SEVENTH FRAMEWORK PROGRAMME
THEME – ICT
[Information and Communication Technologies]

Contract Number:	223854
Project Title:	Hierarchical and Distributed Model Predictive Control of Large-Scale Systems
Project Acronym:	HD-MPC

HD–MPC
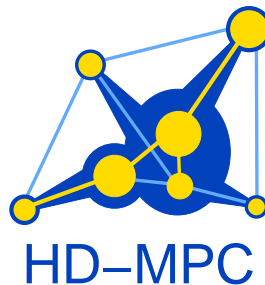
Deliverable Number:	D7.2.3 Part II
Deliverable Type:	Report
Contractual Date of Delivery:	November 1, 2011
Actual Date of Delivery:	**November 1, 2011** (original version)
	November 10, 2011 (extended[1] version)
Title of Deliverable:	**Report that presents the closed-loop validation results for the hydro-power valley, including stability and constraints issues, as well as the HD-MPC demonstration of results – Part II**
Dissemination level:	Public
Workpackage contributing to the Deliverable:	WP7
WP Leader:	Damien Faille
Partners:	TUD, EDF, KUL, RWTH, USE, UNC, SUPELEC, INOCSA
Author(s):	C. Savorgnan, A. Kozma, M.D. Doan, F. Valencia, J.M. Maestre, and M.A. Ridao

---

[1]The main difference with the original version is that decentralized MPC is now also included in the comparison.

Table of contents

Project co-ordinator

Name:	Bart De Schutter
Address:	Delft Center for Systems and Control
	Delft University of Technology
	Mekelweg 2, 2628 Delft, The Netherlands
Phone Number:	+31-15-2785113
Fax Number:	+31-15-2786679
E-mail:	b.deschutter@tudelft.nl
Project web site:	`http://www.ict-hd-mpc.eu`

Executive Summary

This deliverable consists of two parts. Part I describes the work on the hydro-power valley at EDF, while Part II presents the HD-MPC demonstration of results using the public hydro-power valley benchmark.

This part of the deliverable (i.e., Part II) presents the results on the Hydro-Power Valley (HPV) benchmark. The system is a hydro-power plant composed by several subsystems connected together. It is composed by 3 lakes and a river which is divided in 6 reaches which terminate with dams equipped with turbines for power production. The lakes and the river reaches are connected in three different ways: by a duct, ducts equipped with a turbine, and ducts equipped with a pump and a turbine. The river is fed by the an upstream inflows and tributary flows.
Six hierarchical/distributed/decentralized schemes have been tested on the HPV benchmark:

- Distributed multiple shooting

- Fast gradient-based DMPC

- Hierarchical infinite horizon MPC

- Game theory based MPC

- DMPC based on agent negotiation

- Decentralized MPC

A power tracking scenario has been used to test the algorithms: the power output of the system should follow a given reference while keeping the water levels in the lakes and at the dams as constant as possible.
Economic indexes have been defined to compare the different approaches. Also the performance with constraints and communications requirements of the distributed approaches have been considered.
The best results are obtained with the Distributed Multiple Shooting approach, with a nearly perfect tracking and a negligible economic cost. Good results are also obtained with the fast gradient-based DMPC approach and the Hierarchical infinite horizon MPC approach.

Chapter 1

Synopsis

This report is organized as follows. The second chapter introduces the hydro-power valley model. The system is a hydro power plant composed by several subsystems connected together. It is composed by 3 lakes and a river which is divided in 6 reaches which terminate with dams equipped with turbines for power production. The lakes and the river reaches are connected in three different ways: by a duct, ducts equipped with a turbine and ducts equipped with a pump and a turbine. The river is fed by the an upstream inflows and tributary flows. The models of the different component of the system and the proposed subsystem decomposition is presented in the chapter. The following test scenario is considered: the power output of the system should follow a given reference while keeping the water levels in the lakes and at the dams as constant as possible.

The third chapter discusses a class of methods applicable for optimal control of large-scale systems. The proposed approach [19] employs a combination of direct multiple shooting and domain decomposition and is called Distributed Multiple Shooting. This approach presents the best performance results in the application to the hydro-power valley benchmark.

Chapter 4 presents the application of a distributed MPC method based on a distributed accelerated proximal gradient method for solving the dual optimization problem. We first present a framework of networked optimization, in which the cost function is a composite of a strongly convex quadratic cost and a convex non-smooth 1-norm element. We use a dual decomposition approach, the dual problem is solved using the distributed algorithm proposed recently[7].
The new algorithm is distributed in the sense that each subsystem only needs to communicate with its direct neighbors, and there is no need for a master controller. The distributed solution converges to the centralized solution with a fast convergence rate, thus enables real-time implementation in large-scale systems.
In order to apply the new algorithm to the hydro power valley benchmark, we also describe the modeling technique to obtain a suitable linear model, a decentralized model reduction technique that helps to reduce the computational cost. The simulation results show that the new algorithm is capable of solving the power reference tracking problem of the hydro power plant, while respecting the operational constraints. Interestingly, our distributed algorithm requires much shorter computational time than a centralized QP approach, even when we compare the total time of computations.

The fifth Chapter presents a hierarchical control structure designed in order to maintain certain variables of a hydro power valley into a zone while the whole power plant follows a certain reference. The controllers use here are Infinite Horizon Model Predictive Control (IHMPC), these controllers assures stability if the linear system is stable and do not have integrating states. The hierarchical control structure is composed by IHMPC with zone control (coordinator) as controller that is applied

over the entire plant, there are eight IHMPC controllers, one for each subsystem. The coordinator calculates hierarchical input variables and output references that are always into the designed zone, this information is taken by the controller of each subsystem to generate the input variables that are going to be applied, in order to achieve the reference that the coordinator generates. A constraint is added to the IHMPC with zone control formulation, it is used to calculate the power of the system as a function of output variables, and to assure that the variable follows a reference.

The following chapter proposes the use of the game theory to formulate a distributed model predictive control scheme to control the hydro-power valley. In this control scheme the whole system is decomposed into several subsystems able to communicate between them. For each subsystem a local MPC controller is formulated, and all local optimization problems are computed at the same time without using an iterative procedure to determine what control actions should be applied to each subsystem. Since the decisions of the subsystems depend on the decisions of the others, game theory is used as a mathematical framework to formulate and to analyze the distributed MPC problem and to derive a reliable solution.

The last distributed approach is presented in the seventh chapter. The distributed MPC scheme is based on agent negotiation presented in [9]. This control scheme is tailored for distributed linear systems composed of subsystems coupled in the inputs. We assume that the subsystems are controlled by a set of independent agents that are able to communicate and that each agent has access only to the model and the state of one of the subsystems. These assumptions imply that before the agents take a cooperative decision, they must negotiate. At each time sample, following a protocol, agents make proposals to improve an initial feasible solution on behalf of their local cost function, state and model. These proposals are accepted if the global cost improves the cost corresponding to the current solution.

Four quantitative indexes has been used to analyze the performance of the approaches:

- Mean absolute tracking error (MAE) in *MW*

- Mean quadratic tracking error (MQE) MW^2

- Power reference tracking index 1 (J_1) in Euros: two indexes will be used to assess the economic performance of the proposed scheme. In first place, an expression inspired in the index proposed for the power reference tracking scenario is used:

$$\int_0^{86400} c(t) \left| p_r(t) - \sum_{i=1}^{8} p_i(x_i(t), u_i(t)) \right| dt$$

  where $c(t)$ is the cost of the electricity at time t. Note that this expression only focuses on the economical part of the equation (2.24).

- Power reference tracking index 2 (J_2) in Euros: another option that will be used to test the economic performance of the scheme is given by the following expression

$$\int_0^{86400} c(t) max \left(p_r(t) - \sum_{i=1}^{8} p_i(x_i(t), u_i(t)), 0 \right) dt$$
$$+ 0.5 \int_0^{86400} c(t) max \left(\sum_{i=1}^{8} p_i(x_i(t), u_i(t)) - p_r(t), 0 \right) dt$$

In Table 1.1 the indexes for each one of the approaches are shown. Notice that there are very important differences among approaches from an economic point of view.

Control performance	J_1	J_2	MAE	MQE
Distributed multiple shooting	0.20	-	6.31×10^{-5}	6.99×10^{-9}
Fast gradient-based DMPC	2568	2440	1.65	4.85
Hierarchical infinite horizon MPC	2397	1416	1.52	3.21
Game theory-based DMPC	4722	4066	3.19	19.06
DMPC based on agent negotiation	5750	4194	3.72	19.88
Decentralized MPC	3.30×10^5	2.78×10^5	208.70	54.08×10^3

Table 1.1: Table of the quantitative benchmark indices of each tested controller

The best results are obtained with the Distributed Multiple Shooting approach, with a nearly perfect tracking and a negligible economic cost. Good results are obtained also with Fast gradient-based DMPC and with Hierarchical infinite horizon MPC.

Chapter 2

Hydro-Power Valley Description

2.1   System overview

The system we consider is a hydro power plant composed by several subsystems connected together. Figure 2.1 gives an overview of the system which is composed by 3 lakes (L_1, L_2 and L_3) and a river which is divided in 6 reaches (R_1, R_2, R_3, R_4, R_5 and R_6) which 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 $q_{tributary}$.

In the following sections we shall provide a model for all the subsystems. To simplify the system modeling we make the following assumptions:

- the ducts are connected at the bottom of the lakes (or to the bottom of the river bed);

- the cross section of the reaches and of the lakes is rectangular;

- the width of the reaches varies linearly along the reaches;

- the river bed slope is constant along every reach.

2.2   System model

2.2.1   Reach model

The model of the reaches is based on the one-dimensional Saint Venant partial differential equation:

$$\begin{cases} \dfrac{\partial q(t,z)}{\partial z} + \dfrac{\partial s(t,z)}{\partial t} = 0 \\ \dfrac{1}{g}\dfrac{\partial}{\partial t}\left(\dfrac{q(t,z)}{s(t,z)}\right) + \dfrac{1}{2g}\dfrac{\partial}{\partial z}\left(\dfrac{q^2(t,z)}{s^2(t,z)}\right) + \dfrac{\partial h(t,z)}{\partial z} + I_{\mathrm{f}}(t,z) - I_0(z) = 0 \end{cases} \tag{2.1}$$

The two equations in (2.1) express the mass and momentum balance. The variables represent the following quantities:

- z is the spatial variable which increases along the flow main direction;

- $q(t,z)$ is the river flow (or discharge) at time t and space coordinate z;
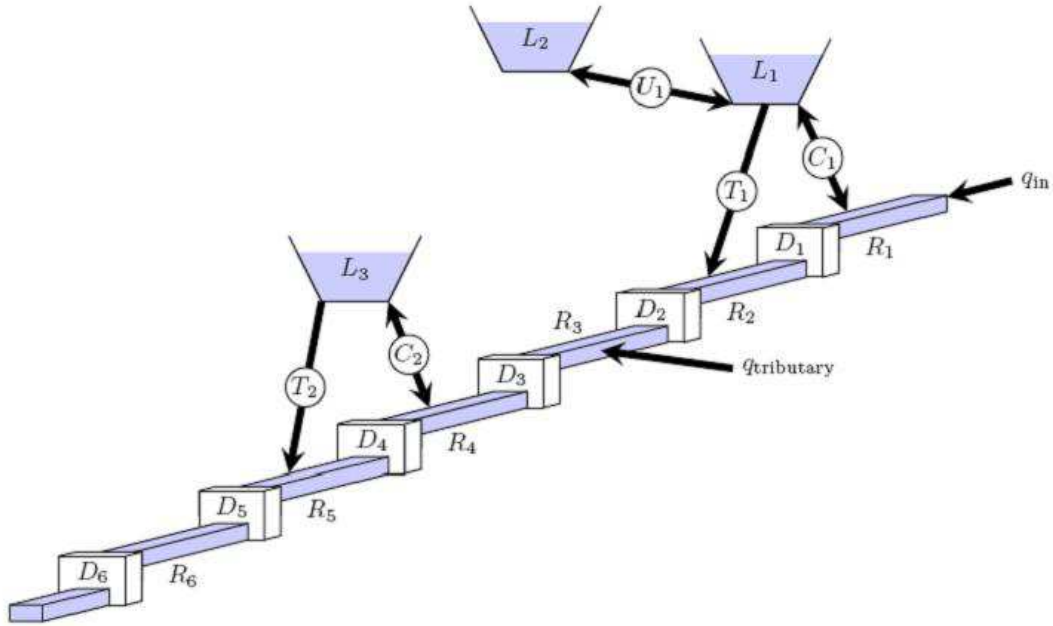
Figure 2.1: Overview of the Hydro power valley.

- $s(t,z)$ is the wetted surface;

- $h(t,z)$ is the water level w.r.t. the river bed;

- g is the gravitational acceleration;

- $I_f(t,z)$ is the friction slope;

- $I_0(z)$ is the river bed slope.

Assuming the cross section of the river is rectangular we can write the following equations:

$$s(t,z) = w(z)h(t,z) \tag{2.2}$$

and

$$I_f(t,z) = \frac{q(t,z)^2 \left(w(z) + 2h(t,z)\right)^{4/3}}{k_{\text{str}}^2 \left(w(z)h(t,z)\right)^{10/3}} \tag{2.3}$$

where $w(z)$ is the river width and k_{str} is the Gauckler-Manning-Strickler coefficient[1].

To take into account lateral inflows, the first equation in (2.1) which expresses the mass balance can be modified as follows

$$\frac{\partial q(t,z)}{\partial z} + \frac{\partial s(t,z)}{\partial t} = q_\text{l}(z) \tag{2.4}$$

where $q_\text{l}(z)$ is the lateral inflow per space unit.

---

[1]The Gauckler-Manning-Strickler coefficient changes accordingly to the kind of river bed surface. In the model we developed k_{str} is constant along the river.

2.2.2 Discretized model

The partial differential equation (2.1) can be converted into an ordinary differential equation with the method of lines. Divide the reach into N cells of length dz and denote by $q_i(t)$ the value of the discharge in the middle of the cell i and by $h_i(t)$ the value of the water level at the beginning of cell i. h_{N+1} represents the water level at the end of the reach.

Denoting by $q_{\text{in}}(t)$ and $q_{\text{out}}(t)$ the water inflow at the beginning of the reach and the water outflow at the end of the reach, we obtain the following set of ordinary differential equations (time dependencies are omitted)

$$
\begin{cases}
\dfrac{\partial h_1}{\partial t} = -\dfrac{1}{w_1}\dfrac{q_1 - q_{\text{in}} - q_{l_1}}{dz/2} \\[2mm]
\dfrac{\partial q_1}{\partial t} = \dfrac{q_1}{w_1 h_1}\dfrac{q_{l_1}}{dz/2} - \dfrac{2q_1}{w_1 h_1}\dfrac{q_1 - q_{\text{in}}}{dz/2} + \left[\dfrac{1}{w_1}\left(\dfrac{q_1}{h_1}\right)^2 - gw_1 h_1\right]\dfrac{h_2 - h_1}{dz} + \\[2mm]
\qquad\quad +gw_1 h_1 I_0 - gw_1 h_1\left[\dfrac{q_1^2\left(w_1 + 2h_1\right)^{4/3}}{k_{\text{str}}^2\left(w_1 h_1\right)^{10/3}}\right]
\end{cases}
$$

$$
\begin{cases}
\dfrac{\partial h_i}{\partial t} = -\dfrac{1}{w_i}\dfrac{q_i - q_{i-1} - q_{l_i}}{dz} \\[2mm]
\dfrac{\partial q_i}{\partial t} = \dfrac{q_i}{w_i h_i}\dfrac{q_{l_i}}{dz} - \dfrac{2q_i}{w_i h_i}\dfrac{q_i - q_{i-1}}{dz} + \left[\dfrac{1}{w_i}\left(\dfrac{q_i}{h_i}\right)^2 - gw_i h_i\right]\dfrac{h_{i+1} - h_i}{dz} + \qquad i = 2,\dots,N \\[2mm]
\qquad\quad +gw_i h_i I_0 - gw_i h_i\left[\dfrac{q_i^2\left(w_i + 2h_i\right)^{4/3}}{k_{\text{str}}^2\left(w_i h_i\right)^{10/3}}\right]
\end{cases}
$$

$$
\dfrac{\partial h_{N+1}}{\partial t} = -\dfrac{1}{w_{N+1}}\dfrac{q_{\text{out}} - q_N}{dz/2}
$$

(2.5)

where w_i represents the river width at the beginning of cell i, w_{N+1} represents the river width at the end of the reach and q_{l_i} is the total lateral inflow of cell i. The river bed slope I_0 is assumed to be constant. Since the width of the reaches changes linearly, the values of w_1 and w_{N+1} are provided in the model data while

$$
w_i = w_1 + \frac{(i-1)(w_{N+1} - w_1)}{N}. \tag{2.6}
$$

Remark 1 *Notice that distance between the beginning of the reaches and the lateral inflow points are given in the last section. They are denoted as $L_{\text{tributary}}$, L_{C_1}, L_{T_1}, L_{C_2} and L_{T_2}.*

2.2.3 Lake model

Denote by $q_{\text{in}}(t)$ and $q_{\text{out}}(t)$ the water inflow and outflow of the lake under consideration, respectively. The volume of water inside the lake varies accordingly to the following equation

$$
\frac{\partial v(t)}{\partial t} = q_{\text{in}}(t) - q_{\text{out}}(t). \tag{2.7}
$$

Since the cross section of the lake is assumed to be rectangular, (2.7) can be equivalently expressed as

$$
\frac{\partial h(t)}{\partial t} = \frac{q_{\text{in}}(t) - q_{\text{out}}(t)}{S}, \tag{2.8}
$$

where $h(t)$ is the water level and S is the lake surface area.

2.2.4 Duct model

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

$$q_{U_1}(t) = S_{U_1} \operatorname{sign}(h_{L_2}(t) - h_{L_1}(t) + h_{U_1}) \sqrt{2g|h_{L_2}(t) - h_{L_1}(t) + h_{U_1}|}, \tag{2.9}$$

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.

Denoting $x = h_{L_2}(t) - h_{L_1}(t) + h_{U_1}$, equation (2.9) can be written as $S_{U_1}\sqrt{2g}\operatorname{sign}(x)\sqrt{|x|}$. The function $\operatorname{sign}(x)\sqrt{|x|}$ is not differentiable for $x = 0$. The following approximation can be used to make the function $q_{U_1}(t)$ differentiable

$$\operatorname{sign}(x)\sqrt{|x|} \approx \frac{x}{(x^2 + \varepsilon^4)^{1/4}}.$$

Notice that for $\varepsilon = 0$ the two functions are equivalent, while keeping ε small we obtain a good approximation ($\frac{1}{\varepsilon}$ corresponds to the derivative of the approximation at $x = 0$).

2.2.5 Turbine model

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

$$p_{\mathrm{t}}(t) = k_{\mathrm{t}} q_{\mathrm{t}}(t) \Delta h_{\mathrm{t}}(t), \tag{2.10}$$

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.

2.2.6 Pump model

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

$$p_{\mathrm{p}}(t) = k_{\mathrm{p}} q_{\mathrm{p}}(t) \Delta h_{\mathrm{p}}(t), \tag{2.11}$$

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.

2.2.7 Modeling of ducts equipped with a turbine and a pump

The ducts C_1 and C_2 are equipped with a pump and a turbine and therefore we can use equations (2.10) and (2.11) to express the amount of power generated or absorbed. However, the turbines and the pumps cannot function together and this should be expressed in the optimal control problems (OCPs) formulated using the hydro power plant. Depending on the OCP formulation and the method used to solve the problem different models can be used. In the remainder of this section we illustrate some possibilities in modeling C_1 (the same model can be used for C_2). We assume that the flow can be determined by the controller.

Discontinuous model

Denote by $q_{C_1}(t)$ the flow through duct C_1. We assume that:

- $q_{C_1}(t) \geq 0$ when C_1 functions as a turbine;

- $q_{C_1}(t) < 0$ when C_1 functions as a pump.

Notice that by using this convention we do not need to express explicitly that C_1 can function as a turbine or a pump alternatively. The power produced can be expressed as

$$p_{C_1}(t) = k_{C_1}(q_{C_1}(t))q_{C_1}(t)\Delta h_{C_1}(t), \tag{2.12}$$

where $\Delta h_{C_1}(t)$ it the duct head which depends on the water level in lake L_1 and reach R_1 and

$$k_{C_1}(q_{C_1}(t)) = \begin{cases} k_{t_{C_1}} & \text{when } q_{C_1}(t) \geq 0 \\ k_{p_{C_1}} & \text{when } q_{C_1}(t) < 0 \end{cases}, \tag{2.13}$$

($k_{t_{C_1}}$ is the turbine coefficient and $k_{p_{C_1}}$ is the pump coefficient). The flow in C_1 is limited:

$$q_{C_1}(t) \in \left[-q_{C_1 p,max}, -q_{C_1 p,min} \right] \cup \left[q_{C_1 t,min}, q_{C_1 t,max} \right], \tag{2.14}$$

where the values $q_{C_1 p,max}$, $q_{C_1 p,min}$, $q_{C_1 t,min}$ and $q_{C_1 t,max}$ are positive (the subscript t stands for turbine, while p stands for pump).

Equation (2.13) and the constraint (2.14) make the model of the C_1 discontinuous and therefore not suitable for many control techniques.

Smoothed model

Equation (2.13) can be written as

$$k_{C_1}(q_{C_1}(t)) = \frac{1}{2} \left((1 + \text{sign}(q_{C_1}(t))) k_{t_{C_1}} + (1 - \text{sign}(q_{C_1}(t))) k_{p_{C_1}} \right) \tag{2.15}$$

and then made smooth using the following approximation

$$\text{sign}(x) \approx \frac{x}{(x^2 + \varepsilon^2)^{1/2}} \tag{2.16}$$

(ε^{-1} corresponds to the derivative of the approximation at $x = 0$). The constraint (2.14) can be simplified by imposing

$$q_{C_1}(t) \in \left[-q_{C_1 p,max}, q_{C_1 t,max} \right]. \tag{2.17}$$

The previous model of C_1 is still highly nonlinear and may not be suitable for linear MPC applications.

Double flow model

Another simplified model can be obtained by introducing two separate variables to express the flow in C_1

- $q_{C_1 p}(t)$: flow when C_1 is functioning as a pump;

- $q_{C_1 t}(t)$: flow when C_1 is functioning as a turbine.

Assuming these new variables are both positive we can write

$$q_{C_1}(t) = q_{C_1 t}(t) - q_{C_1 p}(t) \tag{2.18}$$

and

$$p_{C_1}(t) = (k_{t_{C_1}} q_{C_1 t}(t) - k_{p_{C_1}} q_{C_1 p}(t))\Delta h_{C_1}(t). \tag{2.19}$$

The constraint (2.14) can be rewritten in terms of $q_{C_{1p}}(t)$ and $q_{C_{1t}}(t)$

$$q_{C_{1p}}(t) \in \left[q_{C_{1p,min}}, q_{C_{1p,max}} \right] \tag{2.20}$$

$$q_{C_{1t}}(t) \in \left[q_{C_{1t,min}}, q_{C_{1t,max}} \right]. \tag{2.21}$$

Relaxed model

When the power production is maximized (as in the profit maximization scenario proposed below), the following relaxation can be used

$$\begin{aligned} p_{C_1}(t) &\leq k_{t_{C_1}} q_{C_{1t}}(t) \Delta h_{C_1}(t) \\ p_{C_1}(t) &\leq k_{p_{C_1}} q_{C_{1p}}(t) \Delta h_{C_1}(t) \end{aligned} \tag{2.22}$$

and

$$q_{C_1}(t) \in \left[-q_{C_{1p,max}}, q_{C_{1t,max}} \right] \tag{2.23}$$

This relaxation is meaningful for power maximization since the value of $k_{p_{C_1}} < k_{t_{C_1}}$.

Remark 2 *Using any of the models in Sections 2.2.7, 2.2.7 or 2.2.7 introduces some approximations. In particular, the control inputs corresponding to the solution of an OCP using these simplified models may not respect constraint (2.14). The control values achieved should be therefore modified.*

2.3 Subsystem partition

The system is partitioned into 8 subsystems.

2.3.1 Subsystem 1 ($L_1 + L_2 + U_1 + T_1 + C_1$)

Subsystem 1 is composed by lakes L_1 and L_2 and ducts U_1, T_1 and C_1. Duct C_1 can function as a pump or a turbine.

Define the following quantities:

- $h_{L_1}(t)$ is the water level in lake L_1;

- $h_{L_2}(t)$ is the water level in lake L_2;

- $q_{L_1}(t)$ is the water inflow for L_1 which takes into account rain, small tributaries, etc.;

- $q_{L_2}(t)$ is the water inflow for L_2 which takes into account rain, small tributaries, etc.;

- $q_{T_1}(t)$ is the water discharge going to turbine T_1 (control variable);

- $q_{C_1}(t)$ is the water discharge going through the duct C_1 (control variable);

- h_{T_1} is the height difference of duct T_1;

- h_{C_1} is the height difference of duct C_1;

- $h_{R_2,T_1}(t)$ is the water level in R_2 at the outflow point of duct T_1;

- $h_{R_1,C_1}(t)$ is the water level in R_1 at the outflow point of duct C_1;

- $k_{t_{T_1}}$ is the turbine coefficient of T_1;

- $k_{t_{C_1}}$ is the turbine coefficient of C_1;

- $k_{p_{C_1}}$ is the pump coefficient of C_1

- $p_{S_1}(t)$ is the power produced by subsystem 1.

The equations governing the subsystem behavior can be derived using the equations illustrated in the previous section and setting

- lake L_1

$$q_{\text{in}}(t) = q_{L_1}(t) + q_{U_1}(t)$$
$$q_{\text{out}}(t) = q_{T_1}(t) + q_{C_1}(t)$$

- lake L_2

$$q_{\text{in}}(t) = q_{L_2}(t)$$
$$q_{\text{out}}(t) = q_{U_1}(t)$$

- turbine T_1

$$\Delta h_{\text{t}}(t) = h_{T_1} + h_{L_1}(t) - h_{R_2,T_1}(t)$$

- combined turbine/pump C_1

$$\Delta h_{C_1}(t) = h_{C_1} + h_{L_1}(t) - h_{R_1,C_1}(t).$$

The variables of subsystem 1 are subject to the following constraints

$$h_{L_1\min} \leq h_{L_1}(t) \leq h_{L_1\max}$$
$$h_{L_2\min} \leq h_{L_2}(t) \leq h_{L_2\max}$$
$$q_{T_1\min} \leq q_{T_1}(t) \leq q_{T_1\max}$$
$$q_{C_1}(t) \in \left[-q_{C_{1\text{p,max}}}, -q_{C_{1\text{p,min}}}\right] \cup \left[q_{C_{1\text{t,min}}}, q_{C_{1\text{t,max}}}\right]$$

2.3.2 Subsystem 2 ($L_3 + T_2 + C_2$)

Subsystem 2 is composed by lake L_3 and ducts T_2 and C_2.
　　Define the following quantities:

- $h_{L_3}(t)$ is the water level in lake L_3;

- $q_{L_3}(t)$ is the water inflow for L_3 which takes into account rain, small tributaries, etc.;

- $q_{T_2}(t)$ is the water discharge going to turbine T_2 (control variable);

- $q_{C_2}(t)$ is the water discharge going through the duct C_2. $q_{C_2}(t)$ is positive when C_2 functions as a pump (control variable);

- h_{T_2} is the height difference of duct T_2;

- h_{C_2} is the height difference of duct C_2;

- $h_{R_5,T_2}(t)$ is the water level in R_5 at the outflow point of duct T_2;

- $h_{R_4,C_2}(t)$ is the water level in R_4 at the outflow point of duct C_2;

- $k_{t_{T_2}}$ is the turbine coefficient of T_2;

- $k_{t_{C_2}}$ is the turbine coefficient of C_2;

- $k_{p_{C_2}}$ is the pump coefficient of C_2;

- $p_{S_2}(t)$ is the power produced by subsystem 2.

The equations governing the subsystem behavior can be derived using equations (2.8)–(2.11) and setting

- lake L_3

$$q_{\mathrm{in}}(t) = q_{L_3}(t)$$
$$q_{\mathrm{out}}(t) = q_{T_2}(t) + q_{C_2}(t)$$

- turbine T_2

$$\Delta h_{\mathrm{t}}(t) = h_{T_2} + h_{L_3}(t) - h_{R_5,T_2}(t)$$

- combined turbine/pump C_2

$$\Delta h_{C_2}(t) = h_{C_2} + h_{L_3}(t) - h_{R_4,C_2}(t).$$

The variables of subsystem 2 are subject to the following constraints

$$h_{L_{3\mathrm{min}}} \le h_{L_3}(t) \le h_{L_{3\mathrm{max}}}$$
$$q_{T_{2\mathrm{min}}} \le q_{T_2}(t) \le q_{T_{2\mathrm{max}}}$$
$$q_{C_2}(t) \in \left[-q_{C_{2\mathrm{p,max}}}, -q_{C_{2\mathrm{p,min}}} \right] \cup \left[q_{C_{2\mathrm{t,min}}}, q_{C_{2\mathrm{t,max}}} \right]$$

2.3.3 Subsystem 3 ($R_1 + D_1$), 4 ($R_2 + D_2$), 5 ($R_3 + D_3$), 6 ($R_4 + D_4$), 7 ($R_5 + D_5$), 8 ($R_6 + D_6$)

Subsystems 3, 4, 5, 6, 7, and 8 are composed by a reach and dam. Figure 2.2 represents the structure of the dams. All the flow going through the dams is used by the turbine to produce electricity. The head of the turbines inside the dams can be expressed as difference of the water level before and after the dam. Since the water level after dam D_6 is not part of the model we consider it constant ($h_{D_{6\mathrm{out}}}$).

The constraints on the subsystem variables are

- subsystem 3

$$h_{R_{1\mathrm{min}}} \le h_{R_1}(t) \le h_{R_{1\mathrm{max}}}$$
$$q_{D_{1\mathrm{min}}} \le q_{D_1}(t) \le q_{D_{1\mathrm{max}}}$$

  where $h_{R_1}(t)$ is the water level at the end of reach R_1 and $q_{D_1}(t)$ is the dam discharge which goes to the turbine (the control variable);

- subsystem 4

$$h_{R_{2\mathrm{min}}} \le h_{R_2}(t) \le h_{R_{2\mathrm{max}}}$$
$$q_{D_{2\mathrm{min}}} \le q_{D_2}(t) \le q_{D_{2\mathrm{max}}}$$

  where $h_{R_2}(t)$ is the water level at the end of reach R_2 and $q_{D_2}(t)$ is the dam discharge which goes to the turbine (the control variable);

Figure 2.2: Dam configuration.

- subsystem 5

$$h_{R_{3\min}} \leq h_{R_3}(t) \leq h_{R_{3\max}}$$
$$q_{D_{3\min}} \leq q_{D_3}(t) \leq q_{D_{3\max}}$$

where $h_{R_3}(t)$ is the water level at the end of reach R_1 and $q_{D_3}(t)$ is the dam discharge which goes to the turbine (the control variable);
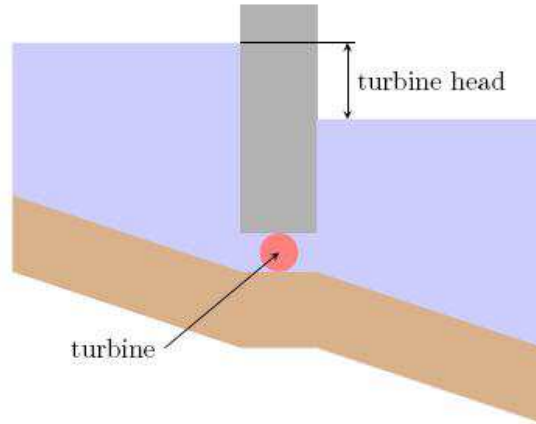
- subsystem 6

$$h_{R_{4\min}} \leq h_{R_4}(t) \leq h_{R_{4\max}}$$
$$q_{D_{4\min}} \leq q_{D_4}(t) \leq q_{D_{4\max}}$$

where $h_{R_4}(t)$ is the water level at the end of reach R_4 and $q_{D_4}(t)$ is the dam discharge which goes to the turbine (the control variable);

- subsystem 7

$$h_{R_{5\min}} \leq h_{R_5}(t) \leq h_{R_{5\max}}$$
$$q_{D_{5\min}} \leq q_{D_5}(t) \leq q_{D_{5\max}}$$

where $h_{R_5}(t)$ is the water level at the end of reach R_5 and $q_{D_5}(t)$ is the dam discharge which goes to the turbine (the control variable);

- subsystem 8

$$h_{R_{6\min}} \leq h_{R_6}(t) \leq h_{R_{6\max}}$$
$$q_{D_{6\min}} \leq q_{D_6}(t) \leq q_{D_{6\max}}$$

where $h_{R_6}(t)$ is the water level at the end of reach R_6 and $q_{D_6}(t)$ is the dam discharge which goes to the turbine (the control variable).

2.4 Control test scenario: Power reference tracking

We assume that the power reference to be followed by the entire system is known 24 hours in advance. Therefore, the prediction horizon is set to 86400 seconds. The inputs of the system can be changed every 30 minutes. The input vectors $u_i(t)$ are constant in this time intervals.

The optimal control problem to be solved reads

$$\min_{x_i,u_i} \quad \int_0^{86400} \gamma \left| p_r(t) - \sum_{i=1}^{8} p_i(x_i(t), u_i(t)) \right| dt + \sum_{i=1}^{8} \int_0^{86400} (x_i(t) - x_{ss,i})^T Q_i (x_i(t) - x_{ss,i}) dt \quad (2.24)$$

where $t_k = 1800k$, f is a function which represents the dynamics of the whole system. The function $p_r(t)$ is the given power reference (piecewise constant).

Remark 3 *Notice that when implementing this scenario the power should be expressed in MW (megawatts).*

In the control test scenario we make the assumption that all the water inflows are constant ($q_{\text{in}}(t) = q_{\text{in}}$, $q_{\text{tributary}}(t) = q_{\text{tributary}}$, $q_{\text{in}_{L_1}}(t) = q_{\text{in}_{L_1}}$, $q_{\text{in}_{L_2}}(t) = q_{\text{in}_{L_2}}$, $q_{\text{in}_{L_3}}(t) = q_{\text{in}_{L_3}}$).

To simplify the description of the two optimal control problem formulations we define

- $x_i(t)$: state vector of subsystem i;

- $u_i(t)$: input vector of subsystem i;

- C_i: set describing the constraints for subsystem i;

- $p_i(x_i(t), u_i(t))$: power produced by subsystem i;

Chapter 3

Distributed Multiple Shooting

3.1 Introduction

In this section we discuss a class of methods applicable for optimal control of large-scale systems. The proposed approach [19] employs a combination of direct multiple shooting and domain decomposition and is called Distributed Multiple Shooting. We regard the optimal control problem

$$\min_{\substack{x,u,z,\\y,e}} \int_0^T \ell(e(t))dt + \sum_{i=1}^M \int_0^T \ell^i(x^i(t),u^i(t),z^i(t))dt \tag{3.1a}$$

$$\text{s.t.}\dot{x}^i(t) = f^i(x^i(t),u^i(t),z^i(t)) \tag{3.1b}$$

$$y^i(t) = g^i(x^i(t),u^i(t),z^i(t)) \tag{3.1c}$$

$$x^i(0) = \bar{x}_0^i \tag{3.1d}$$

$$z^i(t) = \sum_{j=1}^M A_{ij}y^j(t) \tag{3.1e}$$

$$e(t) = r(t) + \sum_{i=1}^M B^i y^i(t) \tag{3.1f}$$

$$p^i(x^i(t),u^i(t)) \geq 0, \quad q(e(t)) \geq 0 \quad t \in [0,T], \tag{3.1g}$$

where $x^i(t),u^i(t)$ and $z^i(t)$ is the state variable, control input variable and coupling input signal, respectively. The signal $r(t)$ can be regarded as a reference signal. Note that the coupling between subsystems is characterized only by (3.1f).

3.2 Discretization

In order to obtain a finite nonlinear program, we have to discretize our continuous signals. In this section we detail how this may be carried out. The signals $r(t), e(t)$ and $y(t)$ are discretized by using Legendre polynomials. For example, the pth element of the $z^i(t)$ signal can be approximated by

$$z_p^i(t) = \Gamma_m(t)^T \mathbf{z}_{m,p}^i, \tag{3.2}$$

where $\Gamma_n(t)$ is the mth-order Legendre basis. Due to the orthogonality of $\Gamma_n(t)$, the discretization can be obtained by performing the integration

$$\mathbf{y}_n^i = \frac{2}{t_{n+1}-t_n} \int_{t_n}^{t_{n+1}} \Gamma_n(t)\left(y^i(t)\right)^T dt. \tag{3.3}$$

The discretization of the state profile is done inside an integrator, thus we only define the initial value x_n^i of each shooting interval and handle the integrator as a function that solves differential equations depending on the initial value, the control input signal and some coupling input coefficients.

We discretize the control input signal by introducing piecewise constant control profile, thus for system i and each shooting interval n we introduce the function $F(x_n^i, u_n^i, \mathbf{z}_n^i)$, which represents a simulator that is able to generate sensitivities as well.

After the discretization steps we obtain a finite nonlinear programming problem.

$$
\min_{\substack{u_n^i, x_n^i, \mathbf{z}_n^i, \\ \mathbf{y}_n^i, \mathbf{e}_n}} \quad \sum_{n=0}^{N-1} \left(L_n(\mathbf{e}_n) + \sum_{i=1}^{M} L_n^i(x_n^i, u_n^i, \mathbf{z}_n^i) \right)
$$
$$
\begin{aligned}
\text{s.t.} \quad & x_{n+1}^i = F_n^i(x_n^i, u_n^i, \mathbf{z}_n^i) & n = 0, \dots, N-1 \\
& \mathbf{y}_n^i = G_n^i(x_n^i, u_n^i, \mathbf{z}_n^i) & n = 0, \dots, N-1 \\
& x_0^i = \bar{x}_0^i \\
& \mathbf{z}_n^i = \Sigma_{j=1}^{M} A_{ij} \mathbf{y}_n^j \\
& \mathbf{e}_n = \mathbf{r}_n + \Sigma_{j=1}^{M} B_{ij} \mathbf{y}_n^j \\
& p^i(x_n^i, u_n^i) \geq 0, \qquad Q_n(\mathbf{e}_n) \geq 0
\end{aligned}
\tag{3.4}
$$

One may solve the result problem with a Sequential Quadratic Programming method, which calculates the linearization of the original problem and employs corrections sequentially to the original optimization variables. The essence of the proposed method is that the evaluation of $F_n^i(x_n^i, u_n^i, \mathbf{z}_n^i)$ along with $\nabla F_n^i(x_n^i, u_n^i, \mathbf{z}_n^i)$ may be divided into $M \times N$ independent tasks with own integration rules.

3.3 Solution methods

In this section we describe the solution method that can be used to solve (3.4). To simplify our discussion we regard the constrained nonlinear programming problem

$$
\begin{aligned}
\min_x \quad & f(x) \\
\text{s.t.} \quad & g(x) = 0
\end{aligned}
\qquad \text{with KKT} - \text{conditions} : \begin{bmatrix} L_x(x, \lambda) \\ g(x) \end{bmatrix} = 0,
$$

where $L_x(x, \lambda) = \nabla f(x) - \nabla g(x) \lambda$.

Sequential Quadratic Programming methods linearize the KKT system in a way and solve the resulting problem sequentially until convergence.

Once we calculate exact linearizations (i.e. exact constraint Jacobians), we have to solve the linear system

$$
\begin{bmatrix} L_x^k \\ g^k \end{bmatrix} + \begin{bmatrix} B^k & -\nabla g^k \\ (\nabla g^k)^T & 0 \end{bmatrix} \begin{bmatrix} x - x_k \\ \lambda - \lambda_k \end{bmatrix} = 0
$$

or equivalently we have to solve the quadratic program

$$
\min_p \frac{1}{2} p^T B^k p + (\nabla f^k)^T p
$$
$$
\text{s.t. } g^k + (\nabla g^k)^T p = 0,
$$

and apply the solution p^* to obtain

$$
\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} := \begin{bmatrix} x_k \\ 0 \end{bmatrix} + \alpha_k p^*
\tag{3.5}
$$

with some $\alpha_k \in (0,1]$. This approach we will regard as *full SQP method*.

By using inexact derivatives $G^k \approx (\nabla g^k)^T$ we may neglect some directions in the Jacobian (i.e. less forward derivatives in the integrators) while retaining convergence [5]. The corresponding linear system to be solved is

$$\begin{bmatrix} L_x^k \\ g^k \end{bmatrix} + \begin{bmatrix} B^k & -(G^k)^T \\ G^k & 0 \end{bmatrix} \begin{bmatrix} x - x_k \\ \lambda - \lambda_k \end{bmatrix} = 0,$$

or equivalently solve the quadratic programming problem

$$\min_p \frac{1}{2} p^T B^k p + d^T p$$
$$\text{s.t. } g^k + G^k p = 0,$$

where $d = L_x^k + (G^k)^T \lambda = \nabla f^k - \nabla g^k \lambda + (G^k)^T \lambda$. Note that $\nabla g^k \lambda$ is computable with one adjoint derivation, which gives the name to the method, *adjoint-based SQP method*.

In the context of Distributed Multiple Shooting, one can divide the variable vector $\mathbf{z}$ into $[\mathbf{z_1}, \mathbf{z_2}]$, where $\mathbf{z_1}$ represents the low-order coefficients of signal $z(t)$ having the following structure in $\nabla F_n^i(x_n^i, u_n^i, \mathbf{z}_n^i)$.

x_n^i			u_n^i			$(\mathbf{z_1})_n^i$		$(\mathbf{z_2})_n^i$	
$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	0	0
$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	0	0
$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	0	0

In this approach [18] there is a tunable parameter that determines the length of $\mathbf{z_1}$. Note that in the extreme case none of the $\mathbf{z}$ columns are calculated, in which case the quadratic programming subproblems decompose to $M \times N$ small quadratic programs. In optimization terminology this approach corresponds to Newton's method in the variables x and u, and a fix-point iteration in the variables $\mathbf{z}$. From the control point of view, this is equivalent to local controllers that cooperate with each other by exchanging the variables $\mathbf{z}$ between neighbors.

3.4 Numerical results

We have solved an optimal control problem on the Hydro Power Valley where the cost function consisted of an $\mathscr{L}_2$ and an $\mathscr{L}_1$ term that correspond to the tracking of the steady state and tracking of the power reference, respectively. The control horizon was 24 hours, which we divided into 48 subintervals.

In Table 3.1 the running time of one SQP iteration is shown. It is clear that by using Multiple Shooting (MS) and Distributed Multiple shooting (DMS) one can solve the same problem in much less running time compared to a serial solution.

In Table 3.2 we show the number of iterations needed to achieve the requested tolerance in the KKT conditions. One can conclude that in this specific application it is sufficient to do Newton-method only on the first coupling coefficient.

In Figure 3.1 the KKT-tolerance is depicted in a logarithmic scale against the iteration counter. Our experiments correspond to what we can expect from SQP convergence theory, namely we achieved linear convergence.

In Figure 3.2 we compared the planned power production and the reference power within 1 day. This can be considered as an accurate tracking, since the maximal tracking error is never larger than 10^{-3}.

Method	Integr.	QP sol.	Σ
fSQP (SS – serial)	8 min 43 s	0.01 s	8 min 43 s
fSQP (MS – 48 cores)	10.88 s	9.93 s	20.81 s
fSQP (DMS – 384 cores)	1.92 s	9.93 s	11.85 s
aSQP(5) (DMS – 384 cores)	1.31 s	9.93 s	11.24 s
aSQP(1) (DMS – 384 cores)	1.17 s	9.93 s	11.10 s
aSQP(0) (DMS – 384 cores)	1.15 s	9.93 s	11.08 s

Table 3.1: Guessed running time of one SQP iteration. fSQP: full Sequential Quadratic Programming method. aSQP(x): adjoint-based Sequential Quadratic Programming method with taking only x out of 10 directions in the Jacobians.

	1E-3	1E-4	1E-5
fSQP	5	6	7
aSQP(5)	5	6	7
aSQP(1)	5	6	7

Table 3.2: Number of iterations needed by different methods in order to achieve certain tolerance in the KKT-conditions

In Figure 3.3 the evolution of the water levels in time are depicted. In the beginning of the prediction horizon an oscillatory behavior may be observed that is due to the aggressivity of the power tracking. Our formulation incorporates the steady state tracking as well (next to power tracking) and thus the system converges to the steady state at the end of the prediction horizon, while respecting hard constraints.

In Figure 3.4 the control plan of the HPV subsystems are shown, the same phenomenon may be observed as in the states, after a while the control profile gets smoother and smoother, driving the whole system to the steady state.

Performance analysis

In this section we characterize the quality of our controller with objective measures.

- Mean absolute tracking error: 6.3074×10^{-5} MW

- Mean quadratic tracking error: 6.9951×10^{-9} MW^2

- Power reference tracking index (EUR):

$$\int_0^{86400} \gamma \left| p_r(t) - \sum_{i=1}^{8} p_i(x_i(t), u_i(t)) \right| dt = 0.1986 \quad \text{EUR} \tag{3.6}$$
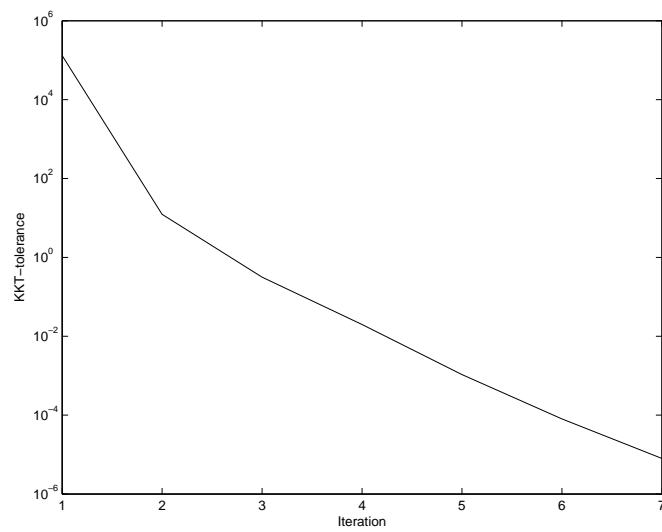
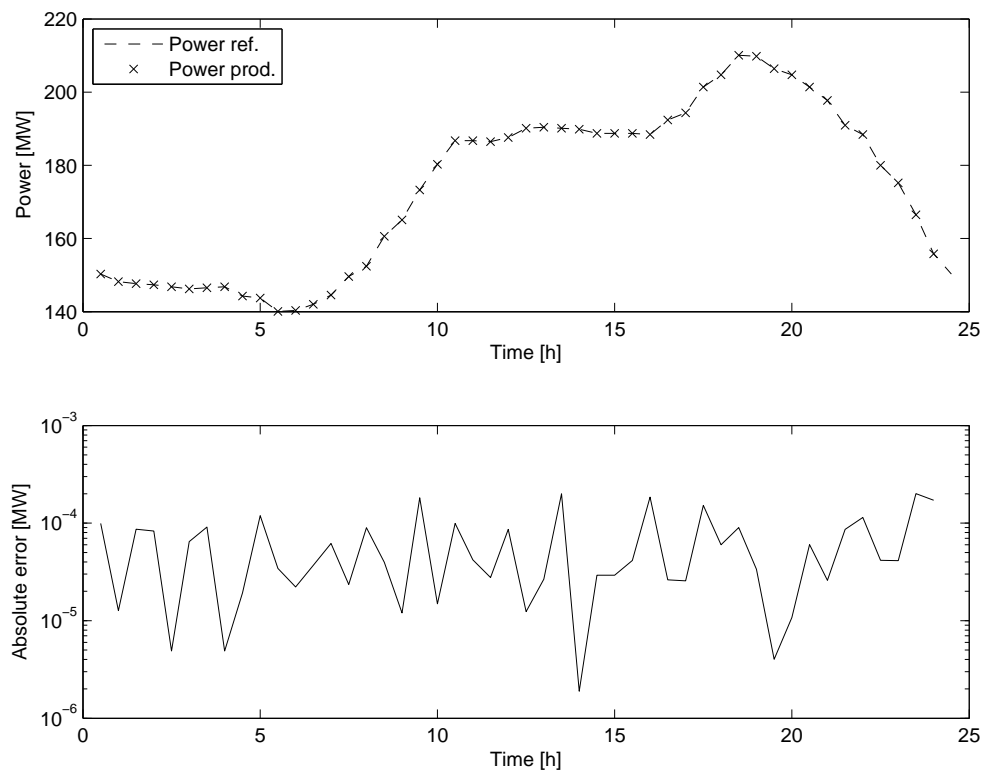Figure 3.1: Converge rate of fSQP and aSQP(5) methods.

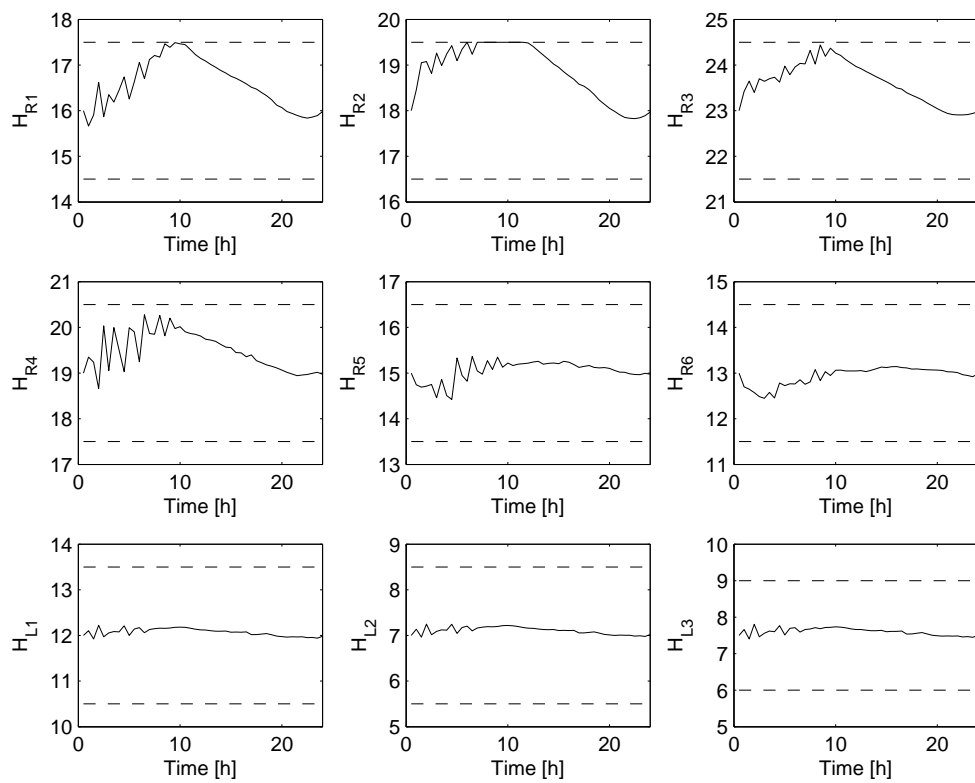Figure 3.2: Comparison of the reference power and the power generated.

Figure 3.3: Water levels in different reaches and lakes together with hard constraints along 24 hours.
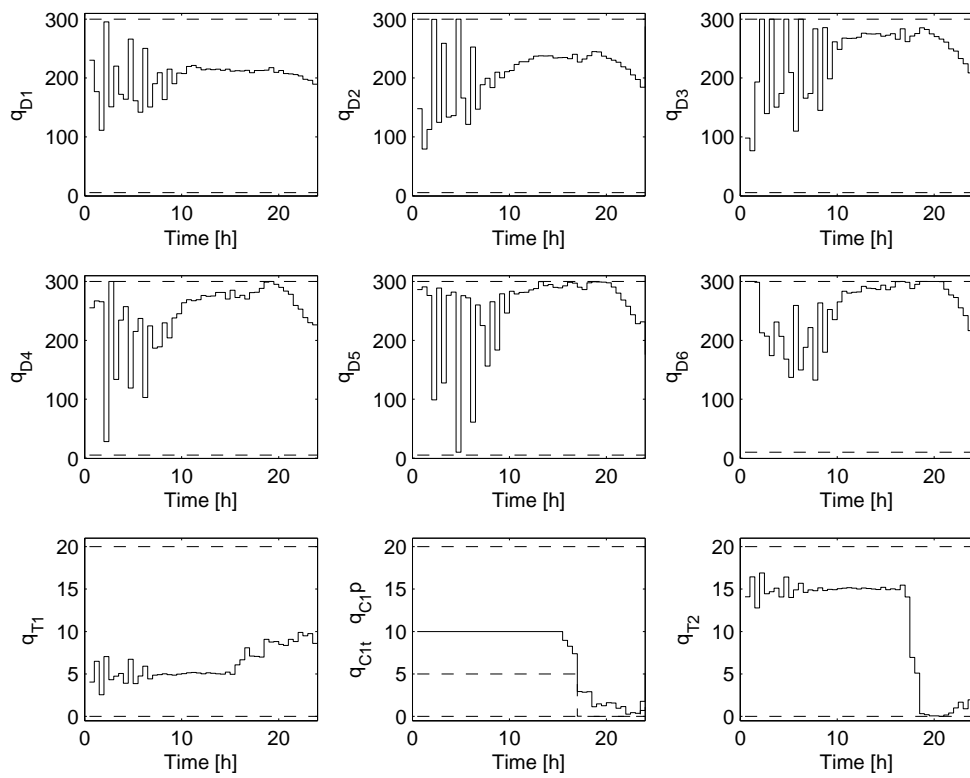
Figure 3.4: Control plan of reaches and lakes together with constraints for 24 hours.

- Optimality measure and constraint violation: The solution found is considered as an optimal solution (though not highly accurate) with

  - KKT tolerance 0.258831,

  - primal infeasability 1.08×10^{-12}

  - dual infeasability 1.67.

- Communication costs: The communication costs consist of sending and receiving vectors and matrices. 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 via MPI interface using double precision.

Chapter 4

Fast Gradient-Based Distributed MPC

In this chapter we present the application of a novel distributed MPC method in the Hydro Power Valley benchmark. Our approach make use of a distributed algorithm for networked optimization that is applicable to a class of convex non-smooth optimization problems [7]. The new algorithm is distributed in the sense that each subsystem only needs to communicate with its direct neighbors, and there is no need for a master controller. The distributed solution converges to the centralized solution with a fast convergence rate, thus enables real-time implementation in large-scale systems. We will first summarize the essence of the distributed optimization technique, then discuss its application to the Hydro Power Valley benchmark.

4.1 Distributed gradient-based algorithm for networked optimization

4.1.1 Problem setup

We consider convex non-smooth optimization problems with the following form:

$$\min_x \; J^h \triangleq \frac{1}{2}x^T H x + g^T x + \gamma \|Px - p\|_1 \tag{4.1}$$
$$\text{s.t. } A_1 x = B_1$$
$$A_2 x \leq B_2$$

where $x \in \mathbf{r}^n$, the matrix H is block-diagonal, positive definite, the matrices $A_1 \in \mathbf{r}^{q \times n}$, $A_2 \in \mathbf{r}^{r \times n}$ and $P \in \mathbf{r}^{m \times n}$ have sparsity structures. The sparsity will facilitate distributed implementation.

Based on the structure of H, we partition the full variable x into the set of M local variables $x_i \in \mathbf{r}^{n_i}$ as $x = [x_1^T, \ldots, x_M^T]^T$ and define corresponding H_i and g_i such that

$$J(x) \triangleq \frac{1}{2}x^T H x + g^T x = \sum_{i=1}^M \frac{1}{2}x_i^T H_i x_i + g_i^T x_i \tag{4.2}$$

Problem (4.1) represents a form of network optimization problems, in which each subsystem (or agent) is associated with a variable x_i. The subsystems are coupled through constraints and the 1-norm term, which is commonly used as a regularization term or as a soft constraint. Let us define the neighborhood $\mathcal{N}_i$ of each subsystem i as the group of all subsystems that couple with subsystem i

through either the constraints or the 1-norm term. Mathematical definition of $\mathcal{N}_i$ can be given as:

$$\mathcal{N}_i = \left\{ j \in \{1,\ldots,M\} \middle| \begin{array}{ccc} A_{1[ij]} & \neq & [0]_{n_i \times n_j}, \text{ or} \\ A_{2[ij]} & \neq & [0]_{n_i \times n_j}, \text{ or} \\ P_{[ij]} & \neq & [0]_{n_i \times n_j} \end{array} \right\} \tag{4.3}$$

where the subscript $[ij]$ refers to the sub-matrix corresponding to the variables x_i and x_j.

Our method will work with positive definite H, for this reason we need the following assumption:

Assumption 1 *We assume that each H_i in (4.2) is a real symmetric positive definite matrix that satisfies the following eigenvalue bounds*

$$\underline{\sigma}_i I \preceq H_i \preceq \bar{\sigma}_i I$$

where $0 < \underline{\sigma}_i \leq \bar{\sigma}_i < \infty$.

Assumption 2 *We assume that there exists a Slater vector, $\bar{x}$ to the optimization problem, i.e. a vector $\bar{x}$ such that $A_1 \bar{x} = B_1$ and $A_2 \bar{x} < B_2$. Further, we assume that $a_l, l = 1,\ldots,q$ are linearly independent.*

Remark 1 *Assumption 2 together with the fact that the cost function J^h is strongly convex implies that the minimum of the optimization problem (4.1) is always attained at a unique point, denoted by x^*.*

4.1.2 Dual problem

In the following section we will present the formulation of a dual problem to (4.1), which will be tackled with the new distributed optimization method.

We start by introducing auxiliary variables x_a to get the following optimization problem which is equivalent to (4.1):

$$\min_{x,x_a} J(x) + \gamma \|x_a\|_1 \tag{4.4}$$
$$\text{s.t. } A_1 x = B_1$$
$$A_2 x \leq B_2$$
$$Px - p = x_a$$

We introduce Lagrange multipliers $\lambda \in \mathbb{R}^q, \mu \in \mathbb{R}^r_{\geq 0}, \nu \in \mathbb{R}^m$ for relaxation of the constraints. The dual function is

$$q(\lambda,\mu,\nu) = \inf_{x,x_a} \left\{ J(x) + \gamma \|x_a\|_1 + \lambda^T (A_1 x - B_1) + \right.$$
$$\left. + \mu^T (A_2 x - B_2) + \nu^T (Px - p - x_a) \right\} \tag{4.5}$$

Let us make use of the definition of the conjugate function (cf. [17]) of a function $f(x)$:

$$f^\star(y) \triangleq \sup_x \left\{ y^T x - f(x) \right\}. \tag{4.6}$$

Using the notation of conjugate functions, we can rewrite (4.5) by rearranging the terms and replacing $\inf_x(\cdot)$ by $-\sup_x(-(\cdot))$ with the following form:

$$q(\lambda,\mu,\nu) = -J^\star(-(A_1^T \lambda + A_2^T \mu + P^T \nu))$$

$$-\sup_{x_a} \left[v^T x_a - \gamma \|x_a\|_1 \right] \tag{4.7}$$

Note that we can compute the conjugate of the scaled 1-norm explicitly:

$$
\begin{aligned}
\sup_{x_a} \left\{ v^T x_a - \gamma \|x_a\|_1 \right\} &= \sup_{x_a} \left\{ \sum_i \left[v^i x_a^i - \gamma |x_a^i| \right] \right\} \\
&= \sum_i \left\{ \sup_{x_a^i} \left[v^i x_a^i - \gamma |x_a^i| \right] \right\} \\
&= \begin{cases} 0 & \text{if } \|v\|_\infty \le \gamma \\ \infty & \text{else} \end{cases}
\end{aligned}
$$

where the superscript i denotes the i-th element in the vector. We see that the conjugate of the scaled 1-norm is an indicator function of a hyper box. In order to have a definite value for the sup operator, we must look for v in the box where the maximal values of all the elements is γ. In this way, we absolutely respect the 1-norm term in the cost function of (4.4), while relaxing only the constraint.

Remark 2 *The fact that the conjugate of the scaled 1-norm becomes a box-constraint for the introduced dual variables, v, is important for distribution reasons. The projection operations onto the box are parallelizable, thus facilitating the distributed computation.*

We introduce the following notations:

$$\mathscr{A} = [A_1^T \ A_2^T \ P^T]^T \qquad\qquad \mathscr{B} = [B_1^T \ B_2^T \ p^T]^T \qquad\qquad z = [\lambda^T \ \mu^T \ v^T]^T$$

where $\mathscr{A} \in \mathbb{R}^{(q+r+m)\times n}$, $\mathscr{B} \in \mathbb{R}^{q+r+m}$ and $z \in \mathbb{R}^{q+r+m}$. The set of feasible dual variables is defined as

$$Z = \mathbb{R}^q \times \mathbb{R}^r_{\ge 0} \times [-\gamma, \gamma]^m \tag{4.8}$$

where $[-\gamma, \gamma]^m$ stands for m times product of the set $[-\gamma, \gamma]$. With these definitions the dual problem can be rewritten in a compact form as follows:

$$q(z) := -J^\star(-\mathscr{A}^T z) - \mathscr{B}^T z.$$

Note that as J is a quadratic function with positive definite H, the conjugate function $J^\star(y)$ has the explicit formula [4]:

$$J^\star(y) = \frac{1}{2}(y-g)^T H^{-1}(y-g) \tag{4.9}$$

and it is differentiable with the gradient:

$$\nabla J^\star(y) = H^{-1}(y-g) \tag{4.10}$$

Under Assumption 2 it is well known (cf. [4, §5.2.3]) that there is no duality gap, i.e. we can get the minimum of (4.4) by solving the dual problem:

$$\max_{z\in Z} q(z) \tag{4.11}$$

To present the algorithm in a more familiar way of convex optimization, instead of maximizing the concave function q, we will focus on minimizing the convex function f which is the opposite of q:

$$f(z) \triangleq -q(z) = J^\star(-\mathscr{A}^T z) + \mathscr{B}^T z$$
$$= \frac{1}{2}(\mathscr{A}^T z + g)^T H^{-1}(\mathscr{A}^T z + g) + \mathscr{B}^T z \tag{4.12}$$

In the next section, we will present the distributed algorithm for solving the dual problem, together with the convergence property.

4.1.3 Distributed dual accelerated proximal gradient method (DDAPG)

Our algorithm is based on a fast gradient technique, the accelerated proximal gradient methods (APG). This method has convergence rate $O(\frac{1}{k^2})$ as developed in [12] and further elaborated and extended in [3, 13, 14, 21, 22].

In this section, we present a distributed algorithm based on the APG method which achieves the same convergence rate with the centralized counterpart. The main idea is to exploit the problem structure such that the APG computations can be distributed to subsystems. The detailed algorithm was discussed in [7], in the following we summarize the distributed accelerated proximal gradient algorithm for solving the dual problem.

Let us partition the constraint matrices to each row as

$$\mathscr{A} = \begin{bmatrix} a_1^T \\ \vdots \\ a_{q+r+m}^T \end{bmatrix}, \quad \mathscr{B} = \begin{bmatrix} b_1 \\ \vdots \\ b_{q+r+m} \end{bmatrix}$$

in which each $a_l, l = 1, \ldots, q+r+m$ is a column vector, corresponding to a scalar constraint in (4.4) as $a_l^T x \leq b_l, l = 1, \ldots, q$ or $a_l^T x = b_l, l = q+1, \ldots, q+r+m$. Recall the definition of the neighborhood in (4.3), it can be seen that if the lth constraint involves subsystem i, then all the non-zero values of a_l only involves i and the neighboring subsystems $j \in \mathcal{N}_i$.

Let us also divide the set $\{1, \ldots, q+r+m\}$ into M subsets $\mathscr{L}_i, i = 1, \ldots, M$, such that each subsystem i will be responsible for all computations concerning the constraints $l \in \mathscr{L}_i$. There are different ways to make this division, the only requirement is that for every $l \in \mathscr{L}_i$, we have $a_{li} \neq 0$, with a_{li} represents the sub-vector of a_l that corresponds to the variable x_i in the lth constraint of (4.4).

The *Distributed Dual Accelerated Proximal Gradient* algorithm is given as below.

Algorithm 1 *Distributed Dual Accelerated Proximal Gradient*

Initialize $z^0 = z^{-1}$ and x^{-1}
In every node, i, the following computations are performed:
For $k \geq 0$

   1. Compute

$$x_i^k = -H_i^{-1}\left(\sum_{j \in \mathcal{N}_i}\left[\sum_{l \in \mathscr{L}_j^1} a_{li} z_l^k\right] + g_i\right) \tag{4.13}$$

$$\bar{x}_i^k = \frac{2k+1}{k+2}x_i^k - \frac{k-1}{k+2}x_i^{k-1} \tag{4.14}$$

---

2. *Send $\bar{x}_i^k$ to each $j \in \mathcal{N}_i$, receive $\bar{x}_j^k$ from each $j \in \mathcal{N}_i$*

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

$$d_l = \sum_{j \in \mathcal{N}_i} a_{lj}^T \bar{x}_j^k - b_l \tag{4.15}$$

$$z_l^{k+1} = z_l^k + \frac{k-1}{k+2}(z_l^k - z_l^{k-1}) + \frac{1}{L}d_l^k, \quad l \leq q \tag{4.16}$$

$$z_l^{k+1} = \max\left\{0, z_l^k + \frac{k-1}{k+2}(z_l^k - z_l^{k-1}) + \frac{1}{L}d_l^k\right\},$$
$$q < l \leq q+r \tag{4.17}$$

$$z_l^{k+1} = \min\left\{\gamma, \max\left[-\gamma, z_l^k + \frac{k-1}{k+2}(z_l^k - z_l^{k-1})\right.\right.$$
$$\left.\left. + \frac{1}{L}d_l^k\right]\right\}, \quad q+r < l \leq q+r+m \tag{4.18}$$

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

The convergence rates for the dual function f and the primal variables when running Algorithm 1 are stated in the following theorem.

Theorem 1 *Algorithm 1 has the following convergence rate properties:*

1. *Denote z^* as an optimizer of the dual problem (4.11). The convergence rate is:*

$$f(z^k) - f(z^*) \leq \frac{2L\|z^0 - z^*\|_2^2}{(k+1)^2}, \forall k \geq 1 \tag{4.19}$$

2. *Denote x^* as the unique optimizer of the primal problem. The rate of convergence for the primal variable is*

$$\|x^k - x^*\|_2^2 \leq \frac{4\bar{\sigma}L\|z^0 - z^*\|_2^2}{\underline{\sigma}(k+1)^2}, \forall k \geq 1 \tag{4.20}$$

The proof of Theorem 1 and more details of Algorithm 1 can be found in [7].

This result enables implementing the algorithm DDAPG in a distributed fashion and achieve fast convergence rate as the centralized APG algorithm. The improvement of the convergence rate is helpful when we want to use this method for solving online optimal control problem, such as distributed model predictive control. In the following section we will apply the new algorithm to DMPC of the Hydro Power Valley benchmark.

4.2 Model construction and distributed MPC configuration for HPV application

Since our proposed method is designed for linear systems, we first need to obtain a linear model of the HPV. We linearize the given nonlinear model at the steady state condition to get the linear model.

Although the approximation property of the linear model is good only near the steady state region, the linear model is well suited for the control, as will be shown by the simulation results.

One of the difficulties for applying a linear MPC approach to the HPV problem is the discontinuity of the power functions associated with the ducts C_1 and C_2. The discontinuity is caused due to the fact that the flow through C_1 (or C_2) can have two directions and the powers generated or consumed do not have equivalent coefficients. To deal with this issue, we use double-flow technique, which means introducing two separate positive variables to express the flow in C_1:

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

- $q_{C_{1t}}(t)$: 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 (by $q_{C_{1p}}(t)$) and consumed (by $q_{C_{1t}}(t)$). This approach allows the optimization solver to deal with continuous variables only. When the solution is obtained, we combine the virtual flows to get the real flow through C_1:

$$q_{C_1}(t) = q_{C_{1t}}(t) - q_{C_{1p}}(t) \tag{4.21}$$

The double-flow approach is also applied for C_2. Consequently, the new linear model has 12 inputs. Another issue of the linear model is that the spatial discretization results in dependencies of adjacent states representing water levels along the reaches, leading to some unobservable and uncontrollable modes. Moreover, the linear model has a large number of states, causing computational burden. We will use balanced truncation for model order reduction[28] so that the reduced model has only observable and controllable modes.

Let us first describe the balancing transformation. Consider a discrete-time linear model with state-space realization:

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k)$$

We can compute the controllability Gramian P as follows:

$$APA^T + BB^T = P \tag{4.22}$$

and the observability Gramian Q as:

$$A^T QA + C^T C = Q \tag{4.23}$$

With a transformation matrix $T, |T| \neq 0$ for state transformation $\bar{x} = Tx$, the Gramians are transformed to:

$$\bar{P} = TPT^T, \quad \bar{Q} = T^{-T} QT^{-1} \tag{4.24}$$

We can find a particular matrix T such that

$$\bar{P} = \bar{Q} = \text{diag}(\sigma_1, \ldots, \sigma_n) \tag{4.25}$$

with $\sigma_i \geq 0, \forall i$. This is called a balancing transformation. The controllability and observability Gramians of the new system realization are equal and diagonal, consisting of entries $\sigma_1, \ldots, \sigma_n$ which are called Hankel singular values.

The truncated model is obtained by removing the modes that correspond to small σ_i, and all the modes of the reduced model are both controllable and observable (corresponding to $\sigma_i > 0$). This model reduction method is called balanced truncation.

Since we want to keep the structure of the HPV model, we perform balanced truncation for each local system. With a particular choice of the modes to be truncated, we obtain a 46-state reduced model that approximately represent the dynamics of the full linear model with 249 states.

Finally, the discrete MPC optimization problem can be cast into the form of problem (4.1). We then use Algorithm 1 to solve the optimization problem at each sampling time in a distributed setting. In particular, we use 8 subsystems as defined in Table 4.1.

Table 4.1: Subsystem configuration

Subsystem	Neighborhood set	Input variables	Output variables
1	$\{1,3,4\}$	$u_1 = [q_{T1}, q_{C1t}, q_{C1p}]^T$	$y_1 = [h_{L1}, h_{L2}]^T$
2	$\{2,6,7\}$	$u_2 = [q_{T2}, q_{C2t}, q_{C2p}]^T$	$y_2 = h_{L3}$
3	$\{3,1,4\}$	$u_3 = q_{D1}$	$y_3 = h_{R1}$
4	$\{4,1,3,5\}$	$u_4 = q_{D2}$	$y_4 = h_{R2}$
5	$\{5,4,6\}$	$u_5 = q_{D3}$	$y_5 = h_{R3}$
6	$\{6,2,5,7\}$	$u_6 = q_{D4}$	$y_6 = h_{R4}$
7	$\{7,2,6,8\}$	$u_7 = q_{D5}$	$y_7 = h_{R5}$
8	$\{8,7\}$	$u_8 = q_{D6}$	$y_8 = h_{R6}$

The control parameters are chosen as follows:

- Time step: $T = 1800s$.

- Horizon length: $N = 10$.

- Simulation time: 48 steps (1 day).

- $\gamma = 500$.

4.3 Simulation results

We made simulations of the HPV control with the proposed DDAPG algorithm. To demonstrate the fast convergence property of DDAPG algorithm, we also solve the optimization problem at each time step in a centralized way, this is done by transforming the problem into quadratic problem and applying the *quadprog* solver of MATLAB. The comparison of computation time of the distributed algorithm and the centralized solver is presented in Figure 4.1. Note that our distributed algorithm is implemented in MATLAB, hence it is a fair to compare it against a solver written in MATLAB, other than a solver written in C. Figure 4.1 shows that the total computation time of DDAPG algorithm is much shorter than the computation time of *quadprog*, this reflexes the fast convergence rate of DDAPG and the efficiency of dealing with the 1-norm element in the cost function.

The power reference tracking is plotted in Figure 4.2, showing that our distributed controller performs tracking very well. This result validates the linear model and the model reduction technique, the reduced linear model is good enough for control of the HPV system.

In Figures 4.3 and 4.4, we see that the input and output constraints are all satisfied. The constraint satisfaction is achieved due to the fact that the solution of the distributed MPC is indeed the centralized

solution of the dual problem, and since there is no duality gap, the dual optimum equals the primal optimum, thus giving us the primal solution of the centralized MPC problem.
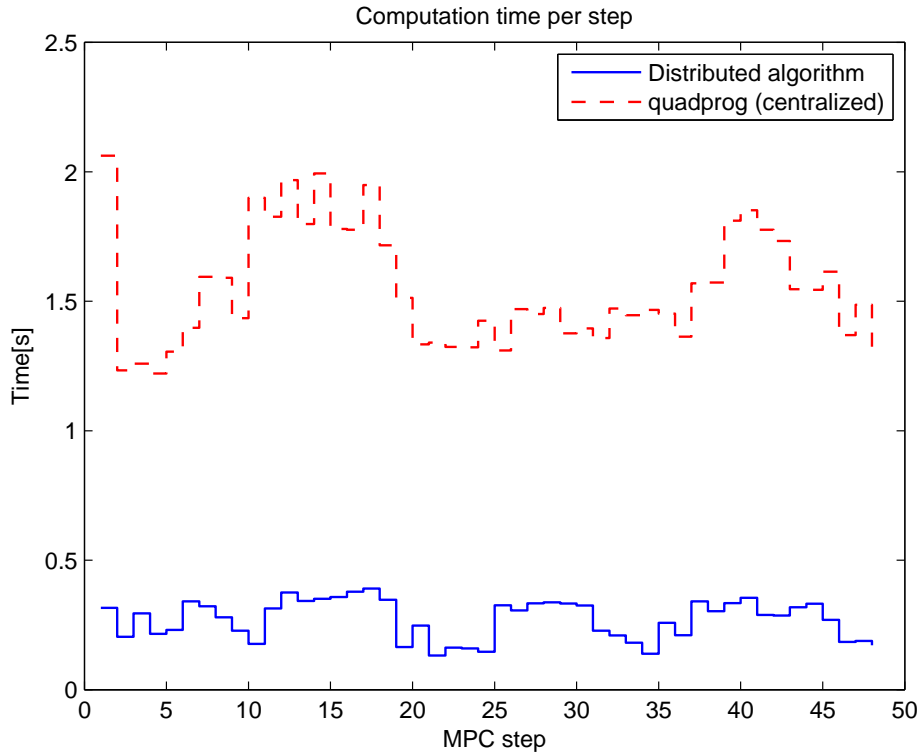

Figure 4.1: Comparison of computation time per each time step

4.4 Performance analysis

In this section we characterize the quality of our controller with objective measures.

- Mean absolute tracking error: $1.65 \, MW$.

- Mean quadratic tracking error: $4.85 \, MW^2$.

- Power reference tracking index (EUR):

$$\int_0^{86400} c(t) \left| p_r(t) - \sum_{i=1}^{8} p_i(x_i(t), u_i(t)) \right| dt = 2568$$

- Power reference tracking index 2 (EUR):

$$\int_0^{86400} c(t) \min\left(p_r(t) - \sum_{i=1}^{8} p_i(x_i(t), u_i(t)), 0 \right) dt$$
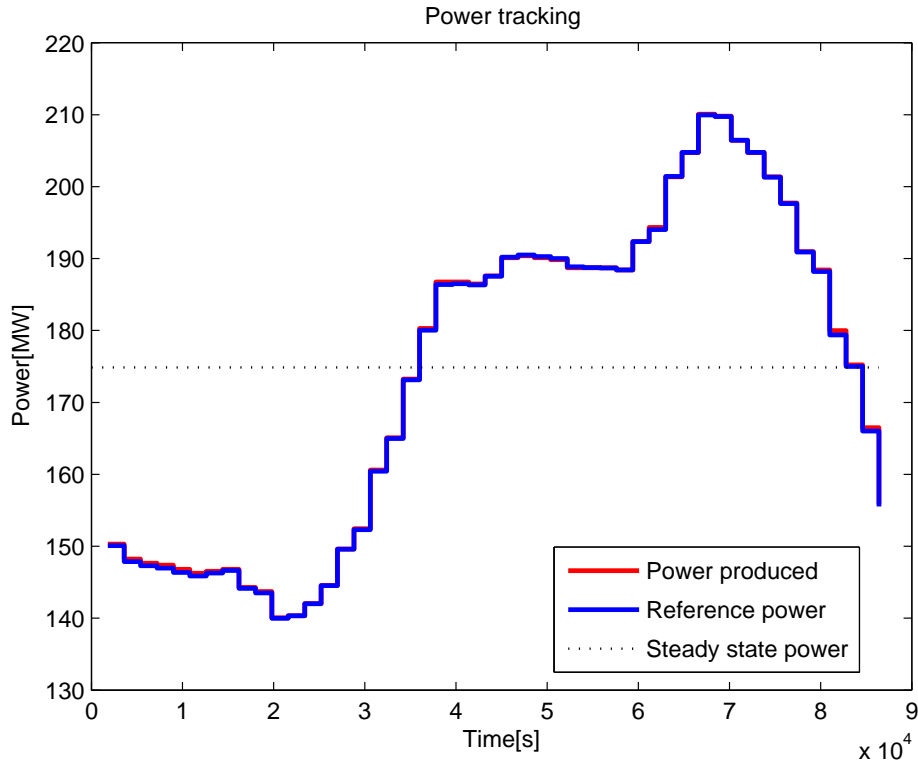
Figure 4.2: Simulation result of reference power tracking

$$+0.5 \int_0^{86400} c(t) \min \left(\sum_{i=1}^{8} p_i(x_i(t), u_i(t)) - p_r(t), 0 \right) dt = 2440$$

- Constraint violation: There is no violation for input constraints. The output constraints are slightly violated, only for the water levels of the reaches R_5 and R_6, with the following indices:

    - Maximum constraint violation of h_{R_5}: 0.02 m. Accumulated constraint violation of h_{R_5} over 48 steps: 0.049 m/day.

    - Maximum constraint violation of h_{R_6}: 0.023 m. Accumulated constraint violation of h_{R_6} over 48 steps: 0.058 m/day.

- Communication costs: The communication costs consist of sending and receiving primal vectors and dual vectors between neighbors in each iteration. The communication costs vary depending on the total number of iterations needed until convergence is obtained. We provide the graphs of the communication costs with respect to each MPC step in Figures 4.5 and 4.6.

4.5   Conclusions

We have presented a distributed MPC method that is applicable to the Hydro Power Valley benchmark. The distributed MPC is based on a distributed accelerated proximal gradient method for solving the
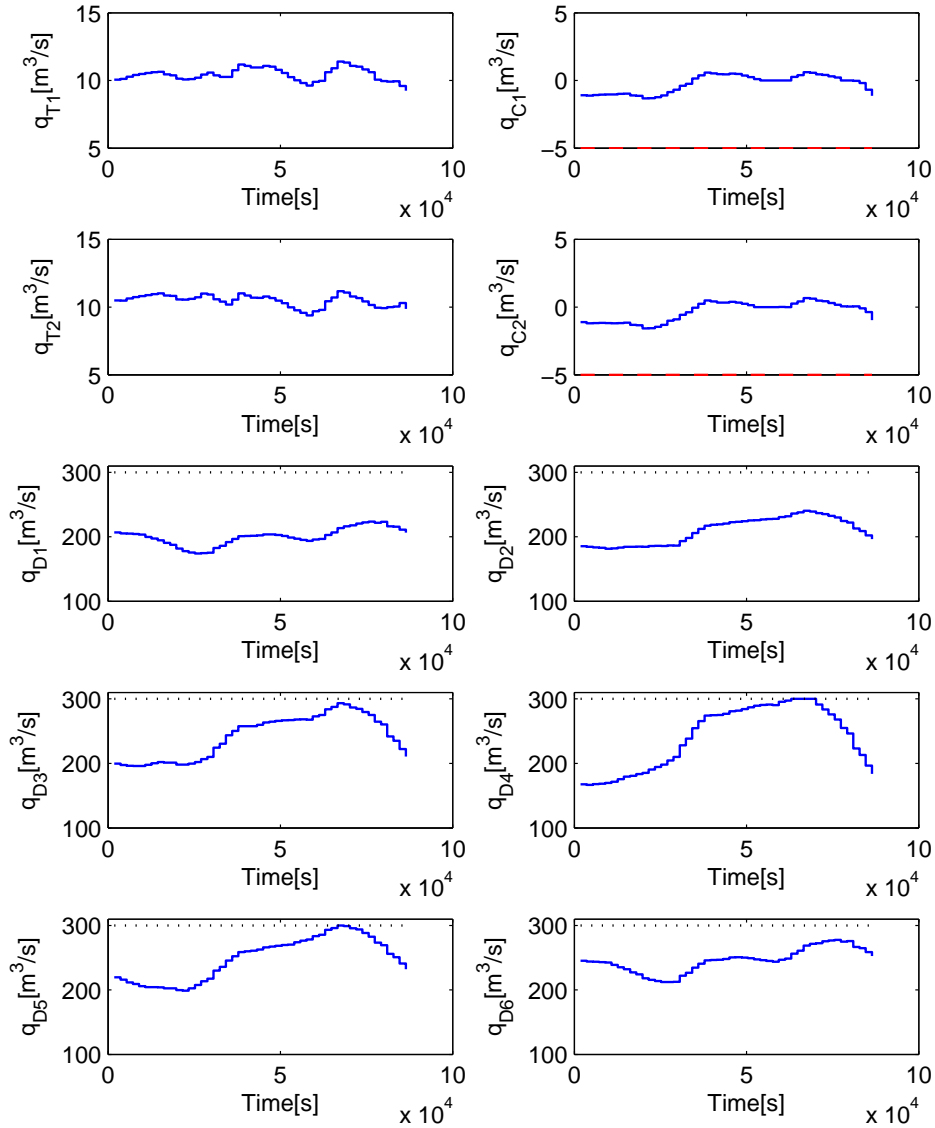
Figure 4.3: Input constraint satisfaction with the DDAPG

dual optimization problem. The distributed solution converges to the centralized solution with a fast convergence rate, thus enable real-time implementation in large-scale systems. We have also shown that the new algorithm is suitable for the control problem of the Hydro Power Valley benchmark. The simulation results has shown that the power reference can be tracked well, while all the operational constraints are satisfied. Moreover, our distributed algorithm requires much shorter computational time than a centralized QP approach, even when we compare the total time of computations.
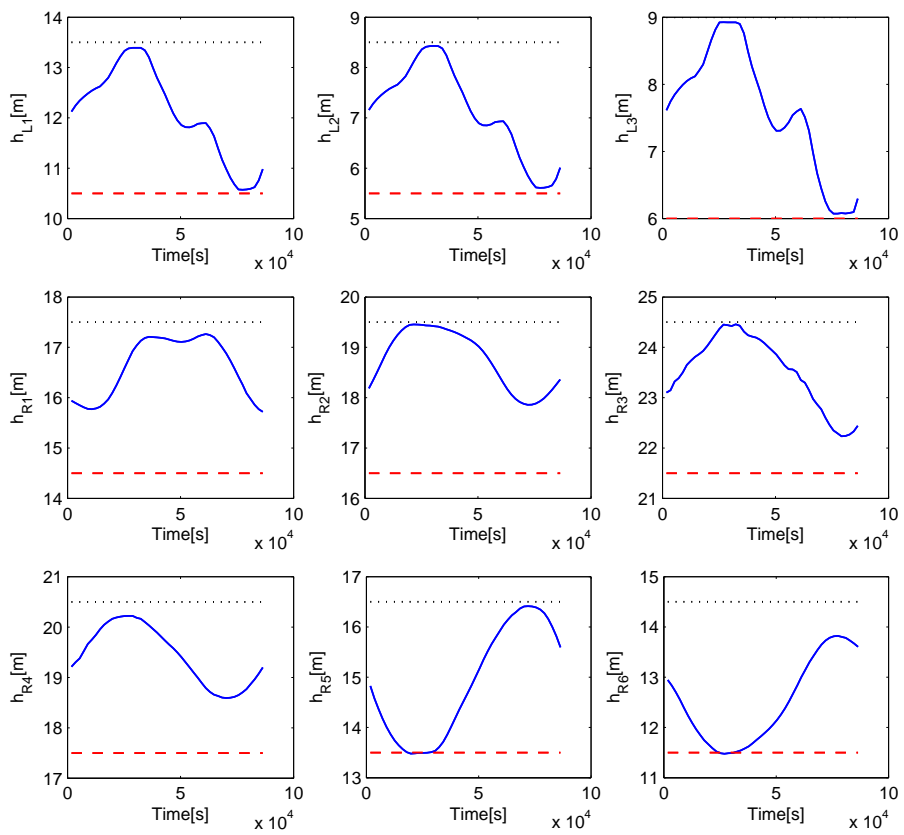
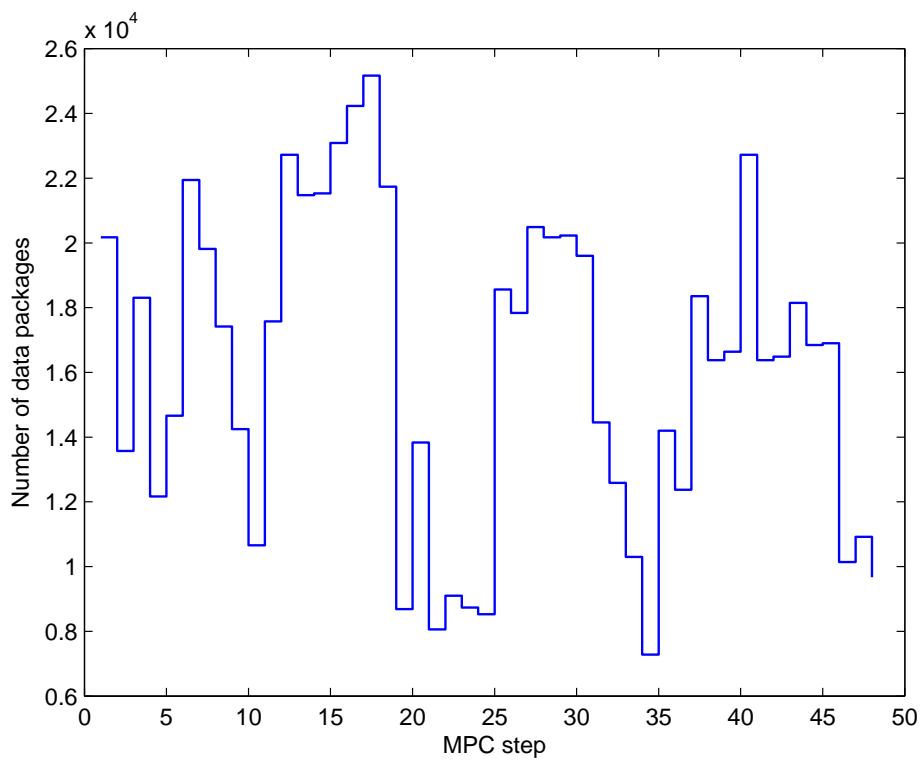Figure 4.4: Output constraint satisfaction with the DDAPG

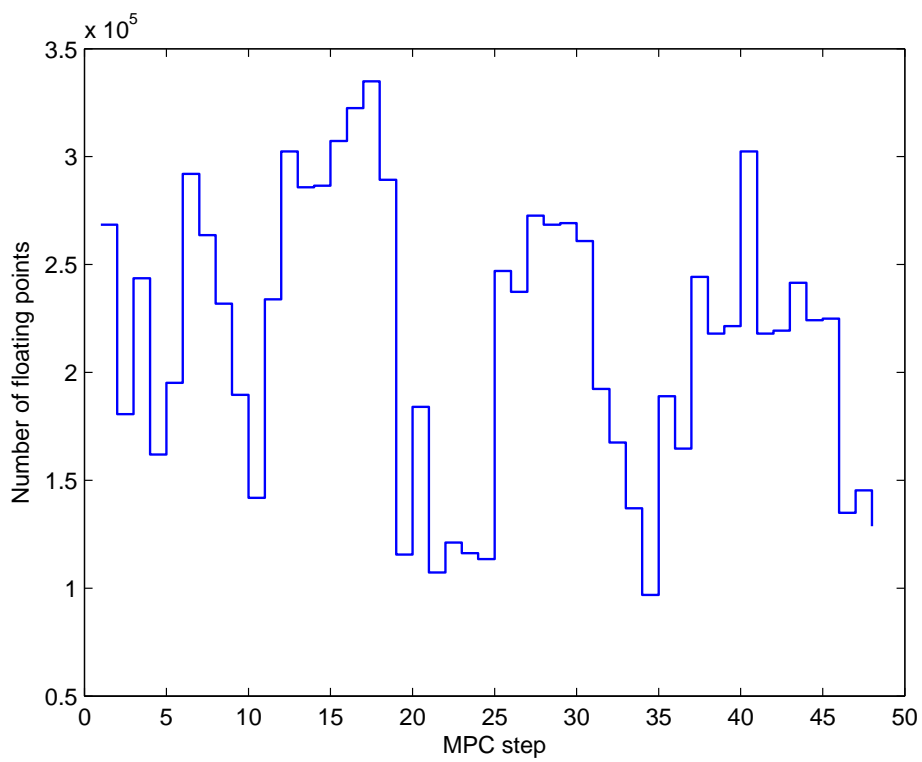Figure 4.5: Number of data packages transmitted between distributed controllers

Figure 4.6: Number of floating point reals transmitted between distributed controllers

Chapter 5

Hierarchical Infinite Horizon Model Predictive Controller

5.1 Introduction

The model of the HPV was linearized in order to develop a hierarchical Infinite Horizon Model Predictive Controller (IHMPC). The resultant model had certain characteristics that made the control troublesome. Problems regarding the controllability and the stability of the system were:

- Controllability: This property was tested finding the controllability matrix and computing its rank. Initial tests showed bad conditioning of the matrix, then it seemed that it was not full rank, but modifying the tolerance of the estimation it presented full rank. This result was caused by the large difference between singular values, i.e. the system is controllable, but there are many states that need large effort to be controlled.

- Integrating system: if the eigenvalue of the discrete linear model have one pure real value exactly in the unitary circle in a complex space, it is said that the system is an integrating system. In this case there were eight singular values that had a pure real value of one, so the system had eight integrating states.

Since IHMPC assures stability only if the discrete linear system is stable, it cannot be applied directly to the system. The approach selected to skip this problem was stabilizing the system using a state feedback controller, with the variable change:

$$u(k) = -Kx(k) + v(k) \tag{5.1}$$

where the state feedback gain matrix K could be found by means of pole placement or with an optimization problem, $v(k)$ being the new manipulated variable. A necessary condition to find the gain matrix was that the system should be stabilizable [26]. With that condition achieved it could be calculated as the solution of an optimization problem as $K = R^{-1}B^T P(t)$, where P was found solving the Ricatti equation [6].

With the proposed modification, the dynamics of the system changed and the state representation became:

$$
\begin{aligned}
x(k+1) &= (A - BK)x(k) + Bv(k) \\
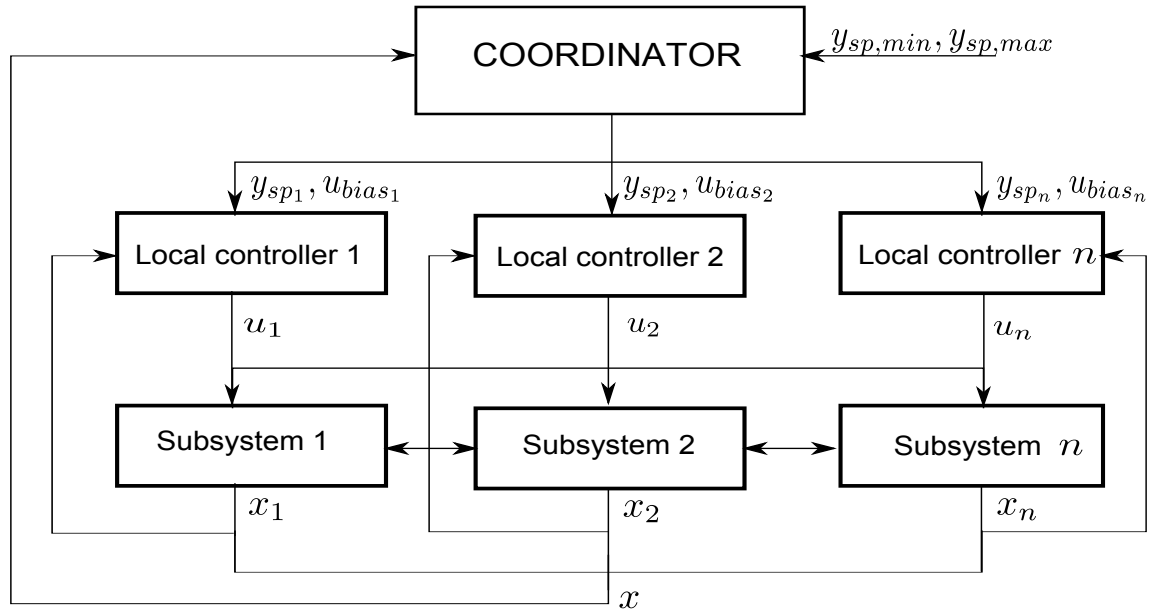&= \widetilde{A}x(k) + Bv(k)
\end{aligned}
\tag{5.2}
$$

Figure 5.1: Hierarchical structure

where the matrix $\widetilde{A} = (A - BK)$ is stable (the norm of all eigenvalues of $\widetilde{A}$ are inside of the unitary circle). Since the plant is stable, the MPC can now control it by using $v(k)$ as manipulated variable. The problem is then reduced to control the new state feedback system. Also, by the change on the manipulated variable the constrains must be expressed in terms of $v(k)$. This was done replacing (5.2) into the constraints for the sates and for the manipulated variables. In the next section the hierarchies used on the design of the proposed hierarchical controller will be explained.

5.2 Hierarchical control approach for hydro-power valley

In this section a hierarchical structure is designed for the HPV problem, it was based on an IHMPC with zone control (coordinator) as controller applied over the entire plant, with eight IHMPCs that controlled the subsystems (see Figure 5.1). The coordinator calculated hierarchical input variables and output references with values inside the designed zone, this information was taken by the controllers for generating the input variables that were applied, achieving the reference that the coordinator generated.

5.2.1 Coordinator of the IHMPC with zone control

The zone control strategy is implemented in applications where the exact values of the controlled outputs are not important, as long as they remain inside a range with specific limits [8]. The MPC optimization problem implemented with the zone control strategy is as follows:

$$\min_{\Delta u_k, y_{sp,k}, \delta_k} J_{k,\infty} = \sum_{j=1}^{\infty} \left(y(k+j) - y_{sp,k} - \delta_k \right)^T Q \left(y(k+j) - y_{sp,k} - \delta_k \right)$$

$$+ \sum_{j=1}^{N_c-1} \Delta u(k+j|k)^T R \Delta u(k+j|k) + \delta_k^T S \delta_k$$

subject to

$$x(k+1) = Ax(k) + B\Delta u(k)$$

$$y(k) = Cx(k) + Du(k)$$

$$y_{sp_{min}} \leq y_{sp,k} \leq y_{sp_{max}} \tag{5.3}$$

$$-\Delta u_{min} \leq \Delta u(k+j|k) \leq \Delta u_{max}$$

$$u_{min} \leq u(k+j|k) \leq u_{max}$$

where N_c is the control horizon, $\Delta u(k)$ is the increment of the manipulated variables, δ_k denotes the slack variable, $y(k)$ is the output of the system, A, B, C, D are the matrices associated with the linear model of the system, and $y_{sp,k}$ is the reference value for the output of the system, with $u(k)$ the control action, Q, R, S wighting matrices, and the subindex *min* and *max* the minimal and maximal values (respectively) of the corresponding variable. In this formulation the overall system model was represented as a discrete, linear time-invariant (LTI) model. For the HPV was only necessary to add the following constraint to 5.3:

$$C_p y = \sum_{i=1}^{8} P_i = P_{ref} \tag{5.4}$$

where C_p is a selection matrix and P_i is the power generated by the i-th subsystem. Finally the optimization problem that solved the coordinator was:

$$\min_{\Delta u_k, y_{sp,k}, \delta_k} J_{k,\infty} = \sum_{j=1}^{\infty} \left(y(k+j) - y_{sp,k} - \delta_k \right)^T Q \left(y(k+j) - y_{sp,k} - \delta_k \right)$$

$$+ \sum_{j=1}^{N_c-1} \Delta u(k+j|k)^T R \Delta u(k+j|k) + \delta_k^T S \delta_k$$

subject to

$$x(k+1) = Ax(k) + B\Delta u(k)$$

$$y(k) = Cx(k) + Du(k)$$

$$C_p y = P_{ref}$$

$$y_{sp_{min}} \leq y_{sp,k} \leq y_{sp_{max}} \tag{5.5}$$

$$-\Delta u_{min} \leq \Delta u(k+j|k) \leq \Delta u_{max}$$

$$u_{min} \leq u(k+j|k) \leq u_{max}$$

where P_{ref} is the power reference of the valley.

5.2.2 Local controllers of the extended IHMPC

The extended infinite horizon MPC is defined as follows [15]:

$$\min_{\Delta u_k, \delta_k} J_{k,\infty} = \sum_{j=1}^{\infty} \left(y_i(k+j) - y_{sp_i} - \delta_k \right)^T Q \left(y_i(k+j) - y_{sp_i} - \delta_k \right)$$

$$+ \sum_{j=1}^{N_c-1} \Delta u_i(k+j|k)^T R \Delta u_i(k+j|k) + \delta_k^T S \delta_k$$

subject to

$$x_i(k+1) = A_{ii} x_i(k) + B_{ii} \Delta u_i(k)$$

$$y_i(k) = C_{ii} x_i(k) + D_{ii} u_i(k)$$

$$-\Delta u_{i\min} \le \Delta u_i(k+j|k) \le \Delta u_{i\max} \tag{5.6}$$

$$u_{i\min} \le u_i(k+j|k) \le u_{i\max}$$

where N_c is the control horizon, A_{ii}, B_{ii}, C_{ii}, D_{ii} are the blocks of the matrices A, B C, D relating the local states and control inputs, $Q \in \Re^{n_{y_i} \times n_{y_i}}$, $R \in \Re^{n_{u_i} \times n_{u_i}}$ are positive definite weighting matrices, $\delta_k \in \Re^{n_{y_i}}$ is a vector of slack variables and $S \in \Re^{n_{y_i} \times n_{y_i}}$ is assumed positive definite. Observe that each slack variable refers to a given controlled output. Weighting matrix S was selected such that the controller pulls to zero the slacks or at least minimize them depending on the number of inputs. To reduce the infinite to finite horizon controller was necessary to add a terminal state penalty $\overline{Q}$ (that is obtained by solving a Lyapunov equation) and a terminal constraints to prevent the cost to become unbounded.

5.3 Simulation results

The control objective of the HPV was to follow a power reference while the levels depicted earlier had to be maintained into a certain zone. So, an infinite horizon MPC with zone control was implemented, the power of each plant had a tracking reference while the levels had a zone defined by the coordinator. In the results presented below both the coordinator and the local control actions were computed at the same time step in a sequential way, i.e. first the coordinator computed the power for each subsystem and the zones for the levels. After that, the local controllers computed the optimal control inputs to be applied on each subsystem.

The hierarchical control structure could manage the power reference problem in the HPV benchmark, with the extra constrain the controllers could follow the total plant power reference (Figure 5.2). The output variables (levels and individual system powers) were supposed to be into a designated zone (Figures 5.3 and 5.4), the simulations show how the controller had to move the output variables to other set point to achieve the power reference. In certain cases output variables went out of the zone, as the levels in the reach 1 (R_1), but immediately the variable was forced to come back. In Figure 5.5 the input variables are shown, it can be seen that the behavior of the variables had a tendency, i.e. the plant appeared to be stable.

From the simulation results it is possible to conclude that the levels variables in the hierarchical control structure could to be controlled into the zone. In this case the local controllers used the information from the coordinator to correct the input variables, in order to achieve the reference that the coordinator sent.
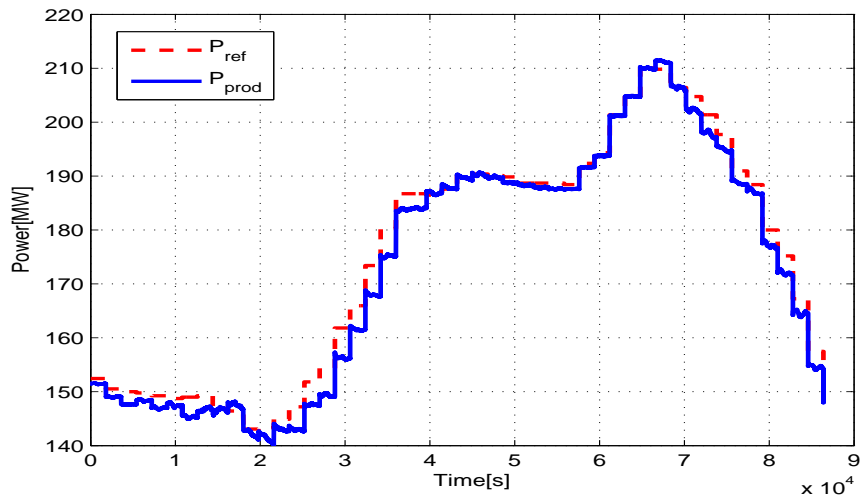
Figure 5.2: Comparison between the power produced by the HPV with the power reference
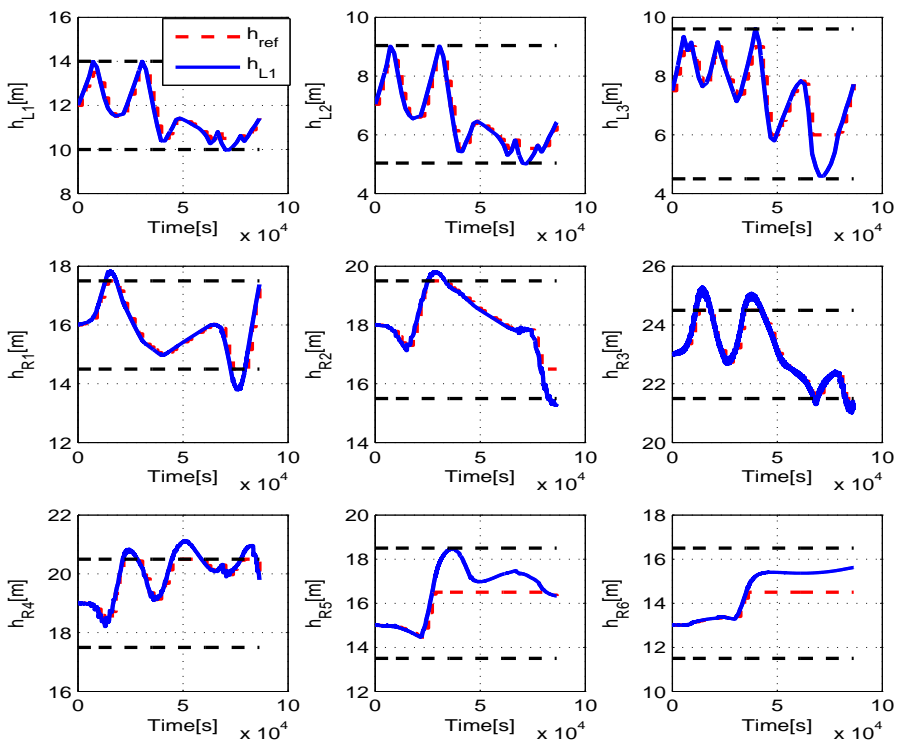

Figure 5.3: Behavior of the levels in the lakes and the levels at the damps of the HPV.
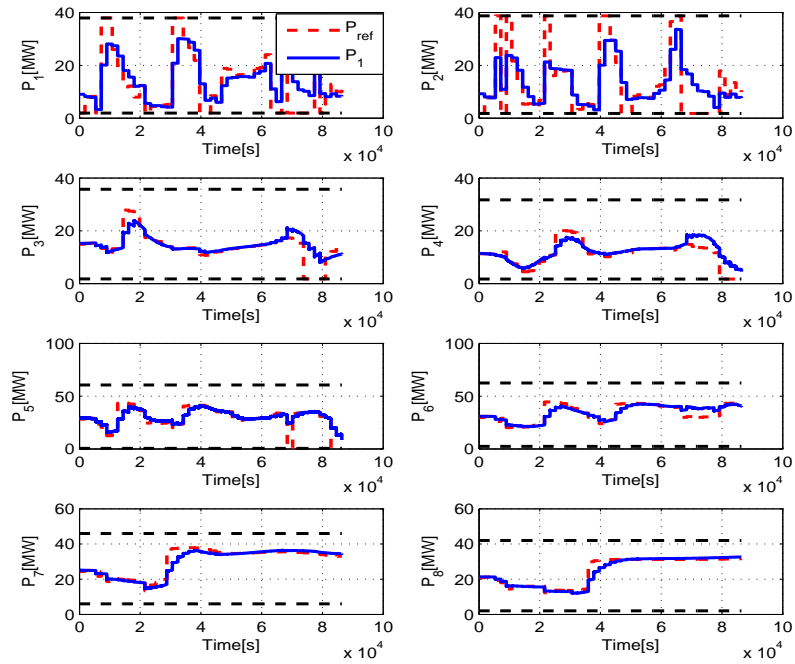
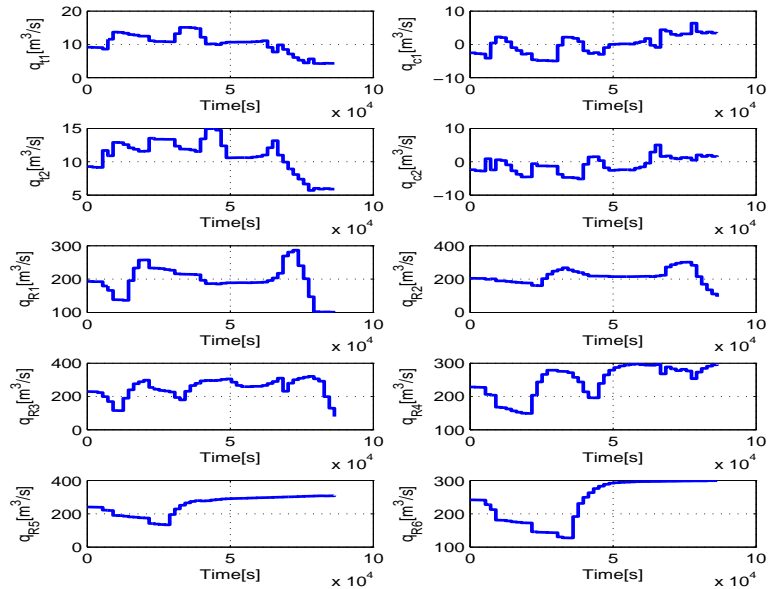Figure 5.4: Behavior of the individual powers of the HPV.

Figure 5.5: Control actions

5.4   Performance analysis

In this section, the performance analysis of the proposed control scheme are presented. The indices and values are:

- Mean absolute tracking error: $1.5191MW$

- Mean square tracking error: $3.2119MW^2$

- Power reference tracking index (economic index 1): 2397

- Power reference tracking index (economic index 2): 1416

- Constraint violation: some of the levels violate the restriction at certain instants, but they do it in a very short time, the total time where there is constrain violation was calculated adding the period of time where the levels violate the restriction, the total time is 177 minutes (almost 3 hours).

- Number of data packets transmitted: since the controller is a hierarchical structure with decentralized local controller, few information is transmitted. The coordinator transmits the values of the output reference and input variable bias for each subsystem. So the coordinator transmit two vectors to each subsystems of size 12×1 for input variables bias and 17×1 for the reference of the output variables.

Chapter 6

Game Theory Based Distributed Model Predictive Controller

6.1   Game formulation of distributed model predictive control

Let us first introduce some notation used in this section. Let Ω_i denote the feasible set of control actions for subsystem i, $i = 1, \ldots, M$, defined as the Cartesian product $\Omega_i = \Pi_{j=0}^{N_u} \Lambda_i$, where Λ_i is the feasible set for the control actions $u_i(k+j)$, for $j = 0, \ldots, N_u$ determined by the physical and operational limits of subsystem i, with N_u being the control horizon. Let $\widetilde{u}(k) = [\widetilde{u}_1^T(k), \ldots, \widetilde{u}_M^T(k)]^T$, where $\widetilde{u}_i(k) = [u_i^T(k), \ldots, u_i^T(k+N_u)]^T$ for $i = 1, \ldots, M$. Let $\phi_i(\widetilde{u}(k); x(k))$ denote the cost function for subsystem i, $i = 1, \ldots, M$, where the notation $(\widetilde{u}(k); x(k))$ indicates that the function ϕ_i depends on $\widetilde{u}(k)$ and $x(k)$ is a parameter whose time evolution is given by the linear state update equation

$$x(k+1) = Ax(k) + Bu(k)$$

where A and B are obtained by linearizing the model describing the behavior of the whole system [24]. For the sake of simplicity of notation we will not indicate the dependence of ϕ_i on $x(k)$ explicitly in the remainder of this paper and thus write $\phi_i(\widetilde{u}(k))$ instead $\phi_i(\widetilde{u}(k); x(k))$.

Assume that $0 \in \Lambda_i$ for $i = 1, \ldots, M$. Assume that Λ_i is closed, convex, and independent of k for $i = 1, \ldots, M$. Assume that the subsystems are able to "bargain"in order to achieve a common goal: to maintain both the local and the whole system performance by driving the states of the system to their reference values. Let χ denotes the constrained stabilizable set, i.e., χ is the set of all initial states x that can be steered to the origin by applying a sequence of admissible control actions. Assume that the initial system state vector $x(k) \in \chi$ (this assumption also is made in [24, 25]).

A game is defined as the tuple $(N, \{\Omega_i\}_{i \in N}, \{\phi_i\}_{i \in N})$, where $N = \{1, \ldots, M\}$ is the set of players, Ω_i is a finite set of possible actions of player i, and $\phi_i : \Omega_1 \times \ldots \times \Omega_M \longrightarrow \mathbb{R}$ is the payoff function of the ith player [1]. So, a DMPC problem can be defined as a tuple $G = (N, \{\Omega_i\}_{i \in N}, \{\phi_i\}_{i \in N})$, where $N = \{1, \ldots, M\}$ is the set of subsystems, Ω_i is the non-empty set of feasible control actions for subsystem i, and $\phi_i : \Omega_1 \times \ldots \times \Omega_M \longrightarrow \mathbb{R}$, where ϕ_i is the cost function of the i-th subsystem. Hence, a DMPC problem is a game in which the players are the subsystems, the actions are the control inputs, and the payoff of each subsystem is given by the value of its cost function.

Since it is assumed that the subsystems are able to "bargain"in order to achieve a common goal, the game G can be analyzed as a bargaining game following the Nash theories about such games. A bargaining game is a situation involving a set of players who have the opportunity to collaborate for mutual benefit by an agreement on a joint plan of action [10, 11]. If an agreement is not possible,

the players carry out an alternative plan which is determined by the information locally available. The benefit perceived by the player when an agreement is not possible is called disagreement point. Mathematically, a bargaining game is defined as follows [16]:

Definition 1 *A bargaining game for N is a pair (S,d) where:*

1. *S is a nonempty closed subset of $\mathbb{R}^M$ (Closedness of the feasible set S is required for mathematical convenience.).*

2. *$d \in \text{int}(S)$, d being the disagreement point.*

3. *$\zeta_i(S) := \max\{\phi_i \colon (\phi_i)_{i \in N} \in S\}$ exists for every $i \in N$.*

Here $\phi_i \colon \mathbb{R}^M \longrightarrow \mathbb{R}$ denotes the profit function of player i for $i = 1,\ldots,M$, S denotes the feasible set of profit functions, and $\zeta_i(S)$ denotes the utopia point of subsystem i for $i = 1,\ldots,M$. Moreover, if the feasible set S is convex then the bargaining game (S,d) is called a convex bargaining game.

Note that Definition 1 is formulated in a static context. Then, an extension of the original definition of bargaining games is required in order to analyze a DMPC problem as a game.

Let a discrete-time dynamic bargaining game refers to a situation where at each time step a static bargaining game (S,d) is solved depending on the dynamic evolution of the decision environment, with dynamic evolution determined by the state vector $x(k) \in \mathbb{R}^n$ and the input vector $u(k) \in \mathbb{R}^m$, with $x(k) \in \mathbb{X}$ and $u(k) \in \mathbb{U}$, $\mathbb{X}$ and $\mathbb{U}$ being the feasible sets for $x(k)$ and $u(k)$ respectively. In this game, we assume that the feasible set and/or the disagreement point can change with time. Mathematically, a discrete-time dynamic bargaining game is defined as follows:

Definition 2 *Discrete-time dynamic bargaining game:*
A discrete-time dynamic bargaining game for N is a sequence of pairs $\{(S(0),\eta(0)),(S(1),\eta(1)),\ldots\}$, denoted by $\{(\Theta(k),\eta(k))\}_{k=0}^{\infty}$ ($\eta(k)$ being the disagreement point at time step k), where:

1. *$\Theta(k)$ is a nonempty closed subset of $\mathbb{R}^M$, for $k = 1,2,3,\ldots$*

2. *$\eta(k) \in \text{int}(\Theta(k))$ for $k = 1,2,3,\ldots$, $\eta(k)$ being the disagreement point.*

3. *$\zeta_i(\Theta(k)) := \max\{\phi_i(k) \colon (\phi_i(k))_{i \in N} \in \Theta(k)\}$ exists for every $i \in N$ at each time step k, i.e., for $k = 1,2,3,\ldots$*

4. *There exists functions $f_i \in \mathbb{R}^{n_i}, g_i \in \mathbb{R}, h_i \in \mathbb{R}$, $i = 1,\ldots,M$, determining the dynamic evolution of the decision environment, the feasible set, and the disagreement point of player i such that*

$$x_i(k+1) = f_i(x(k),u(k))$$
$$\Theta_i(k+1) = g_i(x(k),u(k),\Theta(k))$$
$$\eta_i(k+1) = h_i(x(k),u(k),\eta(k))$$

with $x_i(k) \in \mathbb{X}_i$, $\mathbb{X}_i \subset \mathbb{X}$.

5. *There exists a profit function $\phi(x(k),u(k)) \in \mathbb{R}^M$ such that $\phi(x(k),u(k)) \in \Theta(k)$.*

If g_i is a convex function for $i = 1\ldots,M$, then $\Theta(k)$ is convex and the game $\{(\Theta(k),\eta(k))\}_{k=0}^{\infty}$ is a convex discrete-time bargaining game.

Let $\Upsilon(k) := \{(\phi_1(\widetilde{u}(k)), \ldots, \phi_M(\widetilde{u}(k)))\colon \widetilde{u}_i(k) \in \Omega_i, \ \forall i \in N\}$ denote the feasible set of the DMPC problem. Since Ω_i is time-invariant for $i = 1, \ldots, M$ the feasible set $\Upsilon(k)$ is also time invariant, i.e., $\Upsilon(1) = \Upsilon(2) = \ldots = \Upsilon$. Let us define the disagreement point $\eta(k)$ as $\eta(k) := (\eta_1(k), \ldots, \eta_M((k))$, where

$$\eta_i(k+1) = \begin{cases} \eta_i(k) - \alpha|\eta_i(k) - \phi_i(\widetilde{u}(k))|, & \eta_i(k) > \phi_i(\widetilde{u}(k)) \\ \eta_i(k) + |\phi_i(\widetilde{u}(k)) - \eta_i(k)|, & \eta_i(k) < \phi_i(\widetilde{u}(k)) \end{cases}$$

$\forall i \in N$, with $0 < \alpha \leqslant 1$. Such definition of the disagreement point is based on the negotiation model proposed by [11]. Let the utopia point $\zeta_i(\Upsilon) := \min\{\phi_i(\widetilde{u}(k))\colon \phi_i(\widetilde{u}(k)) \in \Upsilon\}$ exist for every $i \in N$. Then, the DMPC problem can be defined as a discrete-time dynamic bargaining game $\{(\Upsilon, \eta(k))\}_{k=0}^{\infty}$. Note that in $\{(\Upsilon, \eta(k))\}_{k=0}^{\infty}$ only the disagreement point depends of the time step k, and that $\zeta_i(\Upsilon)$ is redefined. Moreover, since Λ_i is assumed closed and convex for $i = 1, \ldots, M$, Ω_i also is closed and convex for $i = 1, \ldots, M$. Thus, if $\phi_i(\widetilde{u}(k))$ is a continuous convex function with respect to $\widetilde{u}(k)$, then the feasible set Υ is closed and convex. Since Υ is time-invariant $\{(\Upsilon, \eta(k))\}_{k=0}^{\infty}$ is a bargaining game with closed and convex feasible set.

In order to derive a solution for a bargaining game an axiomatic approach was proposed in [11]. Such a characterization is based on the symmetry of the bargaining game. A bargaining game (S, d) is called symmetric if $d_1 = d_2 = \ldots = d_M$, and for every $\phi \in S$ any point $\widetilde{\phi} \in \mathbb{R}^M$ arising from ϕ by performing some permutation of its coordinates is also in S. If a bargaining game (S, d) does not satisfy these conditions, then it is called a nonsymmetric bargaining game. For discrete-time dynamic bargaining games, if $\eta_1(k) = \ldots = \eta_M(k)$ for $k = 0, 1, 2, \ldots$, and for every $\phi(k) \in \Theta(k)$, any point $\widetilde{\phi}(k) \in \mathbb{R}^M$ arising from $\phi(k)$ by performing some permutation of its coordinates is also inside $\Theta(k)$ for $k = 0, 1, 2, \ldots$, the game $\{(\Theta(k), \eta(k))\}_{k=0}^{\infty}$ is called symmetric. These conditions are satisfied when $f_i(x(k), u(k)) = f_j(x(k), u(k))$, $g_i(x(k), u(k)) = g_j(x(k), u(k))$, $h_i(x(k), u(k)) = h_j(x(k), u(k))$, and $\mathbb{X}_i = \mathbb{X}_j$ for all $i, j \in N$. However, the symmetry conditions for discrete-time dynamic bargaining games are heavily restrictive in real DMPC applications. Then, in general a DMPC game $\{(\Upsilon, \eta(k))\}_{k=0}^{\infty}$ is nonsymmetric.

Let $\mathbb{R}_{++}^N := \{\phi(\alpha \in \mathbb{R}^N\colon \alpha_i > 0,$ for all $i \in N\}$. Let H denote a wighted hierarchy of N, i.e., H is an ordered $(l+1)$-tuple $H = \langle N^1, \ldots, N^l, w \rangle$, where $(N^1, \ldots, N^l)$ is a partition of N (i.e., the sets N^j, $j = 1, \ldots, l$ are pairwise disjoint nonempty sets whose union equals to N), and $w \in \mathbb{R}_{++}^N$ with $\sum_{i \in N^j} w_i = 1$ for every $j = 1, \ldots, l$ [16]. Let $P(T) := \{\alpha \in T\colon$ there is no $\beta \in T$ with $\beta \leqslant \alpha, \ \beta \neq \alpha\}$ denote the Pareto optimal subset of T. Let $L_+(T, \gamma) := \{i \in L\colon$ there exists $\alpha \in T$ with $\alpha_i < \gamma_i\}$. Let $\arg\max\{f(\alpha)\colon \alpha \in T\} := \{\alpha \in T\colon f(\alpha) \geq f(\beta)$ for all $\beta \in T\}$. Then the non-symmetric Nash bargaining solution of a game $\{(\Theta(k), \eta(k))\}_{k=0}^{\infty}$ at time step k is defined as follows [16, Definition 2.14].

Definition 3 *Non-symmetric Bargaining Solution:*
Let $H = \langle N^1, \ldots, N^l, w \rangle$ be a weighted hierarchy of N. Let Υ^j, $j = 0, \ldots, l$ denote the feasible set for

the partition N^j. Then, the sets Θ^j are defined as follows:

$$\Theta^0 := \{\phi(\widetilde{u}(k)) \in \mathbb{R}^N : \phi(\widetilde{u}(k)) \in P(\Theta), \phi(\widetilde{u}(k)) \leqslant \eta(k)\}$$

$$\Theta^1 := \arg\max\{\Pi(\eta_i(k) - \phi_i(\widetilde{u}(k)))^{w_i} :$$
$$i \in N^1, \ \phi(\widetilde{u}(k)) \in \Theta^0\}$$

$$\Theta^2 := \begin{cases} \arg\max\{\Pi(\eta_i(k) - \phi_i(\widetilde{u}(k)))^{w_i} : \\ \qquad i \in N_+^2(\Theta^1, \eta(k)), \ \phi(\widetilde{u}(k)) \in \Theta^1\} \\ \qquad if \ N_+^2(\Theta^1, \eta(k)) \neq \varnothing \\ \Theta^1 \quad otherwise \end{cases}$$

$$\vdots$$

$$\Theta^l := \begin{cases} \arg\max\{\Pi(\eta_i(k) - \phi_i(\widetilde{u}(k)))^{w_i} : \\ \qquad i \in N_+^l(\Theta^{l-1}, \eta(k)), \ \phi(\widetilde{u}(k)) \in \Theta^{l-1}\} \\ \qquad if \ N_+^l(\Theta^{l-1}, d) \neq \varnothing \\ \Theta^{l-1} \quad otherwise \end{cases}$$

Let $H = \langle N, w \rangle$. Then, according to Definition 3 the nonsymmetric bargaining solution of a DMPC game $\{(\Upsilon, \eta(k))\}_{k=0}^{\infty}$ at time step k can be computed in a centralized way as a solution of the maximization problem (6.1).

$$\max_{\widetilde{u}(k)} \Pi_{i=1}^M (\eta_i(k) - \phi_i(\widetilde{u}(k)))^{w_i}$$

Subject to:

$$\eta_i(k) > \phi_i(\widetilde{u}(k))$$
$$\widetilde{u}(k) \in \Omega \tag{6.1}$$

Maximization problem (6.1) can be written equivalently as (6.2).

$$\max_{\widetilde{u}(k)} \sum_{i=1}^M w_i \log(\eta_i(k) - \phi_i(\widetilde{u}(k)))$$

Subject to:

$$\eta_i(k) > \phi_i(\widetilde{u}(k))$$
$$\widetilde{u}(k) \in \Omega \tag{6.2}$$

Let $\sigma_i(\widetilde{u}_i(k), \widetilde{u}_{-i}(k)) = \phi_i(\widetilde{u}(k))$ for $i = 1, \dots, M$, where $\widetilde{u}_{-i}(k) = [\widetilde{u}_1^T(k), \dots, \widetilde{u}_{i-1}^T(k), \widetilde{u}_{i+1}^T(k), \dots, \widetilde{u}_M^T(k)]$. Then, maximization problem (6.1) can be solved in a distributed way by locally solving the system-wide control problem (6.3).

$$\max_{\widetilde{u}_i(k)} \sum_{r=1}^M w_r \log(\eta_r(k) - \sigma_r(\widetilde{u}_i(k), \widetilde{u}_{-i}(k)))$$

Subject to:

$$\eta_r(k) > \sigma_r(\widetilde{u}_i(k), \widetilde{u}_{-i}(k))$$
$$\widetilde{u}_i(k) \in \Omega_i \tag{6.3}$$

Note that maximization problem (6.3) is equivalent to maximization problem (6.2), considering fixed $\widetilde{u}_{-i}(k)$ and optimizing only in the direction of $\widetilde{u}_i(k)$. This formulation allows to each subsystem take into account the effect of its decisions in the behavior of the whole system and to promote the cooperation among subsystems. The proposed algorithm for solving (6.3) is given in [2, 23]. In the next section a DMPC control based on game theory is presented for an HPV.

6.2 Game theory based control of a hydro-power valley

With the purpose of designing a MPC for the HPV depicted in before, we consider the power tracking scenario proposed in [20]. In this scenario, power output of the system should follow a given reference while keeping the water levels in the lakes and at the dams as constant as possible. So, the global cost function considered for the DMPC is composed by two terms: the first term penalizes the 1-norm of the power tracking error, and the second term penalizes the 2-norm of the deviations of the levels in the lakes and in the dams of their steady state values.

Let T_s denote the sample time. By linearizing and discretizing the HPV model yields:

$$
\begin{aligned}
x(k+1) &= A_d x(k) + B_d u(k) \\
y(k) &= C_d x(k) + D_d u(k)
\end{aligned}
\tag{6.4}
$$

where A_d, B_d, C_d, D_d are the matrices resulting of the linearization of the HPV model, and $y(k) = [p(k), h_D^T(k)]^T$, with $h_D(k) = [h_{D1N_x}, h_{D2N_x}, h_{D3N_x}, h_{D4N_x}, h_{D5N_x}, h_{D6N_x}]$ the levels at the dams (only the levels in the last element of the spatial discretization of the reaches is considered to regulate the levels of the reaches). Note that the power produced by the HPV is piecewise defined respect to $u(k)$ due to the turbine-pump elements. In order to overcome this issue in the linearization, constants k_{des1}, k_{des2} were introduced, virtual inputs $\bar{u}_1(k) \in [-q_{C1\text{pump}}, q_{C1\text{turb}}], \bar{u}_2(k) \in [-q_{C2\text{pump}}, q_{C2\text{turb}}]$ were considered, and a gain compensation

$$
u_p(k) = \left\{
\begin{array}{l}
\frac{k_{desp}}{k_{tCp}} \bar{u}_p(k) \text{ if } \bar{u}_p(k) \geq 0 \\
\frac{k_{desp}}{k_{pCp}} \bar{u}_p(k) \text{ if } \bar{u}_p(k) < 0
\end{array}
\right.
$$

was proposed, where $q_{C1\text{pump}}, q_{C2\text{pump}}, q_{C1\text{turb}}, q_{C2\text{turb}}$ are the maximum pumped flows and maximum turbine flows for the turbine-pump elements C_1, C_2 respectively, $p = 1, 2$ (the values of $q_{C1\text{pump}}, q_{C2\text{pump}}, q_{C1\text{turb}}, q_{C2\text{turb}}$ are given in [20]).

Moreover, note that the dimension of the matrices A_d, B_d depends of N_x which in general is large in order to adequately represent the HPV dynamics. Then a centralized MPC maybe is not suitable and a DMPC is required.

Let N_p be the prediction horizon. Then optimization problem of the power tracking problem can be written as

$$
\min_{\widetilde{u}(k)} \gamma |\widetilde{p}_r(k) - \widetilde{y}_p(\widetilde{u}(k))| + \widetilde{u}^T(k) Q_{uu} \widetilde{u}(k) + h_D^T(k) Q_{ux} \widetilde{u}(k) + h_D^T(k) Q_{xx} h_D^T(k)
$$

Subject to:

$$
\begin{aligned}
& \widetilde{u}(k) \in \Omega \\
& u(k+\nu) = u(k+N_u), \ \forall N_u < \nu < N_p - 1
\end{aligned}
\tag{6.5}
$$

where $\widetilde{p}_r(k) = [p_r(k), \ldots, p_r(k+N_p)]$, $\widetilde{y}_p(\widetilde{u}(k)) = [p(x(k), u(k)), \ldots, p(x(k), u(k+N_p-1))]$, $Q_{uu} = \bar{B}_d^T \bar{Q} \bar{B}_d$, $Q_{ux} = x^T(k) \bar{A}_d^T \bar{Q} \bar{B}_d$, $Q_{xx} = \bar{A}_d^T \bar{Q} \bar{A}_d$, and Ω is the feasible set composed by the input constraints and the mapping using (6.4) of the state constraints to input constraints, with $\bar{A}_d, \bar{B}_d$ the resulting matrices from the prediction of $h_D(k)$ along N_p, and $\bar{Q}$ the Q block diagonal matrix resulting form the transformation of the power tracking problem into (6.5). From [20], it is possible to divide the HPV under study into 8 subsystems:

† Subsystem 1: lakes L_1 and L_2, turbine T_1, and turbine-pump C_1.

† Subsystem 2: lake L_3, turbine T_2, and turbine-pump C_2.

† Subsystems 3-8: reaches R_1 to R_6 respectively.

Let $\sigma_i(\widetilde{u}_i(k), \widetilde{u}_{-i}(k)$ be the local cost function of each subsystem, $\sigma_i(\widetilde{u}_i(k), \widetilde{u}_{-i}(k)$ defined as

$$\sigma_i(\widetilde{u}_i(k), \widetilde{u}_{-i}(k)) = \gamma|\widetilde{p}_r(k) - \widetilde{y}_p(\widetilde{u}_i(k), \widetilde{u}_{-i}(k))|$$
$$+ [\widetilde{u}_i^T(k), \widetilde{u}_{-i}^T(k)]\bar{H}_i[\widetilde{u}_i^T(k), \widetilde{u}_{-i}^T(k)]^T + 2\bar{F}_i[\widetilde{u}_i^T(k), \widetilde{u}_{-i}^T(k)]^T$$

where $\bar{H}_i, \bar{F}_i$ are the resulting matrices of the permutation of the rows and columns of Q_{uu} and Q_{ux} respectively. (the state dependence of $\sigma_i(\cdot)$ was omitted for notational convenience), and $\gamma \in \mathbb{R}$ a constant weight (the term $h_D^T(k)Q_{xx}h_D^T(k)$ was omitted because it is constant respect to the decision variables, then it does not affect the result of the optimization). From [20], the state and input constraints are time independent and only establishes lower and upper boundaries to the states and inputs. So, they are independent for each subsystems, i.e., there is not coupled constraints. Then, for the control of the HPV we have a game $G_{HPV} = \{N, \{\sigma_i(\widetilde{u}_i(k), \widetilde{u}_{-i}(k))\}_{i \in N}, \{\Omega_i\}_{i \in N}\}$, with $N = \{1, \ldots, 8\}$, in which all subsystems have the same goal: to minimize the power tracking error keeping the levels in the lakes and at the dams as close as possible to their steady state values. Hence, the game G_{HPV} can be analyzed and solved as a discrete-time dynamic bargaining game $\{(\Upsilon, \eta(k))\}_{k=0}^{\infty}$, with $\eta(k)$ defined as in Section 6.1. Then, according to (6.3) the distributed solution of the game G_{HPV} is given by the solution of the local optimization problems (6.6).

$$\max_{\widetilde{u}_i(k)} \sum_{r=1}^{8} w_r \log(\eta_r(k) - \sigma_r(\widetilde{u}_i(k), \widetilde{u}_{-i}(k)))$$

Subject to: (6.6)

$$\eta_r(k) > \sigma_r(\widetilde{u}_i(k), \widetilde{u}_{-i}(k))$$
$$\widetilde{u}_i(k) \in \Omega_i$$

Since the power produced by the HPV at time step k is equal to the sum of the powers generated by all subsystems, and assuming that each subsystem communicates the value of the states and inputs to the remaining subsystems, each subsystem is able to compute the power produced by the other subsystems. Hence, the term $\gamma|\widetilde{p}_r(k) - \widetilde{y}_p(\widetilde{u}_i(k), \widetilde{u}_{-i}(k))|$ is reduced to compute the power contribution of subsystem i given the power produced by the remaining subsystems. In the next section the simulation results are presented.

6.3   Simulation results

Based on the formulation presented in Section 6.2, a closed-loop simulation of the HPV described in [20] was performed along 24 hours (simulation time). In this simulation, $k_{des1} = \frac{3}{4}(k_{tC1} + k_{pC1})$, $k_{des2} = \frac{3}{4}(k_{tC2} + k_{pC2})$, $T_s = 1800s$ (30 minutes), $N_p = 48$ (corresponding to a day), $N_u = 32$, $w_{1,2} = \frac{0.4}{2}$, $w_{3-8} = \frac{0.6}{6}$ (the weights of subsystems 1 to 8), $\eta_i(0) = 1 \times 10^5$ (the initial disagreement point of subsystems 1 to 8), $\gamma = 50$, $Q = I$ (I being the identity matrix), and the lower and upper values of the inputs and the states, and the parameter of the HPV model were taken as proposed in [20].

Figure 6.1 shows the comparison between the power produced by the HPV and the power reference, when the proposed DMPC scheme computed the inputs of each subsystem. In this Figure was shown that the power produced by the HPV followed the power reference, satisfying one of the objectives proposed for the control scheme. However, there was an oscillation at the beginning of the experiment due to the transient generated by the change of the power from 175 MW (the equilibrium power) to the initial required power 150 MW.
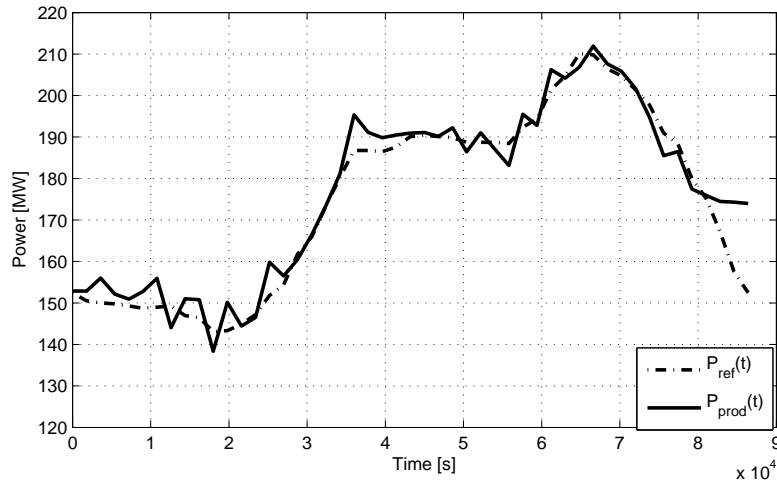
Figure 6.1: Comparison between the power produced by the HPV with the power reference, when the proposed game-theory-based DMPC is used for computing the inputs of the subsystems

In order to maintain some power demand, the levels of the reaches and the lakes had to be modified. In Figure 6.2 the behavior of the levels is presented. At the beginning of the simulation the lakes increased their levels due to the reduction of the power form the equilibrium point to the set point (see first panel of Figure 6.2). When the required power was increased the lakes reached constant levels of water, achieving one of the system objectives. During the whole simulation the reaches maintained their levels as constant as possible (see second panel of Figure 6.2). If it is considered that the reaches also can be used for maritime traffic, maintaining constant their levels guarantees it. This condition was considered in the selection of the weights, by giving more importance to the reaches compared with the lakes; it is evidenced with $\sum_{i=3}^{8} w_i > \sum_{i=1}^{2} w_i$.

The excursions of the levels of the lakes were associated with the behavior of the control inputs (see Figure 6.3). Even though the control inputs remained inside the range defined by the constraints, the control actions of subsystems 1 and 2 had higher variations than the control actions of the remaining subsystems, with respect to their local capability. This produced lager changes in the levels of the lakes than in the levels of the reaches. Recall that subsystems 3 to 8 were power plants and subsystems 1 and 2 were ducts equipped with turbines and turbine-pump elements, with less capability to produce electric power than the power plants.

Finally, in Figure 6.4 the evolution of the disagreement points is presented. In this Figure, the disagreement started at the same point but as they were evolving each subsystem had its own value indicating the non-symmetry of the game G_{HPV}. Figure 6.5 shows a zoom between 3.5×10^4 and 8×10^4, note that all the disagreement points decreased with low frequency oscillations. Such oscillations were associated to the decision process of each subsystem.

6.4   Performance analysis

In order to evaluate the performance of the proposed control scheme, the following indices were proposed:
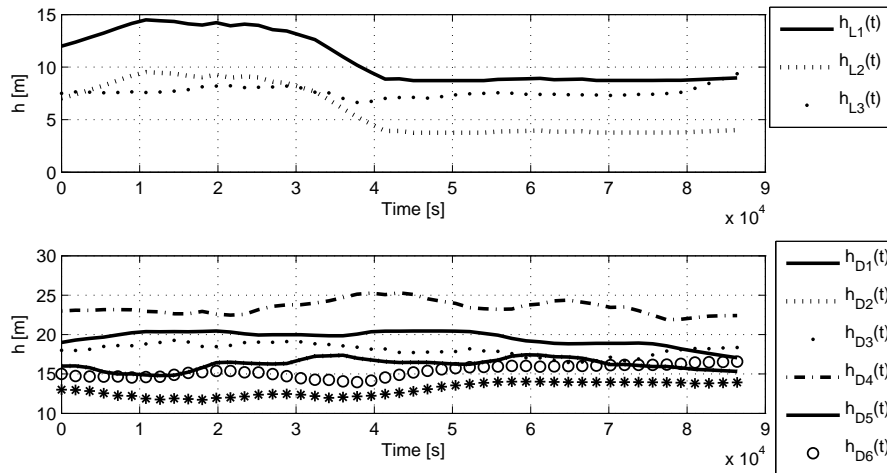
Figure 6.2: Behavior of the levels in the lakes (first panel) and the levels at the dams (second panel) of the HPV. In both panels the levels are inside the values defined by the constraints, although the levels of the lakes (first panel) present large excursions before remaining constant, while the levels of the reaches remains as constant as possible.

- Mean absolute power tracking error: $3.1925MW$

- Mean square tracking error: $19.0696MW^2$

- Power tracking index (economic index 1): 4722

- Power tracking index (economic index 2): 4066

- Constraint violation: with the proposed control scheme there are two constraints which are not respected. These constraints are those regarding the levels of the lakes L_1 and L_2. The constraints regarding lake L_2 violated almost all the time. The deviation is less than 1m in both directions. The constraints regarding lake L_2 are violated from 45000s until the end of the simulation. The deviation is 1m below the minimum allowed level value.

- Number of data packets transmitted: in the proposed control scheme each subsystem has to transmit its disagreement point, the value of the local states, and the current control action. In the case of the HPV: the vector of local inputs is a vector with dimension 32×1 for subsystems 3 to 8 and 64×1 for subsystems 1 and 2, the vector of states has a dimension 40×1 for subsystems 3 to 8, 1×1 for subsystem 2, and 2×1 for subsystem 1, and the disagreement point has a dimension 1×1 for all subsystems (all the vectors are double precision vectors).
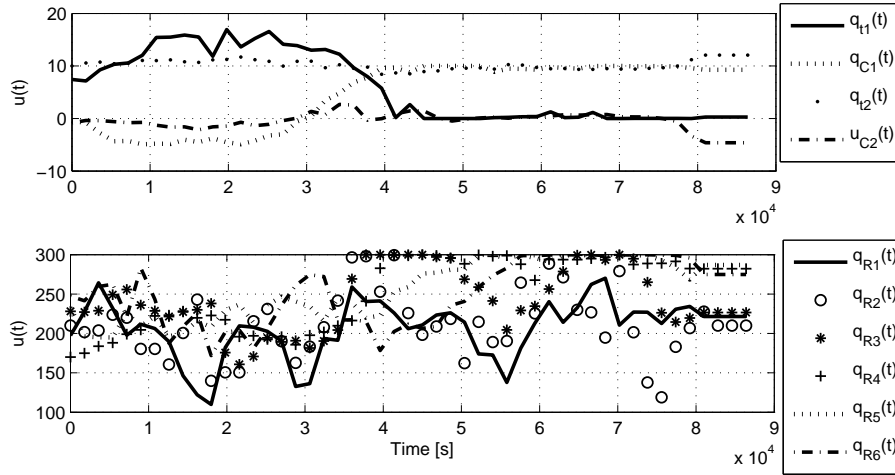
Figure 6.3: Control actions applied to the subsystems. In the first panel the behavior of the control actions applied to subsystems 1 and 2 is presented. In the second panel the behavior of the control actions applied to subsystems 3 to 8 is presented. In both panels the control actions remains inside the range defined by the constraints of the control inputs.
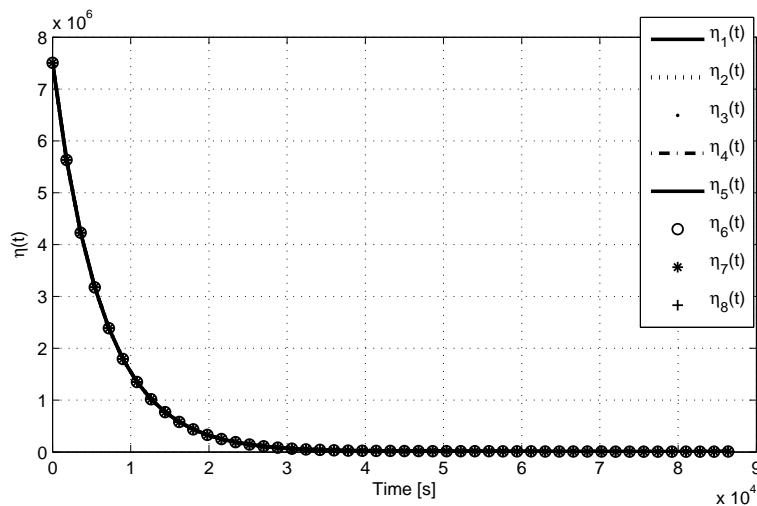

Figure 6.4: Behavior of the disagreement points at the full simulation. Overall evolution
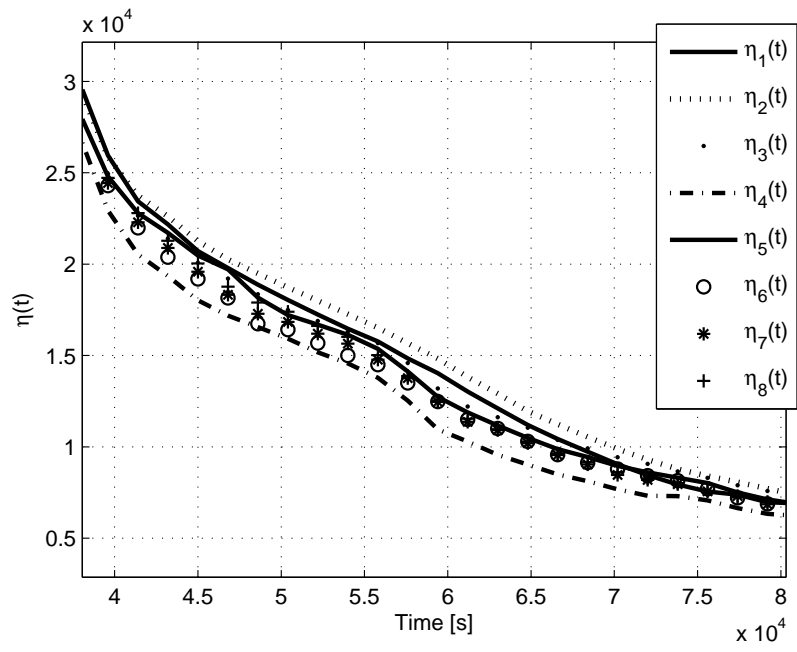
Figure 6.5: Behavior of the disagreement points at the full simulation. Detailed view that allows to evidence the non-symmetry of the game.

Chapter 7

Distributed MPC Based on Agent Negotiation

7.1 Introduction

In this chapter we use the distributed MPC scheme based on agent negotiation presented in [9]. This control scheme is tailored for distributed linear systems composed of subsystems coupled in the inputs. We assume that the subsystems are controlled by a set of independent agents that are able to communicate and that each agent has access only to the model and the state of one of the subsystems. These assumptions imply that before the agents take a cooperative decision, they must negotiate. At each time sample, following a protocol that will be explained later in this section, agents make proposals to improve an initial feasible solution on behalf of their local cost function, state and model. These proposals are accepted if the global cost improves the cost corresponding to the current solution. At this point it is convenient to point out that it is possible to guarantee the stability properties of the proposed distributed controller as it is shown in [9]. Nevertheless, in this paper we use a slightly simplified version of the algorithm which does not guarantee stability. This version of the algorithm has successfully been applied in [27]. Finally, notice that some simplifying assumptions were made in order to adapt the system model, which is non-linear and include state coupling, to the algorithm. In particular, a linear model was used and the coupling in the state was neglected assuming that the difference of water levels remains constant during the control horizon, i.e., the power generation depends only on the value of the manipulated variables.

7.2 DMPC algorithm based on agent negotiation

The control objective of the proposed scheme is to minimize a global performance index defined as the sum of each of the local cost functions. The local cost function of agent i based on the predicted trajectories of its state and inputs is defined as

$$J_i(x_i, \{U_j\}_{j \in n_i}) = \sum_{k=0}^{N-1} L_i(x_{i,k}, \{u_{j,k}\}_{j \in n_i})$$

where $U_j = \{u_{j,k}\}$ is the future trajectory of input j, N is the prediction horizon, $L_i(\cdot)$ with $i \in M_x$ is the stage cost function defined as

$$L_i(x_i, \{u_j\}_{j \in n_i}) = (x_i - x_{r_i})^T Q_i (x_i - x_{r_i})$$
$$+ \sum_{j \in n_i} u_j^T S_{ij} u_j$$

with $Q_i > 0, S_{ij} > 0$. Note that the term x_{r_i} stands for the agent i reference.

We use the notation $x_{i,k}$ to denote the state i, k-steps in the future obtained from the initial state x_i applying the input trajectories defined by $\{U_j\}_{j \in n_i}$.

At the end of the negotiation rounds, the agents decide a set of input trajectories denoted as U^d. The first input of these trajectories is applied and the rest of the values are used to generate the initial proposal U^s for the next sampling time. Note that the last value of these trajectories is repeated so that U^s has the proper size.

We define next the proposed distributed MPC scheme:

- Step 1: Each agent p measures its current state $x_p(t)$. The agents communicate in order to obtain $U^s(t)$ from $U^d(t-1)$. The initial value for the decision control vector $U^d(t)$ is set to the value of the shifted input trajectories, that is, $U^d(t) = U^s(t)$.

- Step 2: Randomly, each agent asks the neighbors affected if they are free to evaluate a proposal (each agent can only evaluate a 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. We will use the superscript p to refer to the agent which is granted permission to make a proposal.

- Step 3: In order to make its proposal, agent p solves:

$$\{U_j^p(t)\}_{j \in n_p} = \arg \min_{\{U_j\}_{j \in n_p}} J_p(x_p, \{U_j\}_{j \in n_p})$$

$$s.t.$$
$$x_{p,k+1} = A_p x_{p,k} + \sum_{j \in n_p} B_{pj} u_{j,k}$$
$$x_{p,0} = x_i(t)$$
$$x_{p,k} \in \mathscr{X}_p, \ k = 0, \dots N$$
$$u_{j,k} \in \mathscr{U}_j, \ k = 0, \dots N-1, \ \forall j \in n_p$$
$$U_j = U_j^d(t), \ \forall j \notin n_{prop}$$

(7.1)

From the centralized point of view, the proposal at time step t of agent p is defined as

$$U^p(t) = \{U_j^p(t)\}_{j \in n_p} \uplus U^d(t)$$

where the operation $\uplus$ stands for the update of the components relatives to $\{U_j^p(t)\}_{j \in n_p}$ in $U^d(t)$.

- Step 4: Each agent i affected by the proposal evaluates the difference between the cost of the new proposal $U^p(t)$ and the cost of the current accepted proposal $U^d(t)$ as

$$\Delta J_i^p(t) = J_i(x_i(t), \{U_j^p(t)\}_{j \in n_i})$$
$$- J_i(x_i(t), \{U_j^d(t)\}_{j \in n_i})$$

This difference $\Delta J_i^p(t)$ is sent back to the agent p. 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.

- Step 5: Once agent p receives the local cost increments from each neighbor, it can evaluate the impact of its proposal $\Delta J^p(t)$, which is given by the following expression

$$\Delta J^p(t) = \sum_{i \in \bigcup_{j \in n_{prop}} m_j} \Delta J_i^p(t)$$
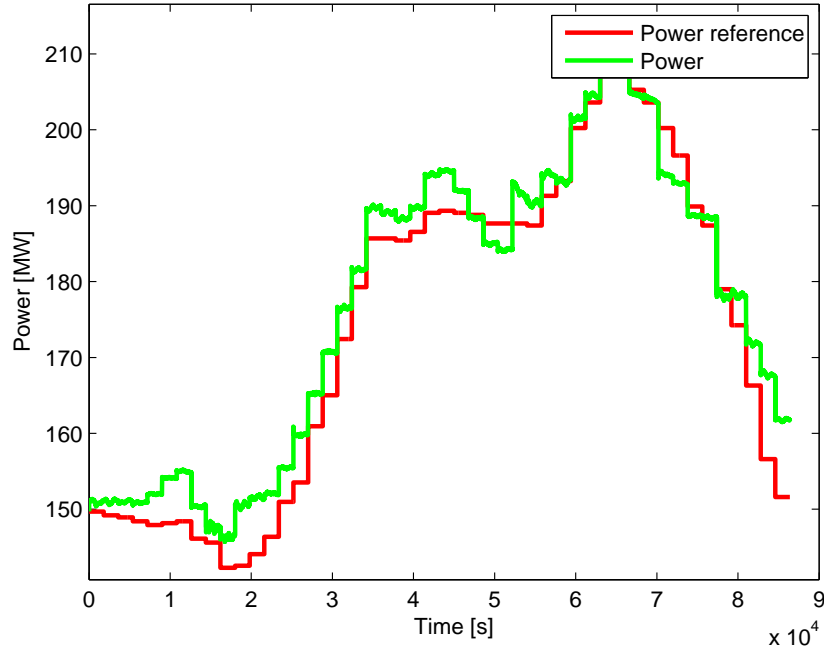
(7.2)

Figure 7.1: Power tracking by the distributed controller based on agent negotiation.

This global cost increment is used to make a cooperative decision on the future inputs trajectories. If $\Delta J^p(t)$ is negative, the agent will broadcast the update on the control actions involved in the proposal and the joint decision vector $U^d(t)$ will be updated to the value of $U^p(t)$, that is $U^d(t) = U^p(t)$. Else, is discarded.

- Step 6: The algorithm goes back to step 1 until the maximum number of proposals have been made or the sampling time ends. We denote the optimal cost corresponding to the decided inputs as

$$J(t) = \sum_{i=1}^{M_x} J_i(x_i(t), \{U_j^d(t)\}_{j \in n_i}) \tag{7.3}$$

- Step 7: The first input of each optimal sequence in $U^d(t)$ is applied and the procedure is repeated the next sampling time.

7.3 Simulation results

In the following pictures, the results of the HPV controlled in closed loop with the distributed controller based on agent negotiation are shown. In Figure 7.1 the power reference and the power generated by the system is shown. Figures 7.2 and 7.3 show respectively the evolution of the levels and the inputs of the system. Notice that the y-axes limits are adjusted to the value of the corresponding upper and lower constraints.
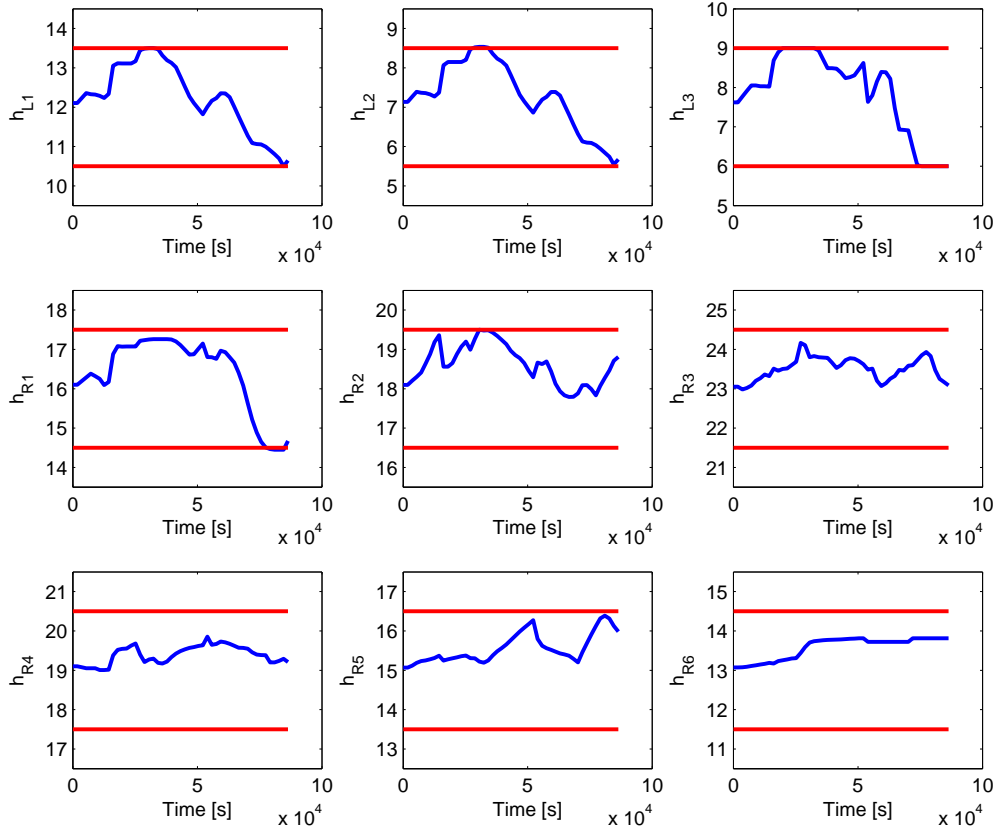
Figure 7.2: Levels by the distributed controller based on agent negotiation.

7.4 Performance analysis

In this section, we study the performance of the proposed scheme using different indices:

- Mean absolute tracking error: 3.72 MW

- Mean quadratic tracking error: 19.88 MW^2

-
$$\int_0^{86400} c(t) \left| p_r(t) - \sum_{i=1}^{8} p_i(x_i(t), u_i(t)) \right| dt = 5750$$

-
$$\int_0^{86400} c(t) \max\left(p_r(t) - \sum_{i=1}^{8} p_i(x_i(t), u_i(t)), 0 \right) dt$$
$$+ 0.5 \int_0^{86400} c(t) \max\left(\sum_{i=1}^{8} p_i(x_i(t), u_i(t) - p_r(t)), 0 \right) dt = 4194$$

- Constraint violation: there are two small violations in one of the constraints of the problem. Specifically, the constraints regarding the water level of lake 2 and the water level at reach 1 are violated . In the first case, the maximum value of the deviation with respect the constraint is 0.035 m and the violation takes place during a total time of 2 hours out of the 24 hours of the simulation. In the second case, the maximum violation is -0.05 m and the total time is again 2 hours.

- Number of data packets transmitted: the total number of proposals that are evaluated each sample time is 50. This means that each agent sends an average number of approximately 6 proposals to its neighbors. Given that these results are obtained for a control horizon N_c=10, each proposal consists of a maximum number of 30 floating point reals.
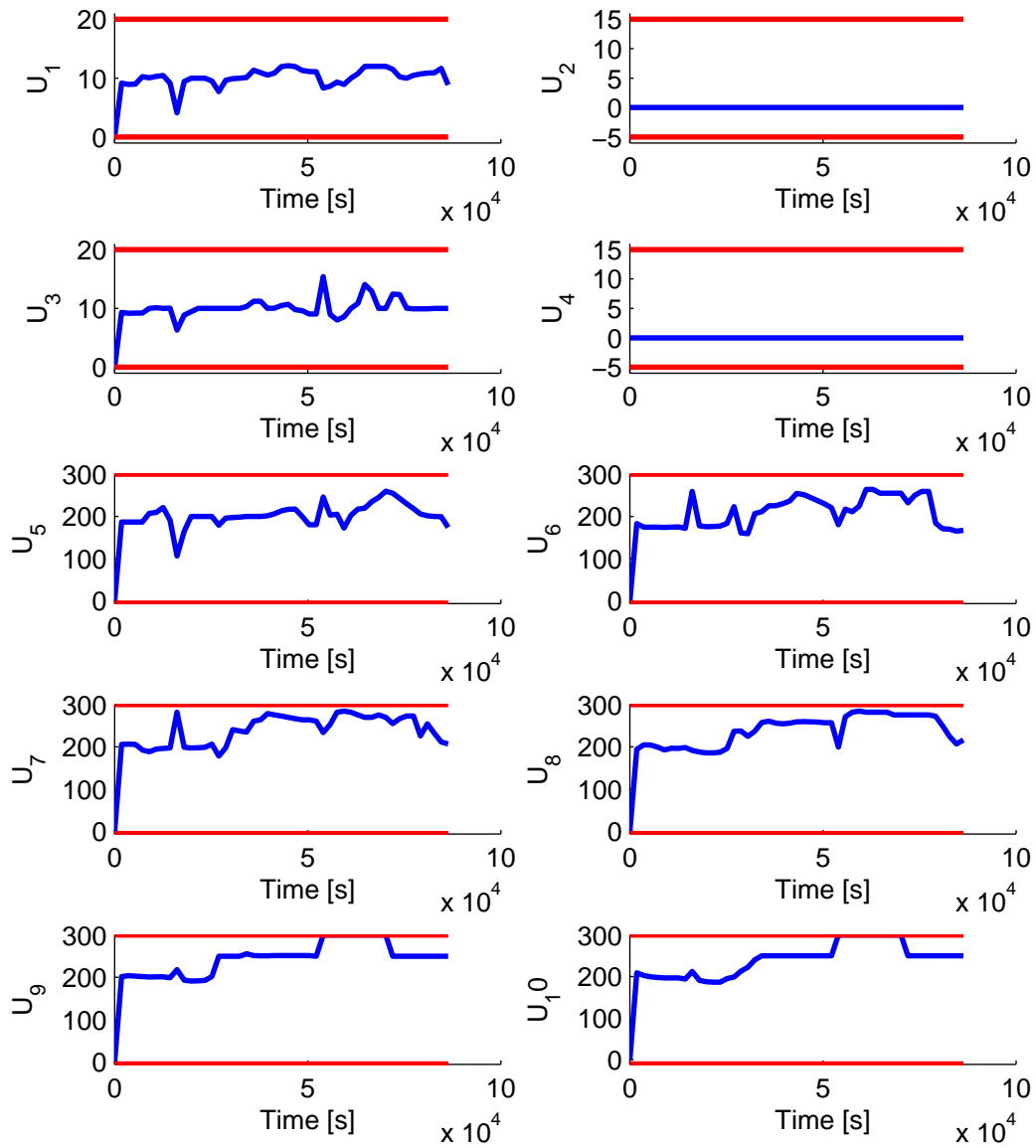
Figure 7.3: Inputs by the distributed controller based on agent negotiation.

Chapter 8

Economic Assessment and Results

This section summarizes the results of the different approaches, analyzing the control performance in the tested scenario and providing indices to compare the algorithms from a economic point of view. We also compare the new approaches with a standard decentralized MPC solution.

8.1   Decentralized MPC simulation

In addition to the results of newly developed hierarchical and distributed approaches, we also apply a decentralized model predictive control to the Hydro Power Valley benchmark, in order to provide a more complete comparison. The decentralized MPC setting is defined as follows:

- There are 8 local controllers, each of them is responsible for the control of one subsystem. Each one can only measure its own output and can only control its own manipulator(s).

- The controllers use linearized local models with double-flow technique in order to apply discrete-time linear MPC control to the local problem.

- No information exchange is allowed. The steady-state inputs and states are the only common information of the local controllers. Any subsystem interaction will be modeled by using the steady-state variables of the other models.

- The power reference tracking is separated into local tracking, with the local power references proportional to the steady-state power of the corresponding subsystem.

With the above conditions, we treat the optimal control problem (2.24) by using local MPC problems as follows:

$$\min_{x_i,u_i} \quad \int_0^{86400} \gamma |p_{r_i}(t) - p_i(x_i(t),u_i(t))|\,dt + \int_0^{86400} (x_i(t)-x_{ss,i})^T Q_i(x_i(t)-x_{ss,i})dt, \quad i=1,\ldots,8$$

where $p_{r_i}(t)$ is the local power reference. The values of $p_{r_i}(t)$ are computed such that the following condition is maintained:

$$\frac{p_{r_i}(t)}{p_r(t)} = \frac{p_{ss,i}}{p_{ss}}, \quad \forall i, \forall t \tag{8.1}$$

in which $p_{ss,i}$ and p_{ss} respectively represent the steady-state powers of the subsystem i and the whole plant.

For the decentralized simulation, we use the horizons $N = N_c = 10$, sampling time $T = 1800$s and $\gamma = 500$. The results are plotted in Figures 8.1, 8.2 and 8.3.
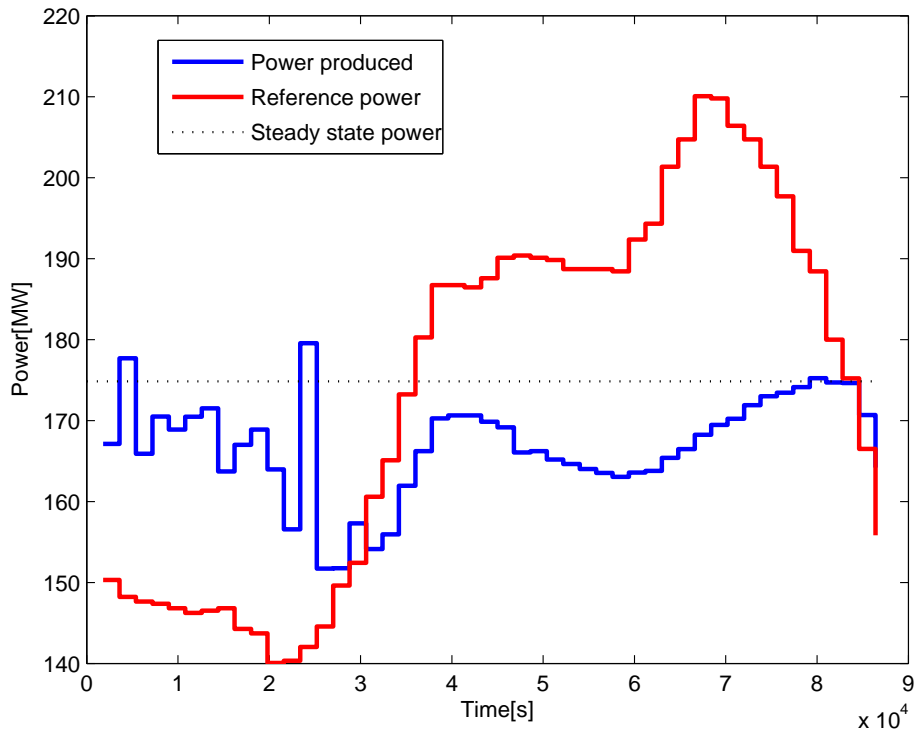
Figure 8.1: Total power reference tracking with decentralized MPC

8.2   Comparison and assessment of the results

Four quantitative indices have been used to analyze the performance of the approaches:

- Mean absolute tracking error (MAE) in *MW*

- Mean quadratic tracking error (MQE) MW^2

- Power reference tracking index 1 (J_1) in Euros: two indices will be used to assess the economic performance of the proposed scheme. In first place, an expression inspired in the index proposed for the power reference tracking scenario is used:

$$\int_0^{86400} c(t) \left| p_r(t) - \sum_{i=1}^{8} p_i(x_i(t), u_i(t)) \right| dt$$

where c(t) is the cost of the electricity at time t. Note that this expression only focuses on the economical part of the equation (2.24).

- Power reference tracking index 2 (J_2) in Euros: another option that will be used to test the
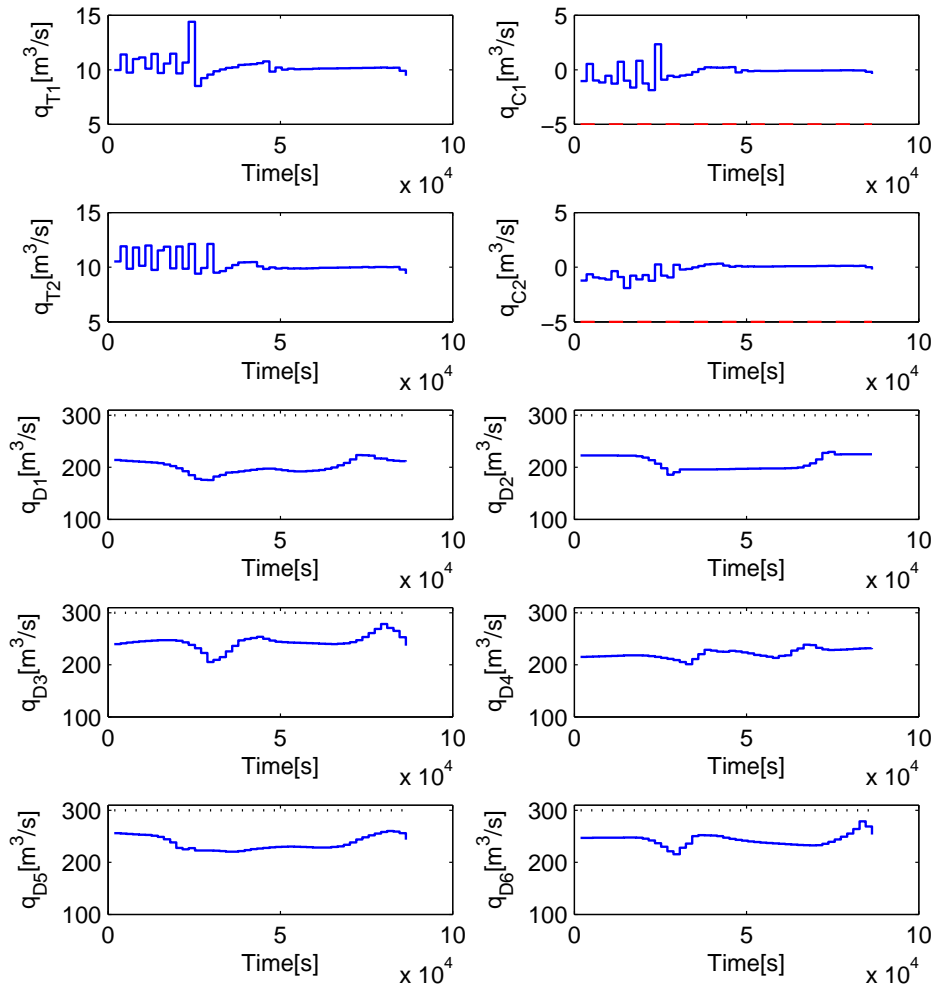
Figure 8.2: Input constraint satisfaction with decentralized MPC

economic performance of the scheme is given by the following expression

$$\int_0^{86400} c(t) max \left(p_r(t) - \sum_{i=1}^{8} p_i(x_i(t), u_i(t)), 0 \right) dt$$
$$+0.5 \int_0^{86400} c(t) max \left(\sum_{i=1}^{8} p_i(x_i(t), u_i(t)) - p_r(t), 0 \right) dt$$

In Table 8.1 the indices for each one of the approaches are shown. Notice that there are very important differences among approaches from an economic point of view.

The best results are obtained with the Distributed Multiple Shooting approach, with a nearly perfect tracking and a negligible economic cost. Good results are obtained also with Fast gradient-based
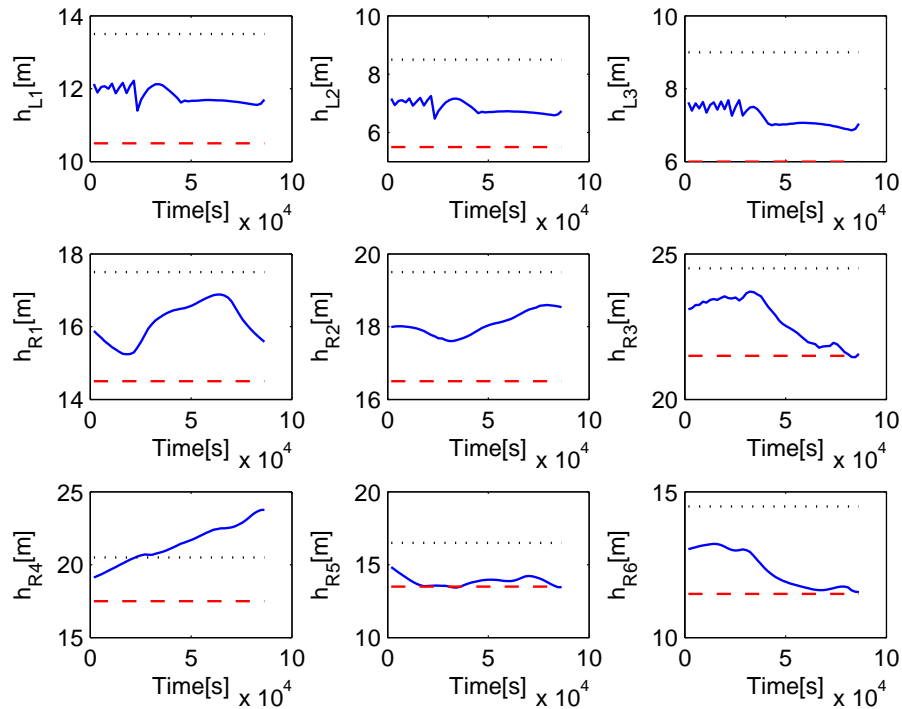
Figure 8.3: Output constraint satisfaction with decentralized MPC

DMPC and Hierarchical infinite horizon MPC. These results are understandable as the Distributed Multiple Shooting approach implements a nonlinear controller that uses a good nonlinear approximation of the centralized model, while Fast gradient-based DMPC and the Hierarchical infinite horizon MPC effectively solve the centralized linear MPC problem.

The DMPC scheme based on agent negotiation shows a poor performance in comparison with the scheme that obtained the best results. This result was expected since this particular scheme is tailored for problems in which there are strong restrictions on the amount of global information each agent has. Specifically, each agent only knows how other agents affect them and use this information in order to make proposals to the others. Hence, the final degree of cooperation is relatively low. This assumption is reasonable for systems in which there are concerns about the information the agents share, e.g.: a supply chain, or in which the composition of the overall system is not known in advance. In addition, the model used by the agents is linear while the system under test presented some important nonlinearities.

The Decentralized MPC scheme heavily suffers from the lack of information exchange, as it could hardly track the total power reference. Moreover, the decentralized MPC may cause instability, as we can see in Figure 8.3 that the water level of R_4 exceeds the upper constraint and cannot be regulated. The unfavorable results of the decentralized MPC signifies the role of information exchange in optimal control of large-scale systems.

Another important issue in HPV systems is that reach and lake levels remain between maximum and minimum values. This is to prevent the risk of flood and to guarantee a minimum ecological

Control performance	J_1	J_2	MAE	MQE
Distributed Multiple Shooting	0.20	-	6.31×10^{-5}	6.99×10^{-9}
Fast Gradient-based DMPC	2568	2440	1.65	4.85
Hierarchical Infinite Horizon MPC	2397	1416	1.52	3.21
Game Theory-based DMPC	4722	4066	3.19	19.06
DMPC based on agent negotiation	5750	4194	3.72	19.88
Decentralized MPC	3.30×10^5	2.78×10^5	208.70	54.08×10^3

Table 8.1: Table of the quantitative benchmark indices of each tested controller

level. In general, most of the approaches present a good behavior regarding constraints violations. Only sporadic small constraint violations appear in some of the approaches:

- Distributed Multiple Shooting: No constraint violations

- Fast gradient-based DMPC: There is no violation for input constraints. The output constraints are slightly violated, only for the water levels of the reaches R_5 and R_6

- Hierarchical Infinite Horizon MPC: some of the levels violate the restriction at certain instants, but they do it in a very short time, the total time where there is constrain violation is 177 minutes (almost 3 hours).

- Game Theory Based MPC: with the proposed control scheme there are two constraints which are not respected. These constraints are those regarding the levels of the lakes L_1 and L_2. The constraints regarding lake L_2 violated almost all the time. The deviation is less than 1m in both directions. The constraints regarding lake L_2 are violated from 45000s until the end of the simulation. The deviation is 1m below the minimum allowed level value.

- DMPC based on agent negotiation: there are two small violations in one of the constraints of the problem. Specifically, the constraints regarding the water level of lake 2 and the water level at reach 1 are violated . In the first case, the maximum value of the deviation with respect the constraint is 0.035 m and the violation takes place during a total time of 2 hours out of the 24 hours of the simulation. In the second case, the maximum violation is -0.05 m and the total time is again 2 hours.

- Decentralized MPC: there is no violations of the input constraints. However the water levels of the reaches R_3, R_4 and R_5 are violated, especially for the reach R_4, the water level blows up.

Finally, communication requirements must be considered in distributed approaches. The following summarizes the communication needs of each one of the methods:

- Distributed Multiple Shooting: The communication costs consist of sending and receiving vectors and matrices. 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 via MPI interface using double precision.

- Fast gradient-based DMPC: the communication costs consist of sending and receiving primal vectors and dual vectors between neighbors in each iteration. The communication costs vary depending on the total number of iterations needed until convergence is obtained. (See Figures 4.5 and 4.6).

- Hierarchical Infinite Horizon MPC: the controller is a hierarchical structure with decentralized local controller, so few information is transmitted. The coordinator transmits the values of the output reference and input variable bias for each subsystem. So the coordinator transmit two vectors to each subsystems of size 12×1 for input variables bias and 17×1 for the reference of the output variables.

- Game Theory Based MPC: each subsystem has to transmit its disagreement point, the value of the local states, and the current control action. The vector of local inputs is a vector with dimension 32×1 for subsystems 3 to 8 and 64×1 for subsystems 1 and 2, the vector of states has a dimension 40×1 for subsystems 3 to 8, 1×1 for subsystem 2, and 2×1 for subsystem 1, and the disagreement point has a dimension 1×1 for all subsystems (all the vectors are double precision vectors).

- DMPC based on agent negotiation: the total number of proposals that are evaluated each sample time is 50. This means that each agent sends an average number of approximately 6 proposals to its neighbors. Given that these results are obtained for a control horizon N_c=10, each proposal consists of a maximum number of 30 floating point reals.

- Decentralized MPC: there is no communications between local controllers.

Bibliography

[1] O. Akira, "A noncooperative approach to general n-person cooperative games," Graduate School of Economics, Hitotsubashi University, Discussion Papers 2005-01, 2005.

[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 M. W., "A comparative analysis of distributed MPC techniques applied to the hd-mpc four tank benchmark," *Journal of Process Control*, vol. 21, no. 5, pp. 800–815, 2011.

[3] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sciences*, vol. 2, no. 1, pp. 183–202, Oct 2009.

[4] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY: Cambridge University Press, 2004.

[5] M. Diehl, F. Jarre, and C. Vogelbusch, "Loss of superlinear convergence for an SQP-type method with conic constraints," *ISIAM Journal on Optimization*, vol. 6, pp. 1201 –1210, 2006.

[6] J. Espinosa, "Control lineal de sistemas multivariables," *Draft version*, 2003.

[7] P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer, "A distributed optimization algorithm with convergence rate $O(1/k^2)$ for distributed model predictive control," Tech. Rep., 2011, submitted to Automatica.

[8] A. González and D. Odloak, "A stable MPC with zone control," *Journal of Process Control*, vol. 19, no. 1, pp. 110–122, 2009.

[9] 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*, 2011.

[10] J. F. Nash, "Equilibrium points in N-persons games," *Proceedings of the National Academy of Sciences of United States of America*, vol. 36, no. 1, pp. 48–40, 1950.

[11] ——, "Two-person cooperative games," *Econometrica*, vol. 21, no. 1, pp. 128–140, 1953.

[12] Y. Nesterov, "A method of solving a convex programming problem with convergence rate O (1/k^2)," *Soviet Mathematics Doklady*, vol. 27, no. 2, pp. 372–376, 1983.

[13] ——, "On an approach to the construction of optimal methods of minimization of smooth convex functions," *Ékonom. i. Mat. Metody*, vol. 24, pp. 509–517, 1988.

[14] ——, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, May 2005.

[15] D. Odloak, "Extended robust model predictive control," *AIChE Journal*, vol. 50, no. 8, pp. 1824–1836, 2004. [Online]. Available: http://dx.doi.org/10.1002/aic.10175

[16] H. J. M. Peters, *Axiomatic bargaining game theory*. Kluwer Academic Publishers, 1992.

[17] K. Rockafellar, *Convex Analysis*. Princeton, NJ: Princeton Univercity Press, 1970.

[18] C. Savorgnan, A. Kozma, J. Andersson, and M. Diehl, "Adjoint-based distributed multiple shooting for large-scale systems," in *Proceedings of the 18th IFAC World Congress*, Milano, Italy, Sep. 2011, pp. 381 – 386.

[19] C. Savorgnan, C. Romani, A. Kozma, and M. Diehl, "Multiple shooting for distributed systems with applications in hydro electricity production," *Journal of Process Control*, vol. 21, pp. 38 –745, 2011.

[20] C. Savorgnan and D. Moritz, "Control benchmark of a hydro-power plant (draft)," Hierarchical and Distributed Model Predictive Control for Large-Scale Systems (HD-MPC), Katholieke Universiteit Leuven, Tech. Rep., March 23 2011.

[21] K.-C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems," *Pacific J. Optim.*, vol. 6, pp. 615–640, 2010.

[22] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," May 2008, submitted to SIAM J. Optim. Available: http://www.math.washington.edu/~tseng/papers/apgm.pdf.

[23] 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.

[24] A. Venkat, J. Rawlings, and S. Wright, "Stability and optimality of distributed model predictive control," in *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference*, Spain, 2005, pp. 6680–6685.

[25] ——, "Stability and optimality of distributed, linear model predictive control. Part I: State feedback," *Texas-Wisconsin Modeling and Control Consortium Technical Report*, vol. 3, 2006.

[26] W.M.Wonham, "On pole assignment in multi-input controllable linear systems," *IEEE Transactions on Automatic Control*, vol. 12, no. 6, pp. 660–665, 1967.

[27] 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*, vol. 21, no. 5, pp. 787–799, 2011.

[28] K. Zhou, J. C. Doyle, and K. Glover, *Robust and optimal control*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.