Hierarchical and Decentralised Model Predictive Control of Drinking Water Networks: Application to Barcelona Case Study

C. Ocampo-Martinez[†,*] D. Barcelli[‡] V. Puig[†] A. Bemporad[‡]

[†]Institut de Robòtica i Informàtica Industrial (CSIC-UPC)
Llorens i Artigas, 4-6, 08028 Barcelona, Spain
[‡,]IMT Institute for Advanced Studies Lucca
Via della Quarquonia 2, Lucca, Italy
[◇]Information Engineering Department, University of Siena
Via Roma 56, 53100 Siena, Italy

Abstract

In this paper, a hierarchical and decentralised model predictive control (DMPC) strategy for drinking water networks (DWN) is proposed. The DWN is partitioned in a set of subnetworks using a partitioning algorithm that makes use of the topology of the network, historic information about the actuator usage and heuristics. A suboptimal DMPC strategy was derived, which consists in a set of MPC controllers, whose prediction model is a plant partition, where each element solves its control problem in a hierarchical order. A comparative simulation study between centralised MPC (CMPC) and DMPC approaches is developed using a case study, which consists in an aggregate version of the Barcelona DWN. Results have shown the effectiveness of the proposed DMPC approach in terms of the scalability of computations with an acceptable admissible loss of performance in all the considered scenarios.

1 Introduction

Drinking water management in urban areas is a subject of increasing concern as cities grow. Limited water supplies, conservation and sustainability policies, as well as the infrastructure complexity for meeting consumer demands with appropriate flow pressure and quality levels make water management a challenging problem. Many modern water systems are operated through centralised telecontrol systems. In most cases, network operation is carried out using heuristic rules and "historic" strategies, which were the product of years of operational experience and empirical results. While these strategies may generally be adequate, the best operational policies may be very complex to determine in large-scale interconnected networks. Thus, decision-support systems for operational control, which are based on mathematical models of network

[*] Corresponding author. Tel.: +34 93 401 5752; fax: +34 93 401 5750. E-mail: cocampo@iri.upc.edu (C. Ocampo-Martinez)

operation and optimal control techniques, provide useful guidance for efficient management of water networks.

Model-based Predictive Control (MPC) has been proved to be one of the most widely accepted advanced control technique for the operational control of water systems [1, 2, 3]. The main reason is that, once the network dynamical model is available, the MPC design just consists in expressing the desired performance specifications through different control objectives and constraints on system variables (e.g., minima/maxima of selected process variables and/or their rates of change), which are necessary to ensure process safety and asset health. The rest of the MPC design is automatic: the given model, constraints and weights define an optimal control problem over a finite time horizon in the future (for this reason the approach is said predictive). This is translated into an equivalent optimisation problem and solved on line to obtain an optimal sequence of future control moves. Only the first of these moves is applied to the process, as at the next time step a new optimal control problem is solved, to exploit the information coming from fresh new measurements using in a receding horizon strategy. In this way, an open-loop design methodology (i.e., optimal control) is transformed into a feedback one.

Nevertheless, the main hurdle for MPC control, as any other control technique, when applied to large-scale networks in a centralized way, is the non-scalability. The reason is that a huge control model is required along with the need of being rebuilt on every change in the system configuration as, for example, when some part of the network should be stopped because of maintenance actions or malfunctions. Subsequently, a model change would require re-tuning the centralised controller. It is obvious that the cost of setting up and maintaining the monolithic solution of the control problem is prohibitive. A way of circumventing these issues might be by looking into *decentralised* MPC (DMPC) or *distributed* MPC techniques, where networked local MPC controllers are in charge of controlling each one a part of the entire system. The main difference between distributed and decentralized MPC is that the former uses negotiations and recomputations of local control actions within the sampling period to increase the level of cooperation, whereas the latter does not (at the benefit of computation time, but at the cost of optimality), for further details see [4].

The success of centralised MPC (CMPC) drives now a new interest in this old area of distributed control, becoming DMPC one of the hottest topics in control during the early 21st century. Thus, two research projects (HD-MPC and WIDE) are currently being carried out in Europe, both focused on the development of decentralised and distributed MPC techniques. Few works have been recently published in this area; see, e.g., [5, 6, 7, 8, 9, 10], among others. However, there is a prior problem to be solved: the system decomposition into subsystems. The importance of this issue has already been noticed in classic-control books addressing the decentralised control of large-scale systems (LSS); see, e.g., [11] or [12]. These references propose some approaches for dealing with the decomposition of dynamical networked systems under certain assumptions, which are related to the level of coupling of the constitutive elements belonging to the considered LSS.

The main contribution of this paper consists in presenting the application in simulation of a hierachical-like DMPC approach to the Barcelona drinking water network (DWN). The aim of DMPC is to reduce the computational burden and increase scalability and modularity with respect to the centralised counterpart, but still maintaining a convenient level of suboptimality with respect to the desired control objectives. Moreover, the advantage of the hierarchical-like DMPC approach is the simplicity of its implementation given the absence of negotiations among controllers which allows for a simple implementation from the networking viewpoint. This fact involves much shorter computational times since only one optimization problem should be solved for each subsystem. Furthermore, each local MPC controller could be converted into its *explicit* form [13] leading to a low online complexity. To apply the proposed DMPC approach, the

DWN is partitioned in a set of subnetworks using a partitioning algorithm that makes use of the topology of the network, the information about the actuator usage and heuristics. The partition approach finds a set of non-overlapping subsystems weakly interconnected.

The paper is structured as follows: Section 2 describes the case study considered in the paper. Section 3 briefly introduces the partitioning approach for dynamical systems used with the case study of this paper. Section 4 presents and discusses the hierarchical-like DMPC strategy applied to the case study. Section 5 discusses the main **simulation** results derived from the application of the proposed control approach over the considered case study. Finally, conclusions and directions for further work are reported in Section 6.

2 Case-study Description

2.1 System description

The Barcelona DWN, managed by Aguas de Barcelona, S.A. (AGBAR), not only supplies drinking water to Barcelona city but also to the metropolitan area. The sources of water are the Ter and Llobregat rivers, which are regulated at their head by some dams with an overall capacity of 600 cubic hectometres. Currently, there are four drinking water treatment plants (WTP): the Abrera and Sant Joan Despí plants, which extract water from the Llobregat river, the Cardedeu plant, which extracts water from Ter river, and the Besòs plant, which treats the underground flows from the aquifer of the Besòs river. There are also several underground sources (wells) that can provide water through pumping stations. Those different water sources currently provide a flow of around 7 m^3/s. The water flow from each source is limited and with different water prices depending on water treatments and legal extraction canons.

The Barcelona DWN is structurally organised in two layers. The upper layer, named as *transport network*, links the water treatment plants with the reservoirs distributed all over the city. The lower layer, named *distribution network* is sectorised in subnetworks. Each subnetwork links a reservoir with each consumer. This paper is focused on the transport network. Thus, each subnetwork of the distribution network is modelled as a demand sector. The demand of each sector is characterised by a demand pattern, which can be predicted by using a time-series model [14]. The control system of the transport network is also organised in two layers (see Figure 1). The upper layer is in charge of the global control of the network, establishing the set-points of the regulatory controllers at the lower layer. Regulatory controllers are of PID type, while the supervisory layer controller is of MPC type. Regulatory controllers hide the network non-linear behaviour to the supervisory controller. This fact allows the MPC supervisory controller to use a control-oriented linear model.

2.2 System management criteria

AGBAR has provided the management policies for the Barcelona DWN, given their knowledge of the system. These management criteria are described below.

2.2.1 Minimising water production and transport costs

The main economic costs associated with drinking water production (treatment) are due to chemicals, legal canons, and electricity costs. The corresponding performance index to be minimised is expressed as

$$f_1(t) = W_e \ (\alpha_1 + \alpha_2(t))\, u(t), \tag{1}$$

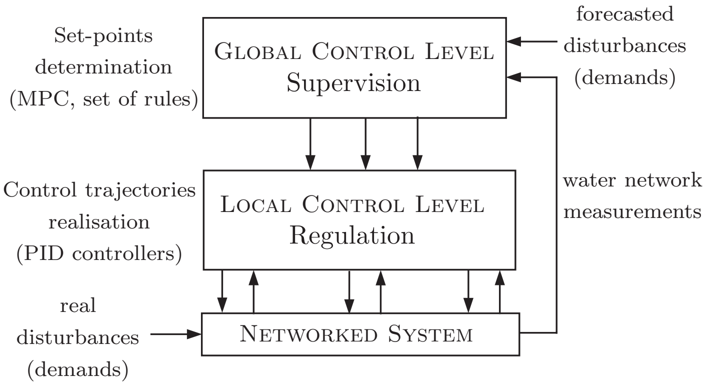

Figure 1: Hierarchical structure for RTC system

where $u \in \mathbb{R}^m$ denotes the manipulated flows through the system actuators, α_1 corresponds to a known vector of dimension $1 \times m$ related to the economic costs of the water according to the source (treatment plant, dwell, etc.), and $\alpha_2(t)$ is a vector of dimension $1 \times m$ associated with the economic cost of the flow through certain actuators (pumps only) and their control cost (pumping). Note the time variance of α_2 due to the fact that pumping electricity costs have different values according to the time of the day. The weight W_e is the penalty associated with economic costs with respect to the other objectives that will be included in the MPC optimisation problem.

2.2.2 Safety storage term

The satisfaction of water demands should be fulfilled at every time instant. However, some risk prevention mechanisms should be introduced in the tank management so that, the stored volume is preferably maintained around a given safety value in case of emergency, and to guarantee future water availability in case of demand forecast estimation errors. A quadratic expression for this concept is used and written as follows:

$$f_2(t) = (x(t) - \beta\, x^{\max})^T\, W_x\, (x(t) - \beta\, x^{\max}), \tag{2}$$

where $x \in \mathbb{R}^n$ denotes the water volumes at network tanks, β is a term which determines the safety volume to be considered for the control law computation and matrix W_x defines the weight of the objective in the cost function. This term prevents the controller from keeping the lowest possible water volumes in the tanks, which would reduce robustness to demand forecast errors.

2.2.3 Smoothness of the control actions

To smooth out the control action of MPC in order to avoid overpressures which can cause structural damage and leaks in the network, the following third term is included in the objective function to penalise variations $\Delta u(t) = u(t) - u(t-1)$ of the control signal between consecutive sampling intervals

$$f_3(t) = \Delta u(t)^T\, W_{\Delta u}\, \Delta u(t), \tag{3}$$

where $W_{\Delta u}$ is a $m \times m$ weight matrix.

2.3 Control-oriented modelling

Control-oriented modelling principles for DWNs have been widely presented in the literature, see [1, 15]. In order to obtain a control-oriented model of the DWN, the constitutive network elements as well as their basic relationships should be discussed. The reader is referred to the aforementioned references and to [16, 17] for further details of DWN modelling and specific insights related to the case study of this paper.

2.3.1 Tanks

A water tank provides the entire system with the storage capability of drinking water. The mass balance expression relating the stored volume x, the manipulated inflows and outflows (including the demand flows as outflows) for the i-th tank can be written as the difference equation

$$x_i(t+1) = x_i(t) + \Delta t \left(\sum_i q_{\text{in},i}(t) - \sum_j q_{\text{out},j}(t) \right), \tag{4}$$

for all discrete-time instant t, where $q_{\text{in},i}(t)$ and $q_{\text{out},j}(t)$ correspond to the i-th inflow and the j-th outflow, respectively, given in m^3/s. The physical constraint related to the range of volume capacities for the i-th tank is expressed as

$$x_i^{\min} \leq x_i(t) \leq x_i^{\max}, \quad \forall t, \tag{5}$$

where $x_i^{\min}$ and $x_i^{\max}$ denote the minimum and the maximum volume capacity, respectively, given in m^3. Since this constraint is physical, it is impossible to send more water to a tank than it can store, or draw more water than the stored amount.

2.3.2 Actuators

There are two types of control actuators: pumps and valves. The manipulated flows through the actuators represent the control input variables of the model, denoted in the sequel as u. Both pumps and valves have lower and upper physical limits that are also model constraints. As in (5), they are expressed as

$$u_i^{\min} \leq u_i(t) \leq u_i^{\max}, \quad \forall t, \tag{6}$$

where $u_i^{\min}$ and $u_i^{\max}$ denote the minimum and the maximum flow capacity, respectively, given in cubic meters per second. It is assumed that there is a local controller, which ensures that the required flow through the actuator is satisfied, following the discussion done for Figure 1.

2.3.3 Nodes

These elements correspond to the network points where water flows are merged or split. Thus, the nodes represent mass balance relations, being modelled as equality constraints related to inflows (from other tanks through valves or pumps) and outflows, these latter being represented not only by manipulated flows but also by demand flows. The expression of the mass conservation in these elements can be written as[1]

$$\sum_i q_{\text{in},i}(t) \; = \; \sum_j q_{\text{out},j}(t). \tag{7}$$

[1] With a slight abuse of notation, the node inflows and outflows are still denoted by q_{in} and q_{out}, respectively, despite they can be manipulated flows and hence denoted by u, if correspond.

2.3.4 Sectors of Consume

A sector of consume represents the water demand made by the network users of a certain physical area. It is considered as a system disturbance. Since the demand shows a periodic behaviour with daily and weekly seasonalities, it can be forecasted by using methods based on time series. In this paper, the demand forecasting algorithm used by the MPC consists in a two-level scheme [14] composed by

- a time-series model to represent the daily aggregate flow values, and
- a set of different daily flow demand patterns according to the day type to cater for different consumption during the weekends and holidays periods. Every pattern consists of 24 hourly values for each daily pattern.

This algorithm runs in parallel with the MPC algorithm. The daily series of hourly-flow predictions are computed as a product of the daily aggregate flow value and the appropriate hourly demand pattern.

Aggregate daily-flow model The aggregate daily-flow model is built on the basis of a time series modelling approach using an ARIMA strategy. A time series analysis was carried out on several daily aggregate series, which consistently showed a weekly seasonality, as well as the presence of deterministic periodic components. A general expression for the aggregate daily flow model, to be used for a number of demands in different locations, was derived using three main components:

- A *weekly-period oscillating signal*, with zero-average value to cater for cyclic deterministic behaviour, implemented using a second-order (two-parameter) model with two oscillating modes ($p_{1,2} = cos(\frac{2\pi}{7}) \pm jsin(\frac{2\pi}{7})$).
- An *integrator* takes into account possible trends and the non-zero mean value of the flow data.
- An autoregressive component to consider the influence of previous flow values within a week. For the general case, the influence of seven previous days is considered. However, after parameter estimation and significance analysis, the models are usually reduced implementing a smaller number of parameters a_i such as the model output y is expressed as
$$y(t) = -a_1 y(t-1) - a_2 y(t-2) - a_3 y(t-3) - a_4 y(t-4).$$

Combining the previous components in the following way:

$$\begin{aligned} \Delta y_{\text{int}}(t) &= y(t) - y(t-1), \\ \Delta y_{\text{osc}}(t) &= \Delta y_{\text{int}}(t) - 2\cos(2\pi/7)\Delta y_{\text{int}}(t-1) + \Delta y_{\text{int}}(t-2), \\ y_p(t) &= -a_1 \Delta y_{\text{osc}}(t-1) - a_2 \Delta y_{\text{osc}}(t-2) - a_3 \Delta y_{\text{osc}}(t-3) - a_4 \Delta y_{\text{osc}}(t-4), \end{aligned}$$

the structure of aggregate daily flow model for each demand sensor is therefore

$$y_p(t) = -b_1 y(t-1) - b_2 y(t-2) - ... - b_7 y(t-7). \tag{8}$$

The parameters $b_1, \ldots, b_7$ should be adjusted using least-squares-based parameter estimation methods and historical data (after pre-processing to obtain fault-free set). In parallel with the forecasting and control module, a data validation module should be considered, which validates the used information.

1-hour flow model The 1-hour flow model is based on distributing the daily flow prediction provided by the time-series model described in previous section using a one-hour-flow pattern that takes into account the daily/monthly variation in the following way:

$$y_{p1h}(t+i) = \frac{y_{pat}(t,i)}{\sum\limits_{j=1}^{24} y_{pat}(t,i)} y_p(j), \qquad i = 1, ..., 24,$$

where $y_{p1h}(t)$ is the predicted flow for the current day j using (8) and $y_{pat}(t)$ is the prediction provided by the one-hour-flow pattern with the flow pattern class day/month of the actual day. Demand patterns are obtained from statistical analysis. See [14] for further details.

2.3.5 Network Model

Considering the expressions presented above, the control-oriented model of a DWN in discrete-time state space can be written as

$$x(t+1) = A\,x(t) + B\,u(t) + B_p\,d(t), \tag{9}$$

where $x \in \mathbb{R}^n$ is the state vector corresponding to the water volumes of the n tanks, $u \in \mathbb{R}^m$ represents the vector of manipulated flows through the m actuators, and $d \in \mathbb{R}^p$ corresponds to the vector of the p demands. A, B, and B_p are the system matrices of suitable dimensions. Since the demands can be forecasted and they are assumed to be known, d is a known vector containing the measured disturbances affecting the system. Additionally, (9) can be rewritten as

$$\begin{aligned} x(t+1) &= A\,x(t) + \Gamma\,\upsilon(t), && \text{(10a)} \\ \begin{bmatrix} E_u & E_d \end{bmatrix} \upsilon(t) &= 0, && \text{(10b)} \end{aligned}$$

where $\Gamma = [B \quad B_p]$, $\upsilon(t) = [u(t)^T \quad d(t)^T]^T$, and E_u, E_d are matrices of suitable dimensions. Notice that (10a) comes from the mass balance in tanks while (10b) at the network nodes (see (7)). Also notice that when all the network flows are manipulated, then A is an identity matrix of suitable dimensions.

This modelling methodology has been applied to the Barcelona DWN aggregate network in Figure 2. From this figure, it can be seen that the network is comprised of 17 tanks (state variables), 61 actuators (26 pumping stations and 35 valves), 11 nodes and 25 main sectors of water demand (model disturbances). The model has been simulated and compared against real behaviour assessing its validity. The detailed information about physical parameters and other system values are reported in [17].

3 DWN Partitioning Approach

The application of DMPC to DWN depends crucially on how the network is decomposed into subsystems. Identifying subsystems is not an easy task in a large-scale network as it involves to find automatically“sufficiently small” sections of the networked plant that are not “too coupled” among them. The partitioning algorithm, proposed in this paper, aims to obtain this decomposition automatically by identifying clusters of elements that are strongly connected with each other but weakly interconnected with the other clusters, in order to represent the whole network as a set of loosely coupled subsystems [18]. The current version of the algorithm is though to be used off-line, that is, the partitioning of the system is static and is not carried out on-line. A further improvement could be to adapt the proposed algorithm such that the partitioning could be done on-line when, for instance, some structural change of the network appears.

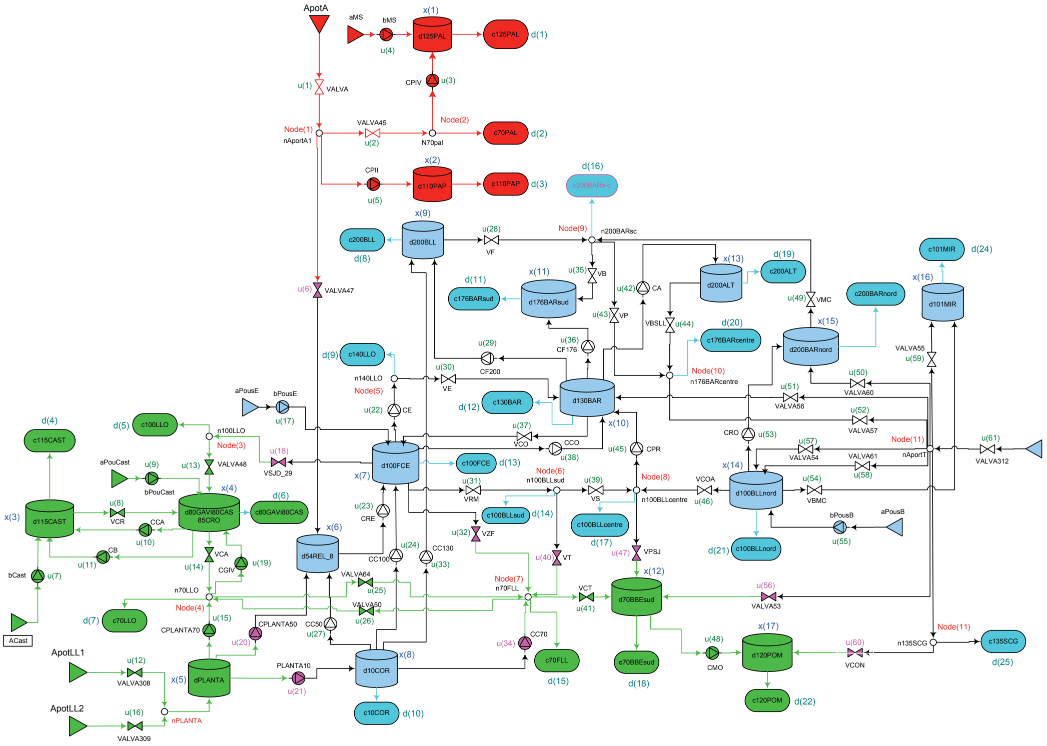

Figure 2: Aggregate case of the Barcelona Drinking Water Network

3.1 Partitioning algorithm

As a starting point, the partitioning algorithm requires the following information of the DWN:

1. The interconnection structure characterised by the matrix

$$M = \begin{bmatrix} A_{sp} & B_{sp} \end{bmatrix}, \tag{11a}$$

where

$$A_{sp} = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}, \quad B_{sp} = \begin{bmatrix} B \\ E \end{bmatrix}. \tag{11b}$$

where A and B are the system matrices in (9), the subscript sp identifies the matrices used for system decomposition, and $E \triangleq [E_1 \quad E_2]$ is the matrix related to the equality constraints (10b). In order to take into account input bounds, new normalised inputs are introduced $\bar{u} \triangleq u/u_{max}$ so that $\bar{u} \in [0, 1]$. Thus, new matrices $\bar{B}$ and $\bar{E}$ are introduced in (11b) to take into account the rescaling. From matrix M, the adjacency matrix Ψ of the network graph can be obtained by replacing the non-zero elements by ones, leaving the null elements unchanged.

2. A threshold value ε is used for determining whether a term, which takes into account the actuator capacity (maximum allowable flow) and its usage frequency, has a negligible *effect* on the entire plant. In this way the less important actuators are filtered out, in order to reduce the coupling degree of the system and identify independent subnetworks.

The partitioning algorithm proceeds by decomposing the matrix M into a set of submatrices, named as *partitions* and denoted by $P_\varepsilon = \left\{M_1, \cdots, M_{n_p}\right\}$. Then, P_ε correspond to a set of subgraphs (subsystems) obtaining by deleting the edges corresponding to elements of M with magnitude no larger than ε. That is, the idea behind the partitioning approach is to neglect less important elements (i.e., links) in matrix M such that the resulting $\tilde{M}$ is less coupled. Ideally, $\tilde{M}$ should lead to a permutation matrix P such that $P'\tilde{M}P$ is block-diagonal. This procedure is repeated iteratively by reducing ε until an enough number of partitions is obtained. Algorithm 1 summarises the steps of the proposed partitioning algorithm.

Partitions can be tuned by means of parameter ε of the proposed approach, which makes the user able to attempt matching the desired number and size of subsystems.

Typically, in the first iteration, Algorithm 1 neglects a high number of elements of M, highly reducing the matrix connectivity degree and obtaining a subsystem decomposition. Then, once the sets of states/inputs relative to each partition are computed, the task of finding a suitable P that block-diagonalises the matrix $P'\tilde{M}P$ is a matter of linear algebra implementation. Every subsystem is composed by sets of state and input variables that are linked, meaning that are in the same block in the $P'\tilde{M}P$ diagonal. Let $\mathbb{X}^i$ and $\mathbb{U}^i$ be respectively the sets of state and input variables assigned to subsystem i, while $L(\mathbb{X}^i)$ and $L(\mathbb{U}^i)$ determine the number of variables for each set. A subsystem is created if both numbers are different than zero. All state and input variables that are not assigned to any of the currently created subsystems, that is, that does not belong to $\mathbb{X}^i$ or $\mathbb{U}^i$, respectively, are available for the next iteration. Otherwise, variables already assigned to a subsystem~~,~~ in the current or in a previous iteration, are *masked*[2] to prevent their reassignment to other subsystem.

Then, a new iteration of the algorithm starts by decreasing ε (e.g., halving ε). Algorithm 1 iterates until all state variables are assigned to a subsystem. Note that the algorithm may terminate even if some inputs are not be assigned to any subsystem, which is due to automatic threshold based neglecting process. Such issue can be managed by manually include unassigned inputs to proper subsystem following engineering insight.

The importance of the mask arises from the structure of the algorithm. In fact, if not excluded, all previously assigned states and inputs would be part of the next iteration partition, introducing couplings and hence increasing the size of the resulting submodels. The aforementioned inclusion easily follows from the decreasing of ε among sequential iterations.

Few remarks on the above algorithm:

1. At any iteration of Algorithm 1, the numerical value of ε is a crucial tuning knob of the approach. A guideline is that the larger is the decreasing step, the larger is the size of the obtained subsystems. Ways for automatically determining the step size are a subject of current research.

2. Matrix E in (11b) defines a constraint among actuators that can be easily taken into account if all the actuators belong to the same subsystem. Otherwise, since each controller manipulates every partition independently from the others, negotiations between controllers would be required to guarantee the fulfillment of node constraints.

3. The use of masks to prevent state reassignment avoids that submodels have overlapping states and inputs: if a state variable is used in a model by a controller, no other controller can use it. The main benefit of this choice is the very low level of coupling between partitions, but the price to pay is a potential decrease of closed-loop performance.

[2] Let us consider a variable to be masked when it does not belong to any set since it has already been classified in a previous iteration.

Algorithm 1 Automatic partitioning algorithm

```
Initialise masks to a neutral value
Initialise the sets of unassigned variables $\mathbb{X}$ and $\mathbb{U}$ with all state and input variables, respectively
Determine the number of unassigned states: $N_x = L(\mathbb{X})$;
Init $\varepsilon$
while $N_x > 1$ do
    Apply masks to $A_{sp}$ and $B_{sp}$
    $M = [A_{sp} \;\; B_{sp}\bar{u}]$
    For all elements of $M$
    if $M_{i,j} < \varepsilon$ then
        $\tilde{M}_{i,j} = 0$;
    else
        $\tilde{M}_{i,j} = 1$;
    end if
    Find $P$ such that $P'\tilde{M}P$ is block diagonal
    Identify parts satisfying $N_{x^i} = L(\mathbb{X}^i) > 0$ and $N_{u^i} = L(\mathbb{U}^i) > 0$ and add to previous ones
    Update $N_x$
    Update masks with updated states and inputs
    Update $\varepsilon$
end while
```

4. The current structure of the algorithm is unsuitable to handle state overlaps because it relies on links between elements that present different degree of coupling. Hence, once the stronger couplings are eliminated (using masking), the weaker ones gain relative importance. State overlaps may be introduced a posteriori based on engineering insight, in order to increase the adherence with respect the original centralised model. Handling overlapping in an automatic way is also a current research topic.

5. In some cases even relatively small connections, i.e., capable of carrying a minor amount of water, are very important for demand satisfaction. A way of accounting for such an issue is to perform a simulation using, for instance, a CMPC controller, and compute the average percentage of use for each actuator. Thus, this information could be used to weight $\bar{u}$ component-wise. The main drawback of this approach is the need of (and dependence on) simulation.

6. Note that the proposed algorithm can be customised by setting different importance levels of states vs. inputs, by weighting the related components in M. By defining

$$M = [W_A' A_{sp} W_A \quad W_B' B_{sp} W_B \quad W_u' \bar{u} W_u],$$

   where W_A, W_B and W_u are weights respectively of A, B and u, it is possible to affect the resulting partitioning outcome.

7. The structure of the proposed algorithm suggests that termination is achieved if the ε value is decreased at each iteration. However, at the current status of development, the algorithm cannot guarantee any property for the resulting partitioning but the assignment of all system-state variables to a subsystem.

The decomposition process of matrix M reported here is similar to the one proposed by the ε-decomposition method in [18]. The underlying idea in both cases is to disconnect those actuators

Table 1: Dimension comparison between the subsystems and the whole network.

Elements	Subsyst 1	Subsyst 2	Subsyst 3	Whole Model
Tanks	2	5	10	17
Actuators	5	22	34	61
Demands	3	7	15	25
Nodes	2	3	6	11

corresponding to interconnections with strength smaller than the prescribed ε, identifying the disconnected subsystems. According to [18], there are s different ε-decompositions P_ε that can be obtained for different values of ε satisfying

$$\max_{i \neq j} |m_{ij}| = \varepsilon_1 < \varepsilon_2 < \cdots < \varepsilon_K = 0,$$

with $K \leq s$, where $s = dim(M)$. Moreover, such decompositions are nested, that is, the partitions obtained satisfy: $P_{\varepsilon_1} \subset P_{\varepsilon_2} \cdots P_{\varepsilon_K}$ with P_{ε_1} being the finest and P_{ε_k} the coarsest. The main novelty of the algorithm presented in this paper is the matrix normalisation taking into account actuator physical/operative limits, and the iterative threshold updating that allows one to take into account weaker coupling without being influenced by the stronger ones.

3.2 Case study partitioning

Using the partitioning algorithm presented in this section, the aggregate model of the Barcelona DWN is decomposed in three subsystems, as depicted in Figure 2 in different colours. The resultant decomposition follows the scheme shown in Figure 3. The subsystems are defined by the following elements:

- *Subsystem 1:* Composed by tanks x_i, $i \in \{1, 2\}$, inputs u_j, $j \in \{1 : 5\}$, demands d_l, $l \in \{1, 2, 3\}$, and nodes n_q, $q \in \{1, 2\}$. It is represented in Figure 2 with red colour and corresponds to Subsystem S_1 in Figure 3.
- *Subsystem 2:* Composed by tanks x_i, $i \in \{3, 4, 5, 12, 17\}$, inputs u_j, $j \in \{7 : 16, 18, 19, 25, 26, 32, 34, 40, 41, 47, 48, 56, 60\}$, demands d_l, $l \in \{4 : 7, 15, 18, 22\}$, and nodes n_q, $q \in \{3, 4, 7\}$. It is represented in Figure 2 with green colour and corresponds to Subsystem S_2 in Figure 3.
- *Subsystem 3:* Composed by tanks x_i, $i \in \{6 : 11, 13 : 16\}$, the inputs u_j, $j \in \{6, 17, 20 : 24, 27 : 31, 33, 35 : 39, 42 : 46, 49 : 55, 57, 58, 59, 61\}$, demands d_l, $l \in \{8 : 14, 16, 17, 19, 20, 21, 23, 24, 25\}$, and nodes n_q, $q \in \{5, 6, 8 : 11\}$. It is represented in Figure 2 with blue colour and corresponds to Subsystem S_3 in Figure 3.

Table 1 collects the resultant dimensions for each subsystem and the corresponding comparison with the dimensions of the vectors of variables for the entire aggregate network.

4 DMPC Approach

Using the Barcelona DWN decomposition which is obtained from the partitioning algorithm in Section 3, a DMPC strategy is implemented in order to manage the networked system. This DMPC strategy considers

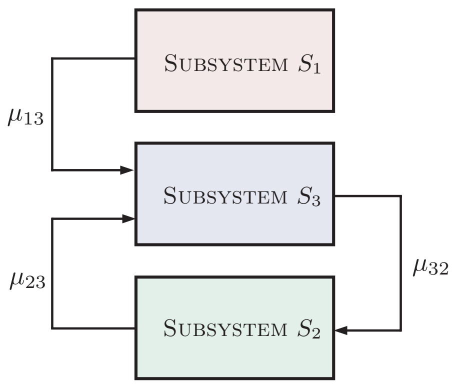

Figure 3: Conceptual scheme of the partitioned Barcelona DWN

- the dynamical system model in (10) split in 3 subsystems;
- the physical constraints (5) and (6) for each subsystem;
- a demand forecasting algorithm (presented in Section 2.3.4);
- a multi-objective cost function including the objectives (1), (2), and (3) and scarlarised using weights. The function can be written as

$$J(t) = \sum_{i=0}^{H_u-1} f_1(t+i|t) + \sum_{i=1}^{H_p} f_2(t+i|t) + \sum_{i=0}^{H_u-1} f_3(t+i|t), \tag{12}$$

  where H_p and H_u correspond to the prediction and control horizons, respectively, index t represents the current time instant while index i represents the predicted time along H_p. In this paper, the prediction horizon is related to the 24-hours demand seasonality. Regarding the value of H_u, it has been set to be equal to H_p, following the criterion of the DWN management company.

In order to explain and discuss the implementation of the solution sequence for the considered hierarchical-like DMPC strategy, denote C_i as the MPC controller related to the subsystem S_i (for $i \in \{1,\dots,3\}$), and μ_{ij} as the set of control actions u (manipulated flows, see (10)) going from S_i to S_j (for $j \in \{1,\dots,3\}$, $i \neq j$). Notice that μ_{ij} not only contains values of each component at time t but also all values over H_u, i.e., if $\mu_{ij} = \{u_a, u_b, \dots\}$, then[3]

$$\begin{aligned} u_a &\triangleq [u_a(t|t) \quad u_a(t+1|t) \ \dots \ u_a(t+H_u-1|t)]^T, \\ u_b &\triangleq [u_b(t|t) \quad u_b(t+1|t) \ \dots \ u_b(t+H_u-1|t)]^T, \\ &\vdots \end{aligned} \tag{13}$$

with $u_a(t+i|t)$ denoting the value of u_a at time $t+i$ (over the control horizon) given t. Once introduced this notation and according to the scheme in Figure 3, the particular sets μ_{ij} for the

[3] With some abuse of the notation, the elements of vector u are denoted with the corresponding discrete-time dependence in order to differentiate the vector from its components.

case study of this paper are defined as

$$\begin{aligned}\mu_{13} &= u_6,\\ \mu_{23} &= [u_{20}, u_{21}]^T,\\ \mu_{32} &= [u_{18}, u_{32}, u_{34}, u_{40}, u_{47}, u_{56}, u_{60}]^T.\end{aligned}$$

According to [11], the *pure* hierarchical control scheme determines a sequence of information distribution among the subsystems, where top-down communication is available from upper to lower levels of the hierarchy. Note that, despite the subsystems coupling (given by the shared links), the main feature of the pure hierarchical control approach relies on the unidirectionality of the information flow between controllers.

Looking at Figure 4, where the directions of sets μ_{ij} are graphically shown, it is possible to realise that the set μ_{32} (red-dashed line in the figure) breaks the mentioned unidirectional flow between MPC controllers. This fact implies that the standard hierarchical control scheme for partitioned LSS cannot be straight applied. To solve this situation and design a DMPC strategy, a hierarchical-like DMPC approach is proposed and conveniently implemented. This strategy follows the hierarchical control philosophy and the sequential way of solving the optimisation subproblems of the corresponding MPC controllers but also considering the appearance of bidirectional information flows. For this purpose, additional constraints and heuristics are taken into account in order to cope with the feature of having the double direction in the flow of information between some of the controllers. In particular, Figure 4 presents the considered hierarchy for the case study of this paper, where the controller at the upper level determines the values of variables shared with controllers at lower level. Notice that Figure 4 also shows why the pure hierarchical control approach cannot be employed since the MPC controller C_2 shares bidirectional information with C_3.

Therefore, the proposed solution sequence of the described hierarchical-like control problem for the aggregate model of the Barcelona DWN at each time step $t \in \mathbb{Z}_{\geq 1}$ is the following:

- C_3 computes the control actions of S_3 and sets μ_{13} and μ_{23}. Set μ_{32} is considered as a set of *virtual demands*[4] within the controller C_3.
- C_1 computes the control actions of S_1 considering μ_{13} as a set of virtual demands.
- In parallel, C_2 computes the control actions of S_2 considering μ_{23} as a set of virtual demands. Additionally, C_2 also computes μ_{32} to be used as a set of virtual demands for C_3 at time step $t + 1$.

Remark 4.1 *Notice that in the proposed DMPC scheme, at the first time step ($t = 1$), the initial values of the control actions belonging to set μ_{32} are not available. Those values can be obtained by solving a constraint satisfaction problem (CSP) defined by the models and constraints of subsystems S_2 and S_3 through the algorithm proposed in [19]. The solution of this CSP provides feasible control actions for the set μ_{32}, which allows starting the solution sequence described above. For subsequent time steps, values of μ_{32} take values computed by C_2 in the previous time step,*

[4] Consider two subsystems S_1 and S_2, which share a set of manipulated flows μ_{12}. According to the notation employed in the paper, those flows come from S_1 to S_2. If the solution sequence of optimisation subproblems — defined by the pre-established hierarchical order — determines that μ_{12} is computed by the MPC controller of S_1, then flows in μ_{12} are considered as *virtual demands* in the controller related to S_2 since their value are now imposed in the same way as the water demands.

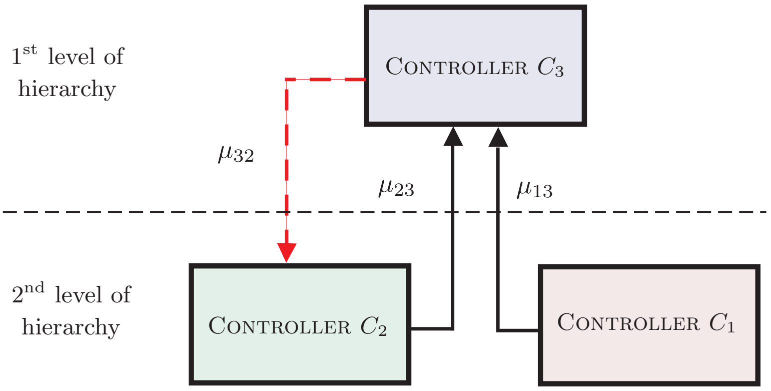

Figure 4: Hierarchy of MPC controllers C_i. Their solution sequence is top-down

i.e., the elements belonging to those sets at time t are now assigned as (see (13)*)*

$$u = \begin{bmatrix} u(t+1|t-1) \\ \vdots \\ u(t+H_u-1|t-1) \\ u(t+H_u-1|t-1) \end{bmatrix}. \tag{14}$$

5 Results

The results obtained by using the proposed DMPC strategy, presented in Section 4, are compared with those obtained when a CMPC strategy is used. The model parameters and measured disturbances (demands) have been supplied by AGBAR. Demands data correspond to the consume of drinking water of the city of Barcelona during the year 2007. Different scenarios are considered by modifying some controller parameters corresponding to different prioritisations of the control objectives. These parameters are the safety volume, denoted as β, and the weight matrices in the cost function (12). Regarding β, it has been set to the following values:

(a) 80% of x^{max}, that is denoted as $\mu = 0.8\, x^{max}$. This value is purely illustrative to show the effectiveness of the MPC controller;

(b) 20% over the minimum tank volumes requested to satisfy the demands[5] (except for tanks x_5, x_6 and x_8 in Figure 2, since they are considered as sources due to their strategic management requirements and network location). This second vector of safety volumes, denoted as η, is more convenient since it keeps the volumes of the tanks as low as possible, satisfying the demands at each time instant. These minimum volumes are taken from previous studies reported in [16].

In particular, consider $\Omega = (\omega_\alpha, \omega_x, \omega_{\Delta u})$ as the 3-tuple of weights associated to the matrices $W_\alpha = \omega_\alpha I$, $W_x = \omega_x I$ and $W_u = \omega_u I$ used in (1), (2) and in (3), respectively. Thus, the following scenarios have been stated:

[5] The minimum volume to satisfy the water demand is based on considering the worst-case scenario, i.e., the minimum volume within tanks that is required to satisfy the maximum network demand with no inflow. Some extra percentage of this volume can be added in order to take into account demand forecast inaccuracy (safety amount).

Table 2: Computation time and performance comparisons

Scenario	CMPC		DMPC	
	$\sum f_1$	Total Time	$\sum f_1$	Total
Scenario 1	49.86	360.02	50.06	253.12
Scenario 2	52.95	373.25	52.99	255.16
Scenario 3	49.94	395.18	50.46	264,55
Scenario 4	49.84	394.24	50.08	279.03
Scenario 5	52.93	396.12	53.01	274.19
Scenario 6	49.97	488.79	50.52	339.84

- **Scenario 1**: $\beta = \mu$ and $\Omega = (1, 0.1, 10^{-3})$;
- **Scenario 2**: $\beta = \mu$ and $\Omega = (1, 1, 10^{-3})$;
- **Scenario 3**: $\beta = \mu$ and $\Omega = (1, 0.1, 0.1)$;
- **Scenario 4**: $\beta = \eta$ and $\Omega = (1, 0.1, 10^{-3})$;
- **Scenario 5**: $\beta = \eta$ and $\Omega = (1, 1, 10^{-3})$;
- **Scenario 6**: $\beta = \eta$ and $\Omega = (1, 0.1, 0.1)$.

All results have been obtained considering four days real-demand scenarios (with 1 hour of sampling time), and $H_p = H_u = 24$. The network has been simulated using the same model than the MPC controller but fed with real water demands. The network model has been calibrated and validated using real data coming from AGBAR databases. All simulations were performed in MATLAB® 7.2 implementations running on an Intel® Core™2, 2.4 GHz machine with 4Gb RAM.

The hierarchical-like DMPC controller is compared with a CMPC in the considered scenarios. The computational burden of each controller was determined as the time required for the QP solver in obtaining the solution when the CMPC is implemented while as the sum of the time used by all the MPC controllers in the DMPC scheme. Table 2 gathers the simulation time (in s) for all the mentioned scenarios. This time regards to the time used for solving the optimisation problem only. As the way of showing the minimum loss of performance, Table 2 also presents the total economic cost related to the fourth simulation day. Notice that the simulations do not consider a warm starting so the first simulation day corresponds to a transient of the system behaviour. The behaviour of the rest of the days corresponds with a steady state (taking into account the cyclic pattern of the demands). In order to clearly illustrate this feature, Table 3 collects the complete information of the economic costs for Scenario 1, considering the discrimination by water and electrical costs (pumping) for each control scheme. These costs have been obtained by replacing the computed optimal actions (flows) and the expression of the economic costs in (1). It is also important to highlight that the economic costs collected in Tables 2 and 3 are given in *economic units* rather than in real values (Euro) due to confidentiality reasons. Additionally, Figure 5 shows a comparative of the evaluation of the cost function $J(t)$ in (12) for both control strategies (CMPC and DMPC), without considering the first simulation day.

Notice from Table 2 that the loss of performance given when using the DMPC strategy is never greater than 2%, what is a remarkable result given the reduction of the computation time, which can achieve up to 35% improvement. Thus, despite the DMPC approach inevitably leads to a small loss of performance, the benefits in terms of time and computational load are significant enough. In this particular application, the CMPC could also satisfy the real-time

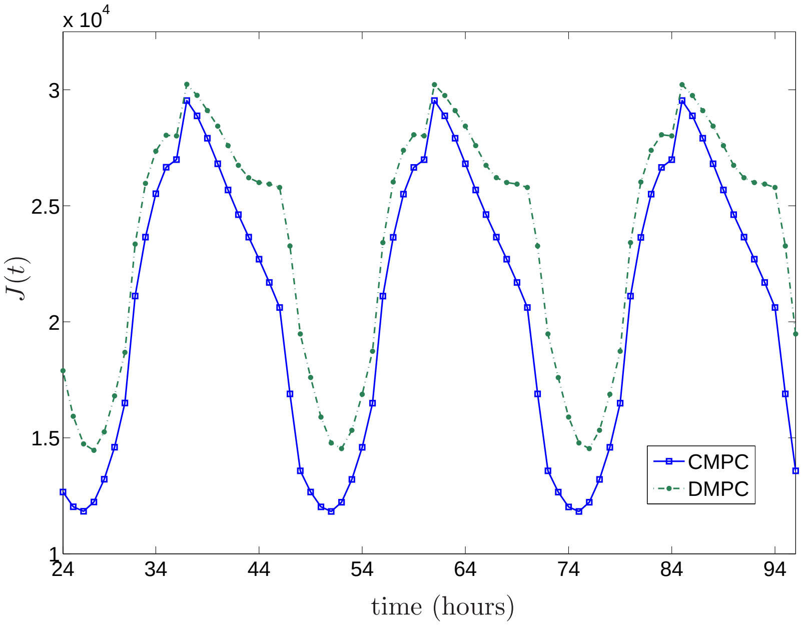

Figure 5: Cost function evaluation for CMPC and DMPC strategies

Table 3: Complete discrimination of economic costs for the Scenario 1

Day	CMPC			DMPC		
	Water	Electric	Total	Water	Electric	Total
1	29.601	22.107	51.709	44.49	18.69	63.84
2	27.733	22.141	49.874	31.99	18.07	50.06
3	27.722	22.138	49.861	31.99	18.07	50.06
4	27.722	22.138	49.860	31.99	18.07	50.06

constraint since the sampling time is one hour. Thus, the main motivation for using DMPC in this application would not be the improvement in computation but the scalability and the potential change adaptability easiness that the control strategy could offer. In fact, according to discussions with the AGBAR company, the main reason for using a DMPC approach in the case study of this paper, additionally to the easier maintenance of the (sub)system models, is that it allows replacing the current legacy control in multiple steps, where the DMPC is implemented on a selected network part only at each step. This ability is important for practical application and maintenance, which allows moving some part of the network to the current legacy control when some malfunction/fault is detected without stopping the supervisory MPC controller.

Regarding Table 3, notice that economic costs reach their steady state after the second day. Moreover, it is worthy to highlight the inverse behaviour of the electrical costs between the CMPC and DMPC cases. In global terms, the performance given by the DMPC is slightly worse, but this consideration should be done with the total cost. When the system is controlled by employing DMPC, the subsystems have no information of the water costs related to external sources. This fact explains that the optimisation over-emphasises the reduction of pumping costs inside the subsystem. By contrast, CMPC has the information of all water costs so it optimizes

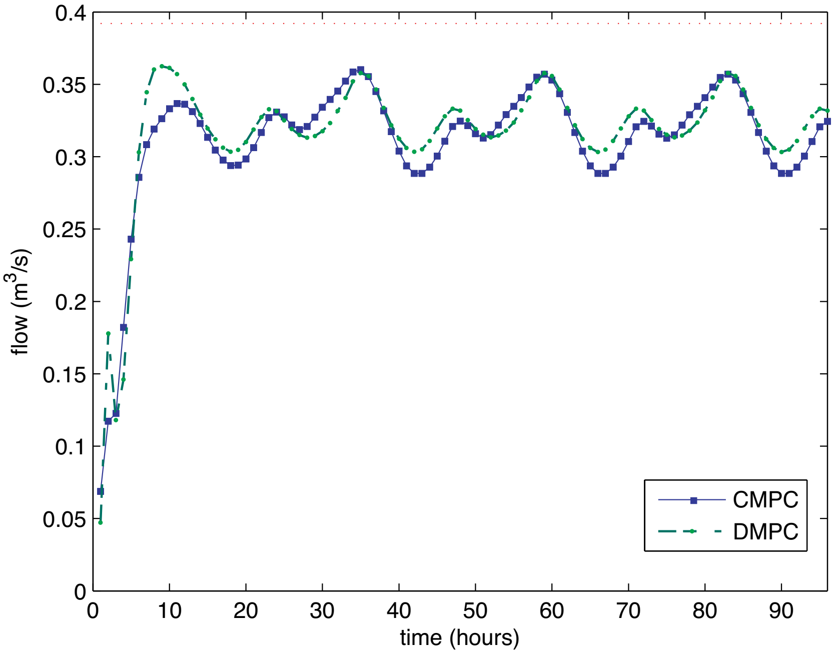

Figure 6: Computed flow related to valve 54 (VBMC)

this factor but *moves* the water inside the network, incrementing the electric costs. Thus, CMPC strategy selects the optimal source regarding water costs while the proposed DMPC approach emphasises local objectives (pumping costs) but is not able to minimize the water costs since it requires a global vision of the system.

In particular, the results show that with a suitable tuning of the MPC controller, no matter the topology considered (centralised or decentralised), the behaviour for some volumes and manipulated flows remain almost the same. This fact can be seen in Figures 6 and 7, where the flow through the valve 54 and the volume in tank 10, respectively, are depicted when they are computed for a DMPC and CMPC schemes. Notice for instance that volume behaviours are quite similar and, given the weight tuning of Scenario 3, they smoothly oscillate around the desired reference, thanks to a convenient compromise of the safety and smoothness terms in the cost function.

6 Conclusions

In this paper, a DMPC strategy for DWN has been proposed. The DWN is decomposed in a set of subnetworks using a partitioning algorithm that makes use of the topology of the network, the information about the actuator usage and heuristics. A hierarchical structure related to the order of execution of the DMPC controllers allows one to take into account global network constraints. A comparative study between the CMPC and DMPC approaches has been developed using as case study the aggregate model of the Barcelona DWN. Results have shown that the partition algorithm, helped by an analysis of the system topology and heuristics, yields a proper segmentation of the whole network without overlapping models. The performances of CMPC and DMPC schemes were compared in terms of economical benefits and computational burden.

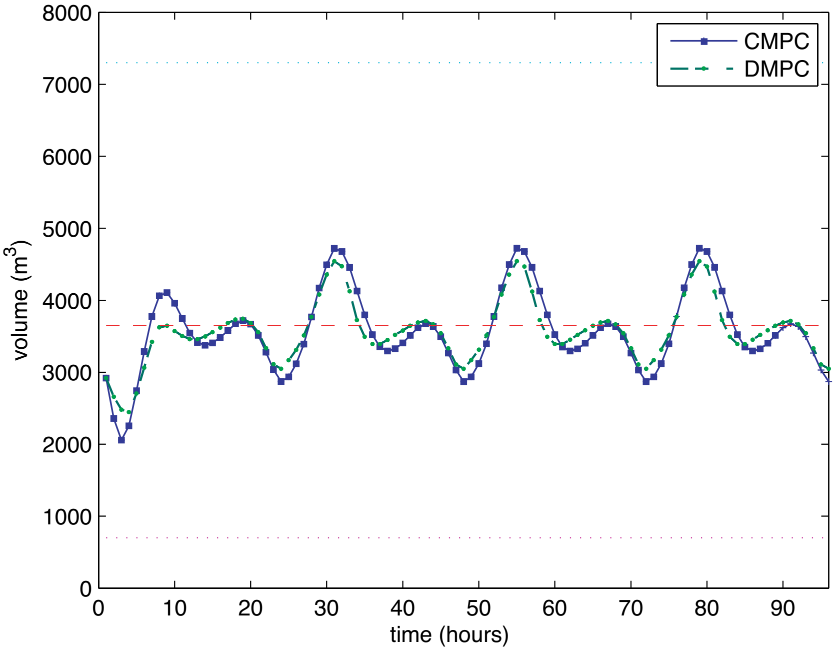

Figure 7: Resultant volume related to tank 10 (d130BAR)

Results have shown the effectiveness of the DMPC strategy in the important reduction of such computational burden despite the loss of performance of the control scheme, which in turn, has resulted to be quite small.

In future research, issues related to the the possibility of allowing the subsystems to overlap will be investigated, in addition to topics of stability of the proposed control approach. From the point of view of the control problem, feasibility issues are not discussed, being this a topic of ongoing research. However, the feasibility of the solution is guaranteed given the topological design of the network, since it has been properly dimensioned to supply the water to each demand sector from the available sources. On the other hand, the proposed partitioning approach provides a first automatic subsystem decomposition but since the partitioning result depends on the value of ε, which should be selected by trial and error. A further improvement would consists in selecting the value of ε through optimisation by establishing criteria to decide whether or not the obtained system decomposition is adequate. Moreover, some restrictions could be added regarding the complete assignment of inputs and states to the different subsystems. Finally, different schemes of hierarchical control and in general, strategies that deal with partitioned systems should be designed and further investigated for being tested over this type of networked systems.

Acknowledgements

This work has been supported by the WATMAN Project (Ref. DPI2009-13744) of the Spanish Science and Technology Ministry, the DGR of Generalitat de Catalunya (SAC group Ref. 2009/SGR/1491) and the European Commission under project “WIDE - Decentralized and Wireless Control of Large-Scale Systems”, contract number FP7-IST-224168.

References

[1] M. Brdys and B. Ulanicki, Operational Control of Water Systems: Structures, algorithms and applications. UK: Prentice Hall International, 1994.

[2] M. Marinaki and M. Papageorgiou, Optimal Real-time Control of Sewer Networks. Secaucus, NJ (USA): Springer, 2005.

[3] P. V. Overloop, Model Predictive Control on Open Water Systems. Delft, The Netherlands: Delft University Press, 2006.

[4] A. Bemporad and D. Barcelli, "Decentralized model predictive control," in Networked Control Systems, A. Bemporad, W. Heemels, and M. Johansson, Eds., Berlin Heidelberg, 2010, pp. 149–178.

[5] T. Keviczky, F. Borrelli, and G. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems," Automatica, vol. 42, no. 12, pp. 2105–2115, December 2006.

[6] J. B. Rawlings and B. T. Stewart, "Coordinating multiple optimization-based controllers: New opportunities and challanges," Journal of Process Control, vol. 18, no. 9, pp. 839–845, October 2008.

[7] R. Negenborn, B. De Schutter, and J. Hellendoorn, "Multi-agent model predictive control for transportation networks: Serial vs. parallel schemes," Engineering Applications of Artificial Intelligence, vol. 21, no. 3, pp. 353–366, April 2008.

[8] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, "Distributed MPC strategies with application to power system automatic generation control," IEEE Tran on Control Systems Technology, vol. 16, no. 6, pp. 1192–1206, November 2008.

[9] R. Scattolini, "Architectures for distributed and hierarchical Model Predictive Control: A review," Journal of Process Control, vol. 19, no. 5, pp. 723–731, May 2009.

[10] D. Barcelli, A. Bemporad, and G. Ripaccioli, "Hierarchical multi-rate control design for constrained linear systems," in Proc. 49th IEEE Conf. on Decision and Control, to appear, 2010.

[11] D. Šiljak, Decentralized control of complex systems. Academic Press, 1991.

[12] J. Lunze, Feedback Control of Large-Scale Systems. Great Britain: Prentice Hall, 1992.

[13] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," Automatica, vol. 38, no. 1, pp. 3–20, 2002.

[14] J. Quevedo, V. Puig, G. Cembrano, and J. Blanch, "Validation and reconstruction of flow meter data in the Barcelona water distribution network," Control Engineering Practice, vol. 11, no. 6, pp. 640–651, June 2010.

[15] C. Ocampo-Martinez, V. Puig, G. Cembrano, R. Creus, and M. Minoves, "Improving water management efficiency by using optimization-based control strategies: the Barcelona case study," Water Science & Technology: Water supply, vol. 9, no. 5, pp. 565–575, 2009.

[16] E. Caini, V. Puig, and G. Cembrano, "Development of a simulation environment for drinking water networks: Application to the validation of a centralized MPC controller for the Barcelona case study," Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Tech. Rep. IRI-TR-03-09, April 2009.

[17] V. Fambrini and C. Ocampo-Martinez, "Modelling and decentralized Model Predictive Control of drinking water networks," Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Tech. Rep. IRI-TR-04-09, April 2009.

[18] M. E. Sezer and D. Siljak, "Nested ε-decomposition and clustering of complex systems," Automatica, vol. 22, no. 3, pp. 321–331, 1986.

[19] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics. London: Springer-Verlag, 2001.