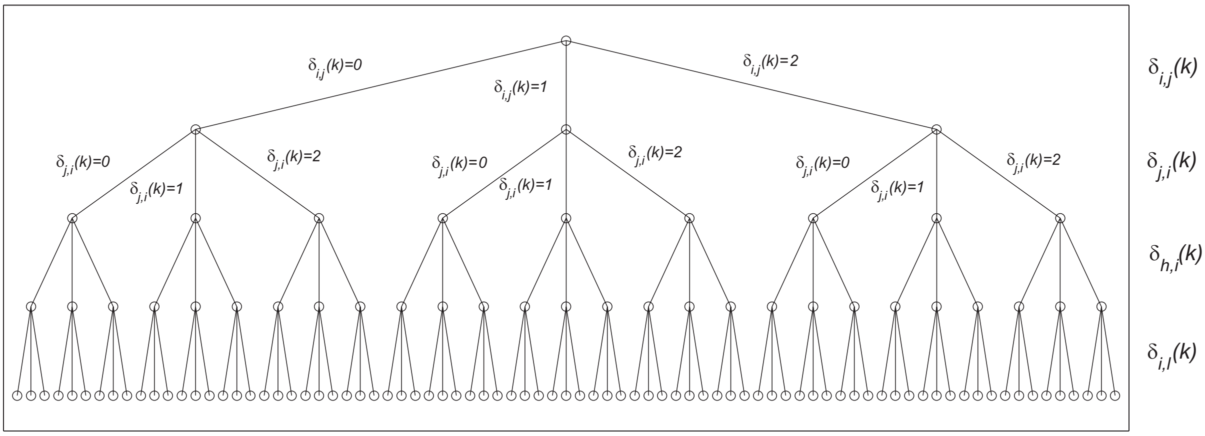


Figure 2: Sequential scheme of the global algorithm.

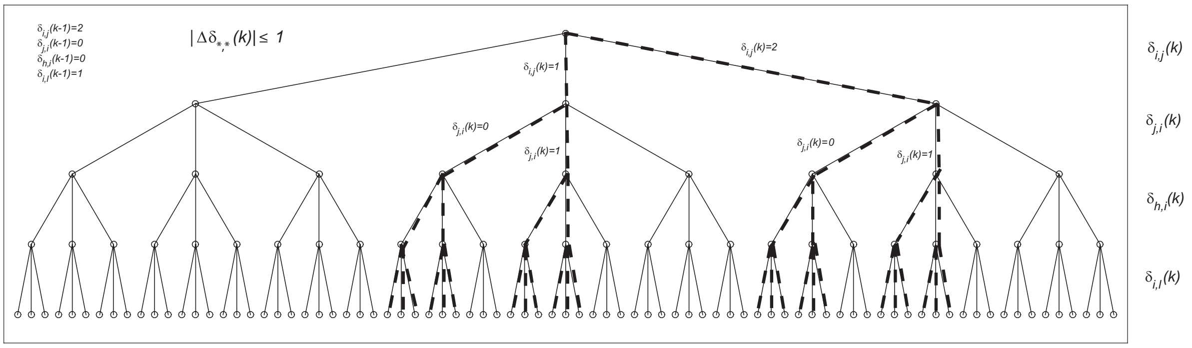
In order to understand the complexity of the combinational problem, consider the possible combinations of NC-MPC structures for a simple system with four possible decision variables are shown in Figure 3a. Each variable $\delta_{i,j}(k)$, $\delta_{j,i}(k)$, $\delta_{h,i}(k)$, and $\delta_{i,l}(k)$ can take three possible values. Then, full enumeration of all the possible combinations leads to 81 possible NC-MPC communication structures. As the full enumeration of all the possible NC-MPC structures is not practical, off-line reduction methods can be considered. One solution is bounding the variations of the communication signals $|\Delta\delta_{i,j}(k)|$, so as to avoid switching directly from fully communication to not communication. In Figure 3b, to reduce the complexity, the case $|\Delta\delta_{i,j}(k)| \leq 1$ is depicted. In this case:

- if $\delta_{i,j}(k-1) = 0$, then $\delta_{i,j}(k) \in \{0, 1\}$;
- if $\delta_{i,j}(k-1) = 1$, then $\delta_{i,j}(k) \in \{0, 1, 2\}$;
- if $\delta_{i,j}(k-1) = 2$, then $\delta_{i,j}(k) \in \{1, 2\}$.

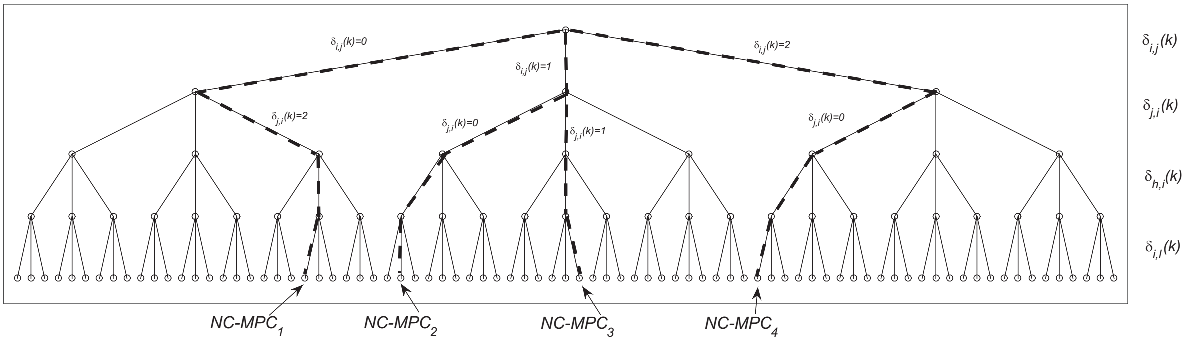
In the figure, $\delta_{i,j}(k-1) = 2$, $\delta_{j,i}(k-1) = 0$, $\delta_{h,i}(k-1) = 0$, and $\delta_{i,l}(k) = 1$. Then, for time step k, $\delta_{i,j}(k)$ can take two values (1 or 2), $\delta_{j,i}(k)$ and $\delta_{h,i}(k)$ can take the values 0 or 1, and $\delta_{i,l}(k)$ can take three possible values (0, 1 or 2). The total number of combinations for this case is 24. Another method to reduce



(a) Diagram to show the complexity of the problem, 81 NC-MPC controllers to be evaluated.



(b) Reduction method to simplify the problem. Thick dashed lines represent the selected solutions. In this case, 24 NC-MPC controllers are evaluated at time step k.



(c) Pruning to evaluate only the most relevant structures. Thick dashed lines represent the selected solutions. In this case, 4 NC-MPC controllers are evaluated at time step k.

Figure 3: Complexity of the optimization problem solved by the supervisory controller.

the complexity of the problem consists in holding any possible variation at least during a period of T time steps. Thus, the supervisory controller operates every step $k = h \cdot T$. Therefore, if $|\Delta\delta_{i,j}(k)| > 0$, then $\Delta\delta_{i,j}(k+t) = 0$ for $t = 1, ..., T$. A third option could be to limit the total number of variations per subsystem, so the communication will change gradually when the subsystem i has many different communication channels. In this way, $\sum_j |\Delta\delta_{i,j}(k)| < \Delta_i$, for a given Δ_i. The drawback of any of those methods or a combined method is the evaluation of still a considerable number of topologies. In this paper we propose to prune the search tree and to only consider a few set of relevant NC-MPC configurations, which are selected based on the application. For example, in Figure 3c a representation of the four more relevant NC-MPC configurations is presented. At the supervisory level, switches among only those NC-MPC structures will be allowed as shown in Figure 4. To obtain a good set of relevant NC-MPC configurations, a simulation-based procedure can be conducted to find the most effective topologies that lead to the best performance. Alternatively, interviews with experienced operators and knowledge based strategies with learning capabilities can be applied to select the best set based on real-life measurements and operation.

The supervisor will instruct the local controllers to calculate control actions under a limited number of communication scenarios (N_c). The sets of control actions proposed by the controllers for each communication scenario are then evaluated in

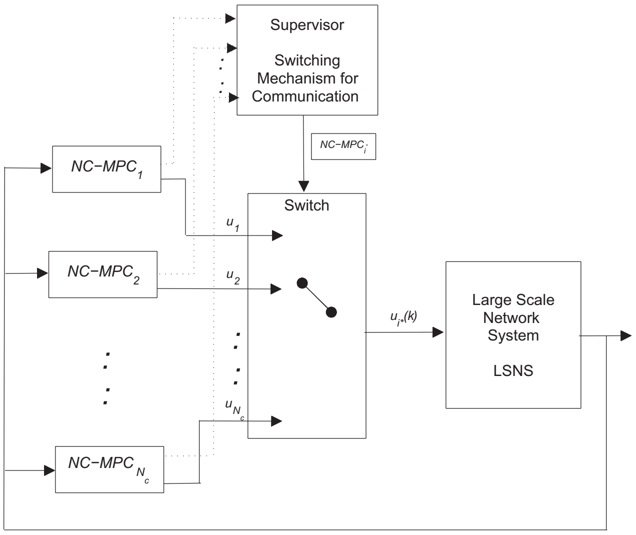


Figure 4: NC-MPC scheme with time-varying topologies.

a global model of the system available for the supervisor. Then, to determine the switches, the supervisor will weigh the solutions of the topology considering the following criteria:

- Minimization of the global objective function for performance, i.e. the first term in (13).
- Minimization of the communication effort over the ranges for i, j, t, given by $\sum_j \sum_i \sum_t \delta_{i,j}(k+t)$ weighted by a cost of the topology. For simplicity of the notation, this term was called $\Lambda_{\text{NC-MPC}_c}(k)$ in (13), comprising the cost of using the topology given by the controller NC-MPC$_c$.
- $\Delta\delta_{i,j}(k+t) = 0$ for $t = 1, ..., T$, to reduce the number of switches over time. We assume the supervisory controller operates every step $k = h \cdot T$.

Once the variables $\delta_{i,j}(k)$ are determined $\forall C_i, C_j \in C$, the supervisor will indicate the communication topology to be followed at time step k. To calculate the control sequences for each communication scenario, the local controllers receive from the supervisory controller the values of the variables $\delta_{i,j}(k)$ and $\delta_{j,i}(k)$ for all the communication channels of subsystem i. Then, in the case subsystem i is not waiting for information coming from upper levels, it will coordinate (or not) the solution of its optimization problem with the other subsystems at the same level of the hierarchy, and then it will transmit the control sequences to lower levels according to the communication instructions.

5. Case Study

In this section, we present simulations performed for a simple benchmark reported in [31]. It is composed of 16 water tanks arranged in a 4×4 matrix and each tank has a pipe that connects it with its direct neighbors. The control objective is to minimize a cost function including communication costs and performance. Figure 5 shows the possible control structures selected over the physical topology of the case study. The following discrete-time linear dynamics are assumed for each tank:

$$x_i(k+1) = x_i(k) + T_s \frac{1}{A_i} \sum_{j \in \mathcal{N}_i} u_{ij}(k), \tag{14}$$

where $x_i(k)$ is the level of the water in tank i and A_i is its cross-sectional area, T_s is the sampling time, $u_{ij}(k)$ is the flow through the pipe connecting tanks i and j, and $\mathcal{N}_i$ is the set of tanks connected to tank i. The parameters of the model are in Table 2. Each tank is governed by an agent that can manipulate the flow of all the outflow pipes it is connected to[3] and that can communicate the control variables to the connected tanks if the selected control structure commands to do so.

5.1. Control Structures

The following seven possible control structures have been selected:

1. *Big inflow coalition*: This option is shown in Figure 5a and represents the biggest possible coalition of subsystems that cooperate in order to coordinate the water inflow to the overall system. Notice that subsystem 1 has the monopoly of the external water inflow. For this reason, the remaining subsystems need the aid of the biggest coalition in case there is not enough water to reach the reference.
2. *Small inflow coalition*: This option is presented in Figure 5b. It corresponds to the case in which the four subsystems closest to the external water inflow are grouped into a coalition and the remaining subsystems work in a decentralized fashion. Again, such coalition could be formed when there is water scarcity in these subsystems. Notice that this option requires less coordination than the first one.
3. *Big outflow coalition*: This option is shown in Figure 5c and represents the biggest possible coalition of subsystems that cooperate in order to coordinate the water outflow leaving the overall system. Notice that subsystem 16 has the monopoly of the external water outflow. For this reason, the rest of the subsystems need the aid of subsystem 16 if there is too much water. Notice as well that, contrary to what happens in the case of water scarcity, in this case the subsystems can pump water to their neighbors.
4. *Small outflow coalition*: This option is presented in Figure 5d. It corresponds to the case in which the four subsystems closest to the external water outflow are grouped into a coalition and the rest of the subsystems work in a decentralized fashion. Again, such coalition could be formed when there is an excess of water in these subsystems. Notice that this option requires less coordination than the third structure.
5. *Control structure with hierarchical relationships*: The fifth possible control structure is depicted in Figure 5e. In particular, this alternative is a variation of option 1. In this

[3]Arrows represent the direction of the water flow.

case, tank 4 receives information from the actions that tank 3 is going to carry out. This information is then taken into account by the corresponding controller in order to calculate its control sequence.

6. *Control structure with information broadcast*: This option, which is represented in Figure 5f, is also a variation of option 1. In this case, however, the agent that regulates the water level of tank 13 also governs input 16. That is, this case represents a control structure in which there is a strict hierarchical relationship between different controllers: one controller is taking control of external inputs. Note that this case is also introduced to establish a comparison with option 5. As it can be seen, the situations of tanks 13 and 4 are symmetrical in the proposed case study. Hence, it is possible to analyze the consequences of two different relationships between controllers: one based on information broadcast and another based on a strict hierarchy in which there is a *transfer* of decision variables from one controller to another.
7. *Decentralized control structure*: The last option is shown in Figure 5g, which corresponds to a fully decentralized control scheme. In this case, there is no coordination among the subsystems and, for this reason, any subsystem with a water level below the reference cannot do anything by itself. On the other hand, subsystems with an excess of water can pump water out of their tanks to the neighboring tanks.

As additional comments, notice that subsystems 4 and 13 are never included in any coalition. This is not a problem for them whenever they have an excess of water, but they depend on their neighbors if they need it, which highlights the importance of the proper coalition formation. In Figure 5, local controllers that cooperate with full communication using a distributed scheme have been grouped into a single and bigger control entity. That is, the communication arrows and the individual agents have been omitted to highlight the fact that, under this type of cooperation, the controllers behave as a centralized unit.

Remark 5.1. *Other control topologies could have been considered and included in the example. Nevertheless, we believe that the above seven topologies allow us to illustrate how the cooperation can be increased and decreased according to the situation of the system regarding its objective. Likewise, this choice also allows us to point out the consequences of the different type of relationships that can be established between the local controllers.*

5.2. Simulation parameters

The simulation is implemented in the following way: each simulation step corresponds to 0.15 s. Every two simulation steps the controllers update their control actions according to the topology selected, which, in turn, can change each ten simulation steps.

The parameters used in the simulation are listed in Table 2. In this example, the control scheme recalculates the most appropriate system partitioning each five time steps (h introduced in previous section). A time step is defined as two times the simulation step. To this end, if the time step index k is a multiple of 5, the following global cost function is minimized:

$$J(\mathbf{u}(k), x_k, \mathbf{d}(k)) = \sum_{m=1}^{16} J_m(\mathbf{u}(k), x_k, \mathbf{d}(k)) + \Lambda_{\mathrm{NC-MPC_c}}(k), \quad (15)$$

where $\Lambda_{\mathrm{NC-MPC_c}}$ stands for the communication costs associated to the partitioning given by the topology c used and J_m is the local cost function that stands for the local objectives that each subsystem has, which is defined as:

$$\begin{aligned} J_m(\mathbf{u}(k), x_k, \mathbf{d}(k)) \quad &= \sum_{l=0}^{N_\mathrm{p}-1} e_m(k+l+1)^\mathrm{T} Q_m \, e_m(k+l+1) \\ &\quad + u_m^\mathrm{T}(k+l) R_m u_m(k+l), \end{aligned} \quad (16)$$

with $e_m(k+l+1) \triangleq x_m(k+l+1) - x_m^\mathrm{r}$. The values corresponding to the reference x_m^r, and the weighting matrices Q_m and R_m are given in Table 2.

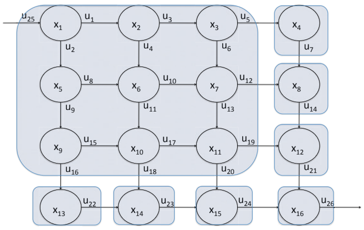
During the remaining four time steps the topology remains constant and the members of each partition calculate jointly their actions in order to minimize the sum of the corresponding $J_m(k)$. The solver used is Quadprog from Matlab. For the integer variables $\delta_{ij}(k)$ explicit enumeration was employed.

Different topologies will have different costs. These cost values represent the coordination efforts made by the control scheme. In particular, no penalty is assigned for topology 7 because it represents a fully decentralized control scheme. Topologies 2 and 4 are slightly penalized because each one introduces cooperation between four local controllers. A stronger penalty is assigned to topologies 1 and 3 because of their greater cooperation degree. Finally, the maximum penalty considered in this case study is assigned to topologies 5 and 6 since they involve an additional communication link in comparison with topology 1.

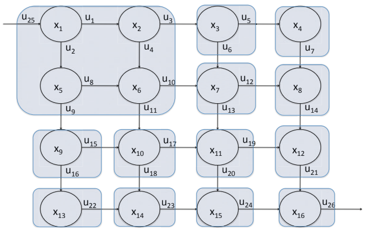
5.3. Results

Figure 6 shows the evolution of the water levels of the system controlled with the proposed switching scheme when the initial level is 0.25 m for each tank. Given that tank 1 is the only one equipped with an external controllable input, the supply of water for all the tanks in the system depends exclusively on this subsystem. Hence, the corresponding controller has an important role in the coordination process needed to supply water for all the tanks.

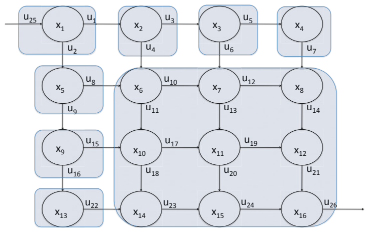
In Figure 6, the evolution of the control topology is also shown. The system starts by using the control structure 2, which makes sense since the amount of water supplied by controller 1 is limited. For this reason, it is only worth to coordinate the actions with the closest neighbors. A few steps later, however, the cooperation grows and the control structure 1 is selected. As the situation of the newly aggregated agents improves, the structure goes back to 2. Once again, the coordination group is enhanced but this time control structure 6 is selected, i.e., controller 13 is given priority and is allowed to govern u_{16} too. In this way, it is able to achieve the desired set point. From that moment on, the control structure goes to 2



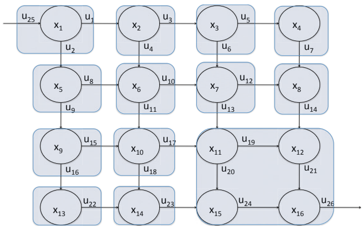
(a) Big inflow coalition



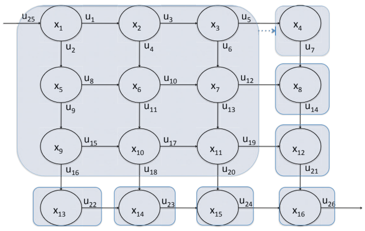
(b) Small inflow coalition



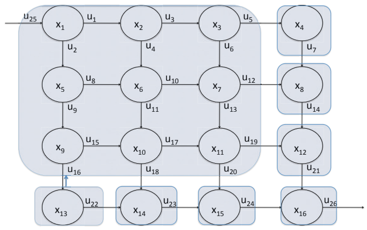
(c) Big outflow coalition



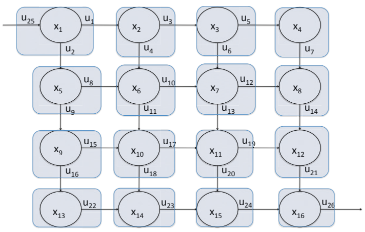
(d) Small outflow coalition



(e) Control structure with hierarchical relationship



(f) Control structure with information broadcast



(g) Decentralized control structure

Figure 5: Possible control structures.

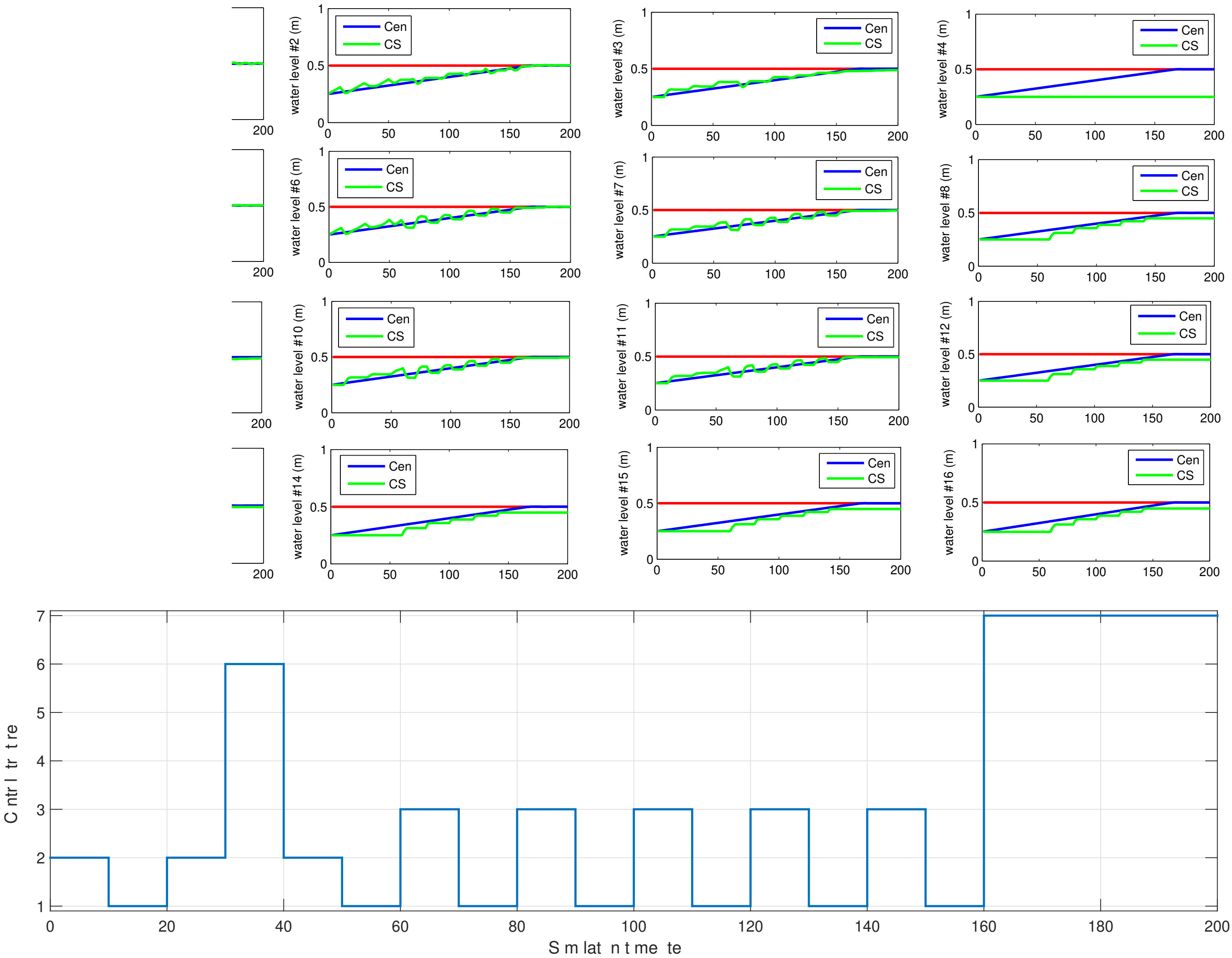


Figure 6: Evolution of the water levels of all the tanks and the control structure when the initial state is 0.25 m for all the tanks. The x-axis corresponds to the simulation step. The reference is 0.5 m. Cen is the Centralized MPC solution, CS is the proposed NC-MPC method. Red represents reference.

and, after that, there is a succession of commutations between control structures 1 and 3. Taking into account that none of the control structures allows centralized coordination, switching in this way is a suitable mechanism to achieve a good performance, i.e. the biggest groups of subsystems for coordinating inflow and outflow alternate in order to distribute the water all over the network. Finally, the last control structure selected is 7, which is the completely decentralized control structure. In this case, there is nothing that can be gained from cooperation, at least taking into account the price of communication.

Another simulation has been performed using an initial level of 0.75 m for each tank. The corresponding results are shown in Figure 7. Note that here the coordination degree required to reduce the excess of water is lower because each controller can pump out water independently. However, constraint satisfaction requires coordination. For this reason, the control structures that are selected in this simulation are 3, 4, and 7.

Finally, controller structure 5 deserves some comments. It is a variation of controller structure 1, in which subsystem 4 receives information from the group of subsystems that work in a coordinated fashion. While this broadcast of information is meaningful for controller 4 to calculate its control action, it does not improve the overall performance. Thus controller 1 is selected more often than 4 because of the additional cost of communications. In addition, it is well known that information broadcast is not as efficient as information exchange in order to improve the overall performance [6].

6. Conclusions

A non-centralized MPC controller that adapts to different operational conditions by switching between topologies is proposed in this paper. Including the changes in the topology explicitly in the predictions leads to an NP-Hard combinatorial mixed-integer optimization problem that we solve for a limited number of cases. This allows to include the dynamic effect of the switching explicitly in the prediction model. The controller was tested on a water distribution system, showing its effective-

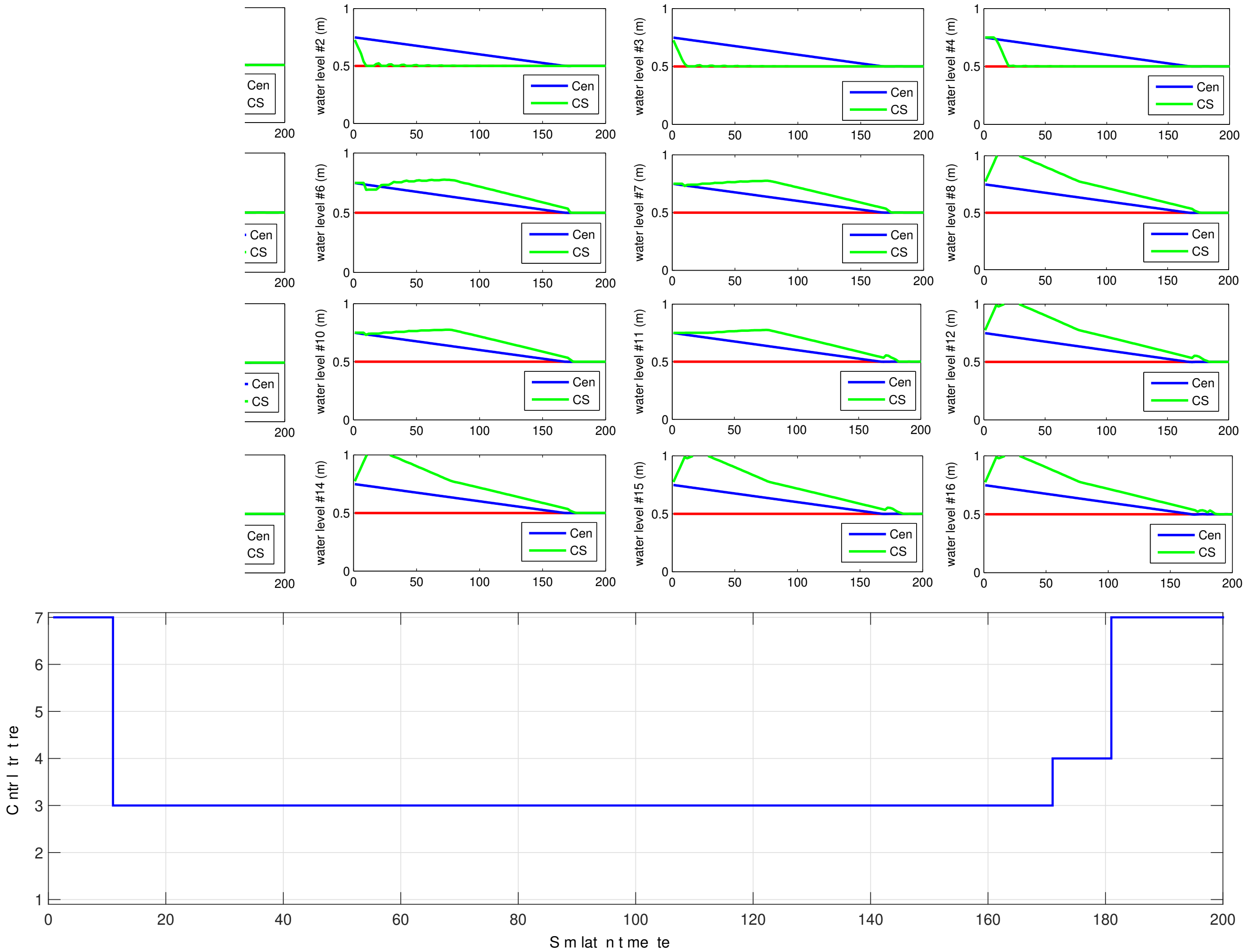


Figure 7: Evolution of the water levels of all the tanks and the control structure when the initial state is 0.75 m for all the tanks. The x-axis corresponds to the simulation step. The reference is 0.5 m. Cen is the Centralized MPC solution, CS is the proposed NC-MPC method. Red represents reference.

Table 2: Model and controller parameters

Parameter	Symbol	Value
System		
Storage area, $\forall i$	A_i	3.14 (m^2)
Simulation time step length	T_s	0.15 (s)
Controller		
Control time step length	T_c	0.3 (s)
Topology switching time step length	T_{ts}	1.5 (s)
Prediction horizon	N_p	5 (s)
Quadratic penalty weight on x_i, $\forall i$	Q_i	1
Quadratic penalty weight on u_i, $\forall i$	R_i	1
Reference tank i, $\forall i$	x_i^r	0.5
Cost topologies 1, 3	$\Lambda_{\text{NC-MPC}_c}$	30
Cost topologies 2, 4	$\Lambda_{\text{NC-MPC}_c}$	10
Cost topologies 5, 6	$\Lambda_{\text{NC-MPC}_c}$	35
Cost topology 7	$\Lambda_{\text{NC-MPC}_c}$	0
Maximum pump capacity	$u_{\max}$	0.5 (m^3/s)
Minimum pump capacity	$u_{\min}$	0 (m^3/s)
Maximum water level	$x_{\max}$	1 (m)
Minimum water level	$x_{\min}$	0 (m)

ness to adapt to different operational topologies according the relative importance of the different topologies.

Several research lines can be proposed from the ideas discussed in this work, including issues related to the partitioning of the dynamical system seen as network composition of elements, as well as robust feasibility and stability when considering switching partitioning and control topologies. Moreover, further analysis about general robustness of the non-centralized control schemes and their influence over the overall system performance arise as topics of current and future interest around this research.

Conflict of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

Research supported by the European 7th Framework Network of Excellence Highly complex and networked control systems (HYCON2), the Secretaria d'Universitats i Recerca i del Departament da Economia i Coneixement of Generalitat de Catalunya, the European COST Action TU1102, and the Spanish project ECOCIS (ref. DPI 2013-48243-C2-1-R). Financial support by the FP7-ICT project DYMASOS (ref. 611281) is also gratefully acknowledged.

References

[1] R. R. Negenborn, P. J. van Overloop, T. Keviczky, B. De Schutter, Distributed model predictive control of irrigation canals, Networks and Heterogeneus Media 4 (2) (2009) 359–380. doi:10.3934/nhm.2009.4.359.

[2] R. R. Negenborn, B. De Schutter, H. Hellendoorn, Multi-agent model predictive control for transportation networks: Serial versus parallel schemes, Engineering Applications of Artificial Intelligence 21 (3) (2008) 353–366. doi:10.1016/j.engappai.2007.08.005.

[3] C. Ocampo-Martinez, V. Puig, G. Cembrano, J. Quevedo, Application of predictive control strategies to the management of complex networks in the urban water cycle [applications of control], IEEE Control Systems Magazine 33 (1) (2013) 15–41. doi:10.1109/MCS.2012.2225919.

[4] J. Maestre, D. Muñoz de la Peña, E. F. Camacho, T. Alamo, Distributed model predictive control based on agent negotiation, Journal of Process Control 21 (5) (2011) 685–697. doi:10.1016/j.jprocont.2010.12.006.

[5] R. Scattolini, Architectures for distributed and hierarchical model predictive control: A review, Journal of Process Control 19 (5) (2009) 723–731. doi:10.1016/j.jprocont.2009.02.003.

[6] R. Negenborn, J. Maestre, Distributed model predictive control: An overview and roadmap of future research opportunities, IEEE Control Systems Magazine 34 (4) (2014) 87–97. doi:10.1109/MCS.2014.2320397.

[7] P. Trodden, A. Richards, Adaptive cooperation in robust distributed model predictive control, in: Proceedings of the 2009 IEEE Control Applications (CCA) and Intelligent Control (ISIC), St. Petersburg, Russia, 8-10 July 2009, pp. 896–901. doi:10.1109/CCA.2009.5280964.

[8] M. Jilg, O. Stursberg, Optimized distributed control and topology design for hierarchically interconnected systems, in: Proceedings of the 2013 European Control Conference, Zurich, Switzerland, 17-19 July 2013, pp. 4340–4346.

[9] T. Sadamoto, T. Ishizaki, J.-I. Imura, Hierarchical distributed control for networked linear systems, in: Proceedings of the 2014 IEEE 53rd Annual Conference on Decision and Control (CDC), 2014, pp. 2447–2452. doi:10.1109/CDC.2014.7039762.

[10] J. M. Maestre, D. Muñoz de la Peña, A. Jiménez Losada, E. Algaba, E. F. Camacho, A coalitional control scheme with applications to cooperative game theory, Optimal Control Applications and Methods 35 (5) (2014) 592–608. doi:10.1002/oca.2090.

[11] F. Fele, J. Maestre, S. Mehdy Hashemy, D. Muñoz de la Peña, E. Camacho, Coalitional model predictive control of an irrigation canal, Journal of Process Control 24 (4) (2014) 314–325. doi:10.1016/j.jprocont.2014.02.005.

[12] F. Muros, J. Maestre, E. Algaba, T. Alamo, E. Camacho, An iterative design method for coalitional control networks with constraints on the shapley value, in: Proceedings of the 19th World Congress IFAC, Cape Town, South Africa, 24-29 August 2014, pp. 1188–1193.

[13] A. Nuñez, C. Ocampo-Martínez, B. De Schutter, F. Valencia, J. López, J. Espinosa, A multiobjective-based switching topology for hierarchical model predictive control applied to a hydro-power valley, in: Proceedings of the 2013 IFAC International Conference on Intelligent Control and Automation Science, Chengdu, China, 2-4 September 2013, pp. 534–539.

[14] A. Anand, L. Samavedham, Optimizing the communication topology in a coordinated model predictive control architecture, in: Proceedings of the 5th International Symposium on Advanced Control of Industrial Processes, Hiroshima, Japan, 28-30 May 2014, pp. 318–323.

[15] C. Ocampo-Martinez, S. Bovo, V. Puig, Partitioning approach oriented to the decentralised predictive control of large-scale systems, Journal of Process Control 21 (5) (2011) 775–786. doi:10.1016/j.jprocont.2010.12.005.

[16] M. E. Sezer, D. Siljak, Nested ε-decomposition and clustering of complex systems, Automatica 22 (3) (1986) 321–331. doi:10.1016/0005-1098(86)90030-0.

[17] J. Anderson, Y. Chang, A. Papachristodoulou, Model decomposition and reduction tools for large-scale networks in systems biology, Automatica 47 (6) (2011) 1165–1174. doi:10.1016/j.automatica.2011.03.010.

[18] S. Kamelian, K. Salahshoor, A novel graph-based partitioning algorithm for large-scale dynamical systems, International Journal of Systems Science 46 (2) (2015) 227 – 245. doi:10.1080/00207721.2013.775395.

[19] H. Li, Y. Shi, Distributed receding horizon control of large-scale nonlinear systems: Handling communication delays and disturbances, Automatica 50 (4) (2014) 1264–1271. doi:10.1016/j.automatica.2014.02.031.

[20] C. Ocampo-Martinez, V. Puig, J. Grosso, S. Montes, Ch. 31 : Multi-layer decentralized model predictive control of large-scale networked systems, distributed model predictive control made easy, Vol. 69 of Intelligent Systems, Control and Automation: Science and Engineering, Springer, 2014, pp. 495–515. doi:10.1007/978-94-007-7006-5.31.

[21] J. Grosso, On model predictive control for economic and robust operation of generalised flow-based networks, Ph.D. thesis, Universitat Politècnica de Catalunya - BarcelonaTech, Barcelona (2015).

[22] R. S. Sánchez-Pena, F. D. Bianchi, Model selection: From LTI to switched-LPV, in: Proceedings of the American Control Conference, Montreal, Canada, 27-29 June 2012, pp. 1561–1566. doi:10.1109/ACC.2012.6315134.

[23] J. Maciejowski, Predictive Control with Constraints, Prentice Hall, U.K., 2002.

[24] J. M. Maestre, R. R. Negenborn (Eds.), Distributed Model Predictive Control Made Easy, Vol. 69 of Intelligent Systems, Control and Automation: Science and Engineering, Springer, 2014. doi:10.1007/978-94-007-7006-5.

[25] P. Frasca, H. Ishii, C. Ravazzi, R. Tempo, Distributed randomized algorithms for opinion formation, centrality computation and power systems estimation: A tutorial overview, European Journal of Control 24 (1) (2015) 2–13. doi:10.1016/j.ejcon.2015.04.002.

[26] Z. Li, Z. Duan, G. Chen, L. Huang, Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint, IEEE Transactions on Circuits and Systems 57 (1) (2010) 213–224. doi:10.1109/TCSI.2009.2023937.

[27] D. Ozan, J. Lunze, Optimal and event-based networked control of physically interconnected systems and multi-agent systems, International Journal of Control 87 (1) (2014) 169–185. doi:10.1080/00207179.2013.825816.

[28] R. Schuh, J. Lunze, Desing of the communication structure of a self-organizing networked controller for heterogeneous agents, in: Proceedings of the 2015 European Control Conference, Linz, Austria, 15-17 July 2015, pp. 2199–2206.

[29] I. Alvarado, D. Limon, D. Muñoz de la Peña, J. Maestre, M. Ridao, H. Scheu, W. Marquardt, R. Negenborn, B. De Schutter, F. Valencia, J. Espinosa, A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark, Journal of Process Control 21 (5) (2011) 800–815. doi:10.1016/j.jprocont.2011.03.003.

[30] J. Maestre, M. Ridao, A. Kozma, C. Savorgnan, M. Diehl, M. Doan, A. Sadowska, T. Keviczky, B. De Schutter, H. Scheu, W. Marquardt, F. Valencia, J. Espinosa, A comparison of distributed MPC schemes on a hydro-power plant benchmark, Optimal Control Applications and Methods 36 (3) (2015) 306–332. doi:10.1002/oca.2154.

[31] J. M. Maestre, F. Muros, F. Fele, E. F. Camacho, An assessment of coalitional control in water systems, in: Proceedings of 14th European Control Conference, Linz, Austria, 15-17 July 2015.