TÉCNICO
LISBOA

Distributed Adaptive Control of a Water Delivery Canal

Tiago José Ribeiro Ricardo

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor: Prof. Dr. João Manuel Lage de Miranda Lemos

Examinatiom Committee

Chairperson: Prof. Dr. João Fernando Cardoso Silva Sequeira
Supervisor: Prof. Dr. João Manuel Lage de Miranda Lemos
Members of the Committe: Prof Dr. Alexandre José Malheiro Bernardino

November 2016

Intelligence is the ability to adapt to change.

Stephen Hawking

Acknowledgments

The work developed in this study was only possible thanks to the help and support of several people that had a tremendous impact in my journey. These years of study and hard work seemed more easy to endure thanks to the support of friends that since the beginning of the course remained close to me, during group works, meals and special moments that I will cherish for the rest of my life. I also would like to thank to my parents for their patience, precious help and support during these difficult years.

I would also like to thank my supervisor, Professor João Miranda Lemos, for being an inspiring and supportive person during many lectures and during the execution of this study, that introduced me to the exciting world of Automatic Control, and for giving me valuable life lessons that I will keep in my mind. At last but not least, I would like to thank to my girlfriend for all these fantastic years and for being an amazing individual that supported me and gave me strength and hope when I needed the most.

Abstract

Water Delivery Canals are structures characterized for their large dimensions and for allowing the water supply and an efficient resource management. Throughout their large extensions, water canals have several local control agents associated with its pools, composed by sensors and actuators. Due to its large dimensions and to the existing sensors and actuators, there is motivation to implement distributed control strategies, that allow a more efficient communication between local controllers, without requiring a central node. Also, existing external factors and disturbances associated with the usage of water for consumption may change the dynamics of the system, which provides motivation for the usage of adaptive algorithms, in which new models are identified using input and output data retrieved from the system.

The objective of this dissertation is therefore to develop distributed adaptive control strategies that are composed by a recursive identification algorithm (RLS) and by control algorithms based on LQG and MPC theory, with coordination and negotiation mechanisms. The sequence of these steps allows the design of controllers with models estimated recursively using data from the system, becoming therefore adaptable to system dynamics changes. The algorithms considered in this study are based in Game Theory concepts, where the decisions of local control agents take into account the knowledge of the neighbouring agents' decisions, converging therefore to the Nash Equilibrium, and in Lagrangian optimization methods.

Keywords

RLS, LQG, MPC, Distributed Control, Adaptive Control, Water Canal.

Resumo

Os canais de água são estruturas caracterizadas pelas suas grandes dimensões, por possibilitarem o fornecimento de água e por permitirem a gestão eficiente deste recurso natural. Ao longo dos canais de água existem diversos agentes de controlo locais, compostos por sensores e actuadores, que actuam sobre um dado troço. As grandes dimensões e a existência de vários sensores e actuadores nos canais de água motivam a implementação de estratégias de controlo distribuídas, que tornam a comunicação entre controladores locais mais eficiente, sem a necessidade de um nó central. Por outro lado, a existência de factores externos e de perturbações associadas à utilização da água para consumo/fins agrícolas alteram a dinâmica do sistema, pelo que a implementação de algoritmos adaptativos possibilita a identificação de novos modelos com os dados de entrada e saída deste.

O objectivo desta dissertação consiste então no desenvolvimento de estratégias de controlo adaptativas e distribuídas que integram um algoritmo recursivo de identificação, os Mínimos Quadrados Recursivos (RLS), e algoritmos de controlo óptimo LQG e MPC, baseados em mecanismos de coordenação e negociação. A sequência de um algoritmo de identificação e de um algoritmo de controlo possibilita o dimensionamento dos ganhos do controlador com base em modelos estimados recursivamente com dados do sistema. Os algoritmos considerados nesta dissertação baseiam-se em conceitos de Teoria de Jogos, em que as decisões dos vários agentes de controlo locais são tomadas com conhecimento das decisões dos agentes vizinhos, convergindo para o equilíbrio de Nash, e em métodos de optimização Lagrangeana.

Palavras Chave

RLS, LQG, MPC, Controlo Distribuído, Controlo Adaptativo, Canal de Água.

Contents

List of Figures

List of Tables

Abbreviations

ARX AutoRegressive with eXogenous input

ARMAX AutoRegressive Moving Average with eXogenous input

NuHCC Núcleo de Hidráulica e Controlo de Canais

D-MPC Distributed Model Predictive Control

D-LQG Distributed Linear-Quadratic-Gaussian

D-ADMM Distributed Alternating Direction Method of Multipliers

LQG Linear-Quadratic-Gaussian

LQR Linear-Quadratic Regulator

LQE Linear-Quadratic Estimator

MIMO Multiple-Input Multiple-Output

MISO Multiple-Input Single-Output

SISO Single-Input Single-Output

PRBS Pseudo-Random Binary Sequence

MPC Model Predictive Control

RHC Receding Horizon Control

RLS Recursive Least Squares

SS State-Space

PI Proportional-Integral

PID Proportional-Integral-Derivative

List of Symbols

N	Time horizon
t	Discrete time
x	State of the system
$\bar{x}$	Augmented state of the system
$\hat{x}$	State estimate
u	Manipulated variable
Δu	Incremental manipulated variable
ΔU	Incremental manipulated variable (Matrix notation)
y	System output
Θ	Parameter Estimates
Δy	Incremental system output
Y	Incremental system output (Matrix notation)
r	Reference signal
w	Markov parameters
$\mathbf{W}$	Markov parameters (Matrix notation)
$\mathbf{\Pi}$	Predictor Matrix
e	Error

1

Introduction

Contents

1.1 Motivation

Although most of the Earth surface is covered by water, the percentage that is fresh is less than 3 percent, and only a part of it is available to human consumption. According to [1] the growth in the world population requires immediate action to manage this natural and essential resource in a sustainable way. Alongside with the increasing demand of water, the issue of water scarcity has become critical in several regions worldwide and has also a severe impact in some economic sectors such as agriculture.

Irrigation systems play an important role in addressing the problem of water scarcity by allowing efficient resource management. These large structures are characterized for being both highly dynamic due to variations in the water levels, disturbances (mud accumulation, vegetation growing, and physical problems), and for being time-variant in several ways [2]. To address these issues, and to allow the system to respond in a both stable and robust way, it is relevant to understand how the different stretches of a water canal interact and how to use the information provided by the different sensors in order to develop appropriate control strategies. However, there are several issues that need to be addressed [3], [4], [5], in order to increase the efficiency of water management in irrigation canals, such as minimizing losses and ensuring water availability according to demand while taking into account existing water level constraints.

Due to their large dimensions, most irrigation canals are characterized by having several sensors and actuators placed across them that allow the development of local control strategies. Therefore, a water delivery canal may be envisaged as a series of different subsystems that are connected and that interact among each other. Although this sounds appealing to the implementation of centralized control strategies, there is the downside of the increasing complexity of the system and the fact that if the controller fails the consequences might be considerable. The communication between subsystems can also be seen as difficult due to the large dimension of the structure, that introduces problems in terms of computational load and communication [6]. These issues provide motivation for the implementation of decentralized solutions, in which each controller computes its manipulated variable without being aware of the other subsystems. Since the lack of communication between subsystems threatens the performance of the whole system and may even yield an unstable behaviour [7], one possible solution to improve the efficiency and tackle the feasibility and communication issues is the implementation of a distributed control strategy. In this approach the interaction between subsystems is taken into account and communication becomes an important ally in order to develop coordination algorithms with the objective of reaching a consensus between subsystems.

While distributed strategies are characterized by the adequate management of local control objectives [3], adding adaptation to these strategies has the advantage of increasing the performance of the system, by contributing to its stability when facing variations in its dynamics. Therefore, the

combination of adaptation and distributed control strategies appears to be an efficient way to manage the water levels in the canal.

1.2   Problem Formulation

The water canal considered in this dissertation belongs to Núcleo de Hidráulica e Controlo de Canais (NuHCC) of Univerisity of Évora, Portugal. In [8] a more extensive description of the canal is presented, alongside with a manual for the SIMULINK non-linear model used in this study. The system is composed of an automatic canal, together with a traditional canal that guarantees water return. The automatic canal is controlled by a central server equipped with a SCADA system that allows remote monitoring and control of the canal, and consists of a series of four pools separated by four vertical gates that are actuated by electrical motors. The first three pools are terminated by an undershot gate, whereas the fourth pool is terminated by an overshot gate.
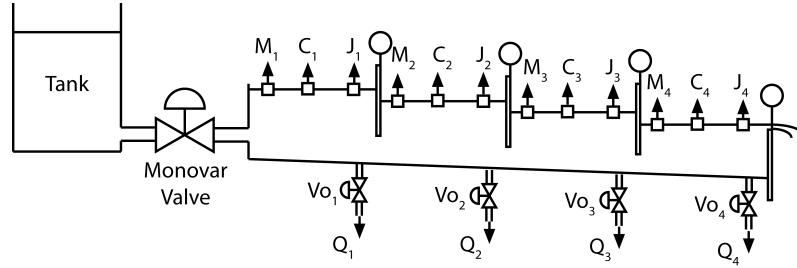
Figure 1.1: Schematic representation of the NuHCC automatic canal.

The pools have approximately the same length ($35m$ for the first three and $36m$ for the fourth pool), and the same height ($0.90m$). In order to measure the upstream (M_i), centre (C_i) and downstream (J_i) water level, three sensors were placed along each pool i, as shown in figure 1.1, although only the last ones are considered in this dissertation. The variables Q_i represent the side flow of off-take i, controlled by valves placed at the canal bottom, with control signals denoted by V_{Oi}. These variables allow the generation of disturbances in the system, being associated, in a practical situation, to water consumption by users.

The dynamics of the water flow in the canal is described by the non-linear hyperbolic Saint-Venant equations [9], obtained using the mass and momentum conservation principles and whose solution is obtained using numerical methods for solving Partial Differential Equations [8]. In this study only the undershot gates, whose schematic representation is shown in figure 1.2 (Lemos, J. M. et al., 2010), are considered and it is assumed that the flow under each is modelled by

$$Q(t) = C_{ds}A(t) * \sqrt{2g(h_u - h_d)}, \tag{1.1}$$

in which C_{ds} is the discharge coefficient, h_u and h_d denote the upstream and downstream water

levels, g is the acceleration of gravity and A, the effective area of the opening of the gate, is given by

$$A(t) = Wu(t) \tag{1.2}$$

in which W represents the width of the gates and $u(t)$ is the position of the undershot gate, at time instant t.
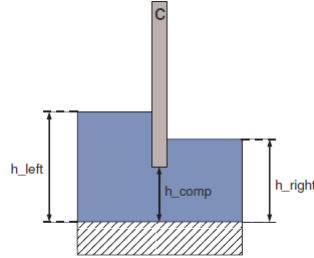
Figure 1.2: Schematic side view representation of the undershot gate. The upstream and downstream water levels are denoted by h_{left} and h_{right} respectively, whereas h_{comp} represents the gate height. [8]

The development of adaptive control algorithms has the purpose of maintaining the water level of each pool at the desired level, even in the presence of existing disturbances and changes in canal dynamics. The adaptive control algorithms implemented in this study take into account estimates of system parameters computed by an identification algorithm that takes into account variations in the dynamics of the system. The parameter identification problem is addressed using the Recursive Least Squares (RLS) algorithm, that recursively estimates the parameters that minimize a predefined least squares function.

The introduction of adaptation in the control algorithms addresses the time-variant dynamic of the system, allowing it to maintain the water levels close to the desired values, even in the presence of external factors or disturbances such as water extraction for agricultural use. The adaptive distributed algorithms developed in this dissertation are based on Linear-Quadratic-Gaussian (LQG) control and Model Predictive Control (MPC) theory. The LQG control algorithms are characterized by an infinite-horizon quadratic cost function that is minimized, and that leads to the controller and state estimator expressions, whereas in the MPC algorithms, a quadratic cost function is only considered for a specific finite-horizon that slides with time, and its minimization may take into account explicit input and state constraints.

The water canal is converted in a network of connected subsystems associated with each pool, in which each system is controlled by a local agent and the computation of manipulated variables takes into account information provided by the neighbours. Although the irrigation canal is a non-linear system with infinite order, in this dissertation it will be approximated by a finite-dimension linear model, and thus the control algorithms developed will address only linear finite dimensional State-Space (SS) models. Since during the period this study was conducted, the water canal was non-operational, the target system considered is the SIMULINK non-linear canal model.

1.3   Literature Review

Throughout the years there have been many studies, [4], [7], [9] that tackle the automatic control of water canals due to strong motivations related with the issues and challenges associated with these structures. The main objectives of automatic control of water canals concern the water level regulation in order to optimize the resource distribution such that the amount of water provided matches the demand [4].

1.3.1   System identification methods

The successful implementation of control algorithms requires accurate models that describe the system dynamics. One of the first challenges to be addressed before developing a suitable control strategy is the identification of the system. Irrigation canals are characterized for having a highly-dynamic and complex behaviour described by the non-linear Saint-Venant equations, whose solution requires approximations provided by numerical methods such as the Preissman Scheme or the Orthogonal collocation [9], [10], [11]. The estimation of model parameters based on the Saint-Venant equations can be made by comparing experimental observations with simulation data and it typically takes into account prior knowledge of the channel structure [11], [12]. The SIMULINK non-linear model of the water canal considered in this dissertation was developed using numerical methods to solve the Saint-Venant equations and was validated and calibrated by comparing simulation results with experimental data [8].

An alternative to the linearization of the Saint-Venant equations is to use linear identification algorithms that require experiments in the system considered to validate a model obtained, taken into account prior physical information, to accurately describe the system dynamics [5], [13]. Models based on the Saint-Venant equations with parameters estimated using experimental data capture the dynamics of the channels but, due to the accuracy and simplicity of linear identification models and to the control algorithms considered in the dissertation, the second approach is preferred [12], [13].

1.3.2   Control algorithms applied to water canals

The control techniques developed and applied to water canals range from centralized to decentralized and distributed approaches [7], with several adaptive strategies also being taken into account [2], [14]. Regarding the system considered in this dissertation, there are several studies, from master thesis to research papers, that tackle the application of control algorithms using as plant the NuHCC water canal [2], [3], [6], [15], [16], [17]. Some of these studies [6], [15], serve as background to the development of the adaptive distributed strategies introduced in this dissertation. The control strategies developed are applied taking into account both linear and non-linear models, and include the application of Proportional-Integral (PI) and Proportional-Integral-Derivative (PID) controllers [18],

[19], optimal Linear-Quadratic-Gaussian (LQG) control algorithms [15], [16], gain-scheduling control [20], MPC algorithms [2], [3], [6], [21] and MUSMAR adaptive control [2].

1.3.3 Distributed control algorithms

The distributed control strategies taken into account in this dissertation were introduced in [6], [15], [17] and [22]. The system is divided into several subsystems (pools and respective gates) that are associated with local controllers in a network structure where the subsystems are connected to their neighbours. Compared with decentralized control strategies, [23], [24], the distributed control approaches have the advantage of considering the interactions between subsystems in the computation of manipulated variables. The control algorithms in [23] and [24] are based in coordination methods that use an augmented Lagrangian in the cost function minimization, whereas the coordination procedure introduced in [6], relies on a distributed optimization algorithm that uses augmented Lagrangian to solve problems in networks of interconnected nodes. A different approach is followed in [15], [17], [22] and [25], where Game Theory concepts are used in the coordination procedure in order to reach a consensus between subsystems.

1.3.4 Other studies regarding water management in water delivery channels

Regarding water management in irrigation channels, there are still several techniques and relevant issues that were addressed throughout the years and that include the usage of different manipulated and measured variables such as water levels, discharges, volumes and gate heights [26]; Fault Tolerant Control (FTC) to guarantee stability in the presence of faults in sensors and actuators [16] and control strategies that address risk mitigation in irrigation systems [27] by executing mitigation actions if risk factors such as inadequate quality of fresh water and failures in gate openings are expected.

1.4 Original Contributions

Three adaptive distributed control algorithms are considered in this dissertation: the first one is a Linear-Quadratic Regulator (LQR) based on optimal control and on Game Theory concepts, in which the coordination of subsystems is achieved with a negotiation step common to all local controllers in order to reach a consensus regarding the computation of manipulated variables, whereas the other two algorithms are based on linear MPC theory, by minimizing a quadratic cost function that is only applied in a pre-defined horizon that slides with time.

The coordination procedures of the adaptive D-MPC strategies introduced in this study rely on the above control structures and are based in different concepts. The first strategy is based on a distributed optimization algorithm that solves problems in networks of interconnected nodes using
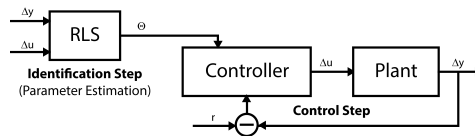
Figure 1.3: Schematic representation of the adaptive controller applied to a linear incremental model.

augmented Lagrangian whereas the second one is based on Game Theory concepts, using a negotiation step similar to the one considered for the D-LQG algorithm.

The identification step of the adaptive controllers is based on the RLS algorithm that provides online estimates of the system parameters. A representation of the general adaptive controller strategy applied to the linear incremental model is shown in figure 1.3, in which input and output increments with respect to the operating point, Δu and Δy, are the variations of the gate position and water levels of a linear subsystem, the reference signal is denoted by r and Θ represents the parameter estimates provided by the RLS algorithm.

1.5 Thesis Outline

This dissertation has the following structure:

Chapter 1 is dedicated to the motivations and problem formulation considered in this dissertation. A literature review is also provided, with references to some of the studies conducted to water canals, followed by a brief presentation of the original contributions.

Chapter 2 addresses the definition of identification strategies, model orders, linear incremental models and of the recursive identification algorithm considered in the adaptive controllers.

Chapter 3 introduces the LQG control theoretical background, followed by the definition of single-gate controllers and centralized multiple-gate controllers, adaptive and non-adaptive.

Chapter 4 addresses the definition of the distributed LQG control algorithm, both in the adaptive and non-adaptive approaches.

Chapters 5 and 6 follow a similar structure as the previous chapters with the first chapter being dedicated to the formulation of the MPC control algorithms and definition of single-gate and centralized multiple-gate controllers whereas the second one is dedicated to the distributed MPC control algorithms, with adaptive and non-adaptive strategies.

2

System Identification

Contents

This chapter adresses the definition of the identification strategies and algorithms used in this study. The first sections are dedicated to the identification of a linear incremental model using simulation data obtained with the SIMULINK non-linear canal model. The last section describes the RLS algorithm considered in the identification step of the adaptive controllers.

2.1 Linear Incremental Model

In order to design the control systems developed in this study, it is necessary to find a model that adequately describes the dynamics of the water canal. The objective is to obtain finite-dimension models that describe the system dynamics in both the SISO and MIMO cases. The parameter estimation from the non-linear SIMULINK model requires the establishment of the orders and of the model structure considered. Since the noise present in the simulation data is not white, one may consider an AutoRegressive Moving Average with eXogenous input (ARMAX) model to describe the system dynamics. However, as previous studies show [15], [16], for the purposes of control design, AutoRegressive with eXogenous input (ARX) models are suitable and the structure is simpler, which makes it preferable.

For a SISO system, the ARX model structure is described by

$$A(q^{-1})\Delta y(t) = B(q^{-1})\Delta u(t - n_K) + e(t), \tag{2.1}$$

where $\Delta y(t) \in \mathbb{R}$ represents the incremental output of the system, $\Delta u(t) \in \mathbb{R}$ the incremental input, $e(t) \in \mathbb{R}$ a Gaussian white noise sequence, q^{-1} is the unit delay operator, and $A(q^{-1})$ and $B(q^{-1})$ are polynomials in the unit delay operator, described by

$$A(q^{-1}) = 1 + a_1 q^{-1} + ... + a_{n_A} q^{-n_A} \quad , \quad B(q^{-1}) = b_0 + b_1 q^{-1} + ... + b_{n_B} q^{-n_B}, \tag{2.2}$$

in which positive integers n_K, n_A and n_B represent the pure delay, and the number of zeros and poles.

The identification of linear incremental models from the non-linear SIMULINK model requires the definition of an equilibrium point, described in table 2.1 for an intake flow of $0.05 m^3/s$.

Table 2.1: Equilibrium Point

Pool	Water Level [m]	Gate Position [m]
1	0.8	0.1072
2	0.72	0.1082
3	0.64	0.1089
4	0.56	0.4022

The first identification strategy used in this dissertation consists of using the *pem* function of MATLAB, that implements the prediction error algorithm over the simulated data. This strategy provides an indication of the orders of the model that correspond to the best results and that will be used in the Recursive Least Squares (RLS) algorithm, used as an identification technique for adaptive control strategies. In order to avoid the excitation of high-frequency modes and non-linearities, the data

was filtered using a third-order low-pass Butterworth filter with a cut-off frequency of $0.1 rad/s$, as described in [16]. The mean and the initial transient of the signal were also removed before applying the identification algorithm. The comparison between the output of the model and the measured one, obtained with the parameter estimates, is accomplished using MATLAB function *compare*, that provides the normalized root mean square error as a fit percentage. The sample time considered was $T_s = 1s$.

2.2 SISO Model Identification

In this first experiment, only the first pool and the corresponding undershot gate were taken into account in order to identify a Single-Input Single-Output (SISO) model. In order to obtain a more reliable model, a suitable signal needs to be applied in the system input, in order to excite it and to provide better parameter estimates. The signal chosen is the Pseudo-Random Binary Sequence (PRBS), since it is characterized for being a variable signal that excites the system along a wide range of frequencies when compared with, for instance, a sequence of square waves, as shown in [16].
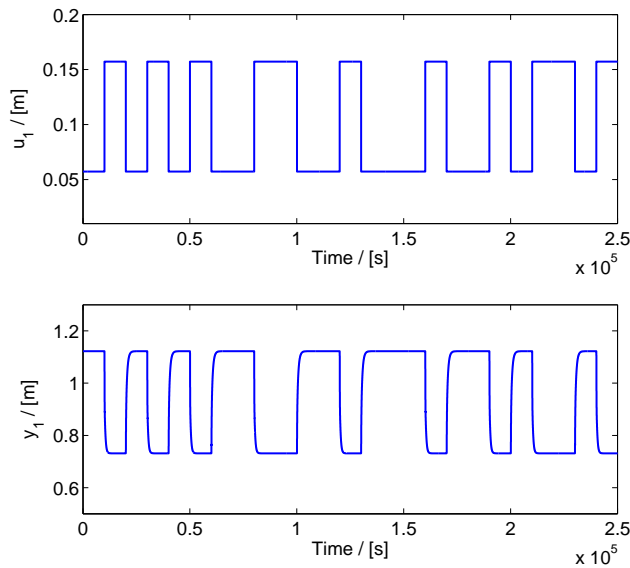
Figure 2.1: Input u_1 and output y_1 signals - first pool. The input signal is a PRBS signal with an amplitude of $0.05m$ applied around the equilibrium point of the pool, defined in table 2.1.

Several simulations were performed with the first pool being excited by a PRBS signal applied around the equilibrium point while the other three gates were kept at their equilibrium points. The simulation results were then filtered, and the mean was removed in order to provide better estimates. The input and output signals obtained are represented in figure 2.1, in which the reverse response of the system may be noticed, with an increase in the gate position u, leading to a decrease in the pool water level y.

After performing several trials, with different combinations of model orders, the one that led to the best fit percentage was $n_A = 4$, $n_B = 2$, $n_K = 1$, leading to the results shown in figure 2.2. During the experiments it was possible to verify that higher amplitude variations in the input signal led to poor identification results, with manifestations of non-linearities and that the period of the input signal had
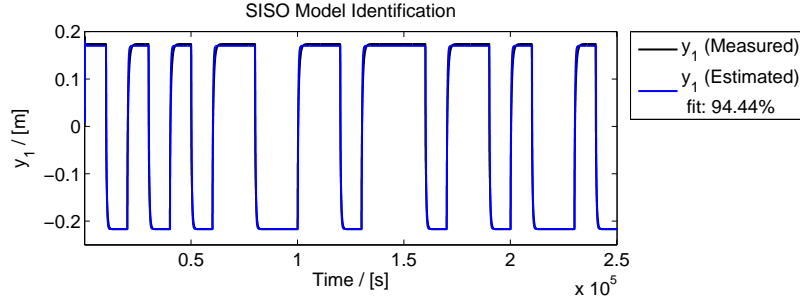
Figure 2.2: Comparison between the measured system output and the simulated output, obtained using the parameter estimates - first pool.

also influence in those results. The frequency of the input signal has to be within a range of values that is sufficiently exciting to provide good estimates, and suitable to the slow system response, in order to allow the stabilization of the water level. Having the parameter estimates and thus knowledge of the transfer function that describes the dynamics of the system, one may obtain an equivalent linear state model to be used in control design.

The conversion between transfer functions and the state model is achieved by either using an analytical approach or by the MATLAB functions *ss2tf* and *tf2ss*. Due to the simplicity of the SISO linear model, the first approach was used, but in the other cases, mostly due to their complexity, MATLAB functions were preferable. The resulting SISO linear incremental SS model is

$$x(t + 1) = Ax(t) + B\Delta u(t) + e(t), \quad \Delta y(t) = Cx(t), \tag{2.3}$$

in which Δu and Δy represent the incremental gate position and water level with respect to the equilibrium point, x is the state of the system, and $A[n_A \times n_A]$, $B[n_A \times n_B]$ and $C[n_C \times n_A]$ are the SS matrices that describe the system dynamics.

2.3 MIMO Model Identification

In order to design a control system for a multi-variable structure that contemplates the first three pools and corresponding undershot gates, one needs to identify a Multiple-Input Multiple-Output (MIMO) linear model from the simulation data. The identification procedure is similar to the one defined in section 2.2, although there are some changes that reflect the interactions between subsystems.

Just like in the previous section the outputs and the inputs of each subsystem are defined as the downstream water levels and gate positions, respectively. But instead of having $A(q^{-1})$ and $B(q^{-1})$ defined as polynomials that represent the zeros and poles of the system, these are now matrices with polynomial entries, with the following underlying assumptions:

- The water level in each pool depends only on the water level in the same pool in previous time instants;

- The inputs only influence the water levels of the respective system and of neighbouring systems.

The MIMO linear model structure is therefore defined as

$$\begin{bmatrix} A_{11}(q^{-1}) & 0 & 0 \\ 0 & A_{22}(q^{-1}) & 0 \\ 0 & 0 & A_{33}(q^{-1}) \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix} = \begin{bmatrix} B_{11}(q^{-1}) & B_{12}(q^{-1}) & 0 \\ B_{21}(q^{-1}) & B_{22}(q^{-1}) & B_{23}(q^{-1}) \\ 0 & B_{32}(q^{-1}) & B_{33}(q^{-1}) \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix} + e(t),$$

(2.4)

with n_A, n_B and n_K representing the model orders, defined as

$$n_A = \begin{bmatrix} n_{A_{11}} & 0 & 0 \\ 0 & n_{A_{22}} & 0 \\ 0 & 0 & n_{A_{33}} \end{bmatrix} \quad , \quad n_B = \begin{bmatrix} n_{B_{11}} & n_{B_{12}} & 0 \\ n_{B_{21}} & n_{B_{22}} & n_{B_{23}} \\ 0 & n_{B_{32}} & n_{B_{33}} \end{bmatrix} \quad , \quad n_K = \begin{bmatrix} n_{K_{11}} & n_{K_{12}} & 0 \\ n_{K_{21}} & n_{K_{22}} & n_{K_{23}} \\ 0 & n_{K_{32}} & n_{K_{33}} \end{bmatrix} . \quad (2.5)$$

The structure of the model (2.4) reflects the assumption that the different canal stretches interact only through the manipulated variables (gate positions) of the adjacent gates. This assumption leads to a good fit of the model and allows the application of the control methods described in subsequent chapters.
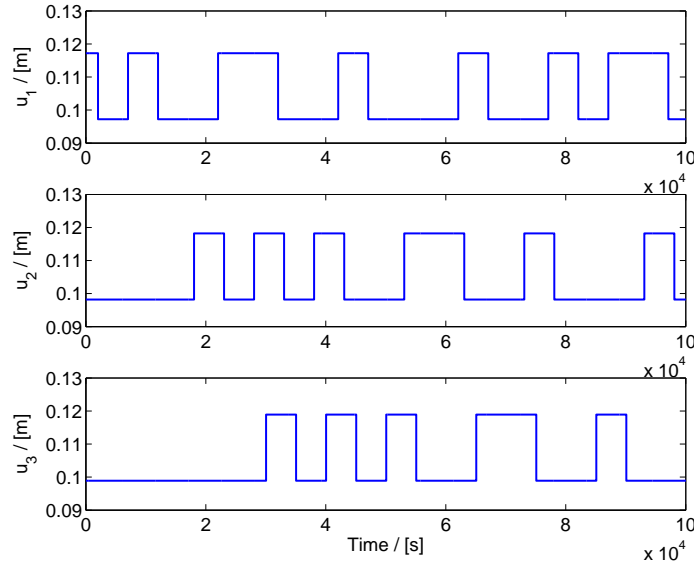
Figure 2.3: Input signals applied to the first three pools to the System - gate position representation.

Using the gate positions as input signals did not result in satisfactory fit percentages (no more than 70%), and thus a solution proposed for this issue, described in [16] was considered. The solution consists of using a variable proportional to the flow drawn by each gate $q(t)$ as input signal. This new variable, $v(t)$, is related with the flow across the gate by the discharge coefficient C_{ds}, as shown in (2.6) and with appropriate input signals it allowed to obtain better results with fit percentages around 91%-96%. The open-loop system response for the three pools is represented in figures 2.3 and 2.4, in which one may see PRBS signals being applied to the inputs, with similar amplitude variations around the equilibrium points defined in table 2.1 but delayed in order to not produce synchronous variations.

$$v(t) = \frac{1}{C_{ds}} q(t) = u(t) * W * \sqrt{2g(h_u - h_d)}.$$

(2.6)

The model identification was performed using the new variable proportional to the flow $v(t)$, and the value of the discharge coefficient C_{ds} was computed as the quotient between the equilibrium point
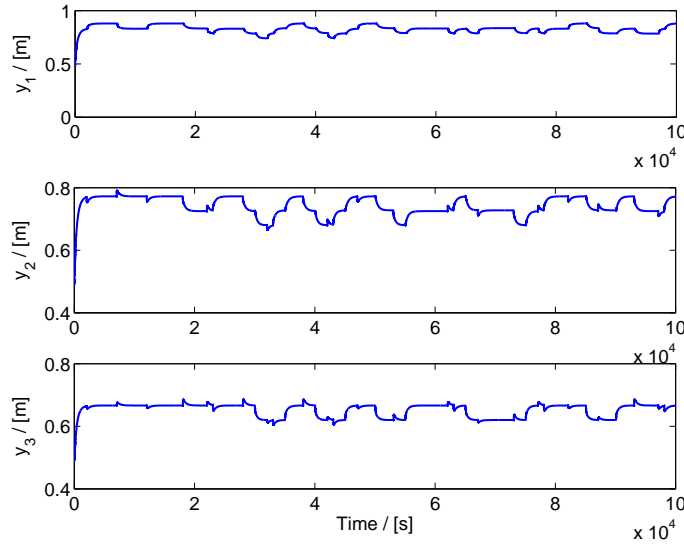
Figure 2.4: Output signals resulting from the open-loop response of the first three pools of the system - water level representation.

of the flow ($0.076m^3/s$) and the intake flow ($0.05m^3/s$). The results obtained are represented in figure 2.5 and by comparing the open-loop responses of the three pools it is possible to verify how the water level in each pool influences the neighbouring subsystems. For example in the first time instants, when the first gate is excited with a negative variation of its position around the equilibrium point which leads to a higher water level in the first pool, the opening of second gate leads to a decrease in the water level of the first pool. The corresponding combination of orders that led to this result is

$$n_A = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix} \quad , \quad n_B = \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & 3 \\ 0 & 4 & 4 \end{bmatrix} \quad , \quad n_K = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 2 \\ 0 & 1 & 1 \end{bmatrix}. \tag{2.7}$$

With the parameter estimates and corresponding transfer function, one may obtain an equivalent state representation of the system using MATLAB function *tf2ss*. However, in order to prevent the appearance of uncontrollable or unobservable states, MATLAB function *minreal* was also used to obtain a minimum realization of the state model. The linear MIMO incremental model structure is given by

$$x(t+1) = Ax(t) + Bv(t) + e(t) \quad , \quad \Delta y(t) = Cx(t) \tag{2.8}$$

$$x(t+1) = \begin{bmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \end{bmatrix} x(t) + \begin{bmatrix} B_{11} & B_{12} & 0 \\ B_{21} & B_{22} & B_{23} \\ 0 & B_{32} & B_{33} \end{bmatrix} v(t) + e(t), \tag{2.9}$$

$$\Delta y(t) = \begin{bmatrix} C_{11} & 0 & 0 \\ 0 & C_{22} & 0 \\ 0 & 0 & C_{33} \end{bmatrix} x(t), \tag{2.10}$$

in which the incremental gate positions $\Delta u_i(t)$ are computed using the relationship between $u_i(t)$ and the variable proportional to the flow drawn by each gate v_i

$$\Delta u_i(t) = \frac{v_i(t)}{W * \sqrt{2g(h_u - h_d)}}. \tag{2.11}$$

14
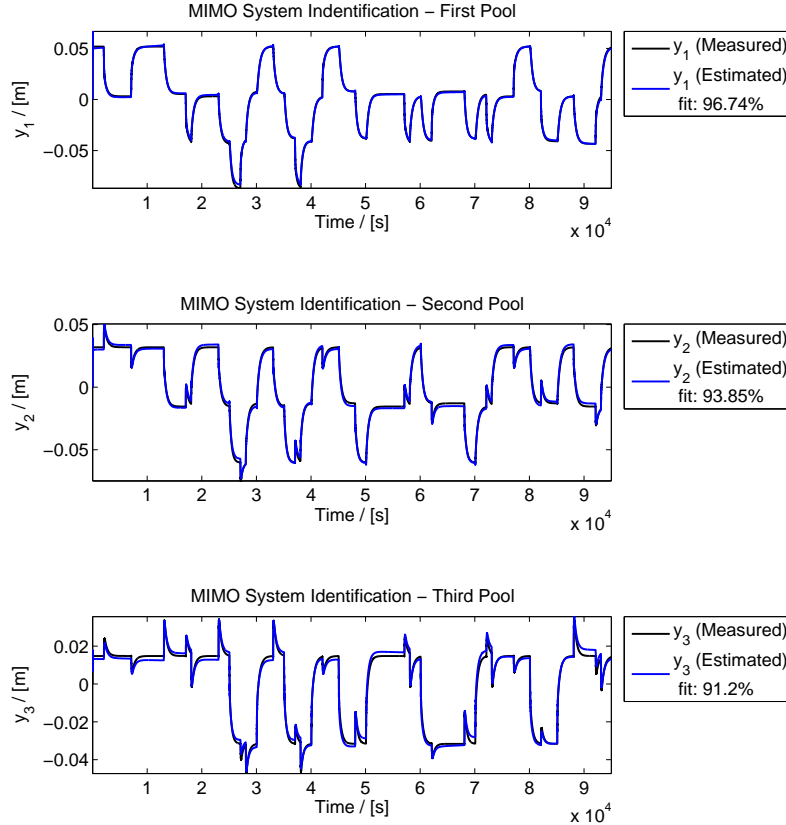
Figure 2.5: MIMO System Identification - Comparison of the open-loop system response simulated and the resulting system output obtained using the parameter estimates for the first three pools.

2.4 MIMO Model Identification with the effect of the side takes

The D-LQG strategy introduced in section 4.1 requires a model composed by several subsystems (pools) connected to their neighbours that interact with their manipulated variables and flow of lateral off-takes. Each subsystem Σ_i, with i denoting the i-th pool, is seen as a MISO system [15], [16], in which its output is the water level y_i of the corresponding pool and its manipulated variable is the position of the respective gate v_i. The interactions between neighbouring subsystems are assumed to be described by the respective gate positions v_{i-1}, v_{i+1}, considered later in the feed-forward control term, and the flows of the lateral off-takes Q_{i-1}, Q_i, Q_{i+1} are handled as accessible disturbances.

Due to the number of inputs, the identification procedure conducted needed to be different than the ones presented before. The simulation time was increased and new input signals had to be applied into the SIMULINK model, representing the lateral off-takes Q_i. Initially, during a pre-defined period of time, only variations in gate positions were considered, and after that time interval only variations in the flows of the lateral off-takes were taken into account. This resulted in identification problems related with different mean values of the system response that corresponded to the different parts of the experiment. The lateral off-take valves were considered to be open in the equilibrium, and thus a new operating point was considered (table 2.2), in which Q_i denotes the flow of lateral off-take of the

i-th pool.
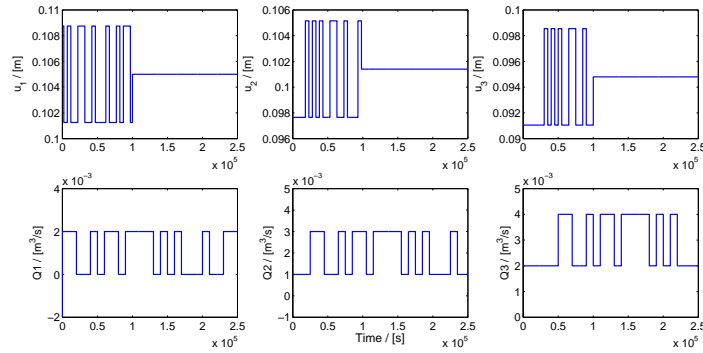
Figure 2.6: Representation of the input signals applied to the SIMULINK model, with Q_i representing flow of the lateral off-take of pool i and u_i the respective gate position.
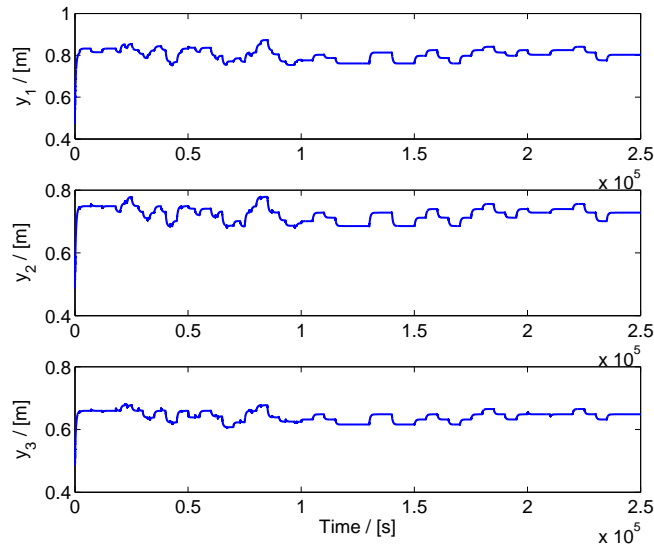
Figure 2.7: Output signals resulting from the open-loop response of the first three pools of the system - water level representation. During the first $1 \times 10^5 s$ both gate positions and flows of lateral off-takes were excited whereas in the rest of the simulation time the gates remained in constant positions.

Table 2.2: Equilibrium Point considering the flow of lateral off-takes

Pool	Water Level [m]	Q_i [m^3/s]	Gate Position [m]
1	0.8	0.001	0.1050
2	0.72	0.002	0.1014
3	0.64	0.003	0.0948
4	0.56	0	0.4150

The identification procedure was divided into two different parts, in which in the first half, PRBS signals were applied simultaneously representing variations in the gate positions and flows of the lateral off-takes. These input signals had different frequencies and were delayed with respect to each other. In the second half of the experiment, gate positions were kept constant at their equilibrium position

while varying the flows of lateral off-takes, as represented in figure 2.6, followed by a representation of the system output in figure 2.7.

The variable proportional to the flow v_i was used again for the model identification and all the signals were filtered and had their mean removed. After performing several experiments, the combination of orders that resulted in the best fit percentages is

$$n_{A1} = 3 \;,\; n_{B1} = \begin{bmatrix} 3 & 4 & 3 & 4 \end{bmatrix} \;,\; n_{K1} = \begin{bmatrix} 1 & 2 & 1 & 1 \end{bmatrix}; \tag{2.12}$$

$$n_{A2} = 3 \;,\; n_{B2} = \begin{bmatrix} 2 & 1 & 2 & 3 & 4 & 3 \end{bmatrix} \;,\; n_{K2} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}; \tag{2.13}$$

$$n_{A3} = 3 \;,\; n_{B3} = \begin{bmatrix} 3 & 4 & 1 & 5 \end{bmatrix} \;,\; n_{K3} = \begin{bmatrix} 1 & 1 & 3 & 1 \end{bmatrix}. \tag{2.14}$$
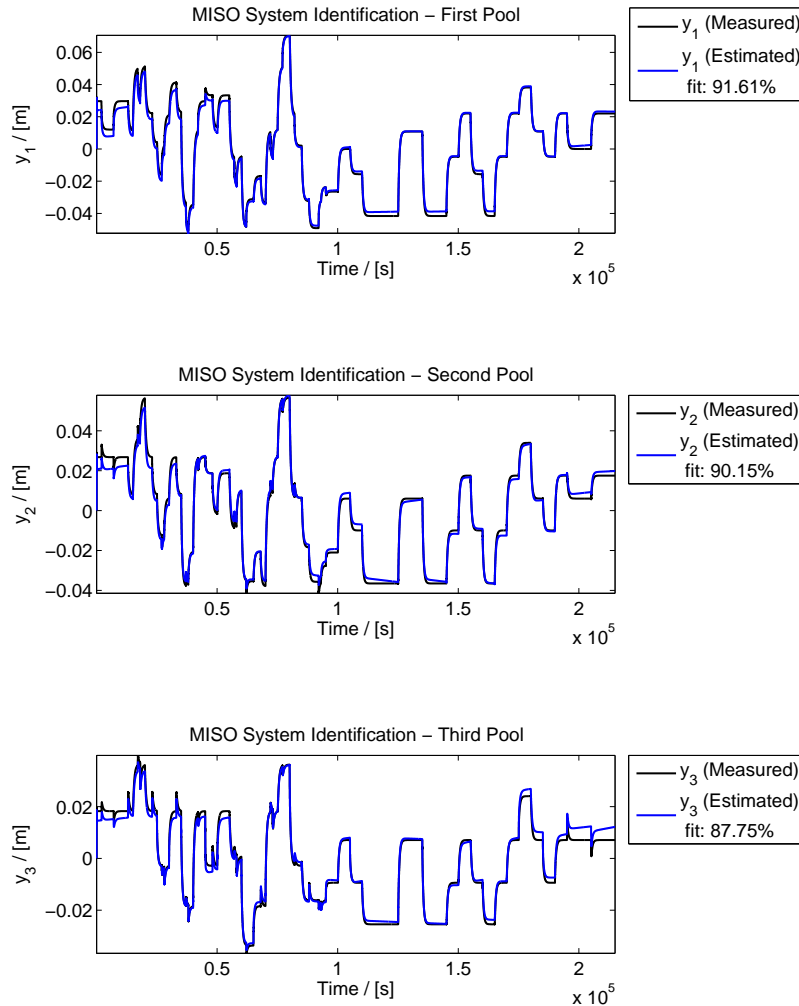
Figure 2.8: MIMO System Identification with the effect of side takes - Comparison of the open-loop system response simulated and the resulting system output obtained using the parameter estimates for the first three pools.

The fit percentages obtained, represented in figure 2.8, are slightly worse than the ones presented

in the other sections, but since they are around 90% and since this identification procedure is only to design controllers and to provide a good indication of the model orders to be used in the adaptive control algorithms, the results obtained are satisfactory. Due to the changes that are required for the distributed LQG control algorithm, the model structure needs to be different, with the B matrix contemplating only the gate positions of the respective pools and the influence of flows from lateral off-takes and positions of neighbouring gates contemplated by matrices Γ_d and Φ_v respectively. The linear incremental model structure is therefore

$$x(t+1) = Ax(t) + Bv(t) + \Phi_v v(t) + \Gamma_d Q(t) + e(t) \quad , \quad \Delta y(t) = Cx(t), \tag{2.15}$$

$$A = \begin{bmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & 0 & 0 \\ 0 & B_{22} & 0 \\ 0 & 0 & B_{33} \end{bmatrix}, \quad \Phi_v = \begin{bmatrix} 0 & B_{12} & 0 \\ B_{21} & 0 & B_{23} \\ 0 & B_{32} & 0 \end{bmatrix}, \quad \Gamma_d = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} & 0 \\ \Gamma_{21} & \Gamma_{22} & \Gamma_{23} \\ 0 & \Gamma_{32} & \Gamma_{33} \end{bmatrix},$$
$$\tag{2.16}$$

$$\Delta y(t) = \begin{bmatrix} C_{11} & 0 & 0 \\ 0 & C_{22} & 0 \\ 0 & 0 & C_{33} \end{bmatrix} x(t). \tag{2.17}$$

The parameter estimates used to define the models introduced in sections 2.2, 2.3 and 2.4 are shown in Appendix A.

2.5   Recursive Least Squares

The results obtained in the previous sections are taken into account for the design of controllers with fixed parameters and provide an indication of the model orders that result in good parameter estimates. However, the off-line application of an identification algorithm is not the approach considered for the parameter estimation step of adaptive controllers. To identify the system dynamics or, in other words, to estimate its parameters, the solution needs to take into account that several observations are required and that memory management should be efficient. This leads us to a recursive solution, which in this dissertation will be the RLS algorithm that uses both incremental input and output data to estimate the parameters [28]. Considering the generic transfer function defined for an ARX model in (2.1), it is possible to write an equivalent difference equation with delayed samples for an incremental model

$$\Delta y(t) = -\sum_{i=1}^{n_A} a_i \Delta y(t-i) + \sum_{i=0}^{n_B} b_i \Delta u(t - n_K - i) + e(t) \tag{2.18}$$

in which $t \geq 0$ is an integer that represents discrete-time, $\Delta u \in \mathbb{R}$ is the incremental manipulated variable, $\Delta y \in \mathbb{R}$ the incremental system output, with increments defined with respect to the operating point, $e \in \mathbb{R}$ represents white Gaussian noise, n_A is the pole order, n_B the zero order and n_K is the system delay. A system is casual if and only if its delay is nonnegative, *i.e.* $n_A \geq n_B$. Discarding systems with instantaneous action, is is $n_A > n_B$. Taking into account expression (2.18), the regressor, φ, is defined as

$$\varphi'(t-1) = \begin{bmatrix} -\Delta y(t-1) & -\Delta y(t-2) & ... & -\Delta y(t-n_A) & \Delta u(t-n_K) & \Delta u(t-n_K-1) & ... & \Delta u(t-n_B) \end{bmatrix}$$
$$\tag{2.19}$$

and the vector of the parameters to be estimated, Θ, is given by

$$\Theta' = \begin{bmatrix} a_1 & a_2 & ... & a_{n_A} & b_0 & ... & b_{n_B} \end{bmatrix}. \tag{2.20}$$

For each observation, the model is described by

$$\Delta y(t) = \varphi'(t-1)\Theta + e(t). \tag{2.21}$$

Given N observations, the estimation of the vector of parameters Θ by $\hat{\Theta}$ is made by minimizing the following cost function:

$$J(\Theta) = \frac{1}{2N} \sum_{t=1}^{N} [\Delta y(t) - \Theta'\varphi(t-1)]^2. \tag{2.22}$$

To obtain the estimate by combining the previous estimates with new data, a recursive estimator is required, with the following elements:

- Vector of estimates $\hat{\Theta}(t-1)$ and previous auxiliary variables $P(t-1)$;

- New data $\Delta y(t)$, $\varphi(t-1)$.

With the combination of these elements it is possible to compute the new estimates $\hat{\Theta}(t)$ and the new auxiliary variables $P(t)$, in which P is the covariance matrix, which is symmetric and positive semi-definite. In order to obtain good parameter estimates one must ensure that the data is adequate in the sense of being sufficiently exciting, thus guaranteeing the decrease of Kalman gain elements K and P. In the first time instants, when the uncertainty regarding the true values of the parameters is large, the values of the covariance matrix P should be high. This leads to higher Kalman gains in the beginning and consequently to a faster convergence of the parameters estimates, as seen in the expression for parameter estimation

$$\hat{\Theta}(t) = \hat{\Theta}(t-1) + K(t)[\Delta y(t) - \varphi'(t-1)\Theta(t-1)], \tag{2.23}$$

in which $\hat{\Theta}$ denotes the vector of parameter estimations at time instant t, φ is the regressor and the Kalman gain is denoted by K.

Higher values of K result in more emphasis to the difference between the experimental value and the estimated value, and thus in a faster convergence of the estimates towards the true values. Throughout time and with the data acquired, the uncertainty regarding the parameters decreases, which results in a smaller value of K and a slower convergence to the real value of the parameters. However, this introduces a new difficulty since the algorithm takes into account recent and past data with the same "weight", and thus when the estimates converge, it takes a considerable amount of time for the estimates to converge towards a new value after a change in the system dynamic. In order to prevent this loss of adaptation from occurring, one may consider defining a forgetting factor, λ, with values between 0 and 1 so that the algorithm weights less data from past. With smaller values of λ, the algorithm tracks better changes in the state and the convergence is faster since it retains less data. With larger values of λ, the algorithm becomes progressively slower to follow changes in the

system dynamic since it retains more data but with less variations in the estimates.

With a fixed forgetting factor λ, the results obtained in the experiments conducted with the SIMULINK canal model were unsatisfactory, with the parameter estimates constantly varying. When the system was in equilibrium, the information provided to the algorithm was not exciting enough to it provide good estimates, and since the input signal applied to the system takes into account its slow response, as described in section 2.2, this situation occurred frequently. In order to prevent this issue, an alternative version of the RLS algorithm with variable forgetting factor, introduced in [29], is considered. In this algorithm, the value of λ depends on the information available and on the current estimates. In order to do so, one must define a new variable ε that denotes the prediction error. One of the required parameters is the mean value of the prediction error, denoted as ε_0, which was defined after conducting several experiments in order to have a measure of the prediction error. In the experiments conducted with the adaptive control algorithms, the parameters of the three subsystems were defined as

$$\lambda_0 = 0.98, \quad \lambda_{min} = 0.98 \quad P_0 = 0.01 \times I_{p \times p}, \quad \varepsilon_0 = 5 \times 10^{-3}, \tag{2.24}$$

in which λ_0 and P_0 are the initial values of the forgetting factor and covariance matrix and p is the number of parameters.

Algorithm 2.1 Recursive Least Squares (RLS) with variable exponential forgetting factor

---

Require: Output signal $\Delta y(t)$, previous Covariance Matrix $P(t-1)$, previous forgetting factor $\lambda(t-1)$ and previous parameter estimates $\hat{\Theta}(t-1)$
function RLS($\Delta y, \hat{\Theta}, P, \varepsilon$)
    for t=1:T
        Read current system output $\Delta y(t)$
        Compute the prediction error $\varepsilon(t) = \Delta y(t) - \varphi'(t-1)\hat{\Theta}(t-1)$
        Define the mean value of the prediction error ε_0
        Compute the Kalman gain $K(t) = \frac{P(t-1)\varphi(t-1)}{\lambda(t-1) + \varphi'(t-1)P(t-1)\varphi(t-1)}$
        Compute the parameter estimates $\hat{\Theta}(t) = \hat{\Theta}(t-1) + K(t)\varepsilon(t)$
        The new forgetting factor λ is given by $\lambda(t) = 1 - [1 - \varphi'(t-1)K(t)]\varepsilon^2(t)/\varepsilon_0$
        If $\lambda(t) < \lambda_{min} \rightarrow \lambda(t) = \lambda_{min}$
        Compute the Covariance Matrix $P(t) = [I - K(t)\varphi'(t-1)]\frac{P(t-1)}{\lambda(t)}$
return $\hat{\Theta}(t)$

3

LQG Control

Contents

In this chapter the LQG controller is described, with the theoretical background followed by the definition of two adaptive controllers applied respectively to the first pool of the canal and to the multi-variable system composed by the first three pools.

3.1 Linear-Quadratic Gaussian Controller

The first control strategy to be considered in this dissertation is the LQG controller, that results from the combination of a Linear-Quadratic Regulator (LQR) and a Linear-Quadratic Estimator (LQE). This observer-controller structure is represented in figure 3.1. The linear incremental model, for a SISO system, considered in the definition of this controller is described by (2.3).
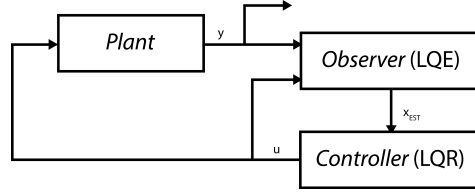
Figure 3.1: Schematic representation of the LQG controller.

3.1.1 Linear-Quadratic Regulator

Assuming that the system state is accessible for direct measure, the LQR control law,

$$\Delta u(t) = -Kx(t), \tag{3.1}$$

is obtained by finding the gain K that minimizes the quadratic cost function

$$J = \sum_{t=0}^{\infty} x(t)^T Q x(t) + \rho \Delta u^2(t), \tag{3.2}$$

in which $Q \in \mathbb{R}^{n_A \times n_A}$ is a positive semi-definite matrix and ρ is a positive scalar quadratic cost weight. In the case of a multi-variable system with several inputs, $\rho \in \mathbb{R}^{n_B \times n_B}$. K is given by

$$K = (I + \rho^{-1}B^T P B)^{-1} \rho^{-1} B^T P A, \tag{3.3}$$

with P denoting the positive definite solution of the algebraic Riccati equation. In MATLAB the state feedback gain K is given by the *dlqr* function from the Control Systems Toolbox.

3.1.2 Linear-Quadratic Estimator

In the previous section, it was assumed that the system state was accessible, which is not true most of the times. In this study, since there is no access to the state, it is necessary to obtain an indirect measure of it. In order to do so, one begins by defining an optimal Linear-Quadratic Estimator (LQE), which provides estimates of the state $\hat{x}$. The state estimate is given by

$$\hat{x}(t|t) = A\hat{x}(t-1|t-1) + B\Delta u(t-1) + M[\Delta y(t) - C(A\hat{x}(t-1|t-1) + B\Delta u(t-1))], \tag{3.4}$$

in which M is the optimal gain matrix obtained by minimizing a cost function that depends on the estimator covariance matrices Q_E and R_E related with the process and measurement noises [16]. M is given by

$$M = PC^T(CPC^T + R_E)^{-1}, \tag{3.5}$$

with P denoting the positive definite solution of the algebraic Riccati equation. In MATLAB, the observer gain M is computed using the function *dlqe* from the Control Systems Toolbox. The state estimates computed by the observer take into consideration the reference signal r, in order to tackle the reference tracking problem. The signal is added to expression (3.4),

$$\hat{x}(t|t) = A\hat{x}(t-1|t-1) + B\Delta u(t-1) + M[\Delta y(t) - C(A\hat{x}(t-1|t-1) + B\Delta u(t-1))] - Mr(t), \tag{3.6}$$

which can be written as

$$\hat{x}(t|t) = \Phi_E \hat{x}(t-1|t-1) + \Gamma_E \Delta u(t-1) - M(e(t)), \quad \Phi_E = A - MCA, \quad \Gamma_E = B - MCB, \quad e(t) = r(t) - \Delta y(t). \tag{3.7}$$

3.1.3 Integral action

In order to guarantee that the system response follows the reference signal, the control design requires the inclusion of an integrator, defined by

$$x_I(t) = \frac{T_s}{q-1}e(t) \iff x_I(t+1) = x_I(t) + T_s e(t), \tag{3.8}$$

in which $e(t)$ is the output error defined by $e(t) = r(t) - \Delta y(t)$, $r(t)$ is the reference signal and T_s is the sampling time, defined as $T_s = 1s$. The state-space model of the system is now described by

$$\bar{x}(t+1) = \bar{A}\bar{x}(t) + \bar{B}\Delta u(t), \quad \bar{x} = \begin{bmatrix} x \\ x_I \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} A & 0 \\ -T_s C & I \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \tag{3.9}$$

with the system output being written as

$$\Delta y(t) = \bar{C}\bar{x}(t), \quad \bar{C} = \begin{bmatrix} C & 0 \end{bmatrix}. \tag{3.10}$$

Expressions (3.8) and (3.9) define the augmented state-space model of the system with integral action, which is used in the computation of manipulated variables. The LQR takes into account the augmented state-space matrices ($\bar{A}$, $\bar{B}$, and $\bar{C}$) to compute the manipulated variables, that can be written as

$$\Delta u(t) = - \begin{bmatrix} K & K_I \end{bmatrix} \begin{bmatrix} x(t) \\ x_I(t) \end{bmatrix}, \tag{3.11}$$

with

$$\begin{bmatrix} K & K_I \end{bmatrix} = (I + \rho^{-1}\bar{B}^T\bar{P}\bar{B})^{-1}\rho^{-1}\bar{B}^T\bar{P}\bar{B}, \tag{3.12}$$

in which $\bar{P}$ denotes the algebraic solution of the Riccati equation using the augmented state-space model. Since this augmented state-space model is unobservable [16], the cost function defined by (3.2) needs to be rewritten, in order to include a term that depends on the integrator state x_I and on a matrix Q_I with dimensions $n_A \times n_A$,

$$J = \sum_{t=0}^{\infty} \bar{x}(t)^T \bar{Q}\bar{x}(t) + \Delta u(t)^T \rho \Delta u(t), \quad \bar{Q} = \begin{bmatrix} Q & 0 \\ 0 & Q_I \end{bmatrix}, \quad Q_I = I. \tag{3.13}$$

Regarding the LQE, the computation of the observer gain M takes into account the original state-space model defined in (2.3) and thus the formulation introduced in section 3.1.2 is still considered.

3.1.4 LQG Controller - Separation Theorem

The combination of the LQR with the LQE results in the Linear-Quadratic-Gaussian (LQG) controller, as previously mentioned. While x is estimated by the LQE, since the augmented state-space system is unobservable, the true value of the integrator state x_I is considered. In order for the LQG controller to work, the regulator and the estimator need to be design separately. According to the Separation Theorem, it is possible to design separately the observer and the regulator, guaranteeing that after coupling them, the closed-loop system poles are the same as the ones obtained separately. In [16], proof that this theorem is valid in this situation is given, and thus it is possible to design the LQR and the LQE separately.

The schematic representation of the LQG controller applied to the system is shown in figure 3.2.
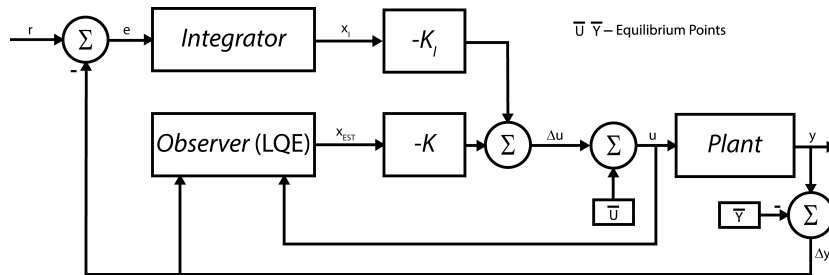
Figure 3.2: Schematic representation of the LQG controller with integral action applied to a linear incremental model.

3.2 Single gate LQG controller

After describing the theoretical background of the LQG controller, the first algorithm developed was applied to the first pool of the canal, taking into consideration the SISO model identified in section 2.2. The LQR and LQE definition requires an initial step in which the suitable controller parameters, ρ, R_E, and Q_E, are defined. In order to find the suitable parameters, several experiments were conducted with different combinations of values, whose results are represented in figure 3.3. In order to verify the influence of ρ, experiments were conducted with $R_E = 1$ and $q = 1$. Higher values of ρ result in a slower system response, while lower values are associated to faster system response with more oscillations. In this scenario, the suitable value for ρ is the one that results in a compromise between the fastest possible system response with less oscillations. Taking into consideration the results represented in figure 3.3, the value chosen was $\rho = 1000$.

With a value defined for ρ, it is still needed to define the estimator, which requires two covariance matrices R_E and Q_E to compute the optimal observer gain M. In the SISO model, R_E is a positive constant and Q_E is a matrix defined as $q^2 B B^T$. With $\rho = 1000$ experiments were conducted to

determine the influence of the estimator parameters. By maintaining $R_E = 1$, it is possible to see that an higher value of q results in a faster system response with more oscillations and higher overshot, and thus the value chosen was $q = 1$. Similar experiments were conducted with the values already defined for ρ and q but with different values for R_E. This parameter as an effect on the estimator similar to the effect of ρ in the regulator, and as it is possible to verify, an higher value of R_E results in a slower system response while lower values are associated to a faster response with more oscillations. The combination of values for the LQG controller parameters is $\rho = 1000$, $q = 1$ and $R_E = 100$. In figure 3.4 it is represented the closed-loop response of the controlled SISO system composed by the first pool and corresponding gate with the parameters defined above.
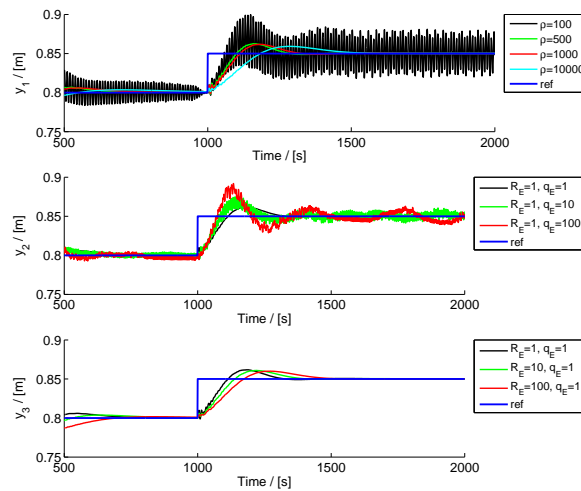
Figure 3.3: LQG controller parameter tunning - Results of the experiments conducted to find the best combination of parameters. Closed-loop response of the SIMULINK non-linear canal model, considering only the first subsystem composed by the first pool and gate. The remaining three gates were kept at their equilibrium positions.
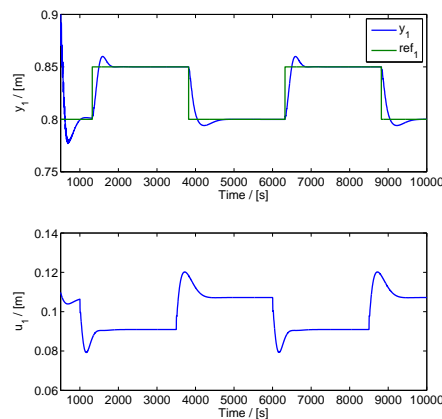
Figure 3.4: Closed-loop response of the SIMULINK non-linear canal model, considering only the first subsystem composed by the first pool and gate, controlled by the SISO LQG controller with $\rho = 1000$, $q = 1$ and $R_E = 100$. The remaining three gates were kept at their equilibrium positions.

3.3 Adaptive single gate LQG controller

After defining the LQG controller, a corresponding adaptive strategy based on the same algorithm was developed. This adaptive algorithm is basically divided into two steps:

- **Identification**: Estimation of SISO model parameters using the RLS algorithm 2.1 with variable exponential forgetting;

- **Control**: Execution of LQG controller applied to a SISO model, defined in section 3.2, that takes into account a model obtained from the parameter estimates.

The identification RLS algorithm takes into account incremental input and output data, $\varDelta u$, and $\varDelta y$, at each time instant t to provide estimates of the system parameters. The model orders are defined in section 2.2, and in order to guarantee that the estimates are closer to convergence, during a pre-specified period of time t_I, only the identification step is activated. During that time period, the system system is being excited by a PRBS signal of amplitude $0.01m$ that remains active during the whole experiment. The period of time considered was $t_I = 7 \times 10^4 s$. After the initial period of time, the controller step is activated and the two steps are combined sequentially, with the RLS algorithm providing the parameter estimates used to define the SISO model that is taken into account by the LQG controller. This strategy is represented in figure 3.5. The controller parameters considered are $\rho = 1000$, $q = 1$ and $R_E = 100$, with $Q_E = qI_{n_A \times n_A}$.

The system response obtained with this adaptive controller is represented in figure 3.6, followed by the parameter estimates in figure 3.7, where it is possible to see that by the time the controller step is activated these are closer to convergence.
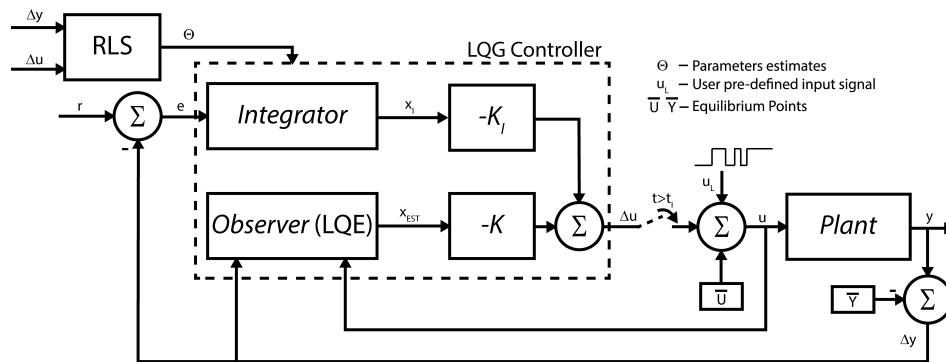
Figure 3.5: Schematic representation of the adaptive LQG controller with integral action applied to a linear incremental model.

3.4 Multiple gate LQG controller

After defining the LQG controller applied to the first gate and pool, it is possible to use the same theoretical background to develop a multi-variable controller applied to the model identified in section 2.3 and described by expression (2.8). The system state x is defined as $x(t) = \begin{bmatrix} x_1(t) & x_2(t) & x_3(t) \end{bmatrix}^T$,
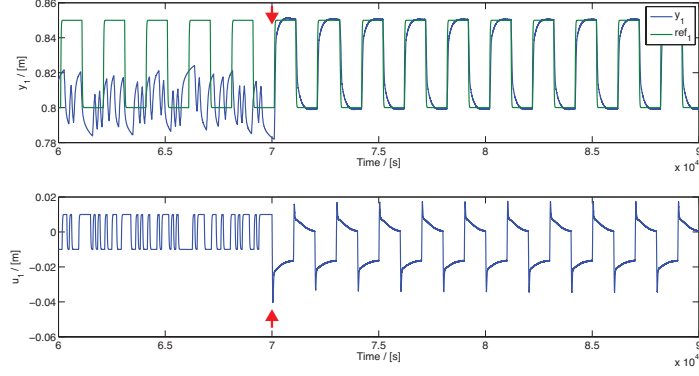
Figure 3.6: Open-loop and closed-loop response of the SIMULINK non-linear canal model between $6 \times 10^4 s$ and $9 \times 10^4 s$, considering only the first subsystem composed by the first pool and gate, controlled by the adaptive SISO LQG controller with $\rho = 1000$, $q = 1$ and $R_E = 100$. The remaining three gates were kept at their equilibrium positions. The controller step is activated at $t = 7 \times 10^4 s$.
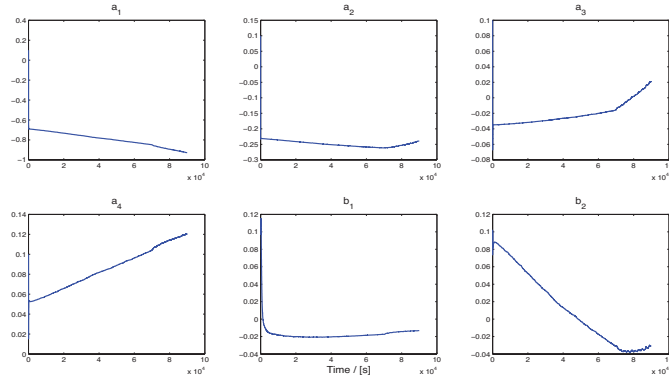
Figure 3.7: Representation of the SISO parameters estimates using the Adaptive LQG controller.

in which x_i is the i-th subsystem state (i-th pool) and the integrator state is defined as $x_I(t) = \begin{bmatrix} x_{I1}(t) x_{I2}(t) x_{I3}(t) \end{bmatrix}^T$. The state-space matrices are now

$$A = \begin{bmatrix} A_1 & 0 & 0 \\ 0 & A_2 & 0 \\ 0 & 0 & A_3 \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} & 0 \\ B_{21} & B_{22} & B_{23} \\ 0 & B_{32} & B_{33} \end{bmatrix}, \quad C = \begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{bmatrix}, \tag{3.14}$$

and as for the controller parameters these are defined as

$$R = \rho I_{3\times3}, \quad \bar{R}_E = R_E I_{3\times3}, \quad \bar{Q}_E = q^2 B B^T, \tag{3.15}$$

in which the parameters associated with each subsystem are assumed to be equal to simplify the controller design. In order to define the controller parameters, several experiments were conducted with different combinations of values, with the corresponding results represented in figures 3.8-3.10.

Taking into consideration the same aspects referred in section 3.2, regarding the parameter tunning of the LQG controller applied to the first subsystem, the suitable combination of values for the parameters is $\rho = 1000$, $q = 1$, $R_E = 100$. In figures 3.11 and 3.12 it is represented the system response with this combination of parameters, in which it is possible to see how the output track the corresponding reference signals and the centralized effect visible whenever occurs a variation in the position of one gate.
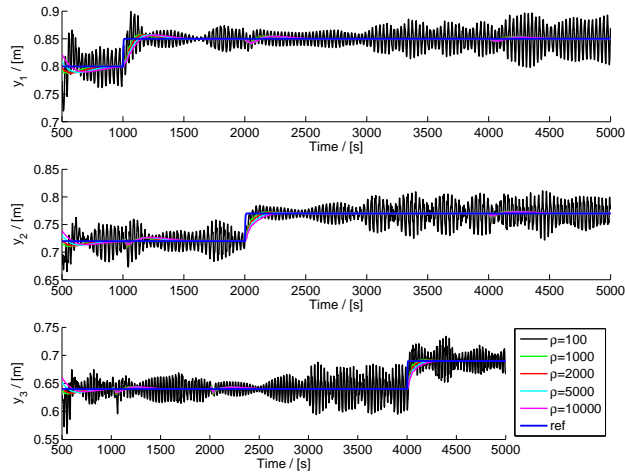
Figure 3.8: LQG controller parameter tunning ρ - Closed-loop response of the SIMULINK non-linear canal model, considering the MIMO model. The fourth gate was kept on its equilibrium position. The experiments were conducted with $q = 1$ and $R_E = 1$.
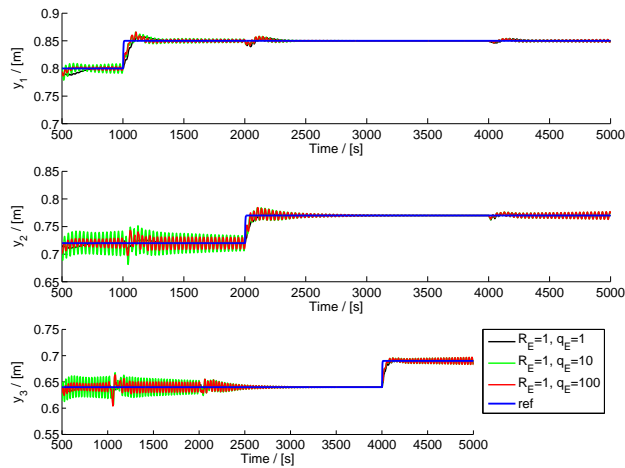
Figure 3.9: LQG controller parameter tunning q - Closed-loop response of the SIMULINK non-linear canal model, considering the MIMO model. The fourth gate was kept on its equilibrium position. The experiments were conducted with $\rho = 1000$ and $R_E = 1$.

3.5 Adaptive multiple gate LQG controller

The MIMO LQG controller defined in the previous section is now considered in the formulation of an adaptive centralized multi-variable control strategy. This strategy is similar to the adaptive algorithm introduced in section 3.3, regarding the sequence of two main blocks: an identification algorithm to provide estimates of the system parameters and a control algorithm that takes into account a model built with those estimates. During a period of time, denoted t_I, only the identification step is activated, and in this case, due to the higher complexity of the model (in terms of number of parameters), the duration was increased to $t_I = 1 \times 10^5 s$. This allows the estimates to be closer to convergence when the control step is activated, avoiding eventual identification issues that may prevent the controller from stabilizing the system. During this time period each subsystem, composed by a pool and its cor-
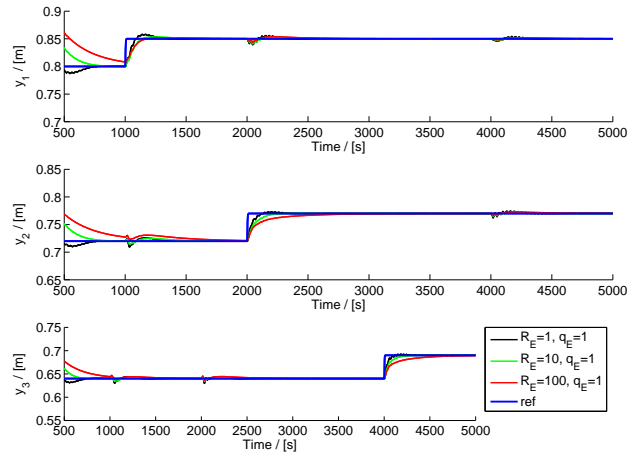
Figure 3.10: LQG controller parameter tunning R_E - Closed-loop response of the SIMULINK non-linear canal model, considering the MIMO model. The fourth gate was kept on its equilibrium position. The experiments were conducted with $\rho = 1000$ and $q = 1$.
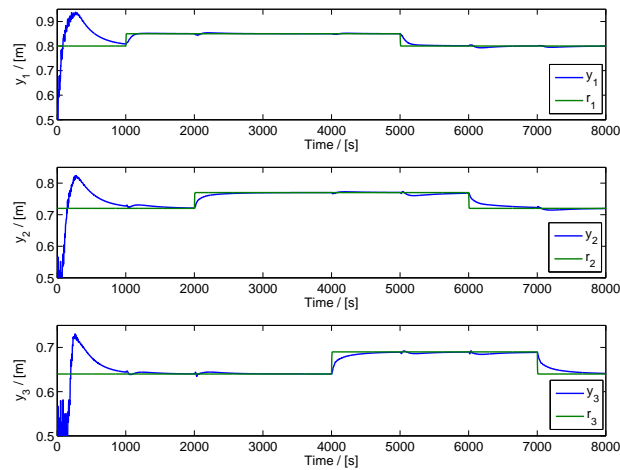
Figure 3.11: Closed-loop response of the SIMULINK non-linear canal model, considering the system composed of the first three gates, controlled by the MIMO LQG controller with $\rho = 1000$, $q = 1$ and $R_E = 100$. The fourth gate was kept at its equilibrium position. (Output and Reference signals)

responding gate, is excited with a PRBS signal of amplitude $0.01m$. The signal remains active during the whole experiment, guaranteeing that the input is being constantly excited, to prevent identification issues such as the covariance blow-up. The neighbouring input signals are asynchronous with the purpose of identifying the influence of each subsystem in its neighbours, similar to the process defined in section 2.3. This strategy is also represented by figure 3.5 and the controller parameters considered are $\rho = 1000$, $q = 1$ and $R_E = 100$, with $R = \rho I_{3\times3}$, $\bar{R}_E = R_E I_{3\times3}$, and $\bar{Q}_E = q I_{3\times3}$.

During the controller definition it was verified that removing the linearization of the input provided better results with the adaptive algorithm, since that new variable "filtered" some of the excitation that is required by the RLS algorithm to estimate the parameters. Therefore, the following adaptive strategies based on the distributed LQG algorithm and on predictive controllers will not contemplate
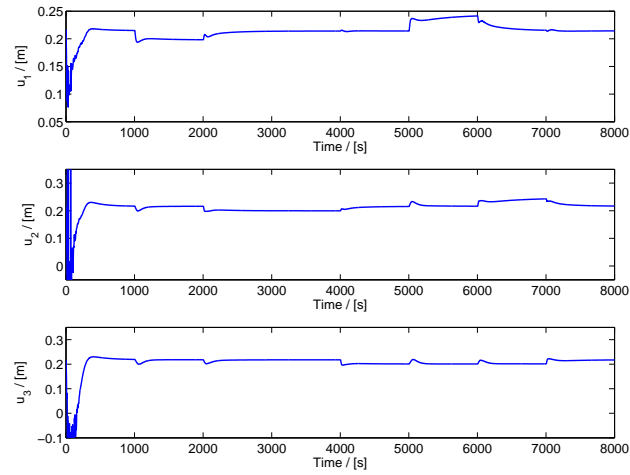
Figure 3.12: Closed-loop response of the SIMULINK non-linear canal model, considering the system composed of the first three gates, controlled by the MIMO LQG controller with $\rho = 1000$, $q = 1$ and $R_E = 100$. The fourth gate was kept at its equilibrium position. (Input signals)

the linearization of the manipulated variable. The results obtained are represented in figures 3.13 and 3.14, in which it is possible to verify that the outputs track the corresponding reference signals and the centralized behaviour of the system whenever occur variations in its subsystems. The parameters estimates are represented in figure 3.15.
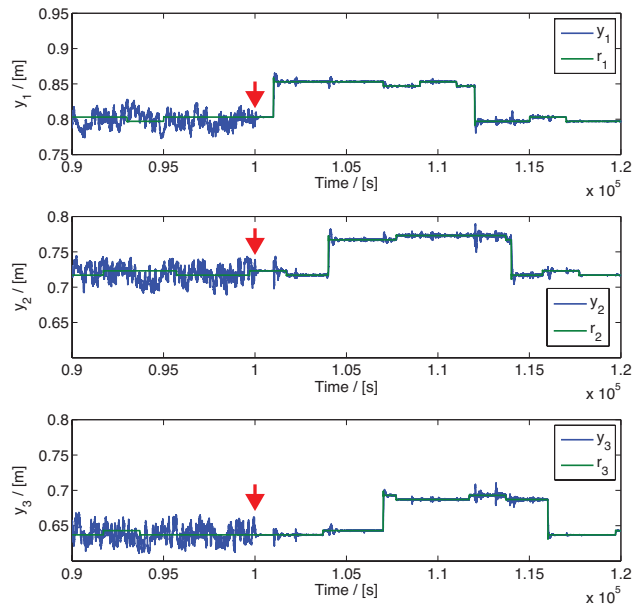
Figure 3.13: Open-loop and closed-loop response of the SIMULINK non-linear canal model, considering the system composed of the first three gates, controlled by the adaptive MIMO LQG controller with $\rho = 1000$, $q = 1$ and $R_E = 100$. The fourth gate was kept at its equilibrium position. The controller step was switched on at $1 \times 10^5 s$. (Output and Reference signals)
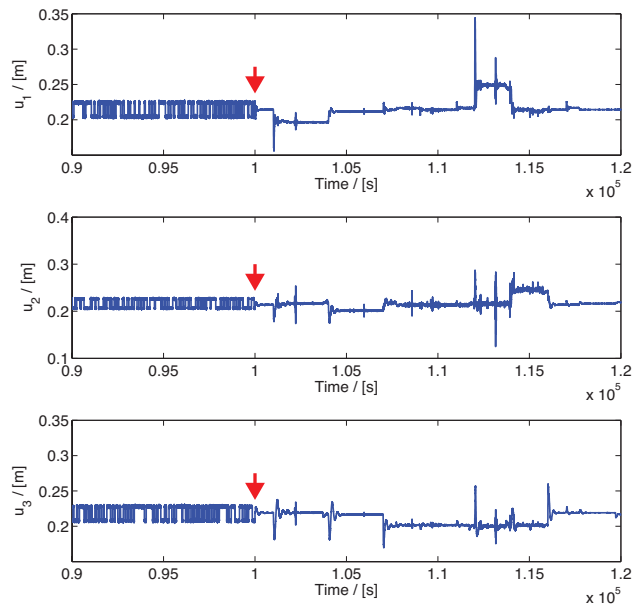
Figure 3.14: Open-loop and closed-loop response of the SIMULINK non-linear canal model, considering the system composed of the first three gates, controlled by the adaptive MIMO LQG controller with $\rho = 1000$, $q = 1$ and $R_E = 100$. The fourth gate was kept at its equilibrium position. The controller step was switched on at $1 \times 10^5 s$. (Input signals)
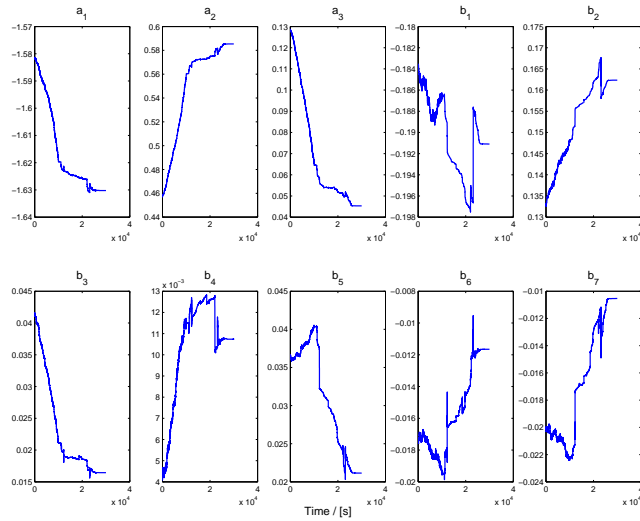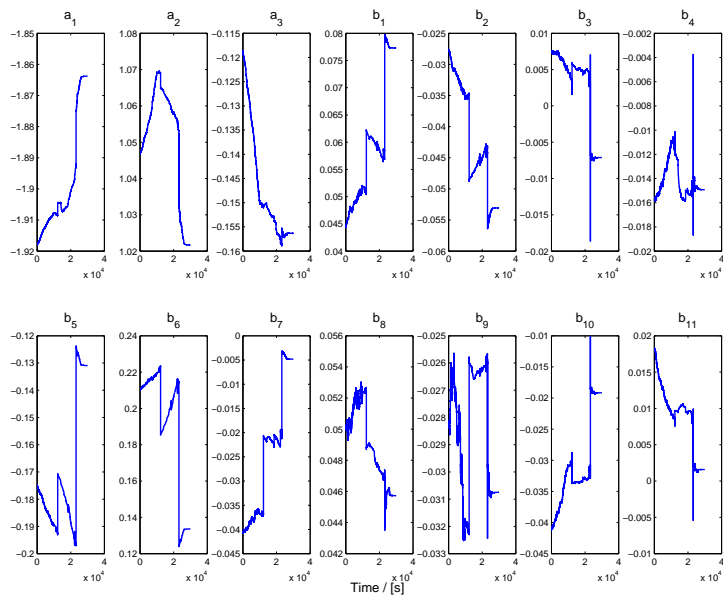
(a) Parameter estimates of the first subsystem.

(b) Parameter estimates of the second subsystem.
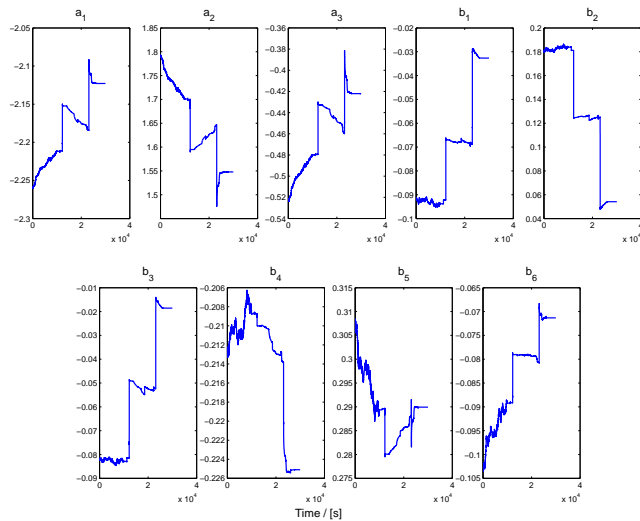
(c) Parameter estimates of the third subsystem.

Figure 3.15: Representation of the MIMO parameters estimates using the adaptive LQG controller between $9 \times 10^4 s$ and $1.2 \times 10^5 s$. The control step is switched on at $1 \times 10^5 s$.

4

Distributed LQG Control

Contents

This chapter is dedicated to the definition of a distributed LQG control strategy based on an iterative procedure that relies on the subsystems coordination, while the local controllers negotiate in order to reach a consensus. This strategy takes into consideration the MIMO state-space model identified in section 2.4, in which the inputs of neighbouring systems and the flow of the lateral off-takes are treated as accessible disturbances.

4.1 Control Law definiton

The state model considered in this section is the model defined by (2.15), in which $v(t)$ and $Q(t)$ denote the accessible disturbances regarding the manipulated variables of neighbouring subsystems and the flow of the lateral off-takes. In order to simplify the notation, the SS model described by (2.15) can be written as

$$x(t+1) = Ax(t) + Bv(t) + \Gamma d(t), \quad \Delta u(t) = Cx(t), \tag{4.1}$$

with

$$\Gamma = \begin{bmatrix} \Phi_v & \Gamma_d \end{bmatrix}, \tag{4.2}$$

in which the entries of Φ_v are the vectors B_{ij} with $i \neq j$, which denote the effect of neighbouring manipulated variables and the entries of Γ_d are the vectors B_{ij} that represent the effect of the lateral off-take flow. The disturbances are denoted by d and defined as

$$d(t) = \begin{bmatrix} v(t) \\ Q(t) \end{bmatrix} = \begin{bmatrix} v_{i-1}(t) & v_i(t) & v_{i+1}(t) & Q_{i-1}(t) & Q_i(t) & Q_{i+1}(t) \end{bmatrix}^T. \tag{4.3}$$

The SS model described above represents each subsystem i, and with integral action, expression (4.1) is written as

$$\bar{x}(t+1) = \bar{A}\bar{x}(t) + \bar{B}v(t) + \bar{\Gamma}d(t), \quad \Delta u(t) = \bar{C}\bar{x}(t), \tag{4.4}$$

with $\bar{\Gamma} = \begin{bmatrix} \Gamma & \underline{0} \end{bmatrix}^T$, and $\bar{A}$, $\bar{B}$, and $\bar{C}$ are defined by (3.9-3.10). The controller structure is represented in figure 4.1, in which it is possible to verify how the local control agents communicate with each other.
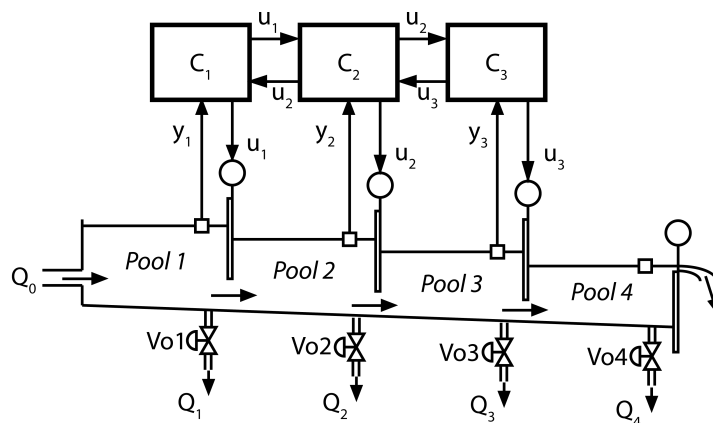
Figure 4.1: Schematic representation of the distributed controller structure.

At each time instant t, each controller has access to the manipulated variables of its neighbours and to the flow of the lateral off-takes of the respective pool and of neighbouring susbsystems. This

information exchange between local control agents is crucial to the coordination procedure. The control strategy introduced in this section was already studied in [15] and [16]. The control law is obtained by minimizing the quadratic cost function defined in (3.13), taking into consideration the new SS MISO model defined by (4.4). The solution for this minimization problem is obtained by applying the discrete-time version of the Pontryagin Minimum Principle, described in the studies previously mentioned and in appendix B.

Taking into account the performance index defined by (B.2) and the cost function (3.2), it is possible to write the Hamiltonian function as

$$H(t) = \lambda^T(t)[\bar{A}\bar{x}(t) + \bar{B}v(t) + \bar{\Gamma}d(t)] - \frac{1}{2}[\bar{x}(t)^T\bar{Q}\bar{x}(t) + v^T(t)\rho v(t)], \tag{4.5}$$

in which λ is the co-state. According to the Pontryagin Minimum Principle, the solution of the stationary condition with respect to v is given by

$$\frac{\partial H(t)}{\partial v(t)} = \lambda^T(t+1)\bar{B} - \rho v(t) = 0 \quad \Leftrightarrow \quad v(t) = \rho^{-1}\bar{B}^T\lambda(t+1). \tag{4.6}$$

Assuming that $\lambda(t) = -\bar{P}\bar{x}(t) + g$, with $\bar{P}$ being a $n_{\bar{A}} \times n_{\bar{A}}$ matrix and g is a vector related with the accessible disturbances, it is possible to compute, at each time instant t, the manipulated variables $v(t)$ as

$$v(t) = -\rho^{-1}\bar{B}^T\bar{P}[\bar{A}\bar{x}(t) + \bar{B}v(t) + \bar{\Gamma}d(t)] + \rho^{-1}\bar{B}^Tg. \tag{4.7}$$

The previous expression can be solved with respect to $v(t)$, resulting in

$$v(t) = -(I + \rho^{-1}\bar{B}^T\bar{P}\bar{B})^{-1}\rho^{-1}\bar{B}^T\bar{P}\bar{A}\bar{x}(t) + (I + \rho^{-1}\bar{B}^T\bar{P}\bar{B})^{-1}\rho^{-1}\bar{B}^T[g - \bar{P}\bar{\Gamma}d(t)], \tag{4.8}$$

and it can be simplified into

$$v(t) = -\begin{bmatrix} K & K_I \end{bmatrix}\bar{x}(t) + v_{ff}(t), \tag{4.9}$$

in which v_{ff} is the vector of feedforward control variables. The computation depends on the partial derivative of the Lagrangian L with respect to the augmented system state $\bar{x}$ and on the co-state and closed-loop state equations. The computation is described in detail in [16] and the resulting expression for the feedforward control action v_{ff} is

$$v_{ff}(t) = K_{ff}d(t) = (I + \rho^{-1}\bar{B}^T\bar{P}\bar{B})^{-1}\rho^{-1}\bar{B}^T[\Phi - \bar{P}\bar{\Gamma}]d(t), \tag{4.10}$$

with

$$\Phi = -I + \bar{A}^T\bar{P}[I + \bar{B}\rho^{-1}\bar{B}^T\bar{P}]^{-1}\bar{B}\rho^{-1}\bar{B}^T - \bar{A}^{T^{-1}}\bar{A}^T\bar{P}[I + \bar{B}\rho^{-1}\bar{B}^T\bar{P}]^{-1}\bar{\Gamma}, \tag{4.11}$$

$$\bar{P} = \bar{A}^T\bar{P}[I + \bar{B}\rho^{-1}\bar{B}^T\bar{P}]^{-1}\bar{A} + \bar{Q} \tag{4.12}$$

The control law defined by (4.9) is represented in figure 4.2.
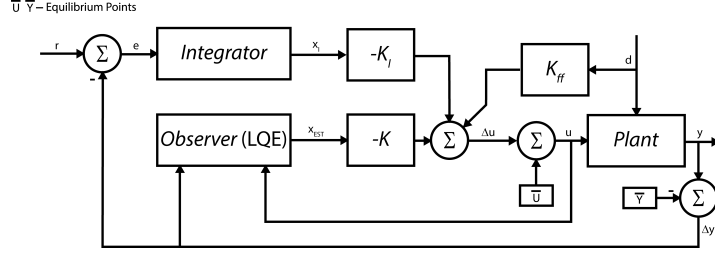
Figure 4.2: Schematic representation of a local controller.

4.2 Coordination procedure

After defining the control law for each local agent it is important to develop a coordination procedure that guarantees that the manipulated variables are computed taking into account information provided by neighbouring controllers. The distributed procedure considered in this study was defined in [3] and [22] and it is applied to serially chained systems, like the water canal. This coordination algorithm is an iterative procedure in which the local control agents communicate with each other to compute the corresponding manipulated variables. In order to define the algorithm, the letter j will be used to define the iterations, i to identify the subsystem and t is the discrete time. The procedure begins by initializing the gate positions with the previous values, while the flow of the lateral off-takes is read from the sensors,

$$d_{i,j=0}(t) = \begin{bmatrix} v_{i-1}(t-1) & v_i(t-1) & v_{i+1}(t-1) & Q_{i-1}(t) & Q_i(t) & Q_{i+1}(t) \end{bmatrix}^T. \tag{4.13}$$

During a number of predefined iterations n_I, the expression (4.9) is used to compute the new manipulated variables $v_{i,j}(t)$,

$$v_{i,j}(t) = - \begin{bmatrix} K & K_I \end{bmatrix} \bar{x}(t) + K_{ff} d_{i,j}(t). \tag{4.14}$$

When the n_I-th iteration is performed, the optimal manipulated variables are defined as $v_{i,opt}(t) = v_{i,n_I}(t)$. Although the disturbance vector d is composed by accessible disturbances associated with the gate positions v and with the flows of lateral off-takes Q, the iterative procedure considers only the gate positions, and thus it can be written as

$$v_{i,j}(t) = \begin{bmatrix} K & K_I \end{bmatrix} \bar{x}(t) + \begin{bmatrix} K_{ff,v} & K_{ff,Q} \end{bmatrix} d_{i,j}(t), \tag{4.15}$$

$$d_{i,j}(t) = \begin{bmatrix} v_j(t) \\ Q(t) \end{bmatrix} = \begin{bmatrix} v_{i-1,j}(t) & v_{i,j}(t) & v_{i+1,j}(t) & Q_{i-1}(t) & Q_i(t) & Q_{i+1}(t) \end{bmatrix}^T. \tag{4.16}$$

This iterative procedure is represented in figure 4.3, and since the local control agents find its optimal manipulated variable with knowledge of their neighbors decisions, this procedure converges to the Nash Equilibrium [22], a situation where no local controller benefits by changing only its manipulated variable. In order for the algorithm to reach convergence, the spectral radius of $K_{ff,v}$ needs to satisfy the condition

$$max|\lambda(K_{ff,v})| < 1, \tag{4.17}$$

where $\lambda(K_{ff,v})$ represents the eigenvalues of the matrix $K_{ff,v}$, defined as

$$K_{ff} = \begin{bmatrix} 0 & \Psi_1 \bar{\Phi}_{v,12} & 0 \\ \Psi_2 \bar{\Phi}_{v,22} & 0 & \Psi_2 \bar{\Phi}_{v,23} \\ 0 & \Psi_3 \bar{\Phi}_{v,32} & 0 \end{bmatrix}, \tag{4.18}$$
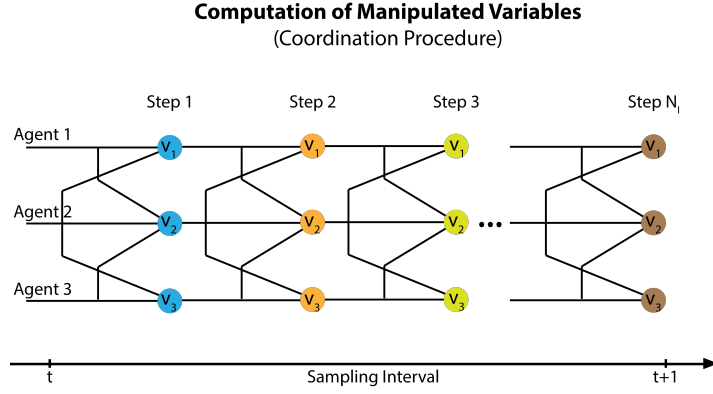
Figure 4.3: Schematic representation of the coordination procedure.

with $\Psi_i = -(\rho_i + \bar{B}_i^T \bar{P}_i \bar{B}_i)^{-1} \bar{B}_i^T \bar{P}_i$.

4.3 Parameter Tunning
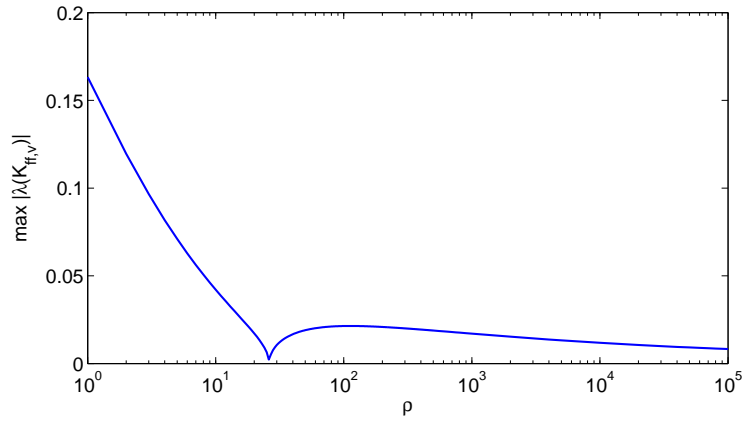
Figure 4.4: Spectral radius of $K_{ff,v}$ as a function of the quadratic cost weights ρ_i. It is assumed that the quadratic weights of the three pools have the same value

In order to verify the condition (4.17), several experiments were conducted with different values of quadratic weights ρ. In order to simplify the controller design it is assumed that the quadratic weights of the three pools, ρ_i, have the same value. The observer parameters are also equal for the three subsystems and these are defined as $R_{E,i} = 1 \times 10^3$ and $q_i = 1$. The number of iterations n_I considered was 10 and the results are represented in figure 4.4, in which it is possible to see that the condition is valid for ρ between 1 and 1×10^5. Maintaining $n_I = 10$ and the same values for the observer, the system response for different values of ρ_i is represented in figures 4.6 and 4.6. It is possible to see that the closed-loop response of the system is only acceptable for $\rho_i \geq 5 \times 10^3$, such that for lower weights the system response was unstable, and thus the value defined was $\rho_i = 1 \times 10^4$. The results of an experiment conducted with all the parameters defined is represented in figures 4.7 and 4.8.
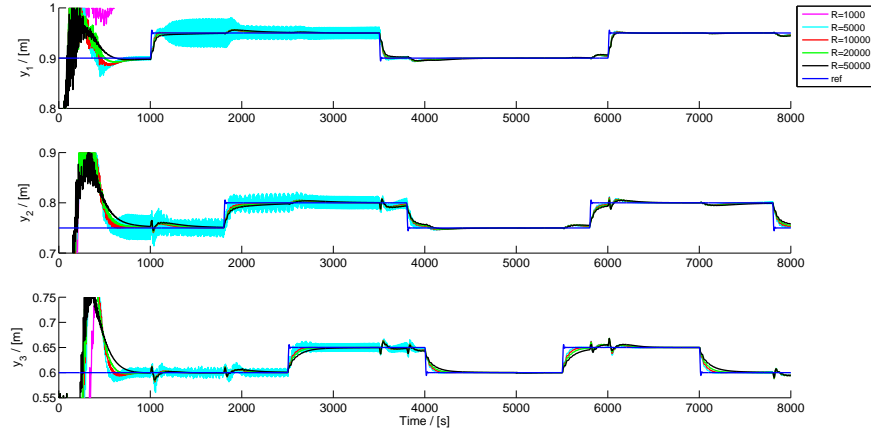
Figure 4.5: Closed-loop response of the SIMULINK non-linear canal model, considering the system composed of the first three gates, controlled by the distributed LQG controller for several values of ρ_i, with $q = 1$, $n_I = 10$ and $R_E = 1000$. The fourth gate was kept at its equilibrium position. (Output and Reference signals)
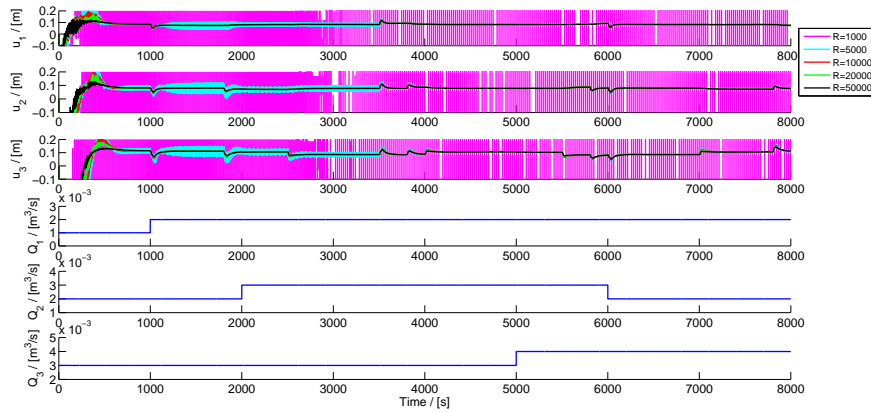
Figure 4.6: Closed-loop response of the SIMULINK non-linear canal model, considering the system composed of the first three gates, controlled by the distributed LQG controller for several values of ρ_i, with $q = 1$, $n_I = 10$ and $R_E = 1000$. The fourth gate was kept at its equilibrium position. (Input signals and flows of lateral off-takes)

4.4 Adaptive Distributed LQG controller

The adaptive distributed control strategy based on the D-LQG algorithm previously defined follows the same principles of the adaptive multi-variable algorithm in section 3.5. The strategy is therefore based in a two-step sequence in which first the RLS estimates the parameters using the input and output data obtained from the plant, and after that, a model defined with those estimates is considered in the controller step, with the D-LQG algorithm.

During a pre-specified period of time t_I, only the identification step is working, in order to guarantee that the estimates are closer to convergence by the time the controller is activated. Without guaranteeing this, it is possible to occur either stability or identification issues, and this was one of the major difficulties while dimensioning the controller. During this period of time, the system is excited by a PRBS signal of amplitude $0.01m$ around the equilibrium points, which is maintained active during the
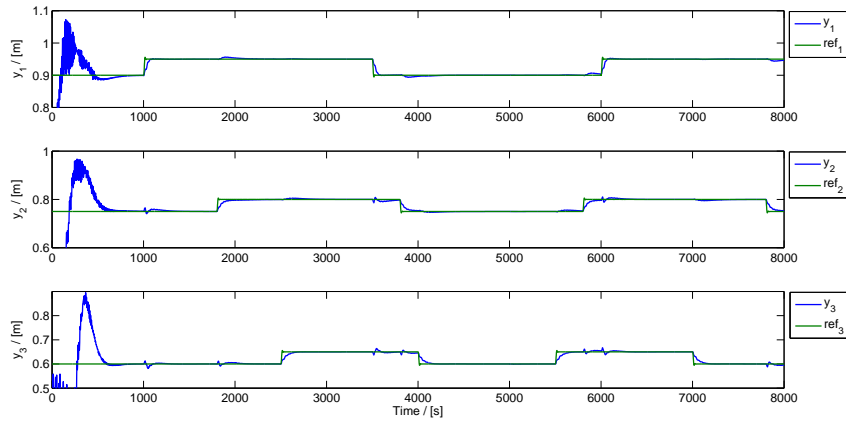
Figure 4.7: Closed-loop response of the SIMULINK non-linear canal model, considering the system composed of the first three gates, controlled by the distributed LQG controller with $\rho_i = 10000$, $q = 1$, $n_I = 10$ and $R_E = 1000$. The fourth gate was kept at its equilibrium position. (Output and Reference signals)
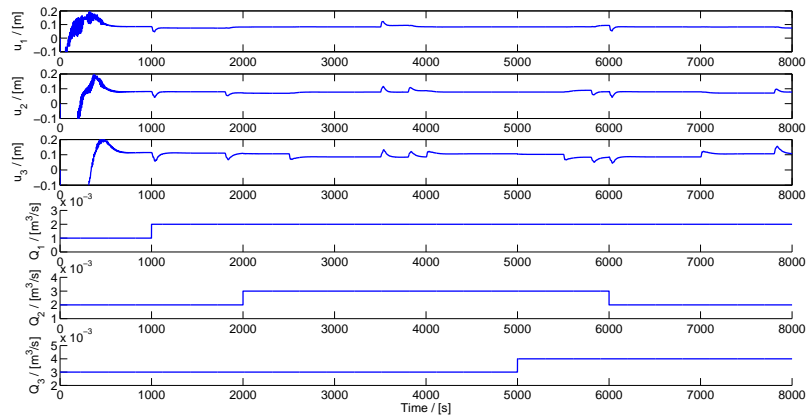
Figure 4.8: Closed-loop response of the SIMULINK non-linear canal model, considering the system composed of the first three gates, controlled by the distributed LQG controller with $\rho_i = 10000$, $q = 1$, $n_I = 10$ and $R_E = 1000$. The fourth gate was kept at its equilibrium position. (Input signals and flows of lateral off-takes)
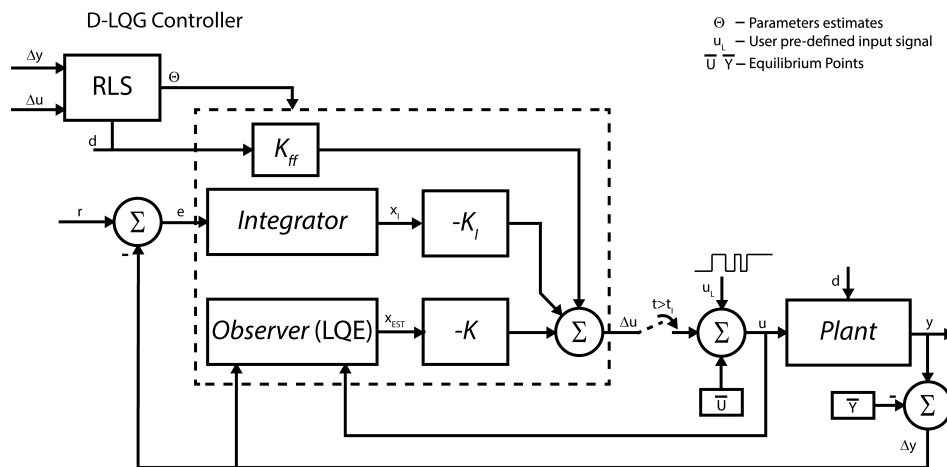
Figure 4.9: Schematic representation of the adaptive D-LQG controller with integral action applied to a linear incremental model.

whole experiment to guarantee that the system is being excited, in order to prevent identification issues. The period of time in which only the identification step is activated was defined as $t_I = 4 \times 10^4 s$. The controller parameters were defined as $\rho_i = 5 \times 10^4$, $R_{E,i} = 1 \times 10^3$, $q_i = 1$, and $n_I = 10$. The parameters were equal for the three subsystems and the algorithm is defined in detail bellow and it is represented in figure 4.9.

Algorithm 4.1 Adaptive D-LQG
___
Initialization of parameter estimates $(\theta_1, \theta_2, \theta_3)$ and respective covariance matrices (P_1, P_2, P_3).

for each time instant t
    Computation of parameters $\theta_i(t)$, using algorithm 2.1 and input and output data $(\Delta u(t), \Delta y(t))$.
    Define the augmented models of each subsystem i, using the parameter estimates $\Theta_i(t)$
        $\bar{x}_i(t+1) = \bar{A}_i \bar{x}(t) + \bar{B}_i v_i(t) + \bar{\Gamma}_i d_i(t)$
        $\Delta y(t) = \bar{C}_i \bar{x}_i(t)$
  if $t > t_I$ **then**
    Regulator and Observer parameters: $\rho_i, R_{E,i}, q_i$
    Number of iterations: n_I
    Computation of Regulator and Integrator gains $\begin{bmatrix} K_i & K_{I,i} \end{bmatrix}$
    Computation of Observer gain M_i
    Computation the state estimates $\hat{x}_i$
    $v_{fb,i}(t) = \begin{bmatrix} K_i & K_{I,i} \end{bmatrix} \begin{bmatrix} \hat{x}_i(t) \\ x_{I,i}(t) \end{bmatrix}$
    Initialize manipulated variables $v_{i,j=0}(t) = v_i(t-1)$
    for $j = 1 : n_I$
        $v_{ff,i,j}(t) = \begin{bmatrix} K_{ff,i,v} & K_{ff,i,Q} \end{bmatrix} d_{i,j}(t)$
    end
    Computation of manipulated variables: $v_i(t) = v_{fb,i}(t) + v_{ff,i,n_I}(t)$

The results obtained during an experiment with the algorithm 4.1 are represented in figures 4.10, 4.11 and 4.12. In figures 4.10 and 4.11 it is represented the open-loop and closed-loop response of the system controlled by the adaptive D-LQG algorithm. It is possible to verify how the outputs of the three pools converge to the reference signals, with the parameter estimates represented in figure 4.12. Comparing with the non-adaptive algorithm, the system response adaptive strategy appears to be a close approximation, even with an additional small amplitude PRBS signal in the input of the system. Regarding the parameters estimates, these begin with the values found in section 2.4, and when the controller step is activated they are closer to convergence.
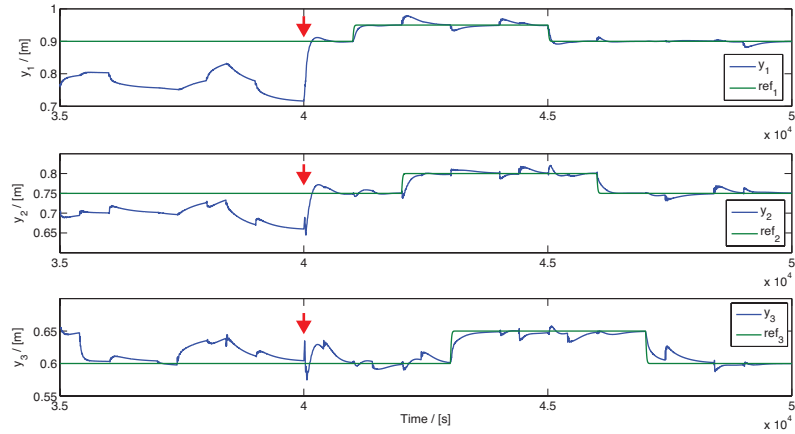
Figure 4.10: Open-loop and closed-loop response of the SIMULINK non-linear canal model, considering the system composed of the first three gates, controlled by the adaptive D-LQG controller with $\rho = 5 \times 10^4$, $q = 1$ and $R_E = 1 \times 10^3$. The fourth gate was kept at its equilibrium position. The controller step was switched on at $4 \times 10^4 s$. (Output and Reference signals)
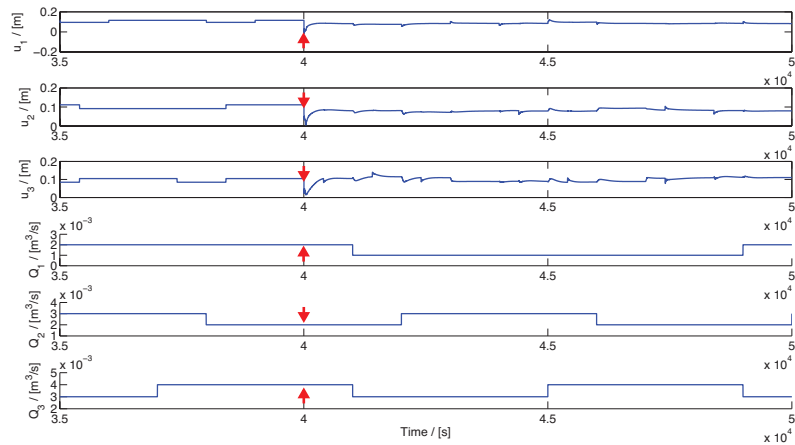
Figure 4.11: Open-loop and closed-loop response of the SIMULINK non-linear canal model, considering the system composed of the first three gates, controlled by the adaptive D-LQG controller with $\rho = 5 \times 10^4$, $q = 1$ and $R_E = 1 \times 10^3$. The fourth gate was kept at its equilibrium position. The controller step was switched on at $4 \times 10^4 s$. (Input signals and flows of lateral off-takes)
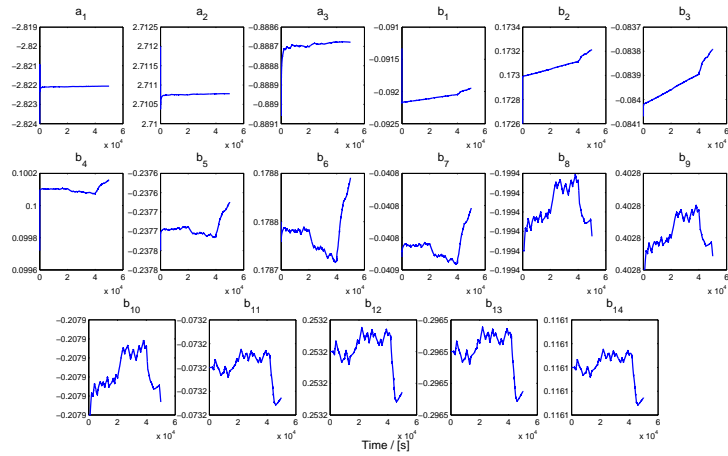
(a) Parameter estimates of the first subsystem.
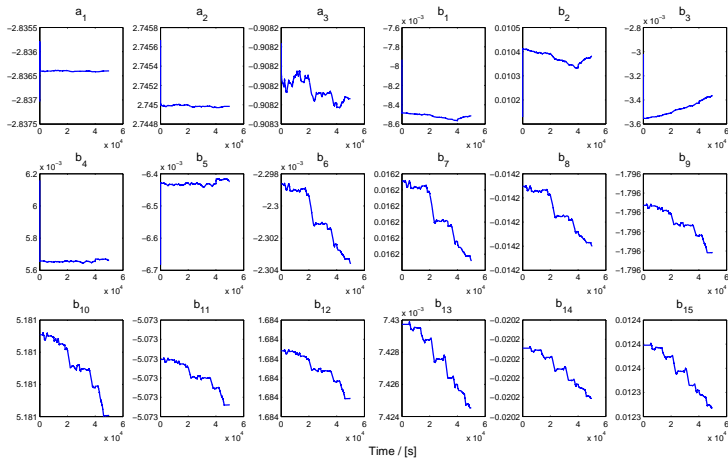
(b) Parameter estimates of the second subsystem.
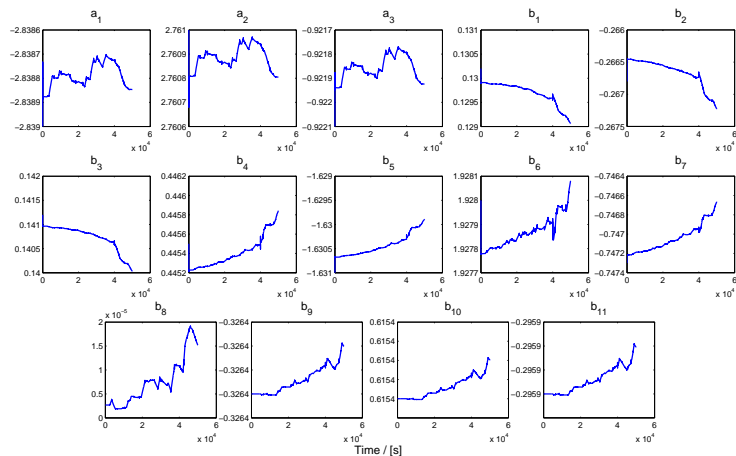
(c) Parameter estimates of the third subsystem.

Figure 4.12: Representation of the MISO parameters estimates using the adaptive D-LQG controller. The control step is switched on at $4 \times 10^4 s$.

5

Model Predictive Control

Contents

This chapter provides the theoretical background required to develop the adaptive and non-adaptive MPC controllers, with a brief introduction followed by the algorithms description for both SISO and MIMO linear systems. This chapter is also dedicated to the adjustment of control parameters and features examples obtained with both linearized and canal models.

5.1   Problem formulation of Model Predictive Control

Model Predictive Control (MPC) or Receding Horizon Control (RHC) is a feedback strategy in which the manipulated variable is obtained using predictions of the system dynamics that take into account its model, with the advantage of handling both input and state constraints. Its control law consists in the optimization of a quadratic cost function that depends on the forecasts of the system behavior during a predefined finite horizon N.

The application of several MPC strategies to control the water delivery canal considered in this study [6], [17], provide an interesting background and motivation to the development of adaptive strategies. The algorithms developed take into account the linear incremental models identified around an equilibrium point of the system, as shown in Chapter 2. The problem formulation of linear MPC algorithms is described in [30] and it begins by considering a linear model such as

$$x(t+1) = Ax(t) + Bu(t), \quad y(t) = Cx(t), \quad u_{min} \leq Du(t) \leq u_{max}, \quad y_{min} \leq Cx(t) \leq y_{max}. \quad (5.1)$$
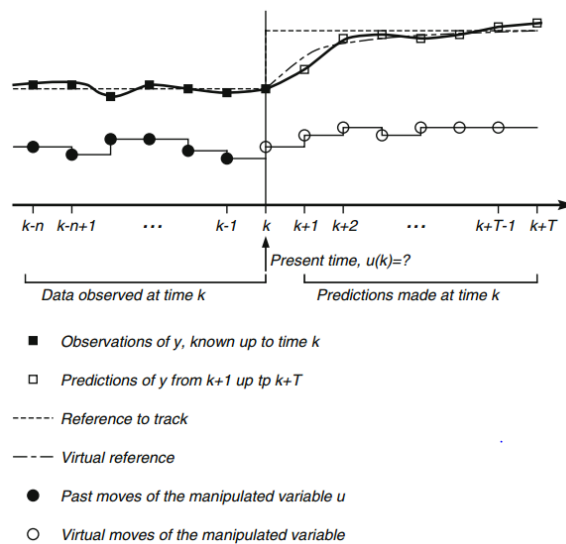
Figure 5.1: Representation of the variables in Model Predictive Control [31].

Figure 5.1 (Lemos, J. M. et al., 2014) represents the variables taken into account by the MPC control algorithm and the process of computing the manipulated variable based on predictions of the system behavior. The objective is to drive the system to a goal state by controlling the input and output variables of the system, u and y, taking into account the input and output constraints. Although the cost function considered in the design of the LQG controllers is applied to an infinite horizon, in

this alternative one considers a quadratic cost function associated to a predefined horizon N [30] that slides alongside with the current state estimates, while taking into account the constraints. The objective is to find a sequence of manipulated variables that minimizes the cost function

$$J_N = \sum_{i=1}^{N} (y(t+i) - r(t+i))^T (y(t+i) - r(t+i)) + \rho u^2(t+i-1). \tag{5.2}$$

The advantages of the RHC strategy when compared with the infinite horizon control are described in [32], and include the handling of constraints, the applicability to a large class of systems, and the fact that it only depends on future values of the reference and of the system parameters in a finite time interval.

5.2 Single gate MPC controller

In order to design a MPC controller for the first pool and corresponding gate, one needs to consider the SISO incremental model defined in (2.3). Since the computation of manipulated variables takes into account an incremental model, the input and output of the system are denoted as Δu and Δy, which represent the variations around an equilibrium point.

5.2.1 SISO Predictor Model

The cost function of the MPC algorithm requires a model for the system predictor. In order to define the predictor one needs to write the SS model equations for the several t time instants

$$
\begin{aligned}
x(t+1) &= Ax(t) + B\Delta u(t) \\
x(t+2) &= A^2 x(t) + AB\Delta u(t) + B\Delta u(t+1) \\
x(t+3) &= A^3 x(t) + A^2 B\Delta u(t) + AB\Delta u(t+1) + B\Delta u(t+2) \\
x(t+i) &= A^i x(t) + \sum_{j=1}^{i} A^{j-1} B\Delta u(k+i-j).
\end{aligned}
\tag{5.3}
$$

The predictor i steps ahead is obtained using the relationship between Δy and x described by (2.3), which results in

$$\Delta y(t+i) = CA^i x(t) + \sum_{j=1}^{i} CA^{j-1} B\Delta u(t+i-j), \tag{5.4}$$

in which $w_j = CA^{j-1}B$ are denoted as the Markov parameters.

In order to simplify the notation and to design the MPC controller, one may write the predictor model in matrix form. Let Y be a vector with the system outputs from time instant $t+1$ to $t+i$ and ΔU a vector with the manipulated variables computed from instant t to $t+i-1$, defined as

$$Y = \begin{bmatrix} \Delta y(t+1) \\ \Delta y(t+2) \\ ... \\ \Delta y(t+i) \end{bmatrix}, \quad \Delta U = \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ ... \\ \Delta u(t+i-1) \end{bmatrix}. \tag{5.5}$$

One may also define $\mathbf{W}$ and $\mathbf{\Pi}$ as

$$\mathbf{W} = \begin{bmatrix} w_1 & 0 & ... & 0 \\ w_2 & w_1 & ... & 0 \\ ... & ... & ... & ... \\ w_i & w_{i-1} & ... & w_1 \end{bmatrix}, \quad \mathbf{\Pi} = \begin{bmatrix} CA \\ CA^2 \\ ... \\ CA^i \end{bmatrix}. \tag{5.6}$$

Since it is not possible to access the system state with the SIMULINK model, one needs to consider an estimate of the state $\hat{x}$ obtained using a LQE introduced in section 3.1.2. With the matrices defined in (5.5) and (5.6), one may write an expression for the model predictor as

$$Y = \mathbf{\Pi}\hat{x}(t) + \mathbf{W}\Delta U. \tag{5.7}$$

5.2.2 Receding horizon cost function

Taking into account the predictor model defined in (5.4), the RHC cost function (5.2) depends on two parameters, the finite horizon N and on the cost weight ρ. Both these parameters need to be defined before the minimization of the cost function. In the case of ρ, the value considered took into consideration the parameter tunning process introduced in the LQG controller section, whereas for the time horizon, several experiments were conducted with different values of N, in order to determine its influence.

One may define the error e of the system output as

$$e = \frac{\sum_{t=1}^{N}(y(t) - r(t))^2}{t_s}, \tag{5.8}$$

with r denoting the reference and t_s the simulation time, in order to analyze how the influence of N in the controller design. The results obtained are represented in figure 5.2, in which one may see e decreasing with the horizon. For $N < 5$ the output did not track the reference. In general, the value of e tends to decrease with an increasing horizon, although the value increases slightly for some values of N. For $N \geq 5$ the system is stable and since the computational load increases with the horizon there is no need to select a higher value of N.
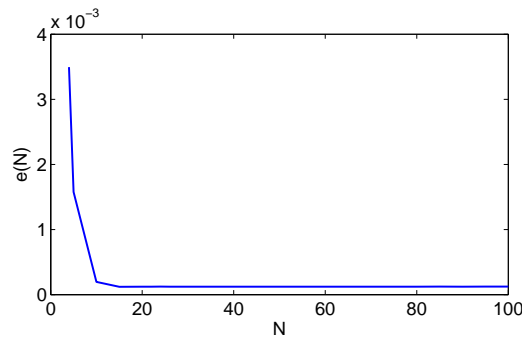
Figure 5.2: Variation of the error e with the horizon N for the SISO model with a quadratic weight $\rho = 1000$.

With the parameters defined, one needs to take into consideration the minimization of the RHC cost function. Since this function depends on the value of the state estimate $\hat{x}$ in the current time

instant, which is known a priori, and on the manipulated variables, one may consider it as a function of the sequence of manipulated variables. The solution for the minimization problem depends on whether the constraints are considered or not.

Since in this dissertation the adaptive controllers require an identification step followed by the computation of the manipulated variables, considering constraints would increase the computational load and thus these are not taken into account. Without constraints, the problem has an algebraic solution derived in [31], and it is also possible to determine the sequence of manipulated variables using MATLAB function *fminunc*, that finds the minimmum of a problem defined by the user. The computation of manipulated variables for the SISO model is accomplished by MATLAB function *fminunc*.

5.2.3 Integral action

After performing several examples, the results obtained were similar to the ones shown in [6], with large static error between the system output and reference. The solution found to prevent this problem was to introduce an integrator in series with the MPC controller, and thus the computed manipulated variable is given by

$$z(t+1) = \Delta u(t) + z(t), \tag{5.9}$$

and the SISO model seen by the controler is represented by

$$\begin{bmatrix} x(t+1) \\ z(t+1) \end{bmatrix} = \begin{bmatrix} A & B \\ \underline{0} & 1 \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Delta u(t), \quad y(t) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix}, \tag{5.10}$$

which can be written in a more compact notation as

$$\bar{x}(t+1) = \bar{A}\bar{x}(t) + \bar{B}\Delta u(t), \quad \Delta y(t) = \bar{C}\bar{x}(t), \tag{5.11}$$

where $\bar{x}$ denotes the augmented state of the system.

From the LQE point-of-view, the system is now described by

$$x(t+1) = Ax(t) + Bz(t), \quad \Delta y(t) = Cx(t). \tag{5.12}$$

The matrices $\mathbf{\Pi}$ and $\mathbf{W}$ are now computed with the augmented model matrices $\bar{A}$, $\bar{B}$, and $\bar{C}$ defined in (5.10).

The results obtained with the linearized model after introducing integral action are shown in figure 5.3, in which the system output converges to the reference without static error. In figure 5.4 it is possible to verify the influence of the integral action with a comparison between the closed-loop system response obtained with and without the integrator.

While implementing the MPC algorithm with integral action into the SIMULINK canal model, the closed-loop system response shows a static error of small amplitude (around $0.005m$) when varying with respect to the equilibrium point, as shown in figure 5.5. The first gate is excited by a sequence of
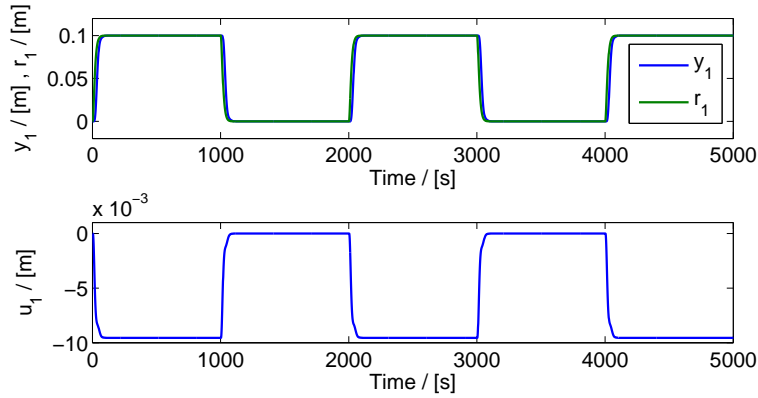
Figure 5.3: Closed-loop response of the linearized SISO model with parameters $\rho = 1000$ and $N = 25$.

square waves as reference signal while the other gates remain in their equilibrium positions. Despite the static error, the system output tracks the reference signal and, by looking to the results obtained with the linearized SISO model, the advantages of introducing integral action are noticeable.

The static error present in the closed-loop response of the SIMULINK canal model persisted in all the experiments conducted with the MPC algorithms (both adaptive and non-adaptive strategies), and a possible explanation to this error is still to be found.
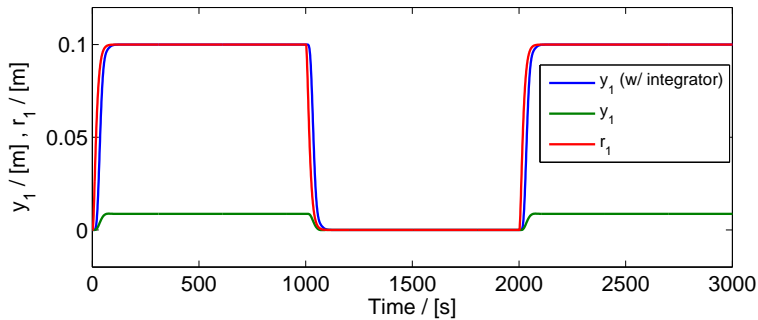
Figure 5.4: Comparison between the closed-loop response of the linearized SISO model with parameters $\rho = 1000$ and $N = 25$ with and without integral action.

5.3 Adaptive single gate MPC controller

In the previous section it was assumed that the system model was known *a priori*, after conducting several experiments to identify its parameters as shown in chapter 2. In order to develop an adaptive control strategy for the SISO MPC controller a similar approach to the one used in section 3.3 is required. The algorithm is now divided into two steps:

- **Identification**: Estimation of SISO model parameters using the RLS algorithm with variable exponential forgetting.

- **Control**: Execution of the MPC control strategy defined in section 5.2 with the resulting SISO model from the previous setp.
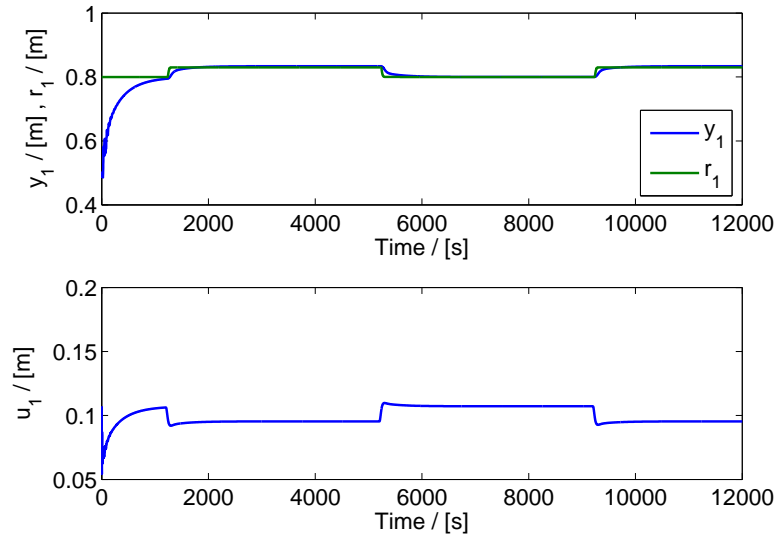
Figure 5.5: Closed-loop response of the SISO model simulated with the SIMULINK canal model (first gate) with parameters $\rho = 1000$ and $N = 25$. The remaining gates were kept on their equilibrium positions.

In order to provide better estimates of the system parameters before connecting the controller, during an initial pre-specified period of time only the identification step is working. At each time instant t, the RLS algorithm provides estimates of the system parameters using the current input and output data, Δu and Δy retrieved from the SIMULINK canal model. After the initial period of time in which only this step is activated, in order to ensure that the estimates are converging, the controller step begins by defining an augmented model with the matrices defined in (5.10) using the estimates obtained in the previous step.

The augmented model is then used to compute the predictor matrices $\mathbf{\Pi}$ and $\mathbf{W}$ defined in (5.6). The predictor (5.7) is then used to minimize the RHC cost function using *fminunc*. This sequence of steps is represented in figure 5.6.

The results obtained with the Adaptive MPC algorithm are shown in figures 5.7, 5.8 and 5.9. Figure 5.7 shows both the open-loop and closed-loop system response from $t = 6 \times 10^4 s$ to $t = 8 \times 10^4 s$, in which at $t_I = 7 \times 10^4 s$ the controller is switched on. During the period of time in which only the identification step is working, the system is excited by a pre-defined input signal in order to estimate its parameters. When the controller is connected, a PRBS signal of small amplitude, $0.001m$, is added to the computed manipulated variable to provide a more exciting input signal. As it is possible to verify, even with the introduction of integral action, the output tracks the reference with static error.

In figure 5.8 the parameters estimates obtained during the experiment are represented and it is possible to verify that when the controller is switched on, the estimates are converging, even though parameters, a_4 and b_2, appear to require more time and excitation to converge. Figure 5.9 shows how the controller reacts to existing disturbances, which in this case are caused by the opening of the lateral offtake valve at $t = 7.2 \times 10^4 s$. The problem with the integral action is also reflected in this
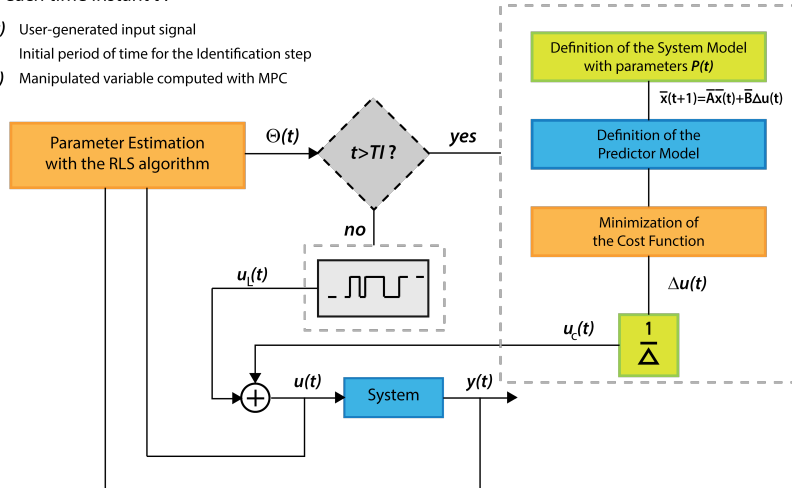
Figure 5.6: Schematic representation of the Adaptive MPC algorithm applied to both SISO and centralized MIMO models.
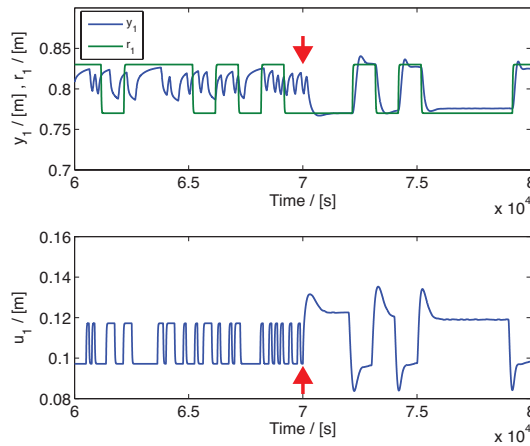
Figure 5.7: Open-loop and closed-loop response of the SISO model simulated with the SIMULINK canal model (first gate) with parameters $\rho = 1000$ and $N = 25$. In this adaptation strategy, during the first $7 \times 10^4 s$ only the identification step is working and then the MPC controller is switched on. The remaining gates were kept on their equilibrium positions.
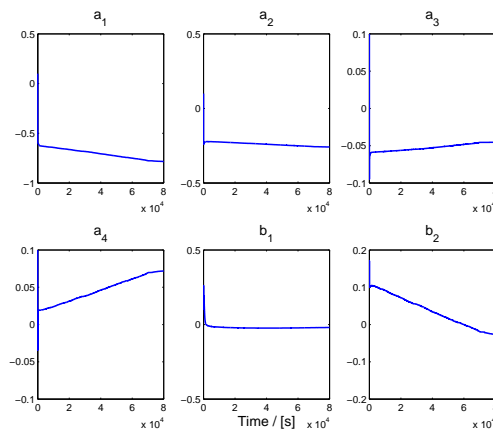
Figure 5.8: Representation of the SISO parameters estimates using the Adaptive MPC controller.
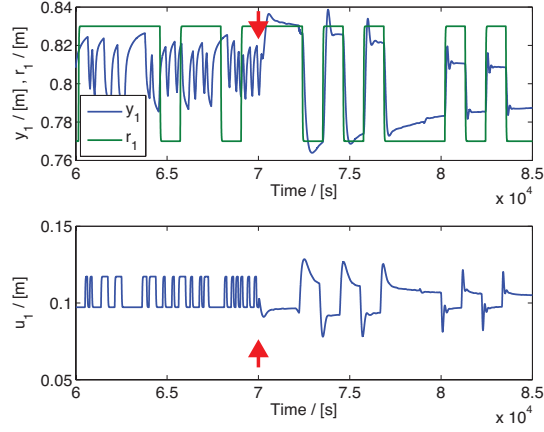
Figure 5.9: Open-loop and closed-loop response of the SISO model simulated with the SIMULINK canal model (first gate) with parameters $\rho = 1000$ and $N = 25$. In this adaptation strategy, during the first $7 \times 10^4 s$ only the identification step is working and then the MPC controller is switched on. The lateral offtake valve is open at $7.2 \times 10^4 s$ with a constant flow of $0.001 m^3/s$. The remaining gates were kept on their equilibrium positions.

example, since even though the controller reacts to the valve opening, the water level in the first pool tends to decrease.

5.4   Multiple gate MPC controller

The previous sections were dedicated to the theoretical background and definition of a MPC controller applied only to the first gate of the water canal while the remaining gates were kept at their equilibrium positions. It is possible, using a similar formulation, to design a centralized MPC algorithm to be implemented into the first three gates, using the MIMO model defined in (2.8).

The introduction of integral action with the centralized MIMO MPC requires an augmented system state defined as

$$\bar{x}(t) = \begin{bmatrix} x_1(t) & z_{11}(t) & z_{12}(t) & x_2(t) & z_{21}(t) & z_{22}(t) & z_{23}(t) & x_3(t) & z_{32}(t) & z_{33}(t) \end{bmatrix}^T, \quad (5.13)$$

in which z_{ij} with $i = j$ are the manipulated variables defined in (5.9) and z_{ij} with $i \neq j$ are copies of the manipulated variables of subsystem j associated with the i-th subsystem. The augmented matrices are written as

$$\bar{A}_i = \begin{bmatrix} A_i & B_i \\ \underline{0} & I \end{bmatrix}, \quad \bar{B}_i = \begin{bmatrix} \underline{0} \\ I \end{bmatrix}, \quad \bar{C}_i = \begin{bmatrix} C_i & \underline{0} \end{bmatrix}, \quad (5.14)$$

with B_i defined as a matrix with the input matrices B_{ij} of the corresponding subsystem i as entries. The augmented state matrices are then used to compute the predictor model followed by the cost function minimization using *fminunc*, as described in section 5.2.

The cost-function is now multi-variable and it is necessary to define values for the quadratic weights related with each subsystem ρ_i and for the time horizon N. For the quadratic cost weights
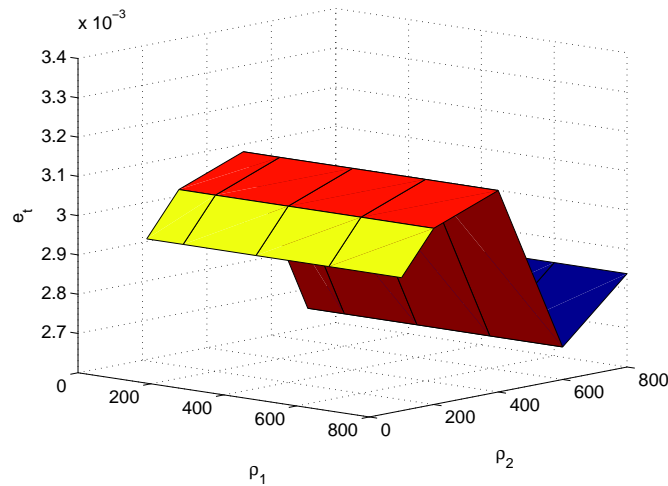
Figure 5.10: Variation of the total output error e_t with different combinations of quadratic cost weights ρ_i. In this experiment the value of ρ_3 was fixed in 400 and $N = 20$.

several experiments were conducted with different combinations of values of ρ_i from 100 to 800. In figure 5.10 it is represented the variation of the total output error for different combinations of ρ_1 and ρ_2, keeping fixed the value of ρ_3. The combination of quadratic cost weights that results in a smaller output error is $\rho_1 = 200$, $\rho_2 = 600$ and $\rho_3 = 400$. A curious fact regarding the results obtained is that the total output error e_t appears to not be influenced by the first quadratic cost weight ρ_1. The influence of the horizon is studied in this case, in terms of the total error variable e_t, which is computed as

$$e_i = \frac{\sum_{t=1}^{N}(y_i(t) - r_i(t))^2}{t_s}, \quad e_t = \frac{e_1 + e_2 + e_3}{3}, \tag{5.15}$$

in which e_i is the output error of the i-th subsystem and t_s is the simulation time. Several experiments were therefore conducted with different values of N in order to determine how it influences the system output error. The results obtained are represented in figure 5.11 in which the output errors of the three subsystems and of the total system are represented for several values of N, between 10 and 100. For $N < 10$, the higher-amplitude oscillations in the output signal during the transitory regime increase the value of the output error and thus these values were not represented. Due to the computational load of the centralized MPC algorithm, it is not necessarily better to select the value of N for which the output error is minimum. Since for N between 30 and 40 the total output error appears to be closer to its minimum value, the selected value for the horizon N is 35.

With values defined for N and for the weights ρ_i two experiments were conducted, the first to test the reference tracking and the second one to test the rejection of disturbances. In figures 5.12 and 5.13 it is shown how the output converges to the reference, in the linearized model, and the corresponding input signals. The interaction between subsystems in the centralized model is seen when the water level in each pool varies, which results in variations in the water level of neighbouring subsystems. When this occurs, the controller acts upon the disturbance in order to drive the output towards the reference. In figures 5.14 and 5.15 the experiment was conducted in the SIMULINK canal model. In both cases it is possible to see that the outputs of the three subsystems converge to the
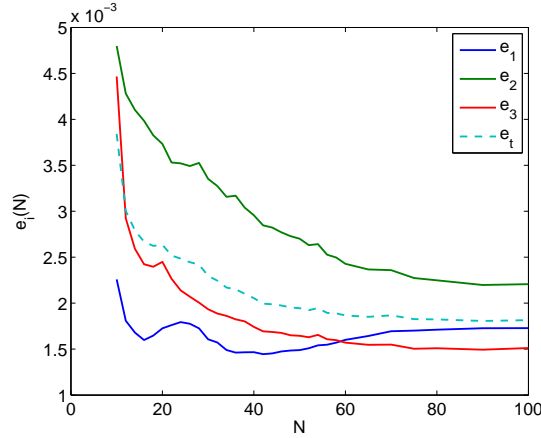
Figure 5.11: Variation of the error e_i of the i-th subsystem with the horizon N for the MIMO model with quadratic weights $\rho_1 = 200, \rho_2 = 100, \rho_2 = 200$.

respective reference and how these systems interact when the input/output of one varies. In the case of the SIMULINK canal model there is again some static error when the output varies alongside its equilibrium point. The error between the output and reference signals seems to vary accordingly with the current water levels of the subsystems, which may indicate that it is influenced by the interaction between control agents.
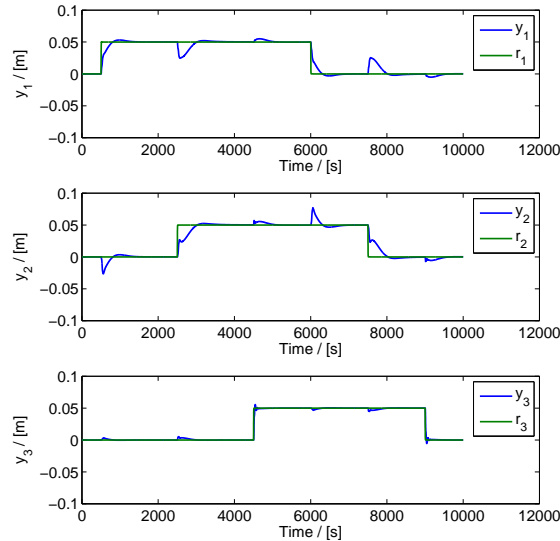
Figure 5.12: Closed-loop response of the linearized MIMO model with parameters $\rho_1 = 200, \rho_2 = 600, \rho_3 = 400$ and $N = 35$. (Output and Reference signals)

The experiment conducted begins with all three gates in their equilibrium position followed by the opening of the first gate in order to track the reference. It is possible to see how the second system reacts to this variation ($t = 500s$), in which the manipulated variable is computed in order to compensate the disturbance related with the interaction between subsystems. At the same time instant, the third gate is mostly affected by the disturbance in the second pool.
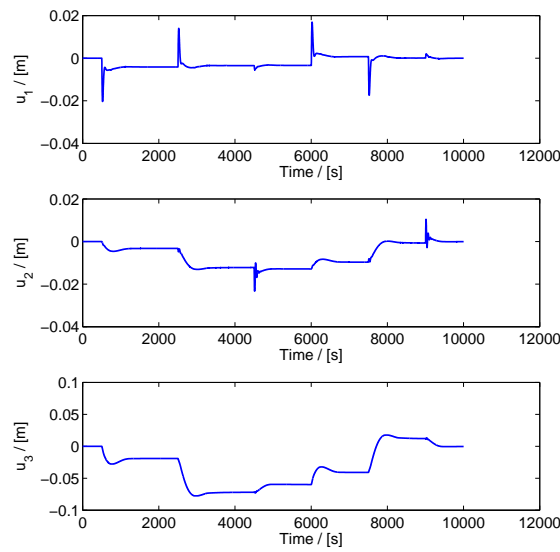
Figure 5.13: Closed-loop response of the linearized MIMO model with parameters $\rho_1 = 200, \rho_2 = 600, \rho_3 = 400$ and $N = 35$. (Input signal)
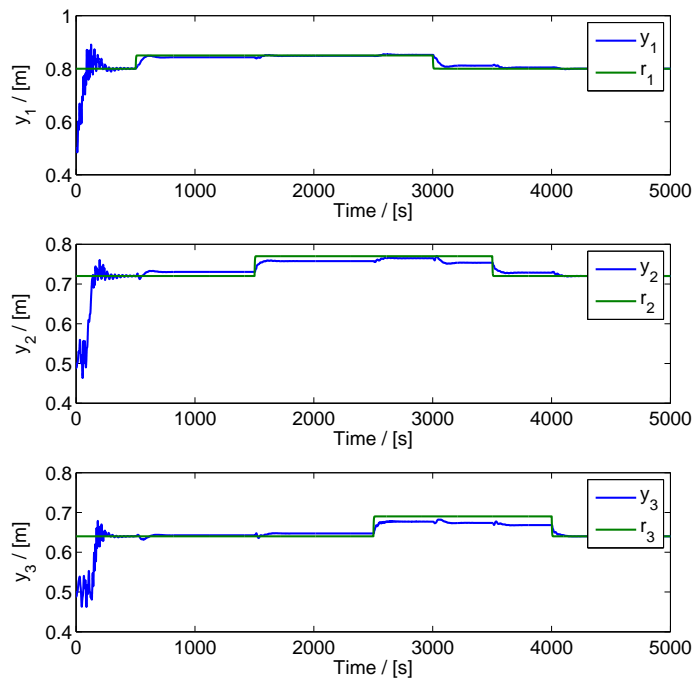
Figure 5.14: Closed-loop response of the MIMO model in an experiment conducted in the SIMULINK canal model with parameters $\rho_1 = 200, \rho_2 = 600, \rho_3 = 400$ and $N = 35$ (Output and Reference signals)

Compared with the algorithm presented in section 5.2, the computational load and time increases when applying this algorithm, mostly due to the larger dimension of the system. The minimization of the quadratic cost function is affected by the size of N, since this parameter influences the matrices used in the calculations. In terms of computational load, an analytical solution could improve the
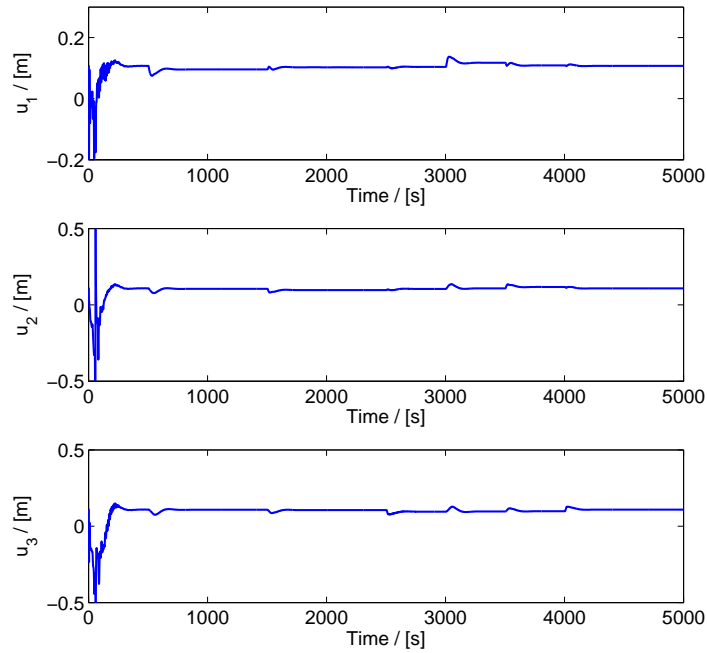
Figure 5.15: Closed-loop response of the MIMO model in an experiment conducted in the SIMULINK canal model with parameters $\rho_1 = 200, \rho_2 = 600, \rho_3 = 400$ and $N = 35$ (Input signals)

performance of the algorithm, since it is no longer required to use MATLAB function *fminunc*.

5.5   Adaptive multiple gate MPC controller

An adaptive version of the centralized multi-variable MPC algorithm is introduced in this section, with a similar structure as the one represented in figure 5.6. Regarding the algorithm introduced in the previous section, nothing changes in the control step, or to be more precise in the computation of the manipulated variables Δu_i, which are still obtained with MATLAB function *fminunc*. The identification step takes into account the parameters estimates of the three subsystems individually. During the execution of the algorithm, at each time instant t there are three executions of the RLS algorithm associated with the three subsystems that take into account the input and output data required to estimate the parameters. This way it is possible to define the system matrices as in (3.14).

Although this strategy has the advantage of being adaptable to the dynamic behavior of the system, the computational load increases with the introduction of the identification step. In the previous section it was already verified this issue with the usage of *fminunc* and thus it is still necessary to have this in mind to chose an appropriate value for N, which remains the same. During an initial period of time, $t_I = 1 \times 10^5 s$, only the identification step is working and thus the system is operating in open-loop with an input excitation defined by the user. After this period of time, with the parameter estimates obtained with the system open-loop response, the control step is switched on. This strategy guarantees that the parameters estimates are converging or at least close to convergence to prevent

issues such as instability.

The results obtained in the experiment are represented in figures 5.16 and 5.17. In order to handle with high-frequency oscillations, visible in experiments conducted with the previously defined quadratic cost weights, these values were increased in order to reduce the effect of the oscillations in the system response. The combination of quadratic weights considered in this example is $\rho_1 = 2000, \rho_2 = 2000$, and $\rho_3 = 1000$. Another possible solution for this issue is the inclusion of a low-pass filter. After $t_I = 1 \times 10^5 s$, when the controller step is activated it is possible to see that the output tracks the reference with static error and that the water levels of each pool are disturbed by the interactions between subsystems. For instance in $t = 1.13 \times 10^5 s$ the output signal of the second subsystem increases, following its reference and this produces a disturbance in the output signal of the first pool that ends up increasing. As for the third subsystem, the variations in the second pool appear to not introduce a relevant disturbance in the output.

The parameters estimates identified with the RLS algorithm are represented in figure 5.18. During the experiment, while the input is being excited by a user predefined signal, the parameter estimates converge, and by the time the control step is activated these are close to convergence. The period of time in which only the identification step is working was increased, in comparison with the algorithm of section 5.3, since some of the parameters required more time to converge. In figure 5.18 the parameter estimates are represented during the time period that the control step is activated and it is possible to see the estimates varying in instants that correspond to the input excitations.
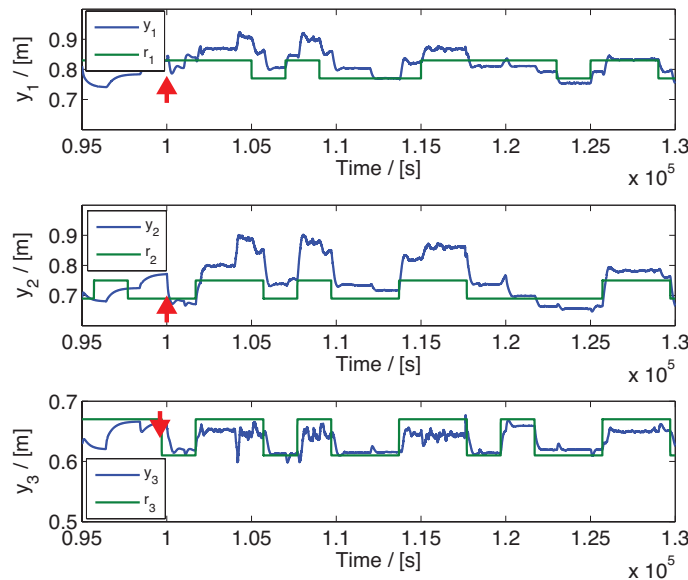
Figure 5.16: Open-loop and closed-loop response of the MIMO model in an experiment conducted in the SIMULINK canal model with Adaptive MPC algorithm with parameters $\rho_1 = 2000, \rho_2 = 2000, \rho_3 = 1000$ and $N = 35$. In $t = 10^5 s$ the controller is switched on. (Output and Reference signals)
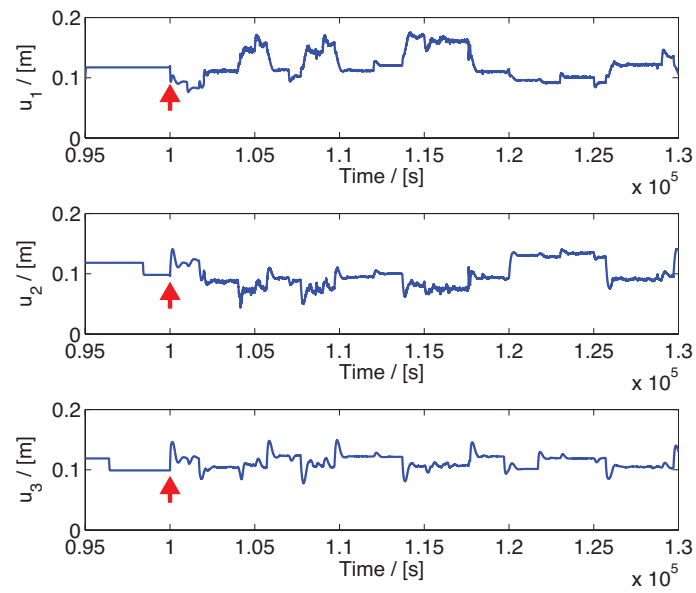
Figure 5.17: Open-loop and closed-loop response of the MIMO model in an experiment conducted in the SIMULINK canal model with Adaptive MPC algorithm with parameters $\rho_1 = 2000, \rho_2 = 2000, \rho_3 = 1000$ and $N = 35$. In $t = 10^5 s$ the controller is switched on. (Input signals)
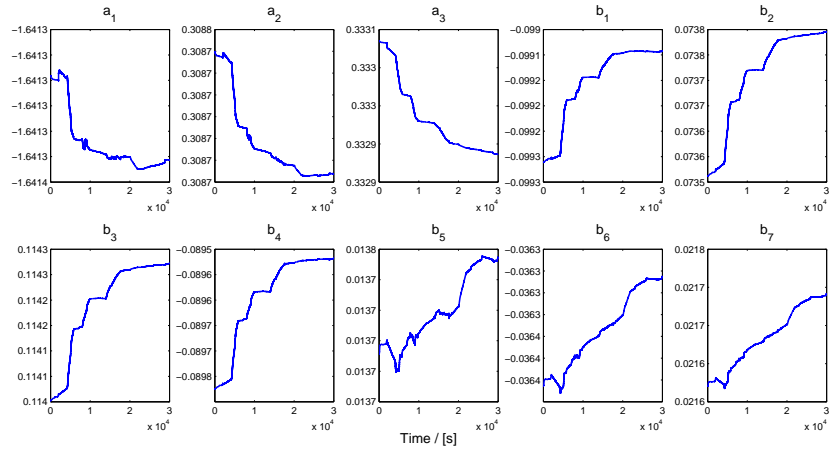
(a) Parameter estimates of the first subsystem.
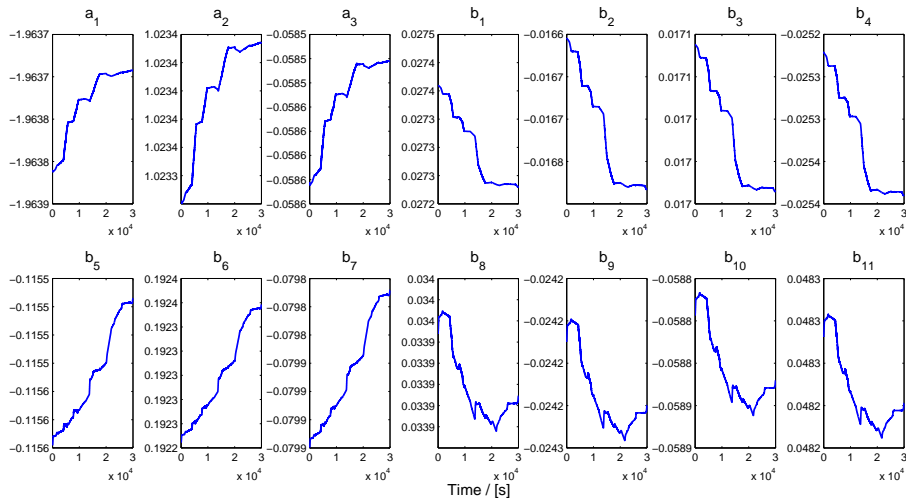
(b) Parameter estimates of the second subsystem.
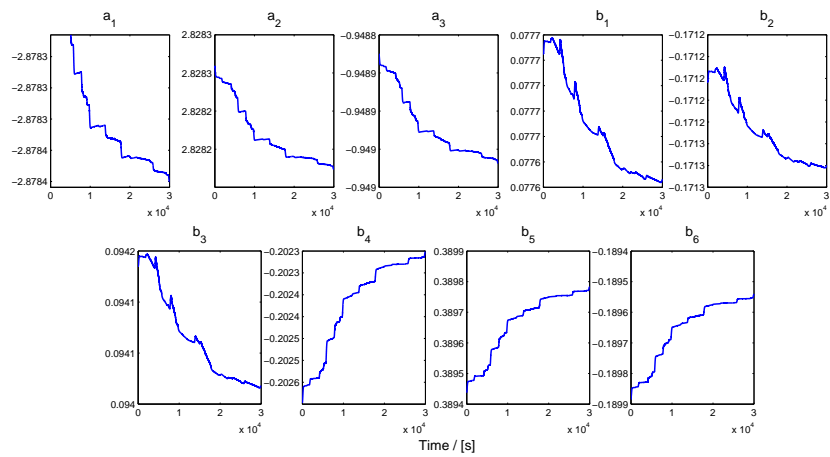
(c) Parameter estimates of the third subsystem.

Figure 5.18: Representation of the MIMO parameters estimates using the Adaptive MPC controller during the time period the control step is switched on.

6

Distributed Model Predictive Control

Contents

This chapter is dedicated to the definition of two Distributed Model Predictive Control strategies, with the objective of reaching a consensus between local controllers, by minimizing a global cost function that is the sum of local cost functions, in order to improve the performance and efficiency of the system. The first technique is based in an efficient distributed algorithm that requires less communications to achieve a desired goal [6], [33], and the second one is a coordination algorithm based on Game Theory concepts, similar to the one introduced in [17].

6.1 D-MPC based on the Distributed Alternating Direction Method of Multipliers

The first D-MPC strategy to be introduced in this dissertation was already described in [6] for input-output models. It is based on a distributed optimization algorithm named Distributed Alternating Direction Method of Multipliers (D-ADMM) that solves problems in networks of interconnected nodes, that represent the subsystems and that have a local cost function J_i associated with them [33]. Taking into consideration the predictor model defined in (5.7) the local cost functions J_i at time instant t are defined as

$$J_i(t) = \|\mathbf{\Pi}\hat{x}_i(t) + \mathbf{W}_{i-1,i}\Delta U_{i-1} + \mathbf{W}_{i,i}\Delta U_i + + \mathbf{W}_{i+1,i}\Delta U_{i+1} - R_i\| + \rho_i * \|\Delta U_i\|, \qquad (6.1)$$

in which R_i is the reference vector. The global cost function J is therefore the sum of all local cost functions J_i. It is assumed that the local cost functions are only accessible by the respective node and that communication is only allowed between neighbouring nodes. In order to minimize the global cost function each node needs to communicate with its neighbours with the purpose of reaching a consensus.

The network structure considered in the problem formulation of D-ADMM is represented in figure 4.1 in which a series of interconnected subsystems (nodes) Σ_i is associated to a local controller C_i with a local cost function that depends on the manipulated variable of the corresponding node and on copies of the manipulated variables of its neighbours $J_i(\Delta U_{i-1}, \Delta U_i, \Delta U_{i+1})$. Considering the multi-variable model defined by (2.8), it is possible to define the models of a subsystem i with integral effect as

$$\begin{bmatrix} x_i(t+1) \\ z_{i-1}(t+1) \\ z_i(t+1) \\ z_{i+1}(t+1) \end{bmatrix} = \begin{bmatrix} A_i & B_i \\ \underline{0} & I \end{bmatrix} \begin{bmatrix} x_i(t) \\ z_{i-1}(t) \\ z_i(t) \\ z_{i+1}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} v_{i-1}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} v_i(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_{i+1}(t), \qquad (6.2)$$

$$\Delta y(t) = \begin{bmatrix} C & \underline{0} \end{bmatrix} \begin{bmatrix} x_i(t) \\ z_{i-1}(t) \\ z_i(t) \\ z_{i+1}(t) \end{bmatrix}, \quad B_i = \begin{bmatrix} B_{i,i-1} & B_{i,i} & B_{i,i+1} \end{bmatrix}, \quad \Delta u(t) = \frac{v(t)}{W * \sqrt{2g(h_u - h_d)}}, \qquad (6.3)$$

which can be written in a more compact form as

$$\bar{x}_i(t+1) = \bar{A}\bar{x}_i(t) + \bar{B}_{i,i-1}v_{i-1}(t) + \bar{B}_{i,i}v_i(t) + \bar{B}_{i,i+1}v_{i+1}(t), \quad \Delta y(t) = \bar{C}\bar{x}(t). \qquad (6.4)$$

In the case of the first and third pool, since they limit the network and only have one neighbour, the augmented model structure changes slightly, since there are only two manipulated variables z considered. With the augmented state matrices it is possible to define the model predictor as

$$Y_i = \mathbf{\Pi}_i \begin{bmatrix} \hat{x}_i \\ z_{i-1}(t) \\ z_i(t) \\ z_{i+1}(t) \end{bmatrix} + \mathbf{W}_{i,i-1} \Delta U_{i-1} + \mathbf{W}_{i,i} \Delta U_i + \mathbf{W}_{i,i+1} \Delta U_{i+1}, \tag{6.5}$$

with

$$\mathbf{W}_{i,j} = \begin{bmatrix} \bar{C}_i \bar{B}_{i,j} & 0 & ... & 0 \\ \bar{C}_i \bar{A}_i \bar{B}_{i,j} & \bar{C}_i \bar{B}_{i,j} & ... & 0 \\ ... & ... & ... & ... \\ \bar{C}_i \bar{A}_i^{i-1} \bar{B}_{i,j} & \bar{C}_i \bar{A}_i^{i-2} \bar{B}_{i,j} & ... & \bar{C}_i \bar{B}_{i,j} \end{bmatrix}, \quad \mathbf{\Pi} = \begin{bmatrix} \bar{C}_i \bar{A}_i \\ \bar{C}_i \bar{A}_i^2 \\ ... \\ \bar{C}_i \bar{A}_i^i \end{bmatrix}. \tag{6.6}$$

The model "seen" by the LQE associated with subsystem i is given by

$$x_i(t+1) = A_i x_i(t) + B_{i,i-1} z_{i-1}(t) + B_{i,i} z_i(t) + B_{i,i+1} z_{i+1}(t), \quad \Delta y(t) = C x(t), \tag{6.7}$$

and the corresponding state estimate can be computed as

$$\hat{x}_i(t) = A_i \hat{x}_i(t) + B_{i,i-1} z_{i-1}(t-1) + B_{i,i} z_i(t-1) + B_{i,i+1} z_{i+1}(t-1) - M_i C_i \hat{x}_i(t) + M_i \Delta y(t). \tag{6.8}$$

The local cost functions associated with each subsystem i are defined as

$$J_i = (Y_i - R_i)^T (Y_i - R_i) + \rho_i \Delta U_i^T \Delta U_i, \tag{6.9}$$

in which Y can be written in a more compact notation as

$$Y_i = \mathbf{\Pi}_i \begin{bmatrix} \hat{x}_i \\ z_{i-1}(t) \\ z_i(t) \\ z_{i+1}(t) \end{bmatrix} + \mathbf{W}_i \bar{\Delta U}_i, \quad \mathbf{W}_i = \begin{bmatrix} \mathbf{W}_{i,i-1} & \mathbf{W}_{i,i} & \mathbf{W}_{i,i+1} \end{bmatrix}, \quad \bar{\Delta U}_i = \begin{bmatrix} \Delta U_{i-1} \\ \Delta U_i \\ \Delta U_{i+1} \end{bmatrix}. \tag{6.10}$$

In order to use the D-ADMM algorithm the following conditions are required:

- Each cost function $J_i : \mathbb{R}^n \to \mathbb{R}^n$ is convex function over $\mathbb{R}$ and each set of constraints is closed and convex;

- The problem is solvable;

- The network is connected and it does not vary with time.

Proof that these conditions [33] are valid in this system and problem are given in [6, Appendix D]. Since it is possible to use D-ADMM to solve the problem considered in this study, the next thing to do is to define the algorithm. In [33] the D-ADMM algorithm defined, associates dual variables to the nodes and in [34] it is shown an application of the algorithm to D-MPC.

The implementation of the D-ADMM considered in this dissertation is the one introduced in [6], where the dual variables are associated to the edges. This requires the definition of two dual variables

(γ_1 and γ_2) and of a cost weight ρ_A related with the cooperation part of the algorithm.

With this being said, the D-ADMM cost functions associated with each subsystem are

$$J_{1,A} = J_1 - \gamma_1 \bar{\Delta U}_1 + \frac{\rho_A}{2}(\bar{\Delta U}_1 - \bar{\Delta U}_2)^T(\bar{\Delta U}_1 - \bar{\Delta U}_2), \tag{6.11}$$

$$J_{2,A} = J_2 - (\gamma_1 - \gamma_2)^T \bar{\Delta U}_2 + \rho_A[(\bar{\Delta U}_2 - \bar{\Delta U}_1)^T(\bar{\Delta U}_2 - \bar{\Delta U}_1) + (\bar{\Delta U}_2 - \bar{\Delta U}_3)^T(\bar{\Delta U}_2 - \bar{\Delta U}_3)], \tag{6.12}$$

$$J_{3,A} = J_3 + \gamma_2 \bar{\Delta U}_3 + \frac{\rho_A}{2}(\bar{\Delta U}_3 - \bar{\Delta U}_2)^T(\bar{\Delta U}_3 - \bar{\Delta U}_2). \tag{6.13}$$

The next step is to replace each J_i by the local cost functions defined in (6.9) and to write expressions (6.11), (6.12) and (6.13) in order to the corresponding manipulated variables. The resulting expressions are given by

$$J_{1,A} = \bar{\Delta U}_1^T(\mathbf{W}_1^T\mathbf{W}_1 + \bar{\rho}_1 + \frac{\rho_A}{2}I)\bar{\Delta U}_1 + \bar{\Delta U}_1^T(2\mathbf{W}_1^T(\mathbf{\Pi}_1\bar{x}_1 - R_1) - \gamma_1 - \rho_A\bar{\Delta U}_2) + \Upsilon_1, \tag{6.14}$$

$$J_{2,A} = \bar{\Delta U}_2^T(\mathbf{W}_2^T\mathbf{W}_2 + \bar{\rho}_2 + 2\rho_A I)\bar{\Delta U}_2 + \bar{\Delta U}_2^T(2\mathbf{W}_2^T(\mathbf{\Pi}_2\bar{x}_2 - R_2) - (\gamma_1 - \gamma_2) - \rho_A(\bar{\Delta U}_1 + \bar{\Delta U}_3)) + \Upsilon_2, \tag{6.15}$$

$$J_{3,A} = \bar{\Delta U}_3^T(\mathbf{W}_3^T\mathbf{W}_3 + \bar{\rho}_3 + \frac{\rho_A}{2}I)\bar{\Delta U}_3 + \bar{\Delta U}_3^T(2\mathbf{W}_3^T(\mathbf{\Pi}_3\bar{x}_3 - R_3) + \gamma_2 - \rho_A\bar{\Delta U}_2) + \Upsilon_3, \tag{6.16}$$

where

$$\bar{\rho}_1 = \rho_1 \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \bar{\rho}_2 = \rho_2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \bar{\rho}_3 = \rho_3 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix}, \tag{6.17}$$

and Υ_i represents the terms that do not depend on the respective manipulated variable. While in chapter 5, the minimization of the RHC quadratic cost function was computed using *fminunc*, in this chapter the computation of the manipulated variables is accomplished with an analytical approach. This strategy was also selected in [6] as a workaround to the large computational load and times verified with MATLAB optimization functions.

The analytical minimization of cost functions $J_{i,A}$ is accomplished by computing the derivative in order to the respective manipulated variable and finding the value for which $\frac{\partial J_{i,A}}{\partial \bar{\Delta U}_i}$ is equal to 0. Before computing the derivatives, expressions (6.14), (6.15), and (6.16) can be written in a more compact way as

$$J_{i,A} = \bar{\Delta U}_i^T \Psi_i \bar{\Delta U}_i + \bar{\Delta U}_i^T \Phi_i + \Upsilon_i. \tag{6.18}$$

The derivative of the D-ADMM cost function $J_{i,A}$ (6.18) in order to $\bar{\Delta U}_i$ is given by

$$\frac{\partial J_{i,A}}{\partial \bar{\Delta U}_i} = 2\bar{\Delta U}_i \Psi_i + \Phi_i. \tag{6.19}$$

The values of the manipulated variables $\bar{\Delta U}_i^*$ that minimize the cost functions are given by expression

$$\frac{\partial J_{i,A}}{\partial \bar{\Delta U}_i} = 0 \quad \Leftrightarrow \quad \bar{\Delta U}_i^* = -\frac{1}{2}\Psi_i^{-1}\Phi_i. \tag{6.20}$$

Like in chapter 5, no constraints were considered in the optimization problem. In this case, the introduction of constraints would made impossible to use the analytical solution. A possible workaround, although not optimal, is introduced in [6] where the values of the manipulated variables and system output are bounded. The D-MPC strategy based on D-ADMM is introduced in algorithm 6.1.

Algorithm 6.1 D-MPC based on D-ADMM with edge-associated dual variables

---

Initialization of manipulated and dual variables: $\gamma_1 = 0; \gamma_2 = 0; \bar{\Delta U}_1 = 0; \bar{\Delta U}_2 = 0; \bar{\Delta U}_3 = 0$.
repeat

$\quad \Phi_1 = 2\mathbf{W}_1^T(\mathbf{\Pi}_1\bar{x}_1 - R_1) - \gamma_1 - \rho_A\bar{\Delta U}_2$
$\quad \Psi_1 = \mathbf{W}_1^T\mathbf{W}_1 + \bar{\rho}_1 + \frac{\rho_A}{2}I$
$\quad \bar{\Delta U}_1 = -\frac{1}{2}\Psi_1^{-1}\Phi_1$
$\quad \Phi_3 = 2\mathbf{W}_3^T(\mathbf{\Pi}_3\bar{x}_3 - R_3) + \gamma_2 - \rho_A\bar{\Delta U}_2$
$\quad \Psi_3 = \mathbf{W}_3^T\mathbf{W}_3 + \bar{\rho}_3 + \frac{\rho_A}{2}I$
$\quad \bar{\Delta U}_3 = -\frac{1}{2}\Psi_3^{-1}\Phi_3$
$\quad \Phi_2 = 2\mathbf{W}_2^T(\mathbf{\Pi}_2\bar{x}_2 - R_2) - (\gamma_1 - \gamma_2) - \rho_A(\bar{\Delta U}_1 + \bar{\Delta U}_3)$
$\quad \Psi_2 = \mathbf{W}_2^T\mathbf{W}_2 + \bar{\rho}_2 + 2\rho_A I$
$\quad \bar{\Delta U}_2 = -\frac{1}{2}\Psi_2^{-1}\Phi_2$
$\quad \gamma_1 = \gamma_1 - \rho_A(\bar{\Delta U}_1 - \bar{\Delta U}_2)$
$\quad \gamma_2 = \gamma_2 - \rho_A(\bar{\Delta U}_2 - \bar{\Delta U}_3)$

until pre-defined maximum number of iterations n_I reached or stopping criteria is met

---

6.1.1 Parameter tuning

After introducing algorithm 6.1, there are two parameters that need to be pre-defined before conducting experiments, a cost weight related with the D-ADMM algorithm, ρ_A, and the maximum number of iterations n_I. In order to select values for these parameters several experiments were conducted with different combinations of values and the results are shown in figure 6.1. It is possible to verify that initially, with less iterations and with a lower weight ρ_A, the output error is higher. The value tends to decrease with more iterations, and with a higher value for ρ_A. From figure 6.1, the combination of values defined to be used in the experiments is $\rho_A = 80$ and $n_I = 20$, since the associated output error is close to the minimum and a smaller maximum number of iterations is better in terms of computational time.

6.1.2 Simulation results

The D-MPC algorithm based on D-ADMM was used in several experiments with both linearized and SIMULINK non-linear canal models, whose results are shown in figures 6.2, 6.3, 6.4 and 6.5. In the case of the experiments conducted with the linearized MIMO model, the system outputs converge towards the reference signals with an integrator being included in series with the MPC controller. The effects of the interactions between subsystems are visible whenever occur variations in the water levels. As for the performance, the computational load and time is lower when compared to the solution used in the centralized MPC controller, with less communication steps required to compute the manipulated variables and the usage of an analytical solution. The results obtained with the SIMULINK non-linear canal model, in figures 6.4 and 6.4 show how the outputs converge to the
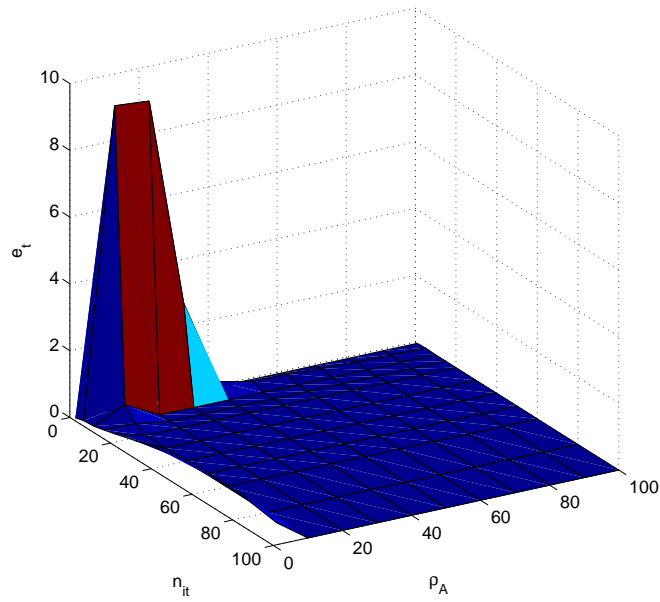
Figure 6.1: Variation of the total output error e_t with different combinations of cost weight ρ_A and the maximum number of iterations n_I. The value of the quadratic cost weights were $\rho_1 = 200$, $\rho_2 = 600$, $\rho_3 = 400$ and $N = 35$.

reference, with an existing error that appears to be related with the current water levels of each subsystem.
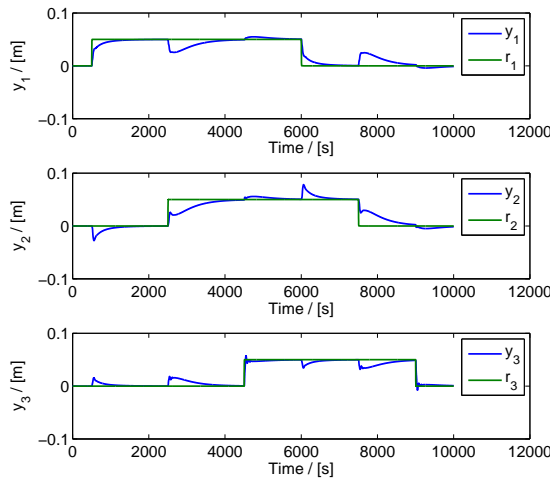
Figure 6.2: Closed-loop response of the linearized MIMO model with parameters $\rho_1 = 200$, $\rho_2 = 600$, $\rho_3 = 400$ and $N = 35$. The maximum number of iterations n_I is 20 and $\rho_A = 80$. (Output and Reference signals)

6.2 Adaptive D-MPC based on the Distributed Alternating Direction Method of Multipliers

Following the implementation of the D-MPC algorithm based on D-ADMM in the previous section, an identification step was added to the controller in order to define a new adaptive control strategy. Like in the previous adaptive control algorithms, the introduction of an identification step using the RLS
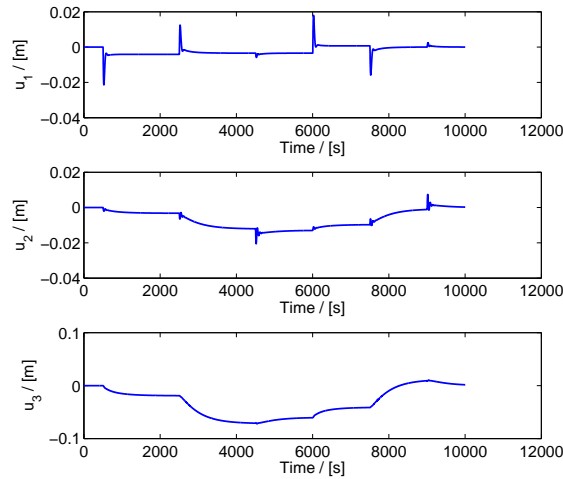
Figure 6.3: Closed-loop response of the linearized MIMO model with parameters $\rho_1 = 200$, $\rho_2 = 600$, $\rho_3 = 400$ and $N = 35$. The maximum number of iterations n_I is 20 and $\rho_A = 80$. (Input signal)
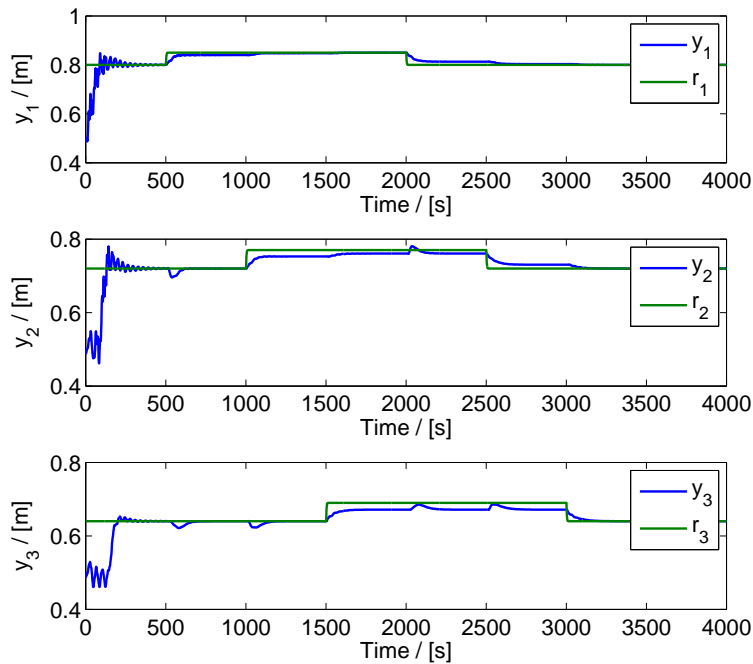
Figure 6.4: Closed-loop response of the system, in an experiment conducted in the SIMULINK canal model, with parameters $\rho_1 = 200$, $\rho_2 = 600$, $\rho_3 = 400$ and $N = 35$. The maximum number of iterations n_I is 20 and $\rho_A = 80$. (Output and Reference signals)

algorithm has the objective of improving the general performance of the controller, by being adaptable to changes in the dynamics of the system. During an initial period of time $t < t_I$, only the identification step is working with the RLS algorithm providing estimates of the parameters using input and output data obtained from the system open-loop response. During this period of time, the input of the system is excited by a signal defined by the user with the objective of having the parameter estimates closer to convergence by the time the controller is switched on. This strategy is implemented in order to
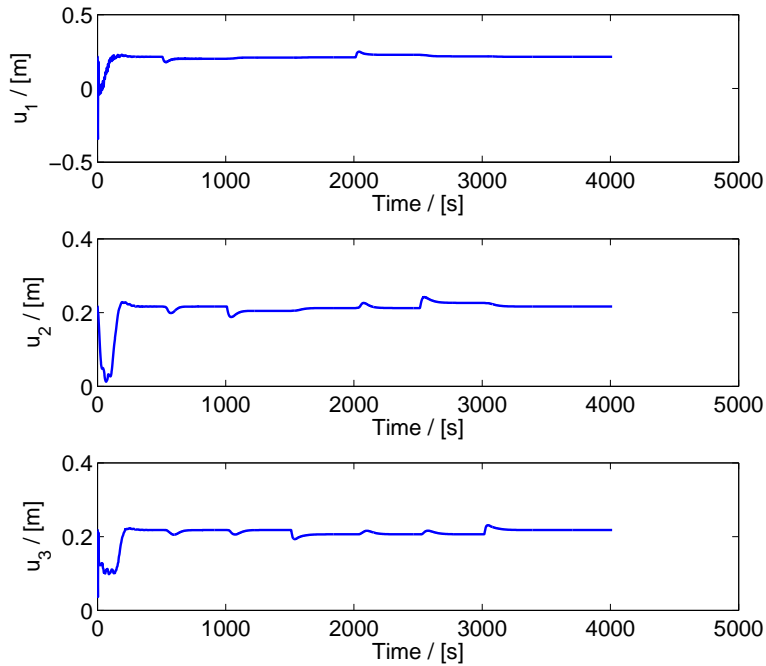
Figure 6.5: Closed-loop response of the system, in an experiment conducted in the SIMULINK canal model, with parameters $\rho_1 = 200$, $\rho_2 = 600$, $\rho_3 = 400$ and $N = 35$. The maximum number of iterations n_I is 20 and $\rho_A = 80$. (Input signal)

prevent stability issues and to allow the controller to act upon a model obtained with better parameter estimates.

After the period of time denoted by t_I the D-MPC algorithm introduced in section 6.1 is switched on and thus the control strategy becomes a sequence of two main steps: an identification step and a control step. At each time instant t, after estimating the values of the parameters, a system model is defined followed the computation of the manipulated variables using algorithm 6.2. As in the previous adaptation strategies a PRBS signal of small amplitude (around $0.001m$) is added to the control variable to provide enough excitation to the identification procedure. The parameters considered for the controller are the ones defined in section 6.1.1.

6.2.1   Simulation results

Several experiments were conducted using this adaptive control algorithm in the SIMULINK model of the water canal. In this D-MPC approach, in order to guarantee that the parameters estimates are closer to convergence to prevent stability issues, the value defined for the time instant in which the controller is switched on is $t_I = 2 \times 10^5 s$. The values of the quadratic cost weights ρ_i were the ones considered in the adaptive centralized controller. In figures 6.6 and 6.7 it is represented the system closed-loop response with the adaptive distributed controller. The higher value for the quadratic cost weights was again the considered solution to handle with existing high-frequency oscillations, and

---

Algorithm 6.2 Adaptive D-MPC based on D-ADMM with edge-associated dual variables

---

Initialization of manipulated and dual variables: $\gamma_1 = 0; \gamma_2 = 0; \bar{\Delta U}_1 = 0; \bar{\Delta U}_2 = 0; \bar{\Delta U}_3 = 0$.
Initialization of parameter estimates $(\theta_1, \theta_2, \theta_3)$ and respective covariance matrices (P_1, P_2, P_3).

for each time instant t

    Computation of parameters $\theta_i(t)$, using algorithm 2.1 and input and output data $(\Delta u(t), \Delta y(t))$.
    Define the augmented models of each subsystem i, using the parameter estimates $\Theta_i(t)$

      $\bar{x}_i(t) = \bar{A}\bar{x}_i(t-1) + \bar{B}_{i,i-1}\Delta u_{i-1}(t-1) + \bar{B}_{i,i}\Delta u_i(t-1) + \bar{B}_{i,i+1}\Delta u_{i+1}(t-1)$
      $\Delta y(t) = \bar{C}\bar{x}(t)$

if $t > t_I$ **then**

    State estimation:

      $\hat{x}_i(t) = A_i\hat{x}_i(t) + B_{i,i-1}z_{i-1}(t-1) + B_{i,i}z_i(t-1) + B_{i,i+1}z_{i+1}(t-1) - M_iC_i\hat{x}_i(t) + M_i\Delta y(t)$

    Predictor model:

      Computation of $\mathbf{\Pi}_i$ and $\mathbf{W}_i$
      $Y_i = \mathbf{\Pi}_i\bar{x} + \mathbf{W}_i\bar{\Delta U}_i$

    Minimization of D-ADMM cost functions:

      repeat
      $\Psi_1 = \mathbf{W}_1^T\mathbf{W}_1 + \bar{\rho}_1 + \frac{\rho_A}{2}I$
      $\Phi_1 = 2\mathbf{W}_1^T(\mathbf{\Pi}_1\bar{x}_1 - R_1) - \gamma_1 - \rho_A\bar{\Delta U}_2$
      $\bar{\Delta U}_1 = -\frac{1}{2}\Psi_1^{-1}\Phi_1$
      $\Psi_3 = \mathbf{W}_3^T\mathbf{W}_3 + \bar{\rho}_3 + \frac{\rho_A}{2}I$
      $\Phi_3 = 2\mathbf{W}_3^T(\mathbf{\Pi}_3\bar{x}_3 - R_3) + \gamma_2 - \rho_A\bar{\Delta U}_2$
      $\bar{\Delta U}_3 = -\frac{1}{2}\Psi_3^{-1}\Phi_3$
      $\Psi_2 = \mathbf{W}_2^T\mathbf{W}_2 + \bar{\rho}_2 + 2\rho_A I$
      $\Phi_2 = 2\mathbf{W}_2^T(\mathbf{\Pi}_2\bar{x}_2 - R_2) - (\gamma_1 - \gamma_2) - \rho_A(\bar{\Delta U}_1 + \bar{\Delta U}_3)$
      $\bar{\Delta U}_2 = -\frac{1}{2}\Psi_2^{-1}\Phi_2$
      $\gamma_1 = \gamma_1 - \rho_A(\bar{\Delta U}_1 - \bar{\Delta U}_2)$
      $\gamma_2 = \gamma_2 - \rho_A(\bar{\Delta U}_2 - \bar{\Delta U}_3)$
      until pre-defined maximum number of iterations n_I reached or stopping criteria is met

---

despite the existing output error, mostly visible in the second pool response, the water levels converged to the reference signals. The implementation of an optimal analytical solution reduced the computational load and time with the downside of not allowing constraints to be considered. By the time instant the controller step is switched on, the parameter estimates are close to convergence, as shown in figure 6.8 and it is also possible to see how these estimates vary whenever occur variations in the system response.
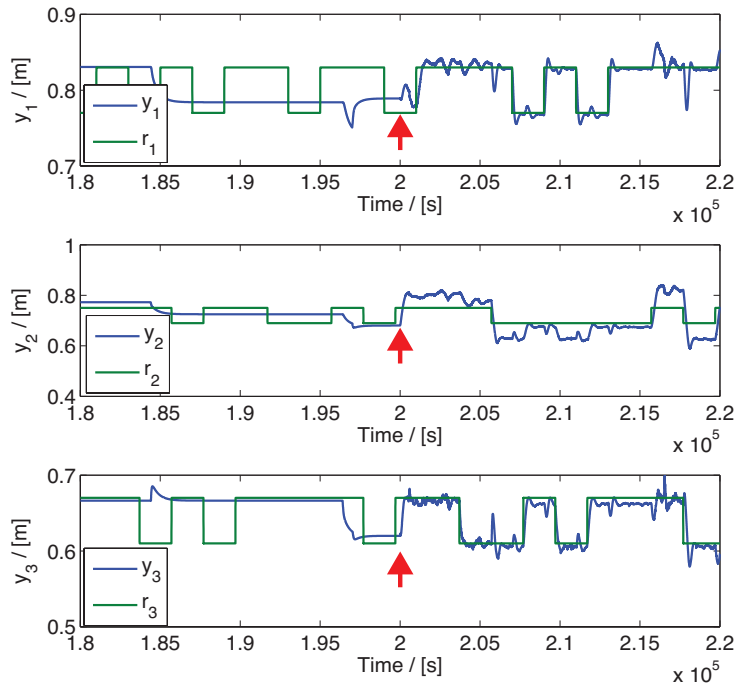
Figure 6.6: Open-loop and closed-loop response of the system with the adaptive D-MPC based on D-ADMM, in an experiment conducted in the SIMULINK canal model, with parameters $\rho_1 = 2000$, $\rho_2 = 1000$, $\rho_3 = 2000$ and $N = 35$. The maximum number of iterations n_I is 20 and $\rho_A = 80$. The controller step is switched on in time instant $t = 2 \times 10^5 s$. (Output and Reference signals)
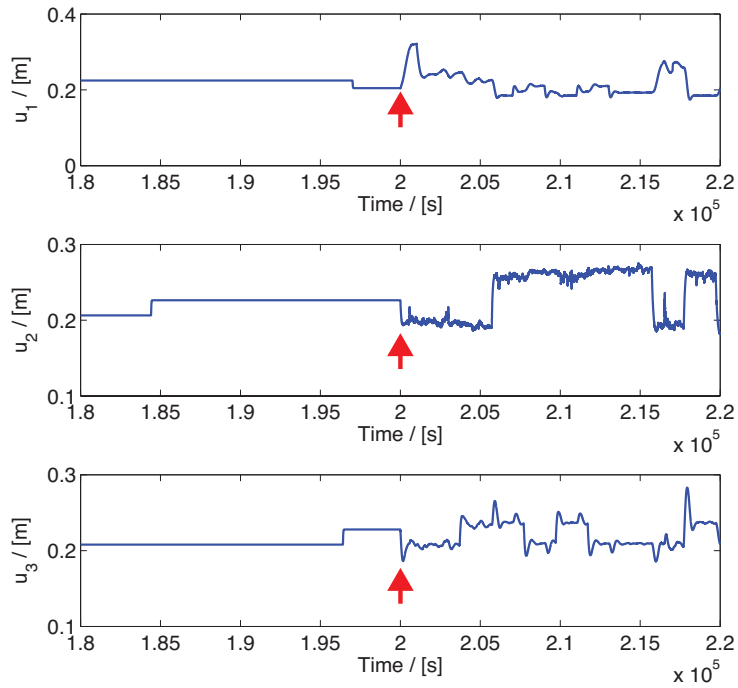
Figure 6.7: Open-loop and closed-loop response of the system with the adaptive D-MPC based on D-ADMM, in an experiment conducted in the SIMULINK canal model, with parameters $\rho_1 = 2000, \rho_2 = 1000, \rho_3 = 2000$ and $N = 35$. The maximum number of iterations n_I is 20 and $\rho_A = 80$. The controller step is switched on in time instant $t = 2 \times 10^5 s$. (Input signal)

In order to verify how the controller rejected disturbances, a different experiment was conducted with the opening of the lateral off-take valves as represented in figure 6.10. Figures 6.9 and 6.10 represent the closed-loop response of the system, in which it is possible to see that in the presence of disturbances, the computed manipulated variables compensate the flow drawn by the lateral off-takes, in order to ensure the tracking of reference signals. This experiment shows that the adaptive distributed controller is also able to reject disturbances.
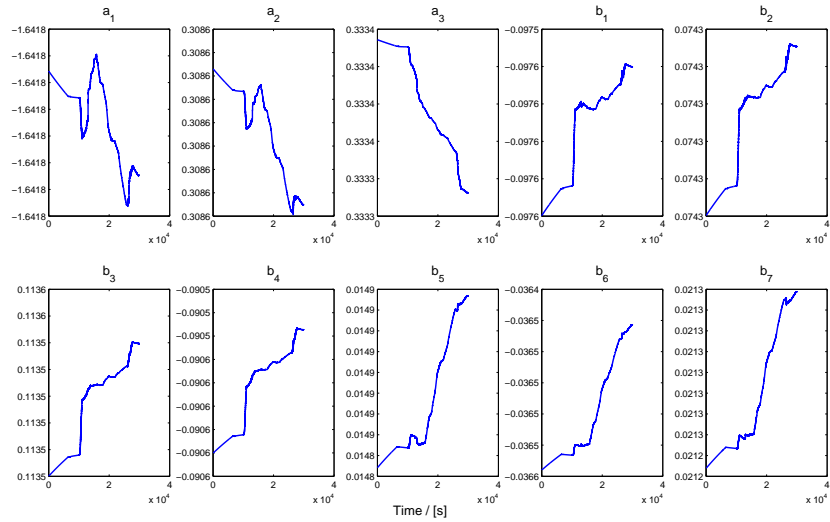
(a) Parameter estimates of the first subsystem.
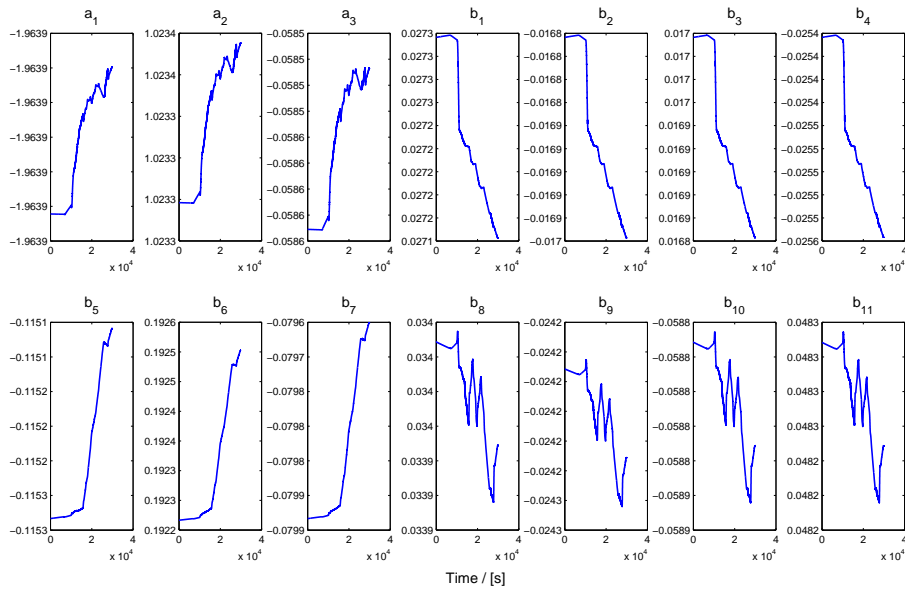
(b) Parameter estimates of the second subsystem.
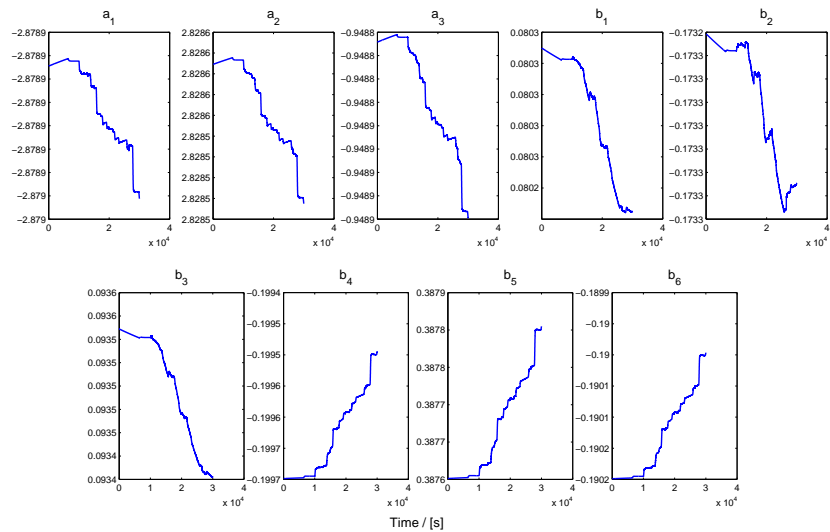
(c) Parameter estimates of the third subsystem.

Figure 6.8: Representation of the MIMO parameters estimates using the Adaptive D-MPC controller based on D-ADMM between $1.8 \times 10^5 s$ and $2.2 \times 10^5 s$. The control step is switched on at $2 \times 10^5 s$.
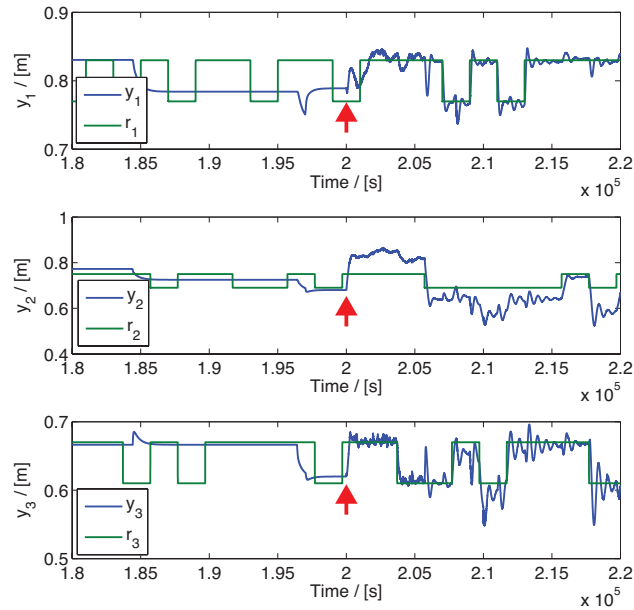
Figure 6.9: Open-loop and closed-loop response of the system, in the presence of disturbances, with the adaptive D-MPC based on D-ADMM, in an experiment conducted in the SIMULINK canal model, with parameters $\rho_1 = 2000$, $\rho_2 = 1000$, $\rho_3 = 2000$ and $N = 35$. The maximum number of iterations n_I is 20 and $\rho_A = 80$. The controller step is switched on in time instant $t = 2 \times 10^5 s$. (Output and Reference signals)
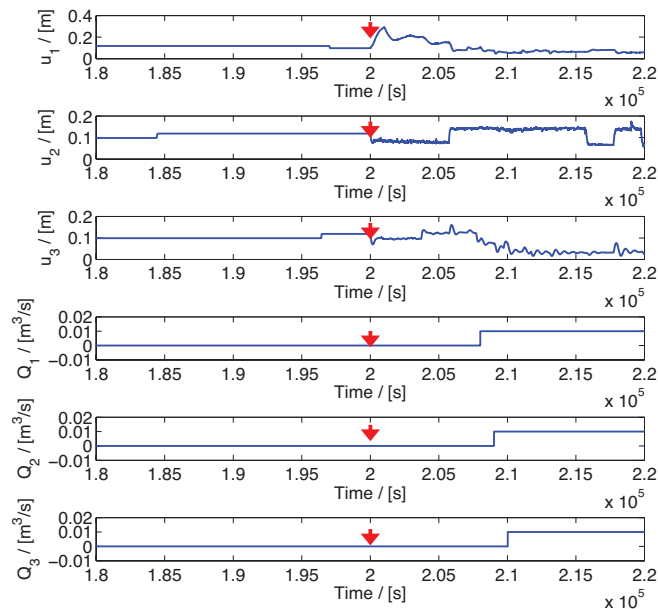
Figure 6.10: Representation of the manipulated variables u_i and flow drawn by the lateral off-takes Q_i, with the adaptive D-MPC based on D-ADMM, in an experiment conducted in the SIMULINK canal model, with parameters $\rho_1 = 2000$, $\rho_2 = 1000$, $\rho_3 = 2000$ and $N = 35$. The maximum number of iterations n_I is 20 and $\rho_A = 80$. The controller step is switched on in time instant $t = 2 \times 10^5 s$.

6.3 D-MPC with neighbouring agent coordination

In this approach, the D-MPC algorithm to be introduced is based on Game Theory concepts, in the sense that each controller must optimize its control variable, taking into account the knowledge of the manipulated variables computed by its neighbours. Since each controller has access to the control inputs of neighbouring subsystems, which are seen as accessible disturbances, the objective of this control strategy is to guarantee an optimal approximation to the minimum of the global cost function. If each controller computes its manipulated variable with knowledge of the control inputs of the neighbouring subsystems, the goal is to reach a situation in which no controller benefits from changing the manipulated variable, the Nash equilibrium [22]. This coordination process, similar to control strategies introduced in [17] and [25], is defined in this study as a simpler alternative, in terms of computational load, to the D-MPC algorithm based on D-ADMM.

The model of each subsystem i, with integral effect, is now written as

$$\begin{bmatrix} x_i(t+1) \\ d_{i-1}(t+1) \\ z_i(t+1) \\ d_{i+1}(t+1) \end{bmatrix} = \begin{bmatrix} A_i & \begin{bmatrix} \Gamma_{i,i-1} & B_i & \Gamma_{i,i+1} \end{bmatrix} \\ \underline{0} & I \end{bmatrix} \begin{bmatrix} x_i(t) \\ d_{i-1}(t) \\ z_i(t) \\ d_{i+1}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} v_{i-1}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} v_i(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_{i+1}(t), \quad (6.21)$$

$$\Delta y(t) = \begin{bmatrix} C & \underline{0} \end{bmatrix} \begin{bmatrix} x_i(t) \\ d_{i-1}(t) \\ z_i(t) \\ d_{i+1}(t) \end{bmatrix}, \quad \Delta u(t) = \frac{v(t)}{W * \sqrt{2g(h_u - h_d)}}, \quad (6.22)$$

in which d_i is the manipulated variable of the i-th subsystem, seen as an accessible disturbance, given by

$$d_i(t) = d_i(t-1) + v_i(t). \quad (6.23)$$

Expression (6.21) can be written in a more compact notation as

$$\bar{x}_i(t+1) = \bar{A}\bar{x}_i(t) + \bar{\Gamma}_{i,i-1}v_{i-1}(t) + \bar{B}_i v_i(t) + \bar{\Gamma}_{i,i+1}v_{i+1}(t), \quad \Delta y(t) = \bar{C}\bar{x}(t). \quad (6.24)$$

This system may also contemplate the disturbances from the lateral off-take valves, using the model considered in section 2.4.

6.3.1 Predictor model

The local cost function associated with the i-th subsystem is given by (6.9) and the predictor model by (6.10), where the matrices $\mathbf{W}$ and $\mathbf{\Pi}$ are defined as

$$\mathbf{W}_{i,j \neq i} = \begin{bmatrix} \bar{C}_i \bar{\Gamma}_{i,j} & 0 & ... & 0 \\ \bar{C}_i \bar{A}_i \bar{\Gamma}_{i,j} & \bar{C}_i \bar{\Gamma}_{i,j} & ... & 0 \\ ... & ... & ... & ... \\ \bar{C}_i \bar{A}_i^{i-1} \bar{\Gamma}_{i,j} & \bar{C}_i \bar{A}_i^{i-2} \bar{\Gamma}_{i,j} & ... & \bar{C}_i \bar{\Gamma}_{i,j} \end{bmatrix}, \quad \mathbf{W}_{i,i} = \begin{bmatrix} \bar{C}_i \bar{B}_{i,i} & 0 & ... & 0 \\ \bar{C}_i \bar{A}_i \bar{B}_{i,i} & \bar{C}_i \bar{B}_{i,i} & ... & 0 \\ ... & ... & ... & ... \\ \bar{C}_i \bar{A}_i^{i-1} \bar{B}_{i,i} & \bar{C}_i \bar{A}_i^{i-2} \bar{B}_{i,i} & ... & \bar{C}_i \bar{B}_{i,i} \end{bmatrix},$$
$$(6.25)$$

$$\mathbf{\Pi} = \begin{bmatrix} \bar{C}_i \bar{A}_i \\ \bar{C}_i \bar{A}_i^2 \\ ... \\ \bar{C}_i \bar{A}_i^i \end{bmatrix}. \quad (6.26)$$

6.3.2 Minimization of cost functions

The objective of this control strategy is to minimize a global cost function that is the sum of all local cost functions of each subsystem i,

$$J(\varDelta U_1, \varDelta U_2, \varDelta U_3) = J_1(\varDelta U_1, \varDelta U_2) + J_2(\varDelta U_1, \varDelta U_2, \varDelta U_3) + J_3(\varDelta U_2, \varDelta U_3). \tag{6.27}$$

This optimization problem is accomplished with a cooperation strategy in which each controller associated with a subsystem computes the control variable that minimizes its cost function with knowledge of its neighbours decisions. With this being said, the local cost functions are given by

$$J_i = (Y_i - R_i)^T (Y_i - R_i) + \rho_i \varDelta U_i^T \varDelta U_i, \tag{6.28}$$

$$Y_i = \mathbf{\Pi}_i \bar{x}_i + \mathbf{W}_{i,i-1} \varDelta U_{i-1} + \mathbf{W}_{i,i} \varDelta U_i + \mathbf{W}_{i,i+1} \varDelta U_{i+1} \tag{6.29}$$

The minimum of each local cost function is solved by computing its derivative in order to the corresponding manipulated variable and finding the solution of $\frac{\partial J_i}{\partial \varDelta U_i} = 0$. This computations result in

$$J_i = \varDelta U_i^T (2\mathbf{W}_{i,i}^T (\mathbf{\Pi} \bar{x}_i + \mathbf{W}_{i,i-1} \varDelta U_{i-1} + \mathbf{W}_{i,i+1} \varDelta U_{i+1})) + \varDelta U_i^T (\mathbf{W}_{i,i}^T \mathbf{W}_{i,i} + \rho_i I) + \Upsilon_i, \tag{6.30}$$

where Υ_i denotes the terms that do not depend on the vector of manipulated variables $\varDelta U_i$. The derivative of the cost function is given by

$$\frac{\partial J_i}{\partial \bar{\varDelta} U_i} = 2\varDelta U_i (\mathbf{W}_{i,i}^T \mathbf{W}_{i,i} + \rho_i I) + 2\mathbf{W}_{i,i}^T (\mathbf{\Pi} \bar{x}_i + \mathbf{W}_{i,i-1} \varDelta U_{i-1} + \mathbf{W}_{i,i+1} \varDelta U_{i+1}). \tag{6.31}$$

The solution of $\frac{\partial J_i}{\partial \varDelta U_i} = 0$ is therefore given by

$$\bar{\varDelta} U_i^* = -\frac{1}{2} (\mathbf{W}_{i,i}^T \mathbf{W}_{i,i} + \rho_i I)^{-1} (2\mathbf{W}_{i,i}^T (\mathbf{\Pi} \bar{x}_i + \mathbf{W}_{i,i-1} \varDelta U_{i-1} + \mathbf{W}_{i,i+1} \varDelta U_{i+1})) \tag{6.32}$$

. This results in the iterative procedure introduced in algorithm 6.3.

Algorithm 6.3 D-MPC with neighbouring agent coordination

Initialization of manipulated variables: $\bar{\varDelta} U_1 = 0; \bar{\varDelta} U2 = 0; \bar{\varDelta} U_3 = 0.$
$\bar{\varDelta} U = \begin{bmatrix} \bar{\varDelta} U_1 & \bar{\varDelta} U_2 & \bar{\varDelta} U_3 \end{bmatrix}^T$
$R_i = \rho_i I$
$M_i = \mathbf{W}_{i,i}^T \mathbf{W}_{i,i} + R_i$
$M = \begin{bmatrix} M_1 & \underline{0} & \underline{0} \\ \underline{0} & M_2 & \underline{0} \\ \underline{0} & \underline{0} & M_3 \end{bmatrix}$
$\Phi = \begin{bmatrix} \underline{0} & 2\mathbf{W}_{1,1}^T \mathbf{W}_{1,2} & \underline{0} \\ 2\mathbf{W}_{2,2}^T \mathbf{W}_{2,1} & \underline{0} & 2\mathbf{W}_{2,2}^T \mathbf{W}_{2,3} \\ \underline{0} & 2\mathbf{W}_{3,3}^T \mathbf{W}_{3,2} & \underline{0} \end{bmatrix}$
$\psi_1 = 2\mathbf{W}_{1,1}^T \mathbf{\Pi}_1 \bar{x}_1$
$\psi_2 = 2\mathbf{W}_{2,2}^T \mathbf{\Pi}_2 \bar{x}_2$
$\psi_3 = 2\mathbf{W}_{3,3}^T \mathbf{\Pi}_3 \bar{x}_3$
$\Psi = \begin{bmatrix} \psi_1 & \psi_2 & \psi_3 \end{bmatrix}^T$
repeat
    $\bar{\varDelta} U = -\frac{1}{2} M^{-1} \Psi - \frac{1}{2} M^{-1} \Phi \bar{\varDelta} U$
until pre-defined maximum number of iterations n_I reached or stopping criteria is met

---

6.3.3 Parameter tuning

The iterative procedure described in algorithm 6.3,

$$\bar{\Delta U} = -\frac{1}{2}M^{-1}\Psi - \frac{1}{2}M^{-1}\Phi\bar{\Delta U},$$ (6.33)

will converge if the spectral radius

$$\lambda_{max} := \max \lambda(M^{-1}\Phi)$$ (6.34)

verifies $|\lambda_{max}| < 1$ [17]. In order to study how the quadratic cost weights ρ_i influence the spectral radius λ_{max}, several experiments were conducted with different combinations of weights. Although in [17] the proposed configuration of the controller assumed that the weights were equal, in this study the assumption is not considered. Therefore, by fixing $\rho_1 = 200$, the value considered in the previous sections, the spectral radius representation with respect to combinations of weights ρ_2 and ρ_3 is shown in figure 6.11.
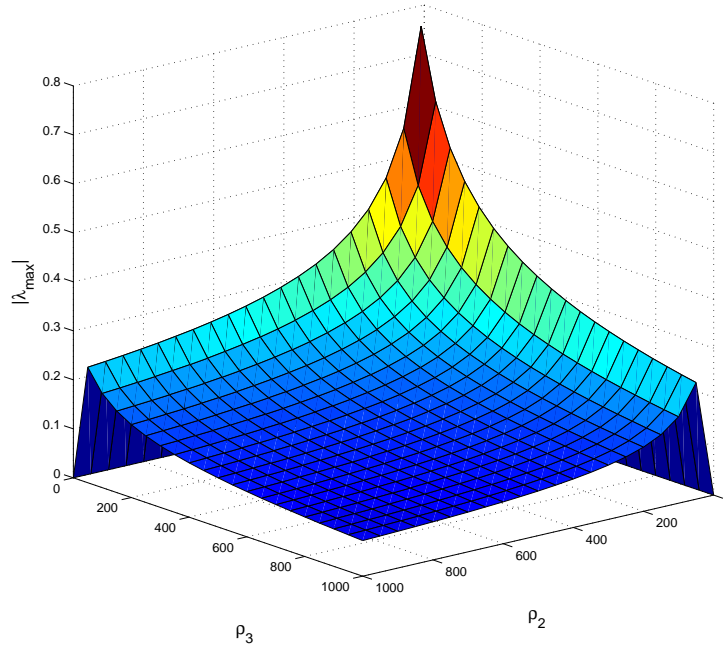
Figure 6.11: Spectral radius λ_{max}.

As it is possible to verify, the spectral radius λ_{max} tends to decrease when the cost weights ρ_i increase, which influences the rate of convergence of the iterative procedure, such that a lower value of λ_{max} results in a fast convergence of the algorithm. The selected values of the quadratic cost weights are $\rho_1 = 200$, $\rho_2 = 800$ and $\rho_3 = 100$. Another parameter that needs to be defined is the maximum number of iterations n_I, and thus several experiments were conducted with different values of n_I in order to determine the associated total output error e_t. The results are shown in figure 6.12 and it is possible to see that the output error tends to decrease with an increasing number of iterations. Since the total output error is smaller for $n_I > 10$, the value considered for the number of maximum iterations is the same as in section 6.1, $n_I = 20$.
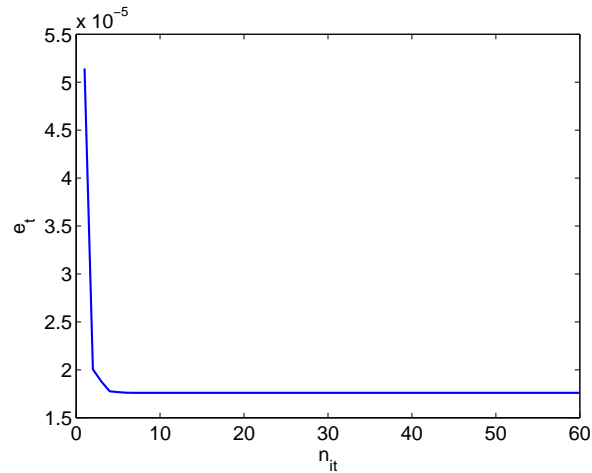
Figure 6.12: Variation of the total output error e_t with the maximum number of iterations n_I.

6.3.4 Simulation results

After defining the parameters required by the controller, two experiments were conducted in both linearized and non-linear SIMULINK canal models, with the purpose of verifying the system response. The experiments are similar to the ones described in section 6.1 and the results are represented in figures 6.13, 6.14, 6.15 and 6.16. As it is possible to verify, the output converges to the reference signal, although in the experiment conducted in the linearized model, the system response appears to be faster and more oscillating than with algorithm 6.1. The results of the experiment conducted in the SIMULINK non-linear canal model in figures 6.15 and 6.16 are similar to the results obtained with the multi-variable MPC algorithms introduced so far, with the output tracking the reference with the same issue regarding the output error, even with integral action, and a similar system response. In terms of performance, the D-MPC algorithm based on Game Theory concepts, at least with an equal number of maximum iterations, appears to have a similar computational time and load when compared with the D-ADMM approach. In comparison with the centralized MPC algorithms, there are practically no differences regarding the system response obtained with both D-MPC strategies.

6.4 Adaptive D-MPC with neighbouring agent coordination

Following the design of a D-MPC algorithm based on Game Theory concepts, comes the definition of a corresponding adaptive control strategy, similar to the one defined in section 6.2. The controller is composed by an identification step and a control step, that is only activated after a predefined time instant t_I. Before t_I only the identification step is activated, using the input and output data from the system to estimate its parameters, using RLS algorithm 6.2. The time instant selected for the control step activation is the same as the one considered in section 6.2, since it was verified that at $t_I = 2 \times 10^5 s$ the parameter estimates are closer to convergence. Regarding section 6.2, the only relevant change is the control algorithm considered, by using the iterative procedure introduced in the previous section. The addition of a PRBS signal with $0.001m$ of amplitude in the system input was again considered, in order to provide enough excitation to the RLS algorithm, by avoiding issues
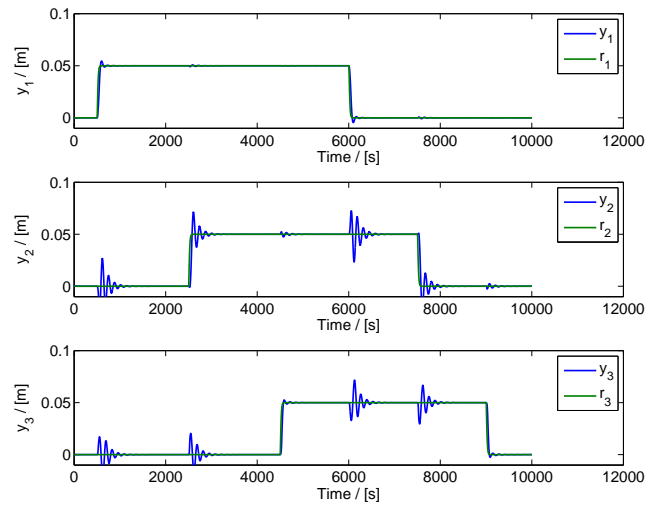
Figure 6.13: Closed-loop response of the linearized MIMO model with parameters $\rho_1 = 200$, $\rho_2 = 800$, $\rho_3 = 100$ and $N = 35$. The maximum number of iterations n_I is 20. (Output and Reference signals)
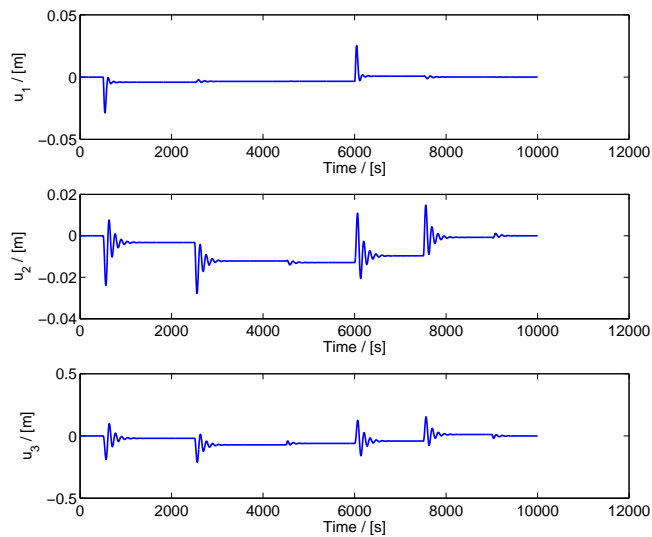
Figure 6.14: Closed-loop response of the linearized MIMO model with parameters $\rho_1 = 200$, $\rho_2 = 800$, $\rho_3 = 100$ and $N = 35$. The maximum number of iterations n_I is 20. (Input signals)

such as the covariance blow-up. Regarding the parameters of the controller, the number of iterations and the finite-time horizon remain the same, whereas the quadratic weight costs considered were the ones used in section 6.2 ($\rho_1 = 2000, \rho_2 = 1000, \rho_3 = 2000$) since this combination worked in the adaptive multi-variable MPC algorithms.

6.4.1 Simulation results

Two experiments were conducted in the SIMULINK non-linear canal model, considering two different situations with respect to existing external disturbances caused by the opening of the lateral
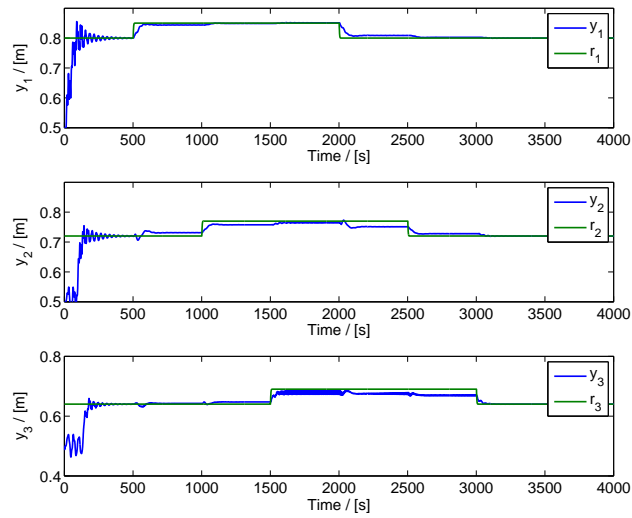
Figure 6.15: Closed-loop response of the SIMULINK non-linear canal model with parameters $\rho_1 = 200$, $\rho_2 = 800$, $\rho_3 = 100$ and $N = 35$. The maximum number of iterations n_I is 20. (Output and Reference signals)

Figure 6.16: Closed-loop response of the SIMULINK non-linear canal model with parameters $\rho_1 = 200$, $\rho_2 = 800$, $\rho_3 = 100$ and $N = 35$. The maximum number of iterations n_I is 20. (Input signals)

off-take valves. The results of the first experiment are represented in figures 6.17 and 6.17. Despite the output error and the oscillatory behavior, the results are practically similar to the ones obtained with the adaptive D-MPC algorithm based on D-ADMM, with a higher output error in the second pool, that appears to be related with the interactions between subsystems. As for the first and third pools, the outputs converge to the reference and appear to be less sensitive to variations in neighbouring subsystems. Regarding the parameter estimates, by the time the controller step is switched on, these are close to convergence, and in figure 6.19 is possible to see how the estimates change with the input excitation.

Another experiment was conducted, with existing disturbances caused by the opening of the lateral

---

Algorithm 6.4 Adaptive D-MPC with neighbouring agent coordination

---

Initialization of manipulated variables: $\bar{\Delta U}_1 = 0$; $\bar{\Delta U}2 = 0$; $\bar{\Delta U}_3 = 0$.
$\bar{\Delta U} = \begin{bmatrix} \bar{\Delta U}_1 & \bar{\Delta U}_2 & \bar{\Delta U}_3 \end{bmatrix}^T$
Initialization of parameter estimates $(\theta_1, \theta_2, \theta_3)$ and respective covariance matrices (P_1, P_2, P_3).

for each time instant t
    Computation of parameters $\theta_i(t)$, using algorithm 2.1 and input and output data $(\Delta u(t), \Delta y(t))$.
    Define the augmented models of each subsystem i, using the parameter estimates $\Theta_i(t)$

$$\bar{x}_i(t+1) = \bar{A}\bar{x}_i(t) + \begin{bmatrix} \bar{\Gamma}_{i,i-1} & \bar{B}_i & \bar{\Gamma}_{i,i+1} \end{bmatrix} V_i(t)$$
$$\Delta y(t) = \bar{C}\bar{x}(t)$$

    if $t > t_I$ **then**
$$R_i = \rho_i I$$
$$M_i = \mathbf{W}_{i,i}^T \mathbf{W}_{i,i} + R_i$$
$$M = \begin{bmatrix} M_1 & \underline{0} & \underline{0} \\ \underline{0} & M_2 & \underline{0} \\ \underline{0} & \underline{0} & M_3 \end{bmatrix}$$
$$\Phi = \begin{bmatrix} \underline{0} & 2\mathbf{W}_{1,1}^T \mathbf{W}_{1,2} & \underline{0} \\ 2\mathbf{W}_{2,2}^T \mathbf{W}_{2,1} & \underline{0} & 2\mathbf{W}_{2,2}^T \mathbf{W}_{2,3} \\ \underline{0} & 2\mathbf{W}_{3,3}^T \mathbf{W}_{3,2} & \underline{0} \end{bmatrix}$$
$$\psi_1 = 2\mathbf{W}_{1,1}^T \mathbf{\Pi}_1 \bar{x}_1$$
$$\psi_2 = 2\mathbf{W}_{2,2}^T \mathbf{\Pi}_2 \bar{x}_2$$
$$\psi_3 = 2\mathbf{W}_{3,3}^T \mathbf{\Pi}_3 \bar{x}_3$$
$$\Psi = \begin{bmatrix} \psi_1 & \psi_2 & \psi_3 \end{bmatrix}^T$$
    repeat
$$\bar{\Delta U} = -\tfrac{1}{2} M^{-1} \Psi - \tfrac{1}{2} M^{-1} \Phi \bar{\Delta U}$$
    until pre-defined maximum number of iterations n_I reached or stopping criteria is met

---

off-take valves, identified in figure 6.21 by Q_i. The results of this experiment are represented in figures 6.20 and 6.21, in which it is possible to see how the controller reacts to the presence of disturbances, compensating the consequent water level decrease in each pool in the computation of the manipulated variables, in order to maintain the reference tracking.
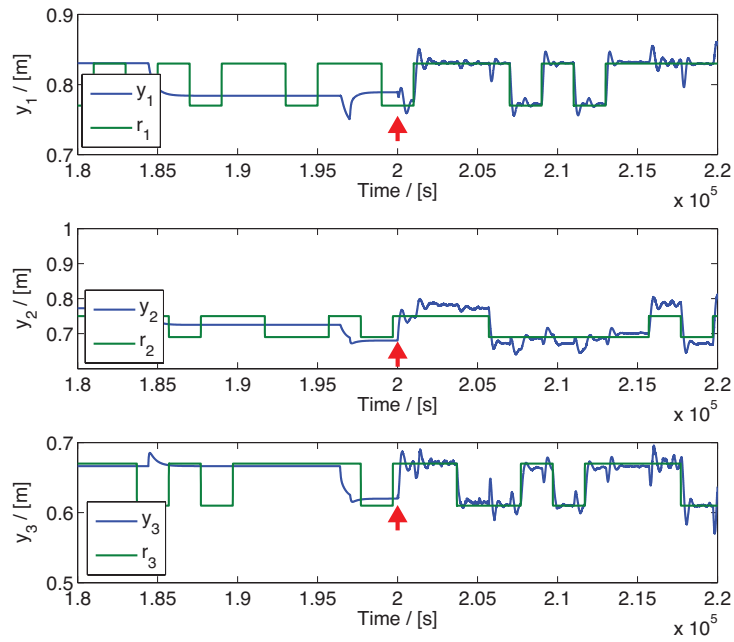
Figure 6.17: Open-loop and closed-loop response of the system with the adaptive D-MPC with neighbouring agent coordination, in an experiment conducted in the SIMULINK canal model, with parameters $\rho_1 = 2000, \rho_2 = 1000, \rho_3 = 2000$ and $N = 35$. The maximum number of iterations n_I is 20. The controller step is switched on in time instant $t = 2 \times 10^5 s$. (Output and Reference signals)
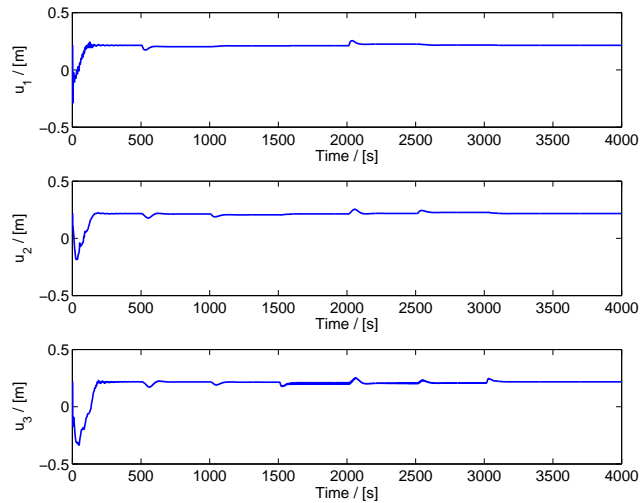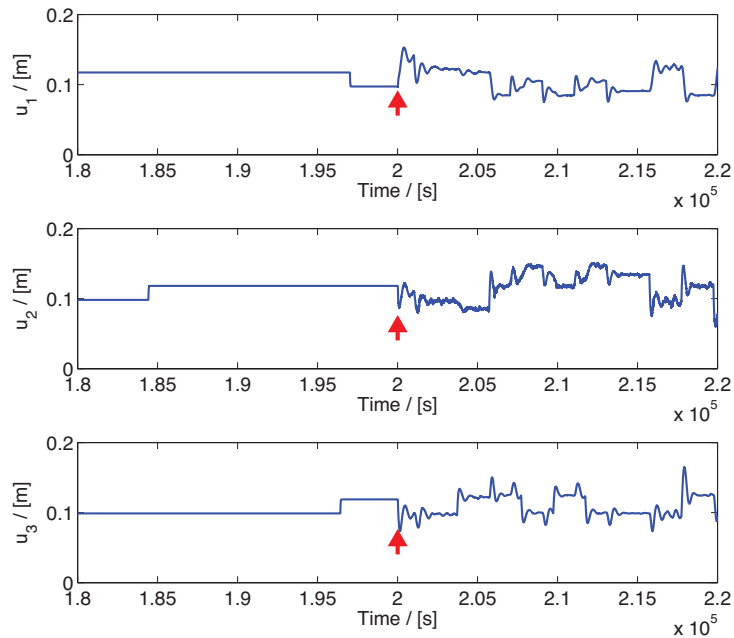

Figure 6.18: Open-loop and closed-loop response of the system with the adaptive D-MPC with neighbouring agent coordination, in an experiment conducted in the SIMULINK canal model, with parameters $\rho_1 = 2000, \rho_2 = 1000, \rho_3 = 2000$ and $N = 35$. The maximum number of iterations n_I is 20. The controller step is switched on in time instant $t = 2 \times 10^5 s$. (Input signals)
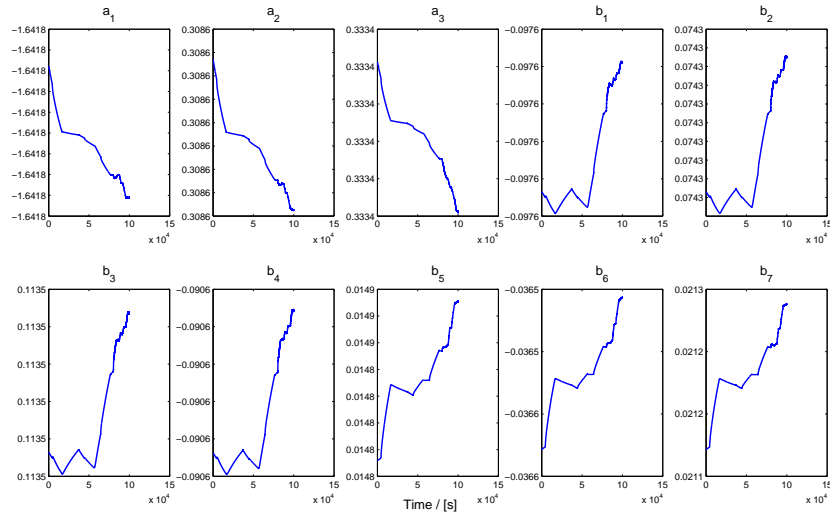
(a) Parameter estimates of the first subsystem.
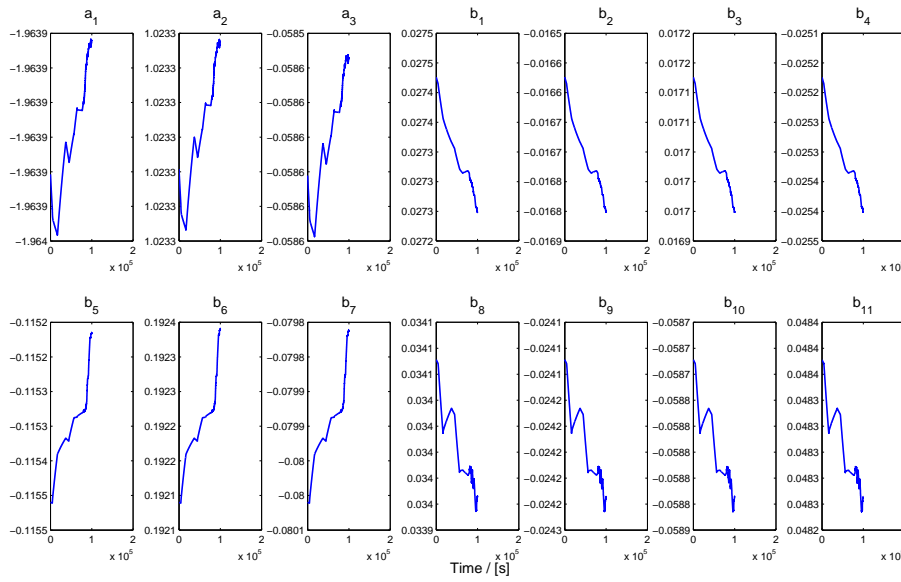
(b) Parameter estimates of the second subsystem.
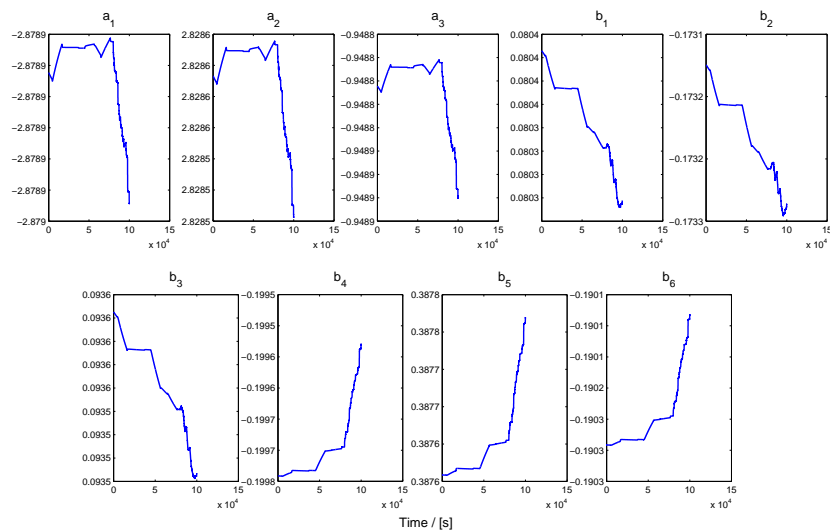
(c) Parameter estimates of the third subsystem.

Figure 6.19: Representation of the MIMO parameters estimates using the Adaptive D-MPC controller with neighbouring coordination between $1.2 \times 10^5 s$ and $2.2 \times 10^5 s$. The control step is switched on at $2 \times 10^5 s$.
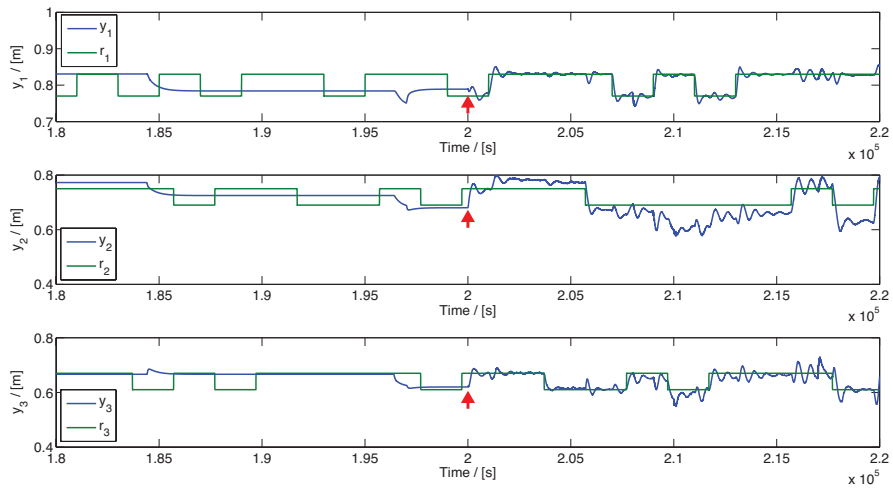
Figure 6.20: Open-loop and closed-loop system response, obtained with the adaptive D-MPC with neighbouring coordination, in an experiment conducted in the SIMULINK canal model, with parameters $\rho_1 = 2000, \rho_2 = 1000, \rho_3 = 2000$ and $N = 35$. The maximum number of iterations n_I is 20. The controller step is switched on in time instant $t = 2 \times 10^5 s$. (Output and Reference signals)
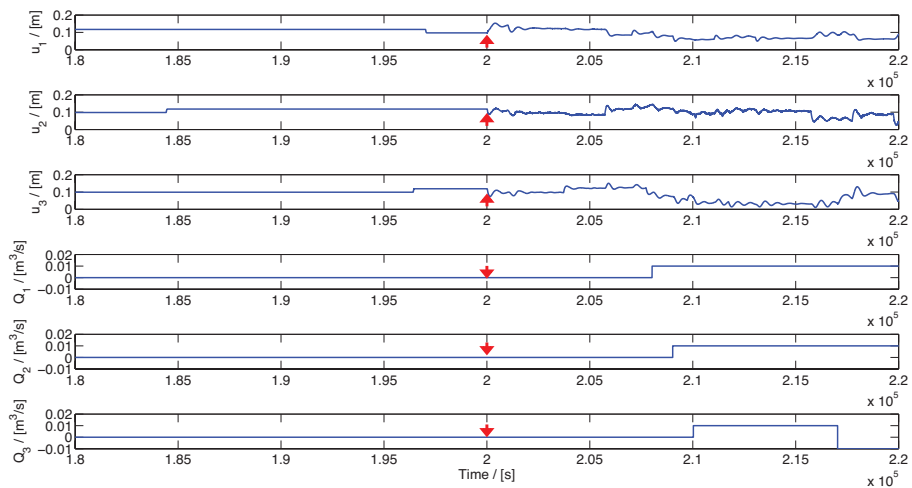
Figure 6.21: Representation of the manipulated variables u_i and flow drawn by the lateral off-takes Q_i, with the adaptive D-MPC with neighbouring coordination, in an experiment conducted in the SIMULINK canal model, with parameters $\rho_1 = 2000, \rho_2 = 1000, \rho_3 = 2000$ and $N = 35$. The maximum number of iterations n_I is 20. The controller step is switched on in time instant $t = 2 \times 10^5 s$.

7

Conclusions and Future Work

The objective of this study was to define adaptive distributed control algorithms based on the LQG and MPC control theory. The algorithms defined here are based on previous studies regarding the water canal considered in this study [6], [15], [16], [17]. The target plant was the SIMULINK non-linear canal model, which gives a good approximation of the canal dynamic behavior. The adaptive control algorithms were tested in a simulation environment, and in the different experiments conducted the results obtained were satisfactory, with a close approximation to the non-adaptive strategies in terms of performance.

Regarding the adaptive strategies considered in this study, one of the difficulties in the design of the controllers were identification problems regarding the selection of the forgetting factor, the initial parameter values and the covariance matrix. In the initial experiments, the RLS algorithm considered had a fixed value for λ. With a fixed value for the forgetting factor, there were several identification issues that prevented the controllers from working properly. A lower value of λ was useful whenever the prediction error was larger, but since the algorithm kept weighting less the past information, due to the slow system response, the parameter estimates and the trace of the covariance matrix started diverging. Since selecting $\lambda = 1$ was not the best approach, an alternative RLS algorithm was considered with a variable forgetting factor that depended on the prediction error, defined in section 2.5. Another difficulty in the definition of the identification algorithm was the parameter initialization. It would be preferable to begin the experiments with a higher uncertainty regarding the parameters of the system, but after performing several experiments, it was verified that the best approach is initializing the parameters closer to convergence with a lower uncertainty.

It would be interesting to define adaptation strategies with a different identification algorithm, such as the Recursive LASSO and compare the results obtained with the RLS algorithm here considered. Also a possible idea for future work is to define a new adaptation strategy in which it is possible to initialize the parameters with a larger uncertainty. Comparing the adaptive and non-adaptive control strategies, the results in general introduced more oscillations and variations in the system response but outputs converged to the corresponding reference signals. Although it increases the computational load and time, with more variables and steps required to compute the control inputs and although it also increases the complexity of the controller with a cost function that depends also on the parameter estimates, adaptation has the advantage of allowing the controller to be more sensitive and adaptable to changes in the system dynamics.

In a system with a time-variant dynamic behavior such as a water canal, with disturbances and several external factors that may produce changes in the system dynamics, the adaptive controllers have the advantage of computing its control inputs with knowledge of those changes in the system dynamics, preventing therefore operational issues.

Regarding the distributed control algorithms, three different approaches were considered, two of

which based on Game Theory concepts, where each local control agent had knowledge of the control inputs of its neighbours and the computation of the manipulated variables was accomplished with a coordination/negotiation step, and the other one based on an efficient distributed algorithm (D-ADMM) in which the computation of the manipulated variables was accomplished using an iterative procedure and augmented Lagrangian function. The results obtained with the three approaches were satisfactory, for the adaptive and non-adaptive algorithms, although it was difficult to define adaptive controllers without stability problems. It was interesting to see how the performance of this distributed techniques was close to the corresponding centralized solutions, with less communication steps between local control agents required.

An unexpected result was obtained using the MPC algorithm, in which the controller, with integral action, is working properly in the linearized model, but when applied to the SIMULINK non-linear canal model, the system output follows the reference with a small error, even with the presence of an integrator. This effect is constant throughout all the experiments conducted with algorithms based on the MPC theory and it appears to be related with either the position of the integrator and the linear incremental model, the incorrect use of sensor measurements and actuators or with the interactions between subsystems, since the error in each pool varies whenever a variation occurs in a neighbouring pool. It is intended to solve this issue with the MPC algorithms with integral action as future work.

Regarding the usage of LQG controllers and strategies based on MPC, the usage of MATLAB functions *dlqr* and *dlqe* from the Control Systems in the design of LQG control algorithms has advantages in terms of computational load and time. Two different optimization solutions were considered in the MPC strategies, in order to compute the manipulated variables. The usage of MATLAB optimization function *fminunc* was considered in the SISO and centralized MIMO algorithms, but the higher computational load and time when compared to an analytical solution made the second one preferred for the adaptive distributed algorithms. Although MPC has the advantage of being able to handle with operational constraints, the optimization process is more costly in terms of computational load. Because of this fact, and since adaptation also increases the computational load of the controller, no constraints were considered in the optimization process.

Due to the integral action issue verified in the MPC algorithms, it is difficult to make a comparison between the adaptive D-LQG and D-MPC algorithms. The system response obtained with the adaptive D-MPC algorithms has more oscillations but the quadratic weights are smaller in comparison with the D-LQG strategy. In general, since the MPC algorithms depend on the quadratic weights and on the value of the finite-time horizon, it is possible to obtain acceptable results with lower quadratic weights and a larger horizon. The adaptive D-MPC strategy based on Game Theory concepts is simpler and requires less variables in the coordination step. Both this D-MPC approach and the D-LQG algorithm have an easy implementation and their coordination procedures converge to the Nash equilibrium. An alternative to this algorithms is the usage of D-ADMM, based on Lagrangian optimization,

which is also efficient in terms of the communication steps required in the coordination procedure and avoids the problem of the Nash equilibrium being far from the global minimum [35].
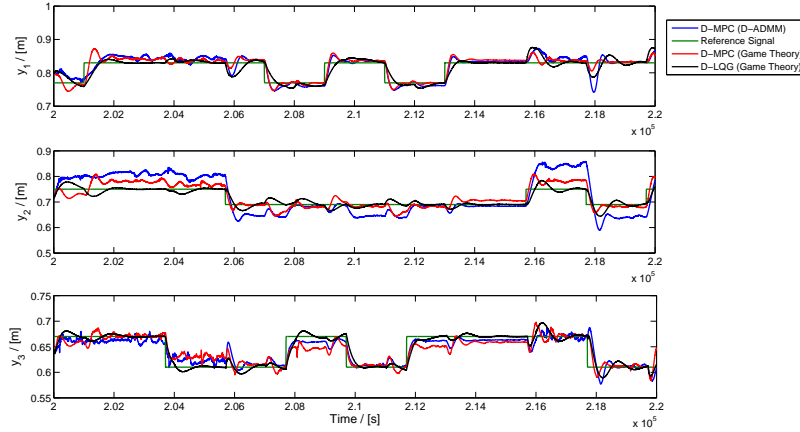
Figure 7.1: Results obtained with the three adaptive distributed strategies.

In general, the adaptive distributed algorithms have similar performances, as it is possible to verify in figure 7.1, where the results of applying the three adaptive distributed control algorithms are represented. The mean and variance of the output error $e(t) = y(t) - r(t)$, represented in table 7.1. As it is possible to verify, the results obtained with the adaptive D-MPC algorithms are similar, with a more oscillatory response than the obtained with the adaptive D-LQG algorithm. The D-LQG controller required less time for the identification algorithm to work isolated, although it was considered a multi-variable model with the effect of side-takes. The adaptive distributed algorithms based on Game Theory concepts have simpler negotiation/coordination strategies, with iterative procedures that require less operations and auxiliary variables, with the drawbacks mentioned above regarding the Nash equilibrium.

Table 7.1: Distributed Adaptive Control Algorithms

Algorithm	$\bar{e}$	σ_e^2
D-MPC (D-ADMM)	0.0036	0.0010
D-MPC (Game Theory)	0.0030	3.2304×10^{-4}
D-LQG (Game Theory)	1.7949×10^{-4}	2.2467×10^{-4}

Since the results shown in this dissertation were obtained in a simulation environment, this introduces a limitation regarding its application on the real plant. As future work, it would also be interesting to test the adaptive algorithms in a real water canal, in order to verify how the controller handles with disturbances and external factors that affect the system dynamics. Although there is still work to be done with new ideas and solutions to explore regarding adaptive and distributed algorithms applied to water canals, three different algorithms were developed and studied with satisfactory

results, complementing the work already developed and creating new challenges for future research.

Bibliography

[1] FAO, *Coping with water scarcity: An action framework for agriculture and food security.* Food and Agriculture Organization of the United Nations, 2012.

[2] M. Rijo, L. Rato, N. Nogueira, F. Machado, and J. Lemos, "Adaptive and non-adaptive model predictive control of an irrigation channel," *Networks and Heterogeneous Media*, vol. 4, no. 2, pp. 303–324, 2009.

[3] J. M. Lemos, L. F. Pinto, L. M. Rato, and M. Rijo, "Multivariable and Distributed LQG Control of a Water Delivery Canal," *Journal of Irrigation and Drainage Engineering*, vol. 139, no. 10, pp. 855–863, 2013. [Online]. Available: http://ascelibrary.org/doi/abs/10.1061/{%}28ASCE{%}29IR.1943-4774.0000621

[4] I. Mareels, E. Weyer, S. K. Ooi, M. Cantoni, Y. Li, and G. Nair, "Systems engineering for irrigation systems: Successes and challenges," *Annual Reviews in Control*, vol. 29, no. 2, pp. 191–204, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1367578805000271

[5] E. Weyer, "System identification of an open water channel," *Control Engineering Practice*, vol. 9, no. 12, pp. 1289–1299, 2001. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0967066101000995

[6] R. P. Costa, "Dual Optimization based Distributed Predictive Control of a Water Delivery Canal," Master's thesis, Instituto Superior Técnico, Lisboa, 2013.

[7] R. R. Negenborn, P.-J. Overloop, T. Keviczky, and B. Schutter, "Distributed model predictive control of irrigation canals," *Networks and Heterogeneous Media*, vol. 4, no. 2, pp. 359–380, 2009.

[8] J. M. Lemos, F. C. Machado, N. M. Nogueira, and P. O. Shirley, "Modelo SIMULINK de um canal piloto - Manual do utilizador," *INESC-ID, Relatório Técnico nº 35/2010*, no. 35, 2010. [Online]. Available: http://ramses.inesc.pt/AQUANET

[9] P. Malaterre and J. Baume, "Modeling and regulation of irrigation canals: existing applications and ongoing researches," *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*, vol. 4, pp. 3850–3855, 1998. [Online]. Available: http://canari.montpellier.cemagref.fr/papers/679.pdf

[10] J. Igreja and J. M. Lemos, "Nonlinear Model Predictive Control of a Water Distribution Canal Pool," *Nonlinear Model Predictive Control*, vol. 384, pp. 521–529, 2009. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-642-01094-1

[11] X. Litrico, V. Fromion, J. P. Baume, C. Arranja, and M. Rijo, "Experimental validation of a methodology to control irrigation canals based on Saint-Venant equations," *Control Engineering Practice*, vol. 13, no. 11, pp. 1425–1437, 2005.

[12] O. Su Ki, M. P. M. Krutzen, and E. Weyer, "On physical and data driven modelling of irrigation channels," *Control Engineering Practice*, vol. 13, no. 4, pp. 461–471, 2005.

[13] K. Eurén and E. Weyer, "System identification of open water channels with undershot and overshot gates," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 16, pp. 638–643, 2005.

[14] J. M. Lemos, "Distributed Adaptive Predictive Control of Water Delivery Networks," *IFAC Proceedings Volumes*, vol. 42, no. 20, pp. 156–161, 2009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1474667015361528

[15] L. M. Pinto, "Distributed LQG control of a Water Delivery Canal," Master's thesis, Instituto Superior Técnico, Lisboa, 2011.

[16] I. Sampaio, "Fault Tolerant Control of a Water Delivery Canal," Master's thesis, Instituto Superior Técnico, Lisboa, 2012.

[17] J. M. Igreja, F. M. Cadete, and J. M. Lemos, "Application of distributed model predictive control to a water delivery canal," *2011 19th Mediterranean Conference on Control & Automation (MED)*, pp. 682–687, 2011.

[18] A. Lacasta, M. Morales-Hernández, P. Brufau, and P. García-Navarro, "Simulation of PID control applied to irrigation channels," *Procedia Engineering*, vol. 70, pp. 978–987, 2014.

[19] S. K. Ooi and E. Weyer, "Control design for an irrigation channel from physical data," *Control Engineering Practice*, vol. 16, no. 9, pp. 1132–1150, 2008.

[20] Y. Bolea and V. Puig, "Gain-scheduling multivariable LPV control of an irrigation canal system," *ISA Transactions*, pp. 1–7, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.isatra.2016.03.009

[21] J. Igreja and J. Lemos, "Nonlinear Model Predictive Control of a Water Distribution Canal Pool," *Nonlinear Model Predictive Control*, vol. 384, pp. 521–529, 2009. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-642-01094-1

[22] J. M. Lemos and L. F. Pinto, "Distributed Linear-Quadratic Control of Serially Chained Systems - Application to a Water Delivery Canal," *IEEE Control Systems Magazine*, vol. 32, no. 6, pp. 26–38, 2012.

[23] H. E. Fawal, D. Georges, and G. Bornard, "Optimal control of complex irrigation systems via decomposition-coordination and the use of augmented Lagrangian," *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 4, pp. 3874 – 3879, 1998.

[24] S. Sawadogo, R. Faye, P. Malaterre, and F. Mora-Camino, "Decentralized predictive controller for delivery canals," *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*, vol. 4, pp. 3880–3884, 1998.

[25] Y. Zhang and S. Li, "Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes," *Journal of Process Control*, vol. 17, no. 1, pp. 37–50, 2007.

[26] P. Malaterre, "Regulation of irrigation canals: characterisation and classification," *Irrigation and Drainage Systems*, vol. 9, no. 4, pp. 297–327, 1995.

[27] A. Zafra-Cabeza, J. M. Maestre, M. a. Ridao, E. F. Camacho, and L. Sanchez, "Hierarchical distributed model predictive control for risk mitigation: An irrigation canal case study," *Proceedings of the 2011 American Control Conference*, vol. 21, pp. 3172–3177, 2011.

[28] G. F. Franklin, M. L. Workman, and J. D. Powell, *Digital Control of Dynamic Systems*, 3rd ed. Wesley, 1998.

[29] S. P. Sanoff and P. E. Wellstead, "Comments on: 'implementation of self-tuning regulators with variable forgetting factors'," *Automatica*, vol. 19, no. 3, pp. 345–346, May 1983.

[30] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control : Theory and Design*. Nob Hill Publishing, 2009.

[31] J. M. Lemos, R. Neves-Silva, and J. M. Igreja, *Adaptive Control of Solar Energy Collector Systems*. Springer, 2014.

[32] W. Kwon and S. Han, *Receding Horizon Control*. Springer, 2005. [Online]. Available: http://link.springer.com/10.1007/b136204

[33] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel, "D-ADMM: A Communication-Efficient Distributed Algorithm For Separable Optimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2718–2723, 2013.

[34] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel, "Distributed Optimization with Local Domains: Applications in MPC and Network Flows," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 2004–2009, 2015.

[35] R. P. Costa, J. M. Lemos, J. F. C. Mota, and J. M. F. Xavier, "D-ADMM based distributed MPC with input-output models," *2014 IEEE Conference on Control Applications, CCA 2014*, pp. 699–704, 2014.

A

System Parameters

Contents

A.1 SISO Model

The SISO linear incremental model is defined by the transfer function

$$A(q^{-1})\Delta y(t) = B(q^{-1})\Delta u(t) + e(t), \qquad (A.1)$$

$$A(q^{-1}) = 1 - 3.614q^{-1} + 4.986q^{-2} - 3.122q^{-3} + 0.7495q^{-4}, \quad B(q^{-1}) = -0.001068q^{-1} + 0.0009421q^{-2}. \qquad (A.2)$$

A.2 MIMO Model

The MIMO linear incremental model is defined by

$$\begin{bmatrix} A_{11}(q^{-1}) & 0 & 0 \\ 0 & A_{22}(q^{-1}) & 0 \\ 0 & 0 & A_{33}(q^{-1}) \end{bmatrix} \Delta y(t) = \begin{bmatrix} B_{11}(q^{-1}) & B_{12}(q^{-1}) & 0 \\ B_{21}(q^{-1}) & B_{22}(q^{-1}) & B_{23}(q^{-1}) \\ 0 & B_{32}(q^{-1}) & B_{33}(q^{-1}) \end{bmatrix} v(t) + e(t), \qquad (A.3)$$

$$A_{11}(q^{-1}) = 1 - 2.851q^{-1} + 2.758q^{-2} - 0.9068q^{-3}, \qquad (A.4)$$

$$B_{11}(q^{-1}) = -0.1815q^{-1} + 0.4018q^{-2} - 0.2707q^{-3} + 0.04811q^{-4}, \qquad (A.5)$$

$$B_{12}(q^{-1}) = 0.02407q^{-2} - 0.04456q^{-3} + 0.02032q^{-4}, \qquad (A.6)$$

$$A_{22}(q^{-1}) = 1 - 2.857q^{-1} + 2.779q^{-2} - 0.9222q^{-3}, \qquad (A.7)$$

$$B_{21}(q^{-1}) = 0.1315q^{-2} - 0.3324q^{-3} + 0.2696q^{-4} - 0.06537q^{-5}, \qquad (A.8)$$

$$B_{22}(q^{-1}) = -0.1548q^{-1} + 0.2994q^{-2} - 0.1476q^{-3}, \qquad (A.9)$$

$$B_{23}(q^{-1}) = 0.0875q^{-2} - 0.2274q^{-3} + 0.1997q^{-4} - 0.06001q^{-5}, \qquad (A.10)$$

$$A_{33}(q^{-1}) = 1 - 2.869q^{-1} + 2.807q^{-2} - 0.9379q^{-3}, \qquad (A.11)$$

$$B_{32}(q^{-1}) = 0.1002q^{-2} - 0.243q^{-3} + 0.1808q^{-4} - 0.03428q^{-5}, \qquad (A.12)$$

$$B_{33}(q^{-1}) = -0.05582q^{-1} - 0.04845q^{-2} + 0.2543q^{-3} - 0.1538q^{-4}. \qquad (A.13)$$

A.3 MIMO Model with the effect of side takes

The linear incremental model is defined as

$$A\Delta y(t) = Bv(t) + \Gamma Q(t) + e(t), \tag{A.14}$$

$$A = \begin{bmatrix} A_{11}(q^{-1}) & 0 & 0 \\ 0 & A_{22}(q^{-1}) & 0 \\ 0 & 0 & A_{33}(q^{-1}) \end{bmatrix}, \quad B = \begin{bmatrix} B_{11}(q^{-1}) & B_{12}(q^{-1}) & 0 \\ B_{21}(q^{-1}) & B_{22}(q^{-1}) & B_{23}(q^{-1}) \\ 0 & B_{32}(q^{-1}) & B_{33}(q^{-1}) \end{bmatrix}, \tag{A.15}$$

$$\Gamma = \begin{bmatrix} \Gamma_{11}(q^{-1}) & \Gamma_{12}(q^{-1}) & 0 \\ \Gamma_{21}(q^{-1}) & \Gamma_{22}(q^{-1}) & \Gamma_{23}(q^{-1}) \\ 0 & \Gamma_{32}(q^{-1}) & \Gamma_{33}(q^{-1}) \end{bmatrix}, \tag{A.16}$$

$$A_{11}(q^{-1}) = 1 - 2.824q^{-1} + 2.712q^{-2} - 0.8889q^{-3}, \tag{A.17}$$

$$B_{11}(q^{-1}) = -0.09133q^{-1} + 0.1726q^{-2} - 0.08407q^{-3}, \tag{A.18}$$

$$B_{12}(q^{-1}) = 0.09972q^{-2} - 0.2378q^{-3} + 0.1788q^{-4} - 0.04088q^{-5}, \tag{A.19}$$

$$\Gamma_{11}(q^{-1}) = -0.1994q^{-1} + 0.4028q^{-2} - 0.2079q^{-3}, \tag{A.20}$$

$$\Gamma_{12}(q^{-1}) = -0.07317q^{-1} + 0.2532q^{-2} - 0.2965q^{-3} + 0.1161q^{-4}, \tag{A.21}$$

$$A_{22}(q^{-1}) = 1 - 2.838q^{-1} + 2.746q^{-2} - 0.9085q^{-3}, \tag{A.22}$$

$$B_{21}(q^{-1}) = -0.007191q^{-1} + 0.01006q^{-2}, \tag{A.23}$$

$$B_{22}(q^{-1}) = -0.002756q^{-1}, \tag{A.24}$$

$$B_{23}(q^{-1}) = 0.006614q^{-1} - 0.006911q^{-2}, \tag{A.25}$$

$$\Gamma_{21}(q^{-1}) = -0.002298q^{-1} + 0.01624q^{-2} - 0.01423q^{-3}, \tag{A.26}$$

$$\Gamma_{22}(q^{-1}) = -1.796q^{-1} + 5.181q^{-2} - 5.073q^{-3} + 1.684q^{-4}, \tag{A.27}$$

$$\Gamma_{23}(q^{-1}) = 0.007431q^{-1} - 0.02024q^{-2} + 0.01235q^{-3}, \tag{A.28}$$

$$A_{33}(q^{-1}) = 1 - 2.839q^{-1} + 2.761q^{-2} - 0.9221q^{-3}, \tag{A.29}$$

$$B_{32}(q^{-1}) = 0.1302q^{-1} - 0.2668q^{-2} + 0.1412q^{-3}, \tag{A.30}$$

$$B_{33}(q^{-1}) = 0.4455q^{-1} - 1.631q^{-2} + 1.928q^{-3} - 0.7473q^{-4}, \tag{A.31}$$

$$\Gamma_{32}(q^{-1}) = 2.677 \times 10^{-6}q^{-2}, \tag{A.32}$$

$$\Gamma_{33}(q^{-1}) = -0.3264q^{-1} + 0.6154q^{-2} - 0.2959q^{-3}. \tag{A.33}$$

B

Pontryagin Minimum Principal

For a discrete-time system described by the non-linear equation

$$x(t+1) = f(x(t), v(t), t), \tag{B.1}$$

in which f is a function that describes the system dynamics. The initial condition $x(0)$ is specified and a performance index is defined as

$$J(u) = \Phi(T, x(T)) + \sum_{t=0}^{\infty} L(x(t), v(t), t), \tag{B.2}$$

in which L denotes the Lagrangian function. The objective is to minimize the performance index J, assuming that $T \leftarrow \infty$ and that there are no constraints either on the final state $x(t)$ or on the manipulated variable v.

The Hamiltonian function H is defined as

$$H(t) = \lambda^T(t+1)f(x(t), v(t), t) + L(x(t), v(t), t), \tag{B.3}$$

in which λ is the co-state. The discrete-time Pontryagin Minimum Principle states that the co-state λ, the optimal control input v and state trajectory x satisfy:

- State equation (B.1) with $x(0)$ specified.

- Adjoint equation:

$$\lambda(t) = \left(\frac{\partial f(x(t), v(t), t)}{\partial x(t)}\right)^T \lambda(t+1) + \left(\frac{\partial L(x(t), v(t), t)}{\partial x(t)}\right)^T \tag{B.4}$$

- Stationary condition:

$$\frac{\partial H(t)}{\partial v(t)} = 0 \tag{B.5}$$

- Co-state terminal condition:

$$\lambda(T) = \frac{\partial \Phi(T, x(T))}{\partial x(T)}. \tag{B.6}$$