Chapter 17
Non-centralized Predictive Control for Drinking-Water Supply Systems

Juan Manuel Grosso, Carlos Ocampo-Martínez and Vicenç Puig

17.1 Introduction

The control schemes proposed in the previous chapters have shown the potential applicability of *centralized* MPC for the economic scheduling control of network flows. Nevertheless, as illustrated with the case study of Barcelona, flow-based networks are generally systems comprised of multiple subsystems and/or large-scale systems. Thus, the centralization of decisions in a single MPC-based agent could be disadvantageous for the reliability of the network operation and the maintenance of the monolithic prediction model. These issues have received a lot of attention from the control research community during the last years. Several *non-centralized* control strategies have been already proposed in the literature, where either large-scale systems are partitioned into subsystems with individual control agents or a plant-wide optimization problem is distributed in a set of smaller optimization problems that are usually coordinated by a master problem. The importance of *system partitioning* and/or *distributed optimization* has already been noticed in classic references addressing the decentralized control of large-scale systems [9, 17] and the decomposition of mathematical programming problems [3]. For distributing the centralized MPC optimization problem, several analytic methods exist, e.g., Dantzig–Wolfe decomposition, Bender's decomposition, and optimality condition decomposition, among other dual or primal decomposition techniques. These analytic decompositions rely strongly on the form of both the constraints and the objective function and are specialized to particular problem structures that might not cover many real

J.M. Grosso · C. Ocampo-Martínez (✉)
Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain
e-mail: cocampo@iri.upc.edu

V. Puig
Research Center Supervision, Safety and Automatic Control (CS2AC-UPC), Terrassa, Spain

V. Puig et al. (eds.), *Real-Time Monitoring and Operational Control of Drinking-Water Systems*, Advances in Industrial Control,
DOI 10.1007/978-3-319-50751-4_17

large-scale flow-based networks. Therefore, as discussed in Chap. 16, graph theory is also used to cope with large-scale networks. Basically, the partitioning of a flow-based network consists in choosing subsets of the global variables to be assigned to different local agents that are in charge of controlling individual partitions/subsystems, as described in [5, 6, 11, 14]. This chapter addresses a large-scale network as a system-of-systems instead of analytically decomposing the global optimization problem; the corresponding partitions will be assumed given from now on.

It has been demonstrated in [16] that exchanging only interaction information (even iteratively) among the local controllers is not enough to guarantee closed-loop stability and/or optimal plant-wide performance due to their competitive behaviour. Hence, for economically optimal operation (or to reduce suboptimality) of the network, cooperation between local controllers must be induced. This can be achieved, e.g., by means of cooperative, coordinated or hierarchical MPC schemes, which incorporate negotiation/coordination mechanisms to approach the centralized solution. A crucial issue in all these non-centralized control schemes is that of guaranteeing recursive feasibility of the optimization problem, especially when addressing dynamically coupled subsystems. Among the non-centralized MPC schemes that have been proposed in the literature (see, e.g., [12] and references therein), one important classification criterion is the information exchange between local agents (e.g., predicted trajectories, prices or dual variables), which in general can be either local or global. On the one hand, there are schemes that use local information and iterative communication to improve performance but guaranteeing feasibility mostly only upon convergence to the global optimal solution. To cope with feasibility losses (e.g., due to early termination of the iterative algorithm), other non-iterative distributed MPC schemes consider the shared variables as local disturbances and rely on the design of (possibly over-conservative) robust local controllers, guaranteeing feasibility of the network at the expense of a worse economic performance. On the other hand, there exist several cooperative approaches inspired in [18], which exchange global information and ensure recursive feasibility of the optimization problem (even with non-iterative communication) by using centralized prediction models. Generally, these cooperative schemes converge asymptotically to the central optimum under certain structural assumptions, e.g., sparse couplings.

Most of the available non-centralized MPC algorithms were proposed to control systems operating under a standard (tracking) cost functions, and only few cooperative (iterative) distributed economic MPC schemes have been recently published (see, e.g., [4, 8]). Differently, this chapter proposes a non-iterative multi-layer distributed economic MPC (ML-DMPC) approach for its application to flow-based networks. This approach is based on a temporal and functional decomposition of the centralized economic scheduling-control problem. The architecture of the proposed ML-DMPC controller lies in the class of hierarchical systems [10]. Specifically, the controller comprises two layers that operate at different timescales and interact to fulfil a set $\mathcal{O}$ of desired control objectives. In a top-down hierarchy, the control structure has a centralized coordinator in the upper layer and a set of local distributed MPC controllers in the lower layer. Contrary to the standard coordinated distributed control structures [10], where the local controllers use local information and communicate iteratively only

with the coordinator to reconstruct the centralized performance, the proposed ML-DMPC scheme considers non-iterative and hierarchical-like neighbour-to-neighbour communication between the local controllers. The coordinator is used to influence (also non-iteratively) the overall performance through economic *intervention parameters*. The ML-DMPC controller aims to improve the performance of a decentralized MPC strategy (but still being globally suboptimal) and to guarantee the recursive feasibility of the involved tractable distributed algorithm.

17.2 Problem Statement

In Chap. 11, a method to obtain the monolithic state-space model of a given flow-based network graph was described. Once the control-oriented model is stated, it is important to determine the objective of performing the partitioning of the physical network no matter what control strategy is followed. For large-scale network flow problems, the partitioning of the system gains sense from the point of view of modularity of the control architecture and the reduction of computational burden. In any case, the way the network elements are interconnected is a key factor for performing the partitioning and control of the overall network since it determines the type of couplings between subsystems and consequently the complexity and rationality of the control strategy.

In the following sections, the overall system is assumed to be decomposed in a set of $M \in \mathbb{Z}_{\geq 1}$ dynamically coupled non-overlapped subsystems denoted by $\mathcal{S}_i$, $i \in \mathbb{Z}_{[1,M]}$. The number M of subsystems is generally a tuning parameter. In this chapter, a two-stage decomposition is performed. In the first stage, a reachability analysis is used to define a set of subsystems that can be supplied only by one source each. These resultant subsystems are here called *anchored subsystems* and are denoted as $\mathcal{S}_i$, $i \in \mathbb{Z}_{[1,r]}$, where $r \leq M$ is the number of flow sources in the network. The remaining elements of the network are grouped in a subsystem denoted as $\tilde{\mathcal{S}}$, which is supplied by the cross-border outflows of the anchored subsystems. Such flows are considered as *pseudosources* of $\tilde{\mathcal{S}}$. In the second stage of the decomposition, subsystem $\tilde{\mathcal{S}}$ is later subdivided into $M - r$ subsystems by means of the graph-based partitioning algorithm proposed in Chap. 16. This algorithm aims at decomposing $\tilde{\mathcal{S}}$ and its corresponding directed graph into subgraphs, in such a way that all resultant partitions have nearly the same number of vertices and a hierarchical/sequential solution order can be stated. Note that another set of pseudosources may appear after the decomposition of $\tilde{\mathcal{S}}$ and, contrary to the first stage of decomposition, each subsystem may have both entering and leaving cross-border flows depending on the interconnections of the resultant $\mathcal{S}_i$ subsystems, $i \in \mathbb{Z}_{[r+1,M]}$. A sketch of the overall decomposition process is depicted in Fig. 17.1.

Particularly, this chapter considers only input-coupled dynamics and input-coupled constraints. Then, each subsystem can be described by the following discrete-time linear model:

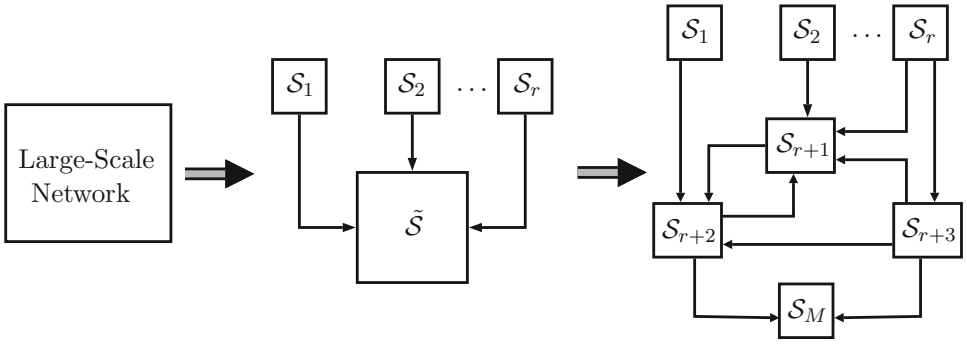


Fig. 17.1 Decomposition of a network with r sources into M subsystems

$$\begin{cases} \mathbf{x}^{[i]}(k+1) = \mathbf{A}_{ii}\mathbf{x}^{[i]}(k) + \mathbf{B}_{ii}\mathbf{u}^{[i]}(k) + \mathbf{B}_{d,ii}\mathbf{d}^{[i]}(k) + \sum\limits_{\substack{j=1 \\ j\neq i}}^{M} \mathbf{B}_{ij}\mathbf{u}^{[j]}(k), & \text{(17.1a)} \\ \mathbf{0} = \mathbf{E}_{u,ii}\mathbf{u}^{[i]}(k) + \mathbf{E}_{d,ii}\mathbf{d}^{[i]}(k) + \sum\limits_{\substack{j=1 \\ j\neq i}}^{M} \mathbf{E}_{u,ij}\mathbf{u}^{[j]}(k), & \text{(17.1b)} \end{cases}$$

for all $k \in \mathbb{Z}_+$ and $i, j \in \mathbb{Z}_{[1,M]}$, 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, respectively, the local state, input and demand vectors of subsystem $\mathcal{S}_i$, $i \in \mathbb{Z}_{[1,M]}$. Local matrices are given by the topology of each subsystem, with $\mathbf{A}_{ii} = \mathbf{I}_{n_{x_i}}$, $\mathbf{B}_{ii} \in \mathbb{R}^{n_{x_i}\times n_{u_i}}$, $\mathbf{B}_{d,ii} \in \mathbb{R}^{n_{x_i}\times n_{d_i}}$, $\mathbf{B}_{ij} \in \mathbb{R}^{n_{x_i}\times n_{u_j}}$, $\mathbf{E}_{u,ii} \in \mathbb{R}^{q_i\times n_{u_i}}$, $\mathbf{E}_{d,ii} \in \mathbb{R}^{q_i\times n_{d_i}}$ and $\mathbf{E}_{u,ij} \in \mathbb{R}^{q_i\times n_{u_i}}$ for all $i, j \in \mathbb{Z}_{[1,M]}$. The decomposition assures that $\sum_{i=1}^{M} n_{x_i} = n_x$, $\sum_{i=1}^{M} n_{u_i} = n_u$, $\sum_{i=1}^{M} n_{d_i} = n_d$ and $\sum_{i=1}^{M} q_i = q$ for all $n_{x_i}, n_{u_i}, n_{d_i}, q_i \in \mathbb{Z}_{\geq 1}$. Similarly, the global constraint sets $\mathcal{X}, \mathcal{U}$ and $\mathcal{D}$ are decomposed to give place to a set of local constraints defined by:

$$\mathbf{x}^{[j]}(k) \in \mathcal{X}_i := \{\mathbf{x}^{[i]} \in \mathbb{R}^{n_{x_i}} \mid \mathbf{0} \leq \mathbf{x}^{[i]} \leq \mathbf{x}^{[i],\max}\}, \quad \text{(17.2a)}$$

$$\mathbf{u}^{[j]}(k) \in \mathcal{U}_i := \{\mathbf{u}^{[i]} \in \mathbb{R}^{n_{u_i}} \mid \mathbf{0} \leq \mathbf{u}^{[i]} \leq \mathbf{u}^{[i],\max}\}, \quad \text{(17.2b)}$$

$$\mathbf{d}^{[j]}(k) \in \mathcal{D}_i := \{\mathbf{d}^{[i]} \in \mathbb{R}^{p_i} \mid \mathbf{0} \leq \mathbf{d}^{[i]} \leq \mathbf{d}^{[i],\max}\}. \quad \text{(17.2c)}$$

Definition 17.1 (*Neighbour and neighbourhood*) A subsystem $\mathcal{S}_j$ is defined as a *neighbour* of subsystem $\mathcal{S}_i$ if and only if $\mathbf{B}_{ij} \neq \mathbf{0}$ or $\mathbf{E}_{u,ij} \neq \mathbf{0}$, $j \in \mathbb{Z}_{[1,M]}$, $j \neq i$. Hence, the *neighbourhood* of $\mathcal{S}_i$ is defined as $\mathcal{N}_i := \{j \in \mathbb{Z}_{[1,M]} \mid \mathbf{B}_{ij} \neq \mathbf{0}$ or $\mathbf{E}_{u,ij} \neq \mathbf{0},\ j \neq i\}$.

Remark 17.1 Note that the overall system model can be obtained by the composition of the above M subsystems, as follows:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{B}_d\mathbf{d}(k), \\ \mathbf{0} = \mathbf{E}_u\mathbf{u}(k) + \mathbf{E}_d\mathbf{d}(k), \end{cases}$$

where the vectors and matrices are now a permutation of the original ones, with

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}(k)^{[1]} \\ \vdots \\ \mathbf{x}(k)^{[M]} \end{bmatrix}, \quad \mathbf{u}(k) = \begin{bmatrix} \mathbf{u}(k)^{[1]} \\ \vdots \\ \mathbf{u}(k)^{[M]} \end{bmatrix}, \quad \mathbf{d}(k) = \begin{bmatrix} \mathbf{d}(k)^{[1]} \\ \vdots \\ \mathbf{d}(k)^{[M]} \end{bmatrix}, \tag{17.3}$$

and

$$A = \begin{bmatrix} \mathbf{I}_{n_{x_1}} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{I}_{n_{x_M}} \end{bmatrix}, \quad B = \begin{bmatrix} \mathbf{B}_{11} & \dots & \mathbf{B}_{1M} \\ \vdots & \ddots & \vdots \\ \mathbf{B}_{M1} & \dots & \mathbf{B}_{MM} \end{bmatrix},$$

$$\mathbf{B}_d = \begin{bmatrix} \mathbf{B}_{d,ii} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{B}_{d,MM} \end{bmatrix}, \quad \mathbf{E}_u = \begin{bmatrix} \mathbf{E}_{u,11} & \dots & \mathbf{E}_{u,1M} \\ \vdots & \ddots & \vdots \\ \mathbf{E}_{u,M1} & \dots & \mathbf{E}_{u,MM} \end{bmatrix},$$

$$\mathbf{E}_d = \begin{bmatrix} \mathbf{E}_{d,ii} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{E}_{d,MM} \end{bmatrix}.$$

Moreover, since the dynamic and static nodes of the network were decomposed into M disjoint subsets, it follows that the global constraint sets can be recovered as Cartesian products, i.e.,

$$\mathcal{X} = \prod_{i=1}^{M} \mathcal{X}_i, \quad \mathcal{U} = \prod_{i=1}^{M} \mathcal{U}_i, \quad \mathcal{D} = \prod_{i=1}^{M} \mathcal{D}_i. \tag{17.4}$$

◊

Before getting through the design of the ML-DMPC strategy, the following preliminary assumptions related to the overall system are stated.

Assumption 17.1 All demands have a *periodic* flow request (with period $T \in \mathbb{Z}_{\geq 1}$) that can be supplied by at least one flow source through at least one flow path.[1]

Assumption 17.2 The required control objectives can be grouped in a set $\mathcal{O} = \mathcal{O}_l \cup \mathcal{O}_g$, which is a composition of a set $\mathcal{O}_l$ of *local* control objectives and a set $\mathcal{O}_g$ 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 17.2 allows to rewrite a centralized general economic stage cost function $J : \mathbb{Z}_+ \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}_+$ in the following form:

[1] A flow path is an ordered sequence of arcs, which may connect sources, intermediate nodes and demands.

$$J(k, \mathbf{x}(k), \mathbf{u}(k) = \sum_{g=1}^{m_g} \lambda_g \ell_g(k, \mathbf{x}(k), \mathbf{u}(k)) + \sum_{l=1}^{m_l} \lambda_l J_l(k, \mathbf{x}(k), \mathbf{u}(k), \tag{17.5}$$

where $\lambda_g, \lambda_l \in \mathbb{R}_+$ are scalar weights that prioritise, within the overall cost function, each global and local control objective, particularly represented by convex functions $J_g : \mathbb{Z}_+ \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}_+$ and $J_l : \mathbb{Z}_+ \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}_+$, respectively. Hence, from (17.1), (17.2) and Remark 17.1, the centralized MPC optimization problem with stage cost (17.5) and prediction horizon N can be rewritten as follows:

$$\min_{\mathbf{u}_k} \sum_{t=0}^{H_p-1} \left(\sum_{g=1}^{m_g} \lambda_g J_g(k, \mathbf{x}(k+t|k), \mathbf{u}(k+t|k)) \right.$$
$$\left. + \sum_{l=1}^{m_l} \lambda_l J_l(k, \mathbf{x}(k+t|k), \mathbf{u}(k+t|k)) \right), \tag{17.6a}$$

subject to:

$$\mathbf{x}^{[i]}(k+t+1|k) = \mathbf{A}_{ii}\mathbf{x}^{[i]}(k+t|k) + \mathbf{B}_{ii}\mathbf{u}^{[i]}(k+t|k)$$
$$+ \mathbf{B}_{d,ii}\mathbf{d}^{[i]}(k+t|k) + \sum_{\substack{j=1 \\ j\neq i}}^{M} \mathbf{B}_{ij}\mathbf{u}^{[j]}(k+t|k), \tag{17.6b}$$

$$\mathbf{0} = \mathbf{E}_{u,ii}\mathbf{u}^{[i]}(k+t|k) + \mathbf{E}_{d,ii}\mathbf{d}^{[i]}(k+t|k) + \sum_{\substack{j=1 \\ j\neq i}}^{M} \mathbf{E}_{u,ij}\mathbf{u}^{[j]}(k+t|k), \tag{17.6c}$$

$$(\mathbf{x}^{[i]}(k+t+1|k), \mathbf{u}^{[i]}(k+t|k)) \in \mathcal{X}_i \times \mathcal{U}_i, \tag{17.6d}$$

$$\mathbf{x}^{[i]}(k|k) = \mathbf{x}^{[i]}(k), \tag{17.6e}$$

for all $i \in \mathbb{Z}_{[1,M]}$ and all $t \in \mathbb{Z}_{[0,H_p-1]}$. The aggregate state and input vectors in the cost function are given by $\mathbf{x}(k+t|k) = (\mathbf{x}^{[1]^T}(k+t|k), \ldots, \mathbf{x}^{[M]^T}(k+t|k))^T$, $mathbfu(k+t|k) = (\mathbf{u}^{[1]^T}(k+t|k), \ldots, \mathbf{u}^{[M]^T}(k+t|k))^T$, respectively. The decision variable is the input sequence $\mathbf{u}_k = \{\mathbf{u}(k+t|k)\}_{t\in\mathbb{Z}_{0,H_p-1}}$.

Thus, the goal of the ML-DMPC approach proposed in this chapter is that of solving (17.6) in a distributed fashion in order to cope with the aforementioned disadvantages of a centralized controller. To do so, a set $\mathcal{C} := \{C_1, \ldots, C_M\}$ of local controllers, their communication network and a coordination mechanism are designed in the following to properly address the effect of couplings between subsystems and to take into account Assumption 17.2.

17.3 Proposed Approach

The whole ML-DMPC set-up consists of the following:

(i) an *upper* layer in charge of achieving the global objectives by solving a centralized optimization problem with a sampling time Δt_1 and
(ii) a lower layer comprising a set of distributed MPC agents that compute the references for the system actuators in order to satisfy the local objectives.

This layer operates with a sampling time Δt_2 ($\Delta t_2 \leq \Delta t_1$). The local controllers solve their associated optimization problem in a hierarchical/sequential fashion and exchange (non-iteratively) in a neighbour-to-neighbour communication strategy the predicted sequence of the inputs affecting the neighbouring subsystems. The upper layer influences the operation of the lower layer by projecting global economic information into the local agents, specifically by modifying the prices/weights of the flow arcs that are shared among the subsystems arising in the lower layer. Figure 17.2 shows the proposed control structure. The ML-DMPC scheme leads to a suboptimal plant-wide performance but with the advantage of a tractable implementation due to a hierarchical-like communication approach that avoids negotiations among local controllers. A formal description of the two optimization layers involved in the ML-DMPC approach and their interaction is given below.

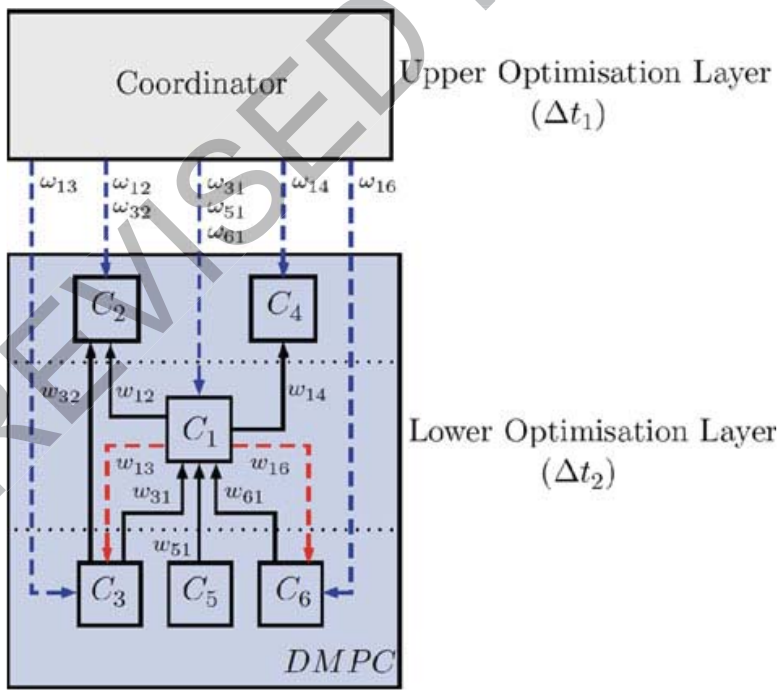


Fig. 17.2 ML-DMPC control architecture

17.3.1 Lower Optimization Layer

Once the network partitioning is performed and the M local models are obtained, it only remains to distribute the original centralized economic MPC problem among the local controllers C_i, considering the given management policies and constraints.

In order to simplify the notation, let rewrite the interaction-oriented local models as rewritten in the following more compact form:

$$\begin{cases} \mathbf{x}^{[i]}(k+1) = \mathbf{A}_{ii}\mathbf{x}^{[i]}(k) + \mathbf{B}_{ii}\mathbf{u}^{[i]}(k) + \mathbf{B}_{d,ii}\mathbf{d}^{[i]}(k) + \tilde{\mathbf{B}}_i\mathbf{w}^{[i]}(k), & \text{(17.7a)} \\ \mathbf{0} = \mathbf{E}_{u,ii}\mathbf{u}^{[i]}(k) + \mathbf{E}_{d,ii}\mathbf{d}^{[i]}(k) + \tilde{\mathbf{E}}_i\mathbf{w}^{[i]}(k), & \text{(17.7b)} \end{cases}$$

for all $i \in \mathbb{Z}_{[1,M]}$, where $\mathbf{w}^{[i]}(k) := (\mathbf{w}_{i_1}^T(k), \ldots, \mathbf{w}_{i_{|\mathcal{N}_i|}}^T(k))^T \in \mathcal{W}_i$ is a vector stacking the flows decided by the controllers of neighbours of subsystem $\mathcal{S}_i$, $\{i_1, \ldots, i_{|\mathcal{N}_i|}\}$ is an ordered sequence of the indices contained in the set $\mathcal{N}_i$ (i.e., $i_1 < \ldots < i_{|\mathcal{N}_i|}$) and $\mathbf{w}_j(k) := \mathbf{T}_{w_j}^T\mathbf{w}^{[j]}(k)$ for all $j \in \mathcal{N}_i$. In the definition of each $\mathbf{w}_j(k)$, the matrix $\mathbf{T}_{w_j} \in \mathbb{R}^{n_{u_j} \times n_{u_{ij}}}$ ($\mathbf{T}_{w_j}^T\mathbf{T}_{w_j} = \mathbf{I}_{n_{u_{ij}}}$) is such that it collects the m_{ij} ($m_{ij} < m_j$) columns of the identity matrix of order n_{u_j}, corresponding to the indices of the rows of $\tilde{\mathbf{u}}^{[j]}(k) \in \mathbb{R}^{n_{u_j}}$ related to the controlled flows decided by the controller C_j and affecting subsystem $\mathcal{S}_i$. Moreover, matrices $\bar{\mathbf{B}}_i$ and $\bar{\mathbf{E}}_i$ are suitably defined to represent the effect of $\mathbf{w}^{[i]}(k)$ on the local state vector $\mathbf{x}^{[i]}(k)$, and the set $\mathcal{W}_i$ is obtained appropriately from $\mathcal{U}_i$. In the sequel, every subsystem $\mathcal{S}_j$ that imposes an outflow $\mathbf{w}_j(k)$ to a subsystem $\mathcal{S}_i$ will be considered as a *virtual demand* of $\mathcal{S}_i$.

Interpretation 1 *At any time instant $k \in \mathbb{Z}_+$ when the controlled flow $\mathbf{u}^{[i]}(k)$ is computed, the controller C_i has knowledge of the state $\mathbf{x}^{[i]}(k)$ and the demands $\mathbf{d}^{[i]}(k)$ and $\mathbf{w}^{[i]}(k)$ imposed by the local and virtual demands, respectively. Future demands $\mathbf{d}^{[i]}(k+t)$ and $\mathbf{w}^{[i]}(k+t)$ might be unknown for all $t \in \mathbb{Z}_{\geq 1}$ and can take arbitrary values in $\mathcal{D}_i$ and $\mathcal{W}_i$, respectively. Nevertheless, the controller C_i has also knowledge of the H_p-step sequences of both the local and virtual demand expectations.*

Each controller C_i will be in charge of deciding only the network flows corresponding to subsystem $\mathcal{S}_i$ by using local and neighbouring information under Interpretation 1. In this chapter, the local problems are defined in such a way that each of them considers a local stage cost function but with a structure similar to the one in (17.5). Specifically, the stage cost function related to each C_i is written as follows:

$$J_i(k, \mathbf{x}^{[i]}(k), \mathbf{u}^{[i]}(k)) = \sum_{g=1}^{m_g} \hat{\lambda}_{g,i}\hat{J}_{g,i}(k, \mathbf{x}^{[i]}(k), \mathbf{u}^{[i]}(k)) + \sum_{l=1}^{m_l} \lambda_{l,i}J_{l,i}(k, \mathbf{x}^{[i]}(k), \mathbf{u}^{[i]}(k)), \quad \text{(17.8)}$$

where each $\hat{J}_{g,i}$, $g \in \mathbb{Z}_{[1,m_g]}$, corresponds to the gth global control objective properly expressed and weighted with a suitable $\hat{\lambda}_{g,i} \in \mathbb{R}_+$ in order to influence controllers C_i to improve plant-wide performance. Moreover, each $J_{l,i}$ is assumed to be the corresponding part of the separable local objectives J_l, $l \in \mathbb{Z}_{[1,m_l]}$, related to the subsystem $\mathcal{S}_i$.

For each subsystem $\mathcal{S}_i$, a portion of control *importance* is removed by its neighbours and added to its local uncertainty in a max-min sense due to the local knowledge considered in Interpretation 1. Hence, before fully devising the distributed MPC controllers operating in the lower layer, the following definition (adjusted from [1, Definition 4.1]) is introduced.

Definition 17.2 Denote a given *network decomposition* with $\mathcal{P} = \{\mathcal{S}_i\}_{i \in \mathbb{Z}_{1,M}}$ and let $\mathcal{C}_\infty^{\mathcal{S}_i}$ be the maximal max-min robust control invariant set for subsystem $\mathcal{S}_i$. Then, the *decentralized max-min robust control invariant set* for the overall system

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{B}_d\mathbf{d}(k), \qquad \forall k \in \mathbb{Z}_+ \tag{17.9a}$$

$$\mathbf{0} = \mathbf{E}_u\mathbf{u}(k) + \mathbf{E}_d\mathbf{d}(k), \qquad \forall k \in \mathbb{Z}_+ \tag{17.9b}$$

subject to constraints

$$\mathbf{x}(k) \in \mathcal{X} = \{x \in \mathbb{R}^{n_x} | \mathbf{0} \leq \mathbf{x} \leq \mathbf{x}^{\max}\}, \qquad \forall k \in \mathbb{Z}_+ \tag{17.10a}$$

$$\mathbf{u}(k) \in \mathcal{U} = \{\mathbf{x} \in \mathbb{R}^{n_u} | \mathbf{0} \leq \mathbf{u} \leq \mathbf{u}^{\max}\}, \qquad \forall k \in \mathbb{Z}_+ \tag{17.10b}$$

and decomposed into Δ is given by $\mathcal{C}_\infty^\Delta = \prod_{i=1}^M \mathcal{C}_\infty^{\mathcal{S}_i}$.

For a given network decomposition $\mathcal{P}$ and local sets $\mathcal{X}_i, \mathcal{U}_i, \mathcal{D}_i$ and $\mathcal{W}_i, i \in \mathbb{Z}_{[1,M]}$, each maximal max-min robust control invariant set $\mathcal{C}_\infty^{\mathcal{S}_i}$ can be explicitly computed for the overall network.

Note that such sets $\mathcal{C}_\infty^{\mathcal{S}_i}$ may result to be empty for a given $\mathcal{P}$ (consequently $\mathcal{C}_\infty^{\mathcal{P}} = \emptyset$), which implies that there is no guarantee that a decentralized control strategy will be feasibility for all times. In such a case, the sets $\mathcal{U}_i$ (accordingly $\mathcal{W}_i$), $i \in \mathbb{Z}_{[1,M]}$, should be properly modified to make possible the decentralized design of $\mathcal{C}_\infty^{\mathcal{P}}$, see e.g., [1].

Assumption 17.3 The local constraint sets arising for a given network decomposition $\mathcal{P} = \{\mathcal{S}_i\}_{i \in \mathbb{Z}_{1,M}}$ are such that

$$\mathbf{B}_{d,ii}\mathcal{D}_i \oplus \bar{\mathbf{B}}_i\mathcal{W}_i \subseteq -\mathbf{B}_{ii}\mathcal{U}_i \quad \text{and} \quad \mathbf{E}_{d,ii}\mathcal{D}_i \oplus \bar{\mathbf{E}}_i\mathcal{W}_i \subseteq -\mathbf{E}_{u,ii}\mathcal{U}_i,$$

for all $\mathcal{S}_i \in \mathcal{P}$. Hence, $\mathcal{C}_\infty^{\mathcal{S}_i} := \big((\mathcal{X}_i \oplus (-\mathbf{B}_{ii}\mathcal{U}_i)) \ominus \big(\mathbf{B}_{d,ii}\mathcal{D}_i \oplus \bar{\mathbf{E}}_i\mathcal{W}_i\big)\big) \cap \mathcal{X}_i \neq \emptyset$.

Even when Assumption 17.3 holds and $\mathcal{C}_\infty^{\mathcal{P}}$ exists, the algebraic equation (17.7b) for each local model acts as a coupling constraint that forbids the design of non-iterative distributed controllers with parallel solution of the local optimization problems. Thus, the distributed MPC algorithm considered in the lower layer of the

proposed ML-DMPC approach involves a non-iterative communication-based MPC design that builds on the hierarchical decentralized MPC approach reported in [13]. The strategy proposed here also follows a hierarchical sequence of solution but considering conditions to deal with the existence of bidirectional complicating flows between neighbour subsystems. The optimization problem to be solved in the lower layer of the ML-DMPC by each local controller C_i, $i \in \mathbb{Z}_{[1,M]}$, with sampling time Δt_2, is defined as follows:

$$\min_{\mathbf{u}_k} \sum_{t=0}^{H_p-1} \left(\sum_{g=1}^{m_g} \hat{\lambda}_{g,i} \hat{J}_{g,i}(k, \mathbf{x}^{[i]}(k+t|k), \mathbf{u}^{[i]}(k+t|k)) + \sum_{l=1}^{m_l} \lambda_{l,i} J_{l,i}(k, \mathbf{x}^{[i]}(k+t|k), \mathbf{u}^{[i]}(k+t|k)) \right), \tag{17.11a}$$

subject to:

$$\mathbf{x}^{[i]}(k+t+1|k) = \mathbf{A}_{ii}\mathbf{x}^{[i]}(k+t|k) + \mathbf{B}_{ii}\mathbf{u}^{[i]}(k+t|k) + \mathbf{B}_{d,ii}\mathbf{d}^{[i]}(k+t|k) + \bar{\mathbf{B}}_i \mathbf{w}^{[i]}(k+t|k), \quad \forall t \in \mathbb{Z}_{[0,H_p-1]} \tag{17.11b}$$

$$\mathbf{0} = \mathbf{E}_{u,ii}\mathbf{u}^{[i]}(k+t|k) + \mathbf{E}_{d,ii}\mathbf{d}^{[i]}(k+t|k) + \bar{\mathbf{E}}_i \mathbf{w}^{[i]}(k+t|k), \quad \forall t \in \mathbb{Z}_{[0,H_p-1]} \tag{17.11c}$$

$$\mathbf{x}^{[i]}(k+1|k) \in \mathcal{C}_\infty^{\mathcal{S}_i}, \tag{17.11d}$$

$$\mathbf{x}^{[i]}(k+t|k) \in \mathcal{X}_i, \quad \forall t \in \mathbb{Z}_{[2,H_p]} \tag{17.11e}$$

$$\mathbf{u}^{[i]}(k+t|k) \in \mathcal{U}_i, \quad \forall t \in \mathbb{Z}_{[0,H_p-1]} \tag{17.11f}$$

$$\mathbf{u}_{(r)}^{[i]}(k|k) = u_{(r)}^{[i]\star}(k+1|k-1), \quad \forall r \in \mathcal{I}_u \tag{17.11g}$$

$$\mathbf{x}^{[i]}(k|k) = \mathbf{x}^{[i]}(k), \tag{17.11h}$$

where $\mathcal{I}_u \subset \mathbb{Z}_+$ is a set containing the indices of all the rows of vector $\mathbf{u}^{[i]}(k)$ related to the inputs decided locally by C_i but affecting neighbours whose controllers C_j are located in higher levels of the predefined hierarchy of solution.

Comparing with the algorithms in [13, 15], problem (17.11) has two subtle but important differences:

1. The incorporation of (17.11d) as a robustness constraint that enforces the predicted state to lie within the maximal max-min robust control invariant set at the first prediction step.
2. The incorporation of (17.11f), restricting those components of the first control action that are decided locally but affect neighbouring subsystems whose controllers are located at higher levels of the solution hierarchy.

As demonstrated in [7, Chap. 6] for a min-max interpretation in a standard centralized MPC controller, the robustness constraint (17.11d) leads to a robust strongly feasible MPC algorithm. Nonetheless, this constraint on its own cannot guarantee recursive

feasibility of the overall distributed MPC solution sequence, because $\mathcal{C}_{\infty}^{\mathcal{S}_i}$ is computed under Interpretation 1, which requires that each controller C_i knows at least the first demand value of its local and virtual demands (i.e., $\mathbf{d}^{[i]}(k)$ and $\mathbf{w}^{[i]}(k)$ when solving at k). This requirement is not fulfilled if controllers C_i are allowed to freely optimize their full input vector without considering their effect in the hierarchical sequence of solution of the non-iterative ML-DMPC approach.

To illustrate this observation, assume that a controller C_j optimizes the flow of a complicating arc affecting a subsystem $\mathcal{S}_i$ whose controller C_i has already solved the ith problem in the solution sequence. Then, the trajectory obtained by C_j could be infeasible (specially due to the equality coupling constraint (17.11c)) for $\mathcal{S}_i$ since $\mathbf{w}^{[i]}(k)$ might be changed and C_i does not have the chance to recompute its solution. Hence, constraint (17.11f) is an extra necessary condition to satisfy Interpretation 1 and to maintain feasibility of the overall sequence of local problems.

17.3.2 Upper Optimization Layer

The fulfilment of a global objective from a local point of view often implies information from the entire network, but this is lost when the system partitioning is performed. Therefore, it is necessary to figure out how to induce cooperation among the set of distributed controllers, considering all the control objectives belonging to $\mathcal{O}$ in a suitable way.

One common way to improve overall closed-loop performance of a decentralized/distributed control scheme is to incorporate a supervisor controller or coordinator on top of the local controllers. Two frequently used coordination methods are the *goal coordination* and the *interaction prediction coordination* (cf., [10]). The fundamental idea behind these approaches is to have independent subproblems containing certain coordinating parameters (e.g., Lagrange multipliers, co-state variables and pseudovariables) in addition to the local decision variables. In both coordination methods, duality theory is used as a standard to construct an equivalent two-level problem to the primal (centralized) optimization problem. Within such framework, the coordinating parameters are updated iteratively by the coordinator based on the local solutions until an optimal solution to the overall system is achieved (cf. [3, 10]). Feasibility of these coordinated control strategies is guarantee only upon convergence.

Contrary to the common methods, the upper optimization layer of the ML-DMPC approach proposed in this chapter is not focused on reconstructing the centralized optimal solution in an iterative manner but to improve the economic performance of the local MPC controllers by intervening in their decision process with a low frequency of intervention. Specifically, this upper layer influences the local solutions by computing, in a non-iterative way, the weight $\omega \in \mathbb{R}^{n_\omega}$ (where n_ω is the number of arcs interconnecting the subsystems) related to the shared links between partitions that appear after the selected network decomposition method (see Fig. 17.1). The weights in ω will affect the first term in the local cost function (17.11a) of each

controller C_i, $i \in \mathbb{Z}_{[1,M]}$. Therefore, to compute ω, a centralized optimization problem based on a temporal and functional decomposition of the network is stated in the upper layer of the ML-DMPC by considering

(i) a static model of the whole network and
(ii) a cost function that only takes into account the global control objectives associated to the system.

The proposed upper optimization layer works with a sampling time $\Delta t_1 = T$, where $T \in \mathbb{Z}_{\geq 1}$ corresponds to the period of the periodic flow requested by local demands (see Assumption 17.1). Thus, when looking at the volume evolution of storage elements, they show a similar behaviour as the flow to the demands; i.e., volumes might also show a periodic behaviour with period T. For this reason, when modelling the network with sampling time Δt_1, it can be assumed that volumes do not change along the time. From now on, subindex c is used to differentiate the temporal scale of the model in the upper layer to that of the lower layer (e.g., $\mathbf{x}_c(k)$ denotes the state at the coordinator level at time instant k with sampling time Δt_1). Hence, storage nodes behave as static nodes in this layer, and the network dynamic model (17.9a) becomes a stationary model, i.e., $\mathbf{x}_c(k) = \mathbf{A_c}\mathbf{x}_c(k) + \mathbf{B_c}\mathbf{u}_c(k) + \mathbf{B}_{d,c}\mathbf{d}_c(k)$.

The stationary model considered by the coordinator is

$$J_{up}(k, \mathbf{x}_c(k), \mathbf{u}_c(k) := \sum_{g=1}^{m_g} \lambda_{g,c} J_{g,c}(\mathbf{x}_c(k), \mathbf{u}_c(k), \tag{17.12}$$

and the upper layer optimization problem is here proposed to be formulated for a flow-based network as the search of the economically optimal path flows from sources nodes to demand nodes.

Definition 17.3 (*Directed path*) Given a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ of a network, a directed path is an ordered sequence of nodes $v_1, v_2, \ldots, v_n$ in which there is an arc (i, j) pointing from each node i in the sequence to its successor node j in the sequence, that is, $\{(v_1, v_2)(v_2, v_3), \ldots, (v_{n-1}, v_n)\}$.

To mathematically and systematically find all flow paths in a given network, this chapter follows the methodology in [2, Appendix A], which exploits the information contained in the node-arc incidence matrix of the network directed graph to construct the path-arc matrix for the given sources and demands. The description of such algorithm is omitted here, and the reader is referred to the aforementioned reference. Once the path-arc matrix is obtained, a constrained optimization problem can be stated to minimize (17.12) in terms of path flows, which are denoted here as $u_p \in \mathbb{R}^{n_p}$ with n_p the number of possible paths.

Hence, the coordinator solves in the upper layer of the ML-DMPC, an optimization problem with the following structure:

$$\min_{u_p} \hat{J}_{up}(\mathbf{x}_c(k), \mathbf{u}_p(k)), \tag{17.13a}$$

subject to:

$$\mathbf{A_p}\mathbf{u}_p(k) \leq \mathbf{b}_p(k), \tag{17.13b}$$

$$\mathbf{A}_{eq}\mathbf{u}_p(k) = \mathbf{b}_{eq}(k), \tag{17.13c}$$

where function $\hat{J}_{up}$ is equivalent to (17.12) but properly expressed in terms of the path flows $\mathbf{u}_p(k)$ by using the graph path-arc matrix. Moreover, constraint (17.13b) is used to consider the physical bounds of each actuator involved in each path, while constraint (17.13c) is used to enforce satisfaction of demands $\mathbf{d}_c(k)$. Matrices $\mathbf{A_p}$ and $\mathbf{A}_{eq}$ and vectors $\mathbf{b}_p$ and $\mathbf{b}_{eq}$ are defined accordingly to the considered bounds and balance constraints.

Throughout this chapter, it has been assumed that the flow at each arc of the network is driven by an actuator. Therefore, by using the optimal solution of problem (17.13) and the information contained in the path-arc matrix of the overall network, it is possible to compute the accumulated cost incurred in traversing all the paths that reach the intermediate nodes from which the arcs interconnecting the M subsystems depart. This accumulated cost information, in addition to Assumption 17.1, allows to define the weight ω as a coordinating economic parameter. This weight is used by the coordinator to project, into the cost function of each local controller C_i, the economic impact (from a global point of view) that each subsystem $\mathcal{S}_i$ will suffer when requesting flow from its neighbour subsystems.

In network flow problems, the global objectives are often given as a composition of economic linear cost functions. In this case, the value of ω can be obtained by following Algorithm 17.1.

Note that Assumption 17.1 and the temporal scale selected for the upper layer make (17.13) independent of the state. Furthermore, the weight ω is more an intervention parameter than a coordination variable since the upper layer does not use any feedback information from the local controllers allocated at the lower layer.

17.3.3 ML-DMPC Algorithm

The sharing of information between the two layers of the proposed ML-DMPC depends on the nature and features of each application. For the case considered in this chapter (i.e., periodic demands), the interaction is unidirectional from the upper optimization layer to the lower optimization layer. 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, $i \in \mathbb{Z}_{[1,M]}$. This updating is performed with a periodicity Δt_1 to consider possible changes in the periodic pattern of demands. In fact, if a given application involves an agreement of predefined demands to be satisfied, the optimization problem of the upper layer needs to be executed only once at the beginning of the operation. In general, the computational time that the upper layer spends is quite low with respect to the computational time of the lower layer.

Algorithm 17.1 Computation of the economic intervention parameter ω

1: Compute the path-arc matrix of the network graph, denoted here by $\mathbf{R}_p \in \mathbb{R}^{n_p \times m}$.
2: Define a matrix $\mathbf{C}_p \in \mathbb{R}^{n_p \times m}$ with the same structure of matrix $\mathbf{R}_p$ but containing in each matrix element the unitary flow cost of each actuator in each possible path.
3: Identify all the arcs interconnecting subsystems $\mathcal{S}_i$, $i \in \mathbb{Z}_{[1,M]}$, and denote with $n_{u_s} \in \mathbb{Z}_+$ the number of such arcs, called from now on as complicating arcs.
4: Solve problem 17.13 and identify from the optimal solution all the paths in which each complicating arc participates, and denote by $n_{p_j} \in \mathbb{Z}_+$, $j \in \mathbb{Z}_{[1,n_{u_s}]}$, the numbers of such paths.
5: Define a set of matrices $\mathbf{T}_{s_j} \in \mathbb{R}^{n_p \times n_{p_j}}$, $j \in \mathbb{Z}_{[1,n_{u_s}]}$, each of them collecting the n_{p_j} columns of the identity matrix of order n_p.
6: Define a set of matrices $\mathbf{R}_{p_j} := \mathbf{T}_{s_j}^T \mathbf{R}_p$ and $\mathbf{C}_{p_j} := \mathbf{T}_{s_j}^T \mathbf{C}_p$ for all $j \in \mathbb{Z}_{[1,n_{u_s}]}$.
7: From the sequential order of the directed paths involved in each matrix $\mathbf{R}_{p_j}$, define a set of matrices $\tilde{\mathbf{R}}_{p_j}$ whose elements will be the same as the ones in matrices $\mathbf{R}_{p_j}$ for all the positions related to the sequential arcs that reach the complicating arcs (these latter included) in each path, and zero in those matrix elements related to the successor arcs.
8: Define the vector $\omega := (\omega_1, \ldots, \omega_{n_{u_s}})^T$, with each of its components computed as

$$\omega_j = \frac{\mathbf{1}_{n_{u_s}}^T \left(\left(\mathbf{C}_{p_j} \circ \mathbf{R}_{p_j} \right) \circ \tilde{\mathbf{R}}_{p_j} \right)^T \mathbf{T}_{s_j}^T \mathbf{u}_p^\star(k)}{\left[\mathbf{R}_{p_j}^T \mathbf{T}_{s_j}^T \mathbf{u}_p(k)^\star \right]_{(r_j)}}, \quad \forall j \in \mathbb{Z}_{[1,n_{u_s}]}$$

where $\mathbf{1}_{n_{u_s}}$ denotes an all-ones column vector of length n_{u_s}, the operator $(\circ)$ indicates the Hadamard product of matrices, and $[\cdot]_{(r_j)}$ is the r_j row of the vector in the brackets with r_j being the position of the associated jth complicating arc in the input vector $\mathbf{u}_c(k)$. Then, ω_j represents a unitary cost per flow unit.

This fact is due to the difference in the nature of the models handled by each layer and the interactions given by the distributed MPC controllers as well as their amount and disposition within the defined hierarchy. Algorithm 17.2 collects the main steps of the proposed ML-DMPC approach. The computational time spend by the scheme corresponds with the sum of maximum times of each hierarchical level of controllers.

One important property desired in the design of any MPC strategy is recursive feasibility. In the following, it is shown that the proposed ML-DMPC algorithm remains feasible for all times if initial feasibility is assumed. The guarantee of feasibility of the approach is unrelated to optimality of the distributed solution.

Theorem 17.1 *Let Assumptions 17.1–17.3 hold and suppose that an initial feasible solution in Step 1 of Algorithm 17.2 exists. Then, each local MPC problem (17.11) solved in Step 3 of Algorithm 17.2 is robust strongly feasible for each subsystem* $\mathcal{S}_i \in \mathcal{P}$.

Proof The proof is by induction, showing that feasibility at time k implies feasibility at time $k+1$. Let $\mathbf{x}^{[i]}(k)$ be a feasible initial condition for each local problem (17.11) and assume that there exists a pair of feasible (not necessarily optimal) state-input trajectories given by $(\mathbf{x}_k^{[i]}, \mathbf{u}_k^{[i]})$ for each subsystem $\mathcal{S}_i \in \mathcal{P}$.

Algorithm 17.2 Non-iterative Multi-layer Distributed Economic MPC

1: **Initialisation:** Set $k = 0$, establish an arbitrary weight ω in the upper layer and send that information to every local controller C_i, $i \in \mathbb{Z}_{[1,M]}$. For each current local state $\mathbf{x}^{[i]}(k)$ and local demand sequence $\mathbf{d}_k^{[i]} = \{\mathbf{d}^{[i]}(k), \bar{\mathbf{d}}^{[i]}(k+1|k), \ldots, \bar{\mathbf{d}}^{[i]}(k+H_p-1|k)\}$, find for all subsystems $\mathcal{S}_i$ a feasible (not necessarily optimal) pair of state and input sequences $(\mathbf{x}_k^{[i]} = \{\mathbf{x}(k+t|k)\}_{t\in\mathbb{Z}_{[0,H_p]}}, \mathbf{u}_k^{[i]} = \{\mathbf{u}(k+t|k)\}_{t\in\mathbb{Z}_{[0,H_p-1]}})$. Apply $\mathbf{u}^{[i]}(k|k)$ in every subsystem and transmit each $\mathbf{u}_k^{[i]}$ to the controllers of the corresponding neighbours of each $\mathcal{S}_i$.

2: **Collecting of information:** After receiving all the neighbour trajectories $\mathbf{u}_k^{[j]}$, $j \in \mathcal{N}_i$, each controller C_i builds the trajectory $\mathbf{w}_k^{[i]} = \{\mathbf{w}^{[i]}(k+t|k)\}_{\in\mathbb{Z}_{[0,H_p-1]}}$, differencing between shared inputs to be imposed by controllers arranged in higher levels of hierarchy and shared inputs planned by controllers arranged in the same or lower levels of hierarchy. These imposed and planned input trajectories are formed locally as $\mathbf{w}_{a,k}^{[i]} = \{\mathbf{w}_a^{[i]\star}(k|k), \ldots, w_a^{[i]\star}(k+H_p-1|k)\}$ and $\mathbf{w}_{a,k}^{[i]} = \{\mathbf{w}_b^{[i]\star}(k+1|k-1), \ldots, \mathbf{w}_b^{[i]\star}(k+H_p-1|k), \mathbf{w}_b^{[i]\star}(k+1|k-1)\}$, respectively, and it is assumed that $\mathbf{w}^{[i]}(k+t|k) = (\mathbf{w}_a^{[i]\,T}(k), \mathbf{w}_b^{[i]\,T}(k)^T$. At each sampling time, obtain $\mathbf{x}^{[i]}(k)$ and $\mathbf{d}_k^{[i]}$ for each subsystem $\mathcal{S}_i$.

3: **Solution of local problems**: Solve each optimization problem 17.11 following a predefined hierarchical sequence.

4: **Implementation of control action:** Each local controller C_i applies $\kappa_i(\mathbf{x}^{[i]}(k), \mathbf{u}_k^{[i]}, \mathbf{d}_k^{[i]}, \mathbf{w}_k^{[i]}) = \mathbf{u}^{[i]\star}(k|k)$ to the associated subsystem $\mathcal{S}_i$. Transmit each $\mathbf{u}_k^{[i]}$ to the controllers of the corresponding neighbours of each $\mathcal{S}_i$.

5: **Updating of the economic intervention parameter:** If $\lfloor k \rfloor_{\mathcal{P}_1} \in \mathbb{Z}_+$, then solve problem 17.13 for the current $\mathbf{d}_k$ and update ω following Algorithm 17.1. Send the new weight to each local controller C_i. Otherwise, go to step 5.

6: Increment k and go to step 2.

Consider now the hierarchical flow of the solution at the next time instant $k+1$. Since each subsystem applied previously the first control action of the initial feasible trajectory $\mathbf{u}_k^{[i]}$, it follows then that $x_{k+1}^{[i]} = \mathbf{x}^{[i]}(k+1|k)$, and from constraint (17.11d), it holds that $\mathbf{x}^{[i]}(k+1) \in \mathcal{C}_\infty^{\mathcal{S}_i}$ for all $i \in \mathbb{Z}_{[1,M]}$. Since $\mathcal{C}_\infty^{\mathcal{S}_i} \neq \emptyset$ by Assumption 17.3, it follows from the invariance property of $\mathcal{C}_\infty^{\mathcal{S}_i}$ that for all $(\mathbf{x}^{[i]}(k+1), \mathbf{d}_{k+1}^{[i]}, \mathbf{w}_{k+1}^{[i]}) \in \mathcal{C}_\infty^{\mathcal{S}_i} \times \mathcal{D}_i^{H_p} \times \mathcal{W}_i^{H_p}$, there exists a control sequence $\mathbf{u}_{k+1}^{[i]} \in \mathcal{U}_i^{H_p}$ such that the constraints in problem (17.11) are satisfied at time instant $k+1$ for all $i \in \mathbb{Z}_{[1,M]}$.

This claim holds only under Interpretation 1, that is, if and only if each controller C_i knows at least the first demand value of its local and virtual demands ($\mathbf{d}^{[i]}(k+1)$ and $\mathbf{w}^{[i]}(k+1)$ when solving at $k+1$). Such requirement is guaranteed by means of constraint (17.11f), which is feasible by the assumption of

existence of any initial feasible trajectory $\mathbf{u}_k$. Therefore, all the local problems solved sequentially by controllers C_i are feasible at $k+1$. Feasibility for all times follows then by induction over k and the assumption of initial feasibility. Consequently, the ML-DMPC approach is strongly feasible, and the claim is proved. □

17.4 Simulations and Results

In order to evaluate the effectiveness of the proposed ML-DMPC approach, the case study related to the model of the Barcelona DWTN is used. In this network, the set $\mathcal{O}_g$ of global control objectives is formed only by the cost function

$$J_E(\mathbf{x}(k), \mathbf{u}(k; \mathbf{c}_u(k), \mathbf{c}_x(k) := \mathbf{c}_u^T(k)\mathbf{W}_e\,\mathbf{u}(k)\Delta t + \mathbf{c}_x^T(k)\mathbf{W}_h\mathbf{x}(k), \tag{17.14a}$$

while the set $\mathcal{O}_l$ of local control objectives is formed by the cost functions

$$J_S(\mathbf{x}(k); \mathbf{s}(k) := \begin{cases} (\mathbf{x}(k) - \mathbf{s}(k)^T\mathbf{W}_s(\mathbf{x}(k) - \mathbf{s}(k) & \text{if } \mathbf{x}(k) \le \mathbf{s}(k) \\ 0 & \text{otherwise,} \end{cases} \tag{17.14b}$$

$$J_\Delta(\Delta\mathbf{u}(k)) := \Delta\mathbf{u}^T(k)\mathbf{W}_{\Delta u}\,\Delta\mathbf{u}(k). \tag{17.14c}$$

The overall network is assumed to be decomposed in six subsystems ($\mathcal{P} = \{\mathcal{S}_1, \ldots, \mathcal{S}_6\}$), which are non-overlapped, output-decentralized and input-coupled (see Fig. 16.3). The model and constraints of each subsystem $\mathcal{S}_i$ are obtained following Sect. 17.2.

The controller C_i of each subsystem $\mathcal{S}_i$ uses the following local multi-objective stage cost in its optimization problem:

$$\begin{aligned} J_i(k, \mathbf{x}^{[i]}(k), \mathbf{u}^{[i]}(k) = \hat{\lambda}_{1,i}\hat{J}_{E,i}(\mathbf{x}^{[i]}(k), \mathbf{u}^{[i]}(k; \mathbf{c}_u^{[i]}(k)) + \lambda_{2,i}J_{\Delta,i}(\Delta\mathbf{u}^{[i]}(k)) \\ +\lambda_{3,i}J_{S,i}(\mathbf{,}^{[i]}(k); \mathbf{x}^{[i]}(k), \mathbf{s}^{[i]}(k)), \end{aligned}$$

where functions $\hat{J}_{E,i}$, $J_{\Delta,i}$ and $J_{S,i}$ are the local economic, safety and smoothness objectives for subsystems $\mathcal{S}_i$ (see Sect. 17.3.1 for the derivation of the local costs). Moreover, $\hat{\lambda}_{1,i}$, $\lambda_{2,i}$ and $\lambda_{3,i}$ are positive scalar weights to prioritise each objective in the aggregate local cost function.

Each local MPC controller operates with a sampling time $\Delta t_2 = 1$ h and a prediction horizon $H_p = 24$ h. The weight $\hat{\lambda}_{1,i}$ and the internal economic parameters of each function $\hat{\ell}_{E,i}$, $i \in \mathbb{Z}_{[1,6]}$, are modified by the upper optimization layer, placing properly each element of the intervention parameter ω (see Algorithm 17.1) in the local cost of the corresponding complicated arcs. The cost function used in the upper optimization layer is given by

$$J_{up}(k, \mathbf{x}_c(k), \mathbf{u}_c(k)) = J_{E,c}(\mathbf{x}_c(k), \mathbf{u}_c(k)), \tag{17.15}$$

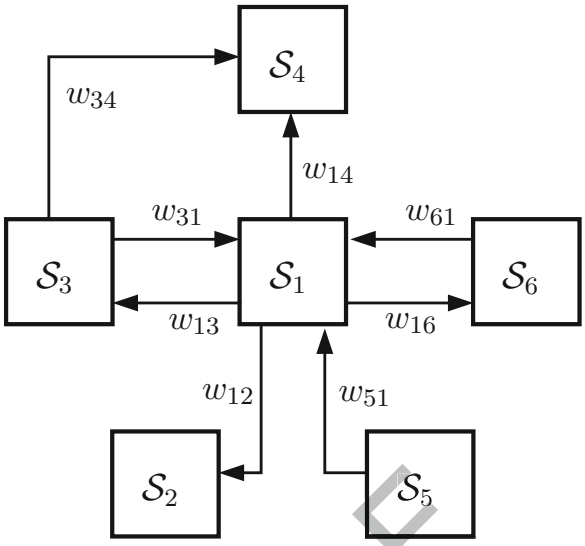


Fig. 17.3 Network subsystems $\mathcal{S}_i$ and their shared connections w_{ij}

which is derived from (17.14a) but expressed in a temporal scale of days (i.e., $\Delta t_1 = 24$ h).

The constraints and the rest of the parameters involved in the optimization problems (i.e., water demands, economic prices of water end electricity, safety thresholds) are set up according to Chap. 2.

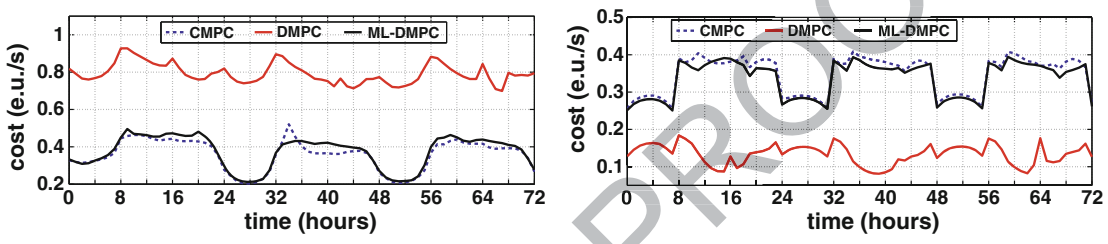
Figure 17.3 shows, in a more compact way, the resulting subsystems and the important couplings between them including their direction. Instead of neglecting the effect of this shared links as classic pure decentralized control schemes do, the ML-DMPC approach applied to the aforementioned case study has the control architecture shown in Fig. 17.2.

The results obtained by applying the ML-DMPC (Algorithm 17.2) are compared with those of applying a centralized MPC (CMPC) approach and a decentralized MPC (DMPC) strategy proposed in [14]. All of the results were obtained for a simulation horizon of 72 h with real data of the network and are summarized in Table 16.2 (Chap. 16) in terms of computational burden and of economic cost as a global management performance indicator. For each MPC approach, the computational time (in seconds) and the water, electric and total cost in economic units (e.u.) are detailed. It can be noticed that an increment of nearly 30% of the total costs of operation occurs when using the one-level hierarchical DMPC strategy reported in [14] with respect to the CMPC baseline. Despite the lower electric costs, the loss of performance in the overall cost is due to the specialized behaviour of local MPC controllers to solve their own optimization problems without knowing the real water supply cost of using shared resources with the neighbours. In contrast, the ML-DMPC outperforms the DMPC results by including the bilevel optimization, which allows to propagate the water cost of sources related with neighbour subsystems to the shared links thanks to the daily centralized control level. With this ML-DMPC approach, the level of suboptimality is acceptable comparing with the CMPC strategy; i.e., total costs are quite similar, but the computational burden is reduced. For this particular application, the computational time of the three approaches is able to satisfy the real-time constraint

Table 17.1 Performance comparisons

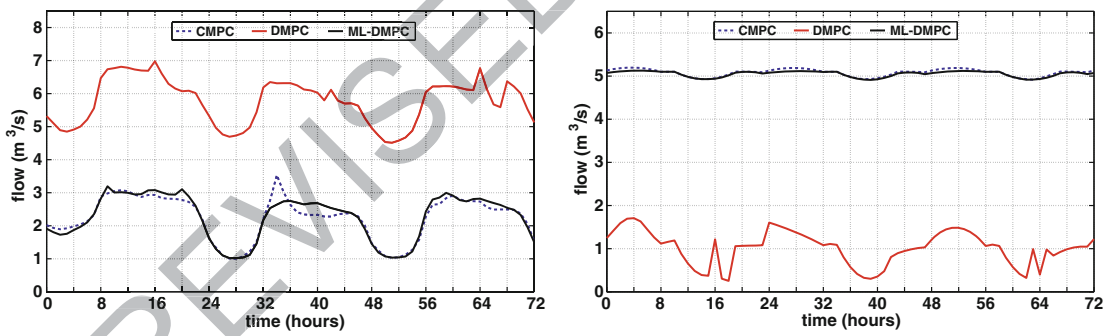
Index	CMPC	DMPC	ML-DMPC
Water cost	93.01	205.55	97.11
Electric cost	90.31	34.58	87.53
Total cost	183.33	240.13	184.65
CPU time	1143	537	540

since the control sampling time is 1 h. Thus, the main motivation for using ML-DMPC is the scalability and easy adaptability of the submodels 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 17.1).

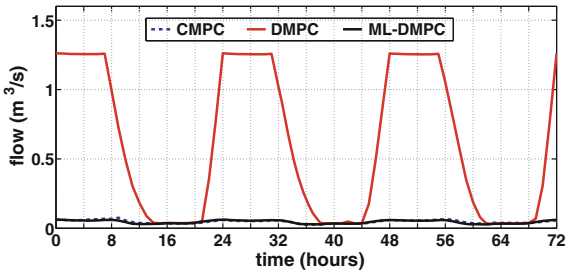


(a) Water cost (b) Electric cost

Fig. 17.4 Economic costs of the three MPC strategies



(a) Ter source (b) Llobregat source



(c) Abrera source

Fig. 17.5 Total flow per water source in the Barcelona DWTN

Due to 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 Fig. 17.5), 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 17.4 shows in detail the evolution of water cost and electric cost, respectively. These results confirm the improvement obtained by including an upper optimization layer to coordinate the local MPCs and face the lack of communication when solving their problems in a tractable way.

17.5 Conclusions

This chapter proposed a non-iterative multi-layer distributed economic MPC approach for large-scale flow-based networks. The control architecture consists in two optimization layers. The upper layer, working with a larger timescale, is in charge of improving the global performance (in general related to an optimal economic cost) by influencing a set of distributed MPC controllers by means of an intervention economic parameter. These distributed controllers are hierarchically arranged in a lower optimization layer and are in charge of determining the set point of the flow actuators to satisfy the local management/control objectives. The system decomposition is based on graph partitioning theory. Results obtained on selected simulation scenarios have shown the effectiveness of the control strategy in terms of system modularity, reduced computational burden and, at the same time, reduced loss of performance in contrast to a CMPC strategy and a hierarchical-like DMPC strategy. Additionally, it has been proved that the proposed approach results in a strongly feasible distributed MPC algorithm. For clarity of presentation, in Algorithm 17.2 it was required that each subsystem calculates its input trajectory at each time step in a hierarchical and sequential order. However, the algorithm works in the same way if non-neighbouring systems located in the same level of hierarchy solve their problems in parallel. Future work will be focused on finding stability conditions under the framework of economic MPC and also on improving the mechanism of coordination to avoid the requirement of plant-wide information in the upper layer of the ML-DMPC approach.

References

1. Barić M, Borrelli F (2010) Decentralized robust control invariance for a network of integrators. In: Proceedings of the American control conference (ACC), Baltimore, MD, USA, June 2010, pp 3939–3944
2. Cheng WC, Hsu NS, Cheng WM, Yeh W (2009) A flow path model for regional water distribution optimization. Water Resour Res 45:1–12

3. Conejo AJ, Castillo E, Minguez R, Garcia-Bertrand R (eds) (2006) Decomposition techniques in mathematical programming. Springer, Berlin
4. Driessen PAAA, Hermans RM, van den Bosch PPJ (2012) Distributed economic model predictive control of networks in competitive environments. In: Proceedings of the IEEE conference on decision and control (CDC), Maui, HI, USA, December 2012, pp 266–271
5. Jilg M, Stursberg O (2013) Optimized distributed control and topology design for hierarchically interconnected systems. In: Proceedings of the European control conference (ECC), July 2013, pp 4340–4346
6. Kamelian S, Salahshoor K (2015) A novel graph-based partitioning algorithm for large-scale dynamical systems. Int J Syst Sci 46(2):227–245
7. Kerrigan EC (2000) Robust constraint satisfaction: invariant sets and predictive control. PhD thesis, Department of Engineering, University of Cambridge, UK, November 2000
8. Lee J, Angeli D (2014) Cooperative economic model predictive control for linear systems with convex objectives. Eur J Control 20(3):141–151
9. Lunze J (1992) Feedback control of large-scale systems. Prentice Hall, Great Britain
10. Mesarović MD, Macko D, Takahara Y (1970) Theory of hierarchical multilevel systems. Academic Press, New York
11. Motee N, Sayyar-Rodsari B (2003) Optimal partitioning in distributed model predictive control. Proc Am Control Conf (ACC) 6:5300–5305
12. Negenborn RR, Maestre JM (2014) Distributed model predictive control: an overview and roadmap of future research opportunities. IEEE Control Syst Mag 34(4):87–97
13. Ocampo-Martinez C, Barcelli D, Puig V, Bemporad A
14. Ocampo-Martinez C, Bovo S, Puig V (2011) Partitioning approach oriented to the decentralised predictive control of large-scale systems. J Process Control 21(5):775–786. Special issue on Hierarchical and distributed model predictive control
15. Ocampo-Martinez C, Puig V, Grosso JM, Montes-de-Oca S (2014) Multi-layer decentralized MPC of large-scale networked systems. In: Distributed Model predictive control made easy. Springer, pp 495–515
16. Rawlings JB, Stewart BT (2008) Coordinating multiple optimization-based controllers: new opportunities and challanges. J Process Control 18(9):839–845
17. Šiljak DD (1991) Decentralized control of complex systems. Academic Press
18. Stewart BT, Venkat AN, Rawlings JB, Wright SJ, Pannocchia G (2010) Cooperative distributed model predictive control. Syst Control Lett 59(8):460–469