



Dual Optimization Based Distributed Predictive Control of a Water Delivery Canal

Rafael Ortega Baptista Pestana da Costa

Thesis to obtain the Master of Science degree in

Electrical and Computer Engineering

Examination Committee

Chairperson: Prof. Dr. João Fernando Cardoso Silva Sequeira

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

Members of the Committee: Prof. Dr. João Manuel de Freitas Xavier

October 2013

“(...)And the Spirit of God moved upon the face of the waters.”

Genesis 1:2

Acknowledgments

I would like to express my gratitude to Instituto Superior Técnico and to all my professors and colleagues for their contribution and help in the past six years. Among those Professor João Miranda Lemos has to be referred, as he was my supervisor in this thesis and also his lessons in the discipline of Modelling and Simulation which were determinant for me in choosing the Systems, Decision and Control Major.

A special word for Professor João Xavier and to João Mota who helped me a lot in the study of the method chosen to deal with the algorithm used in the decentralization.

I would also like to thank all friends and family who gave me support during my studies at Instituto Superior Técnico, specially to my parents.

This work was partly supported by FCT (Portugal) under project ORCHESTRA, contract PTDC/EMS-CRO/2042/2012 and INESC-ID funds under contract PEst-OE/EEI/LA0021/2013.

Abstract

The purpose of this work is to control the water level in a water delivery canal composed of three pools and three undershot gates.

This work proposes a distributed predictive controller for the water delivery canal using an Input-Output model. The distributed algorithm was implemented with the Distributed Alternating Direction Method of Multipliers algorithm.

The controller developed was validated in simulations and it has a similar behaviour to experimental results obtained with other controllers.

Key Words:

Model Predictive Control (MPC), Distributed Alternating Direction Method of Multipliers (D-ADMM), irrigation canal, Input-Output Model

Resumo

O objectivo deste trabalho é controlar o nível de água num canal de rega composto por três secções e três comportas.

Esta tese propõe um controlador preditivo distribuído para o canal, baseado em modelos Entrada-Saída. O algoritmo distribuído foi implementado com o algoritmo Distributed Alternating Direction Method of Multipliers.

O controlador desenvolvido foi validado através de simulações e tem um comportamento semelhante a resultados experimentais obtidos com outros controladores.

Palavras Chave:

Controlo Preditivo, Distributed Alternating Direction Method of Multipliers, Canal de rega, Modelo Entrada Saída

Contents

1	Introduction	13
1.1	Motivation	14
1.2	Literature Review	14
1.3	Problem Formulation	15
1.4	Contributions	15
2	Canal Model	17
2.1	Canal Description	18
2.2	Physical Model	19
2.3	Linearized Model	20
3	Model Predictive Control	23
3.1	MPC construction	24
3.2	Integral Effect	26
3.3	Canal MPC	27
4	Tuning the centralized algorithm	31
4.1	Time Horizon	32
4.2	Control Cost Weights	33
5	Distributed Control	35
5.1	D-ADMM	36
5.2	Node associated Dual variables	38
5.3	Edge associated Dual variables	38
5.4	D-ADMM Parameter Tuning	42
6	Simulation Results	43
7	Conclusions and Future Work	49
Bibliography		51
A	Input-Output Form Coefficients	55
A.1	SISO Coefficients	55
A.2	MISO Coefficients	55
B	Poles and Zeros	57
C	Control costs	61

D D-ADMM conditions	65
D.1 Condition 1	65
D.2 Condition 2	65
D.3 Condition 3	65

List of Figures

2.1	Núcleo de Hidráulica e Controlo de Canais (NuHCC) Canal schematic	18
2.2	Undershoot gate	18
2.3	Overshot gate	19
2.4	Qualitative response of the linearized Multiple Input-Multiple Output (MIMO) system	20
2.5	Frequency response of inputs 1 and 2 in pool 1	21
2.6	Frequency response of inputs 1, 2 and 3 in pool 2	21
2.7	Frequency response of inputs 2 and 3 in pool 3	22
3.1	System and Model Predictive Control (MPC)	24
3.2	System's response, $\rho = 1000$ and $T=50$	26
3.3	System with MPC and integrator	26
3.4	System's response with integral effect, $\rho = 1000$ and $T=50$	27
3.5	System's response to steps with integral effect in the presence of white noise	27
3.6	First Gate response, centralized solution, $\rho_1 = 1000$, $T=50$	29
3.7	Second Gate response, centralized solution, $\rho_2 = 1000$, $T=50$	30
3.8	Third Gate response, centralized solution, $\rho_3 = 1000$, $T=50$	30
4.1	Error evolution with T varying from 15 to 75, centralized solution, $\rho_1 = 500$, $\rho_2 = 500$, $\rho_3 = 500$	32
4.2	Error sum evolution with T varying from 15 to 75, centralized solution, $\rho_1 = 500$, $\rho_2 = 500$, $\rho_3 = 500$	33
4.3	Error sum, $\rho_3 = 200$, $T=45$	34
5.1	Example of communication between controllers	36
5.2	Network topology	36
5.3	First Gate response, D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 200$, $\rho_2 = 600$, $\rho_3 = 200$, $T=45$	40
5.4	Second Gate response, D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 200$, $\rho_2 = 600$, $\rho_3 = 200$, $T=45$	41
5.5	Third Gate response, D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 200$, $\rho_2 = 600$, $\rho_3 = 200$, $T=45$	41
5.6	Error Evolution of D-ADMM solution, $\rho_1 = 200$, $\rho_2 = 600$, $\rho_3 = 200$, $T=45$	42
6.1	Comparison with [10], D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 1000$, $\rho_2 = 1000$, $\rho_3 = 1000$, $T=45$, Gate 1	44
6.2	Comparison with [10], D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 1000$, $\rho_2 = 1000$, $\rho_3 = 1000$, $T=45$, Gate 2	45

6.3	Comparison with [10], D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 1000$, $\rho_2 = 1000$, $\rho_3 = 1000$, T=45, Gate 3	45
6.4	Comparison with [17], D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 1000$, $\rho_2 = 1000$, $\rho_3 = 1000$, T=45, Gate 1	46
6.5	Comparison with [17], D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 1000$, $\rho_2 = 1000$, $\rho_3 = 1000$, T=45, Gate 2	46
6.6	Comparison with [17], D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 1000$, $\rho_2 = 1000$, $\rho_3 = 1000$, T=45, Gate 3	47
B.1	Poles and zeros of the $\frac{y_1}{u_1}$ Single Input-Single Output (SISO) sytstem	57
B.2	Poles and zeros of the $\frac{y_1}{u_2}$ SISO sytstem	57
B.3	Poles and zeros of the $\frac{y_2}{u_1}$ SISO sytstem	58
B.4	Poles and zeros of the $\frac{y_2}{u_2}$ SISO sytstem	58
B.5	Poles and zeros of the $\frac{y_2}{u_3}$ SISO sytstem	58
B.6	Poles and zeros of the $\frac{y_3}{u_2}$ SISO sytstem	59
B.7	Poles and zeros of the $\frac{y_3}{u_3}$ SISO sytstem	59
C.1	Error sum, $\rho_3 = 300$, T=45	61
C.2	Error sum, $\rho_3 = 400$, T=45	61
C.3	Error sum, $\rho_3 = 500$, T=45	62
C.4	Error sum, $\rho_3 = 600$, T=45	62
C.5	Error sum, $\rho_3 = 700$, T=45	62
C.6	Error sum, $\rho_3 = 800$, T=45	63
C.7	Error sum, $\rho_3 = 900$, T=45	63
C.8	Error sum, $\rho_3 = 1000$, T=45	63

List of Acronyms

ADMM Alternating Direction Method of Multipliers

D-ADMM Distributed Alternating Direction Method of Multipliers

LQ Linear Quadratic

MIMO Multiple Input-Multiple Output

MISO Multiple Input-Multiple Output

MPC Model Predictive Control

NuHCC Núcleo de Hidráulica e Controlo de Canais

PI Proportional-Integral

SISO Single Input-Single Output

Chapter 1

Introduction

Contents

1.1	Motivation	14
1.2	Literature Review	14
1.3	Problem Formulation	15
1.4	Contributions	15

1.1 Motivation

Water is considered the most vital resource. The water consumption can be divided in three major areas: human consumption (drinking, cooking, ...), industrial use and agricultural use. As far as agricultural use is concerned, most water usage is for field irrigation. Irrigation techniques were developed throughout history, since early agricultural societies in Mesopotamia and Egypt, to modern day technologies.

The agricultural use of fresh water is accounted for about 66% of water withdrawals and 85% of water consumption worldwide[27]. With an increasingly growing population the volume of water needed to agricultural purposes will increase. Therefore the developing of more efficient methods of irrigation is of utmost importance in order to minimize water losses.

Irrigation systems are usually composed by water canals of very large dimensions, divided into stretches by gates. Although other objectives can be considered in relation to the operation of this type of plants, here we want to design the control algorithm to keep the water level at a desired value at specified points along the canal. Furthermore, due to the spatial distribution of the system a distributed control structure is to be used.

Various techniques have been used to cope with this problem. Model Predictive Control (MPC), a widely used technique in industrial environment, was chosen to control the water level in the canal. Dual optimization techniques will be used to achieve control distribution.

1.2 Literature Review

Control of water delivery canals has been a topic of interest in the control engineering community for the past decades.

Such systems can be modeled using the Saint-Venant equations, that are formed by a system of partial differential equation that embed mass and momentum conservation, [32], [15], [8], [29], [14] or models identified directly from plant data [10], [17], [3], [25].

Several control methods have been used in the context of water delivery canal control with satisfactory results [18]. Proportional-Integral (PI) [13], Linear Quadratic (LQ) [31], fuzzy [5], predictive [1], non linear [4], robust [28], [16] and adaptative [1] control are amongst the techniques used.

Interest in MPC started in the 1980s and it became a widely used technique in industrial control [19]. MPC application to water delivery canals has already been studied [22], [29].

Decentralized and Distributed control are also hot topics in the scientific community. According to [22] decentralized controllers *are developed for local control, without taking into account the effects that local actions have on the overall system performance*. In the same paper Distributed Controller is defined as *local controllers may be designed in such a way that they take into account the effects of local actions at a system-wide level using information exchange*.

There is much work developed both in Decentralized [30], [2], [11] and in Distributed Control [12], [9], [23], [33]. In particular, [7] uses an algorithm where distribution is obtained on the basis of dual control with an augmented lagrangian. Similarly, in this dissertation a Distributed MPC algorithm is obtained based on the Distributed Alternating Direction Method of Multipliers (D-ADMM) optimization algorithm [21], the distinguishing feature being that *i/o* predictive models are employed, while previous works always used state space models.

1.3 Problem Formulation

This dissertation focuses on the development of a distributed controller for a water delivery canal. The canal is divided into subsystems (a pool and a gate) and the goal is to maintain the pool downstream water level at a reference value.

There will be local controllers for each of the three subsystems. Due to the option for distributed control the need for an algorithm that leads to a compromise between local control variables arise.

To address this problem, local cost functions reflecting the deviation os the water level from the reference level and the control variables effort are defined. These local cost functions are minimized in a receeding horizon sense. Furthermore, in order to coordinate directions, a dual method with an agmented lagrangian function is used.

1.4 Contributions

The main contribution of this thesis is the developement of a MPC algorithm that combines local *i/o* predictive models with the D-ADMM algorithm. The application of this controller to a water delivery canal is also studied, in particular by assessing the influence of the parameters that define the algorithm configuration.

Chapter 2

Canal Model

Contents

2.1	Canal Description	18
2.2	Physical Model	19
2.3	Linearized Model	20

In this chapter the canal will be described and the linearized model which will be used to implement the algorithm presented.

2.1 Canal Description

The canal upon which the algorithm will be developed belongs to Núcleo de Hidráulica e Controlo de Canais (NuHCC), of the University of Évora, Portugal. Several MSc Thesis [17], [10] and scientific articles [15], [26] have used this canal to test control systems (either in actual field experiments or with its model). The canal is composed by an automatic canal and a traditional one. The automatic canal is shown in figure 2.1.

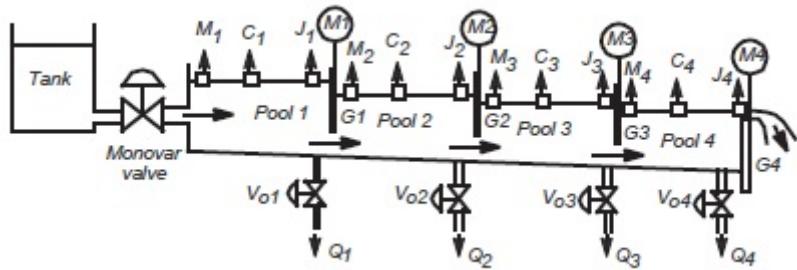


Figure 2.1: NuHCC Canal schematic

This canal is composed by four pools with a gate separating each other. There are three undershot gates 2.2 and one overshot gate 2.3. The overshot gate of the canal is not very reliable, so the algorithm will only be implemented using the three undershot gates. A traditional canal links the water leaving the automatic canal through the overshot gate to a water reservoir upstream pool 1, preventig water losses during field experiments.



Figure 2.2: Undershoot gate



Figure 2.3: Overshot gate

2.2 Physical Model

A canal model can be constructed using the Saint-Venant equations [17], [15], and they have been used by engineers for over 100 years. Under the following assumptions:

- The bed slope S_0 is small enough ($\sin S_0 \approx S_0$);
- Water density is constant;
- Water flow is unidimensional;
- The cross section pressure is hydrostatic;
- There are no lateral discharges;
- The effects of internal viscosity are negligible relatively to external friction;

The next two equations hold:

$$\frac{\partial Q(x, t)}{\partial x} + \frac{\partial A(x, t)}{\partial t} = 0, \quad (2.1)$$

$$\frac{\partial Q(x, t)}{\partial t} + \frac{\partial}{\partial x} \frac{Q^2(x, t)}{A(x, t)} + gA(x, t) \frac{\partial h(x, t)}{\partial x} = gA(x, t)(S_0(x) - S_f(x, t)). \quad (2.2)$$

The first equation is obtained by the mass conservation and the second one by the momentum conservation. Q is the water flow [$m^3 s^{-1}$] through section A [m^2], h is the water depth [m], S_f is the friction slope and g is the gravitational acceleration [ms^{-2}].

The equations that describe the water flow through a gate are

$$Q = K_{ds} A \sqrt{2g(h_b - h_a)} \quad (2.3)$$

for an undershot gate, and

$$Q = BK_{ds} \sqrt{2g(h_b - h_g)^3} \quad (2.4)$$

for an overshot gate.

In these equations K_{ds} is the discharge coefficient, h_b is the water level before the gate, h_a is the water level after the gate, h_g is the gate level, B is the width of the overshot gate and A is the effective area of the orifice for undershot gate.

The use of this equations to model the canal would require a lot of computational effort and their application to the real world may not be totally accurate [32] outside laboratory canals. An identification model will be used instead.

2.3 Linearized Model

It was proved that identification models can have the same performance as models based on the Saint-Venant equations, but requiring a lot less computational effort to achieve said performances [24].

In this work a system identification of the canal [17] will be used, but other authors have been using system identification models in water delivery canal control context [10], [6].

The data to be used is in State Space form and will be converted to input-output form.

There will be two Models to work with. A Single Input-Single Output (SISO) model, used in the initial development of the MPC and Multiple Input-Multiple Output (MISO) models used in the control of each pool for multiple pool control scenario. The coefficients of such models are in A.

The MISO models are model the local behaviour of a Multiple Input-Multiple Output (MIMO) model. The response of the MIMO model to an impulse step in each controller is shown in 2.4.

It can be seen that when there is an impulse in gate 1 the upstream level of pool 1 gets lower and the upstream level of pool 2 gets higher. Pool 3 maintains its level. This is logical, because if gate 1 opens more water flows from pool 1 to pool 2. When an impulse is sent to gate 2, the upstream level in pool 2 gets lower and the upstream level in gate 3 pool higher. In pool 1 the level gets lower too, as the difference between the water level in pool 1 and 2 is bigger now, which will increase the flow through gate 1 (more detailed explanation in [10]). For an impulse in gate 3 the upstream level in pool 3 decreases and the same happens in pool 2.

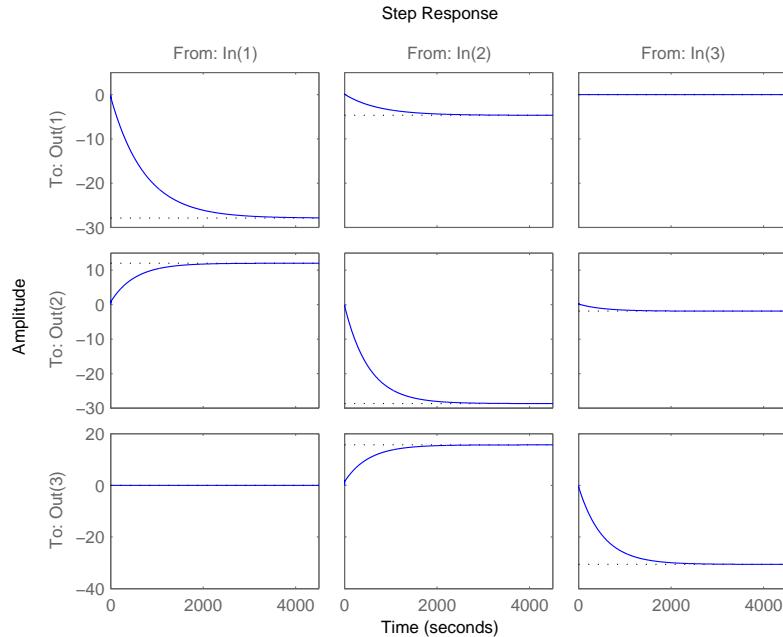


Figure 2.4: Qualitative response of the linearized MIMO system

In figures 2.5 to 2.7 the frequency response of SISO models are shown. The notation s_{ij} stands for the system in which the output is the water level in pool i and the input is opening of gate j.

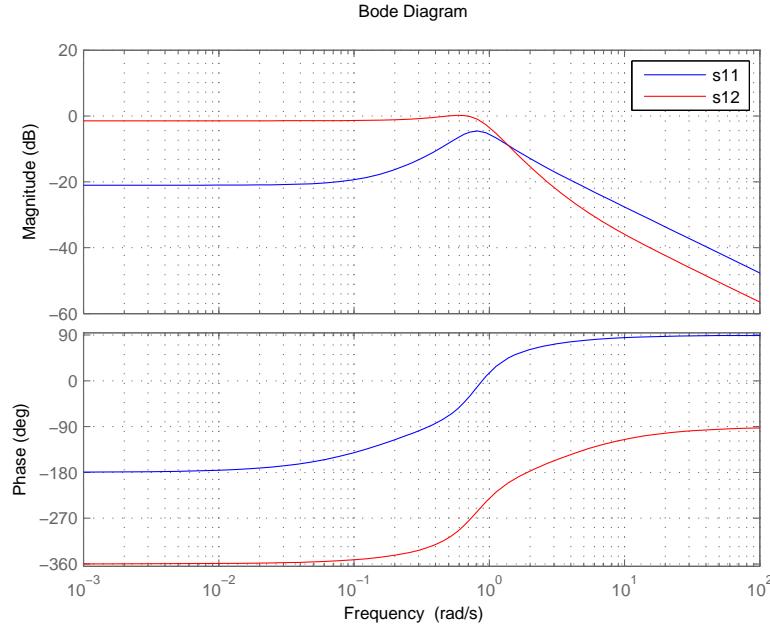


Figure 2.5: Frequency response of inputs 1 and 2 in pool 1

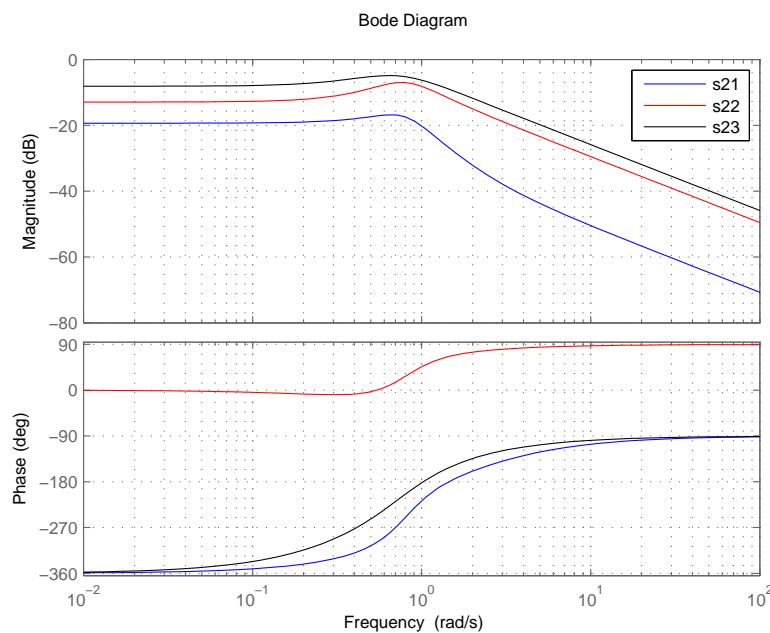


Figure 2.6: Frequency response of inputs 1, 2 and 3 in pool 2

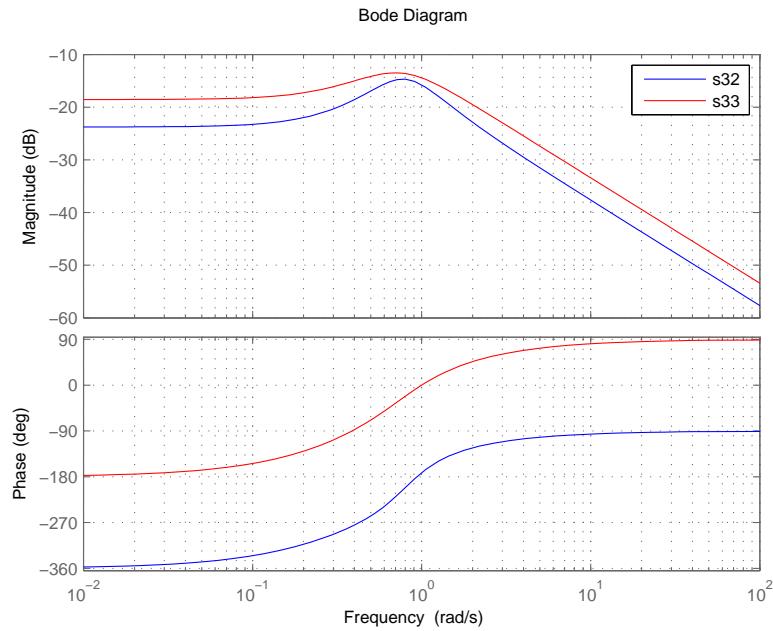


Figure 2.7: Frequency response of inputs 2 and 3 in pool 3

The poles and zeros of the SISO systems are shown in B. All the systems are stable, although small gains would make them unstable. A proportional controller is therefore inappropriate to control such systems.

Chapter 3

Model Predictive Control

Contents

3.1	MPC construction	24
3.2	Integral Effect	26
3.3	Canal MPC	27

This chapter provides the tools required to define the local controllers. The MPC will be described and the problem cost function is defined. Predictors using an *i/o* will be obtained and the problem will be put inmatrix form. The integral effect introduction is explained and the MPC is expanded to cope with the interaction between pools.

3.1 MPC construction

In a first phase a single MPC for the system's linear model will be developed. Having the system as

$$A(q)y(k) = B(q)u(k) + C(q)e(k), \quad (3.1)$$

it can be rewritten as

$$y(k) = -a_1y(k-1) - \dots - a_ny(k-n) + b_0u(k-1) + \dots + b_mu(k-m) + e(k) + \dots + c_ne(k-n). \quad (3.2)$$

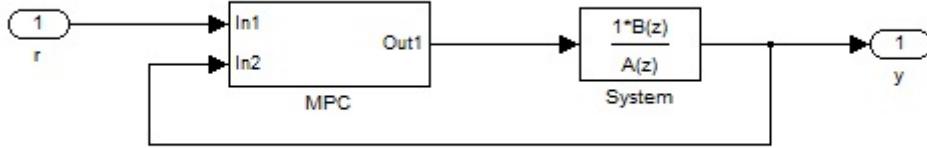


Figure 3.1: System and MPC

The system transfer function is

$$\frac{y(k)}{u(k)} = \frac{B^*(q)}{A(q)}. \quad (3.3)$$

The MPC will compute the system's control variable for an established time horizon, by minimizing the cost

$$J = \sum_{i=1}^N \left[[y(k+i) - r(k+i)]^2 + \rho u(k+i-1)^2 \right]. \quad (3.4)$$

Where $r(k+i)$ is the reference and ρ is a control penalty term. This minimization is subjected to the following constraints:

$$\begin{cases} u_{min} < u < u_{max} \\ slewrate_{min} < u(k+i+1) - u(k+i) < slewrate_{max} \end{cases}$$

The future values of y will be reflected on the cost. A way to predict them becomes necessary.

$$A^*(q^{-1})y(k+1) = B^*(q^{-1})u(k) \quad (3.5)$$

Knowing

$$\frac{1}{A^*(q^{-j})} = F_j^* + q^{-j} \frac{G_j^*}{A^*(q^{-j})}, \quad (3.6)$$

then

$$A^*(q^{-j})F_j^* = 1 - q^{-j}G_j^*. \quad (3.7)$$

The product of (3.5) with F_j^* is

$$A^*(q^{-j})F_j^*y(k+j) = B^*(q^{-j})F_j^*u(k+j-1), \quad (3.8)$$

and by replacing (3.7) in (3.8)

$$\begin{aligned} (1 - q^{-1}G_j^*)y(k+j) &= B^*(q^{-j})F_j^*u(k+j-1) \Leftrightarrow \\ \Leftrightarrow y(k+j) &= G_j^*y(k) + B^*(q^{-j})F_j^*u(k+j-1). \end{aligned} \quad (3.9)$$

and thus a predictor is obtained. It will be replaced in (3.4) and therefore the cost will be dependent on known values of y and u , and also future values of u , for which the cost minimization is intended.

Putting (3.9) in a matrix form

$$\begin{bmatrix} y(k+1) \\ y(k+2) \\ \dots \\ y(k+N) \end{bmatrix} = W \begin{bmatrix} u(k) \\ u(k+1) \\ \dots \\ u(k+M) \end{bmatrix} + \Pi \begin{bmatrix} y(k) \\ \dots \\ y(k-n) \\ u(k-1) \\ \dots \\ u(k-m+1) \end{bmatrix}. \quad (3.10)$$

Matrix Π elements that multiply known values of y and are obtained from G_j^* . The remaining elements of Π and matrix W are obtained by computing $B^*(q^{-j})F_j^*$.

$$\begin{aligned} B^*(q^{-j})F_j^* &= (b_0 + b_1z^{-1} + \dots + b_mz^{-m})(1 + f_1z^{-1} + \dots + f_{j-1}z^{-j+1}) = \\ &= \underbrace{b_0}_{w_1} + \underbrace{(b_0f_1 + b_1)}_{w_2}z^{-1} + \underbrace{(b_0f_2 + b_1f_1 + b_2)}_{w_3}z^{-2} + \dots + \\ &+ \underbrace{(b_0f_{j-1} + b_1f_{j-2} + \dots + b_{j-1})}_{w_j}z^{-j+1} + \\ &+ \underbrace{(b_0f_j + \dots + b_j)}_{\pi_{n+1}^j}z^{-j} + \dots + \underbrace{(b_0f_m + \dots + b_m)}_{\pi_{n+m}^j}z^{-m}. \end{aligned} \quad (3.11)$$

$$W = \begin{bmatrix} w_1 & 0 & \dots & 0 \\ w_2 & w_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ w_j & w_{j-1} & \dots & w_1 \end{bmatrix}; \quad \Pi = \begin{bmatrix} [G_j^*] & \pi_{n+1}^j \\ \dots & \dots \\ \pi_{n+m}^j & \pi_{n+m}^j \end{bmatrix}. \quad (3.12)$$

After obtaining the matrixes a prediction for future values of y is possible. Therefore the cost can be minimized. The MatLab function fmincon, cvx or an analytical approach can be used to obtain the inputs which minimize J . In the figures presented below, an analytical solution was used.

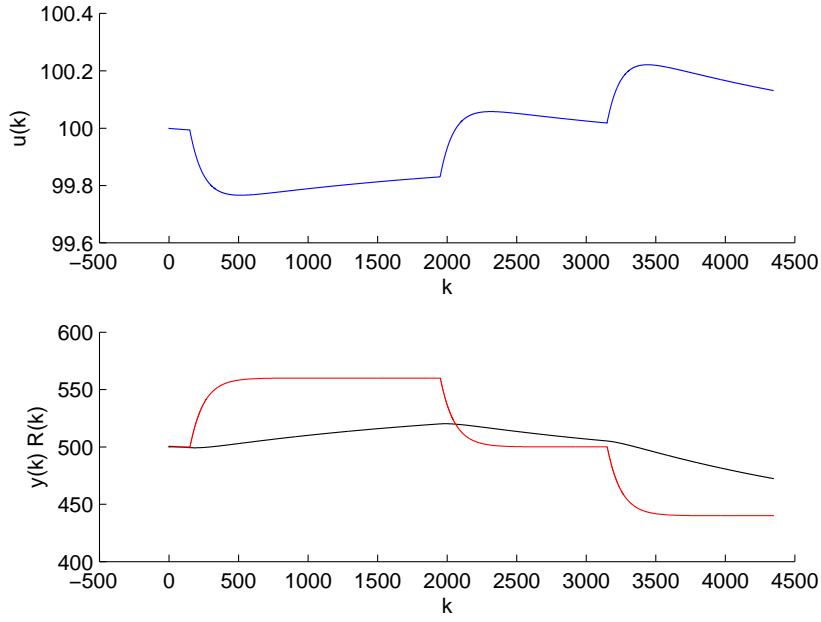


Figure 3.2: System's response, $\rho = 1000$ and $T=50$

As observed in the figure, the response has a very large static error compared to the reference variations, and it takes a very long time to increase and decrease. It takes so much time that, in fact, the changes in the water level are almost inexistent. To prevent this situation an integrator after the MPC's output will be introduced.

3.2 Integral Effect

The matrixes Π and W will change as the system, from the MPC's point of view, is no longer $\frac{B}{A}$ but $\frac{B}{\Delta A}$. The original system (the actuator of the canal gate) will get as input $\Delta u = u(k) - u(k - 1)$. The introduction of an integrator will also allow the rejection of small disturbances. An example of the MPC working with an integrator is shown below.

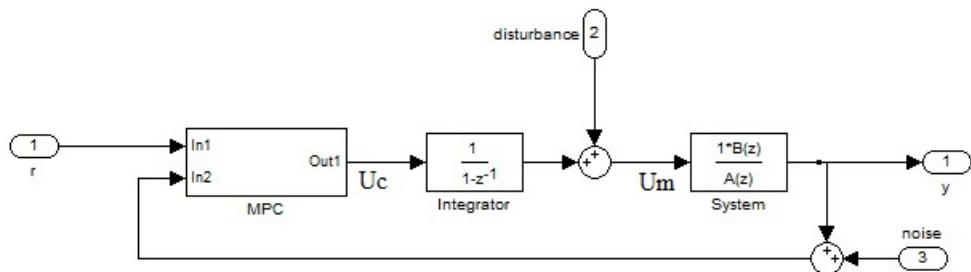


Figure 3.3: System with MPC and integrator

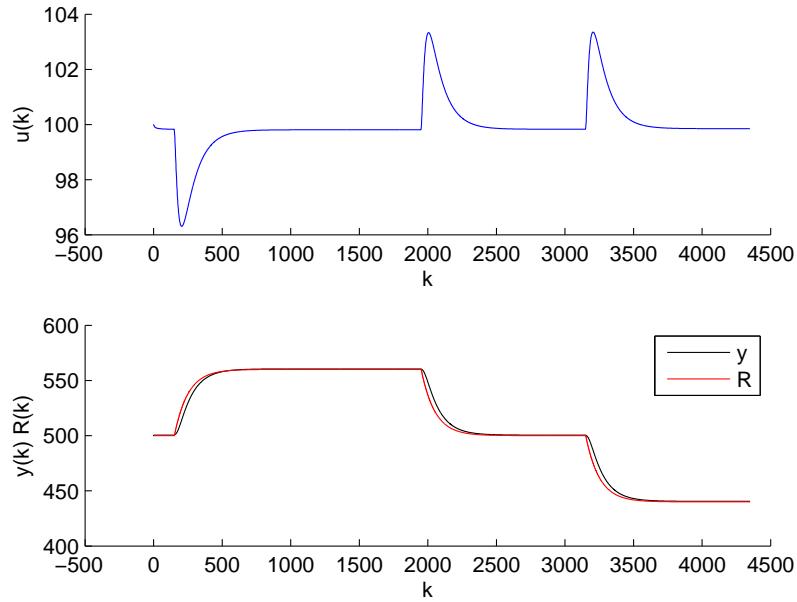


Figure 3.4: System's response with integral effect, $\rho = 1000$ and $T=50$

The system's response in the presence of white noise is shown below.

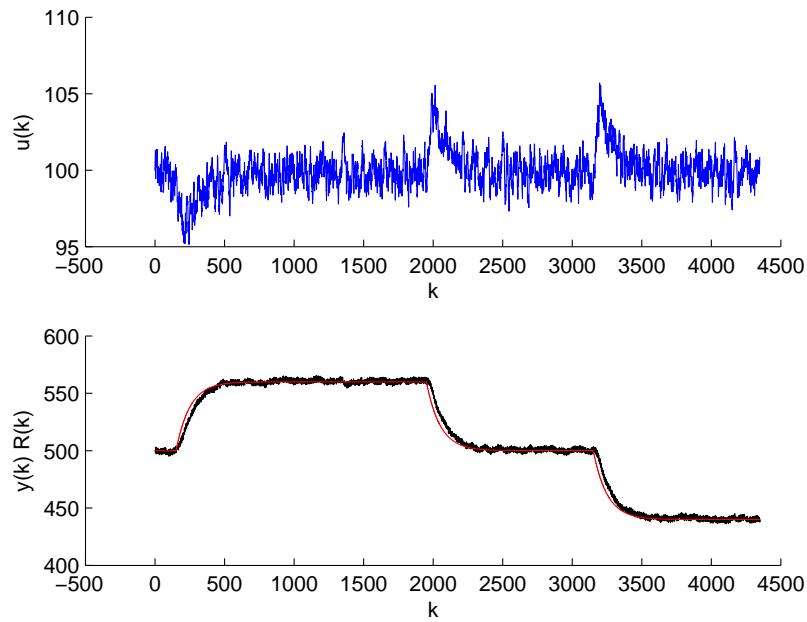


Figure 3.5: System's response to steps with integral effect in the presence of white noise

3.3 Canal MPC

The real system is not composed by a single gate. The flow through a gate that controls a section above or below has an effect on the water level of a canal section. Thus, the problem must be rewritten to get into account these phenomena. Let us analyse more than one canal gate at once. The equation

(3.1) will change to reflect the influence that the canal gates have in another pool.

$$A_1(q^{-1})y_1 = B_1(q)u_1 + P(q)u_2 + Ce_1. \quad (3.13)$$

Let us assume the dimensions of B and P are the same.

$$A_1^*(q^{-1})y_1(k+1) = B_1^*(q^{-1})u_1(k) + P^*(q^{-1})u_2(k) + C^*e_1(q^{-1}). \quad (3.14)$$

Having

$$\frac{1}{A_1^*} = F_j^* + q^{-j} \frac{G_j^*}{A_1^*}, \quad (3.15)$$

one can write

$$(1 - q^{-j}G_j^*)y_1(k+j) = B_1^*F_j^*u_1(k) + P^*F_j^*u_2(k) \Leftrightarrow \quad (3.16)$$

$$y_1(k+j) = G_j^*y_1(k) + B_1^*(q^{-1})F_j^*u_1(k+j-1) + P^*F_j^*u_2(k+j-1). \quad (3.17)$$

The equation 3.17 describes the predictors of y_1 taking into account the effect in y_1 caused by the control variable of gate 2.

Assuming the models are different due to different physical dimensions of the canal sections, and that only the future values of the control variables will affect the water levels in the other pool:

$$\begin{bmatrix} y_1(k+1) \\ \dots \\ y_1(k+T) \\ y_2(k+1) \\ \dots \\ y_2(k+T) \end{bmatrix} = \begin{bmatrix} W_1 & W_{12} \\ W_{21} & W_2 \end{bmatrix} \begin{bmatrix} u_1(k) \\ \dots \\ u_1(k+T-1) \\ u_2(k) \\ \dots \\ u_2(k+T-1) \end{bmatrix} + \begin{bmatrix} \Pi_1 & 0 \\ 0 & \Pi_2 \end{bmatrix} \begin{bmatrix} y_1(k) \\ \dots \\ y_1(k-n+1) \\ u_1(k-1) \\ \dots \\ u_1(k-m) \\ y_2(k) \\ \dots \\ y_2(k-n+1) \\ u_2(k-1) \\ \dots \\ u_2(k-m) \end{bmatrix}. \quad (3.18)$$

To introduce a third gate it will be assumed that the effect of gate $i+2$ in gate i is zero and the contrary is also true. The predictors will be

$$Y = \begin{bmatrix} W_1 & W_{12} & 0 \\ W_{21} & W_2 & W_{23} \\ 0 & W_{32} & W_3 \end{bmatrix} U + \begin{bmatrix} \Pi_1 & 0 & 0 \\ 0 & \Pi_2 & 0 \\ 0 & 0 & \Pi_3 \end{bmatrix} s. \quad (3.19)$$

with:

$$Y = \begin{bmatrix} (y_1)_{k+1}^{k+T} \\ (y_2)_{k+1}^{k+T} \\ (y_3)_{k+1}^{k+T} \end{bmatrix}; \quad (3.20)$$

$$U = \begin{bmatrix} (u_1)_k^{k+T-1} \\ (u_2)_k^{k+T-1} \\ (u_3)_k^{k+T-1} \end{bmatrix}; \quad (3.21)$$

$$s = \begin{bmatrix} (y_1)_k^{k-n+1} \\ (u_1)_{k-1}^{k-m} \\ (y_2)_k^{k-n+1} \\ (u_2)_{k-1}^{k-m} \\ (y_3)_k^{k-n+1} \\ (u_3)_{k-1}^{k-m} \end{bmatrix}; \quad (3.22)$$

A new cost function can be defined for a centralized controller

$$J_{cent} = (Y_{cent} - R_{cent})^2 + \rho \times U_{cent}^2. \quad (3.23)$$

The Matlab function fmincon will be used to minimize

$$J_{cent} = \left\| \begin{bmatrix} W_1 & W_{12} & 0 \\ W_{21} & W_2 & W_{23} \\ 0 & W_{32} & W_3 \end{bmatrix} U + \begin{bmatrix} \Pi_1 & 0 & 0 \\ 0 & \Pi_2 & 0 \\ 0 & 0 & \Pi_3 \end{bmatrix} s - \begin{bmatrix} R_1(k) \\ R_2(k) \\ R_3(k) \end{bmatrix} \right\|^2 + \rho \times \|U\|^2. \quad (3.24)$$

The resulting vector U will minimize J_{cent} in the specified time horizon. The results are shown below.

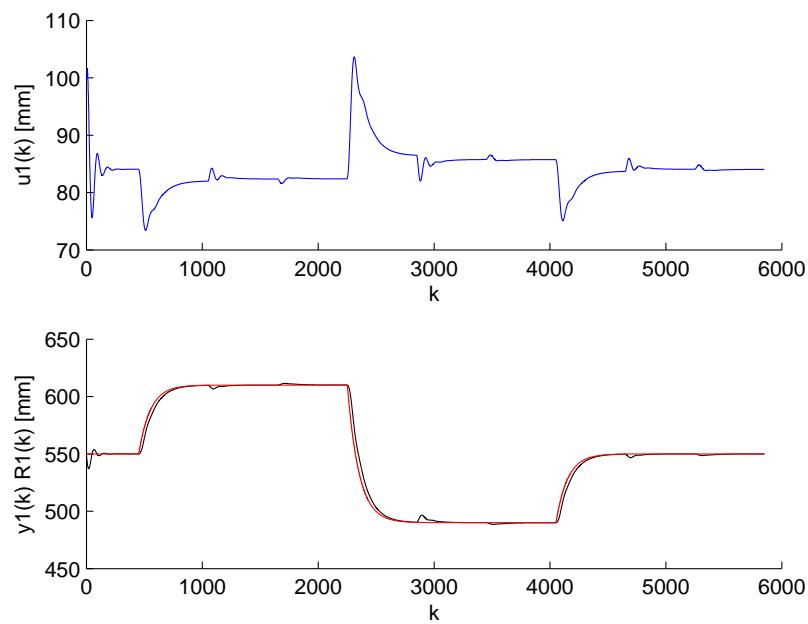


Figure 3.6: First Gate response, centralized solution, $\rho_1 = 1000$, $T=50$

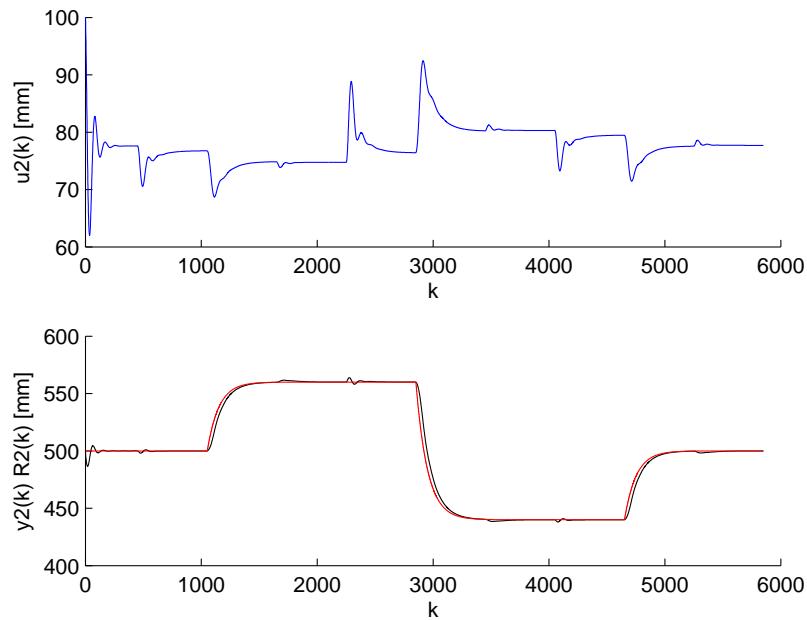


Figure 3.7: Second Gate response, centralized solution, $\rho_2 = 1000$, T=50

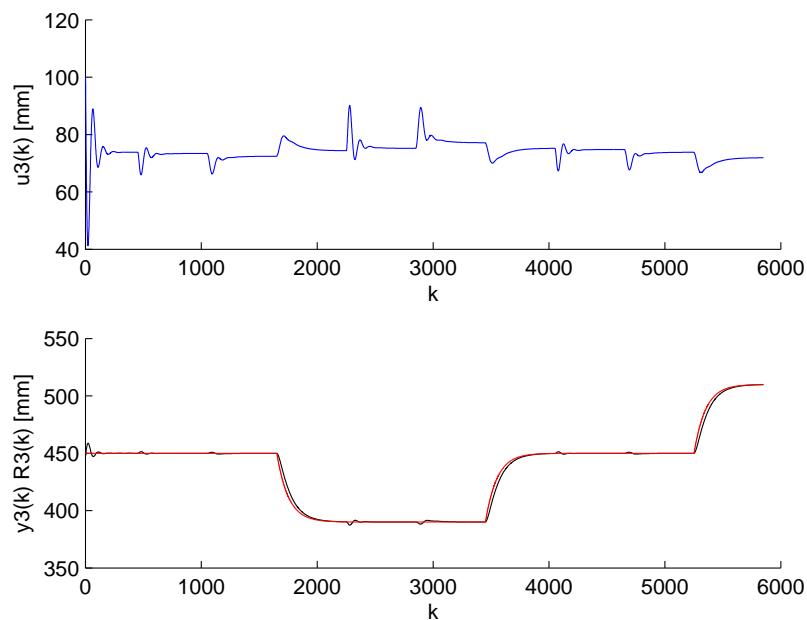


Figure 3.8: Third Gate response, centralized solution, $\rho_3 = 1000$, T=50

Chapter 4

Tuning the centralized algorithm

Contents

4.1 Time Horizon	32
4.2 Control Cost Weights	33

This chapter addresses the tuning of the parameters that configure the centralized algorithm. This parameters are the time horizon and the control cost weights.

To conduct this study the error in each pool will be defined as

$$e_i = \frac{\sum_{k=1}^T (y_i(k) - R_i(k))^2}{h}. \quad (4.1)$$

The total error will be

$$e_t = \frac{\sum_{k=1}^T (y_1(k) - R_1(k))^2 + \sum_{k=1}^T (y_2(k) - R_2(k))^2 + \sum_{k=1}^T (y_3(k) - R_3(k))^2}{h}. \quad (4.2)$$

4.1 Time Horizon

As for the time horizon it was firstly established a range of values for which the solution is stable. Under $T = 15$ and over $T = 75$ stability is not achieved with the centralized algorithm. Below is the graphic representing the error evolution for the values that give a stable result.

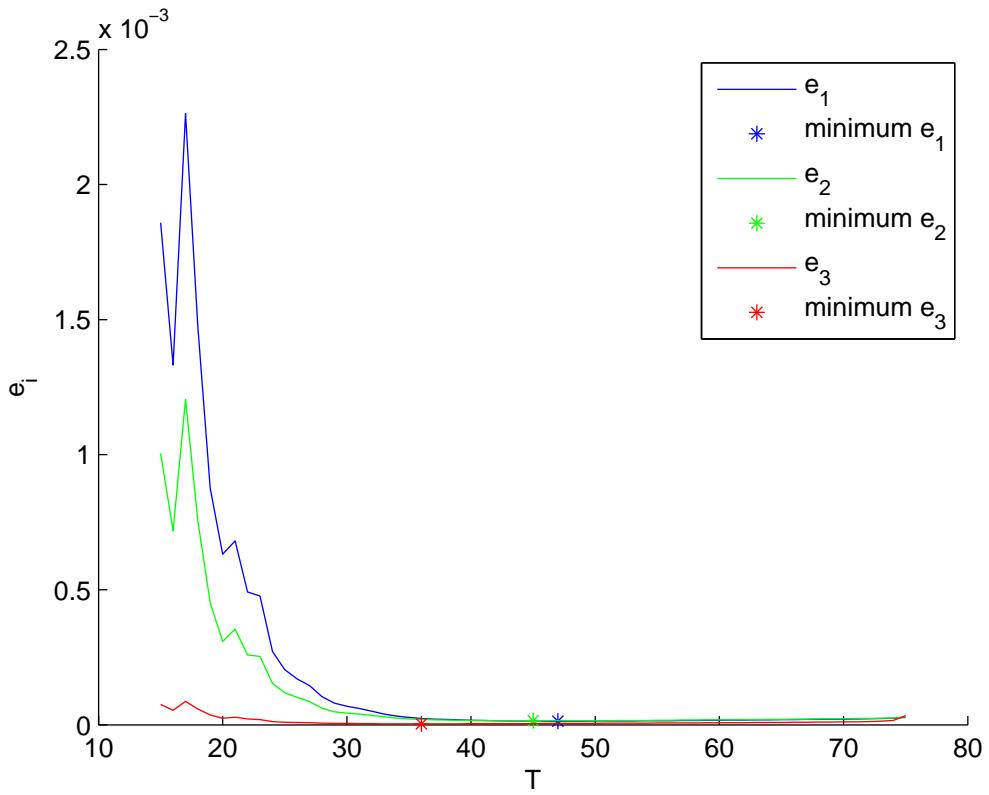


Figure 4.1: Error evolution with T varying from 15 to 75, centralized solution, $\rho_1 = 500$, $\rho_2 = 500$, $\rho_3 = 500$

As shown on the graphic, the error is the lowest for a time horizon of 47, 45 and 36, in respect of pools one, two and three. A graphic representing the total error is shown below, and the time horizon for which that sum is the lowest value is highlighted.

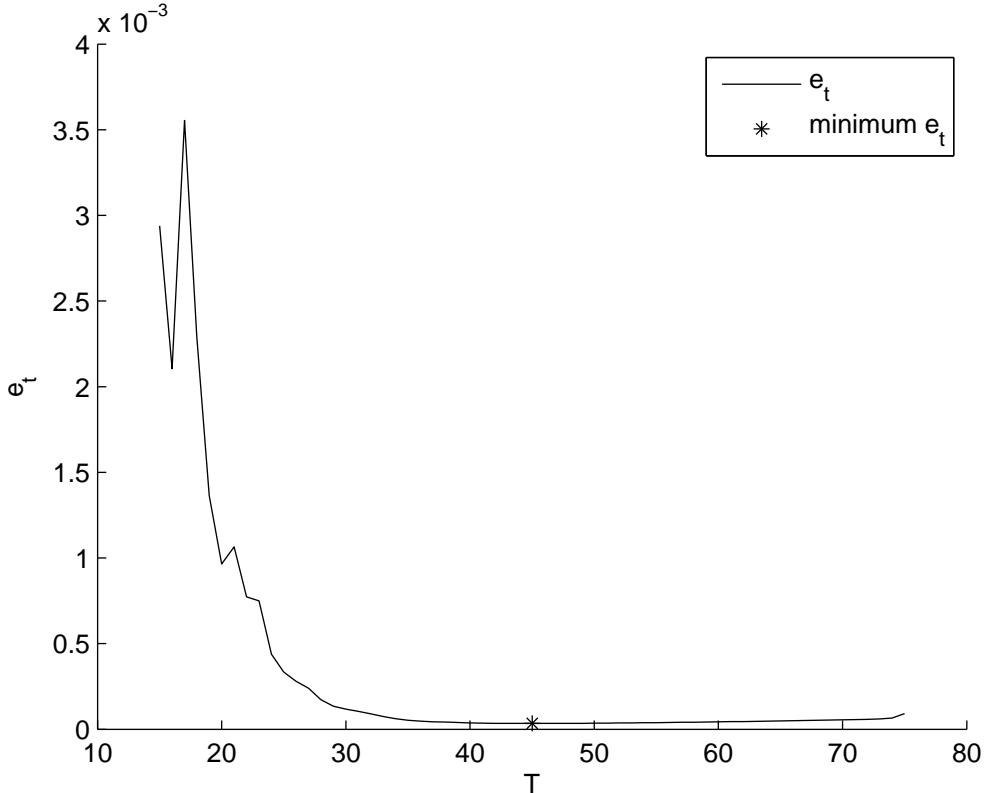


Figure 4.2: Error sum evolution with T varying from 15 to 75, centralized solution, $\rho_1 = 500$, $\rho_2 = 500$, $\rho_3 = 500$

It can be seen in figure 4.2 that a range from 40 to 70 is acceptable for the time horizon. If T is chosen under or over those values the algorithm cannot provide a stable solution (T between 70 and 75 is stable but presents high frequency oscillations that are impossible, making it impossible to apply in). The best value is 45 therefore, in the subsequent simulations, a time horizon of 45 will be used.

4.2 Control Cost Weights

To study the control costs simulations were run to determine the error sum for each set of control costs between 200 and 1000, with jumps of 100. The results are shown with more detail in Appendix C. In figure 4.3 is shown a graphic of the total error evolution in order to ρ_1 and ρ_2 for $\rho_3 = 200$.

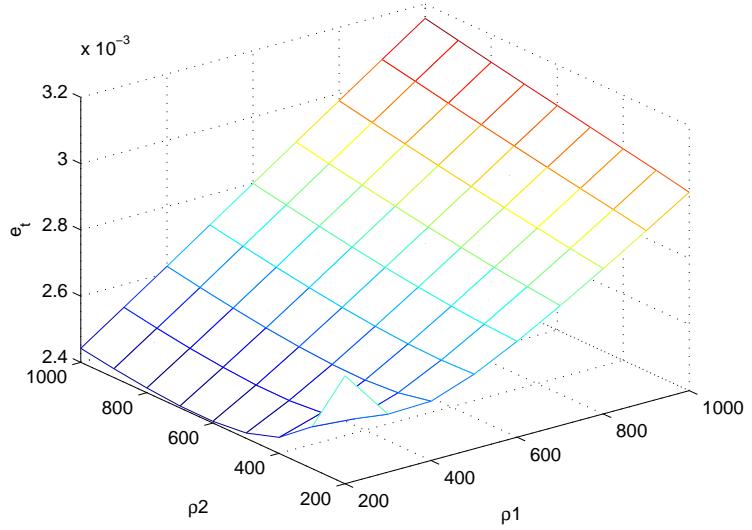


Figure 4.3: Error sum, $\rho_3 = 200$, $T=45$

Chapter 5

Distributed Control

Contents

5.1	D-ADMM	36
5.2	Node associated Dual variables	38
5.3	Edge associated Dual variables	38
5.4	D-ADMM Parameter Tuning	42

In the previous sections a centralized solution for the canal control was developed. To put that approach in practice it will be required that the level measures in each pool are communicated to a central node, the control variables values computed, and then sent to the actuators.

Due to the physical dimensions of such systems this may not be feasible. Alternatively, one can have a local controller for each gate. These local controllers will have access to the values of its own control variable, its neighbours control variables, the upstream and downstream water levels, but no knowledge about the water level in the other pools and the control variables values of controllers other than its neighbours.

This leads to another problem. As stated earlier, the opening or closing of a gate has effect in nearby pools. With the centralized solution there's a full knowledge of the water levels and the control variables. Now each controller computes the future local control variables. There will be local cost functions

$$J_i = \left\| W_i * U_i + W_{i-1,i} * U_{i-1} + W_{i,i+1} * U_{i+1} - R_i \right\|^2 + \rho * \|U_i\|^2. \quad (5.1)$$

Each controller will try to minimize its local cost function in order to its control variable. Distributed control is a hot topic and several methods have been developed [23], [33], [12], [9]. The purpose of such methods is to compute local control variables that will minimize the sum of the local cost functions.

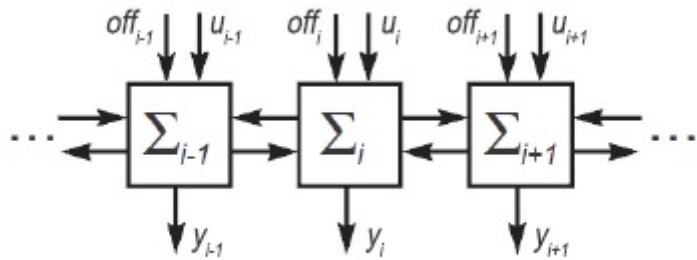


Figure 5.1: Example of communication between controllers

5.1 D-ADMM

A distributed version of the Alternating Direction Method of Multipliers (ADMM) will be used. This algorithm has not been proved to be more efficient than previous ones [21], but empirical data has shown it takes less communication steps to converge to a pre-specified error percentage.

In this problem the network topology is

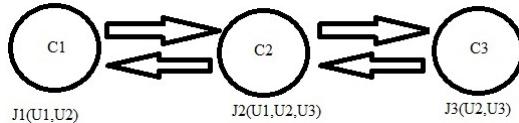


Figure 5.2: Network topology

Each node has a local cost function in order to a subset of U . Each cost function will be minimized subjected to its own set of constraints. Three conditions must be analyzed:

1. Each $J_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex over \mathbb{R}^n , and each set of constraints is closed and convex.
2. The problem is solvable.

3. The network is connected and its topology does not vary with time.

If these three conditions are observed D-ADMM can be used. The analysis to these conditions is in Appendix D.

For each node the cost functions are

$$J_1 = \left\| W_1 U_1 + W_{1,2} U_2 + \Pi_1 s_1 - R_1 \right\|^2 + \rho_1 \|U_1\|^2, \quad (5.2)$$

$$J_2 = \left\| W_2 U_2 + W_{2,1} U_1 + W_{2,3} U_3 + \Pi_2 s_2 - R_2 \right\|^2 + \rho_2 \|U_2\|^2, \quad (5.3)$$

$$J_3 = \left\| W_3 U_3 + W_{3,2} U_2 + \Pi_3 s_3 - R_3 \right\|^2 + \rho_3 \|U_3\|^2. \quad (5.4)$$

This cost functions are not $\mathbb{R}^n \rightarrow \mathbb{R}$. To put the cost functions in that form the control variables will be changed. From now on, each agent involved in the minimization will have an intern replica of all control variables. The notation U_a^b stands for the replica of U_a in agent b. The new control variables are:

$$\bar{U}_1 = \begin{bmatrix} U_1^{1T} & U_2^{1T} & U_3^{1T} \end{bmatrix}^T, \quad (5.5)$$

$$\bar{U}_2 = \begin{bmatrix} U_1^{2T} & U_2^{2T} & U_3^{2T} \end{bmatrix}^T, \quad (5.6)$$

$$\bar{U}_3 = \begin{bmatrix} U_1^{3T} & U_2^{3T} & U_3^{3T} \end{bmatrix}^T. \quad (5.7)$$

The notation of the cost functions will be different:

$$J_1 = (\bar{W}_1 \bar{U}_1 + Y_{10})^T (\bar{W}_1 \bar{U}_1 + Y_{10}) + \bar{U}_1^T \bar{\rho}_1 \bar{U}_1, \quad (5.8)$$

$$J_2 = (\bar{W}_2 \bar{U}_2 + Y_{20})^T (\bar{W}_2 \bar{U}_2 + Y_{20}) + \bar{U}_2^T \bar{\rho}_2 \bar{U}_2, \quad (5.9)$$

$$J_3 = (\bar{W}_3 \bar{U}_3 + Y_{30})^T (\bar{W}_3 \bar{U}_3 + Y_{30}) + \bar{U}_3^T \bar{\rho}_3 \bar{U}_3, \quad (5.10)$$

with

$$\bar{W}_1 = \begin{bmatrix} W_1 & W_{1,2} & 0 \end{bmatrix}, \quad (5.11)$$

$$\bar{W}_2 = \begin{bmatrix} W_{2,1} & W_2 & W_{2,3} \end{bmatrix}, \quad (5.12)$$

$$\bar{W}_3 = \begin{bmatrix} 0 & W_{3,2} & W_3 \end{bmatrix}, \quad (5.13)$$

$$Y_{10} = \Pi_1 s_1 - R_1, \quad (5.14)$$

$$Y_{20} = \Pi_2 s_2 - R_2, \quad (5.15)$$

$$Y_{30} = \Pi_3 s_3 - R_3, \quad (5.16)$$

and

$$\bar{\rho}_1 = \rho_1 \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (5.17)$$

$$\bar{\rho}_2 = \rho_2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (5.18)$$

$$\bar{\rho}_3 = \rho_3 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix}. \quad (5.19)$$

5.2 Node associated Dual variables

In [21] the D-ADMM algorithm is implemented with the dual variables associated to the nodes.

Algorithm 1 D-ADMM with node associated dual variables

Initialization : $\forall p \in V$, set $\gamma_p^1 = x_p^1 = 0$ and $k = 1$

repeat

for all $p \in C_1$ **do do**

 Set $v_p^k = \gamma_p^k - \sum_{j \in N_p} x_j^k$ and find

$x_p^{k+1} = \operatorname{argmin}_{x_p} f_p(x_p) + v_p^k x_p + \frac{D_p \rho_{ad}}{2} \|x_p\|^2$

 Send x_p^{k+1} to N_p

end for

 Repeat for $p \in C_2$, replacing x_j^k by x_j^{k+1} in v_p^k

for all $p \in V$ **do do**

$\gamma_p^{k+1} = \gamma_p^k + \rho_{ad} \sum_{j \in N_p} (x_p^{k+1} - x_j^{k+1})$

end for

$k \leftarrow k + 1$

until some stopping criteria is met

The cost functions will be

$$J_{1_{admm}} = J_1 + (\gamma_1 - \bar{U}_2)^T \bar{U}_1 + \frac{1}{2} \rho_{ad} \|\bar{U}_1\|^2, \quad (5.20)$$

$$J_{2_{admm}} = J_2 + (\gamma_2 - (\bar{U}_1 + \bar{U}_3))^T \bar{U}_2 + \rho_{ad} \|\bar{U}_2\|^2, \quad (5.21)$$

$$J_{3_{admm}} = J_3 + (\gamma_3 - \bar{U}_2)^T \bar{U}_3 + \frac{1}{2} \rho_{ad} \|\bar{U}_3\|^2. \quad (5.22)$$

The use of the MatLab function fmincon is not advised in this context due to large computational times. An analytical solution is much more practical.

5.3 Edge associated Dual variables

Although in [21] the algorithm 1 is presented, in this thesis the D-ADMM will be implemented with the dual variables associated to the edges. Therefore, two dual variables (λ and η) and a cost variable ρ_{ad} will be created.

Reflecting the use of the D-ADMM a change in the cost functions will come as

$$J_{1_{admm}} = J_1 - \lambda^T \bar{U}_1 + \frac{1}{2} \rho_{ad} \|\bar{U}_1 - \bar{U}_2\|^2, \quad (5.23)$$

$$J_{2_{admm}} = J_2 + (\lambda - \eta)^T \bar{U}_2 + \rho_{ad} (\|\bar{U}_2 - \bar{U}_1\|^2 + \|\bar{U}_2 - \bar{U}_3\|^2), \quad (5.24)$$

$$J_{3_{admm}} = J_3 + \eta^T \bar{U}_3 + \frac{1}{2} \rho_{ad} \|\bar{U}_3 - \bar{U}_2\|^2. \quad (5.25)$$

To get cost functions in the form $\mathbb{R}^n \rightarrow \mathbb{R}$ they will come as

$$J_{1_{admm}} = \bar{U}_1 (\bar{W}_1^T \bar{W}_1 + \bar{\rho}_1 + \frac{\rho_{ad}}{2} I) \bar{U}_1^T + (2 \bar{W}_1^T Y_{10} - \lambda - \rho_{ad} \bar{U}_2) \bar{U}_1^T + \epsilon_1, \quad (5.26)$$

$$J_{2_{admm}} = \bar{U}_2 (\bar{W}_2^T \bar{W}_2 + \bar{\rho}_2 + \rho_{ad} I) \bar{U}_2^T + (2 \bar{W}_2^T Y_{20} + (\lambda - \eta) - \rho_{ad} (\bar{U}_1 + \bar{U}_3)) \bar{U}_2^T + \epsilon_2, \quad (5.27)$$

$$J_{3_{admm}} = \bar{U}_3 (\bar{W}_3^T \bar{W}_3 + \bar{\rho}_3 + \frac{\rho_{ad}}{2} I) \bar{U}_3^T + (2 \bar{W}_3^T Y_{30} + \eta - \rho_{ad} \bar{U}_2) \bar{U}_3^T + \epsilon_3. \quad (5.28)$$

The terms in each cost function that do not depend on the minimizing variable are represented by ϵ_i , and not shown here, as the differentiation of the cost functions will make them equal to 0. For simplification purposes

$$J_{1_{admm}} = \bar{U}_1 M_1 \bar{U}_1^T + \Phi_1 \bar{U}_1^T, \quad (5.29)$$

$$J_{2_{admm}} = \bar{U}_2 M_2 \bar{U}_2^T + \Phi_2 \bar{U}_2^T, \quad (5.30)$$

$$J_{3_{admm}} = \bar{U}_3 M_3 \bar{U}_3^T + \Phi_3 \bar{U}_3^T. \quad (5.31)$$

This will lead to

$$\frac{\partial J_{1_{ad}}}{\partial \bar{U}_1^T} = 2 \bar{U}_1 M_1 + \Phi_1, \quad (5.32)$$

$$\frac{\partial J_{2_{ad}}}{\partial \bar{U}_2^T} = 2 \bar{U}_2 M_2 + \Phi_2, \quad (5.33)$$

$$\frac{\partial J_{3_{ad}}}{\partial \bar{U}_3^T} = 2 \bar{U}_3 M_3 + \Phi_3, \quad (5.34)$$

and the minimum for the control variables is

$$\bar{U}_{1_{minimum}} = -\frac{1}{2} M_1^{-1} \Phi_1, \quad (5.35)$$

$$\bar{U}_{2_{minimum}} = -\frac{1}{2} M_2^{-1} \Phi_2, \quad (5.36)$$

$$\bar{U}_{3_{minimum}} = -\frac{1}{2} M_3^{-1} \Phi_3. \quad (5.37)$$

Thus the algorithm for computing the solution using D-ADMM will be as follows.

Algorithm 2 Distributed MPC with dual variables associated to the edges

DualVariablesInitialization : $\lambda = 0; \eta = 0$

ControlVariablesInitialization : $\bar{U}_1 = 0; \bar{U}_2 = 0; \bar{U}_3 = 0$

repeat

$$\Phi_1 = 2\bar{W}_1^T Y_{10} - \lambda - \rho_{ad}\bar{U}_2$$

$$\bar{U}_1 = -\frac{1}{2}M_1^{-1}\Phi_1$$

$$\Phi_3 = 2\bar{W}_3^T Y_{30} + \eta - \rho_{ad}\bar{U}_2$$

$$\bar{U}_3 = -\frac{1}{2}M_3^{-1}\Phi_3$$

$$\Phi_2 = 2\bar{W}_2^T Y_{20} + (\lambda - \eta) - \rho_{ad}(\bar{U}_1 + \bar{U}_3)$$

$$\bar{U}_2 = -\frac{1}{2}M_2^{-1}\Phi_2$$

$$\lambda = \lambda - \rho_{ad}(\bar{U}_1 - \bar{U}_2)$$

$$\eta = \eta - \rho_{ad}(\bar{U}_2 - \bar{U}_3)$$

until pre defined maximum D-ADMM iterations or stopping criterion is met

Applying algorithm 2, using $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 200$, $\rho_2 = 600$, $\rho_3 = 200$, T=45 the following simulation results were obtained.

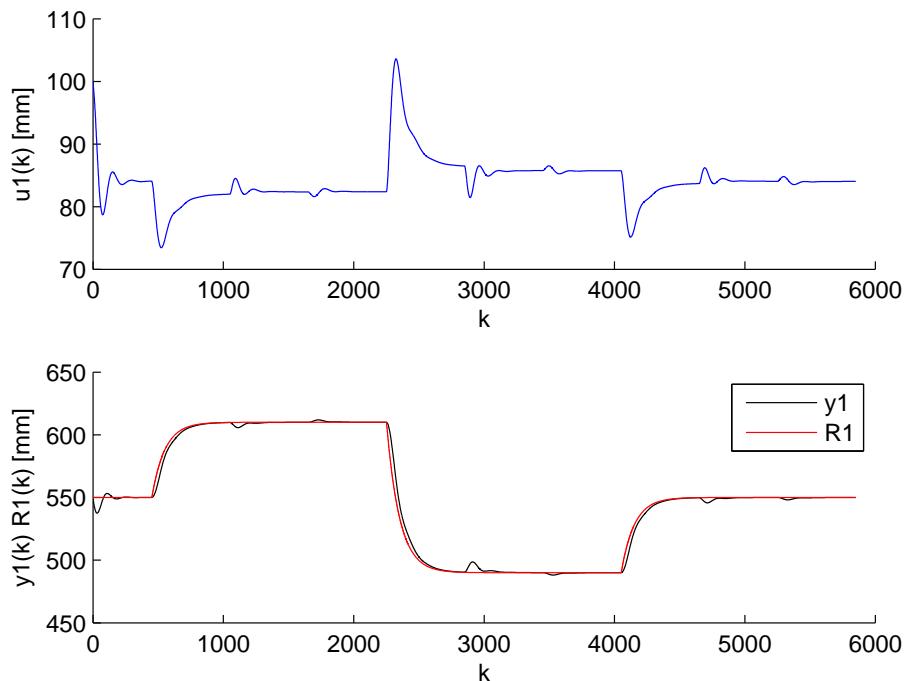


Figure 5.3: First Gate response, D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 200$, $\rho_2 = 600$, $\rho_3 = 200$, T=45

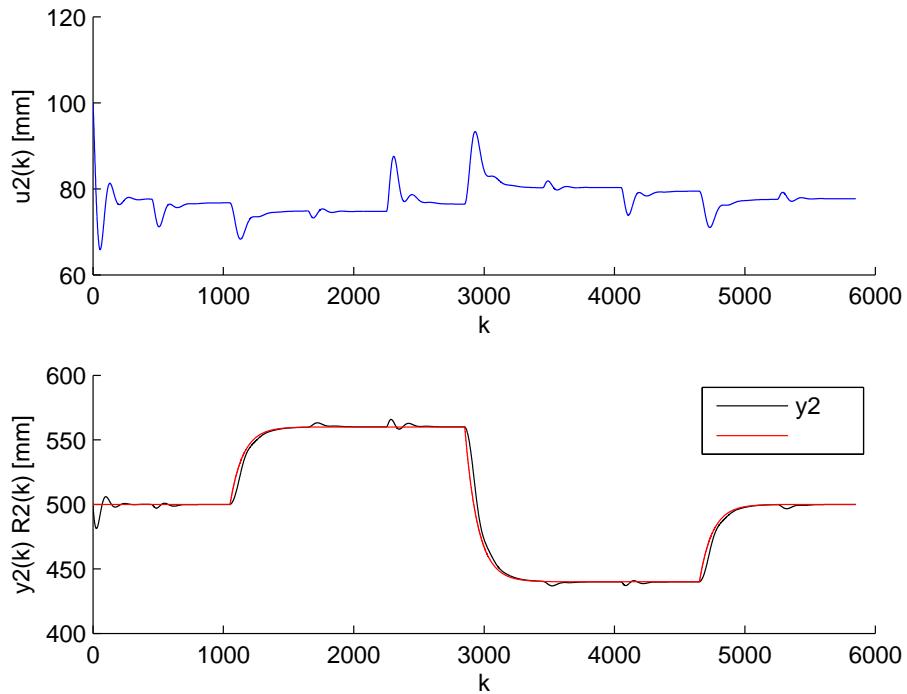


Figure 5.4: Second Gate response, D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 200$, $\rho_2 = 600$, $\rho_3 = 200$, $T=45$

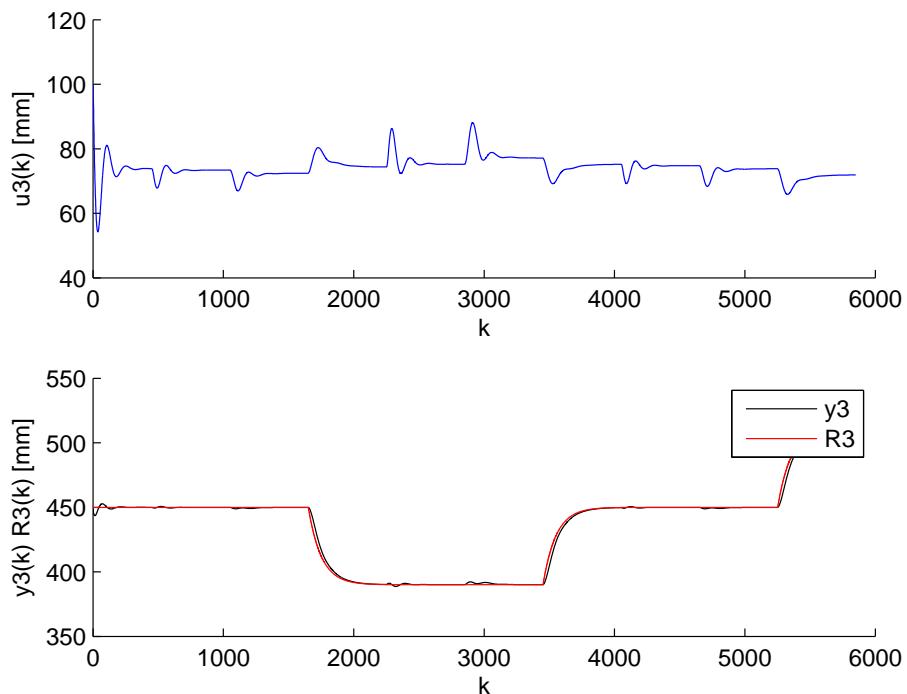


Figure 5.5: Third Gate response, D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 200$, $\rho_2 = 600$, $\rho_3 = 200$, $T=45$

The constraints are not taken into account because dealing with them would make the use of this analytical approach impossible.

The chosen way to deal with them was to force the output to be between acceptable bound values (if it is over a upper bound or below a lower bound, force it to be equal to the bound) and the variation towards the last input not to exceed pre specified values. This will make the problem solution not optimal.

5.4 D-ADMM Parameter Tuning

There are two parameters to take into account when analysing the error evolution in this problem: the number of iterations (n_{it}) and the D-ADMM cost (ρ_{ad}). Tests were conducted to determine the total error (as defined in 4.2) for n_{it} varying from 10 to 100, and ρ_{ad} varying from 20 to 100.

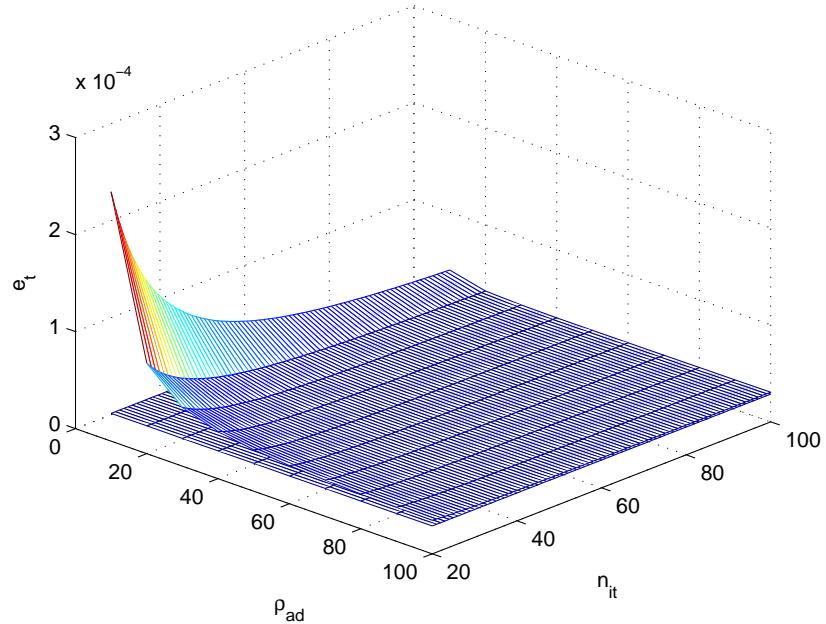


Figure 5.6: Error Evolution of D-ADMM solution, $\rho_1 = 200$, $\rho_2 = 600$, $\rho_3 = 200$, $T=45$

Chapter 6

Simulation Results

For validation purposes simulation results are compared with the experimental results in [10] and [17]. The proposed solution has a faster response than the comparable experimental results, but it presents a higher overshoot. In these simulations the control costs are higher than those considered better in 4.2 because those would make the response too fast.

Figures 6.1 to 6.3 compare the algorithm developed in this work with the one developed in [10]. The initial water levels are 603, 532, 524 mm in pools 1, 2 and 3 respectively. After three minutes the reference in pool 1 changes gradually until it reaches 653mm at minute 6. The effect of this reference change is not interesting to the analysis of the effects controller 1 actions will have in pool 2 because at this point the other pools have not yet attenuated the initial oscillations.

At minute the 30 reference level in pool 2 starts to rise from 532mm until it reaches 582mm at minute 36. Slight oscillations in the other two pools are observed, both of them beginning by a decrease in the level following by an increase with a small overshoot. The oscillations die out quickly (less than two minutes). In order to increase the water level in pool 2, gate 2 closes about 10mm. This will lead to a smaller flow through gate 2, causing water level in pool 3 to decrease. Controller reacts by closing gate 3 about 8mm, retaining water in pool 3 and rising the level back to the reference. In pool 1, contrary to what was expected, the water level also decreases. The models used in this simulations are not the non linear models used in [10] therefore they may not model the phenomena that occur in this experiment.

At minute 60 the reference level in pool 3 rises from 524mm to 572mm. This takes about 6 minutes. The controller closes the gate about 7mm, rising the water level in pool 3. This increases the flow from pool 2 to pool 3 and water level in pool 2 decreases. Controller 2 reacts, closing gate 2 and rising water level in pool 2. The effect in pool 1 is negligible. The reference level in pool 3 goes back to 524mm at minute 120. Once again the effect in pool 1 is negligible. The closing of gate 2 makes the water level in pool 2 to rise. controller 2 compensates this phenomena and the water level goes back to the reference value.

At minute 150 the reference level in pool 2 decreases from 582mm to 532mm. The oscillations in the other pools have a larger amplitude than when the reference level went up.

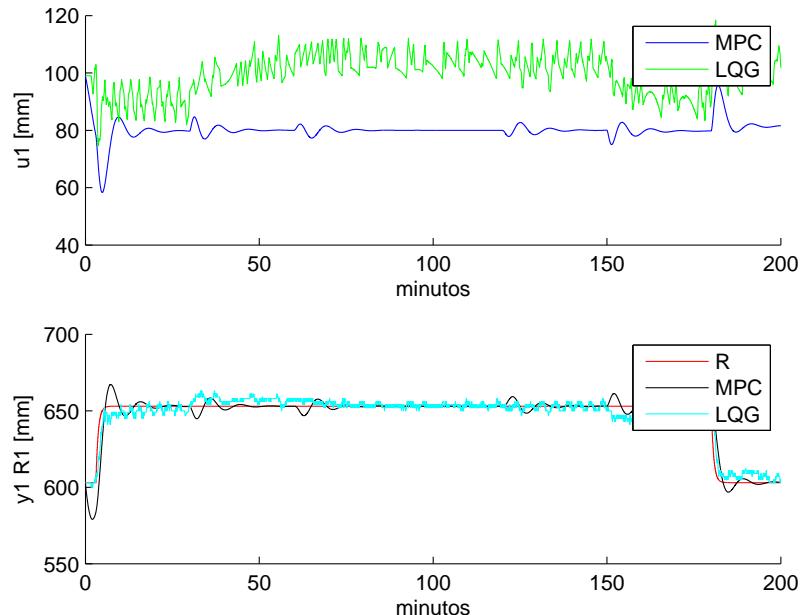


Figure 6.1: Comparison with [10], D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 1000$, $\rho_2 = 1000$, $\rho_3 = 1000$, T=45, Gate 1

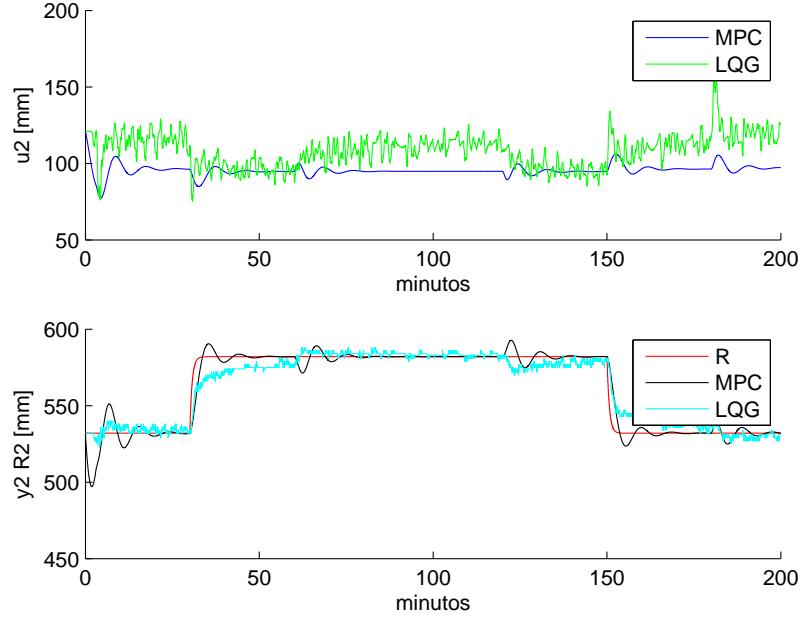


Figure 6.2: Comparison with [10], D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 1000$, $\rho_2 = 1000$, $\rho_3 = 1000$, T=45, Gate 2

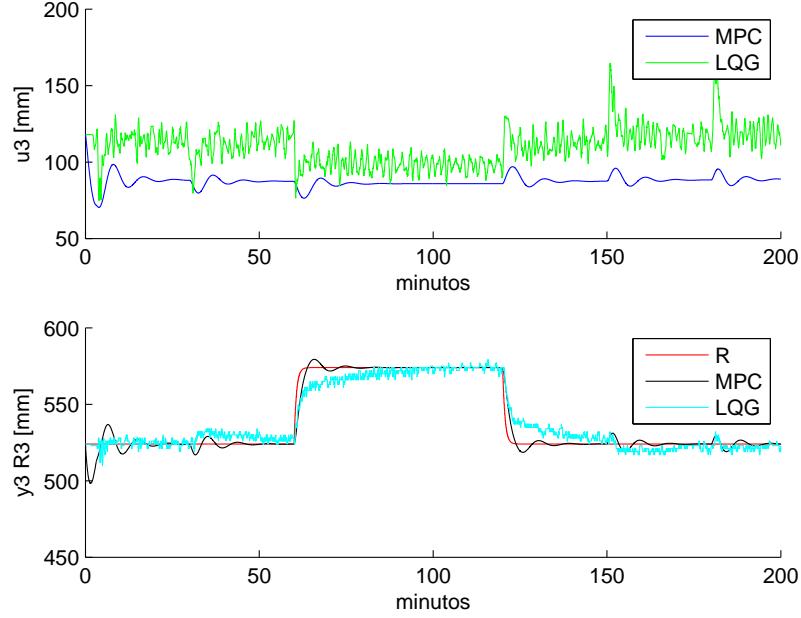


Figure 6.3: Comparison with [10], D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 1000$, $\rho_2 = 1000$, $\rho_3 = 1000$, T=45, Gate 3

Figures 6.4 to 6.6 compare the algorithm developed in this thesis with [17]. Water levels are set initially to 708, 622 and 550mm in pools 1,2 and 3 respectively.

The experience conducted in this field test is much similar to the previous one. The major differences, besides the initial water level values, are the minutes in which the reference level changes occur, and the reference filter (in [10] the reference level changes take about 6 minutes and in [17] it takes 2 minutes). The algorithm developed in this thesis as a behaviour that is more related to [17], which was expectable

since both algorithms use the same canal model.

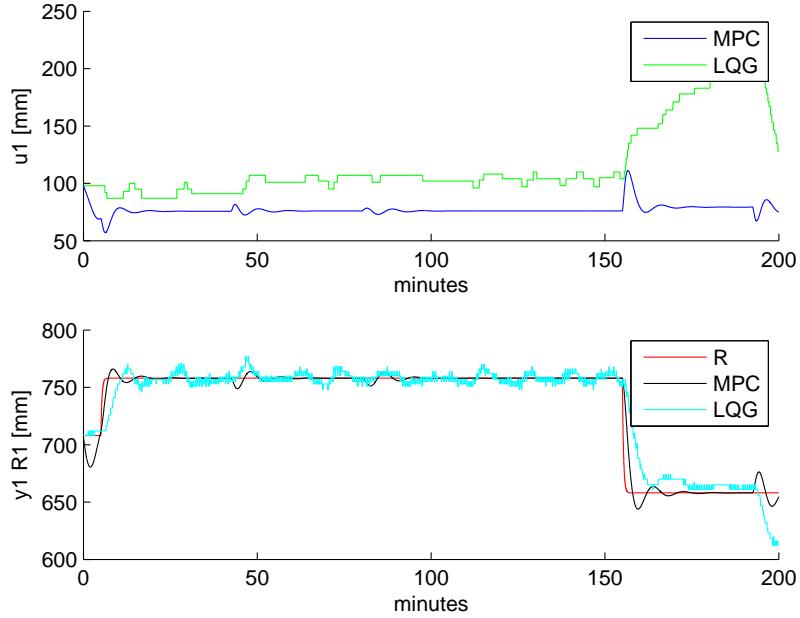


Figure 6.4: Comparison with [17], D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 1000$, $\rho_2 = 1000$, $\rho_3 = 1000$, T=45, Gate 1

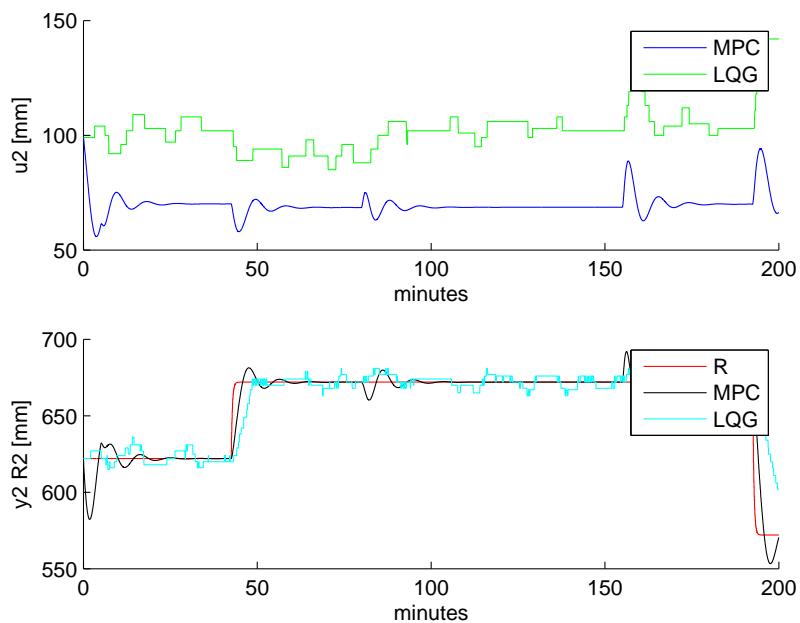


Figure 6.5: Comparison with [17], D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 1000$, $\rho_2 = 1000$, $\rho_3 = 1000$, T=45, Gate 2

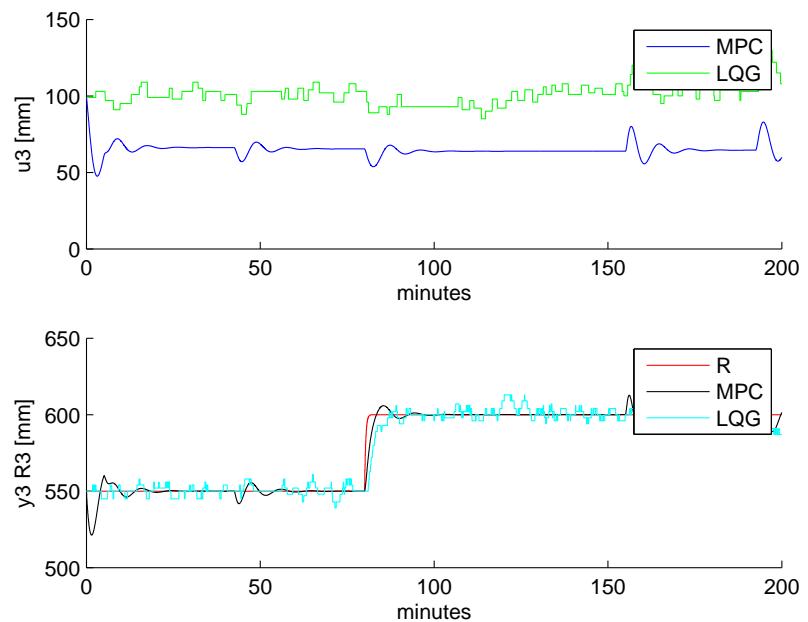


Figure 6.6: Comparison with [17], D-ADMM with $\rho_{ad} = 100$, $n_{it} = 20$ and $\rho_1 = 1000$, $\rho_2 = 1000$, $\rho_3 = 1000$, T=45, Gate 3

Chapter 7

Conclusions and Future Work

The purpose of this work is to develop a distributed predictive control algorithm based on Input-Output models and to apply it in a water distribution canal for agricultural use. The original contribution consists of obtaining a distributed MPC algorithm based on the *i/o* model, as most published works use the state-space approach.

Another innovation was the use of the D-ADMM algorithm in the distributed optimization. This was already implemented in [20] with state models, but in this work it was implemented with Input-Output models.

The resulting distributed MPC algorithm was successfully tested in simulation with a linear model of the canal. The influence of the controller parameters on the resulting performance was studied and the range of acceptable values for the time horizon 4.1, control cost weights 4.2 and D-ADMM 5.4 parameters was established.

The distributed version works with a wide range of D-ADMM cost and number of iterations as shown in figure 5.6. A smaller number of iterations is preferable because it leads to a lower computational time.

The developed algorithm was validated through simulations. In chapter 6 the algorithm performance is compared with the field test results obtained in [17] and [10].

A different way to insert the integral effect will have to be studied to make the best use of the canal sensors measurements. The real opening of each gate is one of the available measurements but, due to the position of the integrator in the system, it cannot be used. In figure 3.3 it can be seen the Real System, the Integrator and the Controller. The gate sensor reads the real value of U_m , but the controller is computing U_c .

If the non linearities are not active (control variable between saturation limits and variation between computed control variable and previous control variable smaller than the slewrate) the value of U_m can be used to compute a replica of U_c , and then use it in the next iteration of MPC as the real value of $u(k-1)$ instead of the one computed in the previous step.

However, this cannot be done when the non linearities occur because a replica of U_c cannot be computed from U_m . This is a downside because the information of the gate sensors is not being used and the controller uses an incorrect value of the control variable in the next iteration. The controller is using the value it computed, with no information that it was not the intended value the system received.

A modification in the integral effect, which could make the computed control value be inserted directly in the actuators, would make possible to use the sensor readings.

As far as the non linearities are concerned a way to incorporate them in the computations, without enlarging the computational time to unacceptable values, would be better than the solution proposed in 5.1.

Bibliography

- [1] Controlo adaptativo e preditivo de canais de rega. F. c. machado and n. m. nogueira. Master's thesis, Instituto Superior Técnico, 2006.
- [2] Alessandro Alessio, Davide Barcellia, and Alberto Bemporad. Decentralized model predictive control of dynamically coupled linear systems. *Journal of Process Control*, 11(5):705–714, 2011.
- [3] Carlos Sepúlveda Toepfer and José Rodellar Benedé. Irrigation canal system identification for control purposes. Technical report, Universitat Politècnica de Catalunya, September 2005.
- [4] J.M. Coron, B. d'Andra Novel, and G. Bastin. A Lyapunov approach to control irrigation canals modeled by Saint-Venant's equations. *ECC99*, 1999.
- [5] Zhiliang Ding, Changde Wang, Guangming Tan, and Guanghua Guan. The application of the fuzzy self-adaptive pid controller to the automatic operation control of water transfer canal system. *Second International Conference on Intelligent Computation Technology and Automation*, 2:822–825, 2009.
- [6] Filipe Alexandre Diogo Maia Cadete. Distributed predictive control of a water delivery canal. Master's thesis, Instituto Superior Técnico, 2011.
- [7] Didier Georges. Decentralized adaptive control for a water distribution system. In *American Control Conference (ACC)*, pages 3346 – 3351, Monreal, QC, June 2012.
- [8] M. Gugat and G. Leugering. Global boundary controllability of the saint-venant system for sloped canals with friction. *Annales de l'Institut Henri Poincare (C) Non Linear Analysis*, 26(1):257–270, 2009.
- [9] J.M. Igreja, J.M. Lemos, F.M. Cadete, L.M. Rato, and M. Rijo. Control of a water delivery canal with cooperative distributed mpc. In *American Control Conference (ACC)*, pages 3346 – 3351, Monreal, QC, June 2012.
- [10] Inês Gama Caldas Sampaio. Fault tolerant control of a water delivery canal. Master's thesis, Instituto Superior Técnico, 2012.
- [11] Tamás Keviczks, Francesco Borrelli, and Gary J. Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Journal Automatica (Journal of IFAC)*, 42(12):2105–2115, 2006.
- [12] Tamás Keviczks, Francesco Borrelli, and Gary J. Balas. A distributed receding horizon control algorithm for dynamically coupled nonlinear systems. *IEEE Transactions on Automatic Control*, 52(7):1249–1263, 2007.
- [13] X. Litrico and V. Fromion. Tuning of robust distant downstream pi controllers for an irrigation canal pool: (i) theory. *Journal of Irrigation and Drainage Engineering*, 132(4):359–368, 2006.

- [14] Xavier Litrico and V. Fromion. Advanced control politics and optimal performance for an irrigation canal. *European Control Conference*, 2003.
- [15] Xavier Litrico, Vincent Fromion, Jean-Pierre Baume, Carina Arranja, and Manuel Rijo. Experimental validation of a methodology to control irrigation canals based on saint-venant equations. *Control Engineering Practice*, 13:1425–1437, 2005.
- [16] Xavier Litrico, Didier Georges, and Jean-Luc Trouvat. Modelling and robust control of a dam-river system. *IEEE SMC98*, 1998.
- [17] Luís Filipe Marques Pinto. Distributed lqg control of a water delivery canal. Master’s thesis, Instituto Superior Técnico, 2011.
- [18] P. Malaterre and J.P.Baum. Modeling and regulation of irrigation canals: existing applications and ongoing researches. *IEEE International Conference*, 4:3850–3855, 1998.
- [19] Manfred Morari and Jay H. Lee. Model predictive control: past, present and future. *Computers and Chemical Engineering*, 23:667–682, 1999.
- [20] João F. C. Mota, João M. F. Xavier, Pedro M. Q. Aguiar, and Markus Püschel. Distributed optimization with local domains: Applications in mpc and num.
- [21] João F. C. Mota, João M. F. Xavier, Pedro M. Q. Aguiar, and Markus Püschel. D-ADMM: A communication-efficient distributed algorithm for separable optimization. *IEEE Transactions on Signal Processing*, 6, 2013.
- [22] Rudy R. Negenborn, Peter-Jules Van Overloop, Tamás Kevinczy, and Bart De Schutter. Distributed model predictive control of irrigation canals. *Networks and Heterogeneous Media*, 4(2):359–380, 2009.
- [23] Alex Olshevsky and John N. Tsitsiklis. Convergence speed in distributed consensus and averaging. *SIAM Journal on Control and Optimization*, 48(1):33–55, 2009.
- [24] Su Ki Ooi, M.P.M. Krutzen, and E.Weyer. On physical and data driven modelling of irrigation channels. *Control Engineering Practice*, 13:461–471, 2005.
- [25] R. Rivas Perez, V. Feliu Batlle, F. Castillo Garcia, and A. Linarez Saez. System identification for control of a main irrigation canal pool. In *Proceedings of the 17th World Congress*, pages 9649–9654. The International Federation of Automatic Control, 2008.
- [26] P.O.Shirley, J.M.Lemos, N.F.Nogueira, and F.J.Machado. Model analysis and control of a four segment irrigation canal. In *Proceedings of the 15th Mediterranean Conference on Control & automation*.
- [27] Igor A. Shiklomanov. Appraisal and assessment of world water resources. *Water International*, 25(1):11–32, 1990.
- [28] J. Shuurmans. *Control of Water Levels in Open-Channels*. PhD thesis, Delft University of Technology, 1997.
- [29] Pedro Silva, Miguel Ayala Botto, João Figueiredo, and Manuel Rijo. Model predictive control of an experimental water canal. In *Proceedings of the European Control Conference 2007*, pages 2977–2984, 2007.

- [30] Yuji Wakasa, Mizue Arakawa, Kanya Tanaka, and Takuya Akashi. Decentralized model predictive control via dual decomposition. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 381–386, 2008.
- [31] E. Weyer. LQ control of an irrigation channel. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 750–755, 2003.
- [32] Erik Weyer. System identification of an open water channel. *Control Engineering Practice*, 9:1289–1299, 2001.
- [33] Hao Zhu, Georgios B. Giannakis, and Alfonso Cano. Distributed in-network channel decoding. *IEEE Transactions on Signal Processing*, 57(10):3970–3983, 2009.

Appendix A

Input-Output Form Coefficients

A.1 SISO Coefficients

The SISO model is described in 3.1. In this work gate 2 was the one treated isolately so, if the noise is not taken into account ($C_2 = 0$):

$$y_2(k+1) = \begin{bmatrix} -1,944 & 0,944 \end{bmatrix} \begin{bmatrix} y_2(k) \\ y_2(k-1) \end{bmatrix} + \begin{bmatrix} -3,637 & 10,442 & -10,099 & 3,289 \end{bmatrix} \begin{bmatrix} u_2(k) \\ u_2(k-1) \\ u_2(k-2) \\ u_2(k-3) \end{bmatrix} \quad (\text{A.1})$$

A.2 MISO Coefficients

$$y_1(k+1) = A_1 \begin{bmatrix} y_1(k) \\ y_1(k-1) \\ y_1(k-2) \end{bmatrix} + B_1 \begin{bmatrix} u_1(k) \\ u_1(k-1) \\ u_1(k-2) \end{bmatrix} + P_{1,2} \begin{bmatrix} u_2(k) \\ u_2(k-1) \\ u_2(k-2) \end{bmatrix} \quad (\text{A.2})$$

$$y_2(k+1) = A_2 \begin{bmatrix} y_2(k) \\ y_2(k-1) \\ y_2(k-2) \end{bmatrix} + B_2 \begin{bmatrix} u_2(k) \\ u_2(k-1) \\ u_2(k-2) \end{bmatrix} + P_{2,1} \begin{bmatrix} u_1(k) \\ u_1(k-1) \\ u_1(k-2) \end{bmatrix} + P_{2,3} \begin{bmatrix} u_3(k) \\ u_3(k-1) \\ u_3(k-2) \end{bmatrix} \quad (\text{A.3})$$

$$y_3(k+1) = A_3 \begin{bmatrix} y_3(k) \\ y_3(k-1) \\ y_3(k-2) \end{bmatrix} + B_3 \begin{bmatrix} u_3(k) \\ u_3(k-1) \\ u_3(k-2) \end{bmatrix} + P_{3,2} \begin{bmatrix} u_2(k) \\ u_2(k-1) \\ u_2(k-2) \end{bmatrix} \quad (\text{A.4})$$

$$A_1 = \begin{bmatrix} 1,6776 & -1,3312 & 0.6522 \end{bmatrix} \quad (\text{A.5})$$

$$A_2 = \begin{bmatrix} -1,7698 & 1,4130 & -0,6416 \end{bmatrix} \quad (\text{A.6})$$

$$A_3 = \begin{bmatrix} -2,0412 & 1,6054 & -0,5633 \end{bmatrix} \quad (\text{A.7})$$

$$B_1 = \begin{bmatrix} -0,4138 & 0,3186 & 0,0580 \end{bmatrix} \quad (\text{A.8})$$

$$B_2 = \begin{bmatrix} -0,3336 & 0,4309 & -0,1449 \end{bmatrix} \quad (\text{A.9})$$

$$B_3 = \begin{bmatrix} -0,2136 & 0,1165 & 0,0665 \end{bmatrix} \quad (\text{A.10})$$

$$P_{1,2} = \begin{bmatrix} 0,0291 & 0,0350 & -0,0703 \end{bmatrix} \quad (\text{A.11})$$

$$P_{2,1} = \begin{bmatrix} 0,1494 & 0,4134 & -0,5428 \end{bmatrix} \quad (\text{A.12})$$

$$P_{2,3} = \begin{bmatrix} 0,1309 & -0,0924 & -0,0416 \end{bmatrix} \quad (\text{A.13})$$

$$P_{3,2} = \begin{bmatrix} 0,5075 & -0,2708 & -0,2211 \end{bmatrix} \quad (\text{A.14})$$

Appendix B

Poles and Zeros

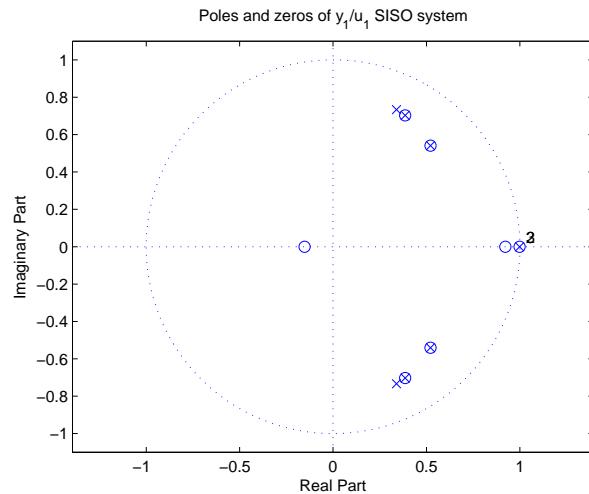


Figure B.1: Poles and zeros of the $\frac{y_1}{u_1}$ SISO system

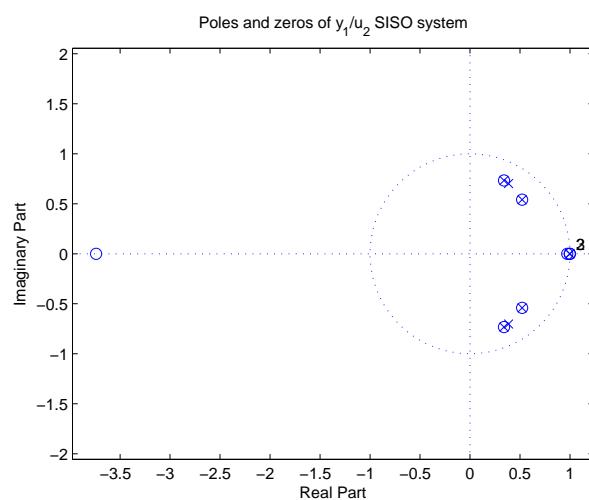


Figure B.2: Poles and zeros of the $\frac{y_1}{u_2}$ SISO system

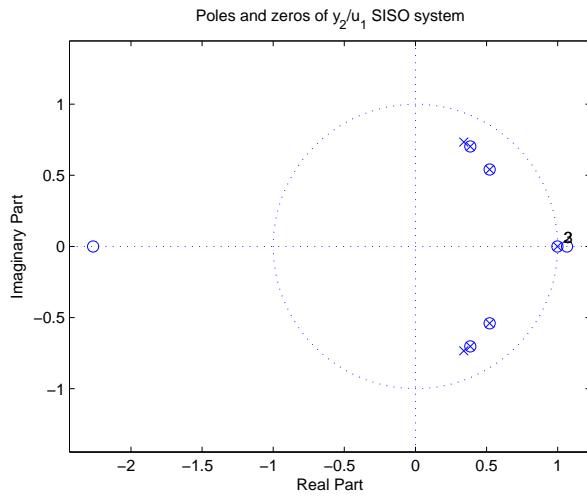


Figure B.3: Poles and zeros of the $\frac{y_2}{u_1}$ SISO system

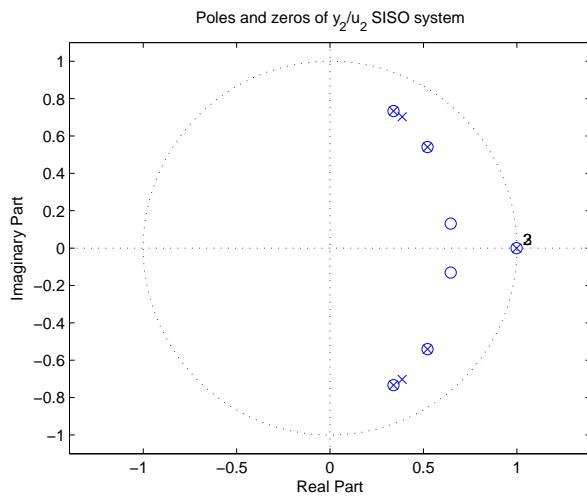


Figure B.4: Poles and zeros of the $\frac{y_2}{u_2}$ SISO system

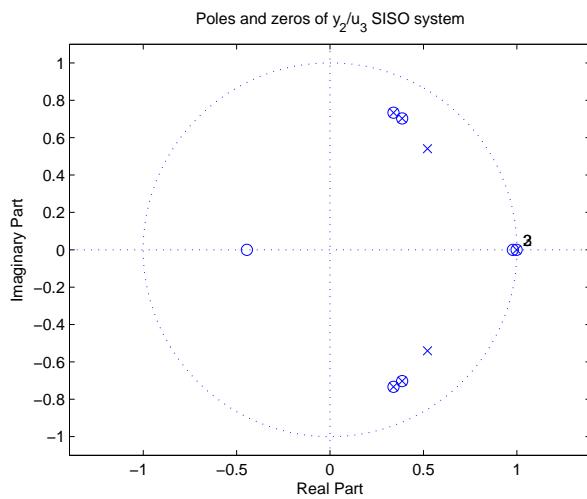


Figure B.5: Poles and zeros of the $\frac{y_2}{u_3}$ SISO system

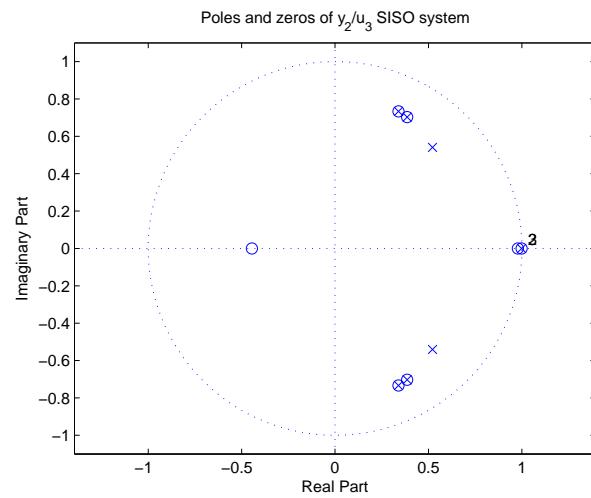


Figure B.6: Poles and zeros of the $\frac{y_3}{u_2}$ SISO system

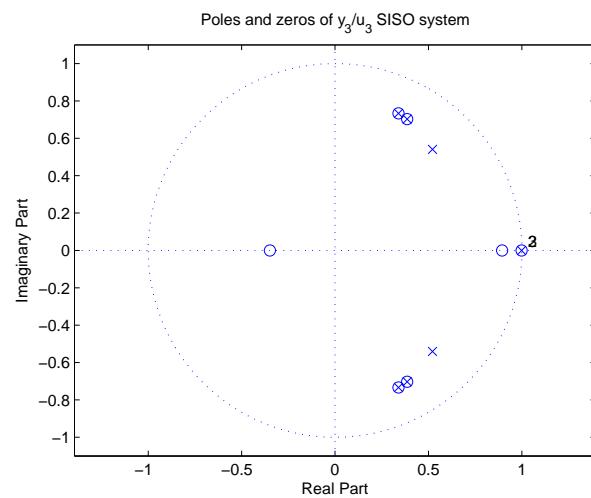


Figure B.7: Poles and zeros of the $\frac{y_3}{u_3}$ SISO system

Appendix C

Control costs

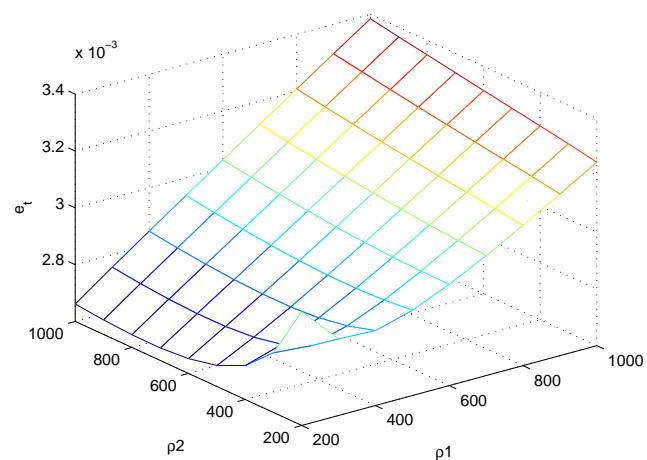


Figure C.1: Error sum, $\rho_3 = 300$, $T=45$

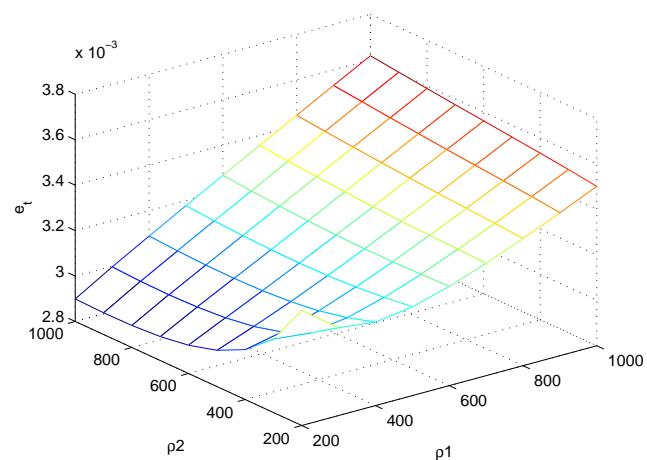


Figure C.2: Error sum, $\rho_3 = 400$, $T=45$

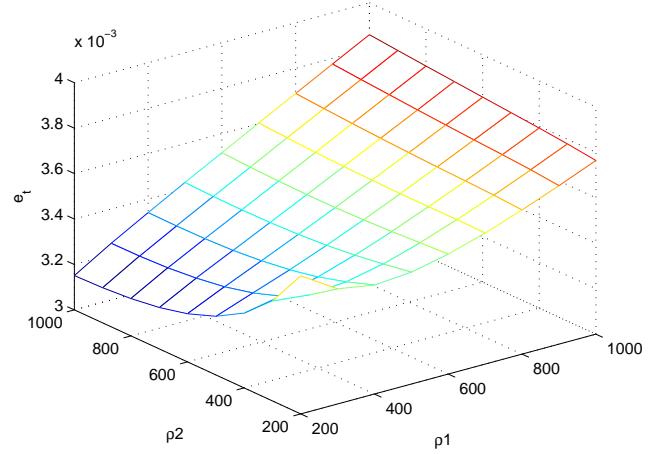


Figure C.3: Error sum, $\rho_3 = 500$, $T=45$

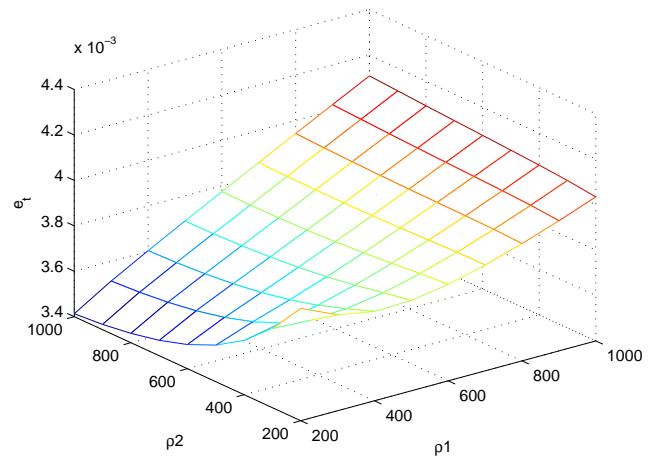


Figure C.4: Error sum, $\rho_3 = 600$, $T=45$

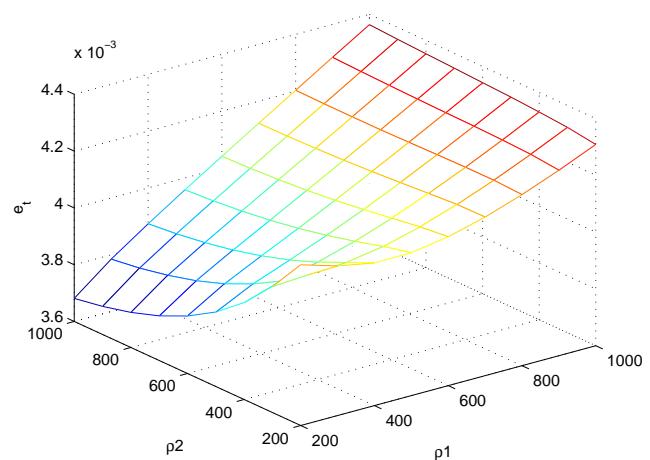


Figure C.5: Error sum, $\rho_3 = 700$, $T=45$

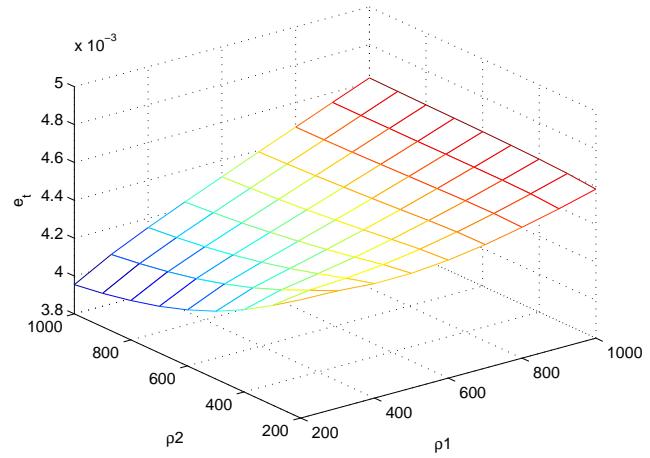


Figure C.6: Error sum, $\rho_3 = 800$, $T=45$

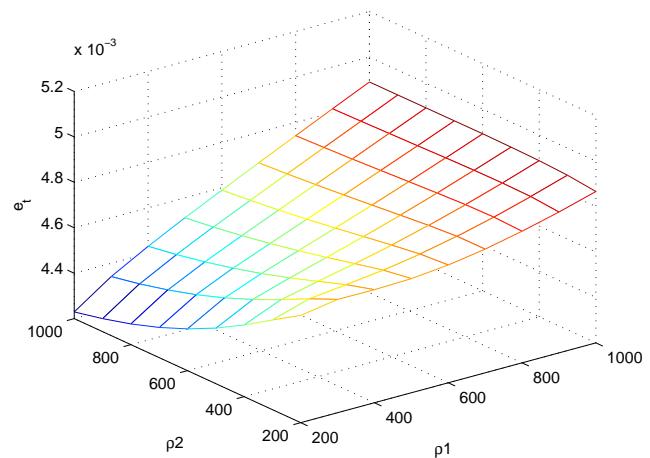


Figure C.7: Error sum, $\rho_3 = 900$, $T=45$

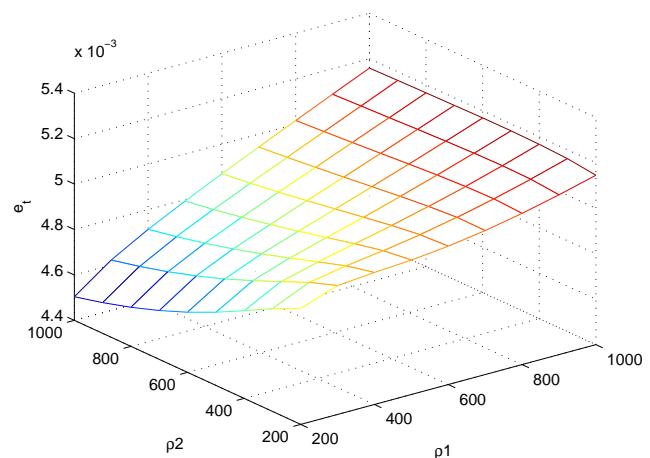


Figure C.8: Error sum, $\rho_3 = 1000$, $T=45$

Appendix D

D-ADMM conditions

D.1 Condition 1

To prove that each $J_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex over \mathbb{R}^n the second derivative (Hessian matrix) will be calculated. If all its eigenvalues are positive the system is convex. The first derivative is

$$\frac{\partial J_{1_{ad}}}{\partial \bar{U}_1^T} = 2\bar{U}_1 M_1 + \Phi_1 \quad (\text{D.1})$$

$$\frac{\partial J_{2_{ad}}}{\partial \bar{U}_2^T} = 2\bar{U}_2 M_2 + \Phi_2 \quad (\text{D.2})$$

$$\frac{\partial J_{3_{ad}}}{\partial \bar{U}_3^T} = 2\bar{U}_3 M_3 + \Phi_3 \quad (\text{D.3})$$

The second is

$$\frac{\partial J_{1_{ad}}^2}{\partial (\bar{U}_1^T)^2} = 2M_1 \quad (\text{D.4})$$

$$\frac{\partial J_{2_{ad}}^2}{\partial (\bar{U}_2^T)^2} = 2M_2 \quad (\text{D.5})$$

$$\frac{\partial J_{3_{ad}}^2}{\partial (\bar{U}_3^T)^2} = 2M_3 \quad (\text{D.6})$$

The eigenvalues are all positive for Hessian matrixes of all three cost functions and therefore they are convex.

D.2 Condition 2

A proof about the existence of solution will not be presented, but in section 3.3 a solution was obtained, so the problem has a solution.

D.3 Condition 3

As seen in figure 5.2 the network is connected. As long as there are no actuator or sensor faults, the topology will not change, and therefore this condition is satisfied.