Delft University of Technology

Delft Center for Systems and Control

Technical report 15-011

A comparison of distributed MPC schemes on a hydro-power plant benchmark*

J.M. Maestre, M.A. Ridao, A. Kozma, C. Savorgnan, M. Diehl, M.D. Doan, A. Sadowska, T. Keviczky, B. De Schutter, H. Scheu, W. Marquardt, F. Valencia, and J. Espinosa

If you want to cite this report, please use the following reference instead:

J.M. Maestre, M.A. Ridao, A. Kozma, C. Savorgnan, M. Diehl, M.D. Doan, A. Sadowska, T. Keviczky, B. De Schutter, H. Scheu, W. Marquardt, F. Valencia, and J. Espinosa, "A comparison of distributed MPC schemes on a hydro-power plant benchmark," *Optimal Control Applications and Methods*, vol. 36, no. 3, pp. 306–332, May–June 2015.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: `https://www.dcsc.tudelft.nl`

*This report can also be downloaded via `https://pub.deschutter.info/abs/15_011.html`

A comparison of distributed MPC schemes on a hydro-power plant benchmark

J. M. Maestre, M. A. Ridao, A. Kozma, C. Savorgnan, M. Diehl, M. D. Doan, A. Sadowska, T. Keviczky, B. De Schutter, H. Scheu, W. Marquardt, F. Valencia, J. Espinosa*

March 21, 2022

In this paper we analyze and compare five distributed model predictive control (DMPC) schemes using a hydro-power plant benchmark. Besides being one of the most important sources of renewable power, hydro power plants present very interesting control challenges. The operation of a hydro-power valley involves the coordination of several subsystems over a large geographical area in order to produce the demanded energy while satisfying constraints on water levels and flows. In particular, we test the different DMPC algorithms using a 24 hour power tracking scenario in which the hydro-power plant is simulated with an accurate non-linear model. In this way, it is possible to provide a qualitative and quantitative comparison between different DMPC schemes implemented on a common benchmark, which is a type of assessment rare in the literature.

1 Introduction

Hydro-power is an important mean of renewable power generation all over the world, being the main source in countries such as Brazil [4]. Besides its contribution in terms of sheer power generation, water power deals naturally with a time-dependent demand and can be used to relieve the side effects of other renewable power sources such as wind or solar energy, which progressively stress the electrical power grid due to their uncontrollable production.

*This research was supported by the European STREP project Hierarchical and distributed model predictive control (HD-MPC, contract number INFSO-ICT-223854), the EU Network of Excellence Highly-complex and networked control systems (HYCON2, FP7/2007-2013 under grant agreement no. 257462) and the FP7-ICT DYMASOS project (under grant agreement No 611281).

Due to the increasing cost of fossil fuels, it is likely that the importance of all these renewable power sources continues to grow in the future to meet the demand for electricity. In this sense, increasing hydro-power production by just adding power plants is not sufficient, since the locations where hydro-power plants can be placed economically and ecologically are limited and often already exhausted in developed countries. Therefore, more attention should be paid to the management of water resources and hydro-power plants, so that the performance and long-term profit of these plants can be maximized while assuring additional requirements such as navigability or flood prevention. To address this challenge, a Hydro-Power Valley (HPV) benchmark[1] [30] – based on a real case study – was created in the EU project "Hierarchical and Distributed Model Predictive Control" (HD-MPC). The main objective of this benchmark is to allow quantification and comparison of distributed control methods.

In the HPV benchmark, there are several power generators placed in a river-lake system that constitute a large hydro-power plant with its corresponding structures – namely, pumps, turbines, and gates – to control the water flows. The challenge is to design efficient controllers for tracking a power production reference, while respecting environmental, operational, and safety constraints on water levels and flow rates. The nonlinear dynamics of the plant, the large spectrum of time delays, and the constraints imposed on the system variables constitute a difficult control problem that needs to be solved in real-time for the operation of the plant. These issues are common in this type of systems and have been reported several times in the literature. For example, in [4] the size of the Brazilian hydropower system and its nonlinearities are stated as major issues for system modelers.

The plants of the HPV benchmark use PID controllers in order to regulate the power and water levels to their corresponding setpoints, which are determined by off-line optimization [13]. The optimization of this type of systems has been studied during decades. For example, in [37], which is a survey carried out in the 80's, approaches based on linear, nonlinear and dynamic programming are reported as proper tools for reservoir management. More recently, [3] surveys the state of the art computational optimization methods for hydropower generation and also for other renewable and sustainable energy sources. In general, the trend is to formulate a model based optimization problem that maximizes the economical profit while taking into account the system constraints and even the potential risks [17].

In the literature and more and more in practice, Model Predictive Control

[1]The HPV benchmark is made public at the project's website: http://www.ict-hd-mpc.eu/

(MPC) [6] is proposed as the most appropriate control technique for this kind of problems due to its capability to handle complex phenomena such as multi-variable interactions, constraints, or delays in a systematic manner. To this end, MPC uses a mathematical model of the system to optimize its expected evolution over a given time horizon according to a cost function while respecting operational constraints. Due to its versatility and high performance, MPC has become very popular in industry [6, 18, 28]. This control technique has also been applied in hydro-power generation. For example, an adaptive MPC scheme is proposed in [29] to maximize the performance in the presence of varying system parameters, while MPC is proposed for regulating the turbine discharge of river power plants in [34]. In addition, real implementations of predictive controllers for hydro-power generation have been reported in [1, 8].

The implementation of MPC controllers has to address some practical challenges. A common concern is the computational burden associated with the optimization problem that has to be solved on-line for the operation of the plant. For example, in [10] Benders decomposition and importance sampling are used in order to ease the computation of a feasible solution for a hydro-power control problem. Another interesting example is [35], where real-time sequential convex programming is applied to the control of a hydro-power plant. Likewise, the fact that the hydro-power systems spread over large geographical areas demands the use of local controllers for the local subsystems that cooperate with each other, i.e., each power plant with its corresponding water system should be controlled locally. These two practical problems can be solved using a non-centralized MPC strategy. In this way, the overall control problem can be partitioned into smaller pieces that are solved by the local controllers. In case the local controllers use a communication network to coordinate the decision making procedure we speak of distributed MPC (DMPC); otherwise, i.e., when there is no communication between the controllers, we speak of decentralized MPC. In this paper, we focus on DMPC because higher control performance can be expected. In addition, DMPC presents other advantages such as its inherent modularity, which provides redundancy and simplifies the maintenance of the control system. The way the centralized problem is distributed and the type and amount of information that the local controllers (also called agents) exchange before attaining a solution to the overall control problem depend on the particular DMPC algorithm used (see [23, 25, 32] for surveys on this topic). Being a natural approach to cope with the control of large-scale systems, DMPC schemes have been proposed for the control of hydro-power networks as well as for other systems with similar dynamics, e.g., irrigation canals [14, 26, 38]. For example, in [9] DMPC with downstream communication is used to control

a cascade of river power plants, and the benefits of the communication are shown via a comparison with the corresponding decentralized MPC scheme. In [31], the structure of a hydro-power problem is exploited by means of distributed multiple shooting method. Other works that deal with hydro-power plants – specifically with the HPV benchmark – can be found in [21, 35]. Finally, it is also common to provide the local controllers with an additional centralized control layer for the sake of supervision and coordination. In this case we speak of Hierarchical DMPC (HDMPC). For example, in [20] an MPC controller that coordinates the actions of the controllers at the river barrages is proposed to optimize the hydro-power utilization of a cascade of river reservoirs. Another example is [39], where a coordinator provides the local controllers with information about the predicted interactions between them. A very recent work is [14], where the coordinator changes the communication topology that connect the local controllers to minimize the communication burden.

In this work, the HD-MPC HPV benchmark will be used to assess the performance of different DMPC schemes for hydro-power generation. The development of this field during the last decade is remarkable and a large number of schemes has been proposed, e.g.: [23] offers details regarding the implementation of 35 different approaches. Most of these schemes have been developed ad hoc for specific situations and there is little information available regarding their performance on different scenarios. Few works such as [2, 16] compare different approaches in the same plant. The lack of results in this context, which is pointed out in [25], hinders a proper comparison between the schemes available in the literature. For this reason, we have chosen a set of schemes that contains the most representative features that characterize the DMPC schemes, namely design approach (top-down and bottom up), model type (nonlinear and linear), architecture (distributed and hierarchical), and computation type (iterative and non-iterative). The schemes compared in this work are: distributed multiple shooting [31], fast gradient-based DMPC [11], game theory based DMPC [12], sensitivity-driven DMPC [33], and DMPC based on agent negotiation [22]. More specifically, we provide a thorough comparison of the schemes in different aspects: the tracking performance in terms of economic quantification, the requirements of communication, the capability of satisfying the constraints, and the computational effectiveness.

The paper is organized as follows. In Section 2, we give an overview of the HPV benchmark. The mathematical models of its components are provided and the control problem is defined. Section 3 briefly presents the different control methodologies that are tested in this paper. In Section 4, we assess the performance of the aforementioned control strategies. Finally, Section 5 provides some concluding remarks.

Finally, in order to simplify the comparison of the schemes the following notation is used throughout the paper:

- $\mathcal{N}$ denotes the set of subsystems in which the overall system is partitioned and the cardinality of this set, $|\mathcal{N}|$, will be used to denote the number of subsystems.
- In the model description, $\mathbf{x}$ is used for states; $\mathbf{u}$ for inputs/actions; $\mathbf{y}$ for outputs/measurements; $\mathbf{d}$ for disturbances.
- The discrete time index is k and t is used as continuous time index.
- T_p is the sampling time.
- N_p and N_c stand respectively for the prediction and the control horizons.
- Reference values are denoted by a bar over the corresponding variable, e.g., $\bar{\mathbf{x}}_i$ stands for the reference of $\mathbf{x}_i$.
- Vectors representing variables over N_p steps in the future are denoted as follows: $\mathbf{x}(k+1:k+N_\mathrm{p})$ represents $[\mathbf{x}^\mathrm{T}(k+1),\ldots,\mathbf{x}^\mathrm{T}(k+N_\mathrm{p})]^\mathrm{T}$.
- $\|\mathbf{z}\|_\mathbf{M}$ is the weighted Euclidean norm, i.e.,$\sqrt{\mathbf{z}^\mathrm{T}\mathbf{M}\mathbf{z}}$

Any variable or notation beyond this basic convention will be explicitly indicated in the text.

2 System overview

The system we consider is a hydro-power plant composed by 8 subsystems connected together. More specifically, in this paper we work with its complete mathematical model [30], which is based on first principles equations. Figure 1 gives an overview of the system, which is composed of 3 lakes (L_1, L_2, and L_3) and a river that is divided in 6 reaches (R_1, R_2, R_3, R_4, R_5, and R_6) that terminate with dams equipped with turbines for power production (D_1, D_2, D_3, D_4, D_5, and D_6). The lakes and the river reaches are connected by a duct (U_1), ducts equipped with a turbine (T_1 and T_2), and ducts equipped with a pump and a turbine (C_1 and C_2). The river is fed by the flows q_in and $q_\mathrm{tributary}$, while the lakes are fed by q_1, q_2, and q_3. The following assumptions have been made in order to simplify the system modeling:

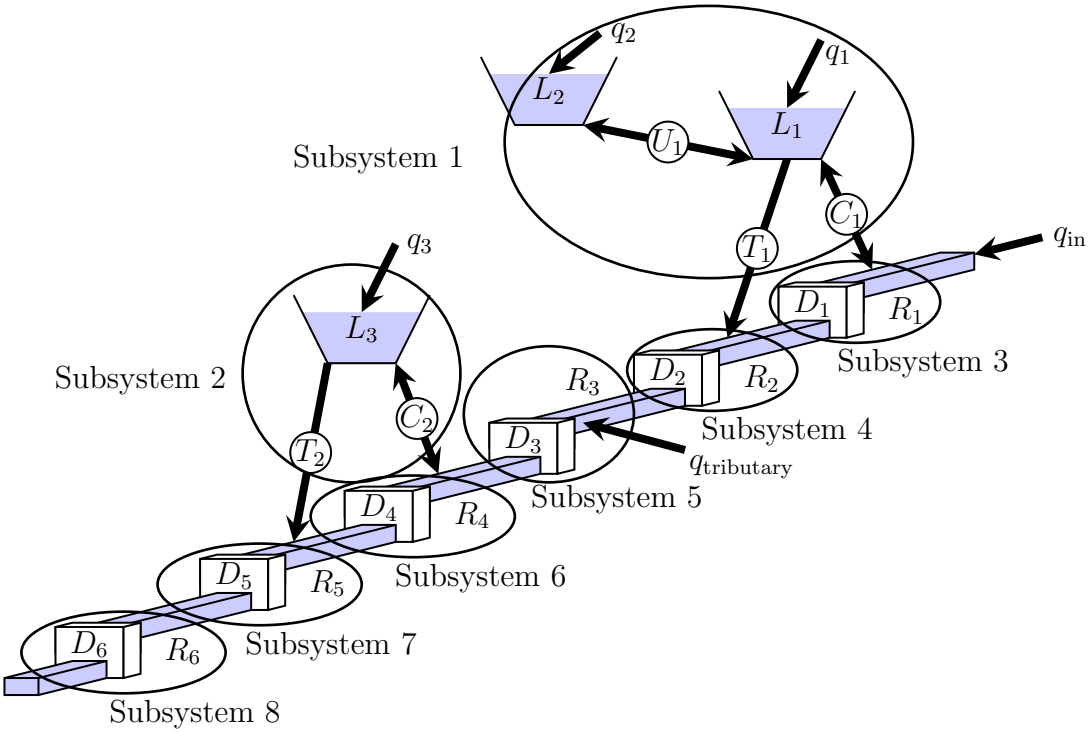


Figure 1: Overview of the hydro-power plant.

- the ducts are connected at the bottom of the lakes (or to the bottom of the river bed);
- the cross sections of the reaches and of the lakes are rectangular;
- the width of the reaches varies linearly along the reaches;
- the river bed slope is constant along every reach.

In the following subsections, we present the models that have been used for the different system components and afterwards we briefly explain how the model is used for control purposes.

2.1 Reach model

The model of the reaches is based on the one-dimensional Saint Venant partial differential equations [7], which correspond to the mass and momentum

balances:

$$\begin{aligned} &\frac{\partial q(t,z)}{\partial z} + \frac{\partial s(t,z)}{\partial t} = q_{\mathrm{l}}(z), \\ &\frac{1}{g}\frac{\partial}{\partial t}\left(\frac{q(t,z)}{s(t,z)}\right) + \frac{1}{2g}\frac{\partial}{\partial z}\left(\frac{q^2(t,z)}{s^2(t,z)}\right) + \frac{\partial h(t,z)}{\partial z} + I_{\mathrm{f}}(t,z) - I_0(z) = 0 \end{aligned} \tag{1}$$

where z is the spatial variable, which increases along the direction of flow, $q(t,z)$ is the water flow (or discharge) at time t and space coordinate z, $s(t,z)$ is the wetted surface, $h(t,z)$ is the water level w.r.t. the river bed, g is the gravitational acceleration, $I_{\mathrm{f}}(t,z)$ is the friction slope, $I_0(z)$ is the river bed slope, and $q_{\mathrm{l}}(z)$ is the lateral inflow per space unit.
Assuming that the cross section of the river is rectangular, we have

$$s(t,z) = w(z)h(t,z) \tag{2}$$

and

$$I_{\mathrm{f}}(t,z) = \frac{q(t,z)^2\left(w(z) + 2h(t,z)\right)^{4/3}}{k_{\mathrm{str}}^2\left(w(z)h(t,z)\right)^{10/3}}, \tag{3}$$

where $w(z)$ is the river width and k_{str} is the Gauckler-Manning-Strickler coefficient[2] [7].

2.2 Lake model

Since the cross section of the lake is assumed to be rectangular, the mass balance for each lake can be transformed into an equation for the water level $h(t)$,

$$\frac{dh(t)}{dt} = \frac{q_{\mathrm{in}}(t) - q_{\mathrm{out}}(t)}{S}, \tag{4}$$

where S is the cross lake surface area, and $q_{\mathrm{in}}(t)$ and $q_{\mathrm{out}}(t)$ denote the water inflow and outflow of the lake.

2.3 Duct model

The flow inside the duct U_1 can be modeled using Bernoulli's law [7]. Assuming that the duct section is much smaller than the lake surface, the flow from lake L_1 to lake L_2 can be expressed as

$$q_{U_1}(t) = S_{U_1}\,\mathrm{sign}(h_{L_2}(t) - h_{L_1}(t) + h_{U_1})\sqrt{2g|h_{L_2}(t) - h_{L_1}(t) + h_{U_1}|}, \tag{5}$$

[2]In general, the Gauckler-Manning-Strickler coefficient depends on the geometry of the river bed surface. In the model used in this work, k_{str} is taken to be constant along the river.

where h_{L_1} and h_{L_2} are the water levels for lakes L_1 and L_2, h_{U_1} is the height difference of the duct, S_{U_1} is the section of the duct, and g is the gravitational acceleration.

2.4 Turbine model

For every turbine, we assume that we can control the turbine discharge directly. The power produced is given by

$$p_{\mathrm{t}}(t) = k_{\mathrm{t}} q_{\mathrm{t}}(t) \Delta h_{\mathrm{t}}(t), \tag{6}$$

where k_{t} is the turbine coefficient, $q_{\mathrm{t}}(t)$ is the turbine discharge, and $\Delta h_{\mathrm{t}}(t)$ is the turbine head.

2.5 Pump model

Pumps can be modeled similarly to turbines. The power absorbed by a pump is given by

$$p_{\mathrm{p}}(t) = k_{\mathrm{p}} q_{\mathrm{p}}(t) \Delta h_{\mathrm{p}}(t), \tag{7}$$

where k_{p} is the pump coefficient, $q_{\mathrm{p}}(t)$ is the pump discharge, and $\Delta h_{\mathrm{p}}(t)$ is the pump head.

2.6 Modeling of ducts equipped with a turbine and a pump

The ducts C_1 and C_2 are equipped both with a pump and a turbine and therefore we can use equations (6) and (7) to express the amount of power generated or absorbed. However, the turbines and the pumps cannot function at the same time. If this fact would be modeled in a straight forward manner, the resulting optimization problem would not be differentiable. In order to overcome this issue, the so-called double flow technique is used, which consists of defining two separate positive variables to express the flow, one for the pump flow and another for the turbine flow. For example, the flow in C_1 could be modeled using these two auxiliary flows:

- $q_{C_{1\mathrm{p}}}(t)$: virtual flow such that C_1 functions as a pump,
- $q_{C_{1\mathrm{t}}}(t)$: virtual flow such that C_1 functions as a turbine.

Using these two flows, the power function associated with C_1 is replaced by two continuous functions that express the power produced ($q_{C_{1\mathrm{t}}}(t)$) and consumed ($q_{C_{1\mathrm{p}}}(t)$). This approach allows the optimization method to deal

with smooth functions only. When a solution is obtained, we combine the virtual flows to get the real flow through C_1:

$$q_{C_1}(t) = q_{C_{1\mathrm{t}}}(t) - q_{C_{1\mathrm{p}}}(t). \tag{8}$$

The duct C_2 is modeled in a similar way.

2.7 Resulting model and optimal control problem

In order to obtain a suitable nonlinear plant-replacement model, the partial differential equations are discretized in space and converted into ordinary differential equations. This model is also used by the distributed multiple shooting method, which is the only nonlinear DMPC method considered in this paper. The rest of the schemes tested in the paper are linear and consequently need a linearized version of the model. To this end, the ordinary differential equations are linearized in the steady state and discretized in time. In any case, the HPV dynamics are described with 249 states, which correspond to water levels and flows, and 10 inputs, which correspond to the flow at the pumps and turbines. In addition, two additional input variables can be considered to simplify the modeling of the ducts equipped with both a pump and a turbine. Finally, note that the state and input variables that are assigned to each local controller correspond only to its local subsystem. The control goal is to minimize the following cost function while following a power reference trajectory $\bar{p}(t)$ during a period of 24 hours:

$$\min_{\mathbf{u}(\cdot),\mathbf{x}(\cdot)} J := \int_0^{24\mathrm{h}} \gamma(t) \left| \bar{p}(t) - P(\mathbf{x}(t), \mathbf{u}(t)) \right| dt \tag{9a}$$

s.t.

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \quad t \in [0, 24h] \tag{9b}$$

$$\mathbf{x}_{\min} \leq \mathbf{x}(t) \leq \mathbf{x}_{\max} \quad t \in [0, 24h] \tag{9c}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}(t) \leq \mathbf{u}_{\max} \quad t \in [0, 24h] \tag{9d}$$

where $\mathbf{x}_i(t)$ is the state of subsystem i; $\mathbf{x}(t) = [\mathbf{x}_i(t)]_{i\in\mathcal{N}}$ is the aggregate variable of all the states of the overall system; $\mathbf{u}(t) = [\mathbf{u}_i(t)]_{i\in\mathcal{N}}$ is the aggregate variable of all the inputs; the vectors $\mathbf{u}_{\min}$, $\mathbf{u}_{\max}$, $\mathbf{x}_{\min}$ and $\mathbf{x}_{\max}$ represent the operational constraints; P is a function that calculates the power produced; $\mathbf{Q}_i$ is a weighting matrix; $\gamma(t)$ is the price of power, which is used to weight the deviation from the power production reference; $f()$ stands for the model used in the prediction. Finally, it must be noticed that the price $\gamma(t)$ and the power reference $\bar{p}(t)$ are piecewise-constant functions with their values changing every 30 minutes.

3 DMPC schemes

In this section, we briefly introduce the basics and the rationale of the DMPC schemes that are compared on the HPV benchmark. For more details about these schemes, the reader is encouraged to consult the original references. Some of the key features that allow to differentiate the DMPC schemes used in this work are:

- Type of approach, or the perspective used to design the scheme: bottom-up or top-down. In the first case, there is a group of autonomous agents that cooperate in a common control problem while in the second case a centralized control problem is partitioned into smaller problems.
- Architecture, or how the coordination between local controllers is structured. Some schemes are based one peer to peer communication while others need a higher layer that works as a coordinator. In the last case we can speak of a hierarchical DMPC.
- Type of model used to represent the plant dynamics.
- Computation type, or how the joint control actions are calculated, in an iterative or non-iterative fashion.
- Theoretical properties such as stability or optimality.

In Table 1 we can see that there are differences in the schemes considered with respect to these key features. Moreover, there are significant differences as well in other key issues such as the assumptions made by each scheme about the structure of the cost function. These differences derive from the fact that most schemes in the literature have been proposed with a particular problem set up in mind. For this reason, there is an unavoidable adaptation step in order to apply a given scheme to a concrete problem such as the HPV benchmark.

Scheme	Approach	Model Type	Architecture	Computation	Optimality	Stability
DMS	Top-down	Nonlinear	Hierarchical	Iterative	Yes	No
DMPC-BAN	Bottom-up	Linear	Distributed	Iterative	No	Yes
DAPG	Top-down	Linear	Distributed	Iterative	Yes	No
S-DMPC	Top-down	Linear	Hierarchical	Iterative	Yes	No
GT-DMPC	Bottom-up	Linear	Distributed	Non-iterative	No	Yes

Table 1: Summary of scheme basic features.

Remark 1 *Table 1 summarizes the main features of the different DMPC schemes considered in a very simplified way. In particular, we say that a scheme is optimal or stable if there are mathematical results available in this regard. However, some restrictions may apply regarding stability because of the assumptions made in the corresponding proofs. In addition, optimality for these schemes is meant only in an open-loop sense. The reader is encouraged to consult the original references to see the specific details.*

3.1 Distributed Multiple Shooting (DMS)

This approach employs a combination of direct multiple shooting [5] and domain decomposition and is regarded as Distributed Multiple Shooting [31]. In order to simplify and parallelize the computations, this method decomposes the original nonlinear optimal control problem in both the space and the time domains. The optimal control problem considered is

$$\min_{\substack{\mathbf{x},\mathbf{u},\\ \mathbf{z},\mathbf{y}}} \sum_{i=1}^{|\mathcal{N}|} \int_0^{N_\mathrm{p}T_\mathrm{p}} \ell_i(\mathbf{x}_i(t),\mathbf{u}_i(t),\mathbf{z}_i(t))dt \tag{10a}$$

s.t.

$$\dot{\mathbf{x}}_i(t) = f_i(\mathbf{x}_i(t),\mathbf{u}_i(t),\mathbf{z}_i(t)) \quad \forall i \in \mathcal{N} \tag{10b}$$

$$\mathbf{y}_i(t) = g_i(\mathbf{x}_i(t),\mathbf{u}_i(t),\mathbf{z}_i(t)) \quad \forall i \in \mathcal{N} \tag{10c}$$

$$\mathbf{x}_i(0) = \bar{\mathbf{x}}_i^0 \quad \forall i \in \mathcal{N} \tag{10d}$$

$$\mathbf{z}_i(t) = \sum_{j=1}^{|\mathcal{N}|} \mathbf{A}_{ij}\mathbf{y}_j(t) \quad \forall i \in \mathcal{N} \tag{10e}$$

$$p_i(\mathbf{x}_i(t),\mathbf{u}_i(t)) \geq 0 \quad t \in [0, N_\mathrm{p}T_\mathrm{p}], \forall i \in \mathcal{N}, \tag{10f}$$

where $\mathbf{z}_i(t)$ is the coupling input signal, which is characterized by (10e) with given coupling matrices $\mathbf{A_{ij}}$. Note that the rest of the optimization problem is decoupled.
In order to obtain a finite nonlinear program, the states are discretized by an integrator. The time domain $[0, N_\mathrm{p}T_\mathrm{p}]$ is divided into N subintervals called shooting intervals such that

$$0 = t_0 < t_1 < \cdots < t_N = N_\mathrm{p}T_\mathrm{p}. \tag{11}$$

For the n-th shooting interval and the i-th subsystem we define the initial state $\mathbf{x}_{i,n}$ and control input $\mathbf{u}_{i,n}$ and handle the integrator as a function $F_{i,n}$ that solves differential equations depending on these optimization variables and some coupling input coefficients that we detail now.

The variables that represent the coupling between subsystems, i.e., $\mathbf{y}(t)$ and $\mathbf{z}(t)$ are discretized by using Legendre polynomials. For example, the p-th element of $\mathbf{y}_i(t)$, $(\mathbf{y}_i(t))_p$, can be approximated by polynomial coefficients y_i^p as

$$(\mathbf{y}_i(t))_p = (\Gamma_m(t))^{\mathrm{T}} y_i^p, \tag{12}$$

where $\Gamma_m(t)$ is the m-th order Legendre basis. Due to the orthogonality of $\Gamma_m(t)$, the coefficient matrix $\mathbf{Y}_{i,n}$ corresponding to the signal $y_i(t)$ with $t \in [t_n, t_{n+1}]$ can be obtained by calculating the quadrature

$$\mathbf{Y}_{i,n} = \frac{2}{t_{n+1} - t_n} \int_{t_n}^{t_{n+1}} \Gamma_m(t) \left(y^i(t)\right)^T dt. \tag{13}$$

Note that on the right-hand side the integrand is a matrix and the quadrature formula has to be evaluated element-wise. Now we have all the ingredients to build the nonlinear programming problem (NLP)

$$\min_{\substack{\mathbf{u}_{i,n}, \mathbf{x}_{i,n}, \\ \mathbf{Z}_{i,n}, \mathbf{Y}_{i,n},}} \sum_{n=0}^{N-1} \left(\sum_{i=1}^{|\mathcal{N}|} L_{i,n}(\mathbf{x}_{i,n}, \mathbf{u}_{i,n}, \mathbf{Z}_{i,n}) \right) \tag{14a}$$

s.t.

$$\mathbf{x}_{i,n+1} = F_{i,n}(\mathbf{x}_{i,n}, \mathbf{u}_{i,n}, \mathbf{Z}_{i,n}) \quad n \in [0, N-1], \forall i \in \mathcal{N} \tag{14b}$$

$$\mathbf{Y}_{i,n} = G_{i,n}(\mathbf{x}_{i,n}, \mathbf{u}_{i,n}, \mathbf{Z}_{i,n}) \quad n \in [0, N-1], \forall i \in \mathcal{N} \tag{14c}$$

$$\mathbf{x}_{i,0} = \bar{\mathbf{x}}_i^0 \quad \forall i \in \mathcal{N} \tag{14d}$$

$$\mathbf{Z}_{i,n} = \sum_{j=1}^{|\mathcal{N}|} \bar{\mathbf{A}}_{ij} \mathbf{Y}_{j,n} \quad \forall i \in \mathcal{N} \tag{14e}$$

$$p_i(\mathbf{x}_{i,n}, \mathbf{u}_{i,n}) \geq 0 \quad \forall i \in \mathcal{N}. \tag{14f}$$

One can solve this problem with a Sequential Quadratic Programming method [27], which calculates the linearization of the original problem and employs corrections sequentially to the original optimization variables. The essence of distributed multiple shooting is that the evaluation of $F_{i,n}(\mathbf{x}_{i,n}, \mathbf{u}_{i,n}, \mathbf{Z}_{i,n})$ along with $\nabla F_{i,n}(\mathbf{x}_{i,n}, \mathbf{u}_{i,n}, \mathbf{Z}_{i,n})$ may be divided into $|\mathcal{N}| \times N$ independent tasks with own integration rules. This provides a massive parallelizability compared to a serial classical method.
Finally, a description of the main steps needed to implement DMS is given in Algorithm 3.1 [19].

Algorithm 3.1 ***Distributed Multiple Shooting method***

Problem preparation

1. *Introduce time mesh* $0 = t_0 \leq \cdots \leq t_N = N_p$ *and optimization variables* $\mathbf{x}_{i,n}$ *and* $\mathbf{u}_{i,n}$.

2. *Choose the order of the Legendre basis function* m *and introduce optimization variables* $\mathbf{Z}_{i,n}$ *and* $\mathbf{Y}_{i,n}$.

3. *Reformulate local dynamic equations* f_i *for each time interval* $[t_n, t_{n+1})$*: plug* $\Gamma_m(\hat{t})^T \mathbf{Z}_{i,n}$ *into* $\mathbf{z}_i(t)$.

4. *Extend local dynamic equations with quadratures that calculate local output approximations and the objective function.*

Algorithm

1. *Evaluate and linearize* $L_{i,n}(\cdot), F_{i,n}(\cdot)$ *and* $G_{i,n}(\cdot)$ *in the actual linearization point using* $|\mathcal{N}| \cdot N$ *parallel processes.*

2. *Collect linearizations at the dedicated optimizer process.*

3. *Determine the next linearization point using an NLP solver.*

4. *Communicate the new linearization point to the* $|\mathcal{N}| \cdot N$ *processes.*

5. *If convergence is achieved then quit, otherwise go to Step 1.*

3.2 DMPC based on agent negotiation (DMPC-BAN)

The DMPC scheme based on agent negotiation is proposed in [22] and its goal is to minimize a global performance index defined as the sum of the local cost functions. It assumes that each subsystem is controlled by an agent that has access only to the local model and state information. Communication between agents is allowed such that they can negotiate within the sampling period to take a cooperative decision. However, we must note that the original version of the scheme proposed in [22] optimizes a cost function whose structure is different from (9). For this reason it has been necessary to make some adjustments in order to provide a suitable solution for the HPV benchmark. In particular, the absolute value operation of (9) is replaced by a square. In addition, the water levels are assumed to be constant during the optimization problem. These changes allows us to define the local cost function for each agent i,

$$J_i(\mathbf{x}_i(0), \mathbf{u}_{\mathcal{N}_i}(0 : N_{\mathrm{p}} - 1)) = \\ \sum_{l=0}^{N_{\mathrm{p}}-1} \left(\|\mathbf{x}_i(l+1) - \bar{\mathbf{x}}_i\|^2_{\mathbf{Q}_i} + \sum_{j \in \mathcal{N}_i} \left(\|\mathbf{u}_j(l)\|^2_{\mathbf{R}_{ij}} + \mathbf{R}_j^{\mathrm{T}} \mathbf{u}_j(l) \right) \right), \tag{15}$$

where $\mathcal{N}_i$ are the set of inputs that affect the dynamics of agent i, $\mathbf{u}_{\mathcal{N}_i}(0 : N_\mathrm{p} - 1)$ is the future trajectory of inputs in $\mathcal{N}_i$, and $\mathbf{Q}_i > 0, \mathbf{R}_{ij} > 0$, and $\mathbf{R}_j$ are weighting matrices. Algorithm 3.2 details how this control strategy works.

Algorithm 3.2 ***DMPC Based on Agent Negotiation***

1. *An initial joint decision vector at time step k is built, $\mathbf{u}^\mathrm{d}_{\mathcal{N}_i}(k : k + N_\mathrm{p} - 1)$. To this end, the final joint decision vector at time step $k - 1$ is shifted such that the components corresponding to its first time step are discarded and new components are aggregated by repeating the control actions corresponding to the last step in the control horizon[3]. Likewise, the state is transmitted to the neighbors such that they all can build the matrices that characterize (15).*

2. *Randomly, agents make proposals to their neighbors. A proposal is simply a possible update of the components related with $\mathcal{N}_i$ in $\mathbf{u}^\mathrm{d}_{\mathcal{N}_i}(k : k + N_\mathrm{p} - 1)$, i.e.:*

$$\min_{\mathbf{u}_{\mathcal{N}_i}(k:k+N_\mathrm{p}-1)} \quad J_i(x_i, \mathbf{u}_{\mathcal{N}_i}(k : k + N_\mathrm{p} - 1)) \tag{16a}$$

s.t.

$$\mathbf{x}_i(l+1) = A_i \mathbf{x}_p(l) + \sum_{j \in \mathcal{N}_i} B_{pj} \mathbf{u}_j(l), \quad l \in [k, k + N_\mathrm{p} - 1] \tag{16b}$$

$$\mathbf{x}_i(l) \in \mathcal{X}_i, \quad l \in [k+1, k+N_\mathrm{p}] \tag{16c}$$

$$\mathbf{u}_j(l) \in \mathcal{U}_j, \quad \forall j \in \mathcal{N}_i, \quad l \in [k, k + N_\mathrm{p} - 1] \tag{16d}$$

$$\mathbf{u}^\mathrm{i}_j(l) = \mathbf{u}^\mathrm{d}_j(l), \ \forall j \notin \mathcal{N}_i, \quad l \in [k, k + N_\mathrm{p} - 1] \tag{16e}$$

3. *Agent i asks all the agents affected by its proposal if they are free to evaluate it (each agent can only evaluate one proposal at the time). If all the neighbors acknowledge the petition, the algorithm continues. If not, the agent waits a random time before trying again.*

4. *Each agent j affected by the proposal evaluates the difference between the cost of the new proposal $\mathbf{u}^\mathrm{i}_{\mathcal{N}_i}(k : k + N_\mathrm{p} - 1)$ and the cost of the current accepted proposal $\mathbf{u}^\mathrm{d}_{\mathcal{N}_i}(k : k + N_\mathrm{p} - 1)$ as*

$$\Delta J^i_j = J_j(\mathbf{x}_j(k), \mathbf{u}^\mathrm{i}_{\mathcal{N}_i}(k : k + N_\mathrm{p} - 1)) - J_j(\mathbf{x}_j(l), \mathbf{u}^\mathrm{d}_{\mathcal{N}_i}(k : k + N_\mathrm{p} - 1)) \tag{17}$$

[3]See [22] for a more sophisticated aggregation of components which in addition can be used to guarantee the stability of the closed-loop system.

This difference ΔJ_j^i is sent back to the proposer agent i. If the proposal does not satisfy the constraints of the corresponding local optimization problem, an infinite cost increment is assigned. This implies that unfeasible proposals will never be chosen.

5. *Agent i receives the local cost increments from all the agents affected by its proposal, such that it can evaluate its impact $\Delta J^i(k)$ as*

$$\Delta J^i = \textstyle\sum_j \Delta J_j^i \tag{18}$$

If ΔJ^i is negative, $\mathbf{u}^{\mathrm{d}}_{\mathcal{N}_i}(k : k + N_{\mathrm{p}} - 1) = \mathbf{u}^{\mathrm{i}}_{\mathcal{N}_i}(k : k + N_{\mathrm{p}} - 1)$ and agent i broadcasts the update in the joint decision vector. Otherwise, the proposal is discarded.

6. *Go to Step 1 until the maximum number of proposals have been made or the time available for placing proposals is over.*

7. *Apply the first component of $\mathbf{u}^{\mathrm{d}}_{\mathcal{N}_i}(k : k+N_{\mathrm{p}}-1)$ and repeat the procedure in the next time step.*

3.3 DMPC based on distributed optimization using accelerated proximal gradient method (DAPG)

The scheme presented here is a distributed optimization algorithm that has fast convergence properties [11, 15]. It assumes that each subsystem is controlled by an agent that can communicate with a group of neighboring agents, and that the whole group of control agents cooperates to obtain a globally optimal solution of the MPC at every sampling time.
The algorithm presented here is based on the accelerated gradient method [11, 15], which can deal with problems posed in a general form as

$$\min_{\tilde{\mathbf{x}}_{\mathrm{dec}}} \quad J := \frac{1}{2}\|\tilde{\mathbf{x}}_{\mathrm{dec}}\|^2_{\mathbf{H}} + \gamma|\mathbf{P}\tilde{\mathbf{x}}_{\mathrm{dec}} - \mathbf{p}| \tag{19a}$$

s.t.

$$\mathbf{A}_1\tilde{\mathbf{x}}_{\mathrm{dec}} = \mathbf{B}_1 \tag{19b}$$

$$\mathbf{A}_2\tilde{\mathbf{x}}_{\mathrm{dec}} \leq \mathbf{B}_2 \tag{19c}$$

where $\tilde{\mathbf{x}}_{\mathrm{dec}}$ contains all decision variables stacked up for the entire prediction period. The matrices $\mathbf{A}_1 \in \mathbb{R}^{q\times n}$, $\mathbf{A}_2 \in \mathbb{R}^{r\times n}$ and $\mathbf{P} \in \mathbb{R}^{m\times n}$ have sparse structures, and the matrix $\mathbf{H} \in \mathbb{R}^{n\times n}$ is positive definite and block-diagonal with block elements $\mathbf{H}_i \in \mathbb{R}^{n_i\times n_i}$.

Problem (9) has to be adapted first to the form (19) to properly deal with the absolute term representing the total power-reference tracking to make the problem separable. To this end, the subsystems are provided with additional flexibility in choosing the appropriate local power-reference to be tracked. The main idea is that each subsystem "trades" an amount of power with its neighbors. Thus, we define for each pair (i, j) a pair of power exchange variables for each predicted sampling time:

$$\delta_{ij}(l) = \delta_{ji}(l), \quad l = 0, \ldots, N_{\mathrm{p}} - 1. \tag{20}$$

Here, we assign either subsystem i or subsystem j to "lead" the exchange between them. A simple way to do that is to let the subsystem with a smaller index lead the exchange, i.e. agent i leads exchange for its neighbors $j \in \Delta_i = \{j | j \in \mathcal{N}, j > i\}$. Then it is possible to replace the absolute term in (9) by:

$$\sum_{i=1}^{8} \left| \bar{p}_i(l) + \sum_{j \in \Delta_i} \delta_{ij}(l) - \sum_{j \in \mathcal{N} \setminus \Delta_i} \delta_{ij}(l) - \mathbf{P}_i \mathbf{x}(l) \right| \tag{21}$$

with $\bar{p}_i$ the nominal power reference for subsystem i, and subject to the constraints (20) for all pairs of (i, j). In other words, the local power reference for each subsystem i deviates from the nominal value by adding the exchange amounts of the links that i manages and subtracting the exchange amounts of the links that affect i but are decided upon by its neighbors. Notice that the new cost function in (21) is decomposable, and the additional constraints (20) can be dualized easily without expanding the neighbor set of each subsystem. Hence, the optimization problem can be cast into (19), such that vector $\tilde{\mathbf{x}}_{\mathrm{dec}}$ contains also the new decision variables δ_{ij}. The constraints (19b) and (19c) represent dynamical equations, physical bound constraints on water levels and flows, and the additional constraints (20).
The dual problem of (19) is to minimize the convex function

$$f(z) = \frac{1}{2} \|\mathbf{z}\|^2_{\tilde{\mathbf{A}} \mathbf{H}^{-1} \tilde{\mathbf{A}}^T} + \tilde{\mathbf{B}}^T \mathbf{z} \tag{22}$$

with

$$\tilde{\mathbf{A}} = [\mathbf{A}_1^T \ \mathbf{A}_2^T \ \mathbf{P}^T]^T \qquad \tilde{\mathbf{B}} = [\mathbf{B}_1^T \ \mathbf{B}_2^T \ \mathbf{p}^T]^T \qquad \mathbf{z} = [\lambda^T \ \mu^T \ \nu^T]^T$$

where λ, μ, and ν are Lagrange multipliers for the constraints (19b), (19c), and $\mathbf{P}\tilde{\mathbf{x}}_{\mathrm{dec}} - \mathbf{p} = 0$, respectively.
We also denote each column of $\tilde{\mathbf{A}}$ by

$$\tilde{\mathbf{A}} = [\mathbf{a}_1 \ \cdots \ \mathbf{a}_{q+r+m}]^T \tag{23}$$

with $\mathbf{a}_l \in \mathbb{R}^n, l = 1, \ldots, q+r+m$. Note that each $\mathbf{a}_l, l = 1, \ldots, q+r+m$ is composed of the components that correspond to the variables of subsystems, i.e.,

$$\mathbf{a}_l = [\mathbf{a}_{l1}^T, \ldots, \mathbf{a}_{l8}^T]^T, \quad l = 1, \ldots, q+r+m \tag{24}$$

in which $\mathbf{a}_{li} \in \mathbb{R}^{n_{\tilde{\mathbf{x}}_i}}, i = 1, \ldots, 8$, where $n_{\tilde{\mathbf{x}}_i}$ is the size of the i^{th} subsystem variable $\tilde{\mathbf{x}}_{\text{dec},i}$.
In order to implement the proposed distributed algorithm, we introduce the sets $\mathcal{L}_i, i = 1, \ldots, 8$, containing indices $l \in \{1, \ldots, q+r+m\}$ of constraints that are assigned to subsystem i. Each constraint should be in only one set $\mathcal{L}_i$. It is decided that $l \in \mathcal{L}_i$ if $\mathbf{a}_{li} \neq 0$.
With proper choice for the step size L (see below), the minimization of (22) is solved by the following distributed algorithm.

Algorithm 3.3 ***Distributed Accelerated Proximal Gradient***
Initialize $\mathbf{z}^0 = \mathbf{z}^{-1}$ *and* $\tilde{\mathbf{x}}_{\text{dec}}^{-1}$ *with the last values from the previous sampling step. For the first sampling step, these variables are initialized as zeros.*
In every node, i*, the following computations are performed:*
For $p = 0, 1, 2, \ldots,$

1. *Compute*

$$\tilde{\mathbf{x}}_{\text{dec},i}^p = -\mathbf{H}_i^{-1} \sum_{j \in \mathcal{N}_i} \left(\sum_{l \in \mathcal{L}_j} z_l^p \mathbf{a}_{li} \right) \tag{25a}$$

$$\mathbf{v}_i^p = \frac{2p+1}{p+2} \tilde{\mathbf{x}}_{\text{dec},i}^p - \frac{p-1}{p+2} \tilde{\mathbf{x}}_{\text{dec},i}^{p-1} \tag{25b}$$

2. *Send* $\mathbf{v}_i^p$ *to each* $j \in \mathcal{N}_i$*, receive* $\mathbf{v}_j^p$ *from each* $j \in \mathcal{N}_i$

3. *Compute with each* $l \in \mathcal{L}_i$

$$c_l^p = \sum_{j \in \mathcal{N}_i} \mathbf{a}_{lj}^T \mathbf{v}_j^p - b_l \tag{26a}$$

$$z_l^{p+1} = z_l^p + \frac{p-1}{p+2}(z_l^p - z_l^{p-1}) + \frac{1}{L} c_l^p, \text{ if } l \leq q \tag{26b}$$

$$z_l^{p+1} = \max \left\{ 0, z_l^p + \frac{p-1}{p+2}(z_l^p - z_l^{p-1}) + \frac{1}{L} c_l^p \right\}, \text{ if } q < l \leq q+r \tag{26c}$$

$$z_l^{p+1} = \min \left\{ \gamma, \max \left[-\gamma, z_l^p + \frac{p-1}{p+2}(z_l^p - z_l^{p-1}) + \frac{1}{L} c_l^p \right] \right\}, \tag{26d}$$

$$\text{if } q+r < l \leq q+r+m \tag{26e}$$

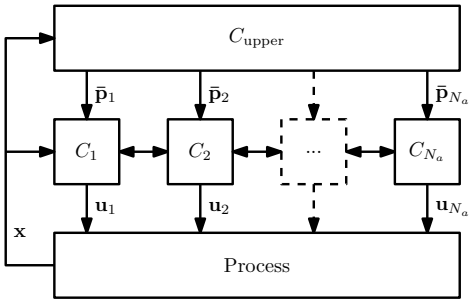


Figure 2: Two layer control architecture with S-DMPC

4. *Send* $\{z_l^{p+1}\}_{l\in\mathcal{L}_i}$ *to each* $j \in \mathcal{N}_i$,
   receive $\{z_l^{p+1}\}_{l\in\mathcal{L}_j}$ *from each* $j \in \mathcal{N}_i$.

The appropriate choice of the step size L is discussed in [15]. Algorithm 3.3 has been shown to converge to the optimal solution of problem (19), and its convergence rate is faster than standard gradient-based algorithms [15].

3.4 Two layer architecture with sensitivity-driven DMPC (S-DMPC)

As the optimal control problem arising in the HPV benchmark involves multiple time scales, i.e., a long time horizon for the prediction of the 24 hour load cycle and shorter time scales arising from high-frequency disturbances caused by plant-model mismatch (here between the nonlinear plant-replacement model and the linear controller model), we propose a two-layer control architecture as depicted in Fig. 2.
The upper layer controller C_{upper} is an standard MPC with a linear model and a quadratic cost function, sampled every 30 minutes, with a prediction and control horizon of 24 hours, i.e., $N_{\text{p}} = N_{\text{c}} = 48$. However, the non-standard power tracking objective (9) is reformulated as a quadratic cost function

$$J^{\text{upper}} = \sum_{l=k}^{k+N_{\text{p}}-1} \left(\bar{p}(l) - \sum_{i=1}^{|\mathcal{N}|} p_i(l) \right)^2 \tag{27}$$

for the upper layer controller in order to make it solvable with standard QP solvers. The purpose of this controller is to distribute the power-load among the power plants to be tracked by the lower-layer distributed controller, i.e., to provide reference trajectories $\bar{p}_i,\ i \in \mathcal{N}$.

On the lower layer, the sensitivity-driven DMPC (S-DMPC) algorithm [33] is implemented for higher-frequency disturbance rejection and is sampled every 90 seconds[4]. For this purpose, the linear quadratic constrained control problem is transcribed offline into a standard QP of the form

$$\min_{\mathbf{u}(k)} \frac{1}{2} \|\mathbf{u}(k)\|_{\mathbf{H}}^2 + \mathbf{f}^T(\mathbf{x}(k))\, \mathbf{u}(k) \tag{28a}$$

s.t.

$$\mathbf{A}\, \mathbf{u}(k) + \mathbf{b}(\mathbf{x}(k)) \geq 0 \ , \tag{28b}$$

where $\mathbf{u}(k)$ and $\mathbf{x}(k)$ indicate the deviations of the input and the initial state at time index k from the nominal steady-state values.
For the online algorithm of S-DMPC, the QP (28) is decomposed into $|\mathcal{N}|$ smaller problems resulting in $|\mathcal{N}|$ distributed controllers for the $|\mathcal{N}|$ subsystems. Each distributed controller considers only part of the QP, namely the control parameters related to the corresponding subsystem, as well as the constraints related to that subsystem. Consequently the matrices and vectors are partitioned as follows: $\mathbf{H} = (\mathbf{H}_{i,j})_{i,j\in\mathcal{N}}$, $\mathbf{A} = (\mathbf{A}_{i,j})_{i,j\in\mathcal{N}}$, $\mathbf{f} = (\mathbf{f}_i)_{i\in\mathcal{N}}$, $\mathbf{b} = (\mathbf{b}_i)_{i\in\mathcal{N}}$, and $\mathbf{u}_k = (\mathbf{u}_{k,i})_{i\in\mathcal{N}}$. Then, each of the controllers iteratively solves the following smaller QP, involving only local decision variables and constraints:

$$\begin{aligned}\min_{\mathbf{u}_i(k)} J_i \triangleq & \frac{1}{2} \|\mathbf{u}_i(k)\|_{\mathbf{H}_{i,i}}^2 + \mathbf{f}_i^T(\mathbf{x}(k))\, \mathbf{u}_i(k) \\ & + \left(\sum_{j\in\mathcal{N}\setminus\{i\}} \mathbf{u}_j^{Tp}(k)\, \mathbf{H}_{j,i} - \lambda_j^{Tp}\, \mathbf{A}_{j,i} \right) \mathbf{u}_i(k), \end{aligned} \tag{29a}$$

s.t.

$$\mathbf{A}_{i,i}\, \mathbf{u}_i(k) + \sum_{j\in\mathcal{N}\setminus\{i\}} \mathbf{A}_{i,j}\, \mathbf{u}_j^p(k) + \mathbf{b}_i(\mathbf{x}(k)) \geq 0. \tag{29b}$$

Here the superscript p indicates the Lagrange multipliers λ_i and the optimal control parameters $\mathbf{u}_i(k)$ of the p-th iteration. The cost function in (29a) is composed of two parts: on the one hand it includes the local objective function, in addition it contains linear information on the other controllers. The full algorithm of the lower layer S-DMPC controller is summarized in the following, while for details of the S-DMPC algorithm we refer to [33]:

Algorithm 3.4 ***Sensitivity-driven DMPC [33]***

[4]The distribution of the power-load with the objective function (27) cannot be solved with the S-DMPC algorithm, as the Hessian matrix of that function is only positive-semidefinite and thus the algorithm is not necessarily applicable [33].

1. *Set time* $k := 0$ *and fix the initial system state* $\mathbf{x}(k)$.

2. *Transcribe the optimal control problem to compute* $\mathbf{H}$, $\mathbf{f}$, $\mathbf{A}$, *and* $\mathbf{b}$; $\mathbf{H}$ *and* $\mathbf{A}$ *do not depend on the initial state* $\mathbf{x}(k)$ *and need to be computed only once.*

3. *Select inital parameters* $\mathbf{u}_i^0(k)$ *and an estimate of the initial Lagrange multipliers* $\lambda_i^0(k)$ *and set* $p := 0$.

4. *Send the control parameters* $\mathbf{u}_i^p(k)$ *and the Lagrange multipliers* $\lambda_i^p(k)$, $\forall i \in \mathcal{N}$, *to the distributed controllers.*

5. *Solve QP* (29) $\forall i \in \mathcal{N}$ *to obtain the minimizer* $\mathbf{u}^{p+1}(k)$ *and the Lagrange multiplier* $\lambda^{p+1}(k)$.

6. *Increase iteration index* p, *i.e.* $p := p + 1$ *and go back to 4.*

7. *Stop iteration, if* $\mathbf{u}^p(k)$ *satisfies some convergence criterion.*

8. *Apply the calculated optimal system inputs to the distributed plant.*

9. *Set time index* $k := k + 1$, *determine the new initial state* $\mathbf{x}(k)$ *and go back to 2.*

3.5 Bargaining Approach to Optimal Control (GT-DMPC)

This DMPC scheme [2, 12, 36] is based on the axiomatic theory developed by J. Nash for bargaining games [24]. In particular, the scheme assumes that the whole system model can be decomposed into $|\mathcal{N}|$ linear subsystems such that

$$\mathbf{x}_i(k+1) = \sum_{j=1}^{|\mathcal{N}|} \mathbf{A}_{ij}\mathbf{x}_j(k) + \mathbf{B}_{ij}\mathbf{u}_j(k) \tag{30}$$

where $\mathbf{x}_i(k)$ and $\mathbf{u}_i(k)$ are respectively the state and the input vector of each subsystem. In this scheme the MPC optimization problem is written as

$$\min_{\mathbf{u}_{\mathcal{N}}(k:k+N_\mathrm{p}-1)} \sum_{i=1}^{|\mathcal{N}|} J_i(\mathbf{u}_{\mathcal{N}}(k); \mathbf{x}(k)) \tag{31a}$$

s.t.

$$\mathbf{u}_i(k) \in \Omega_i, \ \forall \ i \in \mathcal{N} \tag{31b}$$

where Ω_i is a set of the feasible control actions determined by the physical and operational limits of subsystem i and

$$J_i(\mathbf{u}_{\mathcal{N}}(k);\mathbf{x}(k)) = \|\mathbf{u}_{\mathcal{N}}^T(k)\|^2_{\mathbf{Q}uui} + 2\mathbf{x}^T(k)\mathbf{Q}_{xui}\mathbf{u}_{\mathcal{N}}(k) + \|\mathbf{x}^T(k)\|^2_{\mathbf{Q}xxi}$$

being $\mathbf{Q}_{uui} \geq 0, \forall i \in \mathcal{N}$.[5]

From a game-theoretical point of view, it is possible to describe the interactive decision making problem as the following strategic form game:

$$G_{\text{DMPC}} = (\mathcal{N}, \{\Omega_i\}_{i\in\mathcal{N}}, \{J_i(\mathbf{u}_{\mathcal{N}}(k))\}_{i\in\mathcal{N}}). \tag{32}$$

$\mathcal{N}$ is the set of players, Ω_i is the set of actions of player i, and $J_i(\mathbf{u}_{\mathcal{N}}(k)) : \Omega_1 \times \ldots \times \Omega_M \to \mathbb{R}$ determines the cost assigned to each player as the outcome of the game. In addition, let us define the disagreement point $\eta_i(k)$ as the maximum cost player i is willing to accept in order to collaborate with the rest. This value is updated at each time step according to

$$\eta_i(k+1) := \begin{cases} \eta_i(k) - \alpha(\eta_i(k) - J_i(\mathbf{u}_{\mathcal{N}}(k))) \text{ if } \eta_i(k) \geq J_i(\mathbf{u}_{\mathcal{N}}(k)) \\ \\ J_i(\widetilde{\mathbf{u}}(k)) \text{ if } \eta_i(k) < J_i(\mathbf{u}_{\mathcal{N}}(k)) \end{cases}$$

for some $\alpha \in \mathbb{R},\ 0 < \alpha < 1$. According to this rule, whenever the local controller i decides to cooperate, i.e., when its disagreement cost $\eta_i(k)$ is greater than $J_i(\mathbf{u}_{\mathcal{N}}(k))$, its disagreement point is reduced by a certain amount. Otherwise $J_i(\mathbf{u}_{\mathcal{N}}(k))$ is directly assigned as the new value of the disagreement point. Notice that the latter case implies an increment of the player's disagreement cost $\eta_i(k)$ and hence there is an incentive for cooperation. Given that each player is willing to maximize the difference between its current costs and its disagreement point, the solution of this bargaining game can be computed from

$$\min_{\mathbf{u}_{\mathcal{N}}(k)} \sum_{i=1}^{|\mathcal{N}|} \omega_i \log(\eta_i(k) - J_i(\mathbf{u}_{\mathcal{N}}(k))) \tag{33a}$$

s.t.

$$\eta_i(k) > J_i(\mathbf{u}_{\mathcal{N}}(k)) \tag{33b}$$

$$\mathbf{u}_i(k) \in \Omega_i \tag{33c}$$

where $\omega_i > 0,\ \sum_{i=1}^{|\mathcal{N}|} \omega_i = 1$. In order to solve (33) in a distributed fashion it is required that subsystems communicate to each other the current values of their states, control actions and disagreement points. Once the subsystems have this information, the optimization problem of (33) is locally solved by

[5]For simplicity we will just write $J_i(\mathbf{u}_{\mathcal{N}}(k))$ instead of $J_i(\mathbf{u}_{\mathcal{N}}(k);\mathbf{x}(k)))$ in the remainder of this subsection.

considering fixed values of the inputs and states of the remaining subsystems. The particular negotiation model employed by this scheme is non-iterative and it is based on that presented in [24] for two-person games. Regarding the closed-loop stability of the overall system, in [36] conditions are derived. The steps to solve the DMPC game are summarized in Algorithm 3.5 [2, 12, 36].

Algorithm 3.5 ***Bargaining approach***

1. *At time step k, each subsystem broadcasts the values of the state $\mathbf{x}_i(k)$.*
2. *With the information received, each subsystem solves the local optimization problem (33).*
3. *Let $\mathbf{u}_i^*(k)$ denote optimal control actions for subsystem i. If (33) is feasible, subsystem i selects the first control action of $\mathbf{u}_i^*(k)$ as a control action. Otherwise, subsystem i continues with the previous agreed trajectory. To this end, $\mathbf{u}_i^*(k-1)$ is shifted so that its first component is discarded and zeros are inserted in the last one.*
4. *Each subsystem updates its disagreement point.*
5. *Each subsystem broadcasts its updated control action $\mathbf{u}_i^*(k)$ and disagreement point $\eta_i(k)$.*
6. *Go to Step* 1.

4 Results

In this section, we show the results from the tests that we have performed with the schemes described in Section 3. In order to measure their performance, we use the following indicators:

- Integral of absolute tracking error (IATE): the absolute value of the tracking error with respect to the power reference integrated during the whole simulation, i.e.,

$$\int_0^{24\text{h}} \left| \bar{p}(l) - P(\mathbf{x}(t), \mathbf{u}(t)) \right| dt. \tag{34}$$

- Economic tracking error (ETE): the absolute value of the tracking error with respect to the power reference weighted by the corresponding electricity price integrated during the whole simulation, i.e.,

$$\int_0^{24\text{h}} \gamma(t) \left| \bar{p}(l) - P(\mathbf{x}(t), \mathbf{u}(t)) \right| dt. \tag{35}$$

Note that this indicator corresponds with the objective function to be minimized (9).

- Economic tracking error 2 (ETE2): a variation of ETE in which the power overproduction gets half the penalty of underproduction, i.e.,

$$\begin{aligned}&\int_0^{24\text{h}} \gamma(t)\Big(\max\left(\bar{p}(l) - P(\mathbf{x}(t), \mathbf{u}(t)), 0\right)\\&+\tfrac{1}{2}\max\left(P(\mathbf{x}(t) - \bar{p}(l), \mathbf{u}(t)), 0\right)\Big)\, dt.\end{aligned} \tag{36}$$

  The rationale of this second economic indicator is to *reward* the power overproduction with respect to the underproduction, which is worse from the point of view of demand satisfaction.

- Communication costs: the raw communication burden of the scheme measured as the number of floats transmitted at each time step.

- Constraint violation: the integral of the constraints violation. As it can be seen in Figures 4-8, the water levels of the lakes and the reaches, which are depicted in blue, do not always satisfy their corresponding constraints, depicted in red. This parameter sums the area between the blue and the red lines whenever there is a violation. Hence, it provides a numerical indicator regarding the level of constraint satisfaction of each controller. Finally, note that input constraints are always satisfied because inputs are decision variables.

- Sum of absolute input distance with respect to setpoint (SAIDS): it is the sum for all the inputs of a given scheme of the absolute difference between the input value and its corresponding steady state value during the simulated period. The rationale of this indicator is to provide information regarding which schemes implement control actions closer to the steady state values.

$$\sum_{i=1}^{10}\sum_{k=1}^{47}\left|u_i(k+1) - \bar{u}_i\right|. \tag{37}$$

- Sum of absolute input increments (SAII): this indicator sums for each control scheme the absolute difference between consecutive samples of its inputs during the simulated period. The rationale of the indicator is to provide information regarding which schemes cause the most and the least abrupt changes in the input signals.

$$\sum_{i=1}^{10}\sum_{k=1}^{47}\left|u_i(k+1) - u_i(k)\right|. \tag{38}$$

It is worth noting that the computation time has not been used for comparison in this work because the controllers were not designed to optimize the computation time. In addition, the type of implementation has not been uniform for all the controllers (for example, one used C++ and others Matlab). For this reason, we have focused instead on the number of decision variables and the communication burden of each controller to provide the reader with indicators regarding the complexity of each scheme and its corresponding implementation. In any case, the computation times of all the schemes were much lower than the sampling time of the plant.
In the following, we will present the results for each of the schemes considered, which are summarized in Table 2. In Figure 3, the planned power production and the reference power within 1 day are shown. Note that some of the power curves are barely noticeable because they follow the reference with great precision. In addition, the difference between the power generated by each scheme and the reference is shown in Figure 4. Likewise, Figures 5 to 8 show the evolution of the inputs and outputs of the system (blue lines) together with their corresponding constraints (red lines).

4.1 DMS

This scheme obtains the best results in the benchmark. The IATE obtained by DMS was 0.06, and the ETE and ETE2 indices were respectively 4 and 3. This result is not surprising since the DMS scheme solves the centralized nonlinear control problem in a distributed fashion without any model mismatch and without unknown disturbances. More specifically, the solution found is optimal (though not highly accurate) because it converges to that of the equivalent centralized optimization problem, with KKT tolerance 0.258831, primal infeasability 1.08×10^{-12}, and dual infeasability 1.67. For this reason, we can use its results as a bound on the performance that can be achieved by other schemes based on a linear model.
The evolution of the water levels and the inputs of this scheme can be seen in Figure 5. These outstanding results are obtained after solving a nonlinear problem which has altogether 20661 variables, 2154 inequalities, 20025 equalities (besides the 249 states and 12 manipulated variables of the model, 162 auxiliary variables are used by this method) whose corresponding computations are distributed in a total of 8×48 processes. Finally, regarding the communication burden of this scheme, the centralized controller sends $6 \times 8 \times 48 = 2034$ vectors, and receives $5 \times 8 \times 48 = 1920$ vectors and $5 \times 8 \times 48$ matrices in each iteration via a message passing interface that uses double precision. In terms of the floating numbers transmitted, the centralized controller sends 16368 floats and receives 6010800.

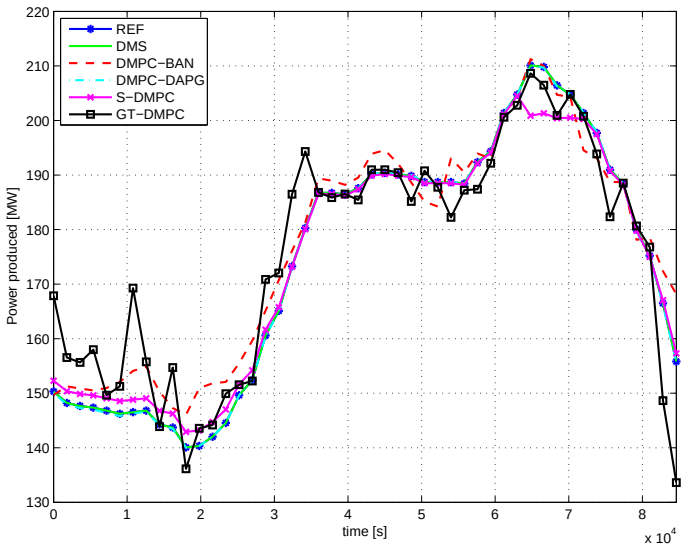


Figure 3: Comparison of the reference power and the power generated.

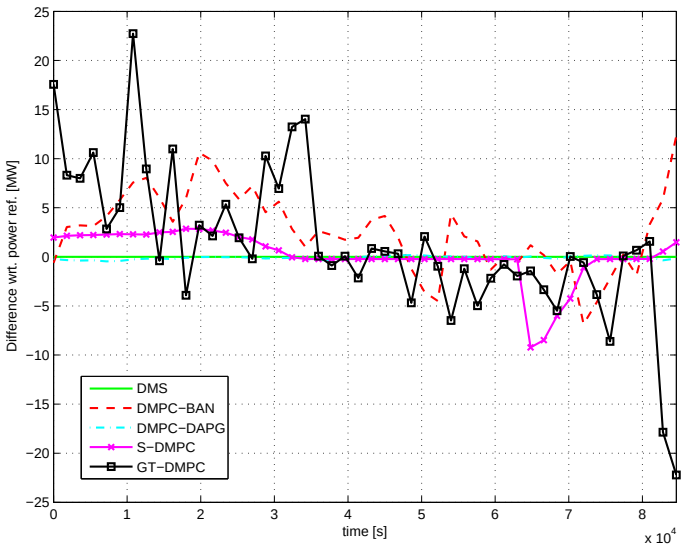


Figure 4: Difference between the power generated and the power reference.

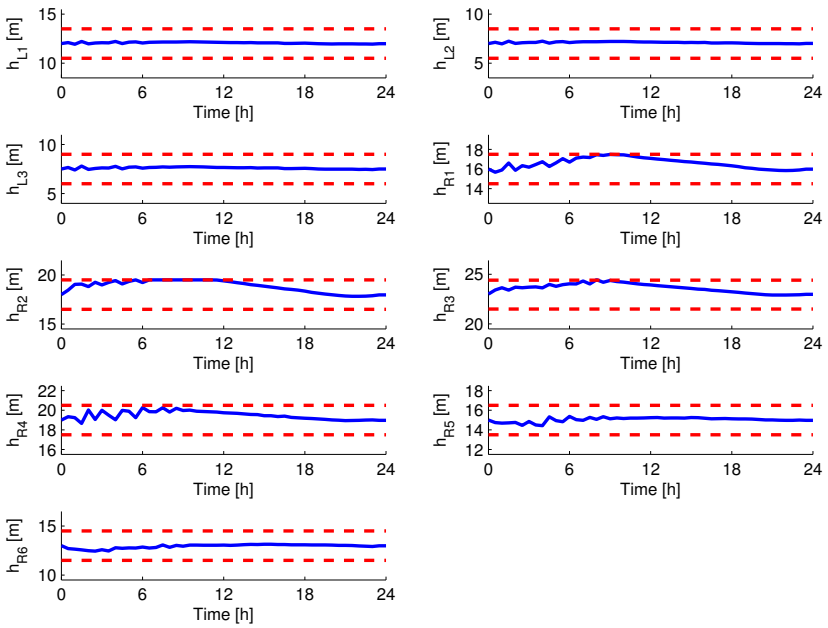
Finally, the SAIDS and SAII indicators for this scheme were respectively 12637.91 and 10489.28.

4.2 DMPC-BAN

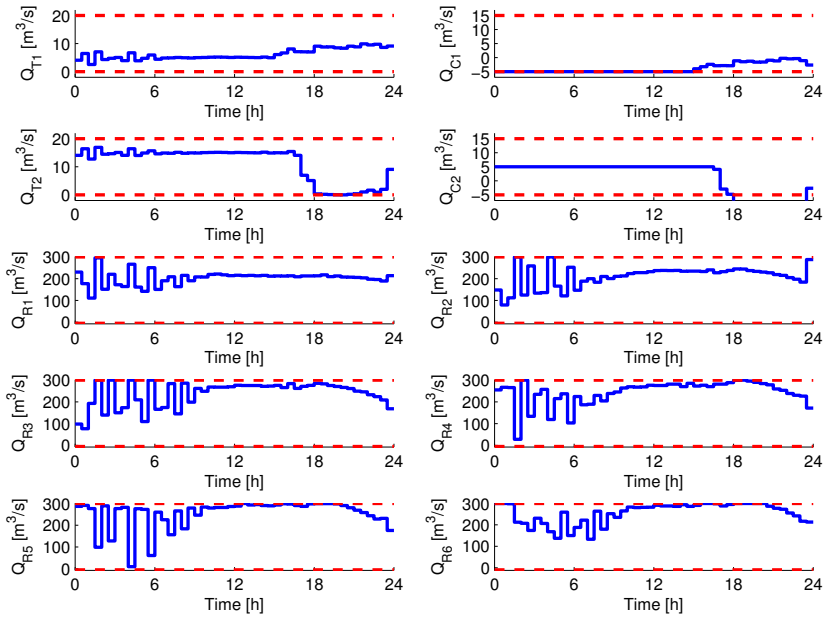
The IATE, ETE, and ETE2 obtained by this scheme were, respectively, 89.09, 5808 and 3388. These results can be explained because of the strict assumptions made by the scheme: all the agents use a linear model and have strict local information. Hence, the cooperating capabilities are limited. Nevertheless, notice that these values are in the same order of magnitude of other more sophisticated schemes. The evolution in time of the water levels and the inputs is depicted in Figure 6.
Regarding the communication burden of the scheme, the total number of proposals at each sample period was set to 50. Given that the control horizon N_c was set to 10, each proposal was composed of a maximum of 30 floating point reals. In addition, at the beginning of each time step, the value of water levels is transmitted so that the power generation matrix can be calculated, i.e., 8 additional floats are transmitted. Regarding the computational burden, each local controller only has to solve its optimization problem once. A total number of 100 optimization variables is calculated. Finally, we must point out that there were two small violations during the simulations, one in the water level of lake L_2 (0.035m) and another one in the water level at reach R_1 are violated (-0.05m). In total, the constraint violation integral is 0.13.
Finally, the SAIDS and SAII indicators for this scheme were respectively 8152.11 and 2915.82.

4.3 DAPG

The IATE, ETE and ETE2 obtained by this scheme were, respectively, 3.86, 250, and 229. As it can be seen in Figure 3, this distributed controller follows the reference accurately, which is not surprising since again this controller solves a centralized control problem in a distributed fashion. The evolution of the water levels and the inputs are shown in Figure 7.
A bound on the communication burden of this scheme is not easy to calculate since it depends on the total number of iterations needed until convergence is obtained. In the simulation performed an average of 580 iterations was needed for convergence at each time step. Taking into account that a total of 2100 floats were trasmitted each iteration, there are approximately 1.2 millions floats transmitted in a sample. Regarding the constraint violations, we must point out that while there is no violation for input constraints, there

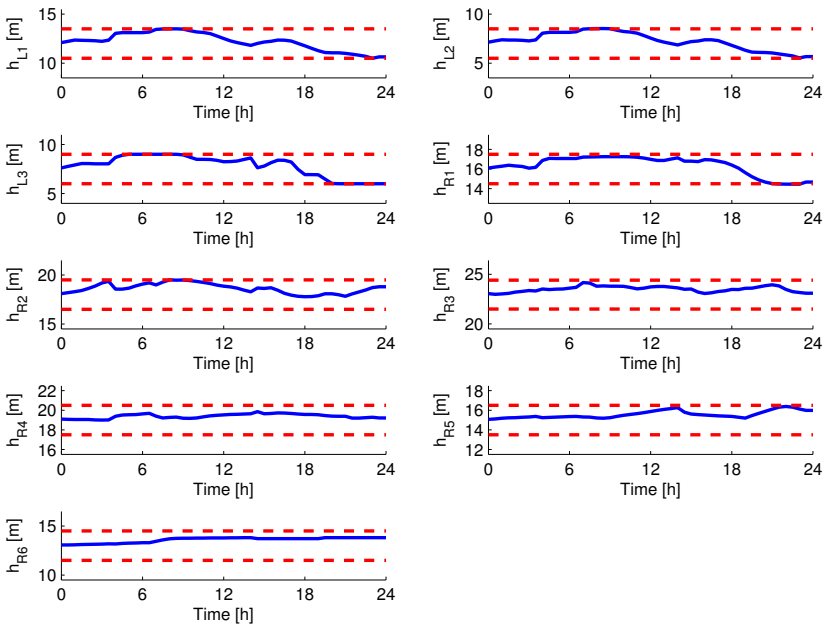


(a)

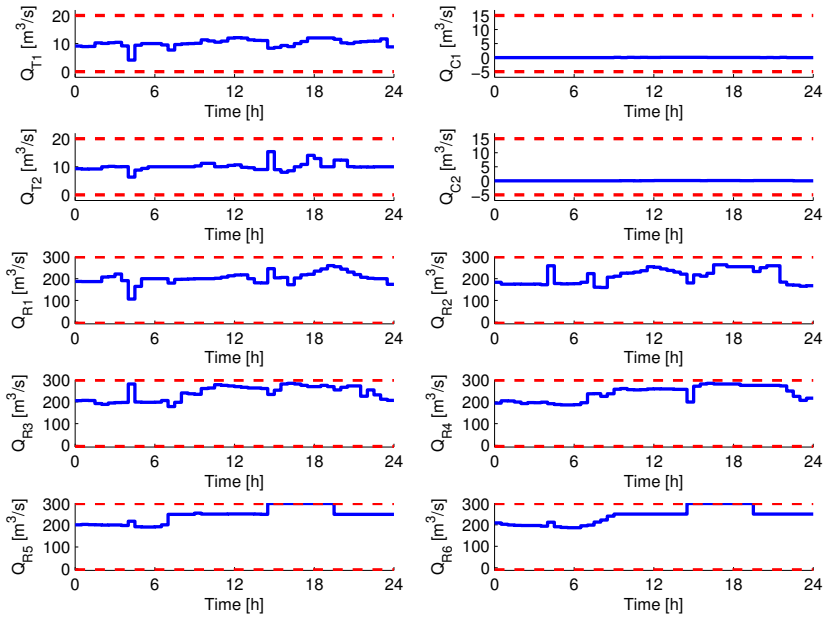


(b)

Figure 5: a) Water levels in different reaches and lakes together with the constraints along 24 hours corresponding to the DMS scheme. b) Control plan of reaches and lakes together with constraints for 24 hours corresponding to the DMS scheme

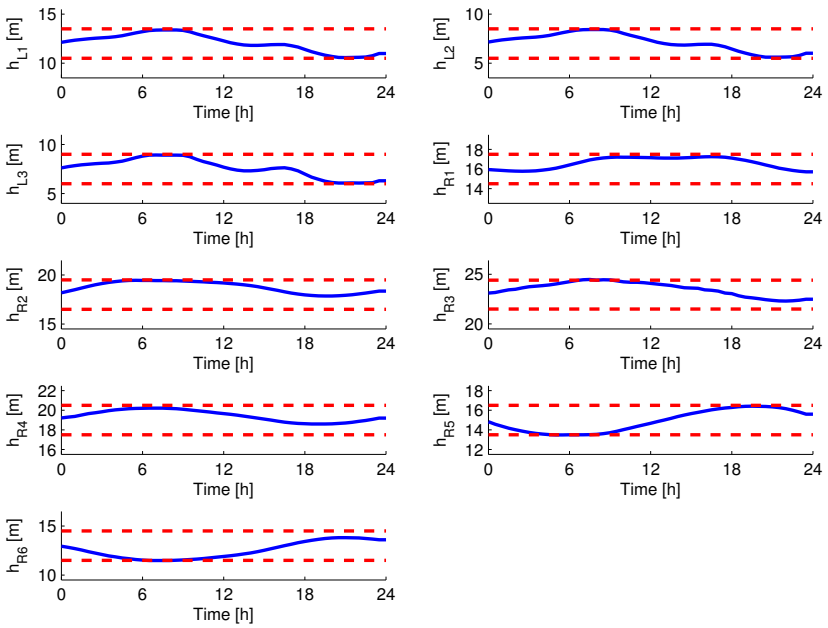


(a)

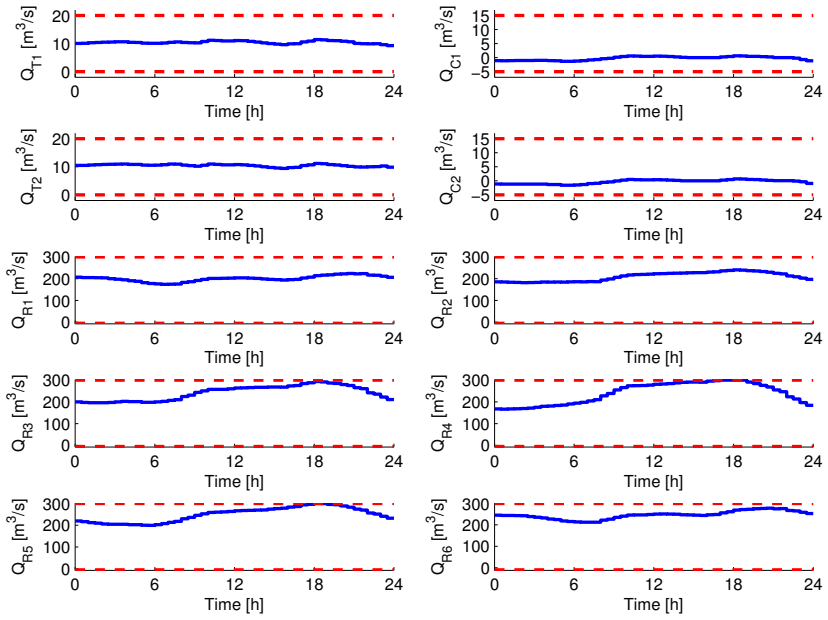


(b)

Figure 6: a) Water levels in different reaches and lakes together with the constraints along 24 hours corresponding to the DMPC-BAN scheme. b) Control plan of reaches and lakes together with constraints for 24 hours corresponding to the DMPC-BAN scheme.



(a)



(b)

Figure 7: a) Water levels in different reaches and lakes together with the constraints along 24 hours corresponding to the DAPG scheme. b) Control plan of reaches and lakes together with constraints for 24 hours corresponding to the DAPG scheme.

is a small violation in the output constraints of the water levels of the reaches R_5 and R_6. The integral of these violations is 0.62.
Finally, the SAIDS and SAII indicators for this scheme were respectively 7434.22 and 1034.83.

4.4 S-DMPC

The IATE, ETE and ETE2 indices obtained by this scheme were, respectively, 36.72, 2419 and 1788. As it can be seen in Figure 3, the performance of the distributed controller is good, in fact similar to that of DAPG. The additional centralized layer plays an important role because it improves the coordination of the local controllers and the overall performance. The evolution of the water levels and of the inputs are shown in Figure 8. Regarding the communication burden, the following holds:

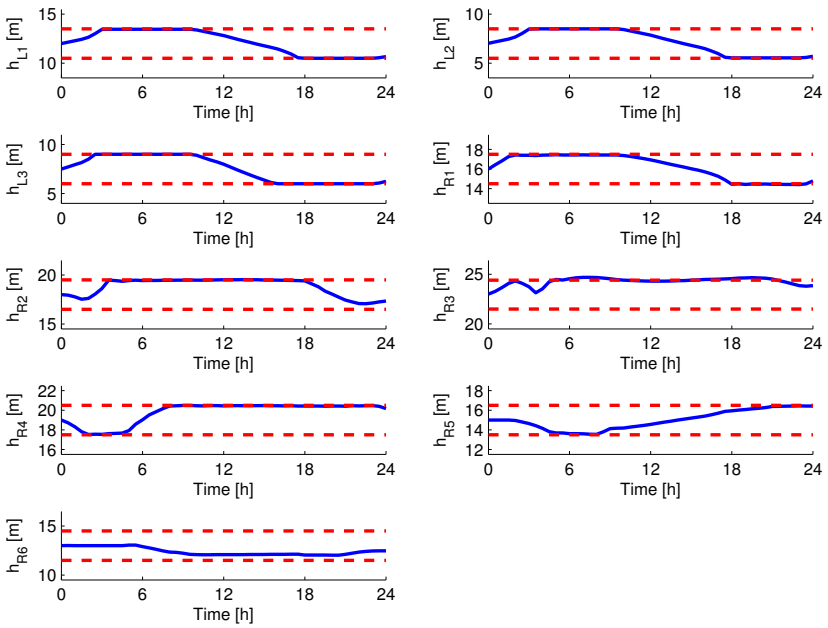
- The initial state has to be transmitted once per sampling period (249 floats).
- For each iteration (2 in the simulation), each of the 480 optimization variables floats has to be broadcasted, i.e., 960 floats are transmitted at each time step.
- For each iteration, each of the 864 Lagrange multipliers that correspond to the evolution constrained output variables have to be transmitted, i.e., 1728 floats are transmitted at each time step.

In total, 2937 floats are transmitted through the network at each time step. This communication burden corresponds to the following set up: 2 iterations per time step, $N_{\mathrm{c}} = 48$, and $N_{\mathrm{p}} = 48$.
Finally, the SAIDS and SAII indicators for this scheme were respectively 8951.34 and 2731.54.

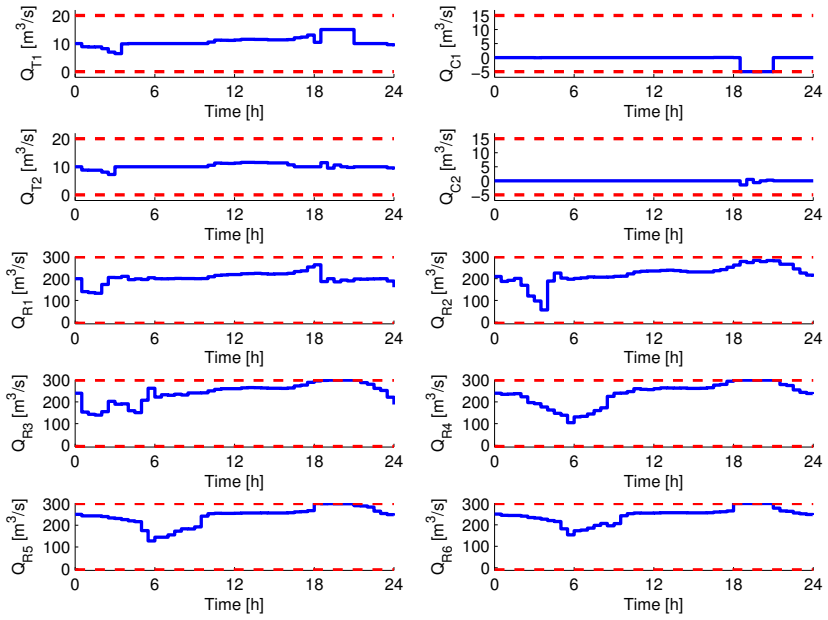
4.5 GT-DMPC

The IATE, ETE and ETE2 obtained by this scheme were, respectively, 116.39, 7998 and 5184. The evolution of the water levels and the inputs is shown in Figure 9.
Regarding the communication burden of this scheme, the subsystems have to transmit their disagreement points, states, and inputs. In total, 500 floats are transmitted at each time step due to the fact that:

- Subsystems 1 and 2 have a 64 decision variables corresponding to the inputs. In addition, the state of subsystems 1 and 2 has respectively 2 and 1 components.

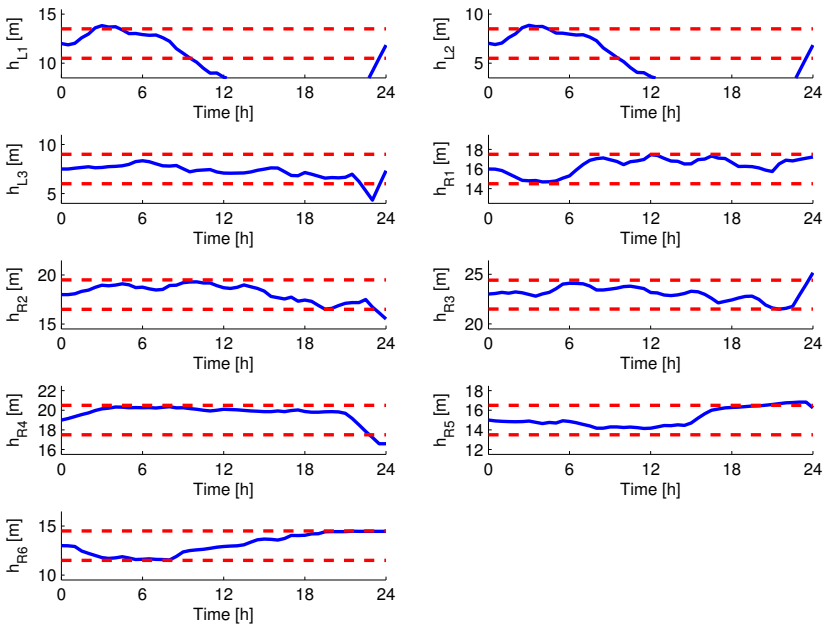


(a)

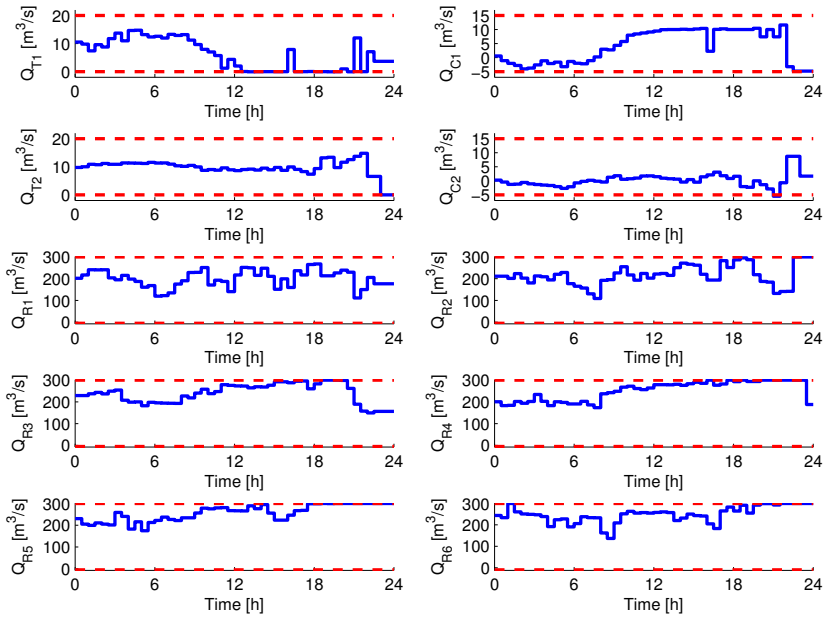


(b)

Figure 8: a) Water levels in different reaches and lakes together with the constraints along 24 hours corresponding to the S-DMPC scheme. b) Control plan of reaches and lakes together with constraints for 24 hours corresponding to the S-DMPC scheme.



(a)



(b)

Figure 9: a) Water levels in different reaches and lakes together with the constraints along 24 hours corresponding to the GT-DMPC scheme. b) Control plan of reaches and lakes together with constraints for 24 hours corresponding to the GT-DMPC scheme.

- Subsystems 3 to 8 have 40 state variables and a decision vector with 32 components.
- The disagreement point only has 1 component for each subsystem.

Finally, we must point out that there were several constraints violated in the simulations. The most important violations occur with the water levels in lake L_1 and L_2, which are violated during several hours. The integral of the constraint violation is 60.48.
Finally, the SAIDS and SAII indicators for this scheme were respectively 11267.17 and 5517.10.

Scheme	IATE [MW·h]	ETE [€]	ETE2 [€]	Comm. Cost. [#floats/sample]	Violations [m·h]	SAIDS $[m^3/s]$	SAII $[m^3/s]$
DMS	0.06	4	3	6027168	0.02	12637.91	10489.28
DMPC-BAN	89.09	5808	3388	1500	0.13	8152.11	2915.82
DAPG	3.86	250	229	1218000	0.62	7434.22	1034.83
S-DMPC	36.72	2419	1788	2937	2.10	8951.34	2731.54
GT-DMPC	116.39	7998	5184	500	60.48	11267.17	5517.10

Table 2: Comparative results

4.6 Comparative assessment

In tables 1 and 2, the main properties of the schemes considered in this work and performance indicators regarding their application in the HPV have been shown. The first, and more meaningful property that has to be discussed, is the way in which each scheme copes with the HPV control problem. In particular, the amount of centralized information that each scheme exploits is directly related with the performance it obtains. For example, as it can be seen in Table 1, DMS, DAPG and S-DMPC follow a top-down approach, i.e., the overall control problem is considered and partitioned into smaller control problems that are solved by local controllers. The concrete way in which the problem is partitioned depends on the particular scheme. A clear advantage of this approach is that it allows to provide guarantees regarding the optimality of the solution attained. More specifically, all these methods provide the same solution of the corresponding centralized control problem if they are allowed to iterate until they converge. Not surprisingly, Table 2 shows that these three schemes obtain the best results regarding the tracking performance indicators considered. However, it must be noted that DMS obtains a clear victory due to its non-linear nature and the massive amount of communication it uses.

The control schemes with the lowest performance in the HPV benchmark are DMPC-BAN and GT-DMPC. Nevertheless, given that these schemes follow a bottom-up approach, they have to be analyzed separatedly. In this case, the approach followed starts with a set of local controllers whose control tasks are coupled. Hence, the goal of this type of schemes is to promote cooperation between the controllers so that the overall performance is increased, but each local controller is still a separate entity and restrictions in the cooperation may apply for this reason. For example, in DMPC-BAN each agent proposes solutions based only on local information that may or may not be accepted by the neighbors. Likewise, GT-DMPC is based on a bargaining procedure used in game theory where each agent has a disagreement point, i.e., a maximum acceptable cost to collaborate with other agents. Why to consider then this approach in a benchmark like HPV? Despite that the HPV benchmark may seem closer to a monolithical control problem that can be partitioned into smaller control problems – and hence more suitable for a top-down approach –, the bottom-up approach is the way to go whenever there are several separated entities involved that are willing to cooperate only partially. In particular, constraints in the amount or the type of information shared, or the fact that an agent is not willing to sacrifice its local objective in order to increase the overall performance, are realistic restrictions that may arise in practice.

An additional topic related with DMPC-BAN and GT-DMPC is that these are the only schemes considered that have a formal proof regarding stability (see Table 1). The fact that stability has not been formally addressed for a scheme does not make it unstable. In fact, there are many schemes in the literature that do not count with this type of formal demonstration (see [23] for a classification of schemes regarding this issue). In our experiments, all schemes showed stable closed-loop behaviors.

Another interesting issue is the amount of communication burden used by each scheme. In Table 2, a certain correlation can be seen between the performance of each scheme and the amount of communication used. In general, communication is the mean to attain the necessary coordination in the control tasks of each local controller. For example, iterative schemes normally improve their performance as the local controllers iterate. Moreover, note that the worst results in the HPV benchmark are obtained by the only non-iterative scheme considered, which highlights the importance of communication.

The type of control actions generated by the different schemes deserves also attention. In certain applications, it is not admissible to have abrupt changes in the inputs due to several reasons: it may compromise the lifetime of the actuators, there may be physical constraints, etc. For these reasons, the indi-

cators SAIDS and SAII have been proposed. However, we must remark that none of the controllers was designed to optimize these performance indicators. Hence, they are introduced to have an initial assessment of the type of input signals generated by each scheme. As it can be seen in Table 2, DMS is the most demanding scheme regarding SAIDS and SAII. In the case of SAIDS, only GT-DMPC obtains a relatively similar result. The rest of schemes are at certain distance, being DAPG the scheme that works closer to the steady state point. Regarding SAII, DMS is by far the scheme with the most abrupt changes in the control signals. Despite of its 50% lower SAII, GT-DMPC obtains again the second worst result. DMPC-BAN and S-DMPC show a similar performance, with an additional reduction 40% with respect to GT-DMPC. The best result is obtained by DAPG, whose SAII is a 10% of that of DMS and which clearly outperforms the rest of controllers from this point of view.

Finally, once the performance indicators have been discussed in the context of DMPC, it is necessary to discuss the particular advantages and disadvantages of the various approaches in relation to their implementation in hydro-power plants. As it is stated in [13], the control of the real HPV is based on PID controllers whose setpoints are calculated by off-line optimization. In general, it is reasonable to expect an improvement on the performance of the overall system if the optimization is performed on-line using the most recent information and in a coordinated fashion, i.e., there is an incentive to use this type of schemes in this context. To this end, it is necessary to provide the local controllers with a communication infrastructure that allows the exchange of information. The scheme with the highest communication demand (DMS) requires each controller to exchange several megabytes during each time step, which is feasible even in case that wireless radio communication is used for this purpose. The choice of the scheme is closely related to the type of approach that is needed: if the goal is to maximize the performance, a top-down scheme that computes a centralized optimization problem in a distributed fashion is recommended; however, if each agent must preserve a certain independence, then a bottom-up scheme may be preferred. Likewise, the performance of the controller with respect to indicators such as the violations, SAIDS, and SAII is also important in this choice,although it must be noticed that a proper tuning or a modification of the optimization problem may help to improve the results presented in Table 2 with respect to these indicators.

5 Conclusions

We have studied the performance of different DMPC schemes on a hydropower benchmark, a type of system that can benefit from the application of this control approach. The comparison, however, was not straight forward. In the first place, it has been necessary to adapt the different MPC algorithms to the particularities of the HPV benchmark. As it has been shown, the changes needed to adapt the schemes were important in some cases, which highlights the fact that implementation problems arise no matter how general a formulation seems. For this reason it is important that control engineers choose carefully when looking for a DMPC algorithm: there are many schemes available but most of them have been designed with a certain framework in mind. In this sense, it is noteworthy that we have tested five schemes with important differences between them. In particular, the schemes considered range from linear to nonlinear, iterative to non-iterative, or hierarchical to fully-distributed, which provides an idea about the relationship between the control performance and the nature of the controllers. Not surprisingly, the best results were obtained by the schemes that are closer to a mere distribution of the computations needed to solve the equivalent centralized control problem. On the other hand, the least optimal results correspond to controllers tailored to cope with distributed problems in which the agents have limited information about the rest of the subsystems and the optimization is based on local objectives. However, our results also show that performance comes at the cost of a higher communication burden, which may be a problem in applications where the transmission of several megabytes per sampling time is not possible.

In any case, the current state of the art of DMPC schemes offers a wide range of techniques able to provide a reasonable overall performance even when there are constraints in the information exchange between the controllers. Future work should focus on the benchmarking of distributed control schemes on larger systems with a possibly dynamic composition, i.e. systems composed of a variable number of subsystems, as this is the nature of many other real distributed control problems such as that of controlling a smart grid.

References

[1] T. Ackermann, D. Schwanenberg, M. Natschke, and J. Köngeter. Control strategy for river-power-plants based on optimization. In J.C. Refs-

gaard and E. A. Karalis, editors, *Operational Water Management*, pages 285–288. Taylor & Francis, 1997.

[2] I. Alvarado, D. Limon, D. Muñoz de la Peña, J. M. Maestre, F. Valencia, H. Scheu, R. R. Negenborn, M. A. Ridao, B. De Schutter, J. Espinosa, and W. Marquardt. A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark. *Journal of Process Control*, 21(5):800–815, 2011.

[3] Raul Banos, Francisco Manzano-Agugliaro, FG Montoya, Consolacion Gil, Alfredo Alcayde, and Julio Gómez. Optimization methods applied to renewable and sustainable energy: A review. *Renewable and Sustainable Energy Reviews*, 15(4):1753–1766, 2011.

[4] Mario TL Barros, Frank TC Tsai, Shu-li Yang, Joao EG Lopes, and William WG Yeh. Optimization of large-scale hydropower system operations. *Journal of Water Resources Planning and Management*, 129(3):178–188, 2003.

[5] H. G. Bock and K. J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings of the 9th IFAC World Congress*, pages 243–247, Budapest, 1984. Pergamon Press.

[6] E. F. Camacho and C. Bordons. *Model Predictive Control in the Process Industry. Second Edition.* Springer-Verlag, London, England, 2004.

[7] V. T. Chow. *Open-Channel Hydraulics.* McGraw-Hill Book Co. Inc, 1959.

[8] J. M. Compas, P. Decarreau, G. Lanquetin, J. L. Estival, N. Fulget, R. Martin, and J. Richalet. Industrial applications of predictive functional control to rolling mill, fast robot, river dam. In *Proceedings of the Third IEEE Conference on Control Applications*, volume 3, pages 1643 –1655, Aug 1994.

[9] A. Şahin and M. Morari. Decentralized model predictive control for a cascade of river power plants. In R. R. Negenborn, Z. Lukszo, and H. Hellendoorn, editors, *Intelligent Infrastructures*, volume 42 of *Intelligent Systems, Control and Automation: Science and Engineering*, pages 463–485. Springer Netherlands, 2010.

[10] G. B. Dantzig and G. Infanger. Intelligent control and optimization under uncertainty with application to hydro-power. *European Journal of Operational Research*, 97(2):396–407, 1997.

[11] M. D. Doan, T. Keviczky, and B. De Schutter. Application of distributed and hierarchical model predictive control in hydro power valleys. In *IFAC Conference on Nonlinear Model Predictive Control*, Noordwijkerhout, Netherlands, August 2012.

[12] F. Valencia, J. Espinosa, B. De Schutter, and K. Staňková. Feasible-cooperation distributed model predictive control scheme based on game theory. In *18th IFAC World Congress*, 2010.

[13] D. Faille. Deliverable 7.2.1 of the HD-MPC project: Control specification for hydro power valleys. Technical report, Électricité de France, 2009.

[14] Filiberto Fele, José M Maestre, S Mehdy Hashemy, David Muñoz de la Peña, and Eduardo F Camacho. Coalitional model predictive control of an irrigation canal. *Journal of Process Control*, 24(4):314–325, 2014.

[15] P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer. Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica*, 49:829–833, 2013.

[16] R. M. Hermans, A. Jokic, M. Lazar, A. Alessio, P. P.J. van den Bosch, I. A. Hiskens, and A. Bemporad. Assessment of non-centralised model predictive control techniques for electrical power networks. *International Journal of Control*, 85(8):1162–1177, 2012.

[17] Liu Hongling, Jiang Chuanwen, and Zhang Yan. A review on risk-constrained hydropower scheduling in deregulated power market. *Renewable and Sustainable Energy Reviews*, 12(5):1465–1475, 2008.

[18] M. Kano and M. Ogawa. The state of the art in chemical process control in Japan: good practice and questionnaire survey. *Journal of Process Control*, 20(9):969–982, Oct 2010.

[19] A. Kozma, C. Savorgnan, and M. Diehl. Distributed multiple shooting for large scale nonlinear systems. In J. M. Maestre and R. R. Negenborn, editors, *Distributed Model Predictive Control Made Easy*, volume 69. Springer, 2014.

[20] H. Linke. A model-predictive controller for optimal hydro-power utilization of river reservoirs. In *IEEE International Conference on Control Applications (CCA)*, pages 1868 –1873, sept. 2010.

[21] J. M. Maestre, M. D. Doan, D. Muñoz de la Peña, P. J. van Overloop, T. Keviczky, M. Ridao, and B. De Schutter. Benchmarking the operation of a hydro power network through the application of agent-based model predictive controllers. In *Proceedings of the 10th International Conference on Hydroinformatics*, Jul. 2012.

[22] J. M. Maestre, D. Muñoz de la Peña, E. F. Camacho, and T. Alamo. Distributed model predictive control based on agent negotiation. *Journal of Process Control*, 21(5):685–697, 2011.

[23] J. M. Maestre and R. R. Negenborn, editors. *Distributed Model Predictive Control Made Easy*, volume 69 of *Intelligent Systems, Control and Automation: Science and Engineering*. Springer, 2014.

[24] J. F. Nash. Two-person cooperative games. *Econometrica*, 21(1):128–140, 1953.

[25] R Negenborn and J Maestre. Distributed model predictive control: An overview and roadmap of future research opportunities. *Control Systems Magazine, IEEE*, 34(4):87–97, 2014.

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

[27] M. J. D. Powell. Algorithms for nonlinear constraints that use Lagrangian functions. *Mathematical Programming*, 14(3):224–248, 1978.

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

[29] G. Ramond, D. Dumur, A. Libaux, and P. Boucher. Direct adaptive predictive control of an hydro-electric plant. In *Proceedings of the 2001 IEEE International Conference on Control Applications*, pages 606–611, 2001.

[30] C. Savorgnan and M. Diehl. Control benchmark of a hydro power plant. Technical report, K. U. Leuven, 2011.

[31] C. Savorgnan, C. Romani, A. Kozma, and M. Diehl. Multiple shooting for distributed systems with applications in hydro electricity production. *Journal of Process Control*, 21(5):738–745, 2011.

[32] R. Scattolini. Architectures for distributed and hierarchical model predictive control - a review. *Journal of Process Control*, 19:723–731, 2009.

[33] H. Scheu and W. Marquardt. Sensitivity-based coordination in distributed model predictive control. *Journal of Process Control*, 21(5):715 – 728, 2011.

[34] C. Setz, A. Heinrich, P. Rostalski, G. Papafotiou, and M. Morari. Application of model predictive control to a cascade of river power plants. In *Proceedings of the 17th IFAC World Congress*, pages 11978–11983, 2008.

[35] Q. Tran Dinh, C. Savorgnan, and M. Diehl. Real-time sequential convex programming for nonlinear model predictive control and application to a hydro-power plant. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 5905–5910, Dec. 2011.

[36] F. Valencia. *Game Theory Based Distributed Model Predictive Control: An Approach to Large-Scale Systems Control*. PhD thesis, Facultad de Minas, Universidad Nacional de Colombia, Medellín, 2012 (Draft).

[37] William W-G Yeh. Reservoir management and operations models: A state-of-the-art review. *Water resources research*, 21(12):1797–1818, 1985.

[38] A. Zafra-Cabeza, J. M. Maestre, M. A. Ridao, E. F. Camacho, and L. Sánchez. A hierarchical distributed model predictive control approach in irrigation canals: a risk mitigation perspective. *Journal of Process Control*, 21(5):787–799, 2011.

[39] J. Zarate Florez, J. J. Martinez, G. Besancon, and D. Faille. Explicit coordination for MPC-based distributed control with application to hydro-power valleys. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 830–835, Dec. 2011.