Universitat Politècnica de Catalunya

Thesis Proposal

Doctoral Program

Automatic Control, Robotics and Computer Vision

MULTI-LAYER DECENTRALIZED PREDICTIVE CONTROL FOR LARGE-SCALE NETWORKED SYSTEMS

Juan Manuel Grosso Pérez

Advisors:
Dr. Carlos Ocampo Martínez
Dr. Vicenç Puig

January, 2013

Multi-Layer and Decentralized Predictive Control for Large-Scale Networked Systems

Abstract

In this thesis, a multi-layer decentralized model predictive control (ML-DMPC) strategy is proposed to be researched for its application to large-scale networked systems (LSNS). The approach aims to exploit the periodic nature of the system disturbance and the availability of both static and dynamic models of the LSNS. The topology of the controller will be structured in two layers. First, an upper layer will be in charge of achieving the global objectives from a set $\mathcal{O}$ of control objectives given for the LSNS. This layer will work with a sampling time Δt_1, corresponding to the disturbance period. Second, a lower layer, with a sampling time Δt_2, $\Delta t_1 > \Delta t_2$, will be 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 will allow 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 will result in a centralized optimization problem for considering the global control objectives, followed of a decentralized scheme for reaching the local control objectives. Each decentralized local MPC controller will be designed with robustness to measured stochastic disturbances and actuator health degradation, and with self-tuning capabilities. This control design aims to improve both the performance and the overall reliability of the LSNS. In addition to formalize the synthesis of the ML-DMPC controller, this thesis aims to analyse conditions for feasibility and stability guarantees. The proposed ML-DMPC approach will be applied to a real case study: the water transport network of Barcelona (Spain), in order to assess the results in terms of system modularity, computational burden and sub-optimality of the system performance in comparison with a centralized MPC (CMPC) strategy.

Contents

1 Introduction

1.1 Motivation

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 due to its social, economic and environmental impact. The leading control technique for the management of LSNS is model predictive control (MPC) [1, 16], which is a suitable approach to deal with challenging multi-criteria problems of real dynamical systems due its systematic and practical formulation, see [9, 2]. Nevertheless, the ever growing complexity of mathematical models (dimensionality, information structure constraints, uncertainty), turns these problems costly or even intractable to solve in practice, specially when controlling LSNS seeking the best operational policies, where centralized MPC algorithms are not prepared enough to face the important computational burden when design aspects (i.e. prediction horizons, weights, and system topology size) have to be continually redefined. Therefore, newer and more efficient methods will be investigated in this thesis, exploiting the structural properties of the plant in order to obtain suitable design and control strategies, to develop decision-support tools for the management of LSNS.

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 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 controlling a part of the entire system.

The work to be performed in this thesis is focused on decentralized model predictive control (DMPC), where a LSNS is decomposed into smaller subsystems that are controlled using local MPC schemes. This partitioning decreases the computational effort but also degrades the performance since the coordination of all the control actions is lost. Thus, a communication and supervisory strategy will be proposed in this thesis to arrange the tasks of the subsystems to provide a feasible solution to the overall system, driven towards the performance of a centralized controller. In addition, to achieve a flexible an reliable controller, it will be also explored the current MPC tuning strategies for weight adaptation, in order to prioritize the multi-criteria optimization problems behind the predictive controllers according to global and local management goals. It is proposed also to identify and isolate the parts that can be solved off-line or can even be explicitly computed. The rest of the control law will be computed on-line, given its dependence of the measurements and/or changing information. All together will allow to define multi-layer and decentralized MPC laws with less computational burden, fact that in turn allows to modify design parameters without affecting the performance of the closed-loop control law. The desired control specifications will be expressed through different performance indexes associated to common objectives such as reductions in control energy and economic costs, assurance of network functionality, among many others.

1.2 State of the Art

LSNS are formed by the interconnection of several subsystems, whose different spatial, temporal and functional characteristics could make them significantly heterogeneous. The optimal management of such systems with a centralized control structure may be a cumbersome task due to the required inherent computational complexity, due to robustness and reliability problems and due to communication limitations. Therefore, many control structures have been researched and reported in literature over the last forty years, i.e., completely decentralized structures, distributed control systems with exchange of information among local controllers and coordinated or hierarchical multi-layer structures. For a survey on the state of the art on the aforesaid MPC architectures the reader is referred to [4, 5, 10, 19, 21] and references therein.

In a pure *decentralized control architecture* the input and output variables of a given system are decomposed and grouped into disjoint sets for which local regulators are designed to operate in a completely independent fashion. While centralized MPC leads to a plant-wide optimum, it is computationally intensive, it is relatively difficult to implement, tune, and maintain, and it is characterized by poor fault tolerance. Decentralized MPC is flexible, reliable, and easy to implement and maintain, but it leads to solutions that are not plant-wide optimum. On the other hand, a *distributed control architecture* improves the global performance by allowing communication between local controllers to achieve a consensus that leads to an approximation of the centralized solution. Nevertheless, the distributed scheme is affected by the quality and the reliability of the communication process, and by the negotiation algorithm that the local controllers follow. LSNS require control laws whose computation is efficient, and whose implementation entails a minimal amount of information exchange between the subsystems, therefore, current research is oriented toward designing robust and computationally feasible *coordinated/hierarchical schemes* that work toward combining the advantages of both the centralized and decentralized control strategies while addressing their drawbacks. The decentralized structure of the system is maintained, but the performance is driven toward that of a centralized scheme. The coordinator or higher-layers of the control scheme coordinates the actions of the individual decentralized MPCs, relaying information among the various individual controllers to account for the interaction effects that exist between the different subsystems of the complex LSNS.

1.3 Thesis Objectives

Despite the wide literature in the context of decentralized control, hierarchical multilayer systems and the related MPC approaches, systematic design methods guaranteeing well-assessed properties are still lacking and only ad-hoc solutions tailored on some specific industrial problems have emerged. Therefore, this thesis primary aims to formalize an answer to the early question stated by [10]:

Given a time interval over which a two-level system's performance is to be observed and during which there are n time instants at which the coordinator can influence the decentralized local controllers, what should the coordinator's strategy be so that, after each coordination instant, the coordinator's action results in an improved overall performance, or, in the absence of adversities, the overall performance improves monotonically over the given time interval.

To do so, the following objectives are proposed:

- Objective 1: Develop a formal mathematical framework for the design of constrained ML-DMPC controllers for LSNS.
- Objective 2: Design a non-iterative and tractable algorithm to coordinate the decentralized controllers of the ML-DMPC architecture.
- Objective 3: Analyse conditions to guarantee feasibility and stability of the proposed ML-DMPC approach.
- Objective 4: Incorporate exogenous and endogenous uncertainties in order to design robust ML-DMPC controllers.
- Objective 5: Design a tuning methodology for the decentralized MPC controllers.
- Objective 6: Validate the proposed control strategy using a simulator of the drinking water network of Barcelona as case study.

2 Working Plan

In this section, a research schedule is proposed for this doctoral research. The estimated duration of the pending research is 24 months. During this period, the whole work will be split into several small tasks to fulfil the stated objectives. The tasks are listed below and the working plan is shown in Table 1.

- Task 1: Conceptualization, i.e., identify the structural concepts which will be the subject of the mathematical studies, set the scope of the framework to be developed and provide a conceptual foundation for the problem of coordination in multi-layer and decentralized control architectures for LSNS.
- Task 2: Formalization, i.e., analyse how the results presented in [15] can be generalized and in which direction, taking into account the impact that the specificity of the framework has on a particular conclusion. Through a mathematical investigation of certain aspects of LSNS where hierarchies may arise in the decision-making process, this task aims to abstract, to define, and to formalize (within a control systems theory framework and in reference to structural properties of systems) the decentralized predictive control strategy proposed in this thesis.
- Task 3: Coordination analysis. This task aims to study deeper the current coordination strategy presented in [15] in order to enhance or generalize it, taking into account selected partitioning schemes and the coupled constraints and shared limited resources that arise with decentralization.
- Task 4: Feasibility analysis.
- Task 5: Derivation of conditions for stability guarantees.
- Task 6: Incorporation of mechanisms to guarantee robustness to exogenous uncertainty.
- Task 7: Incorporation of mechanisms to guarantee robustness to endogenous uncertainty.

- Task 8: Adaptation, i.e., develop tuning strategies for the control scheme, specially for the local controllers, capable of varying weights and/or thresholds in accordance to the variation of disturbances and/or actuator degradation.
- Task 9: Validation of the proposed control strategy with a real case study: the drinking water network of Barcelona.
- Task 10: Summarizing of intermediate results and writing of papers to be submitted to conferences and/or journals of high-impact.
- Task 11: Writing of the PhD thesis document.
- Task 12: Research stage and cooperation with other national or international research groups.

Table 1: Timetable proposal for the pending research period

	2013				2014			
	Jan-Mar	Apr-Jun	Jul-Sep	Oct-Dec	Jan-Mar	Apr-Jun	Jul-Sep	Oct-Dec
Task 1								
Task 2								
Task 3								
Task 4								
Task 5								
Task 6								
Task 7								
Task 8								
Task 9								
Task10								
Task11								
Task12								

3 Background

3.1 Fundamentals of Model Predictive Control

3.1.1 General Considerations

Model Predictive Control (MPC) stands for a family of methods that select control actions based on optimisation problems. It is one of the most successful control technologies applied in a wide variety of application areas due to its capability to explicitly incorporate constraints and define multiple performance objectives within a single control problem.

The tractability of an MPC problem, specially when dealing with LSNS, is defined by the nature of the elements that are involved in the predictive and optimisation strategy. The use of a *cost function* allows to describe the desired behaviour of the system and is generally defined under two purposes: stability and performance. It serves also to specify preferences in a multi-objective optimal control problem. This element is application-dependant but there exist within the MPC literature common cost functions which are convex and results in a easy to solve problem. Common choices are based on linear (i.e., $\|\cdot\|_1$, and $\|\cdot\|_\infty$) and quadratic norm costs (i.e., $\|\cdot\|_2$), which are usually weighted. The explicit handling of *constraints* is the key strength of MPC. It can be found in different applications the following types of constraints: linear (used to upper/lower bound variables), convex quadratic (used to bound a variable to lie within an elipsoid), probabilistic (used to deal with uncertainty and to reduce conservatism of worst-case approaches), second order cones, switched constraints (used when the inclusion of the constraint depends on meeting a predefined condition), non-linear constraints (compromises any other type of constraint and are very difficult to handle when solving the optimisation problem). The most critical element in the MPC framework is the *dynamic model* of the system, since the robustness and performance of the controller depends on the model which can be deterministic or stochastic, linear or non-linear, continuous or discrete or hybrid. Further details on MPC theory, design and applications can be found in [2], [9], among others.

3.1.2 General MPC Problem Setting

A generic MPC framework is given by the following optimal control problem:
Problem 1 (**Generic MPC problem**).

$$J^* \triangleq \min_{\mathbf{u}(k:k+N_p-1)} V_f(\mathbf{x}(k+N_p)) + \sum_{i=0}^{N_p-1} l_i(\mathbf{x}(k+i),\mathbf{u}(k+i)), \quad \text{Cost function} \qquad (1)$$

subject to

$$\begin{aligned}
\mathbf{x}(k+i+1) &= f(\mathbf{x}(k+i),\mathbf{u}(k+i)), && \text{Dynamics} \qquad (2)\\
(\mathbf{x}(k+i),\mathbf{u}(k+i)) &\in \mathbb{X}\times\mathbb{U}, && \text{Constraints} \qquad (3)\\
\mathbf{x}(k+N_p) &\in \mathbb{X}_f, && \text{Terminal constraint} \qquad (4)
\end{aligned}$$

where $k \in \mathbb{Z}_{\geq 0}$ is the current time instant, $N_p \in \mathbb{Z}_+$ is the prediction horizon, $i \in \mathbb{Z}_0^{N_p-1}$ is the prediction time step ahead of the current time instant, $\mathbf{x} \in \mathbb{R}^{n_x}$ is the system state, $\mathbf{u} \in \mathbb{R}^{n_u}$ is the control input, and $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is the arbitrary system state evolution function. Moreover, $l_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}_{\geq 0}$ is the stage cost, $V_f : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ is the terminal cost, and $\mathbb{X} \subset \mathbb{R}^{n_x}$ and $\mathbb{U} \subset \mathbb{R}^{n_u}$ are compact sets containing the origin and representing the set of *feasible* states and the set of *admissible* control actions, respectively. When $N_p = \infty$, the MPC law solves a *infinite horizon* optimal control problem, while with $N_p < \infty$, the MPC law solves a *finite horizon* optimal control problem. □

At each time instant, MPC uses a system model and all currently available information (present and past) to predict the system future evolution over a given prediction horizon and solve an open-loop optimisation problem to calculate a sequence of future control actions that must satisfy system constraints to achieve the desired performance; only the first control move is applied. At the next time step, the overall procedure is repeated over a shifted prediction horizon using updated system measurements to compensate for modelling errors and/or disturbances. This scheme is referred as receding horizon strategy and it is summarised in Algorithm 1.

Algorithm 1 Receding Horizon Strategy

1: measure the state $\mathbf{x}(k)$ at time k
2: compute $\underrightarrow{\mathbf{u}}^*(\mathbf{x}(k)) \triangleq [\mathbf{u}^*(k), \ldots, \mathbf{u}^*(k+H_p-1)]$ by solving Problem 1 with horizon N_p
3: apply the first element $\mathbf{u}_{\mathrm{MPC}}(k) \triangleq \mathbf{u}^*(k)$ to the system
4: proceed to time step $k+1$
5: go to 1.

Remark 1. Although perfect state measurement is generally assumed available, the generic strategy and the approaches developed in this thesis can be used in practice with an estimate of the state obtained by an observer or any state estimator. $\diamondsuit$

Remark 2. The basic MPC strategy uses open-loop predictions and computes input control sequences, but it is also possible to use closed-loop predictions and optimise over control laws, i.e., affine functions of previous states or disturbances if measured. $\diamondsuit$

3.2 Modeling Framework for Multi-Layer Decentralized Control

This thesis adopts the modeling framework presented in [15]. The fundamentals of such control-oriented model used to design decentralized controllers for LSNS are extracted and stated below. The control system architecture of a LSNS may be defined in two levels as shown in Figure 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 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.

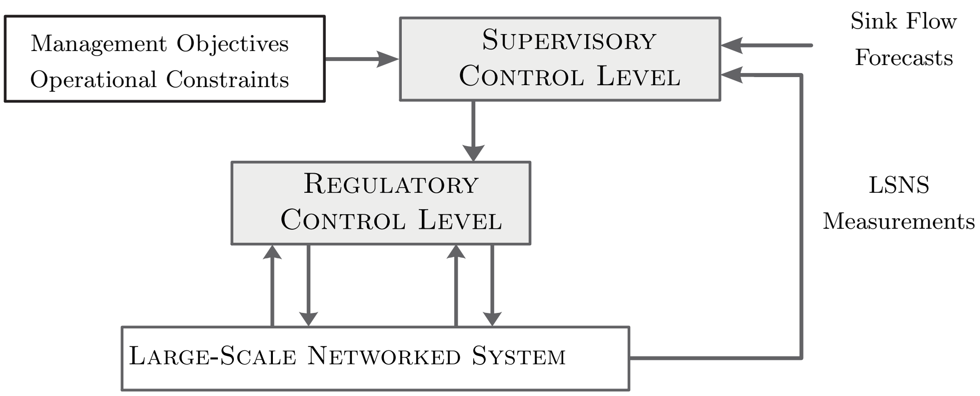


Figure 1: Control architecture for a LSNS

In a general way, a proper control-oriented model of a given system is defined such that it captures its main behavior, being as simple as possible in order to save computational burden when such model is used for control design purposes. This thesis 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 adopted in this thesis 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

$$x(k+1) = x(k) + \Delta t\,(q_{\text{in}}(k) - q_{\text{out}}(k)), \tag{5}$$

where x denotes the stored flow volume, q_{in} and q_{out} denote the net inflow and outflow, respectively; Δt is the considered sampling time and index $k \in \mathbb{Z}_{\geq 0}$ represents the discrete time instant. Notice that (5) 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

$$x_{\min} \leq x(k) \leq x_{\max}, \qquad \forall\, k, \tag{6}$$

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.,

$$q_{\text{in}}(k) = \sum_i q_{\text{out},i}(k). \tag{7}$$

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

$$\sum_j q_{\text{in},j}(k) = q_{\text{out}}(k). \tag{8}$$

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

$$q_{\min,\Lambda_i} \leq q_{\Lambda_i}(k) \leq q_{\max,\Lambda_i} \tag{9}$$

might be considered, where q_{Λ_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 (9) 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 (5), with the associated constraint such as in (6).

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

$$q_{\min} \leq q(k) \leq q_{\max}, \qquad \forall\, k, \tag{10}$$

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 (10).

Remark 3. Regarding storage elements, when their outflow is not manipulated, its expression corresponds with

$$q_{\text{out}}(k) = h(x(k)), \tag{11}$$

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 (11) and rewriting the right-hand side of (10) as

$$q(k) \leq \min\{q_{\max}, h(x(k))\}, \qquad \forall\, k. \tag{12}$$

Moreover, in the scenario where $x_{\min} \neq 0$ and the outflow of the storage element is manipulated, the left-hand side of (10) should be rewritten as

$$\min\{q_{\min}, h(x(k))\} \leq q(k), \qquad \forall\, k, \tag{13}$$

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

3.2.1 Control-oriented Model

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_α sources are considered as inflows. Stating the volume in storage elements as the state variable $\mathbf{x} \in \mathbb{R}^{n_x}$, the flow through the handling elements as the manipulated inputs $\mathbf{u} \in \mathbb{R}^{n_u}$, and the demanded flow as *additive* measured disturbances $\mathbf{d} \in \mathbb{R}^{n_d}$, an LSNS may be abstracted and described by the following set of linear (or linearised) discrete difference-algebraic equations (DAE) for all time instant $k \in \mathbb{Z}_{\geq 0}$:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{B}_d\mathbf{d}(k), \tag{14a}$$
$$\mathbf{0} = \mathbf{E}_x\mathbf{x}(k) + \mathbf{E}_u\mathbf{u}(k) + \mathbf{E}_d\mathbf{d}(k), \tag{14b}$$

where the difference equations in (14a) describe the dynamics of storage elements, and the algebraic equations in (14b) describe the static relations (i.e., mass balance at intersection nodes) in the network. Moreover, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{B}_d$, $\mathbf{E}_x$, $\mathbf{E}_u$, $\mathbf{E}_d$, are time-invariant matrices of suitable dimensions 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

$$\mathbf{x}_{\text{min}} \leq \mathbf{x}(k) \leq \mathbf{x}_{\text{max}}, \quad \forall k, \tag{15a}$$
$$\mathbf{u}_{\text{min}} \leq \mathbf{u}(k) \leq \mathbf{u}_{\text{max}}, \quad \forall k, \tag{15b}$$

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

Remark 4. 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. ♢

3.2.2 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. This thesis considers a graph-theory-based algorithm proposed in [12] for the automatic partitioning of LSNS into subsystems. The algorithm transforms the dynamical model of the given system into a graph representation. Once the equivalent graph has been obtained, the problem of graph partitioning is then solved. The resultant partitions are composed of a set of non-overlapping subgraphs such that their sizes, in terms of number of vertices, are similar and the number of edges connecting them is minimal. To achieve this goal the algorithm applies a set of procedures based on identifying the highly connected subgraphs with balanced number of internal and external connections. Some additional pre-filtering and post-filtering routines are also needed to be included to reduce the number of obtained subsystems.

Thus, the overall system (14) 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 $i \in \{1, \ldots, M\}$ as

$$\mathbf{x}_i(k+1) = \mathbf{A}_i\mathbf{x}_i(k) + \mathbf{B}_i\mathbf{u}_i(k) + \mathbf{B}_{d,i}\mathbf{d}_i(k) + \mathbf{B}_{\mathrm{sh},i}\boldsymbol{\mu}_i(k), \tag{16a}$$

$$\mathbf{0} = \mathbf{E}_i\mathbf{u}_i(k) + \mathbf{E}_{d,i}\mathbf{E}_i(k) + \mathbf{E}_{\mathrm{sh},i}\boldsymbol{\mu}_i(k), \tag{16b}$$

where $\mathbf{x}_i \in \mathbb{R}^{n_{x_i}}$, $\mathbf{u}_i \in \mathbb{R}^{n_{u_i}}$ and $\mathbf{d}_i \in \mathbb{R}^{n_{d_i}}$ are the local states, inputs and disturbances 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 $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{B}_{d,i}$, $\mathbf{E}_{u,i}$, and $\mathbf{E}_{d,i}$ are dictated by each subsystem topology. In the same way, the previously defined overall constraints (15) are partitioned for each i-th subsystem as

$$\mathbf{x}_{\min,i} \leq \mathbf{x}_i(k) \leq \mathbf{x}_{\max,i}, \quad \forall k, \tag{17a}$$

$$\mathbf{u}_{\min,i} \leq \mathbf{u}_i(k) \leq \mathbf{u}_{\max,i}, \quad \forall k. \tag{17b}$$

Moreover, it may occur that the n_α 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, $j \in \{1, \ldots, n_\alpha\}$ 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 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 [12]. 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 ω, 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 4 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.

3.2.3 MPC Problem Formulation

From the LSNS model in (14), 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 2. An MPC controller design is based on the solution of the open-loop multi-objective optimization problem (OOP)

$$\min_{\mathbf{u}^*(k:k+N_p-1)} J(k) \triangleq \sum_{m=1}^{|\mathcal{O}|} \gamma_m J_m(\mathbf{u}(k : k + N_p - 1), \mathbf{x}(k + 1 : k + N_p)), \tag{18a}$$

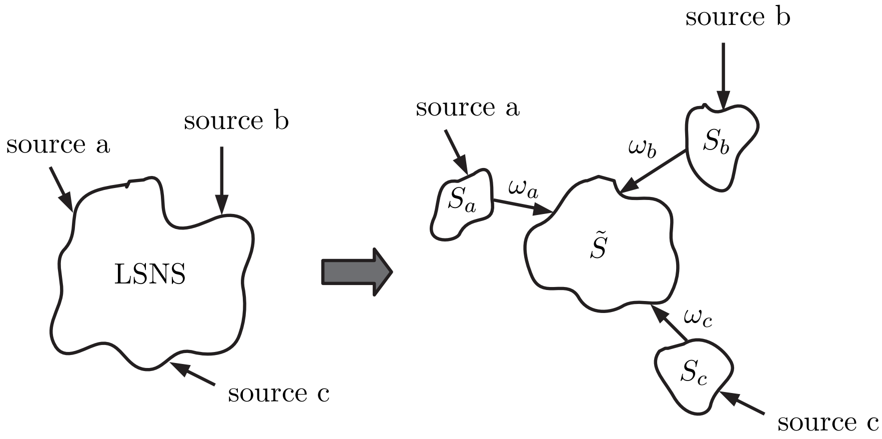


Figure 2: Scheme of LSNS partitioning ($n_\alpha = 3$)

subject to system model (14), system constraints (15) over N_p, and a set of n_c operative constraints given by management policies of the system and condensed on the form

$$\mathbf{G}_1\mathbf{x}(k+1:k+N_p) + \mathbf{G}_2\mathbf{u}(k:k+N_p-1) + \mathbf{G}_3\mathbf{d}(k:k+N_p-1) \leq \mathbf{g}, \tag{18b}$$

where $J(\cdot) : \mathbb{R}^{n_u(N_p-1)\times n_x N_p} \mapsto \mathbb{R}$ in (18a) is the cost function collecting all control objectives of the set $\mathcal{O}$ and γ_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, $\mathbf{G}_1 \in \mathbb{R}^{n_c N_p \times n_x N_p}$, $\mathbf{G}_2 \in \mathbb{R}^{n_c N_p \times n_u(N_p-1)}$, $\mathbf{G}_3 \in \mathbb{R}^{n_c N_p \times n_d(N_p-1)}$, and $\mathbf{g} \in \mathbb{R}^{n_c N_p}$. $\square$

Assuming that Problem 2 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

$$\mathbf{u}_{\text{MPC}}(\mathbf{x}(k)) \triangleq \mathbf{u}^*(k), \tag{19}$$

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 (19) is known in the MPC literature as *the MPC law* [9].

Besides, the decomposition of the original problem leads to design an MPC controller $C_i \in \mathcal{C}$, with $i = \{1, \ldots, M\}$, for each of the M subsystems. This fact also leads to split the cost function (18a). Therefore, each subsystem considers the local cost function

$$J_i(k) = \sum_{m=1}^{|\mathcal{O}|} \gamma_{m,i} J_{m,i}(\mathbf{u}_i(k:k+N_p-1), \mathbf{x}_i(k+1:k+N_p)), \tag{20}$$

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

$$\mathbf{G}_{1,i}\mathbf{x}_i(k+1:k+N_p) + \mathbf{G}_{2,i}\mathbf{u}_i(k:k+N_p-1) + \mathbf{G}_{3,i}\mathbf{d}_i(k:k+N_p-1) \leq \mathbf{g}_i. \tag{21}$$

4 Multi-Layer Decentralized Model Predictive Control (ML-DMPC)

The content of this section have been recently presented in [15] and is presented here as a basis for the theoretical formalization that this thesis aims to develop in the context of ML-DMPC of LSNS.

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

Assumption 2. *All sinks show a periodic flow request, whose period is* $T = \Delta t_1$*.*

Assumption 3. *The set* $\mathcal{O}$ *of control objectives is defined as*

$$\mathcal{O} = \mathcal{O}_l \cup \mathcal{O}_g, \tag{22}$$

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 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.

First of all, in order to develop this idea, the cost function related to the centralized MPC (CMPC) in Problem 2 can be rewritten as

$$\begin{aligned} J(k) = \sum_{j=1}^{m_g} \gamma_j J_j(\mathbf{u}(k:k+N_p-1), \mathbf{x}(k+1:k+N_p)) \\ + \sum_{m=1}^{m_l} \gamma_m J_m(\mathbf{u}(k:k+N_p-1), \mathbf{x}(k+1:k+N_p)). \end{aligned} \tag{23}$$

The approach proposed in this thesis consists in designing a decentralized MPC scheme, where each controller $C_i \subset \mathcal{C}$ considers a newer version of (20) taking into account the structure of

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

(23). Hence, the cost function related to each C_i is written as

$$J_i(k) = \sum_{j=1}^{m_g} \hat{\gamma}_{j,i} \hat{J}_{j,i}(\mathbf{u}_i(k : k + N_p - 1), \mathbf{x}_i(k + 1 : k + N_p)) \\ + \sum_{m=1}^{m_l} \gamma_{m,i} J_{m,i}(\mathbf{u}_i(k : k + N_p - 1), \mathbf{x}_i(k + 1 : k + N_p)), \quad (24)$$

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 Δt_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 Δt_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 $\mathcal{O}$.

4.2 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 ω related to pseudo-sources and discussed in Section 3.2.2 (see Figure 2). These weights ω will determine the prioritization weights $\hat{\gamma}_{j,i}$ in (24) for the controller design at each subsystem S_i. Therefore, to compute the set of ω, 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 Δt_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 Δt_1. For this reason, when modeling the network at sampling time Δt_1, it can be assumed that volumes do not change, i.e., the dynamics of storage elements (5) are modified considering $\mathbf{x}(k+1) = \mathbf{x}(k)$. Hence, storage elements behave as nodes and the network dynamic model (5) becomes a static model (set of algebraic equations). Having this model and the functional

$$J_{up}(k) = \sum_{j=1}^{m_g} \gamma_j J_j(\mathbf{u}(k : k + N_p - 1), \mathbf{x}(k + 1 : k + N_p)), \quad (25)$$

Problem 2 is properly formulated in order to obtain the desired weights ω 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].

4.3 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 (24), 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 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 layer of the hierarchy (see [20]). Note that, despite the subsystem 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 [11], 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 [11] implies that the controller C_i will be allocated in a different layer according to the flow request of its corresponding subsystem S_i. Considering the simple topology in Figure 2, this fact means that the controller $C_{\tilde{\mathcal{S}}}$ will be at the top of the hierarchy, while controllers C_a, C_b, and C_c will share the bottom layer. All controllers work with a sampling time Δt_2 and the computational time spend by the scheme corresponds with the sum of maximum times of each hierarchical layer of controllers (e.g., $\tau_{\text{total}} = \tau_{C_{\tilde{\mathcal{S}}}} + \max(\tau_{C_a}, \tau_{C_b}, \tau_{C_c})$ for the scheme in Figure 2, where τ denotes the computational time). Special considerations should be done for the treatment of bidirectional shared flows [13, 11].

4.4 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

[2]This thesis 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.).

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 Δt_1. Algorithm 2 collects the main steps of the proposed ML-DMPC approach.

Algorithm 2 ML-DMPC Approach

k=0
loop
    set $x(k)$
    $(\omega, \tilde{\gamma}) \Leftarrow$ **solve** Problem 1 with (25)
    while $\frac{k}{\Delta t_1} \notin \mathbb{Z}$
        $u_{\mathrm{MPC},i} \Leftarrow$ **solve** Problem 1 with (24) and using ω, $\tilde{\gamma}$
    end loop
end loop

5 Preliminary Results

5.1 Case-study Description

Preliminary simulation-based results of the control approach proposed in this thesis have been recently reported in [15]. The results are obtained for a real case study of a large-scale system, specifically: the Barcelona 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 [14].

5.1.1 System Management Criteria

The operational goals in the management of the Barcelona DWN 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 network under smooth control actions.

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

$$J_E(k) \triangleq |(\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2(k))^{\mathrm{T}}\mathbf{u}(k)|, \tag{26a}$$
$$J_S(k) \triangleq \|\boldsymbol{\xi}(k)\|^2, \tag{26b}$$
$$J_U(k) \triangleq \|\Delta\mathbf{u}(k)\|^2, \tag{26c}$$

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 $\boldsymbol{\xi} \triangleq \min\{\mathbf{0}, \mathbf{x} - \mathbf{x}_s\} \in \mathbb{R}^{n_x}$ 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 [14].

5.1.2 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 3.2, this system can be generally described in state-space form by (14), 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 (14b).

The states and control inputs are subject to (15); 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

$$\mathbf{x}(k) \geq \mathbf{x}_s(k) - \boldsymbol{\xi}(k) \geq \mathbf{0}, \quad \forall k, \tag{27}$$

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 $\boldsymbol{\xi} \in \mathbb{R}^{n_x}$ 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.

5.2 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 [13]. Thus, the overall system is assumed to be

[3] The performance indices considered for the case study may vary or generalized with the corresponding manipulation to include other control objectives.

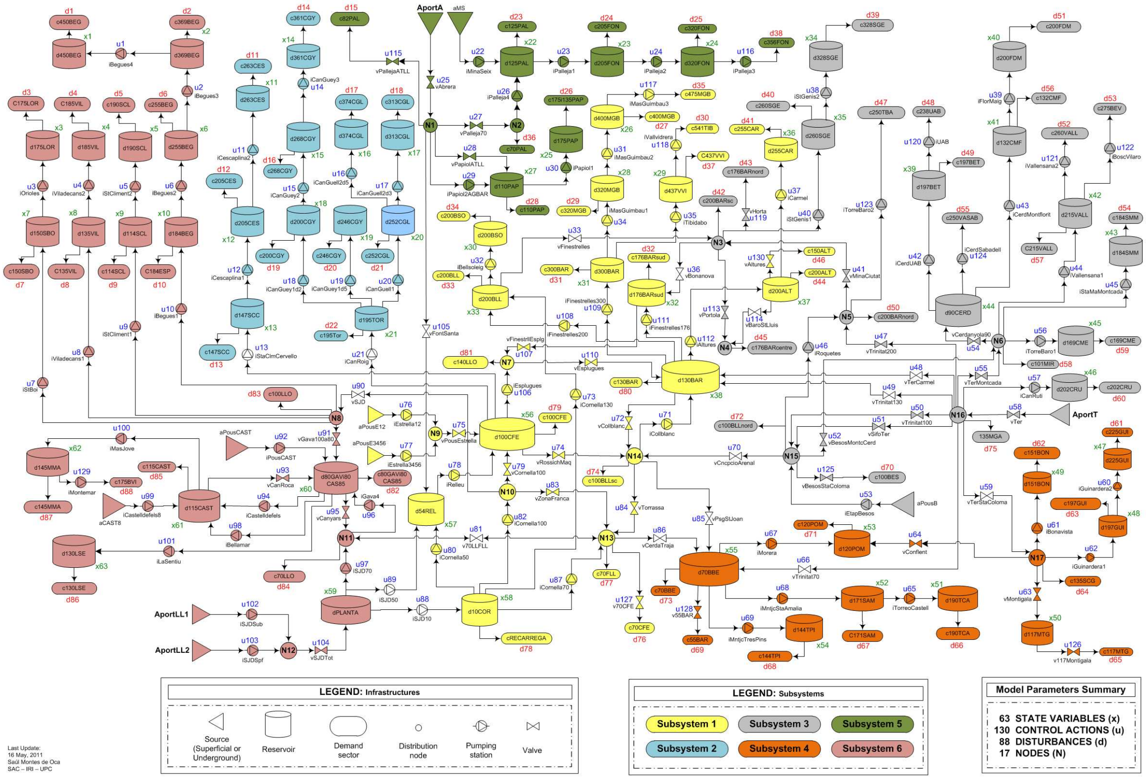


Figure 3: Partition of the Barcelona DWN

composed of six subsystems which are non-overlapped, output-decentralized and input-coupled (see Figure 5.2). The model of each subsystem is obtained for $i \in \{1, \dots, 6\}$ following Section 3.2.2 and expressed by (16). In the same way, the hard constraints of the overall DWN are partitioned and expressed by (17), while for each i-th subsystem the safety constraints are expressed by

$$\mathbf{x}_i(k) \geq \mathbf{x}_{s,i}(k) - \boldsymbol{\xi}_i(k) \geq \mathbf{0}, \quad \forall k. \tag{28}$$

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:

$$J_i^*(k) = \min_{\mathbf{u}_i^*(k:k+N_p-1)} \rho_i \left(\gamma_1 J_{E,i} + \gamma_2 J_{S,i} + \gamma_3 J_{U,i}\right), \tag{29}$$

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, $J_{S,i} \triangleq \sum_{l=1}^{N_p} \|\boldsymbol{\xi}_i(k+l)\|^2$ 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, ρ_i is a positive scalar weight to prioritize subsystems, γ_1, γ_2, and γ_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 $\mathbf{u}_i$, $\boldsymbol{\xi}_i$ and $\Delta\mathbf{u}_i$ are the i-th subsystem local variables previously defined. It can be noticed in Figure 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 4 is applied here.

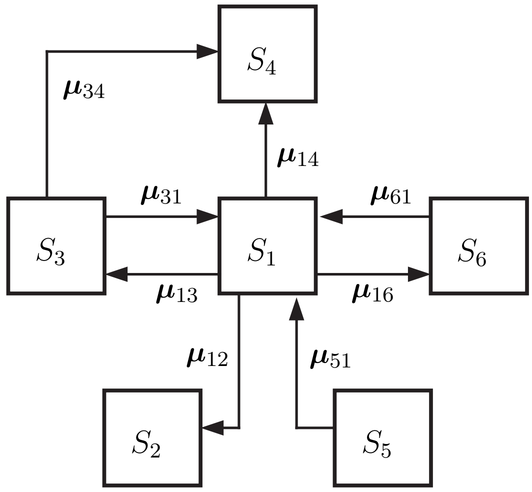


Figure 4: Network subsystems S_i and their sets of shared connections $\boldsymbol{\mu}_{ij}$

The results obtained by applying the ML-DMPC are contrasted with those of applying a CMPC approach and a non-multiayer DMPC strategy proposed in [13]. For this case study, the multi-layer optimization scheme follows Section 4, 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 (26a) 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 sampling time, and to constraints (15) and (27). 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 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 4 to perform the minimization of the local cost functions (29) subject to (16), (17), and (28), 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 (29) are $\rho_{1:6} = 1$, $\gamma_1 = 100$, $\gamma_2 = 10$ and $\gamma_3 = 0.005$. See [13] 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 levels of safety volume and control action smoothness. In order to implement the ML-DMPC approach, the demand forecasting algorithms presented in [18, 14] 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 [17].

5.3 Simulation and Preliminary results

The results of the CMPC, DMPC and ML-DMPC strategies applied to the Barcelona DWN are summarized in Table 2 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 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-multilayer 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-level optimization which allows to propagate the water cost of sources related with neighbors 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.

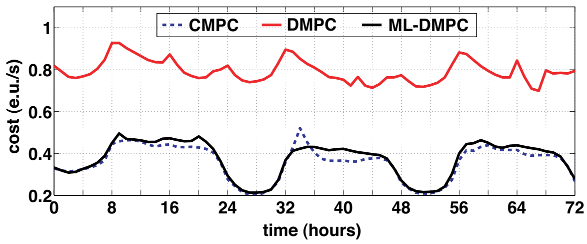
Table 2: Performance comparisons

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

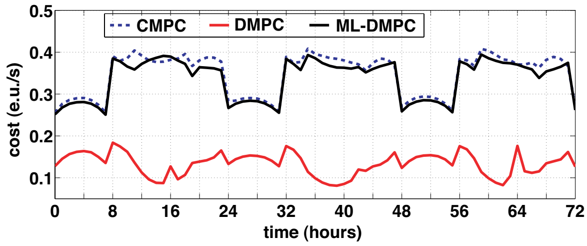
Due the difference of price between water sources and the impact of electric costs on the overall economic performance, the CMPC and ML-DMPC strategies decide to use more water from the Llobregat source despite the consequent pumping of more water through the network (see Figures 6), but achieving a 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 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.

5.4 Other results and contributions

Even in a multi-layer decentralized architecture, local MPC controllers have to be properly designed. Therefore, in addition to the work presented in [15], three approaches have been

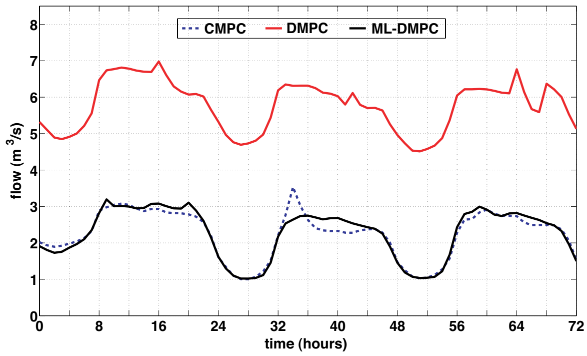


(a) Water cost

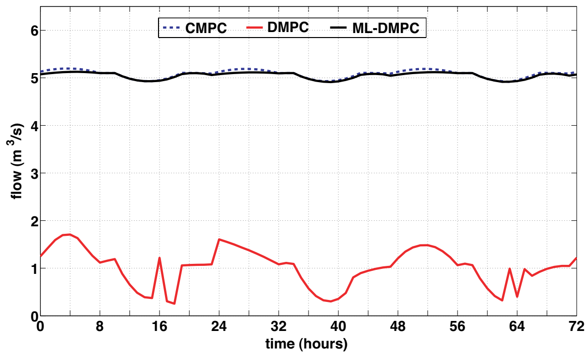


(b) Electric cost

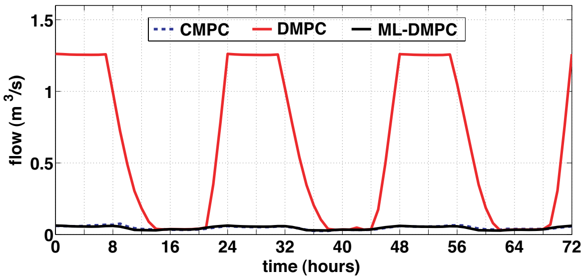
Figure 5: Economic costs of the three MPC strategies



(a) Ter source



(b) Llobregat source



(c) Abrera source

Figure 6: Total flow per water source in the Barcelona DWN

developed for the design of local controllers in the aforementioned case study by the author and the advisors of this proposal. First, a reliability-based MPC (RB-MPC) consisting in a two-layer finite horizon optimization problem that integrates, within the MPC framework, the online computation of dynamic safety stocks and the online management of actuator health degradation is proposed in [8]. Second, a chance-constrained MPC (CC-MPC) to cope with additive stochastic disturbances and to incorporate an explicit risk management mechanism that leads to a robust controller with less conservatism than common set theory-based approaches is proposed in [6]. Third, a learning-based MPC (LB-MPC) is proposed in [7] to add self-tuning capabilities to the local MPC controllers, allowing for adaptation and improvement of system performance.

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] Wei-Chen Cheng, Nien-Sheng Hsu, Wen-Ming Cheng, and William W.-G. Yeh. A flow path model for regional water distribution optimization. *Water Resources Research*, 45(9):W09411, September 2009.

[4] P.D. Christofides, R. Scattolini, D. Muñoz de la Peña, and J. Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, (0):–, 2012.

[5] W. Findeisen, F.N. Bailey, M. Brdys, K. Malinowski, P. Tatjewski, and A. Wozniak. *Control and Coordination in Hierarchical Systems*. Wiley and Sons, 1980.

[6] J.M. Grosso, C. Ocampo-Martinez, and V. Puig. Chance-constrained model predictive control for large-scale networks. *IEEE Transactions on Control Systems Technology*, 2012. (submitted).

[7] J.M. Grosso, C. Ocampo-Martinez, and V. Puig. Learning-based tuning of supervisory model predictive control for drinking water networks. *Engineering Applications of Artificial Intelligence*, 2012. (submitted).

[8] J.M. Grosso, C. Ocampo-Martinez, and V. Puig. A service reliability model predictive control with dynamic safety stocks and actuators health monitoring for drinking water networks. In *Proceedings of the IEEE Conference on Decision and Control*, 2012.

[9] J.M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, Great Britain, 2002.

[10] M.D. Mesarovic, D. Macko, and Y. Takahara. *Theory of Hierarchical Multilevel Systems*. Academic Press, 1970.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] C. Ocampo-Martinez, V. Puig, J.M. Grosso, and S. Montes de Oca. *Distributed MPC Made Easy*, chapter Multi-layer Decentralized Predictive Control of Large-Scale Networked Systems. Springer-Verlag, 2013.

[16] P.J. Van Overloop. *Model Predictive Control on Open Water Systems*. Delft University Press, Delft, The Netherlands, 2006.

[17] 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.

[18] 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.

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

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

[21] Piotr Tatjewski. Advanced control and on-line process optimization in multilayer structures. *Annual Reviews in Control*, 32(1):71–85, 2008.