# Chapter 30
# Multi-layer decentralized MPC of large-scale networked systems

C. Ocampo-Martinez, V. Puig, J.M. Grosso, S. Montes-de-Oca

**Abstract** In this chapter, a multi-layer decentralized model predictive control (ML-DMPC) approach is proposed and designed for its application to large-scale networked systems (LSNS). This approach is based on the periodic nature of the system disturbance and the availability of both static and dynamic models of the LSNS. Hence, the topology of the controller is structured in two layers. First, an upper layer is in charge of achieving the global objectives from a set $\mathcal{O}$ of control objectives given for the LSNS. This layer works with a sampling time $\Delta t_1$, corresponding to the disturbances period. Second, a lower layer, with a sampling time $\Delta t_2$, $\Delta t_1 > \Delta t_2$, is in charge of computing the references for the system actuators in order to satisfy the local objectives from the set of control objectives $\mathcal{O}$. A system partitioning allows to establish a hierarchical flow of information between a set $\mathcal{C}$ of controllers designed based on model predictive control (MPC). Therefore, the whole proposed ML-DMPC strategy results in a centralized optimization problem for considering the global control objectives, followed of a decentralized scheme for reaching the local control objectives. The proposed approach is applied to a real case study: the water transport network of Barcelona (Spain). Results obtained with selected simulation scenarios show the effectiveness of the proposed ML-DMPC strategy in terms of system modularity, reduced computational burden and, at the same time, the admissible loss of performance with respect to a centralized MPC (CMPC) strategy.

---

C. Ocampo-Martinez, V. Puig, J.M. Grosso
Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Technical University of Catalonia, Barcelona, Spain, e-mail: {cocampo,vpuig,jgrosso}@iri.upc.edu

S. Montes-de-Oca
Automatic Control Department, Technical University of Catalonia, Barcelona, Spain, e-mail: saul.montes.de.oca@upc.edu

## 30.1 Introduction

Large-scale networked systems (LSNS) are very common in the modern societies to transport for example, water, electricity, gas, oil, among others. Thus, their optimal management is a subject of increasing interest that due to its social, economic and environmental impact. The leading control technique for the management of LSNS is model predictive control (MPC) [1, 11]. The main reason for its success is due to after obtaining the network dynamical model, the MPC design just consists in expressing the 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 and follows multiple approaches reported in the literature; see, e.g., [2, 6], among many others.

Traditional MPC procedures assume that all available information is centralized. In fact, a global dynamical model of the system must be available for control design. Moreover, all measurements must be collected in one location to estimate all states and to compute all control actions. However, when considering LSNS, these assumptions usually fail to hold, either because gathering all measurements in one location is not feasible, or because the computational needs of a centralized strategy are too demanding for a real-time implementation. This fact might lead to a lack of scalability. Subsequently, a model change would require the re-tuning of the centralized controller. Thus, 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 either *decentralized* or *distributed* MPC techniques, where networked local MPC controllers are in charge of the control of part of the entire system. Those techniques have became one of the hottest topics in control during the early 21$^{st}$ century, opening the door to the research towards solving new open issues and related problems of the strategy. Many works have been published in this area; see, e.g., [4, 14, 15, 17], among others.

In order to apply either decentralized or distributed MPC approaches to LSNS, 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 decentralized control of large-scale systems, see, e.g., [16] or [5]. The decomposition of the system in subsystems could be carried out during the modeling of the process by identifying subsystems as parts of the system on the basis of physical insight, intuition or experience. But, when a large-scale complex system with many states, inputs and outputs is considered, it may be difficult, even impossible, to obtain partitions by physical reasoning. A more appealing alternative is to develop systematic methods, which can be used to decompose a given system by extracting information from its structure, which is represented as a graph. Then, this structural information can be analyzed by using methods coming from graph theory. Consequently, the problem of system decomposition into subsystems leads to the problem of graph partitioning, i.e., the decomposition of graph into subgraphs. However, the development of graph partitioning algorithms within the framework of decentralized or distributed MPC is still very incipient and available methods are currently quite limited.

In this chapter, a multi-layer decentralized MPC (ML-DMPC) approach is proposed and designed for its application to LSNS. This approach is based on the assumption that the disturbances affecting the system have a periodic behavior. Moreover, the approach is also based on the the availability of both static and dynamic models of the LSNS. Hence, the optimization problem behind the controller is defined to have two layers: the former or *upper* layer, working with a sampling time $\Delta t_1$ related to the period of the system disturbances, is in charge of achieving the global objectives from a set $\mathcal{O}$ of control objectives to be fulfilled by the networked system. The latter layer, also named the *lower* layer, with a sampling time $\Delta t_2$, $\Delta t_1 > \Delta t_2$, is in charge of computing the references for the system actuators in order to satisfy the local objectives from the set of control objectives $\mathcal{O}$. The system partitioning allows to establish a hierarchical flow of information between the set $\mathcal{C}$ of local MPC controllers. Therefore, the whole proposed ML-DMPC strategy results in a centralized optimization problem for considering the global control objectives, followed by a decentralized scheme for reaching the local control objectives. The advantage of this hierarchical-like DMPC approach is the simplicity of its implementation given the absence of negotiations among controllers. To apply the proposed DMPC approach, the network is decomposed into subsystems by using a novel automatic decomposition algorithm reported in [8], which is based on graph partitioning. The proposed ML-DMPC approach is applied to a real case study: the water transport network of Barcelona (Spain). Results obtained with selected simulation scenarios show the effectiveness of the proposed ML-DMPC strategy in terms of system modularity, reduced computational burden and, at the same time, the admissible loss of performance with respect to a centralized MPC (CMPC) strategy.

This chapter is structured as follows: Section 30.2 describes boundary conditions on considered system, control objectives, and constraints. Section 30.3 describes the ML-DMPC approach. Section 30.4 illustrates the proposed approach in the aforementioned case study. Finally, conclusions and some directions for further research are reported in Section 30.5.

## 30.2 Boundary Conditions

### 30.2.1 Control-oriented Modeling Framework

Before establishing the fundamentals of the control-oriented modeling framework proposed in this chapter, the statement of the general framework for controlling a LSNS is discussed. The control system architecture of a LSNS may be defined in two levels as shown in Figure 30.1. The upper level consists in a supervisory controller that is in charge of the global control of the networked system, establishing references for regulatory controllers (of PID type) at the lower level. Regulatory

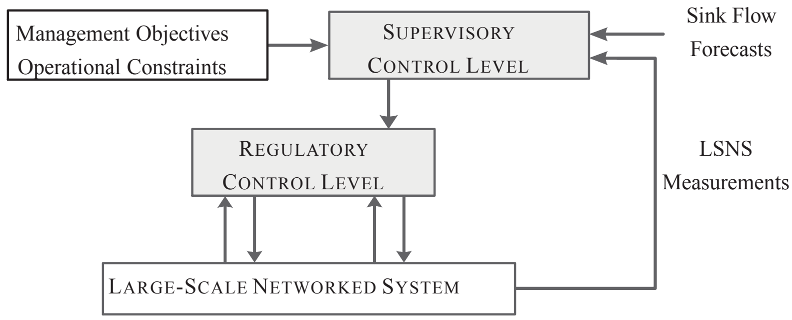



**Figure 30.1:** Control architecture for a LSNS

controllers hide the non-linear behavior of the system to the supervisory controller. This fact may allow the supervisory level to use a control-oriented model.

In a general way, a proper control-oriented model of a given system is defined such that it captures its main behaviors, being as simple as possible in order to save computational burden when such model is used for control design purposes. This chapter considers the use of the control-oriented model with a model-based optimization-based control strategy with constraints. This latter implies not only dynamic and static equations in the mathematical expression of the behavior of the system, but also inequality constraints may be added. In general, these inequalities are associated to bounds in the operational ranges of the physical variables of the system (inputs, states, and outputs). However, some of those inequality constraints may also relate system variables between them together with system disturbances.

The framework of control-oriented modeling of LSNS that is proposed in this chapter relies on the concept of *flow* between or through the constitutive elements of the system. In this framework, the flow is understood in the sense of movement of the raw material related to the use or function of the networked system. In order to have a model structure where the flow concept has sense, it is necessary to define a set of basic elements to be associated with the physical LSNS.

**Storage Element:** As its name indicates, this element represents the fact of storing the material/data flow, what implies a *volume* given in discrete time by the difference equation

$$x(k+1) = x(k) + \Delta t\,(q_{\text{in}}(k) - q_{\text{out}}(k)), \tag{30.1}$$

where $x$ denotes the stored flow volume, $q_{\text{in}}$ and $q_{\text{out}}$ denote the net inflow and outflow, respectively; $\Delta t$ is the considered sampling time and index $k \in \mathbb{Z}_{\geq 0}$ represents the discrete time instant. Notice that (30.1) adds the dynamic nature to the control-oriented model of the whole LSNS. Moreover, this element is not defined to store infinity quantity of flow, what implies a working regime bounded by the storing constraints

$$x_{\min} \leq x(k) \leq x_{\max}, \qquad \forall k, \tag{30.2}$$

where $x_{\min}$ and $x_{\max}$ are the minimum and maximum volume that the element is able to store, respectively.

**Node Element:** This element, also called *junction*, corresponds to a point where flows are either propagated or merged. Propagation means that the node has one inflow and some outflows. Merging means that two or more inflows are merged into a larger outflow. Thus, two types of nodes may be considered:

- Nodes with one inflow and multiple outputs (splitting nodes), i.e.,

$$q_{\text{in}}(k) = \sum_i q_{\text{out},i}(k). \tag{30.3}$$

- Nodes with multiple inputs and one output (merging nodes), i.e.,

$$\sum_j q_{\text{in},j}(k) = q_{\text{out}}(k). \tag{30.4}$$

Mixed nodes can be described from the basic ones described above, i.e., complex nodes with several inflows and outflows may be defined. Notice that this element would add static relations to the control-oriented model of the whole LSNS. However, some LSNS do not show the behavior modeled by nodes, hence static relations are not always present in the control-oriented model.

**Flow source:** This element provides the raw material that flows through the network. It may be considered either:

- as an exogenous inflow to the networked system. In that case, constraints such as

$$q_{\min,\Lambda_i} \leq q_{\Lambda_i}(k) \leq q_{\max,\Lambda_i} \tag{30.5}$$

  might be considered, where $q_{\Lambda_i}$ denotes the inflow from the $i$-th source; $q_{\min,\Lambda_i}$ and $q_{\max,\Lambda_i}$ correspond to minimum and maximum inflow, respectively. For simplicity and compactness of the control-oriented model, constraints in (30.5) are associated to flow handling elements (described below) directly connected to sources;
- or as an external storage element, what implies an expression for its volume $x_\Lambda(k)$ such as in (30.1), with the associated constraint such as in (30.2).

**Sink:** In this framework, a sink is the element where the flow goes to. From a general point of view, sinks are related to the measured disturbances of the system since they ask for flow according to a given profile. The networked system should be managed in such a way that those elements receive the flow they request.

**Link:** This element, also called *arc*, represents the general way of connecting two elements which share a flow, e.g., a source with a node, an storage element with a sink, etc. The flow through these elements can be constrained by the range

$$q_{\min} \leq q(k) \leq q_{\max}, \qquad \forall k, \tag{30.6}$$

where $q_{\min}$ and $q_{\max}$ are the minimum and maximum flow through a link, respectively.

**Flow Handling Element:** In this framework, this element manipulates flow either between storage elements or between a storage element and a node, and viceversa. Hence, flow handling elements are links where the flow is manipulated. Handling elements between storage elements and sinks as well as between nodes and sinks are not considered since the flow handled has to be equal to the flow requested from the sink and, therefore, there is no place for different options. Notice that the flow through these elements is also constrained following (30.6).

*Remark 30.1.* Regarding storage elements, when their outflow is not manipulated, its expression corresponds with

$$q_{\text{out}}(k) = h(x(k)), \tag{30.7}$$

where $h$ should be determined according to the nature of the particular case study. Notice that this relation can be made more accurate (but also more complex) if $h$ is considered to be nonlinear, thus yielding nonlinear constrained control-oriented model. This latter can be seen considering (30.7) and rewriting the right-hand side of (30.6) as

$$q(k) \leq \min\{q_{\max}, h(x(k))\}, \qquad \forall k. \tag{30.8}$$

Moreover, in the scenario where $x_{\min} \neq 0$ and the outflow of the storage element is manipulated, the left-hand side of (30.6) should be rewritten as

$$\min\{q_{\min}, h(x(k))\} \leq q(k), \qquad \forall k, \tag{30.9}$$

which also implies a non-convex constraint within the control-oriented model of the LSNS. $\Diamond$

Consider a given LSNS being represented as the interconnection of $n_x$ storage elements, $n_u$ flow handling elements, $n_d$ sinks and $n_q$ intersection nodes. The $n_\alpha$ sources are considered as inflows. Stating the volume in storage elements as the state variable and the flow through handling elements as the manipulated inputs, an LSNS may be generally described in state-space form by the following linear discrete-time dynamic model:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \boldsymbol{\Gamma}\boldsymbol{\upsilon}(k), \tag{30.10a}$$

$$\mathbf{0} = \mathbf{E}_x\mathbf{x}(k) + \mathbf{E}_\upsilon\boldsymbol{\upsilon}(k), \tag{30.10b}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ corresponds to the state vector (stored volumes), $\boldsymbol{\Gamma} \triangleq [\mathbf{B} \quad \mathbf{B}_p]$, and $\boldsymbol{\upsilon}(k) := [\mathbf{u}(k)^{\mathrm{T}} \quad \mathbf{d}(k)^{\mathrm{T}}]^{\mathrm{T}}$. In turn, $\mathbf{u} \in \mathbb{R}^{n_u}$ is the vector of control inputs (manipulated flows), and $\mathbf{d} \in \mathbb{R}^{n_d}$ corresponds to the vector of measured disturbances (flows to sinks). Moreover, $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$, $\mathbf{B}_p \in \mathbb{R}^{n_x \times n_d}$, are state-space system matrices for mass flow balances in storage elements (30.10a), and $\mathbf{E}_x \in \mathbb{R}^{n_q \times n_x}$ and $\mathbf{E}_\upsilon \in \mathbb{R}^{n_q \times (n_u+n_d)}$ are matrices for static flow balances in nodes (30.10b). All vectors and matrices are dictated by the network topology. Notice that $\mathbf{E}_x = 0$ when outflows from storage elements are manipulated. In general, states and control inputs are subject to constraints of the form

$$\mathbf{x}_{\min} \leq \mathbf{x}(k) \leq \mathbf{x}_{\max}, \quad \forall k, \tag{30.11a}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}(k) \leq \mathbf{u}_{\max}, \quad \forall k, \tag{30.11b}$$

where $\mathbf{x}_{\min} \in \mathbb{R}^{n_x}$ and $\mathbf{x}_{\max} \in \mathbb{R}^{n_x}$ denote the vectors of minimum and maximum volumes, respectively, while $\mathbf{u}_{\min} \in \mathbb{R}^{n_u}$ and $\mathbf{u}_{\max} \in \mathbb{R}^{n_u}$ denote the vectors of minimum and maximum flows through flow handling elements, respectively.

*Remark 30.2.* Notice that manipulated flows may be defined as bidirectional flows. This means that minimum flows of these manipulated links may be negative. In order to cope with this situation, a bidirectional link can be replaced with two separate unidirectional links with null minimum flow, associated with each direction of the original link. Although this approach simplifies the control setup, it might add complexity to the optimization problem related to the optimization-based controller since the number of optimization variables gets higher. ◊

### 30.2.2 Model Decomposition

Once the control-oriented model is stated, it is important to determine the objective of performing the partition of the networked system no matter what control strategy is followed. In this aspect, the availability of centralized information is fundamental. When all the information about the whole set of system variables is available, the partitioning gains sense from the point of view of modularity of the control architecture and the reduction of computational burden. However, when this global information is not fully available, a control topology based on the partition of the system should be designed given the *physical dispersion* the LSNS might show.

Consider again an LSNS formed by the interconnection of several elements from those proposed beforehand. The way they are interconnected is a key factor for performing the partitioning since it determines the type of variables the resultant subsystems would share. Thus, if for instance the outflow for an storage element from a subsystem $S_a$ is not manipulated and the corresponding flow is just the inflow of a subsystem $S_b$, then the shared variable corresponds to a system state. Otherwise, it would be a control input.

This chapter considers the partitioning algorithm proposed in [8]. This algorithm, based on graph partitioning, aims at decomposing (30.10) into subsystems. In order to do so, the graph representation of (30.10) is determined by using the system topology, what yields its *incidence matrix* $I_M$. This matrix describes the connections (edges) between the graph vertices (here represented by storage elements, sources, sinks, and nodes). Once $I_M$ is obtained from the system digraph, the problem of the decomposition into subsystems is formulated in terms of partitioning the corresponding graph into subgraphs such that all subgraphs have nearly the same number of vertices and there exist few connections between subgraphs.

These features, motivated by the posterior design of a decentralized/distributed control strategy, guarantee that the obtained subgraphs have a similar size, fact that balances computations between the controllers and allows minimizing communications/interactions between them.

Thus, the overall system (30.10) is assumed to be decomposed in $M \triangleq |\mathcal{N}|$ subsystems collected in the set $\mathcal{N}$, which are not overlapped, output decentralized and input coupled (therefore, $\mathbf{E}_x = 0$). The model of the $i$-th subsystem is stated below for $i \in \{1,\dots,M\}$ as

$$\mathbf{x}_i(k+1) = \mathbf{A}_i\mathbf{x}_i(k)+\mathbf{\Gamma}_i\boldsymbol{\upsilon}_i(k)+\mathbf{B}_{\mathrm{sh},i}\boldsymbol{\mu}_i(k), \tag{30.12a}$$

$$\mathbf{0} = \mathbf{E}_{\upsilon,i}\boldsymbol{\upsilon}_i(k)+\mathbf{E}_{\mathrm{sh},i}\boldsymbol{\mu}_i(k), \tag{30.12b}$$

where $\mathbf{x}_i \in \mathbb{R}^{n_{x_i}}$ and $\boldsymbol{\upsilon}_i \in \mathbb{R}^{n_{u_i}+n_{d_i}}$ are the local states and inputs of the subsystem $S_i$, respectively, and $\boldsymbol{\mu}_i \in \mathbb{R}^{n_{\mu_i}}$ is the vector of shared inputs between $S_i$ and other subsystems. Moreover, $\mathbf{B}_{\mathrm{sh},i}$ and $\mathbf{E}_{\mathrm{sh},i}$ are matrices whose dimensions depend on the number of shared inputs of $S_i$. The decomposition should assure that $\sum_i n_{x_i} = n_x$, $\sum_i n_{u_i} = n_u$, $\sum_i n_{d_i} = n_d$ and $\sum_i n_{q_i} = n_q$. Matrices $\mathbf{A}_i$, $\mathbf{\Gamma}_i$, $\mathbf{E}_{\upsilon,i}$, are dictated by each subsystem topology. In the same way, the previously defined overall constraints (30.11) are partitioned for each $i$-th subsystem as

$$\mathbf{x}_{\min,i} \leq \mathbf{x}_i(k) \leq \mathbf{x}_{\max,i}, \quad \forall k, \tag{30.13a}$$

$$\mathbf{u}_{\min,i} \leq \mathbf{u}_i(k) \leq \mathbf{u}_{\max,i}, \quad \forall k. \tag{30.13b}$$

Moreover, it may occur that the $n_\alpha$ flow sources of the LSNS determine the amount of $M$ since the sinks (and therefore storage elements and nodes) related to each subsystem $S_j$, $j \in \{1,\dots,n_\alpha\}$ are only supplied by a unique source. Therefore, this topological dependency determines subsystems around a flow source, resulting to be a natural criterion for performing system decomposition. Thus, as seen in Figure 30.2, the initial LSNS might be decomposed in two stages. In the first stage, subsystems tied with flow sources are determined. From now on, these subsystems are called *anchored subsystems* (AS). It can be seen that there will be as many anchored subsystems as number of sources in the network. Remaining elements are associated in a resultant subsystem namely $\tilde{S}$, where storage elements might be fed from two or more flow sources. In the second stage, subsystems $\tilde{S}$ is now decomposed by following the algorithm proposed in [8]. Notice that, at this point, the shared connections of $\tilde{S}$ that correspond to inflows, my be considered as *pseudo-sources* of $\tilde{S}$. Therefore, depending on the management/control objectives related to the LSNS, it is possible to add some additional criteria to each AS outflow (or $\tilde{S}$ inflow). These criteria can be associated to a weighting factor $\omega$, which is related to each pseudo-source of $\tilde{S}$ and would be determined within the design of the control strategy for the LSNS (see Section 30.3 below). Notice that a second set of pesudo-sources would appear after performing the decomposition of $\tilde{S}$, but their treatment can follow the same procedure considered for the first set of pseudo-sources.

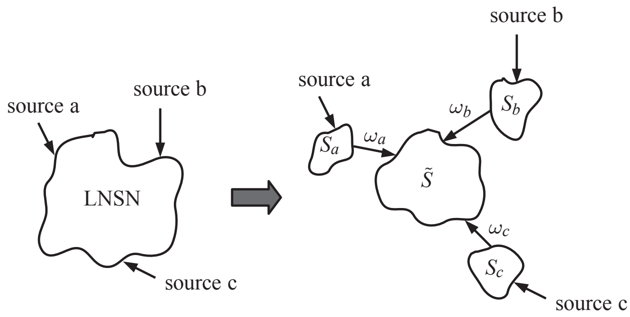



**Figure 30.2:** Scheme of LSBS partitioning ($n_\alpha = 3$)

### 30.2.3 MPC Problem Formulation

From the LSNS model in (30.10), let $\mathbf{u}(k:k+N_p-1)$ be the sequence of control input over a fixed-time prediction horizon $N_p$. Hence, the following problem is proposed.

**Problem 30.1.** An MPC controller design is based on the solution of the open-loop multi-objective optimization problem (OOP)

$$\min_{\mathbf{u}^*(k:k+N_p-1)} J(k) \triangleq \sum_{m=1}^{|\mathcal{O}|} \gamma_m J_m(\mathbf{u}(k:k+N_p-1), \mathbf{x}(k+1:k+N_p)), \tag{30.14a}$$

subject to system model (30.10), system constraints (30.11) over $N_p$, and a set of $n_c$ operative constraints given by management policies of the system and condensed on the form

$$\mathbf{G}_1\mathbf{x}(k+1:k+N_p) + \mathbf{G}_2\boldsymbol{\upsilon}(k:k+N_p-1) \leq \mathbf{g}, \tag{30.14b}$$

where $J(\cdot) : \mathbb{R}^{n_u(N_p-1)\times n_x N_p} \mapsto \mathbb{R}$ in (30.14a) is the cost function collecting all control objectives of the set $\mathcal{O}$ and $\gamma_m$ are positive scalar weights to prioritize the $m$-th control objective $\mathcal{O}_m \in \mathcal{O}$, particularly represented by $J_m$ within the whole cost function. Moreover, $\mathbf{G}_1 \in \mathbb{R}^{n_c \times n_x N_p}$, $\mathbf{G}_2 \triangleq [\mathbf{G}_{2,u} \quad \mathbf{G}_{2,d}] \in \mathbb{R}^{n_c \times \{n_u(N_p-1)+n_d(N_p-1)\}}$, and $\mathbf{g} \in \mathbb{R}^{n_c N_p}$. □

Assuming that Problem 30.1 is feasible, i.e., there is an optimal solution given by the sequence of control inputs $\mathbf{u}^*(k:k+N_p-1) \neq \emptyset$, and then the receding horizon philosophy sets

$$\mathbf{u}_{\text{MPC}}(\mathbf{x}(k)) \triangleq \mathbf{u}^*(k), \tag{30.15}$$

and disregards the computed inputs from $k+1$ to $k+N_p-1$, with the whole process repeated at the next time instant $k \in \mathbb{Z}_{\geq 0}$. Expression (30.15) is known in the MPC literature as *the MPC law* [6].

Besides, the decomposition of the original problem leads to design an MPC controller $C_i \in \mathcal{C}$, with $i = \{1,\dots,M\}$, for each of the $M$ subsystems. This fact also leads to split the cost function (30.14a). Therefore, each subsystem considers the local cost function

$$J_i(k) = \sum_{m=1}^{|\mathcal{O}|} \gamma_{m,i} J_{m,i}(\mathbf{u}_i(k:k+N_p-1), \mathbf{x}_i(k+1:k+N_p)), \tag{30.16}$$

where $m = \{1,\ldots,|\mathcal{O}|\}$, and $\gamma_{m,i}$ are scalar weights that prioritize local objectives within each subsystem. In the same way, operational constraints may be properly split along the subsystems and expressed as

$$\mathbf{G}_{1,i}\mathbf{x}_i(k+1:k+N_p) + \mathbf{G}_{2,i}\boldsymbol{\upsilon}_i(k:k+N_p-1) \leq \mathbf{g}_i. \tag{30.17}$$

## 30.3 Description of the Approach

### 30.3.1 Preliminary Assumptions

Once the control-oriented model is obtained and decomposed into subsystems, the natural step forward consists in designing the decentralized control strategy considering the given management policies and constraints. Before getting through the proposed methodology for designing such controllers based on predictive control, the following assumptions regarding the LSNS and its management are stated.

**Assumption 30.1** *All sinks can be supplied by at least one flow source through at least one flow path*[1].

**Assumption 30.2** *All sinks show a periodic flow request, whose period is* $T = \Delta t_1$.

**Assumption 30.3** *The set* $\mathcal{O}$ *of control objectives is defined as*

$$\mathcal{O} = \mathcal{O}_l \cup \mathcal{O}_g, \tag{30.18}$$

*where* $\mathcal{O}_l$ *corresponds with the set of local control objectives and* $\mathcal{O}_g$ *with the set of global control objectives. Moreover,* $m_l \triangleq |\mathcal{O}_l|$, $m_g \triangleq |\mathcal{O}_g|$, *and hence* $m_l + m_g = |\mathcal{O}|$.

Assumption 30.3 introduces the diversity on the nature of the control objectives of the LSNS. This fact determines the way the decentralized controller is designed since the fulfillment of a global objective from a local point of view should imply information from all the LSNS, fact that is avoided when the system partitioning is performed. Therefore, it is necessary to figure out how to *transform* the formulation of a global objective in a centralized control scheme towards the statement of a set of decentralized controllers $\mathcal{C}$ considering all the control objectives in $\mathcal{O}$ in a suitable way.

[1] A flow path is formed by a finite set of links, which may connect sources, nodes, sinks, and storage elements.

First of all, in order to develop this idea, the cost function related to the centralized MPC (CMPC) in Problem 30.1 can be rewritten as

$$J(k) = \sum_{j=1}^{m_g} \gamma_j J_j(\mathbf{u}(k:k+N_p-1), \mathbf{x}(k+1:k+N_p)) \\ + \sum_{m=1}^{m_l} \gamma_m J_m(\mathbf{u}(k:k+N_p-1), \mathbf{x}(k+1:k+N_p)). \quad (30.19)$$

The approach proposed in this chapter consists in designing a decentralized MPC scheme, where each controller $C_i \subset \mathcal{C}$ considers a newer version of (30.16) taking into account the structure of (30.19). Hence, the cost function related to each $C_i$ is written as

$$J_i(k) = \sum_{j=1}^{m_g} \hat{\gamma}_{j,i} \hat{J}_{j,i}(\mathbf{u}_i(k:k+N_p-1), \mathbf{x}_i(k+1:k+N_p)) \\ + \sum_{m=1}^{m_l} \gamma_{m,i} J_{m,i}(\mathbf{u}_i(k:k+N_p-1), \mathbf{x}_i(k+1:k+N_p)), \quad (30.20)$$

where $\hat{J}_{j,i}(\cdot)$ corresponds to the $j$-th global control objective properly expressed in order to reflect its influence in the local controller. Moreover, $\hat{\gamma}_{j,i}$ is a weight that prioritizes the global objectives that must be filled within the optimization problem.

Thus, the design of the entire control topology gives rise to a twofold optimization problem behind the general MPC topology. This twofold problem consists of two layers operating at different time scales: an *upper layer* works with a sampling time $\Delta t_1$, corresponding to the disturbance period. This layer is in charge of achieving the global objectives from a set $\mathcal{O}$ of control objectives given for the networked system. On the other hand, a *lower layer*, with a sampling time $\Delta t_2$, $\Delta t_1 > \Delta t_2$, is in charge of computing the references for the system actuators in order to satisfy the local objectives from the set of control objectives $\mathcal{O}$.

## 30.3.2 Upper Optimization Layer

This layer is designed to take into account the global control objectives in a proper way, i.e., considering information of the entire system in order to fulfill them. This layer is in charge of computing weights $\omega$ related to pseudo-sources and discussed in Section 30.2.2 (see Figure 30.2). These weights $\omega$ will determine the prioritization weights $\hat{\gamma}_{j,i}$ in (30.20) for the controller design at each subsystem $S_i$. Therefore, to compute the set of $\omega$, a CMPC problem is stated by considering: (i) a static model of the whole LSNS, and (ii) a cost function that only takes into account the global control objectives associated to the system. Regarding the system static model, the upper optimization layer works with a sampling time $\Delta t_1$, corresponding to the periodicity in the flow requested by sinks. Thus, when looking at the volume evolution of storage elements, they show the parallel behavior as the flow to the sinks, i.e., volumes also show a periodic behavior with period $\Delta t_1$. For this reason, when modeling the network at sampling time $\Delta t_1$, it can be assumed that volumes do not change, i.e., the dynamics of storage elements (30.1) are modified considering $\mathbf{x}(k+1) = \mathbf{x}(k)$. Hence, storage elements behave as nodes and the network dynamic model (30.1) becomes a static model (set of algebraic equations). Having this model and the functional

$$J_{up}(k) = \sum_{j=1}^{m_g} \gamma_j J_j(\mathbf{u}(k:k+N_p-1), \mathbf{x}(k+1:k+N_p)), \tag{30.21}$$

Problem 30.1 is properly formulated in order to obtain the desired weights $\omega$ and, indeed, any weight for any arc of any path within the LSNS. To mathematically and systematically find all flow paths in an LSNS, its structure is used by means of node-arc incidence matrices, which represent both the flow balances and the graph structure [3].

### 30.3.3 Lower Optimization Layer

Having a decentralized predictive controller $C_i \in \mathcal{C}$ for each subsystem $S_i \in \mathcal{N}$ with a cost function as in (30.20), the shared inputs for all subsystems in $\mathcal{N}$ are written as $\boldsymbol{\mu}_{ij}$, whose directionality is defined from $S_i$ to $S_j$, $i \neq j$. Additionally, $\boldsymbol{\mu}_{ij}$ not only contain values of each component at time step $k$ but also all values over $N_p$[2]. The fact of having available this complementary information of the shared variables allows to use predicted values of manipulated flows instead of starting a negotiation procedure between subsystems in order to find their value (following the distributed control philosophy). Besides, the implementation of the hierarchical DMPC approach requires that subsystem models are modified to coordinate with other subsystems. To introduce such modification, the following concept is introduced.

**Definition 30.1 (Virtual sink).** Consider two subsystems $S_1$ and $S_2$, which share a set of manipulated flows $\boldsymbol{\mu}_{12}$. According to the notation employed here, those flows go from $S_1$ to $S_2$. If the solution sequence of optimization subproblems — defined by the pre-established hierarchical order — determines that $\boldsymbol{\mu}_{12}$ is computed by $C_1$, then flows in $\boldsymbol{\mu}_{12}$ are considered as *virtual sinks* in $C_2$ since their values are now imposed in the same way as the flow to sinks. ■

The *pure* hierarchical control scheme determines a sequence of information propagation among the subsystems, where top-down communication is available from upper to lower level of the hierarchy (see [16]). Note that, despite the subsystem

[2] This chapter considers $N_u = N_p$, where $N_u$ denotes the control horizon. In the case that $N_u < N_p$, it is still necessary to know the values for shared variables from $N_u$ until $N_p$, no matter the way they are considered (e.g., keeping constant their value at time instant $N_u$, make them null, etc.).

**Algorithm 30.1** ML-DMPC Approach

```
k=0
loop
    set x(k)
    (ω, γ̃) ⇐ solve Problem 1 with (30.21)
    while k/Δt_1 ∉ ℤ
        u_MPC,i ⇐ solve Problem 1 with (30.20) and using ω, γ̃
    end loop
end loop
```

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. However, it may happen that some shared links have defined their flow direction such as bottom-up communications within the hierarchy, which breaks the mentioned unidirectional flow between DMPC controllers. This fact implies that the standard hierarchical control scheme for partitioned LSNS cannot be straight applied. To solve this situation and to design a DMPC strategy, a hierarchical-like DMPC approach, proposed by [7], has been considered and conveniently implemented over the partitioned system. This strategy follows the hierarchical control philosophy and the sequential way of solving the optimization subproblems of the corresponding MPC controllers but also considering the appearance of bidirectional information flows.

The hierarchy defined by the approach of [7] implies that the controller $C_i$ will be allocated in a different level according to the flow request of its corresponding subsystem $S_i$. Considering the simple topology in Figure 30.2, this fact means that the controller $C_{\tilde{S}}$ will be at the top of the hierarchy, while controllers $C_a$, $C_b$, and $C_c$ will share the bottom level. All controllers work with a sampling time $\Delta t_2$ and the computational time spend by the scheme corresponds with the sum of maximum times of each hierarchical level of controllers (e.g., $\tau_{\text{total}} = \tau_{C_{\tilde{S}}} + \max(\tau_{C_a}, \tau_{C_b}, \tau_{C_c})$ for the scheme in Figure 30.2, where $\tau$ denotes the computational time). Special considerations should be done for the treatment of bidirectional shared flows [7, 9].

### 30.3.4 Interaction of Layers

The sharing of information between layers depends on the nature and features of each application. In general, the computational time that the upper layer spends is quite low with respect to the computational time of the lower layer. This fact is due to the difference in the nature of the models handled by each layer and the interactions given by the DMPC controllers as well as their amount and disposition within the defined hierarchy. Once the optimization problem related to the upper layer is solved, the resultant parameters are properly updated for each optimization problem behind each $C_i \in \mathcal{C}$. This updating is performed with a periodicity $\Delta t_1$. Algorithm 30.1 collects the main steps of the proposed ML-DMPC approach.

# 30.4 Application Results

## 30.4.1 Case-study Description

The approach presented in this work is assessed with a case study of a real large-scale system, specifically the Barcelona drinking water network (DWN). This network supplies potable water to the Metropolitan Area of Barcelona (Catalunya, Spain). In general the water network operates as a pull interconnected system driven by endogenous and exogenous flow demands; different hydraulic elements are used to collect, store, distribute and serve drinking water to the associated population. For further details about this network, the reader is refered to [10].

#### 30.4.1.1 System Management Criteria

The operational goals in the management of the Barcelona DWN have been provided by AGBAR due their knowledge of the system. These policies are of three kinds: *economic*, *safety*, *smoothness*, and are respectively stated as follows:

1. To provide a reliable water supply in the most economic way, minimizing water production and transport costs,
2. To guarantee the availability of enough water in each reservoir to satisfy its underlying demand, keeping a safety stock to face uncertainties and avoid stock-outs.
3. To operate the transport network under smooth control actions.

These objectives are assessed by minimizing the following performance indices [3]:

$$J_E(k) \triangleq |(\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2(k))^{\mathrm{T}} \mathbf{u}(k)|, \tag{30.22a}$$

$$J_S(k) \triangleq \|\boldsymbol{x_i}(k)\|^2, \tag{30.22b}$$

$$J_U(k) \triangleq \|\Delta \mathbf{u}(k)\|^2, \tag{30.22c}$$

where $J_E \in \mathbb{R}_{\geq 0}$ represents the economic cost of network operation taking into account water production cost $\boldsymbol{\alpha}_1 \in \mathbb{R}^{n_u}$ and water pumping cost $\boldsymbol{\alpha}_2 \in \mathbb{R}^{n_u}$ which change every time instant $k$ according to the variable electric tariff; $J_S \in \mathbb{R}_{\geq 0}$ is a performance index which penalizes the amount of volume $\boldsymbol{x_i} \triangleq \min\{\mathbf{0}, \mathbf{x} - \mathbf{x}_s\} \in \mathbb{R}^{n_x}$ that goes down from $\mathbf{x}_s$, a predefined safety volume threshold; $J_U \in \mathbb{R}_{\geq 0}$ represents the penalization of control signal variations $\Delta\mathbf{u}(k) \triangleq \mathbf{u}(k) - \mathbf{u}(k-1)$ to extend actuator life and assure a smooth operation; and $\|\cdot\|$ is the Euclidean norm, i.e., $\|\mathbf{z}\| = \sqrt{\mathbf{z}^{\mathrm{T}}\mathbf{z}}$. More details about the management criteria of this case study can be found in [10].

[3] The performance indices considered in this work may vary or generalized with the corresponding manipulation to include other control objectives.

#### 30.4.1.2 Control-oriented Modelling

Consider a DWN being represented as the interconnection of $n_x$ tanks, $n_u$ actuators (pumps and valves), $n_d$ sectored demands and $n_q$ intersection nodes; according to Section 30.2.1, this system can be generally described in state-space form by (30.10), where $\mathbf{x} \in \mathbb{R}^{n_x}$ is the state vector of water stock volumes in m$^3$, $\mathbf{u} \in \mathbb{R}^{n_u}$ is the vector of manipulated flows in m$^3$/s, and $\mathbf{d} \in \mathbb{R}^{n_d}$ corresponds to the vector of disturbances (sectored water demands) in m$^3$/s. In the particular case of the Barcelona DWN, the outflows from storage elements are manipulated, hence, $\mathbf{E}_x = 0$ in (30.10b).

The states and control inputs are subject to (30.11); this polytopic hard constraints are due to the physical limits in tanks (minimum and maximum volume capacities) and the operational limits in actuators (minimum and maximum flow capacities). For safety and service reliability, in the Barcelona DWN states are also subject to soft constraints

$$\mathbf{x}(k) \geq \mathbf{x}_s(k) - \boldsymbol{x_i}(k) \geq \mathbf{0}, \quad \forall k, \tag{30.23}$$

where $\mathbf{x}_s \in \mathbb{R}^{n_x}$ is a vector of safety volume thresholds in m$^3$, estimated empirically, above which is desired to keep the reservoirs to avoid stock-outs, and $\boldsymbol{x_i} \in \mathbb{R}^{n_x}$ represents the amount of volume in m$^3$ that goes down from the desired safety thresholds.

The Barcelona DWN model contains a total amount of 63 tanks and 114 manipulated actuators. Moreover, the network has 88 demand sectors and 17 pipes intersection nodes. Both the demand episodes and the network calibration/simulation set-up are provided by AGBAR. See the aforementioned references for further details of DWN modeling and specific insights related to this case study.

### 30.4.2 ML-DMPC Setup

This section presents the results of applying the proposed ML-DMPC approach to the partitioned model of the Barcelona DWN developed in [9]. Thus, the overall system is assumed to be composed of six subsystems which are non-overlapped, output-decentralized and input-coupled (see Figure 30.3). The model of each subsystem is obtained for $i \in \{1,\ldots,6\}$ following Section 30.2.2 and expressed by (30.12). In the same way, the hard constraints of the overall DWN are partitioned and expressed by (30.13), while for each $i$-th subsystem the safety constraints are expressed by

$$\mathbf{x}_i(k) \geq \mathbf{x}_{s,i}(k) - \boldsymbol{x_i}_i(k) \geq \mathbf{0}, \quad \forall k. \tag{30.24}$$

The decomposition of the original problem also leads to split the cost function. Therefore, each subsystem will be solving, at each time step, the following local multi-objective optimization problem:

$$J_i^*(k) \triangleq \min_{\mathbf{u}_i^*(k:k+N_p-1)} \rho_i \left(\gamma_1 J_{E,i} + \gamma_2 J_{S,i} + \gamma_3 J_{U,i}\right), \tag{30.25}$$

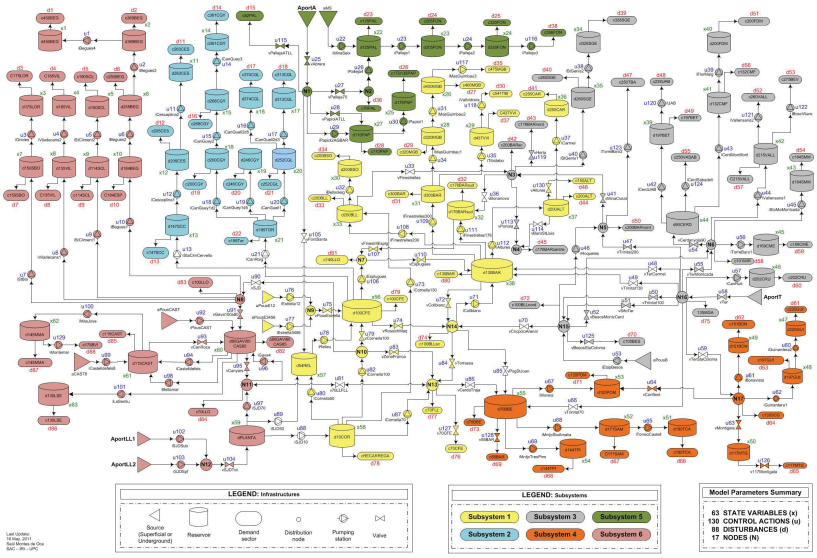



**Figure 30.3:** Partition of the Barcelona DWN

where $J_{E,i} \triangleq \sum_{l=0}^{N_c-1}(\boldsymbol{\alpha}_{1,i} + \boldsymbol{\alpha}_{2,i}(k+l))\mathbf{u}_i(k+l)$ is the economic objective, $J_{S,i} \triangleq \sum_{l=1}^{N_p} \|\boldsymbol{x}_{\boldsymbol{i}i}(k+l)\|^2$ is the safety objective, $J_{U,i} \triangleq \sum_{l=0}^{N_c-1} \|\Delta\mathbf{u}_i(k+l)\|^2$ is the smoothness objective, $N_p$, $N_c \in \mathbb{Z}_{\geq 0}$ are the prediction and control horizon respectively, $\rho_i$ is a positive scalar weight to prioritize subsystems, $\gamma_1$, $\gamma_2$, and $\gamma_3$ are positive scalar weights to prioritize each objective in the aggregate local cost function, $l$ is the time step within the receding horizon, and $\mathbf{u}_i$, $\boldsymbol{x}_{\boldsymbol{i}i}$ and $\Delta\mathbf{u}_i$ are the $i$-th subsystem local variables previously defined. It can be noticed in Figure 30.4, in a more compact way, the resulting subsystems and the important couplings between them including their direction. Instead of neglecting the effect of this shared links as classic pure decentralized control schemes do, the multi-layer hierarchical coordination described in Section 30.3 is applied here.

The results obtained by applying the ML-DMPC are contrasted with those of applying a CMPC approach and a non multi-layer DMPC strategy proposed in [9]. For this case study, the optimization scheme follows Section 30.3, resulting in a *bi-layer* problem which is set up as follows:

- First, the *upper layer* works with a daily time scale and it is in charge of achieving the optimal water source selection. This layer, named *Daily Centralized Control* is a centralized optimization problem with time step $\Delta t = 24$h, which minimizes the cost function (30.22a) subject to a daily model of the DWN represented by $\mathbf{x}(k+1) = \mathbf{x}(k)$, due to the periodic behavior of states at this time layer, and to constraints (30.11) and (30.23). The objective of this upper layer is to determine and fix in an appropriate way the unitary costs of the critical shared variables that act as sources in the partitioned model, in order to enforce the global economic

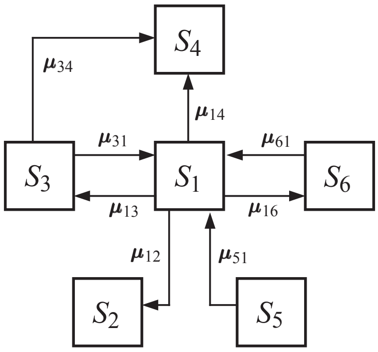



**Figure 30.4:** Network subsystems $S_i$ and their sets of shared connections $\boldsymbol{\mu}_{ij}$

objective by sequentially coordinating subsystems, allowing them to solve their own problems and achieving the solution of the original system.

- Second, the *lower layer* works with an hourly time scale to cope with the DMPC of the original problem. This layer, named *Hourly Decentralized MPC Control* follows the hierarchical coordination scheme proposed in Section 30.3 to perform the minimization of the local cost functions (30.25) subject to (30.12), (30.13), and (30.24), in order to obtain the control policies to operate the DWN and achieve the desired performance. In this hourly layer, following the criterion of the DWN management company, each local MPC controller works with common prediction and control horizons $N_p = N_c = 24$h. The weights of the cost function (30.25) are $\rho_{1:6} = 1$, $\gamma_1 = 100$, $\gamma_2 = 10$ and $\gamma_3 = 0.005$. See [9] for details on the hierarchical DMPC solution sequence.

The results are obtained for 72 hours (July 24-26, 2007). Simulations have been carried out using Matlab® 7.1 (R14SP3). The computer used to run the simulations is a PC Intel® Core™ 2 running at 2.4GHz with 4GB of RAM. The tuning of design parameters has been done in a way that the highest priority objective is the economic cost, which should be minimized while maintaining adequate layers of safety volume and control action smoothness. In order to implement the ML-DMPC approach, the demand forecasting algorithms presented in [10, 13] are used to calculate the disturbance vector involved in each control problem. For more details about the twofold-layer optimization problem applied to the Barcelona DWN, the reader is referred to [12].

### 30.4.3 Main Results

The results of the CMPC, DMPC and ML-DMPC strategies applied to the Barcelona DWN are summarized in Table 30.1 in terms of computational burden and of economic cost as a global management performance indicator. For each MPC approach, the computation time (in seconds) and the water, electric and total cost in economic

**Table 30.1:** Performance comparisons

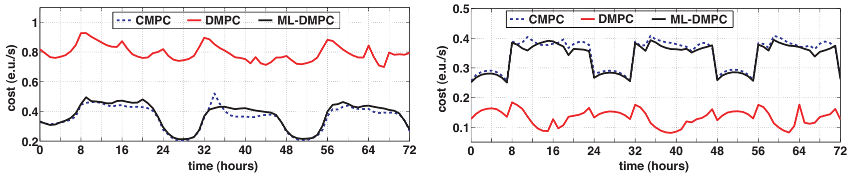
| INDEX | CMPC | DMPC | ML-DMPC |
|---|---|---|---|
| Water Cost | 93.01 | 205.55 | 97.11 |
| Electric Cost | 90.31 | 34.58 | 87.53 |
| Total Cost | 183.33 | 240.13 | 184.65 |
| CPU time | 1143 | 537 | 540 |

units (e.u.), is detailed. It can be noticed that an increment of nearly 30% of the total costs of operation occurs when using the non multi-layer hierarchical DMPC strategy with respect to the CMPC baseline. Despite the lower electric costs, the loss of performance in the overall cost is due to the specialized behavior of local MPC controllers to solve their own optimization problems without knowing the real water supply cost of using shared resources with the neighbors. In contrast, the ML-DMPC outperforms the DMPC results by including the bi-layer optimization which allows to propagate the water cost of sources related with neighbor subsystems to the shared links thanks to the daily centralized control layer. With this ML-DMPC approach the level of sub-optimality is very low comparing with the CMPC strategy, i.e., total costs are very similar, but the computational burden is reduced. For this particular application, the computation time of the three approaches is able to satisfy the real-time constraint since the control sampling time is 1h. Thus, the main motivation for using ML-DMPC is the scalability and easy adaptability of the sub-models if network changes, as well as the modularity of the control policy that leads to face some malfunction/fault without stopping the overall supervisory MPC strategy.

Due the difference of price between water sources and the impact of electric costs on the overall economic performance, the CMPC and ML-DMPC strategies decide to use more water from the Llobregat source despite the consequent pumping of more water through the network (see Figures 30.6), but achieving a lower total cost, while the hierarchical DMPC decides to exploit in each subsystem their own water source (which could be expensive) and minimize the pumping operation cost. Figure 30.5 shows in detail the evolution of water cost and electric cost, respectively. These results confirm the improvement obtained by including an upper layer optimization to coordinate the local MPCs and face the lack of communication when solving their problems in a tractable way.

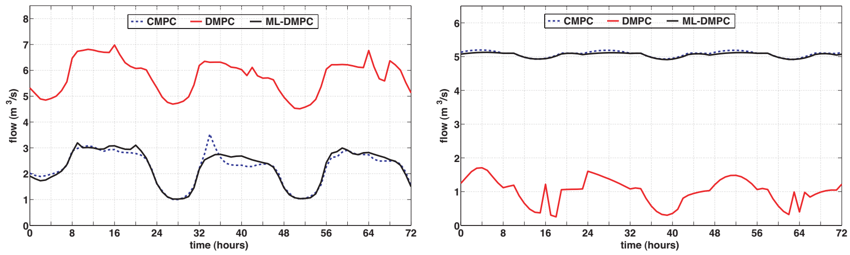
## 30.5 Conclusions and Future Research

This paper has proposed a multi-layer DMPC approach for large-scale networked systems. The upper layer, working with a larger time scale, is in charge of achieving the global control (in general related to an optimal economic cost). On the other

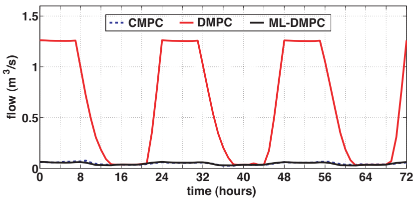



**(a)** Water cost **(b)** Electric cost

**Figure 30.5:** Economic costs of the three MPC strategies




**(a)** Ter source **(b)** Llobregat source




**(c)** Abrera source

**Figure 30.6:** Total flow per water source in the Barcelona DWN

hand, the lower layer is in charge of determining the set-point of the actuators to satisfy the local management/control objectives. The system decomposition is based on graph partitioning theory. Results obtained on selected simulation scenarios has shown the effectiveness of the control strategy in terms of system modularity, reduced computational burden and, at the same time, the reduced loss of performance in contrast to a CMPC strategy and a hierarchical-like DMPC strategy previously presented by the authors. Future work is focused on the formalization of the proposed approach in terms of feasibility, robustness and stability.

**Acknowledgements** This work has been partially funded by the Spanish Ministry of Science and Technology through the WATMAN project (DPI2009-13744).

# References


1. M. Brdys and B. Ulanicki. *Operational Control of Water Systems: Structures, algorithms and applications*. Prentice Hall International, UK, 1994.
2. E.F. Camacho and C. Bordons. *Model Predictive Control*. Springer-Verlag, London, second edition, 2004.
3. W.C. Cheng, N.S. Hsu, W.M. Cheng, and W.W.G. Yeh. A flow path model for regional water distribution optimization. *Water Resources Research*, 45:1 – 12, 2009.
4. T. Keviczky, F. Borrelli, and G.J. Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42(12):2105–2115, December 2006.
5. J. Lunze. *Feedback Control of Large-Scale Systems*. Prentice Hall, Great Britain, 1992.
6. J.M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, Great Britain, 2002.
7. C. Ocampo-Martinez, D. Barcelli, V. Puig, and A. Bemporad. Hierarchical and decentralised model predictive control of drinking water networks: Application to the Barcelona case study. *IET Control Theory & Applications*, 6(1):62–71, 2012.
8. C. Ocampo-Martinez, S. Bovo, and V. Puig. Partitioning approach oriented to the decentralised predictive control of large-scale systems. *Journal of Process Control*, 21(5):775 – 786, 2011.
9. C. Ocampo-Martinez, V. Puig, and S. Bovo. Decentralised MPC based on a graph-partitioning approach applied to Barcelona drinking water network. In *Proceedings of the IFAC World Congress*, Milano (Italy), 2011.
10. 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*, 9(5):565–575, 2009.
11. P.J. Van Overloop. *Model Predictive Control on Open Water Systems*. Delft University Press, Delft, The Netherlands, 2006.
12. V. Puig, C. Ocampo-Martinez, and S. Montes de Oca. Hierarchical temporal multi-layer decentralised mpc strategy for drinking water networks: Application to the Barcelona case study. In *Proceedings of the IEEE Mediterranean Control Conference*, Barcelona (Spain), 2012.
13. 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*, 11(6):640–651, June 2010.
14. J. B. Rawlings and B. T. Stewart. Coordinating multiple optimization-based controllers: New opportunities and challanges. *Journal of Process Control*, 18(9):839–845, October 2008.
15. R. Scattolini. Architectures for distributed and hierarchical Model Predictive Control: A review. *Journal of Process Control*, 19(5):723–731, May 2009.
16. D.D. Šiljak. *Decentralized control of complex systems*. Academic Press, 1991.
17. 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*, 16(6):1192–1206, November 2008.