Distributed model predictive control using optimality condition decomposition and community detection

P. Segovia[1]*, V. Puig[2,3], E. Duviella[1], L. Etienne[1]

[1] *CERI Digital Systems, IMT Lille Douai, F-59000 Lille, France*

[2] *Research Center for Supervision, Safety and Automatic Control (CS2AC), Universitat Politècnica de Catalunya (UPC), Rambla Sant Nebridi 22, 08222 Terrassa, Spain*

[3] *Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens i Artigas 4-6, 08028 Barcelona, Spain*

Abstract

This work regards the development of a distributed model predictive control strategy for large-scale systems, as centralized implementations often suffer from non-scalability. The decomposition of the overall system into minimally coupled subsystems as well as their coordination are based on optimality condition decomposition (OCD) and community detection. The OCD approach allows to solve the associated control subproblems in parallel in an iterative manner until the required degree of accuracy is attained. The proposed strategy is tested on two different systems, the quadruple-tank system and the Barcelona drinking water network, which allow to highlight the effectiveness of the approach.

Keywords: Large-scale systems, model predictive control, distributed control, optimality condition decomposition, community detection.

1. Introduction

The design and operation of technological processes have experienced a great development in the last decades, giving rise to large-scale systems, which are characterized by the spatial distribution of their constitutive elements and the

*Corresponding author: P. Segovia (pablo.segovia@imt-lille-douai.fr)

transmission of energy and matter among these elements. Therefore, the ever-increasing complexity of these systems requires the use of advanced control strategies to govern their behavior [1].

One of the techniques that has received more attention is model predictive control (MPC). A large number of methods fall under the umbrella of MPC, sharing several features such as the explicit use of an approximate model of the process to predict the effect of a certain action on the system, the definition of a set of operational goals to quantify the system performance and the receding strategy that shifts the prediction horizon towards the future [2]. Furthermore, MPC has been widely applied in the industry given its intuitiveness and versatility, a fact that can be testified by the large number of industrial applications reviewed in [3].

Although MPC has proven to perform remarkably well in a vast array of domains and test cases, its implementation is often impractical when performed in a centralized manner (CMPC). Indeed, the distinct features that characterize large-scale systems often result in a non-scalable model, i.e., the maintenance and update of a huge centralized model at every change in the system configuration constitutes an arduous task [4]. Moreover, the reliability of the network might be jeopardized if the decisions are centralized in a single controller [5]. These reasons foster the advancement of non-centralized control strategies, which divide the control effort among several local controllers, each in charge of a portion of the system. This approach allows to reduce the control design complexity and the computational demand. MPC is no exception to this paradigm, which can be realized by the extensive body of literature on non-centralized MPC. A comprehensive overview of methods and applications can be found in [6–8] and references therein.

In order to design a non-centralized MPC, the overall system must be divided into a set of smaller subsystems. The goals of the decomposition are twofold: first, to ensure that each subproblem is much smaller than the overall problem, i.e., there are fewer decision variables and constraints in the subproblems than in the overall problem; and second, to ensure that each subproblem is only

coupled to a few other subproblems, i.e., each subproblem shares variables with just a few other subproblems [9]. However, the choice of decomposition is, in turn, a sensitive matter, as it has a decisive effect on the performance and the computation time, which can be considerably improved by selecting an appropriate decomposition [10].

System decomposition allows for distributed resolution of the subproblems. A large number of distributed optimization techniques have been reported in the literature, and can be classified into two different groups, according to whether they are based on augmented Lagrangian decomposition or decentralized solution of the KKT conditions for local optimality [11]. Two of the most popular techniques are the alternating direction method of multipliers (ADMM) and the optimality conditions decomposition (OCD), and pertain to the first and second class of approaches, respectively. Although somewhat similar in essence, there are some differences between these two approaches. First, the Lagrangian function employed in ADMM is augmented with additional terms associated to the constraint residuals [12], which is not done in the case of the OCD. Instead, OCD maintains all coupling constraints in its own subproblem while relaxing them in adjacent subproblems [13]. Moreover, while ADMM duplicates variables shared by multiple subproblems and adds equality constraints to ensure consistency of the local solutions, OCD assigns each variable to a specific subproblem, and considers that those variables assigned to the subproblem are the only ones allowed to change in its resolution [14]. As a result, ADMM requires a central coordinator to manage the dual variable update step, while OCD does not [14].

Stability and convergence of the solution yielded by distributed optimization approaches need to be ensured. These topics have been extensively studied, and several results on stability [15–17] and convergence [18–20] have been derived. In particular, very recent results have been reported in [21], where an ADMM-based computationally-efficient distributed optimization algorithm with guaranteed stability and convergence is provided. This is a novel result, as it tackles the case of distributed optimization under non-convex constraints

and provides a generic algorithm for distributed nonlinear MPC with localized model information.

Summary of the paper and contribution

This work regards the development of a distributed MPC (DMPC) strategy for large-scale systems. The approach is divided in two parts: first, a decomposition and coordination method is introduced to divide the overall system into a set of subsystems that are minimally coupled to each other. Indeed, the system partitioning problem is carefully examined, as its outcome is closely tied to the control performance. Then, the control subproblems are synthesized by considering the non-centralized MPC framework, and their solutions are coordinated using the selected policy. Stability and convergence of the solution are analyzed, and the performance of the approach is tested on two different case studies.

The contributions of this work with respect to the state of the art are listed below:

- Some preliminary results regarding the proposed DMPC approach were obtained and discussed in [22]. However, while several aspects such as the stability and the convergence of the solution yielded by the method were not explicitly addressed, these properties are duly analyzed in this work.
- Following the above discussion on different distributed optimization approaches, the OCD approach is selected. Its original derivation was illustrated on *static* optimization problems, i.e., devoid of the temporal component that is intrinsic to control problems. Therefore, this work adapts the method to solve DMPC problems. A review of the literature reveals that the integration of the OCD approach in a DMPC scheme is a novel approach: although some preliminary results are reported in [23], the proposed method is not demonstrated for a DMPC.
- On the other hand, the decomposition strategy employed in [22] only took into account the structural properties of the system, thus only considering

the couplings introduced by the dynamics. Conversely, the decomposition methodology is refined in this work, using the Karush-Kuhn-Tucker (KKT) optimality conditions and some ideas from community detection, which allows to consider couplings that may be introduced by the cost function.

- The use of community detection to solve DMPC problems is relatively recent [24–29]. However, the cost function is usually required to be completely separable, either by problem construction or by manipulation of the original problem, introducing auxiliary variables and constraints. Conversely, the proposed approach only requires the cost function not to be completely coupled, thus overriding the need to manipulate the problem, which might render its formulation more intricate.

- The cost function is taken into account at the decomposition stage by considering the optimality conditions of the overall problem. Then, the OCD determines the coordination policy as a result of the obtained decomposition. By contrast, other existing decomposition approaches do not determine directly how the subproblems should be coordinated, and thus require to use an additional coordination strategy.

- The proposed approach allows for a rather direct extension to the robust control case considering tube-based approaches using the results in [30].

The remainder of this paper is structured as follows: Section 2 introduces some necessary background material. Section 3 states the problem and identifies the required steps to solve it. Section 4 details the proposed approach: the OCD and the community detection frameworks are described first, allowing to synthesize the DMPC. Moreover, the convergence and stability of the solution are also analyzed. Finally, Section 5 tests the control approach on the quadruple-tank process and the Barcelona drinking water network (DWN), which allows to draw conclusions and outline future research directions in Section 6.

Notation

Throughout this paper, let $\mathbb{Z}_{\geq 0}$ and $\mathbb{R}^n$ denote the set of natural non-negative scalars and the space of n-dimensional column vectors with real entries, respectively. Scalars are denoted by either lowercase or uppercase letters; vectors, by bold lowercase letters; matrices, by bold uppercase letters; and sets, by calligraphic symbols. Set inclusion is indicated with the symbol $\subseteq$, whereas the union and intersection of multiple sets are denoted with $\bigcup_{i=0}^{N} \mathcal{X}_i$ and $\bigcap_{i=0}^{N} \mathcal{X}_i$, respectively. Moreover, $\|\mathbf{x}\|_{\mathbf{K}}^2 \triangleq \mathbf{x}^\intercal \mathbf{K} \mathbf{x}$, and $\max(\mathbf{x})$ and $|\mathbf{x}|$ indicate the maximum entry and the absolute value of $\mathbf{x}$, respectively. Furthermore, transposition is denoted with the superscript $^\intercal$; and element-wise relations of vectors, with the operators $<$, $\leq$, $=$, $\geq$ and $>$.

2. Preliminaries

2.1. Model predictive control

MPC is an optimization-based control technique that employs a dynamical representation of the process to solve a finite-horizon optimization problem at each time instant over a certain horizon. As a result, a control sequence that minimizes a cost function subject to physical and operational constraints is obtained [31].

Consider that the system to be controlled can be described using the general linear discrete-time state-space representation

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{1a}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k, \tag{1b}$$

where $k \in \mathbb{Z}_{\geq 0}$ denotes the current discrete-time instant, the vectors $\mathbf{x}_k \in \mathbb{R}^{n_x}$, $\mathbf{u}_k \in \mathbb{R}^{n_u}$ and $\mathbf{y}_k \in \mathbb{R}^{n_y}$ represent the system states, control inputs and system outputs, respectively, and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are time-invariant matrices of suitable dimensions.

Then, an MPC can be designed to fulfill the operational goals associated to (1). The optimal control sequence to be applied to the system is given by the solution of the following open-loop optimization problem:

$$\min_{\{\mathbf{u}_{i|k}\}_{i=k}^{k+H_p-1},\, \{\mathbf{x}_{i|k}\}_{i=k}^{k+H_p}} J\left(\mathbf{u}_{i|k}, \mathbf{x}_{i|k}\right) \tag{2a}$$

subject to

$$\mathbf{x}_{i+1|k} = \mathbf{A}\mathbf{x}_{i|k} + \mathbf{B}\mathbf{u}_{i|k}, \qquad i \in \{k, ..., k+H_p-1\}, \tag{2b}$$

$$\mathbf{u}_{i|k} \in \mathcal{U}, \qquad i \in \{k, ..., k+H_p-1\}, \tag{2c}$$

$$\mathbf{x}_{i|k} \in \mathcal{X}, \qquad i \in \{k, ..., k+H_p-1\}, \tag{2d}$$

$$\mathbf{x}_{k+H_p|k} \in \mathcal{X}_{H_p}, \tag{2e}$$

$$\mathbf{x}_{k|k} = \mathbf{x}_k, \tag{2f}$$

where $\{\mathbf{u}_{i|k}\}_{i=k}^{k+H_p-1} \triangleq \{\mathbf{u}_{k|k}, \mathbf{u}_{k+1|k}, \cdots, \mathbf{u}_{k+H_p-1|k}\}$ and $\mathbf{x}_{i|k}$ is defined in the same manner, with k the current time instant, i the time instant along the prediction horizon, and $k+i|k$ the predicted value of the variable at instant $k+i$ using information available at instant k. Moreover, $J\left(\mathbf{u}_{i|k}, \mathbf{x}_{i|k}\right)$ allows to determine the cost throughout the prediction horizon H_p, and $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ represent the feasible sets according to the physical and operational constraints.

Remark 1. A terminal constraint set and a terminal cost are added to (2) to stabilize the plant [32]. In particular, the terminal constraint set $\mathcal{X}_{H_p}$ in (2e) can be defined as an invariant ellipsoidal set of the form [30]

$$\mathcal{X}_{H_p} = \left\{ \mathbf{x}_{k+H_p|k} \in \mathbb{R}^{n_x} \,\middle|\, \mathbf{x}_{k+H_p|k}^{\intercal} \mathbf{Q} \mathbf{x}_{k+H_p|k} \leq 1 \right\}, \tag{3}$$

whereas the terminal cost is formulated as $\left\|\mathbf{x}_{k+H_p|k}\right\|_{\mathbf{Q}}^2$, and $\mathbf{Q}$ is chosen as the corresponding LQR gain [33].

The optimal control sequence (with respect to the chosen criteria) is given by $\{\mathbf{u}_{i|k}\}_{i=k}^{k+H_p-1}$, provided that the problem is feasible. However, only $\mathbf{u}_{k|k}$ is

applied to the system, according to the receding philosophy

$$\mathbf{u}_k^{MPC} \triangleq \mathbf{u}_{k|k}, \tag{4}$$

which is repeated at the next time instant to exploit the most recent measurements, thus transforming the original open-loop strategy into a closed-loop one.

2.2. Lagrangian relaxation

Consider that the control problem (2) is reformulated as follows:

$$\min_{\mathbf{z}} \; f(\mathbf{z}) \tag{5a}$$

subject to

$$\mathbf{a}(\mathbf{z}) = \mathbf{0}, \tag{5b}$$

$$\mathbf{b}(\mathbf{z}) \leq \mathbf{0}, \tag{5c}$$

where $\mathbf{z} \in \mathbb{R}^{n_z}$, $f(\mathbf{z}) : \mathbb{R}^{n_z} \to \mathbb{R}$, $\mathbf{a}(\mathbf{z}) : \mathbb{R}^{n_z} \to \mathbb{R}^{n_a}$ and $\mathbf{b}(\mathbf{z}) : \mathbb{R}^{n_z} \to \mathbb{R}^{n_b}$. Note that $\mathbf{z}$ includes all decision variables in (2).

The original problem can be transformed by using the method of Lagrange multipliers to facilitate its resolution. Then, the Lagrangian function associated to (5) can be defined as [34]

$$\mathcal{L}(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f(\mathbf{z}) + \boldsymbol{\lambda}\mathbf{a}(\mathbf{z}) + \boldsymbol{\nu}\mathbf{b}(\mathbf{z}), \tag{6}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ are the Lagrange multiplier vectors of suitable dimensions associated to $\mathbf{a}(\mathbf{z}) = \mathbf{0}$ and $\mathbf{b}(\mathbf{z}) \leq \mathbf{0}$, respectively [35].

This function can then be used to formulate a relaxed version of the original problem [36]:

$$\min_{\mathbf{z}} \; \mathcal{L}(\mathbf{z}, \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}}), \tag{7}$$

where $\bar{\boldsymbol{\lambda}}$ and $\bar{\boldsymbol{\nu}}$ indicate fixed values of $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$, respectively. The main advantage of (7) over the original problem (5) is that it can generally be decomposed into subproblems

$$\min_{\mathbf{z}^{(i)}} \sum_{i=1}^{N} \mathcal{L}^{(i)} \left(\mathbf{z}^{(i)}, \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}} \right), \tag{8}$$

which can be solved in parallel and yield an approximate solution of (5).

Remark 2. $\mathbf{z}^{(i)}$ denotes the subset of variables that belong to the i-th subproblem, with $\mathbf{z}^{(i)} \in \mathbf{z}$ and $\bigcup_{i=1}^{N} \mathbf{z}^{(i)} = \mathbf{z}$. Moreover, $\bigcap_{i=1}^{N} \mathbf{z}^{(i)} = \emptyset$ corresponds to the particular case in which the overall problem decomposes into completely decoupled subproblems.

The convergence of the solution for the subproblems (8) is achieved by updating the values of the Lagrange multipliers in each subproblem after each iteration. The i-th subproblem has a set of multipliers, each linked to another subproblem's complicating constraint that includes an optimization variable of the i-th subproblem. These multipliers are assigned a seed value and updated (after the subproblems are solved) using the errors in the corresponding complicating constraints. This procedure is repeated in an iterative manner until the required degree of accuracy is attained.

3. Problem statement

The problem at stake consists in deriving a control approach based on (2) that guarantees the convergence and stability of the solution and whose implementation is suitable for large-scale systems. Indeed, (2) is formulated in a centralized manner, i.e., the overall problem is solved by a single decision unit, a strategy that might suffer from implementation issues when applied to large-scale systems. Therefore, it is desirable to reformulate the problem using a non-centralized approach. These can be classified into two main groups according to the availability of information and interactions among local controllers:

- Decentralized control approaches solve each subproblem by ignoring the interactions among subsystems. Depending on the existing degree of coupling among subsystems, such strategies might lead to a poor overall performance [37].

- Distributed control approaches, on the other hand, take explicitly into account the effects of local actions at the systemwide level. Indeed, the developments in information and communication technologies enable the exchange of information among local controllers, thus allowing for cooperation and negotiation with the aim to achieve the best global performance [38]. These problems are solved in an iterative manner until the desired level of convergence is achieved, thus prioritizing the optimality of the results over computation time.

Given the fact that the performances offered by distributed approaches are closer to the optimal centralized performance than those offered by decentralized ones, the former will be considered. However, distributed control methods require a number of decision units, each solving a local optimization problem (subproblem), as well as a mechanism that allows to coordinate the solutions of the subproblems. Indeed, partitioning the overall problem into subproblems generally leads to couplings in the optimization variables, i.e., variables that appear in more than one subproblem. Therefore, it is desirable that the local controllers are able to share their solution with each other, which allows to solve again their local problem with updated information. This iterative process is repeated until the desired level of convergence for the coupled variables is achieved. A number of coordination methods based on the Lagrangian relaxation approach introduced in Section 2.2 have been reported in the framework of distributed control.

The steps required to solve the considered problem are outlined below:

- Divide the overall system into a set of smaller subproblems. The resulting couplings among subproblems define the required information exchanges.

- Devise a strategy that allows for the coordination of the coupled subproblems. This strategy will be based on the Lagrangian relaxation approach presented in Section 2.2.

- Synthesize controllers in a distributed manner using the concepts introduced in Section 2.1.

The convergence and stability properties will also be analyzed to determine the applicability of the proposed methodology.

4. Proposed approach

4.1. Optimality condition decomposition

The OCD can be considered as a particular implementation of the Lagrangian relaxation method, and seeks to decompose a problem into a set of smaller subproblems and coordinate them using the optimality conditions of the overall problem [39].

The overall problem (5) is restated for convenience:

$$\min_{\mathbf{z}} \; f(\mathbf{z}) \tag{9a}$$

subject to

$$\mathbf{b}(\mathbf{z}) \leq 0. \tag{9b}$$

Remark 3. For the sake of simplicity, only the inequality constraints are preserved in the formulation, as the case with equality constraints can be dealt with in a similar manner [39].

The OCD assumes that (9) can be decomposed into a set of smaller subproblems, which can be expressed by conveniently reformulating the overall problem as follows:

$$\min_{\{\mathbf{z}^{(i)}\}_{i=1}^{N}} \sum_{i=1}^{N} f^{(i)}\left(\mathbf{z}^{(i)}\right) \tag{10a}$$

subject to

$$\mathbf{h}\left(\mathbf{z}^{(1)},...,\mathbf{z}^{(N)}\right) \leq \mathbf{0}, \tag{10b}$$

$$\mathbf{g}^{(i)}\left(\mathbf{z}^{(i)}\right) \leq \mathbf{0}, \qquad i \in \{1,...,N\}, \tag{10c}$$

where N is the number of subproblems that can be identified and $\mathbf{z}^{(i)}$ denotes the set of variables that belong to the i-th subproblem. Moreover, (10b) constitutes the set of complicating constraints, as it contains variables from several subproblems. Indeed, if these constraints were removed from (10), the overall problem would easily decompose into N subproblems that could be solved independently.

Remark 4. The OCD requires the cost function to be separable in groups of variables.

A relaxed version of (10) is given by

$$\min_{\{\mathbf{z}^{(i)}\}_{i=1}^{N}} \sum_{i=1}^{N} f^{(i)}\left(\mathbf{z}^{(i)}\right) + \sum_{i=1}^{N} \boldsymbol{\lambda}^{(i)}\mathbf{h}^{(i)}\left(\mathbf{z}^{(1)},...,\mathbf{z}^{(N)}\right) \tag{11a}$$

subject to

$$\mathbf{h}^{(i)}\left(\mathbf{z}^{(1)},...,\mathbf{z}^{(N)}\right) \leq \mathbf{0}, \qquad i \in \{1,...,N\}, \tag{11b}$$

$$\mathbf{g}^{(i)}\left(\mathbf{z}^{(i)}\right) \leq \mathbf{0}, \qquad i \in \{1,...,N\}. \tag{11c}$$

Remark 5. The solution remains unaffected regardless of how the complicating constraints are distributed among the N subproblems.

Then, the relaxed problem (11) decomposes into N subproblems after fixing the values of the variables that pertain to other subproblems. Moreover, the i-th subproblem is formulated as follows:

$$\min_{\mathbf{z}^{(i)}} \quad f^{(i)}\left(\mathbf{z}^{(i)}\right) + \sum_{j=1, j\neq i}^{N} f^{(j)}\left(\bar{\mathbf{z}}^{(j)}\right) \tag{12a}$$

$$+ \sum_{j=1, j\neq i}^{N} \bar{\boldsymbol{\lambda}}^{(j)} \mathbf{h}^{(j)}\left(\bar{\mathbf{z}}^{(1)}, ..., \bar{\mathbf{z}}^{(i-1)}, \mathbf{z}^{(i)}, \bar{\mathbf{z}}^{(i+1)}, ..., \bar{\mathbf{z}}^{(N)}\right)$$

subject to

$$\mathbf{h}^{(i)}\left(\bar{\mathbf{z}}^{(1)}, ..., \bar{\mathbf{z}}^{(i-1)}, \mathbf{z}^{(i)}, \bar{\mathbf{z}}^{(i+1)}, ..., \bar{\mathbf{z}}^{(N)}\right) \leq \mathbf{0}, \tag{12b}$$

$$\mathbf{g}^{(i)}\left(\mathbf{z}^{(i)}\right) \leq \mathbf{0}, \tag{12c}$$

where the overlined variables indicate fixed values.

These subproblems are obtained when the complicating constraints of the j-th subproblem (which involve variables from the i-th block) are relaxed in the i-th subproblem, while the complicating constraints in the i-th subproblem are kept as constraints. Then, the coordination of the subproblems to satisfy the complicating constraints is straightforward since these are included in the subproblems, and can be achieved by updating the multipliers as

$$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \alpha \mathbf{h}_i, \; i \in \{1, ..., N\}, \tag{13}$$

where α is a suitable update constant and $\mathbf{h}_i$ is evaluated using the last solution.

An interesting feature of the OCD is the fact that the decomposition step not only yields the set of subproblems, but also dictates the coordination policy that needs to be applied. Indeed, the resulting subproblems might be independent (no couplings), which would allow for fully decentralized control architectures; hierarchically structured, i.e., lower block triangular decompositions [37], which would benefit from a hierarchical control approach; or interconnected, which would require a distributed control approach. Hence, the degree of communication and coordination among subproblems depends on each case.

The name of the method comes from the fact that the overall problem can be decomposed by manipulating the first-order KKT optimality conditions of the overall problem [35]:

$$\nabla_{\mathbf{z}^{(i)}} f_i\left(\mathbf{z}_*^{(i)}\right) + \sum_{i=1}^{N} \nabla_{\mathbf{z}^{(i)}}^{\intercal} \mathbf{h}^{(i)}\left(\mathbf{z}_*^{(1)}, ..., \mathbf{z}_*^{(N)}\right) \boldsymbol{\lambda}_*^{(i)} \tag{14a}$$

$$+ \sum_{i=1}^{N} \nabla_{\mathbf{z}^{(i)}}^{\intercal} \mathbf{g}^{(i)}\left(\mathbf{z}_*^{(i)}\right) \boldsymbol{\nu}_*^{(i)} = 0, \qquad i \in \{1, ..., N\},$$

$$\mathbf{h}^{(i)}\left(\mathbf{z}_*^{(1)}, ..., \mathbf{z}_*^{(N)}\right) \leq \mathbf{0}, \qquad i \in \{1, ..., N\}, \tag{14b}$$

$$\left(\mathbf{h}^{(i)}\left(\mathbf{z}_*^{(1)}, ..., \mathbf{z}_*^{(N)}\right)\right)^{\intercal} \boldsymbol{\lambda}_*^{(i)} = 0, \qquad i \in \{1, ..., N\}, \tag{14c}$$

$$\boldsymbol{\lambda}_*^{(i)} \geq \mathbf{0}, \qquad i \in \{1, ..., N\}, \tag{14d}$$

$$\mathbf{g}^{(i)}\left(\mathbf{z}_*^{(i)}\right) \leq \mathbf{0}, \qquad i \in \{1, ..., N\}, \tag{14e}$$

$$\left(\mathbf{g}^{(i)}\left(\mathbf{z}_*^{(i)}\right)\right)^{\intercal} \boldsymbol{\nu}_*^{(i)} = 0, \qquad i \in \{1, ..., N\}, \tag{14f}$$

$$\boldsymbol{\nu}_*^{(i)} \geq \mathbf{0}, \qquad i \in \{1, ..., N\}, \tag{14g}$$

where $_*$ denotes the optimal value, and $\boldsymbol{\lambda}_*^{(i)}$ and $\boldsymbol{\nu}_*^{(i)}$ are associated to (10b) and (10c), respectively.

Given the above considerations, the OCD can be viewed as a partitioning and coordination strategy that relies on the formulation and convenient manipulation of the matrix of KKT conditions associated to the overall problem to determine a set of smaller subproblems. However, the OCD addresses neither the block identification task nor the manner to obtain the optimal partitioning. Therefore, the use of a strategy that yields the optimal partitioning and complements the OCD is proposed hereunder.

4.2. Optimal partitioning using community detection

The matrix of KKT conditions can be equivalently expressed using the associated graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes, i.e., the elements of the system, and $\mathcal{E}$ is the set of edges, i.e., the set of arcs that connect the nodes. In the context of a control problem, the states, inputs and outputs of the system can be viewed as nodes, whereas the edges are the equations that link the variables.

Therefore, the problem of partitioning the matrix of KKT conditions can be addressed using methods stemming from graph theory and network science. The approach followed in this work is based on community detection, a problem that seeks to determine communities, i.e., groups of nodes, within complex systems, such that its nodes are significantly more coupled to nodes in the same community than nodes belonging to other communities [40]. The notion of modularity, which measures the difference defined by the previous and rather vague description, is formally defined as follows:

$$M = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta \left(c_i, c_j \right), \tag{15}$$

where M is the modularity of the resulting partitioning, m is the total weight of all edges in the graph, A_{ij} is the weight of the edge that links the i-th and j-th nodes, k_i and k_j are the total weights of the edges that link the i-th and j-th nodes with the rest of the network, respectively, c_i and c_i are the communities to which the i-th and j-th nodes are assigned, respectively, and $\delta \left(c_i, c_j \right)$ is the Kronecker delta defined as

$$\delta \left(c_i, c_j \right) = \begin{cases} 1 \text{ if the } i\text{-th and } j\text{-th nodes belong to the same community,} \\ 0 \text{ otherwise.} \end{cases} \tag{16}$$

Therefore, system partitioning in the community detection framework is formulated as a modularity maximization problem. Although this is generally an NP-hard integer program, several efficient algorithms with performances close to the optimal one have been proposed [10]. The algorithm applied in this paper is known as fast unfolding [41], which consists in a two-phase approach that is repeated in an iterative manner until the partitioning cannot be improved. Given a weighted network, i.e., a graph in which each edge is assigned a value, the steps performed by the fast unfolding algorithm are summarized below:

- The initial partition is such that each node is assigned a different community.

- The first step is known as *modularity optimization*, and consists in computing the gain of modularity that would result from placing the i-th node in a neighboring community, i.e., a community to which the i-th node is linked:

$$\Delta M = \left(\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right) - \left(\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right), \tag{17}$$

  where $\sum_{in}$ is the sum of the weights of the edges of the destination community, $\sum_{tot}$ is the sum of the weights of the edges incident to the destination community, k_i is the sum of the weights of the edges incident to the i-th node, $k_{i,in}$ is the sum of the weights of the edges from the i-th node to the destination community and m is the total sum of the weights of all network edges.

- The second step, also referred to as *community aggregation*, consists in adding the i-th node to the neighboring community for which the maximal positive ΔM is obtained. If such situation is not encountered, it stays in the original community. This is done for each node in the network, and concludes the first iteration of the fast unfolding algorithm. Note also that the same node might be considered more than once throughout the process.

- To prepare the algorithm for the second iteration, the resulting network is built, where the nodes are now the communities obtained in the previous iteration. Moreover, the weights of the edges among communities are updated adding the sum of individual edges. Note also that this procedure creates self-loops.

Figure 1 illustrates the described approach. Note that the community detection problem is solved in only one iteration for this particular network, as a second iteration reveals $\Delta M < 0$ for each node in Fig. 1(c). Moreover, nodes

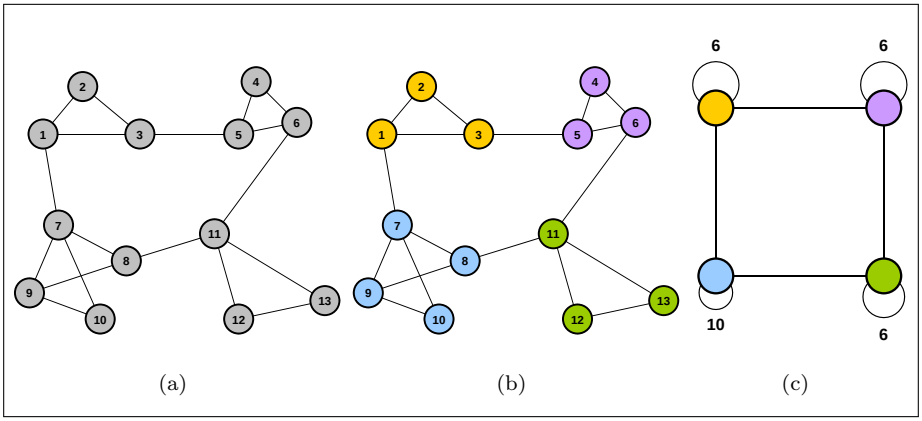

Figure 1: (a) Original network (each node belongs to a different community). (b) Original network after modularity optimization. (c) Resulting network after community aggregation.

are labeled in Figs. 1(a) and 1(b) for the purpose of identification, while the weights of the edges represent the number of connections between a pair of nodes (which might be variables or communities), and the weight of an unlabeled edge equals one.

As stated before, the fast unfolding algorithm is to be applied to the graph associated to the matrix of KKT conditions, which allows to take into account the couplings introduced by the constraints and the cost function. As a result, this approach reveals communities of system variables, probably allowing for physical interpretation. Nevertheless, it must be observed that there exists a matrix of KKT conditions for every time instant. Although it might seem necessary to apply the algorithm at every instant, in practice it is only required to repeat the partitioning procedure in the event of a change in the system, as the links among variables might be different.

The complete partitioning approach based on OCD and community detection is illustrated by means of Fig. 2. Note that the original problem can be expressed using the matrix of KKT conditions. Then, its associated graph is analyzed using community detection, yielding a rearranged matrix of KKT conditions that allows to obtain a problem that decomposes into smaller sub-

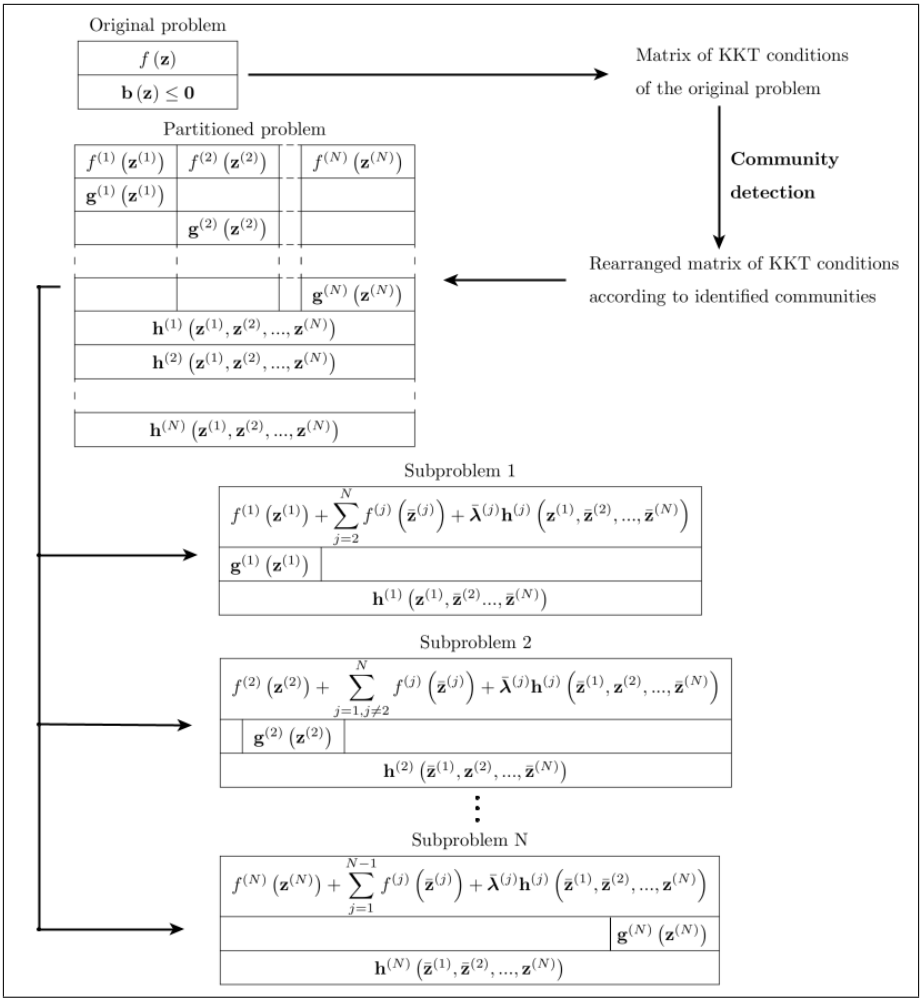

Figure 2: Overview of the OCD using community detection

problems. In turn, these subproblems are solved in parallel, thus providing an approximate solution to the original problem.

4.3. The DMPC-OCD

The last step towards the final solution consists in incorporating the coordination mechanism into the formulation of the subproblems, thus tackling the general case in which the subproblems are coupled. This can be achieved by combining the CMPC formulation and the coordination ideas based on OCD

and community detection. Moreover, note that the original OCD formulation is extended, as it lacks the temporal component inherent to the control framework.

With all this, the l-th control subproblem reads as follows:

$$\min_{\substack{\left\{\mathbf{u}_{i|k}^{(l)}\right\}_{i=k}^{k+H_p-1},\\ \left\{\mathbf{x}_{i|k}^{(l)}\right\}_{i=k}^{k+H_p}}} J^{(l)}\left(\mathbf{u}_{i|k}^{(l)}, \mathbf{x}_{i|k}^{(l)}\right) + \left\|\mathbf{x}_{k+H_p|k}^{(l)}\right\|_{\mathbf{Q}^{(l)}}^2 + \tag{18a}$$

$$\sum_{\substack{m\neq l,\\ \mathcal{V}^{(m)}\cap\mathcal{V}^{(l)}\neq\emptyset}} \boldsymbol{\lambda}^{(m)}\left(J^{(m)}\left(\mathbf{u}_{i|k}^{(m)}, \mathbf{x}_{i|k}^{(m)}\right) + \left(\mathbf{x}_{i+1|k}^{(m)} - \mathbf{A}^{(m)}\mathbf{x}_{i|k}^{(m)} - \mathbf{B}^{(m)}\mathbf{u}_{i|k}^{(m)}\right)\right)$$

subject to

$$\mathbf{x}_{i+1|k}^{(l)} = \mathbf{A}^{(l)}\mathbf{x}_{i|k}^{(l)} + \mathbf{B}^{(l)}\mathbf{u}_{i|k}^{(l)},\ i \in \{k, ..., k+H_p-1\}, \tag{18b}$$

$$\mathbf{u}_{i|k}^{(l)} \in \mathcal{U}^{(l)},\ i \in \{k, ..., k+H_p-1\}, \tag{18c}$$

$$\mathbf{x}_{i|k}^{(l)} \in \mathcal{X}^{(l)},\ i \in \{k, ..., k+H_p-1\}, \tag{18d}$$

$$\mathbf{x}_{k+H_p|k}^{(l)} \in \mathcal{X}_{H_p}^{(l)}, \tag{18e}$$

$$\mathbf{x}_{k|k}^{(l)} = \mathbf{x}_k^{(l)}, \tag{18f}$$

where the superscripts l and m denote information relative to the l-th and m-th subproblems, respectively, $\mathbf{Q}^{(l)}$ is an appropriate submatrix of $\mathbf{Q}$ and can be obtained according to the partitioning, and $\mathcal{V}^{(l)}$ and $\mathcal{V}^{(m)}$ represent the sets of the graph nodes (system variables) associated to the l-th and m-th subproblems, respectively. Note that the l-th and m-th subproblems can only be coupled in the variables through the state equation, which is stated by means of the condition $\mathcal{V}^{(m)} \cap \mathcal{V}^{(l)} \neq \emptyset$. Thus, these are the sole constraints that must be relaxed.

Remark 6. The decision variables in (18) are denoted with the superscript l. Conversely, the variables denoted with the superscript m are optimized in the m-th subproblem and treated as parameters in the l-th subproblem.

4.4. Convergence and stability of the DMPC-OCD

4.4.1. Convergence

The convergence properties of the proposed method are discussed following the derivation reported in [39]. Note that separable constraints are not taken into account at this step, as they do not have an influence on the decomposition procedure.

The search direction strategy can be defined, in the context of an optimization process, as an iterative approach that aims at finding a local minimum of a cost function. Then, the search directions $\boldsymbol{\Delta}^{(l)}$, $l \in \{1, ..., N\}$, can be computed in a centralized manner by solving

$$\underbrace{\begin{bmatrix} \mathbf{KKT}^{(1,1)} & \mathbf{KKT}^{(1,2)} & \cdots & \mathbf{KKT}^{(1,N)} \\ \mathbf{KKT}^{(2,1)} & \mathbf{KKT}^{(2,2)} & \cdots & \mathbf{KKT}^{(2,N)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{KKT}^{(N,1)} & \mathbf{KKT}^{(N,2)} & \cdots & \mathbf{KKT}^{(N,N)} \end{bmatrix}}_{\mathbf{KKT}} \begin{bmatrix} \boldsymbol{\Delta}^{(1)} \\ \boldsymbol{\Delta}^{(2)} \\ \vdots \\ \boldsymbol{\Delta}^{(N)} \end{bmatrix} = - \begin{bmatrix} \nabla_{\mathbf{z}^{(1)}, \boldsymbol{\lambda}^{(1)}} \mathcal{L} \\ \nabla_{\mathbf{z}^{(2)}, \boldsymbol{\lambda}^{(2)}} \mathcal{L} \\ \vdots \\ \nabla_{\mathbf{z}^{(N)}, \boldsymbol{\lambda}^{(N)}} \mathcal{L} \end{bmatrix}, \tag{19}$$

where $\mathbf{KKT}$ is the matrix of KKT conditions, $\boldsymbol{\Delta}^{(l)} = \left(\Delta \mathbf{z}^{(l)}, \Delta \boldsymbol{\lambda}^{(l)}\right)$, $l \in \{1, ..., N\}$, $\mathcal{L}$ is the Lagrangian function for the overall problem and the several blocks within $\mathbf{KKT}$ are the Newton matrices of the form [34]

$$\mathbf{KKT}^{(l,l)} = \begin{bmatrix} \nabla^2_{\mathbf{z}^{(l)} \mathbf{z}^{(l)}} \mathcal{L} & \left(\nabla_{\mathbf{z}^{(l)}} \mathbf{h}^{(l)}\right)^\intercal \\ \nabla_{\mathbf{z}^{(l)}} \mathbf{h}^{(l)} & \mathbf{0} \end{bmatrix}, \tag{20a}$$

$$\mathbf{KKT}^{(l,m)} = \begin{bmatrix} \nabla^2_{\mathbf{z}^{(l)} \mathbf{z}^{(m)}} \mathcal{L} & \left(\nabla_{\mathbf{z}^{(m)}} \mathbf{h}^{(l)}\right)^\intercal \\ \nabla_{\mathbf{z}^{(l)}} \mathbf{h}^{(m)} & \mathbf{0} \end{bmatrix}, \tag{20b}$$

$$\mathbf{KKT}^{(m,l)} = \left(\mathbf{KKT}^{(l,m)}\right)^\intercal. \tag{20c}$$

However, the approach followed by the OCD consists in solving a decompos-

able and approximate version of (19) that can be formulated as

$$\underbrace{\begin{bmatrix} \mathbf{KKT}^{(1,1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{KKT}^{(2,2)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{KKT}^{(N,N)} \end{bmatrix}}_{\overline{\mathbf{KKT}}} \begin{bmatrix} \boldsymbol{\Delta}^{(1)} \\ \boldsymbol{\Delta}^{(2)} \\ \vdots \\ \boldsymbol{\Delta}^{(N)} \end{bmatrix} = - \begin{bmatrix} \nabla_{\mathbf{z}^{(1)},\boldsymbol{\lambda}^{(1)}} \mathcal{L} \\ \nabla_{\mathbf{z}^{(2)},\boldsymbol{\lambda}^{(2)}} \mathcal{L} \\ \vdots \\ \nabla_{\mathbf{z}^{(N)},\boldsymbol{\lambda}^{(N)}} \mathcal{L} \end{bmatrix} \tag{21}$$

and the $\mathbf{KKT}^{(l,l)}$ blocks within $\overline{\mathbf{KKT}}$ are defined as in (20a).

Theorem 1. *Let* $\mathbf{KKT}$ *and* $\overline{\mathbf{KKT}}$ *be the centralized matrix of KKT conditions and its approximate version, respectively. Then, the sufficient condition for convergence of the solution provided by the OCD reads as*

$$\rho\Big(\mathbf{I} - \overline{\mathbf{KKT}}^{-1}\mathbf{KKT}\Big) < 1, \tag{22}$$

where $\rho(\cdot)$ *denotes the spectral radius of the matrix evaluated at the solution.*

Proof. See [42, Chapter 4.2.1] for the proof of general convergence results yielded by iterative methods that are employed to solve large linear systems. □

Remark 7. The convergence condition (22) must be checked for every matrix of KKT conditions, as follows from the statement in Section 4.2.

4.4.2. Stability

The terminal constraint set (3) must be designed to be structured [16], i.e., the following additional property needs to be taken into account in its design:

$$\mathbf{x}_{k+H_p|k}^{\intercal} \mathbf{Q} \mathbf{x}_{k+H_p|k} = \sum_{l=1}^{N} \left(\mathbf{x}_{k+H_p|k}^{(l)}\right)^{\intercal} \mathbf{Q}^{(l)} \mathbf{x}_{k+H_p|k}^{(l)}, \tag{23}$$

and $\mathbf{Q}^{(l)}$ must be positive semi-definite, $l \in \{1, ..., N\}$.

The desired structure for $\mathcal{X}_{H_p}$ is attained by considering local invariant sets of the form [30]

$$\mathcal{X}_{H_p}^{(l)} = \left\{ \mathbf{x}_{k+H_p|k}^{(l)} \in \mathbb{R}^{n_{x^{(l)}}} \;\middle|\; \left(\mathbf{x}_{k+H_p|k}^{(l)}\right)^{\intercal} \mathbf{Q}^{(l)} \mathbf{x}_{k+H_p|k}^{(l)} \leq \beta^{(l)} \right\}, \tag{24}$$

which implies that the condition $\mathbf{x}_{k+H_p|k} \in \mathcal{X}_{H_p}$ is equivalent to

$$\forall i \in \{1, ..., N\} \;\; \exists \beta^{(l)} \geq 0 \; : \mathbf{x}_{k+H_p|k}^{(l)} \in \mathcal{X}_{H_p}^{(l)}, \sum_{l=1}^{N} \beta^{(l)} \leq 1. \tag{25}$$

Then, $\mathbf{Q}$ can be synthesized from the local $\mathbf{Q}^{(l)}$ matrices by solving an additional optimization problem [30, Eq. (23)]. In short, this approach employs the decomposition of the overall problem, which has already been performed, to obtain the $\mathbf{Q}^{(l)}$ matrices, one for each subproblem. Then, the overall matrix $\mathbf{Q}$ can be built by conveniently stacking $\mathbf{Q}^{(l)}$, and its final structure is such that it decomposes in the already defined block matrices $\mathbf{Q}^{(l)}$. Note that this approach allows to obtain both the terminal constraint set and terminal cost.

4.5. Summary of the DMPC-OCD approach

Certain steps comprised in the proposed DMPC-OCD approach may be computed offline, while the simulation of the control problem is performed online. Algorithms 1 and 2 summarize the offline and online parts, respectively. Note that the states obtained in Algorithm 2 after applying $\mathbf{u}_k^{MPC\ (l)}$ allow to resume the simulation at the next time instant. On the other hand, the stop criterion is usually formulated as a maximum admissible error that the complicating constraints should not violate. Naturally, the required degree of accuracy of the solutions has an impact on the number of iterations that need to be performed.

Remark 8. Algorithms 1 and 2 are formulated assuming that the system conditions do not change throughout the simulation, which allows for an offline overall problem decomposition and computation of the terminal constraint set and terminal cost. The noncompliance of this scenario would require to perform these steps online.

Algorithm 1 Offline calculation of the terminal costs

1: Formulate the overall control problem (2)
2: Decompose (2) using the community detection algorithm applied to the KKT conditions formulated for the overall problem (Fig. 2)
3: Determine the terminal constraint set and the terminal cost taking into account the decomposed problem obtained in step 2 (Section 4.4.1)

Algorithm 2 Online DMPC-OCD

Require: Set of subproblems (18)

1: Initialize states and Lagrange multipliers in all subproblems
2: **for** $k = 1 : t_{sim}$ **do**
3: Perform one iteration for each subproblem
4: **while** the stop criterion is not met **do**
5: Exchange last solution among coupled subproblems
6: Update Lagrange multipliers using (13)
7: Perform a new iteration for each subproblem
8: **end while**
9: Set $\mathbf{u}_k^{MPC\ (l)} \triangleq \mathbf{u}_{k|k}^{(l)}$ and apply it, $l \in \{1, ..., N\}$
10: Obtain $\mathbf{x}_{k+1}^{(l)}$, with $l = 1, ..., N$
11: **end for**

5. Case study

The effectiveness of the proposed approach is tested on two different systems, the quadruple-tank system and the Barcelona DWN. One the one hand, considering the quadruple tank system allows to make use of a well-established process for which several benchmarks exist, thus making it possible to compare and validate the results. On the other hand, the Barcelona DWN is a large-scale system that has also been considered to test control approaches in the last years.

Both case studies are described next, each accompanied by the obtained results and corresponding analysis. Moreover, their implementation is carried out in MATLAB, using YALMIP [43] as parser.

5.1. The quadruple-tank system

The quadruple-tank system, depicted in Fig. 3, is a multivariable process reported in detail in [44] and widely employed for the purpose of control education. Although not strictly a large-scale system, the quadruple-tank system possesses a number of features that makes it a very suitable case study in the context of this work. On the one hand, it is a highly coupled system, which allows to test the derived system partitioning and coordination strategy. On the other hand, the system states are directly measurable, which eliminates the need to use an observer in combination with the controller, thus simplifying the design task. Last but not least, the same benchmark was used in [45] to test the performance of several predictive control strategies, with which the DMPC-OCD can be compared by considering the same physical parameters and control objectives.

5.1.1. System description

The main operational goal of the quadruple-tank process consists in steering the water levels of the two lower tanks to the setpoints by conveniently adjusting the voltages applied to two pumps.

The dynamics of the system can be described by

$$\frac{dh_1}{dt} = -\frac{a_1}{S}\sqrt{2gh_1} + \frac{a_3}{S}\sqrt{2gh_3} + \frac{\gamma_a}{S}q_a, \tag{26a}$$

$$\frac{dh_2}{dt} = -\frac{a_2}{S}\sqrt{2gh_2} + \frac{a_4}{S}\sqrt{2gh_4} + \frac{\gamma_b}{S}q_b, \tag{26b}$$

$$\frac{dh_3}{dt} = -\frac{a_3}{S}\sqrt{2gh_3} + \frac{1-\gamma_b}{S}q_b, \tag{26c}$$

$$\frac{dh_4}{dt} = -\frac{a_4}{S}\sqrt{2gh_4} + \frac{1-\gamma_a}{S}q_a, \tag{26d}$$

where h_i $[m]$ and a_i $[m^2]$ denote the water level and the discharge constant of the i-th tank, respectively, with $i \in \{1, 2, 3, 4\}$, S $[m^2]$ is the cross section of the tanks, q_j $[m^3h^{-1}]$ and γ_j (dimensionless) indicate the flow and the ratio of the three-way valve associated to the j-th pump, with $j \in \{a, b\}$, respectively,

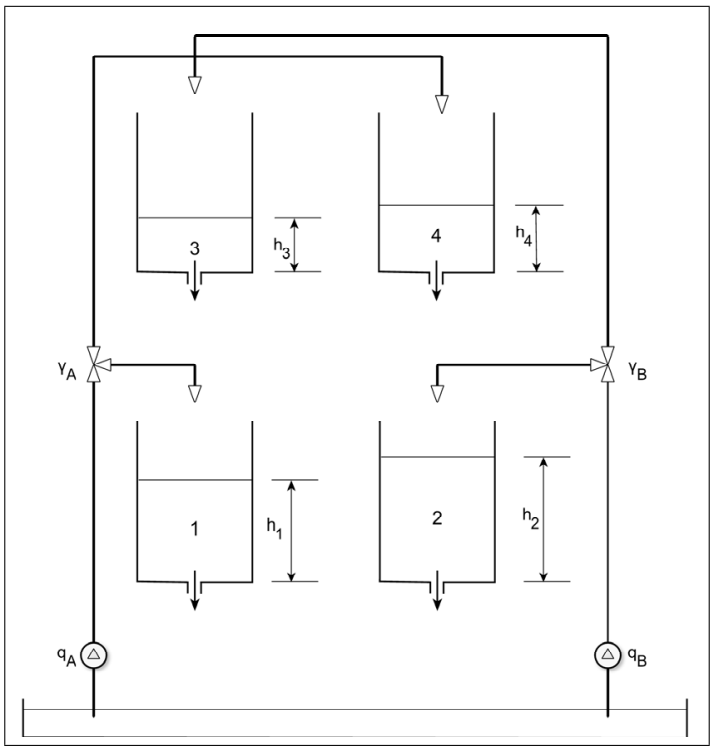

Figure 3: Schematic representation of the quadruple-tank system

and g $[ms^{-2}]$ is the acceleration of gravity. As stated before, the values of all parameters are as in [45].

Equation (26) is used as the simulation model. On the other hand, the DMPC-OCD is tested over a linear prediction model that can be obtained by linearizing (26) around the equilibrium levels given in Table 1 in [45]. Note that the linear prediction model is expressed in terms of deviation variables around the operating point as follows:

$$x_i = h_i - h_i^0,\ i \in \{1,2,3,4\} \tag{27a}$$

$$u_1 = q_a - q_a^0, \tag{27b}$$

$$u_2 = q_b - q_b^0. \tag{27c}$$

Then, the prediction model reads as

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}_c\mathbf{x} + \mathbf{B}_c\mathbf{u}, \tag{28a}$$

$$\mathbf{y} = \mathbf{C}_c\mathbf{x}, \tag{28b}$$

with $\mathbf{x} = [x_1\ x_2\ x_3\ x_4]^\intercal$, $\mathbf{u} = [u_1\ u_2]^\intercal$, $\mathbf{y} = [x_1\ x_2]^\intercal$ and

$$\mathbf{A}_c = \begin{bmatrix} \frac{-1}{\tau_1} & 0 & \frac{1}{\tau_3} & 0 \\ 0 & \frac{-1}{\tau_2} & 0 & \frac{1}{\tau_4} \\ 0 & 0 & \frac{-1}{\tau_3} & 0 \\ 0 & 0 & 0 & \frac{-1}{\tau_4} \end{bmatrix}, \mathbf{B}_c = \begin{bmatrix} \frac{\gamma_a}{S} & 0 \\ 0 & \frac{\gamma_b}{S} \\ 0 & \frac{1-\gamma_b}{S} \\ \frac{1-\gamma_a}{S} & 0 \end{bmatrix}, \quad \mathbf{C}_c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \tag{29}$$

where $\tau_i = S/\left(a_i\sqrt{2h_i^0/g}\right)$ $[s]$ is the time constant of the i-th tank. This continuous-time prediction model is discretized with a sampling time $T_s = 5$ s.

5.1.2. Experimental design

As stated before, the control strategy aims at maintaining the water levels of the lower tanks as close as possible to the desired values. Fig. 4 depicts the references considered, which are designed to test how the system responds to setpoint changes that involve admissible values that are relatively far from one another. Moreover, it can be observed that the initial setpoints correspond to the linearization values considered in the computation of the prediction model.

On the other hand, the following performance index is considered:

$$\begin{aligned} J = &\sum_{k=1}^{t_{sim}} \big(h_1(k) - s_1(k)\big)^2 + \big(h_2(k) - s_2(k)\big)^2 + \\ &\sum_{k=1}^{t_{sim}} 0.01\big(q_a(k) - q_a^s(k)\big)^2 + 0.01\big(q_b(k) - q_b^s(k)\big)^2, \end{aligned} \tag{30}$$

where q_a^s and q_b^s correspond to the values of the controlled inputs and are computed for the setpoints s_1 and s_2 in steady-state conditions.

Remark 9. The computation of the performance index (30) begins once the operating point is reached, i.e., at $t = 2700$ s.

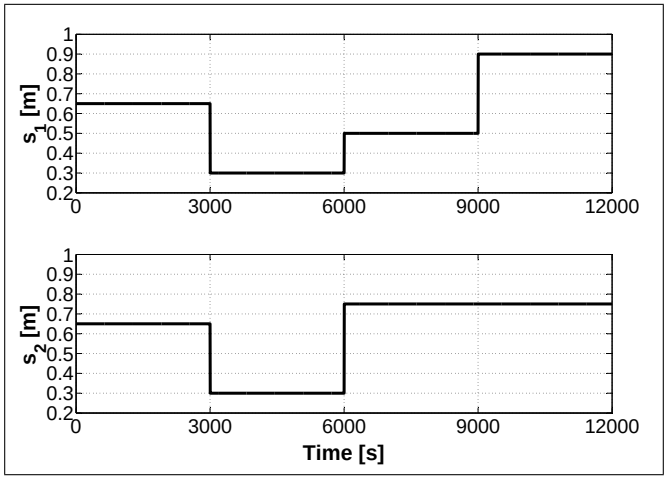

Figure 4: Reference water levels for tanks 1 and 2

5.1.3. Results

The CMPC associated to the quadruple-tank system can be formulated as follows:

$$\min_{\{\mathbf{u}_{i|k}\}_{i=k}^{k+H_p-1},\, \{\mathbf{x}_{i|k}\}_{i=k}^{k+H_p}} J\left(\mathbf{u}_{i|k}, \mathbf{x}_{i|k}\right) \tag{31a}$$

subject to

$$\mathbf{x}_{i+1|k} = \mathbf{A}\mathbf{x}_{i|k} + \mathbf{B}\mathbf{u}_{i|k},\ i \in \{k, ..., k+H_p-1\}, \tag{31b}$$

$$\mathbf{u}_{i|k} \in \mathcal{U},\ i \in \{k, ..., k+H_p-1\}, \tag{31c}$$

$$\mathbf{x}_{i|k} \in \mathcal{X},\ i \in \{k, ..., k+H_p-1\}, \tag{31d}$$

$$\mathbf{x}_{k+H_p|k} \in \mathcal{X}_{H_p}, \tag{31e}$$

$$\mathbf{x}_{k|k} = \mathbf{x}_k, \tag{31f}$$

with

$$J\left(\mathbf{u}_{i|k}, \mathbf{x}_{i|k}\right) = \sum_{i=k}^{k+H_p-1} \left\|\mathbf{x}_{i|k}\right\|_{\mathbf{P}}^2 + \left\|\mathbf{u}_{i|k}\right\|_{\mathbf{R}}^2 + \left\|\mathbf{x}_{k+H_p|k}\right\|_{\mathbf{Q}}^2, \tag{32}$$

and $\mathcal{X}$ and $\mathcal{U}$ correspond to the admissible sets of states and inputs, respectively, and can be computed using the physical limits provided in [45]. Moreover, the cost function (31a) is adapted from (30) employing deviation variables, and the weighting matrices (of suitable dimensions) are: $\mathbf{P} = \mathbf{I}$, $\mathbf{R} = 0.01\mathbf{I}$ and $\mathbf{Q}$ is computed as indicated in Remark 1.

The KKT conditions associated to (31) can be formulated using (14). Note that (31a) is completely separable, i.e., its structure does not introduce any coupling among variables. Therefore, the graph associated to the matrix of KKT conditions is equivalent to the graph of the system, and can be constructed as shown in [27]. First, the interactions between variables and constraints can be modeled by means of the bipartite network depicted in Fig. 5(a). Two sets of nodes (variables and constraints) are considered, and there is an edge that connects a variable v_i and a constraint c_j provided that v_i appears in c_j. Note that the four constraints correspond to the prediction model, as the rest of constraints are not coupled. This graph can be simplified down to a variable unipartite graph, depicted in Fig. 5(b), that disregards the constraint nodes: the interactions between the different variable nodes are now described by edges whose weights are equal to the number of shared constraints among the pair of variables.

The same information can be expressed in a compact manner using the edge weight matrix

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}. \tag{33}$$

Then, the fast unfolding algorithm[1] is applied to the graph using (33). The

[1] `https://perso.uclouvain.be/vincent.blondel/research/louvain.html`

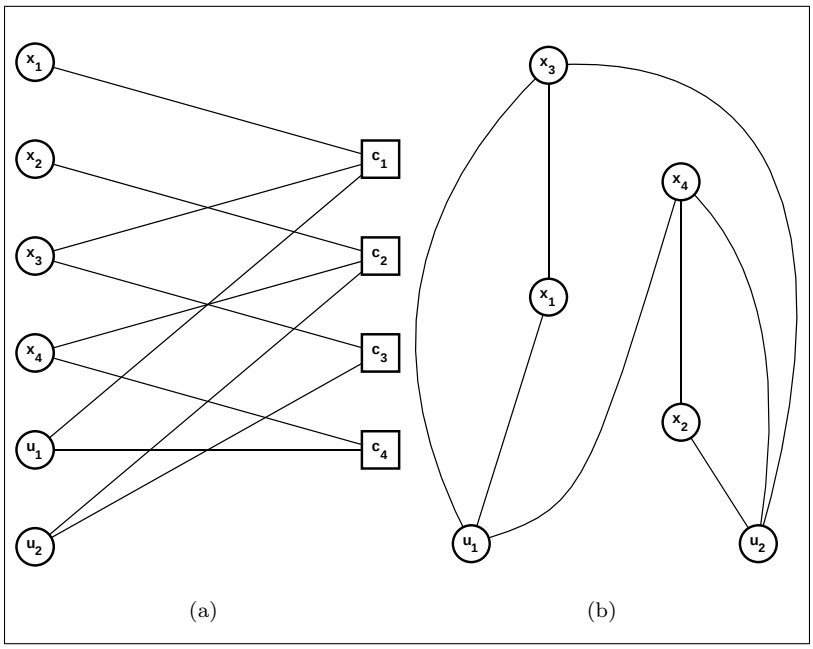

Figure 5: (a) Variable-constraint bipartite graph. (b) Equivalent variable unipartite graph.

results obtained after performing the steps described in Section 4.2 are illustrated in Fig. 6. A second iteration of the method shows that the partitioning cannot be further refined. Therefore, the analysis of the variables that belong to each community allows to conclude that the first subsystem comprises tanks 1 and 3, whereas subsystem 2 consists of tanks 2 and 4. Equivalently, $\mathbf{x}^{(1)} \triangleq \{x_1, x_3\}$ and $\mathbf{x}^{(2)} \triangleq \{x_2, x_4\}$. The same subsystems are identified in [45] using another approach, which allows to validate the system partitioning results. Note that the subsystems are coupled only through the inputs, and both of them have an effect on the dynamics of both subsystems.

The subproblems that result from the system partitioning can then be formulated as follows:

$$\min_{\substack{\left\{\mathbf{u}_{i|k}^{(l)}\right\}_{i=k}^{k+H_p-1}, \\ \left\{\mathbf{x}_{i|k}^{(l)}\right\}_{i=k}^{k+H_p}}} J^{(l)}\left(\mathbf{u}_{i|k}^{(l)}, \mathbf{x}_{i|k}^{(l)}\right) + \tag{34a}$$

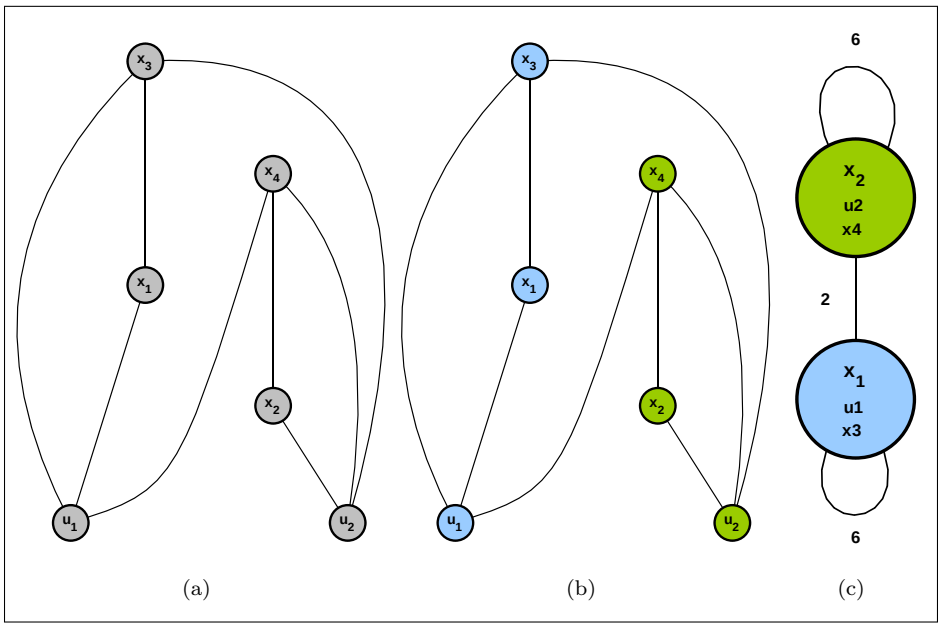

Figure 6: (a) Original graph associated to the quadruple-tank system. (b) Original graph after modularity optimization. (c) Resulting graph after community aggregation.

$$\boldsymbol{\lambda}^{(m)}\left(J^{(m)}\left(\mathbf{u}_{i|k}^{(m)},\mathbf{x}_{i|k}^{(m)}\right)+\left(\mathbf{x}_{i+1|k}^{(m)}-\mathbf{A}^{(m)}\mathbf{x}_{i|k}^{(m)}-\mathbf{B}^{(m)}\mathbf{u}_{i|k}^{(m)}\right)\right)$$

subject to

$$\mathbf{x}_{i+1|k}^{(l)}=\mathbf{A}^{(l)}\mathbf{x}_{i|k}^{(l)}+\mathbf{B}^{(l)}\mathbf{u}_{i|k}^{(l)},\ i\in\{k,...,k+H_p-1\}, \tag{34b}$$

$$\mathbf{u}_{i|k}^{(l)}\in\mathcal{U}^{(l)},\ i\in\{k,...,k+H_p-1\}, \tag{34c}$$

$$\mathbf{x}_{i|k}^{(l)}\in\mathcal{X}^{(l)},\ i\in\{k,...,k+H_p-1\}, \tag{34d}$$

$$\mathbf{x}_{k+H_p|k}^{(l)}\in\mathcal{X}_{H_p}^{(l)}, \tag{34e}$$

$$\mathbf{x}_{k|k}^{(l)}=\mathbf{x}_{k}^{(l)}, \tag{34f}$$

with

$$J^{(l)}\left(\mathbf{u}_{i|k}^{(l)},\mathbf{x}_{i|k}^{(l)}\right)=\sum_{i=k}^{k+H_p-1}\left\|\mathbf{x}_{i|k}^{(l)}\right\|_{\mathbf{P}^{(l)}}^{2}+\left\|\mathbf{u}_{i|k}^{(l)}\right\|_{\mathbf{R}^{(l)}}^{2}+\left\|\mathbf{x}_{k+H_p|k}^{(l)}\right\|_{\mathbf{Q}^{(l)}}^{2}. \tag{35}$$

In the context of this particular problem, $l = 1$ implies $m = 2$ and vice versa.

The solution of the DMPC-OCD is depicted in Fig. 7, showing that the proposed controlled strategy is able to keep the water levels close to the time-varying setpoints with no significant error. Moreover, the evolution of the levels of the upper tanks is also provided, although these are only required to stay within operating limits. Naturally, several iterations are required at every time instant to meet the stop criterion, which is formulated as

$$\sqrt{\left(\max\left(\left|\mathbf{h}_*^{(1)}\right|\right)\right)^2 + \left(\max\left(\left|\mathbf{h}_*^{(2)}\right|\right)\right)^2} \leq 10^{-9}, \tag{36}$$

where $\mathbf{h}_*^{(1)}$ and $\mathbf{h}_*^{(2)}$ correspond to the values of the complicating constraints (the subsystems' dynamics) after substituting the solutions. The satisfaction of (36) does not require more than four iterations in the context of this problem. On the other hand, the average computation time (model definition, conversion to solver-specific format and resolution) for the CMPC equals 0.024 seconds, while it takes 0.028 seconds in the case of the DMPC-OCD. It can then be seen how the distributed approach does not improve the computation time of the centralized approach. This is due to the fact that the quadruple-tank system is not a large-scale system, and therefore the sizes of the overall system and the subsystems are rather similar. The need for the DMPC-OCD approach to solve iteratively until convergence causes the increased computation time. Therefore, the benefits that the proposed approach can offer will become more evident in larger case studies such as the one considered in the next section.

Remark 10. The DMPC-OCD yields a vector of optimal values with length equal to H_p. The stop criterion (36) is formulated for the worst-case scenario as it considers the largest errors for both subsystems, even if these are not obtained at the same time instant within the prediction horizon.

On the other hand, the performance index (30) for the DMPC-OCD is computed, and equals 31.88. The same MPC is tested using a centralized implementation for the sake of comparison, yielding a performance index of 31.36.

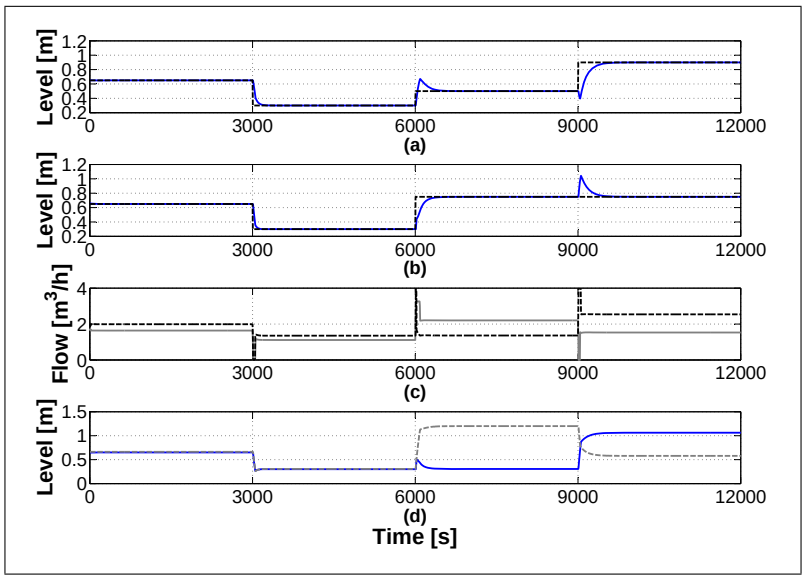

Figure 7: DMPC-OCD results: (a) y_1 (solid blue) and s_1 (dashed black). (b) y_2 (solid blue) and s_2 (dashed black). (c) q_a (solid gray) and q_b (dashed black). (d) h_3 (solid blue) and h_4 (dashed gray).

Thus, a rather small decrease of performance can be observed for the DMPC-OCD with respect to its centralized counterpart, probably due to the fact that the stop criterion allows for a small error at every time instant that adds up over time. Moreover, the performance index of the DMPC-OCD is similar to those reported in the benchmark. All these reasons allow to conclude that the proposed control approach performs as desired.

5.2. The Barcelona DWN

The Barcelona DWN has been used as case study in several research publications [4, 5, 46, 47]. In contrast to the quadruple-tank system, this is an example of a large-scale system, the kind of systems for which the proposed approach has been derived.

5.2.1. System description

Without going into detail, the Barcelona DWN consists of several water treatment plants that extract water from rivers and aquifers, treat it and supply

drinking water to the Barcelona metropolitan area. The reader is referred to [4, Section 2.1] for a more exhaustive description of the system.

The Barcelona DWN control-oriented model can be derived by considering the constitutive elements and their interactions:

- *Tanks* are the elements that allow to store water so that the demand may always be satisfied. The expression that describes the dynamics of the l-th tank reads as follows:

$$x_{k+1}^{(l)} = x_k^{(l)} + \Delta t \left(\sum_i q_k^{l,i} - \sum_j q_k^{l,j} \right), \tag{37}$$

  where $x_k^{(l)}$ is the volume of the l-th tank at time instant k, Δt is the sampling time, and $q_k^{l,i}$ and $q_k^{l,j}$ are the inflows and outflows of the l-th tank at time instant k, respectively. Note that these flows may correspond to manipulated flows and/or demands. Moreover, the constraint on the capacity of the l-th tank is given by

$$\underline{x}^{(l)} \leq x_k^{(l)} \leq \overline{x}^{(l)}, \tag{38}$$

  where $\underline{x}^{(l)}$ and $\overline{x}^{(l)}$ denote the minimum and maximum capacities of the l-th tank, respectively.

- Pumps and valves are used as *actuators* and allow to convey water to satisfy the demands. Their physical limits, which constrain the performance of the system, can be expressed as

$$\underline{u}^{(m)} \leq u_k^{(m)} \leq \overline{u}^{(m)}, \tag{39}$$

  where $u_k^{(m)}$ denotes the flow supplied by the m-th actuator at time instant k, and $\underline{u}^{(m)}$ and $\overline{u}^{(m)}$ are the minimum and maximum capacities of the m-th actuator, respectively.

- *Nodes* correspond to the network locations where water flow mergings and splittings occur. These mass balances determine the static system

behavior, and are incorporated into the model as equality constraints. Then, the mass balance at the n-th node can be expressed as

$$\sum_i q_k^{n,i} = \sum_j q_k^{n,j}, \tag{40}$$

where $q_k^{n,i}$ and $q_k^{n,j}$ are the inflows and outflows of the n-th node at time instant k, respectively. Again, note that q may denote both manipulated flows and demands.

- Consumer *demands* act as system disturbances. As daily and weekly trends can be detected in the data, approximate demands can be forecast using time series methods [48].

With all this, a discrete-time state-space network model can be written as follows:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{B}_d\mathbf{d}_k, \tag{41a}$$

$$\mathbf{0} = \mathbf{E}\mathbf{u}_k + \mathbf{E}_d\mathbf{d}_k, \tag{41b}$$

where $\mathbf{x}_k$, $\mathbf{u}_k$ and $\mathbf{d}_k$ denote the tank volumes, manipulated controls and demands, respectively, and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{B}_d$, $\mathbf{E}$ and $\mathbf{E}_d$ are the model time-invariant matrices of suitable dimensions. Note that (41a) comes from (37), and (41b), from (40).

Moreover, a simplified representation of the Barcelona DWN is provided by means of Fig. 8. The system consists of seventeen tanks, sixty-one actuators, eleven nodes and twenty-five demand sectors. Note also that node 11 is split in two parts for the sake of a better visualization, but in practice it corresponds to a single location.

5.2.2. Experimental design

The management objectives regard constant demand satisfaction while minimizing costs and reducing wear and tear of the equipment. The set of operational objectives can be formally defined as follows:

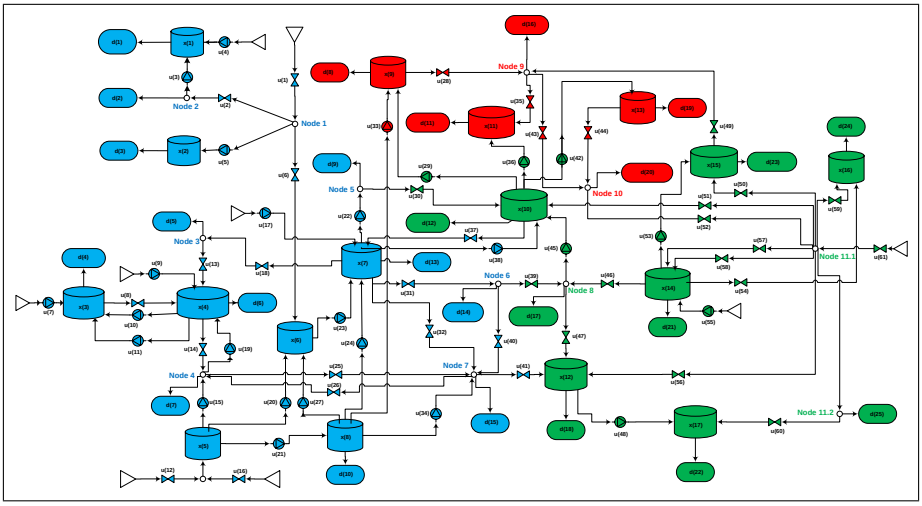

Figure 8: Schematic representation of the Barcelona DWN

- Cost minimization due to water extraction, treatment and distribution, which can be expressed as

$$J_k^{(1)} = \mathbf{W}_e \left(\boldsymbol{\alpha} + \boldsymbol{\beta}_k \right) \mathbf{u}_k, \tag{42}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ denote the known cost vectors related to water extraction, treatment and distribution, and $\mathbf{W}_e$ is the penalty associated to the economic objective. Note that the time dependency of $\boldsymbol{\beta}$ is due to different pumping electricity costs throughout the day.

- Ensure safety storage volumes: the water volumes in the tanks shall always guarantee demand satisfaction and therefore be greater than safety storage volumes, which may be computed based on demand forecasts. However, allowing water volumes in the tanks to fall below the safety volumes for reduced periods of time might improve the overall performance. This relaxation is penalized as

$$J_k^{(2)} = \mathbf{s}_k^{\mathsf{T}} \mathbf{W}_s \mathbf{s}_k, \tag{43}$$

where $\mathbf{s}_k$ is the vector of relaxed volumes at time instant k and $\mathbf{W}_s$ is the penalty associated to the safety objective. Note also that this relaxation

requires to modify (38) as

$$\underline{\mathbf{x}}_k - \mathbf{s}_k \leq \mathbf{x}_k \leq \overline{\mathbf{x}}_k \tag{44}$$

and add the following constraint:

$$\mathbf{s}_k \geq \mathbf{0}. \tag{45}$$

- Smoothness of the control signal, aiming at extending the useful life of pumps and valves. This can be expressed as

$$J_k^{(3)} = (\Delta \mathbf{u}_k)^\intercal \, \mathbf{W}_u \Delta \mathbf{u}_k, \tag{46}$$

where $\Delta \mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$, and $\mathbf{W}_u$ is the penalty associated to the smoothness objective.

Then, the simulation is designed as in [4]: a four-day simulation is considered, with a sampling time of one hour and a prediction horizon of twenty-four hours to account for the daily seasonality. Moreover, $\mathbf{W}_e$, $\mathbf{W}_s$ and $\mathbf{W}_u$ are selected according to the first scenario in the same reference.

5.2.3. Results

Before proceeding with the results, it is worth noting that the steps followed are the same as those reported for the quadruple-tank system in Section 5.1.3. Therefore, some content and/or mathematical formulation is not repeated here for the sake of convenience.

The CMPC for the Barcelona DWN can be formulated by constructing a multi-objective cost function with (42), (43) and (46), and gathering the constraints (39), (41), (44) and (45). Then, the centralized optimization problem can be formally derived by slightly modifying (31)–(32).

Once the graph associated to the matrix of KKT conditions for the centralized problem has been obtained, it can be partitioned using the fast unfolding algorithm. As a result, the three communities depicted in different colors in Fig. 8 are identified:

- The first subsystem (red) consists of tanks x_a, a=$\{9, 11, 13\}$, controls u_b, b=$\{28, 33, 35, 43, 44\}$, nodes N_c, c=$\{9, 10\}$, and demands d_d, d=$\{8, 11, 16, 19, 20\}$.

- The second subsystem (green) consists of tanks x_a, a=$\{10, 12, 14{:}17\}$, controls u_b, b=$\{29, 30, 36, 39, 42, 45{:}61\}$, nodes N_c, c=$\{8, 11\}$, and demands d_d, d=$\{12, 17, 18, 21{:}25\}$.

- The third subsystem (blue) consists of tanks x_a, a=$\{1{:}8\}$, controls u_b, b=$\{1{:}27, 31, 32, 34, 37, 38, 40, 41\}$, nodes N_c, c=$\{1{:}7\}$, and demands d_d, d=$\{1{:}7, 9, 10, 13{:}15\}$.

An inspection of the centralized model reveals that the subsystems are coupled only through the inputs. Moreover, the couplings are as follows:

$$
\begin{aligned}
\boldsymbol{\mu}_{1,3} &= \{\mathbf{u}_{33}\},\\
\boldsymbol{\mu}_{2,1} &= \{\mathbf{u}_{29},\, \mathbf{u}_{36},\, \mathbf{u}_{42},\, \mathbf{u}_{49}\},\\
\boldsymbol{\mu}_{2,3} &= \{\mathbf{u}_{30},\, \mathbf{u}_{39}\},\\
\boldsymbol{\mu}_{3,2} &= \{\mathbf{u}_{37},\, \mathbf{u}_{38},\, \mathbf{u}_{41}\},
\end{aligned}
$$

where $\boldsymbol{\mu}_{i,j}$ denotes the set of manipulated controls that have been assigned to the i-th subproblem (and are thus optimized in the i-th subproblem) but which also have an effect on the j-th subsystem, with $i, j = \{1, 2, 3\}$. Note that similar results are reported in [4] using a different partitioning approach, as the number of coupled controls is the same.

Then, a subproblem for each of the identified subsystems can be formulated by modifying (34)–(35). To illustrate the approach, and in the same spirit as [4], the performances of the centralized and distributed approaches are compared. To this end, the evolution of the volume of a tank, a manipulated control and the total cost are depicted in Figs. 9, 10 and 11, respectively. It can be seen that the results obtained using both approaches are rather similar for both the tank volume and the flow through the valve. Indeed, Fig. 11 reveals that the

performance of the distributed approach is only slightly worse than that of the centralized counterpart, which yields the optimal performance. Such small decrease of performance comes with the benefit of the reduction in computation time that can be achieved if the subproblems are solved in parallel. Indeed, the computation time for a single DMPC-OCD iteration is equal to the computation time for the largest subproblem. In the case of the Barcelona DWN, the average computation time (defined in the same way as for the quadruple-tank system) for the CMPC equals 0.521 seconds, while it takes 0.468 seconds in the case of the DMPC-OCD. While the reduction in time might not be impressive, this is due to the fact that the partitioning approach yields one significantly-larger subsystem, a fact that affects the total computation time. However, this partitioning offers modularity maximization and information-sharing minimization, thus providing an interesting trade-off. Furthermore, (36) is adapted to achieve convergence, which does not require more than five iterations. This shows that the proposed approach scales well for large-scale systems, as only one iteration less was required to achieve convergence for the quadruple-tank system.

Two final comments to conclude the analysis of the results: on the one hand, the system is characterized by a daily periodic behavior, both in the consumer demands (higher during the day and lower during the night) and in the pumping electricity costs for the water company, which follow the same trend as demands. This fact can be realized in Fig. 10, where the flow through the eighteenth actuator (from the seventh tank to the fifth demand) is higher during the day and lower during the night, and also in the cost function in Fig. 11. On the other hand, it is worth mentioning that the results regarding the evolution of the cost function differ a little from those in [4], as a scaling factor was used to present the results in economic units rather than in euros for confidentiality reasons. The same approach has been followed in this work, but the same scaling factor may not have necessarily been applied.

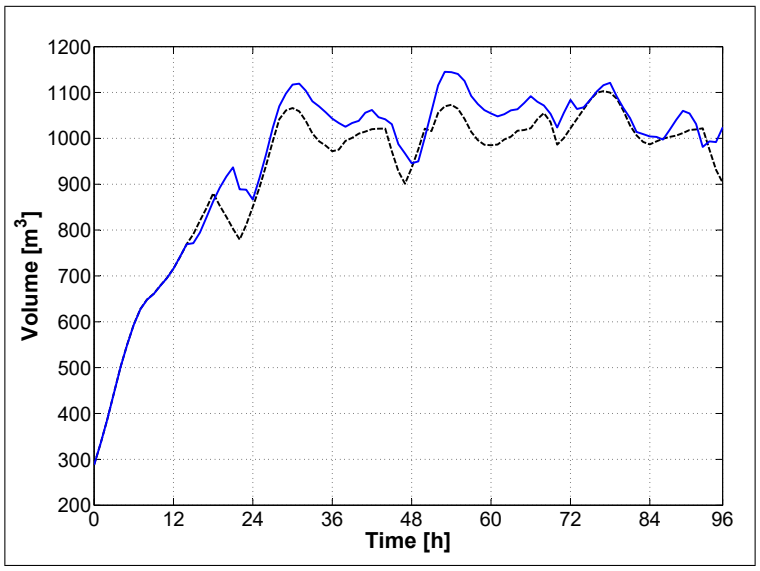

Figure 9: Evolution of the water volume of tank 2 for centralized (blue solid line) and distributed (black dashed line) implementations

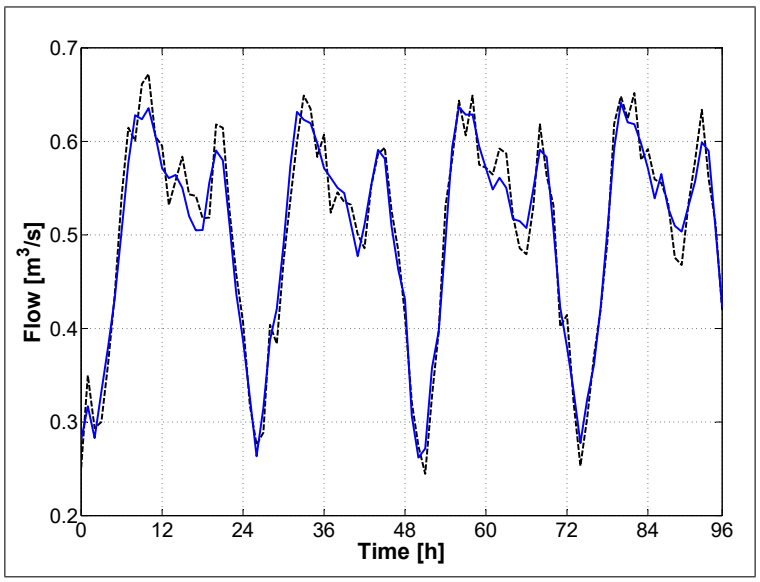

Figure 10: Evolution of the flow through valve 18 for centralized (blue solid line) and distributed (black dashed line) implementations

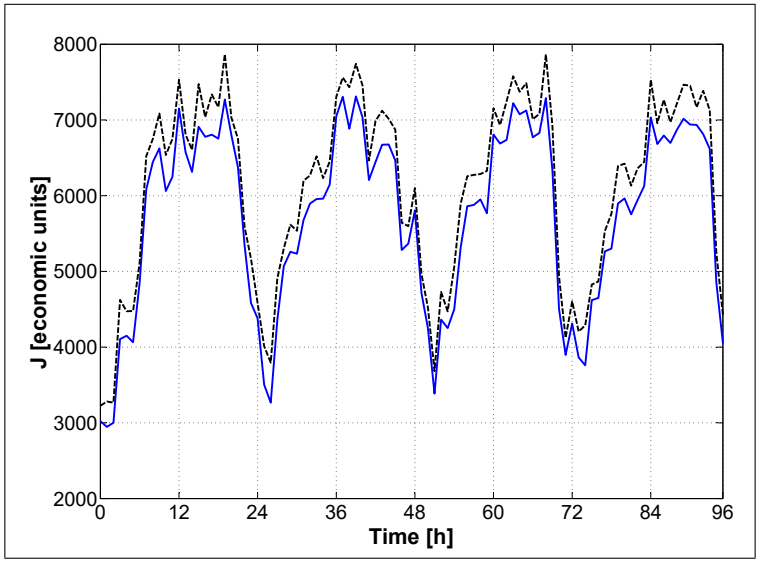

Figure 11: Evolution of the cost function for centralized (blue solid line) and distributed (black dashed line) implementations

6. Conclusions

This work has presented a DMPC approach for large-scale systems, given the fact that centralized implementations often suffer from non-scalability and spatial distribution. The overall problem is decomposed using a particular implementation of the well-known Lagrangian relaxation procedure known as OCD. Indeed, the KKT conditions of the overall system are formulated and manipulated to determine a set of smaller subproblems whose solution converges to the centralized optimal solution. However, and despite the fact that the OCD assumes that the problem can be decomposed, it does not provide the optimal system partitioning. Therefore, a heuristic approach known as community detection is used to determine how the system should be partitioned, as it yields a close-to-optimal performance. Then, the overall problem can be decomposed by manipulating the graph associated to the matrix of KKT conditions, thus allowing to take into account the couplings introduced by the cost function in the partitioning step. Finally, the subproblems can be coordinated and iteratively

solved until the errors due to the couplings are below a certain threshold. The proposed approach is tested on the quadruple-tank system and the Barcelona DWN, as each of them allows to highlight different aspects. The results allow to conclude that the methodology derived in this work provides a satisfactory performance, as it allows to fulfill the operational goals.

Although the proposed approach can be applied to any large-scale system, it has been derived with the inland waterways case study in mind. Indeed, preliminary results on centralized control and state estimation for these systems have been reported in [49]. However, inland waterways are characterized by several features, e.g., complex dynamics, large time delays and resonance phenomena, that make it reasonable to test the approach on simpler systems first. Moreover, the inland waterways case study would also require distributed state estimation since the states cannot be measured, which would add complexity to the final design. Once the performance of this approach on inland waterways has been validated, several ideas might be explored. One possibility could be to extend the formulation to propose a robust strategy considering tube-based approaches. On the other hand, the design of such controllers and observers could be integrated in a hierarchical approach such as the one derived in [50], where additional aspects such as tidal periods and discrete-valued actuators are taken into consideration. Finally, the DMPC-OCD could be used in combination with the fault diagnosis approach presented in [51] to propose a control reconfiguration strategy, aiming at ensuring that the system continues to perform even in the presence of faults. However, this would require to extend the proposed partitioning approach, as it would need to be performed online every time that a fault was diagnosed. Indeed, the faulty components would need to be disconnected for maintenance, which would result in changes in the network topology that would require to perform online re-partitioning.

Acknowledgments

This work has been partially supported by ARMINES through the contract 1908V/1700661 and by the IIW (Institution Intercommunale des Wateringues). This work also falls within the framework of the CHEEF2 project: the support from the Cluster MEDEE is gratefully acknowledged.

References

[1] J. B. Rawlings, B. T. Stewart, Coordinating multiple optimization-based controllers: New opportunities and challenges, Journal of Process Control 18 (9) (2008) 839 – 845.

[2] E. F. Camacho, C. Bordons, Model Predictive Control, Springer, London, 1998.

[3] S. J. Qin, T. A. Badgwell, A survey of industrial model predictive control technology, Control Engineering Practice 11 (7) (2003) 733 – 764.

[4] C. Ocampo-Martínez, 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.

[5] J. M. Grosso, C. Ocampo-Martínez, V. Puig, Non-centralized Predictive Control for Drinking-Water Supply Systems, in: Real-time Monitoring and Operational Control of Drinking-Water Systems, Springer, 2017, pp. 341–360.

[6] R. Scattolini, Architectures for distributed and hierarchical Model Predictive Control - a review, Journal of Process Control 19 (5) (2009) 723 – 731.

[7] P. D. Christofides, R. Scattolini, D. Muñoz de la Peña, J. Liu, Distributed model predictive control: A tutorial review and future research directions, Computers & Chemical Engineering 51 (2013) 21–41.

[8] J. M. Maestre, R. R. Negenborn, Distributed model predictive control made easy, Vol. 69, Springer Sciene & Business Media, 2014.

[9] E. Camponogara, D. Jia, B. H. Krogh, S. Talukdar, Distributed model predictive control, IEEE Control Systems Magazine 22 (1) (2002) 44–52.

[10] A. Allman, W. Tang, P. Daoutidis, DeCODe: a community-based algorithm for generating high-quality decompositions of optimization problems, Optimization and Engineering 20 (4) (2019) 1067–1084.

[11] A. Kargarian, J. Mohammadi, J. Guo, S. Chakrabarti, M. Barati, G. Hug, S. Kar, R. Baldick, Toward distributed/decentralized DC optimal power flow implementation in future electric power systems, IEEE Transactions on Smart Grid 9 (4) (2018) 2574–2594.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, , J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning 3 (1) (2011) 1–122.

[13] Y. Wang, S. Wang, L. Wu, Distributed optimization approaches for emerging power systems operation: a review, Electric Power Systems Research 144 (2017) 127 – 135.

[14] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, J. Lavaei, A survey of distributed optimization and control algorithms for electric power systems, IEEE Transactions on Smart Grid 8 (6) (2017) 2941–2962.

[15] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, G. Pannocchia, Cooperative distributed model predictive control, Systems & Control Letters 59 (8) (2010) 460 – 469.

[16] C. Conte, C. N. Jones, M. Morari, M. N. Zeilinger, Distributed synthesis and stability of cooperative distributed model predictive control for linear systems, Automatica 69 (2016) 117 – 125.

[17] Z. Wang, C. J. Ong, Distributed Model Predictive Control of linear discrete-time systems with local and global constraints, Automatica 81 (2017) 184 – 195.

[18] K. I. Tsianos, The role of the network in distributed optimization algorithms: Convergence rates, scalability, communication/computation tradeoffs and communication delays, Ph.D. thesis, Dept. Elect. Comp. Eng., McGill Univ. (2013).

[19] B. Gharesifard, J. Cortés, Distributed continuous-time convex optimization on weight-balanced digraphs, IEEE Transactions on Automatic Control 59 (3) (2014) 781–786.

[20] A. Nedić, A. Olshevsky, Distributed optimization over time-varying directed graphs, IEEE Transactions on Automatic Control 60 (3) (2015) 601–615.

[21] W. Tang, P. Daoutidis, Fast and stable nonconvex constrained distributed optimization: The ELLADA algorithm (2020). arXiv:2004.01977.

[22] P. Segovia, L. Rajaoarisoa, F. Nejjari, E. Duviella, V. Puig, A communication-based distributed model predictive control approach for large-scale systems, in: 2019 IEEE 58th Conference on Decision and Control (CDC), 2019, pp. 8366–8371.

[23] J. J. Yamé, F. Gabsi, T. Darure, T. Jain, F. Hamelin, N. Sauer, Optimality condition decomposition approach to distributed model predictive control, in: 2019 American Control Conference (ACC), 2019, pp. 742–747.

[24] S. S. Jogwar, P. Daoutidis, Community-based synthesis of distributed control architectures for integrated process networks, Chemical Engineering Science 172 (2017) 434 – 443.

[25] D. B. Pourkargar, A. Almansoori, P. Daoutidis, Impact of decomposition on distributed model predictive control: A process network case study, Industrial & Engineering Chemistry Research 56 (34) (2017) 9606–9616.

[26] W. Tang, D. B. Pourkargar, P. Daoutidis, Relative time-averaged gain array (RTAGA) for distributed control-oriented network decomposition, AIChE Journal 64 (5) (2018) 1682–1690.

[27] W. Tang, A. Allman, D. B. Pourkargar, P. Daoutidis, Optimal decomposition for distributed optimization in nonlinear model predictive control through community detection, Computers & Chemical Engineering 111 (2018) 43 – 54.

[28] W. Tang, P. Daoutidis, Network decomposition for distributed control through community detection in input–output bipartite graphs, Journal of Process Control 64 (2018) 7 – 14.

[29] P. Daoutidis, W. Tang, A. Allman, Decomposition of control and optimization problems by network structure: Concepts, methods, and inspirations from biology, AIChE Journal 65 (10) (2019) e16708.

[30] C. Conte, M. N. Zeilinger, M. Morari, C. N. Jones, Robust distributed model predictive control of linear systems, in: 2013 European Control Conference (ECC), 2013, pp. 2764–2769.

[31] J. B. Rawlings, D. Q. Mayne, Model predictive control: theory and design, Nob Hill Pub. Madison, Wisconsin, 2009.

[32] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, Constrained model predictive control: Stability and optimality, Automatica 36 (6) (2000) 789 – 814.

[33] D. Limon, I. Alvarado, T. Alamo, E. F. Camacho, MPC for tracking piecewise constant references for constrained linear systems, Automatica 44 (9) (2008) 2382 – 2387.

[34] D. P. Bertsekas, Constrained optimization and Lagrange multiplier methods, Academic Press, 2014.

[35] S. Boyd, L. Vandenberghe, Convex optimization, Cambridge University Press, 2004.

[36] N. Chatzipanagiotis, D. Dentcheva, M. M. Zavlanos, An augmented Lagrangian method for distributed optimization, Mathematical Programming 152 (1-2) (2014) 405–434.

[37] D. D. Siljak, Decentralized control of complex systems, Courier Corporation, 2011.

[38] R. Negenborn, P. J. van Overloop, T. Keviczky, B. De Schutter, Distributed model predictive control of irrigation canals, Networks and Heterogeneous Media 4 (2) (2009) 359–380.

[39] A. J. Conejo, E. Castillo, R. Mínguez, R. García-Bertrand, Decomposition techniques in mathematical programming: engineering and science applications, Springer Science & Business Media, 2006.

[40] A. Allman, W. Tang, P. Daoutidis, Towards a generic algorithm for identifying high-quality decompositions of optimization problems, in: 13th International Symposium on Process Systems Engineering (PSE 2018), Vol. 44 of Computer Aided Chemical Engineering, Elsevier, 2018, pp. 943 – 948.

[41] V. D. Blondel, J. L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment 2008 (10) (2008) P10008.

[42] Y. Saad, Iterative methods for sparse linear systems, Vol. 82, SIAM, 2003.

[43] J. Löfberg, YALMIP: a toolbox for modeling and optimization in MATLAB, in: IEEE International Symposium on Computer Aided Control Systems Design, Taipei, Taiwan, 2004.

[44] K. H. Johansson, The quadruple-tank process: a multivariable laboratory process with an adjustable zero, IEEE Transactions on Control Systems Technology 8 (3) (2000) 456–465.

[45] I. Alvarado, D. Limón, D. Muñoz de la Peña, J. M. Maestre, M. A. Ridao, H. Scheu, W. Marquardt, R. 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.

[46] J. M. Grosso, C. Ocampo-Martínez, V. Puig, B. Joseph, Chance-constrained model predictive control for drinking water networks, Journal of Process Control 24 (5) (2014) 504–516.

[47] J. Barreiro-Gomez, C. Ocampo-Martinez, N. Quijano, Dynamical tuning for MPC using population games: a water supply network application, ISA Transactions 69 (2017) 175 – 186.

[48] J. Quevedo, V. Puig, G. Cembrano, J. Blanch, J. Aguilar, D. Saporta, G. Benito, M. Hedo, A. Molina, Validation and reconstruction of flow meter data in the Barcelona water distribution network, Control Engineering Practice 18 (6) (2010) 640 – 651.

[49] P. Segovia, L. Rajaoarisoa, F. Nejjari, E. Duviella, V. Puig, Model predictive control and moving horizon estimation for water level regulation in inland waterways, Journal of Process Control 76 (2019) 1–14.

[50] P. Segovia, E. Duviella, V. Puig, Multi-layer model predictive control of inland waterways with continuous and discrete actuators, in: 21st IFAC World Congress, 2020, accepted.

[51] P. Segovia, J. Blesa, K. Horváth, L. Rajaoarisoa, F. Nejjari, V. Puig, E. Duviella, Modeling and fault diagnosis of flat inland navigation canals, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 232 (6) (2018) 761–771.