Fault Tolerant Control of a Water Delivery Canal

Inês Gama Caldas Sampaio

Maio de 2012
Acknowledgments

I would first like to express my appreciation to Instituto Superior Técnico and all my professors for their contribution to my graduation and in particular my supervisor to whom I express my profound gratitude for all his advice, endless patience and understanding.

I would also like to thank my family, especially my parents, for always being supportive and patient, which, I admit, is not always easy.

A special thanks to João Ribas, for without his presence and support my everyday life would not be the same.

Finally, I would like to thank all my friends, without whom the last five years would have been harder and less joyful. In particular for the last six months, I owe a huge thank you to Tiago Simão, whose patience and dedication I most admire and without whose help and reassuring words developing this work would have been a tougher task.

This work has been performed in the framework of project AQUANET - Decentralized and reconfigurable control for water delivery multipurpose canal systems, funded by FCT - Portugal under contract PTDC/EEA-CRO/102102/2008.
Abstract

This dissertation proposes solutions to the problems of designing controllers for a water delivery canal that are tolerant to actuator and sensor faults. It is considered that an actuator fault has occurred when a gate suddenly blocks, regardless of the command received by the controller. Sensor faults are assumed to occur when a sensor reads a value different from the true water level value.

The controllers considered, which are both centralized and distributed for each fault case, are used to control the water level downstream of each pool of the canal, of which we only take into account the first three pools. The centralized controllers are multivariable Linear-Quadratic-Gaussian (LQG) that use a linear model of the canal. When no actuator fault has been detected, that controller has as inputs the three gate positions and as outputs the water levels of the three pools. After an actuator fault is detected, only the water levels of pools 1 and 3 are controlled and the positions of gates 1 and 3 are the only ones used by the controller. Distributed control uses as many single-gate LQG controllers as the number of pools considered and they negotiate with their neighbors in order to guarantee coordination, reaching a Nash equilibrium. Actuator faults are detected by comparison of the measured gate position with the respective command previously sent.

To deal with sensor faults, we reconstruct the lost signal from the value read by the central sensor of the same pool. Sensor faults are detected based on the difference between the value read by the sensor and the reconstructed signal.

Keywords

System identification, Linear Quadratic Gaussian control, actuator and sensor faults, centralized control, distributed control
Resumo

A presente dissertação propõe soluções para o problema do projecto de controladores tolerantes a falhas nos actuatores e nos sensores, aplicado a um canal de distribuição de água. Considera-se falha num actuador o bloqueio de uma comporta, independentemente do comando recebido. Assume-se que ocorreu uma falha num sensor quando o sensor lê um valor que não é o verdadeiro valor do nível da água.

Para cada tipo de falha, consideram-se controladores centralizados e distribuídos, que controlam o nível da água a jusante de cada trecho do canal, do qual apenas se consideram os três primeiros. Os controladores centralizados são LQG multivariáveis que usam um modelo linear do canal. Quando não ocorre falha, este controlador terá como entradas as posições das três portas e, como saídas, os níveis da água nos três trechos. Quando é detectada uma falha num actuador, o controlador usado tem apenas as posições das comportas 1 e 3 como entradas e os respectivos níveis da água como saída, sendo apenas estes dois controlados. O controlo distribuído requer tantos controladores LQG como o número de trechos considerado, os quais negoceiam com os seus vizinhos, de modo a garantir coordenação. As falhas nos actuadores são detectadas por comparação da posição da comporta medida com o comando previamente enviado.

Quando um sensor falha, é necessário reconstruir o sinal perdido, o que se consegue a partir de um outro sensor situado ao centro do mesmo trecho. As falhas nos sensores são detectadas por comparação do valor lido pelo sensor com o valor reconstruído.

Palavras Chave

Identificação de sistemas, Controlo Linear Quadrático Gaussiano, falhas nos actuatores e nos sensores, controlo centralizado, controlo distribuído
# Contents

1 Introduction  
1.1 Motivation .......................... 3  
1.2 Literature review .................. 3  
1.3 Problem formulation .............. 5  
1.4 Main contributions ................. 5  
1.5 Dissertation outline ............... 5  

2 System Identification .............. 7  
2.1 Canal description .................. 9  
2.2 Model identification .............. 10  
2.2.1 SISO system identification ........ 11  
2.2.2 MIMO system identification .......... 13  
2.2.3 MIMO system identification assuming actuator faults ........... 17  
2.2.4 MISO system identification ........ 19  
2.2.5 MISO system identification assuming actuator faults ........... 22  

3 LQG Control .......................... 23  
3.1 LQG controller ..................... 25  
3.2 Linear-Quadratic Regulator .......... 25  
3.3 Kalman Filter ...................... 27  
3.4 Separation principle ............... 28  
3.5 Single gate LQG controller ........ 29  
3.5.1 Constraining closed-loop poles .......... 31  

4 Actuator Faults ....................... 33  
4.1 Centralized control tolerant to actuator faults ................... 35  
4.1.1 Three gate LQG controller ........ 35  
4.1.2 Two gate LQG controller .......... 38  
4.1.3 Actuator Fault Detection .......... 40  
4.1.4 Reconfiguration ................. 42
### Contents

4.2 Distributed control tolerant to actuator faults .............................................. 45  
  4.2.1 Distributed LQ control with accessible disturbances .......................... 45  
  4.2.2 Coordination between controllers ....................................................... 51  
  4.2.3 LQG controllers structure ................................................................. 51  
  4.2.4 Reconfiguration .................................................................................. 55  

5 Sensor Faults ................................................................................................. 59  
  5.1 Sensor fault definition ............................................................................. 61  
  5.2 Signal reconstruction ............................................................................... 61  
  5.3 Sensor Fault Detection ........................................................................... 62  
  5.4 Centralized control tolerant to sensor faults ........................................... 64  
  5.5 Distributed control tolerant to sensor faults ........................................... 64  

6 Experimental Results ..................................................................................... 69  
  6.1 Actuator faults ......................................................................................... 71  
  6.2 Sensor faults ............................................................................................ 74  

7 Conclusions .................................................................................................... 81  
  7.1 Main points ............................................................................................... 83  
  7.2 Future work .............................................................................................. 84  

A Butterworth filter coefficients ....................................................................... 89  

B Pontryagin’s Minimum Principle in discrete time ............................................ 93
List of Figures

2.1 Schematic of the NuHCC automatic canal. .................................................. 9
2.2 Open-loop response of pool 1 in the SIMULINK model, when a PRBS is applied to
the input around an equilibrium value of 5.96 cm. ........................................ 11
2.3 Comparison between the data collected from the open-loop simulation of the non-
linear model and the response of the ARX model, after filtering the data. .......... 12
2.4 Open-loop response of pools 1, 2 and 3 when a PRBS is applied to each gate
position around the equilibrium values 5.96 cm, 6.01 cm and 6.09 cm, for gates 1,
2 and 3, respectively, while the fourth gate is kept at the position 28.77 cm. The
SIMULINK model of the canal was used. ...................................................... 15
2.5 Time evolution of the variables proportional to the flow under gates 1, 2 and 3 in
the open-loop simulation of the SIMULINK model. A PRBS is applied to each gate
position around the equilibrium values 5.96 cm, 6.01 cm and 6.09 cm, for gates 1, 2
and 3, respectively, while the fourth gate is kept at the position 28.77 cm. ......... 16
2.6 Comparison between the data collected from the open-loop simulation of the SIMULINK
model and the response of the ARX model, after filtering the data. ................. 17
2.7 Open-loop response of the SIMULINK model, taking into account pools 1 and 3,
when gate 2 is kept at a height of 6 cm and gate 4 at 28.77 cm. Water levels above,
variables proportional to the flow under gates below. ................................... 18
2.8 Comparison between the data collected from the open-loop simulation of the SIMULINK
model and the response of the ARX model. The inputs are the positions of gates
1 and 3, the outputs are the water levels of pools 1 and 3 and gate 2 is stuck at a
height of 6 cm, while gate 4 is kept at a position of 28.77 cm. ....................... 19
2.9 Comparison between the data collected from the open-loop simulation of the SIMULINK
model and the response of the ARX model, without assuming simplifications in the
order matrices. The inputs are the positions of gates 1 and 3, the outputs are the
water levels of pools 1 and 3 and gate 2 is stuck at a height of 6 cm, while gate 4 is
kept at a position of 28.77 cm. ................................................................. 19
2.10 Decomposition of the main system (canal) into subsystems (pools) which interact
with each other. ......................................................................................... 20
List of Figures

2.11 Comparison between the data collected from the open-loop simulation of the SIMULINK model and the response of the ARX models, after filtering the data. .................. 21

2.12 Comparison between the data collected from the open-loop simulation of the SIMULINK model and the response of the ARX models, after filtering the data. .................. 22

3.1 Schematic of a Linear-Quadratic Regulator including integral action. .................. 26

3.2 Block diagram of the canal subject to a centralized multivariable LQG controller, where 
\[ u = f(v) \] stand for the variable change of equation (2.7). .................. 29

3.3 Closed-loop response of the controlled SIMULINK model, only taking into account the water level of pool 1 and the position of gate 1, for different values of \( \rho \), considering full access to the state. Gates 2, 3 and 4 are kept at the positions 6.01 cm, 6.09 cm and 28.77 cm, respectively. .................. 30

3.4 Closed-loop response of the controlled SIMULINK model, only taking into account the water level of pool 1 and the position of gate 1, with estimated state, for different values of \( r_n \), while keeping \( \rho = 10^3 \) and \( q = 1 \). Gates 2, 3 and 4 are kept at the positions 6.01 cm, 6.09 cm and 28.77 cm, respectively. .................. 30

3.5 Closed-loop response of the controlled SIMULINK model, only taking into account the water level of pool 1 and the position of gate 1, with estimated state, for different values of \( q \), while keeping \( \rho = 10^3 \) and \( r_n = 10^2 \). Gates 2, 3 and 4 are kept at the positions 6.01 cm, 6.09 cm and 28.77 cm, respectively. .................. 31

3.6 Closed-loop response of the controlled SIMULINK model, only taking into account the water level of pool 1 and the position of gate 1, with estimated state, for different values of \( \alpha \), while keeping \( \rho = 10^3 \), \( r_n = 10^2 \) and \( q = 1 \). Gates 2, 3 and 4 are kept at the positions 6.01 cm, 6.09 cm and 28.77 cm, respectively. .................. 32

3.7 Closed-loop response of the controlled SIMULINK model, only taking into account the water level of pool 1 and the position of gate 1. The LQG parameters are \( \rho = 10^3 \), \( r_n = 10^2 \), \( q = 1 \) and \( \alpha = 1.005 \). Gates 2, 3 and 4 are kept at the positions 6.01 cm, 6.09 cm and 28.77 cm, respectively. .................. 32

4.1 Reconfigurable controller with actuator fault detection. The integrator is included in the blocks that represent the controllers. Block \( u = f(v) \) represents equation (2.1), which relates the gate position with the flow passing under it. .................. 35

4.2 Multivariable controller in normal operation. .................. 36

4.3 Closed-loop response of the controlled SIMULINK model, using centralized control. Only the water levels of pools 1, 2 and 3 and the positions of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controller parameters are \( \rho_i = 10^3 \), \( r_{ni} = 10^2 \), \( q = 1 \) and \( \alpha = 1.005 \). .................. 37

4.4 Multivariable controller after a fault in gate 2 is detected. .................. 38
List of Figures

4.5 Closed-loop response of the controlled SIMULINK model, using centralized control, when gate 2 is kept at the position of 6 cm. Only the water levels of pools 1, 2 and 3 and the position of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controllers parameters are $\rho_i = 10^3$, $r_{ni} = 10^2$, $q = 1$ and $\alpha = 1.005$. ................................................................. 39

4.6 Time evolution of $\Pi$ (blue), $\Pi_{max}$ (red) and fault occurrence (green) above. Fault detection indicator below. ................................................................. 41

4.7 Effects of the variation of $\gamma$ on the number of false alarms and the time for detection, while keeping $\Pi_{max} = 5$ (a, b, c), and effects of the variation of $\Pi_{max}$ on the number of false alarms and the time for detection, while keeping $\gamma = 0.95$ (d, e, f). .................. 42

4.8 Discrete states in controller reconfiguration. ................................................................. 43

4.9 Closed-loop response of the controlled SIMULINK model, using centralized control, when gate 2 remains at a position of nearly 4 cm after time instant 8900 s. Only the water levels of pools 1, 2 and 3 and the position of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controllers parameters are $\rho_i = 10^3$, $r_{ni} = 10^2$, $q = 1$ and $\alpha = 1.005$. .................................................. 44

4.10 Block diagram of the canal subject to the fault controller. A bumpless transition is accomplished by initializing the integrator such that the controller has as output the last gate position values read from the canal. ................................................................. 46

4.11 Closed-loop response of the controlled SIMULINK model, using centralized control, when gate 2 remains at a position of nearly 4 cm after time instant 8900 s. Only the water levels of pools 1, 2 and 3 and the position of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controllers parameters are $\rho_i = 10^3$, $r_{ni} = 10^2$, $q = 1$ and $\alpha = 1.005$. Bumpless transition between controllers. .................................................. 47

4.12 Schematics of the Distributed LQG control applied to the canal. ............................... 48

4.13 Distributed controller in normal operation. ................................................................. 52

4.14 Block diagram of a pool subject to a LQG controller with feedforward action. .......... 52

4.15 Closed-loop response of the controlled SIMULINK model, using distributed control. Only the water levels of pools 1, 2 and 3 and the positions of the respective gates are taken into account, while gate 4 is kept constant at the position of 28.77 cm. The controllers parameters are $\rho_i = 10^3$, $r_{ni} = 10^2$, $q = 1$ and $\alpha = 1.005$. .................. 53

4.16 Distributed controller after a fault is detected in gate 2. ........................................ 54
4.17 Closed-loop response of the controlled SIMULINK model, using distributed control, when gate 2 is kept constant at a position of 6 cm. Only the water levels of pools 1, 2 and 3 and the positions of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controllers parameters are $\rho_i = 10^3$, $r_{ni} = 10^2$, $q = 1$ and $\alpha = 1.005$. .......................... 56

4.18 Closed-loop response of the controlled SIMULINK model, using distributed control, when gate 2 remains at a position of nearly 4 cm after time instant 8900 s. Only the water levels of pools 1, 2 and 3 and the positions of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controllers parameters are $\rho_i = 10^3$, $r_{ni} = 10^2$, $q = 1$ and $\alpha = 1.005$.................. 57

4.19 Closed-loop response of the controlled SIMULINK model, using distributed control, when gate 2 remains at a position of nearly 4 cm after time instant 8900 s. Only the water levels of pools 1, 2 and 3 and the positions of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controllers parameters are $\rho_i = 10^3$, $r_{ni} = 10^2$, $q = 1$ and $\alpha = 1.005$. Bumpless transition between controllers. .......................... 58

5.1 Reconstruction of the value read by the sensor placed at the end of pool 2, $J_2$, from the value read by the sensor placed at the center of pool 2, $C_2$. ................. 61

5.2 Effect of the variation of $\gamma$ on the number of false alarms and the time of fault detection and recovery, while keeping $\Pi_{max} = 0.03 m$ (a, b, c), and effect of the variation of $\Pi_{max}$ on the number of false alarms and the time of fault detection and recovery, while keeping $\gamma = 0.9$ (d, e, f). .......................... 63

5.3 Closed-loop response of the controlled SIMULINK model, when sensor 2 fails at time 4700 s, using centralized control. Only the water levels of pools 1, 2 and 3 and the positions of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controller parameters are $\rho_i = 10^3$, $r_{ni} = 10^2$, $q = 1$ and $\alpha = 1.005$. .......................... 65

5.4 Comparison between the real value of the water level at the end of pool 2 (green) and the reconstructed value (red). The reference is presented in blue. ................. 66

5.5 Estimation error between the real value of the water level downstream of pool 2 and the one reconstructed through the value read by the sensor placed at the center of pool 2. The threshold is represented in red. .......................... 66

5.6 Estimation error between the value read by the sensor placed at the end of pool 2 and the value reconstructed through the readings of the sensor placed at the center of pool 2. The threshold is represented in red and the vertical orange line signals the fault occurrence. .......................... 66
5.7 Closed-loop response of the controlled SIMULINK model, when sensor 2 fails at time $4700 \, s$, using distributed control. Only the water levels of pools 1, 2 and 3 and the positions of the respective gates are taken into account, while gate 4 is kept at the position of $28.77 \, cm$. The controller parameters are $\rho_i = 10^3$, $r_n_i = 10^2$, $q = 1$ and $\alpha = 1.005$. ................................................................. 67

6.1 Closed-loop response of the canal subject to centralized control. Reconfiguration after a fault in gate 2 is detected (yellow mark). Both controllers parameters are $\rho_i = 1000$, $r_n_i = 100$, $q = 1$ and $\alpha = 1.004$. Water levels above, with the respective references, and the gate positions below, their true value in green and the command sent in blue. The red vertical line signals the fault occurrence. ....................... 71

6.2 Closed-loop response of the canal subject to distributed control. Reconfiguration after a fault in gate 2 is detected (yellow mark). Both controllers parameters are $\rho_i = 300$, $r_n_i = 100$, $q = 1$ and $\alpha = 1.004$. Water levels above, with the respective references, and the gate positions below, their true values in green and the command sent in blue. The red vertical line signals the fault occurrence. ....................... 73

6.3 Closed-loop response of the canal subject to centralized control. Signal reconstruction used after a fault is detected in sensor 2 (first yellow line) and until the fault recovery is detected (second yellow line). The controller parameters are $\rho_i = 1000$, $r_n_i = 100$, $q = 1$ and $\alpha = 1.004$. Water levels above, with the respective references, and the gate positions below, their true value in green and the command sent in blue. The red vertical line signals the fault occurrence. ....................... 74

6.4 Performance index (above) and fault detection indicator (below), when centralized control is used. ................................................................. 75

6.5 Comparison between the real value of the water level at the end of pool 2 (blue) and the reconstructed signal (green), using centralized control. Reconstructed signal built from the central sensor and flows drawn by gates 1 and 2. ....................... 76

6.6 Comparison between the real value of the water level at the end of pool 2 (blue) and the reconstructed signal (green), using centralized control. Only the value read by the sensor placed at the center of pool 2 was used in the reconstruction. .... 76

6.7 Closed-loop response of the canal subject to distributed control. Signal reconstruction used after a fault occurs in sensor 2 (first yellow line) and until the fault recovery is detected (second yellow line). The controllers parameters are $\rho_i = 300$, $r_n_i = 100$, $q = 1$ and $\alpha = 1.004$. Water levels above, with the respective references, and the gate positions below, their true value in green and the command sent in blue. The red vertical line signals the fault occurrence. ....................... 77

6.8 Performance index (above) and fault detection indicator (below), when distributed control is used. ................................................................. 78
6.9 Comparison between the real value of the water level at the end of pool 2 (blue) and the reconstructed signal (green), using distributed control. Reconstructed signal built from the central sensor and flows drawn by gates 1 and 2. . . . . . . . . . . . . . . . . . . . . . . . . . 78
6.10 Comparison between the real value of the water level at the end of pool 2 (blue) and the reconstructed signal (green), using distributed control. Only the value read by the sensor placed at the center of pool 2 was used in the reconstruction. . . . . . . . . . . . . . . . . . . . . . . . . . 78
List of Acronyms

**ARMAX** AutoRegressive Moving Average with eXogenous input

**ARX** AutoRegressive with eXogenous input

**BJ** Box-Jenkins

**FTC** Fault Tolerant Control

**LQ** Linear-Quadratic

**LQE** Linear-Quadratic Estimator

**LQG** Linear-Quadratic-Gaussian

**LQR** Linear-Quadratic Regulator

**LTI** Linear Time Invariant

**MIMO** Multiple-Input Multiple-Output

**MISO** Multiple-Input Single-Output

**NuHCC** Núcleo de Hidráulica e Controlo de Canais

**OE** Output-Error

**P** Proportional

**PI** Proportional Integral

**PID** Proportional Integral Derivative

**PLCs** Programmable Logic Controllers

**PRBS** Pseudo-Random Binary Sequence

**SISO** Single-Input Single-Output
# Introduction

## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Motivation</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Literature review</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Problem formulation</td>
<td>5</td>
</tr>
<tr>
<td>1.4 Main contributions</td>
<td>5</td>
</tr>
<tr>
<td>1.5 Dissertation outline</td>
<td>5</td>
</tr>
</tbody>
</table>
1. Introduction
1.1 Motivation

In this chapter the problem of faults occurring in water delivery systems is introduced as the motivation for the writing of this dissertation. After a short literature review, which enframes the reader in the subject and justifies some of the choices made, the problem to be solved is explained and the contributions of this dissertation exposed.

1.1 Motivation

Freshwater scarcity has become a major concern throughout the world. This is mainly due to the significant rise in water consumption that has been verified, where irrigation systems assume great responsibility. These systems entail a large socio-economic impact since a high percentage of the Gross Domestic Product of several countries stems from agricultural irrigation products.

In irrigation systems, the major difficulty is the management of the available water, that may lead to significant waste if not attended carefully. It is therefore important to improve control systems of irrigation channels in order to minimize these losses. Since water demand for irrigation varies with time, we seek a robust and stable control system that is able to respond quickly enough to sudden variations in the water levels, so that those can be maintained at the desired values. Even though good results have already been achieved in this matter, physical problems, such as actuator or sensor failure, will most likely destabilize the regular operation of the controlled system and lead to unstable behaviors that may result in substantially more waste. For this reason, it is essential to develop Fault Tolerant Control (FTC) systems.

This type of systems that can be divided into several subsystems that interact with their neighbors, especially in large scale, justify the use of distributed control. In large scale systems, centralized control, where one controller alone acquires the information of the global system, usually resulting in a better performance, can become impractical in some cases. On the one hand, the numerous inputs and outputs require a large scale computation effort; on the other hand, in the event that the centralized controller fails, the entire system stays unbridled. The distributed framework, where a controller is considered for each subsystem, benefits from requiring less computational effort, while being error-tolerant and flexible to the system structure. Furthermore, it does not need the information of the global system.

1.2 Literature review

Control of water delivery canals has been a subject under intensive research for the last decades. Systems of these type have been modeled either through system identification [27], [11], or using the Saint-Venant equations [23], [20], [2], having both approaches been proved able to adequately capture the dynamics of real open water canals. However, due to the complexity of the Saint-Venant equations, system identification methods are usually preferred for control design and prediction purposes [23].
1. Introduction

Irrigation systems are commonly controlled by Proportional (P), Proportional Integral (PI) or Proportional Integral Derivative (PID) controllers [22], [28], [18], [19], as they present advantages regarding ease of design and implementation, or by Linear-Quadratic (LQ) controllers [29], [21], [17]. Compared to the former, the latter present more degrees of freedom and, however requiring more design effort, they lead to a better performance [28], [29]. For this reason, the controllers to be used in this work are of LQ type, which are still considered simple, and at the same time are reliable and ensure stability for the nominal plant model.

In what concerns distributed control, although usually presenting a lower performance compared to centralized control, this type of control is preferred in large-scale systems that can be divided into subsystems that interact mainly with their neighbors [37]. In particular with regard to irrigation systems, distributed control has been a subject of growing interest [8], [12], [17], and results have been obtained that prove centralized and distributed control performances to be comparable [8]. Furthermore, the distributed structure presents by itself some degree of fault tolerance since, in the event that an actuator or a sensor fails, only the subsystem involved is affected [7].

The concept of FTC has been largely discussed over the last twenty years [1], [5], [35], [34], [32], [13], [25]. It consists in developing techniques that prevent that a simple fault becomes a serious problem and can range from simple appropriate retuning to complex reconfiguration [1]. Different reconfiguration techniques have been presented in [13], [32], [34] and [35], the last two focusing in fault detection and diagnosis as well. Another type of approach models the faults as disturbances that are estimated and compensated by the controller [36].

With regard to water delivery systems, reconfiguration schemes to mitigate both sensor and actuator fault effects in irrigation channels are described in [9]. This reconfiguration approach makes use of the redundancy inherent in the system to recover from both type of faults. In sensor faults, the signal provided by the faulty sensor is replaced by an estimated value obtained from the other available measurements and through the use of an observer. Actuator faults are dealt with by reconfiguring the control loop, accompanied by a relaxation of the control objective. Both these approaches are considered in this work.

An aspect of great importance in the control reconfiguration technique that is to be developed in this dissertation is bumpless transfer between controllers. This issue arises upon switching from one controller acting in closed-loop to another one waiting to take over, when the controllers are conducted by different control laws and have potentially different outputs. A way to avoid the transient caused by the instantaneous switching is to force the initial value of the output of the off-line controller to be the same as that of the controller in closed-loop [31]. Such control systems in which switching occurs, are considered hybrid, as they combine continuous and discrete behaviors and logic decisions, and they raise issues related to stability that must be taken into account [14].

Other topics found in the literature concerning irrigation channels are control loop monitoring
1.3 Problem formulation

This dissertation focuses on the design of control systems for a water delivery canal that are tolerant to actuator and sensor faults. The water delivery canal is divided into several pools and the aim is to maintain the downstream water level of each pool at a desired value. Both centralized and distributed control options are discussed.

The centralized control system tolerant to actuator faults consists of two multivariable LQG controllers, one that is used when all gate actuators are functioning properly and the other one to be used in the event a fault is detected. In order to detect the occurrence of a fault, a fault detection algorithm is also to be developed.

Still concerning actuator faults, but through the usage of distributed control, two controllers for each pool are needed, one for the case that a fault has been detected and the other one for the case that it has not. The same fault detection algorithm is used.

As to the control systems tolerant to sensor faults, the controllers that compose both the centralized and the distributed control systems are the same as the ones used in the control systems tolerant to actuator faults when no faults occur. When a sensor fails, i.e., when it no longer reads the correct water level, the signal lost needs to be reconstructed and the reconstructed signal must replace the sensor readings as input to the controller. Once again a fault detector is needed, so as to switch from the value read by the sensor to the reconstructed signal.

In this work the problem of fault isolation is considered to be out of scope.

1.4 Main contributions

The main contributions of this dissertation consist in the experimental demonstration of control systems tolerant to actuator and sensor faults applied to a water delivery canal. Both centralized and distributed control architectures are proposed for such purpose. Results with both centralized and distributed LQG controllers are presented, considering actuator faults and sensors faults separately.

1.5 Dissertation outline

The present dissertation is organized as follows.

Chapter 1 begins by explaining the reasons that motivated this work and, after a review of the literature available on this matter, the problem to be solved and the dissertation main contributions are presented.
1. Introduction

Chapter 2, after a brief description of the canal, is where the identification of the several systems to be considered is exposed. Each controller design is based on a linear model obtained through system identification methods.

Chapter 3 presents the LQG controller, the one to be used in the present study. A single gate example is used to evaluate the influence of each of the controller parameters in the water level response.

Chapter 4 develops both centralized and distributed control systems tolerant to actuator faults. The controllers to be used are described, as well as the fault detection algorithm, and the results of several simulations of a nonlinear model of the canal subject to those control systems are presented.

Chapter 5 describes a solution to the sensor signal reconstruction and fault detection problems. Some results of simulations of the nonlinear model of the canal are shown for both the centralized and the distributed control systems.

Chapter 6 presents some of the experimental results obtained, for control systems tolerant to actuator faults, using both centralized and distributed control and for control systems tolerant to sensor faults, using both centralized and distributed control.

Chapter 7 draws conclusions and points out the work that is still to be developed on the matter.
2
System Identification

Contents

2.1 Canal description .......................................................... 9
2.2 Model identification ..................................................... 10
2.1 Canal description

In this chapter a short description of the canal is elaborated and the linear models on which the controllers to be designed are based are presented.

2.1 Canal description

The canal on which this study is based belongs to Núcleo de Hidráulica e Controlo de Canais (NuHCC) of the University of Évora, Portugal. This system has an automatic canal and a return canal (traditional canal). The automatic canal consists of four pools separated from each other by undershot gates, whereas the last pool is terminated by an overshot one, as shown in figure 2.1. The traditional canal closes the circuit by returning the water drained by gate 4 to a reservoir placed upstream of the automatic canal.

![Figure 2.1: Schematic of the NuHCC automatic canal.](image)

The automatic canal is 141 m long and has a trapezoidal shape, with a cross-section of bottom width 0.15 m, side slope 1:0.15 (V:H) and depth 0.90 m. Its average longitudinal bottom slope is about $1.5 \times 10^{-3}$ and the maximum flow admitted 90 l/s.

Three level sensors are placed along each pool, measuring the correspondent upstream, center and downstream water levels, which are represented in figure 2.1 by $M_i$, $C_i$ and $J_i$, respectively. In the same figure, $Q_i$ represents the flow of offtake $i$ (lateral hole placed downstream of pool $i$, near the bottom of the canal), which is controlled by the motorized butterfly valve $V_{oi}$.

Assuming $u_i$ to be the position of the undershot gate $i$, the flow under this gate can be written as a function of $u_i$ as

$$q_i(t) = C_{ds} W u_i(t) \sqrt{2g(h_{up\text{stream},i} - h_{down\text{stream},i})},$$  \hspace{1cm} (2.1)

where $q_i(t) \in \mathbb{R}$ is the flow under gate $i$, $C_{ds} \in \mathbb{R}$ is the discharge coefficient, $W = 0.49 m$ is the width of the gates, $h_{up\text{stream},i} \in \mathbb{R}^+$ is the water level upstream of gate $i$ and $h_{down\text{stream},i} \in \mathbb{R}^+$ is the water level downstream of gate $i$ [16]. This expression requires all variables to be expressed in SI units.

A more extensive description of the canal can be found in [16].
2. System Identification

The control problem to be addressed throughout this dissertation consists in taking the water level downstream of each pool, $J_i$, to the desired value, by manipulating the position of the gates. Each of the actual gate positions $u_{r,i}$ is manipulated by a command signal $u_i$. However, the Programmable Logic Controllers (PLCs) that command gate motors are programmed such that $u_{r,i}$ only changes in response to $u_i$ if $|u_i - u_{r,i}| \geq 0.5\text{mm}$. This dead zone entails a nonlinearity that limits the controller achievable precision when tracking a reference and it can even lead to small amplitude oscillations.

In this work, we establish a reference for each pool, all consisting in square waves but with different equilibrium points, which the respective water level is intended to follow, while rejecting disturbances. These disturbances derive from the lateral offtakes, which in a non-experimental canal would serve a purpose to irrigation.

2.2 Model identification

In order to design a control system, a model that describes the relevant dynamics of the canal is needed. The model can be obtained either by using the St. Venant equations or by using system identification models based on operational data from the canal. A nonlinear model based on the St. Venant Equations has already been developed for the present irrigation canal in SIMULINK. From this nonlinear model, a linear one can be built based on parametric methods that estimate parameters of model structures such as

- AutoRegressive with eXogenous input (ARX)
- AutoRegressive Moving Average with eXogenous input (ARMAX)
- Box-Jenkins (BJ)
- Output-Error (OE).

Previous studies on this same canal proved ARX to be the best choice [24], [6]. Since noise attached to the data collected from simulation of the canal nonlinear model is not white [24], ARX would not be the most appropriate structure if the ultimate goal was identification. As the aim is to obtain a suitable control system and the other models do not improve much the results when compared to the ones obtained with ARX, the latter is preferable since it is the simplest one.

Once the method to be used is selected, the next step is to choose an appropriate signal to excite the system, or in this case the nonlinear model created. The Pseudo-Random Binary Sequence (PRBS) was selected, as it excites the plant over a range of frequencies that is wider than that of most other signals, allowing a better identification of the system transients. In order to avoid nonlinearities, the signal amplitude must be small and its period must be large enough for the water level to stabilize. Furthermore, since we are dealing with a slow process, the sampling
2.2 Model identification

period must also be large. It has been established in [24] that an adequate value for the sampling period is 2 s.

The value of 30 l/s was chosen in the present study for intake flow.

2.2.1 SISO system identification

We first consider the identification of the Single-Input Single-Output (SISO) system that consists of the first pool of the NuHCC canal. The input of this system is the position of gate 1 and the output is the downstream level of pool 1. In order to obtain data to identify the system, we use the SIMULINK model of the canal and apply a PRBS to the input around an equilibrium point, while the other gate positions are kept constant. The equilibrium points around which the system is to be operated are 70 cm, 60 cm, 50 cm and 40 cm, for the water levels of pools from 1 to 4, respectively. The respective gate positions that lead to those levels are 5.96 cm, 6.01 cm, 6.09 cm and 28.77 cm. The data collected from the simulation of the nonlinear model is shown in figure 2.2. According to figure 2.1, the downstream water level of pool 1 is represented by \( J_1 \) and the respective gate position by \( u_1 \). We can observe that the system has an inverse response, meaning that when the gate opens (gate position assumes a higher value), the water level decreases, and the other way around, which is in accordance with the fact that the gate is placed at the end of the pool.

![Figure 2.2: Open-loop response of pool 1 in the SIMULINK model, when a PRBS is applied to the input around an equilibrium value of 5.96 cm.](image)

Identification using an ARX model is based on the computation of least squares estimates of the parameters in the model

\[
A(q^{-1}) y(t) = B(q^{-1}) u(t - n_k) + e(t),
\]

where \( y(t) \in \mathbb{R} \) is the output, \( u(t) \in \mathbb{R} \) is the input, \( e(t) \in \mathbb{R} \) is a Gaussian white noise sequence, \( q^{-1} \) represents the backward shift operator, \( n_k \) the pure time delay and

\[
A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_{n_a} q^{-n_a},
\]
2. System Identification

\[ B(q^{-1}) = b_0 + b_1 q^{-1} + \ldots + b_n q^{-n_b}, \]

in which \( a_i, i = 1, \ldots, n_a \) and \( b_i, i = 0, \ldots, n_b \) are real parameters, with \( n_a, n_b \) positive integers.

The aim of the identification process is to find the parameters \( a_i \) and \( b_i \), given the orders of the polynomials \( A(q^{-1}) \) and \( B(q^{-1}) \), \( n_a \) and \( n_b \), and the delay \( n_k \).

Since the least squares method admits data to have zero mean, before the identification process the mean needs to be removed from the data, as well as the initial transient. The best orders found, i.e., the orders that lead to the best fit between data collected from the nonlinear model and the ARX model response to that same data are \( n_a = 4 \) and \( n_b = 2 \) and the delay \( n_k = 1 \).

In order to quantify the adjustment of the model response to the data, we define a measure of fit given by

\[ \text{fit(\%)} = \frac{\| y_{\text{data}} - \hat{y}_{\text{model}} \|}{\| y_{\text{data}} - \bar{y}_{\text{data}} \|} \times 100, \]

where \( y_{\text{data}} \) is the output of the validation data, \( \bar{y}_{\text{data}} \) is the mean of the output of the validation data and \( \hat{y}_{\text{model}} \) is the model output.

For this case a fit of 90.83\% was obtained between the validation data and the model response. Since ARX assumes the data noise to be white, when in fact it is not, it is expected that the reduction of noise improves the previous value. With this in mind, a fifth order butterworth filter with bandwidth 0.1 rad/s is applied to the data before the identification. The filter coefficients can be found in appendix A. Figure 2.3 shows the comparison between the validation data and the new model response, which presents a fit of 94.57\%. The orders that lead to this result are \( n_a = 4 \) and \( n_b = 4 \), with a delay of \( n_k = 1 \).

On the other hand, if an ARMAX structure is used, the result obtained for the fit between data and the model response is barely improved. As opposed to the least squares method, the maximum likelihood method provides unbiased estimates in the presence of colored noise. With respect to the ARX model, the ARMAX model has an extra \( C(q^{-1}) \) polynomial in the equation and the latter is given by

\[ A(q^{-1}) y(t) = B(q^{-1}) u(t - n_k) + C(q^{-1}) e(t), \quad (2.3) \]

where

\[ C(q^{-1}) = 1 + c_1 q^{-1} + \ldots + c_{n_c} q^{-n_c}. \]
2.2 Model identification

in which $c_i$, $i = 1, \ldots, n_c$, is a real parameter, with $n_c$ a positive integer.

Without filtering the data before the identification, the resulting ARMAX model response fitted the output of the validation set in 94.33%. Filtering the data before identification did not improve the response, in fact it even worsen it slightly, as a fit of 93.98% was now obtained for the same validation set.

Since the best result obtained was accomplished with an ARX model and this structure is simpler than an ARMAX one, hereafter the identification of systems is to be performed with the least squares method, bearing in mind that the data must be filtered beforehand. The filter to be used in data filtering for identification purposes is the fifth order butterworth presented, unless stated otherwise.

2.2.2 MIMO system identification

So as to design a centralized control system for the canal under study, we need a Multiple-Input Multiple-Output (MIMO) linear model of it. Considering only pools 1, 2 and 3, the inputs to this model are the positions of the respective gates and the outputs are the downstream water levels of each pool.

The ARX model structure in the multivariable form is

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

where $y(t) \in \mathbb{R}^{n_y}$, with $n_y$ the number of outputs, $u(t) \in \mathbb{R}^{n_u}$, with $n_u$ the number of inputs, $e(t) \in \mathbb{R}^{n_y}$ is a Gaussian white noise sequence, $A(q^{-1}) \in \mathbb{R}^{n_y \times n_y}$ and $B(q^{-1}) \in \mathbb{R}^{n_y \times n_u}$. Each entry of matrix $A(q^{-1})$ is a polynomial and its $ml$ element, with $m, l = 1, \ldots, n_y$, is given by

$$a_{ml}(q^{-1}) = 1 + \sum_{i=1}^{n_{am}} \alpha_{ml}^i q^{-i},$$

in which $\alpha_{ml}^i \in \mathbb{R}$ and $n_{am}$ is a positive integer that represents the order of the $ml$ polynomial. Similarly, each entry of matrix $B(q^{-1})$ is a polynomial and its $mj$ element, with $j = 1, \ldots, n_u$, is given by

$$b_{mj}(q^{-1}) = \sum_{i=0}^{n_{bm}} \beta_{mj}^i q^{-n_{km} - i},$$

where $\beta_{mj}^i \in \mathbb{R}$, $n_{km}$ is a positive integer that represents the delay and $n_{bm}$ is a positive integer that represents the order of the $m-j$ polynomial.

In order to match the distributed control algorithm to be used, matrix $A(q^{-1})$ is forced to have the structure

$$A(q^{-1}) = \begin{bmatrix}
1 + \sum_{i=1}^{n_{a1}} \alpha_{11}^i q^{-i} & 0 & 0 \\
0 & 1 + \sum_{i=1}^{n_{a2}} \alpha_{22}^i q^{-i} & 0 \\
0 & 0 & 1 + \sum_{i=1}^{n_{a3}} \alpha_{33}^i q^{-i}
\end{bmatrix}.$$
2. System Identification

This means that the water level of one pool is not influenced by the water levels of the other pools. Instead, the values of each water level only depend on the values of the same water level in previous time instants.

In the same way, it also happens that not all inputs influence each water level. In fact, the position of gate 3 barely affects the water level of pool 1 and a change in the position of gate 1 is barely noticed by the water level of pool 3. Therefore, the structure of matrix \( B(q^{-1}) \), whose \( mj \) entry gives the influence of input \( j \) on output \( m \), becomes

\[
B(q^{-1}) = \begin{bmatrix}
\sum_{i=0}^{n_{a_{11}}} \beta_{11}^i q^{-nk_{11}-i} & \sum_{i=0}^{n_{a_{12}}} \beta_{12}^i q^{-nk_{12}-i} & 0 \\
\sum_{i=0}^{n_{b_{11}}} \beta_{21}^i q^{-nk_{21}-i} & \sum_{i=0}^{n_{b_{12}}} \beta_{22}^i q^{-nk_{22}-i} & \sum_{i=0}^{n_{b_{13}}} \beta_{23}^i q^{-nk_{23}-i} \\
0 & \sum_{i=0}^{n_{b_{21}}} \beta_{31}^i q^{-nk_{31}-i} & \sum_{i=0}^{n_{b_{22}}} \beta_{32}^i q^{-nk_{32}-i} & \sum_{i=0}^{n_{b_{23}}} \beta_{33}^i q^{-nk_{33}-i}
\end{bmatrix}.
\]

In order to find an ARX model that approximates the system, the orders of the polynomials, \( n_{a_{ml}} \) and \( n_{b_{mj}} \), and the delays \( n_{k_{mj}} \) must be chosen. These orders and delays can be organized into three matrices that have the same structure as matrices \( A(q^{-1}) \) and \( B(q^{-1}) \). Those matrices can be written as

\[
N_A = \begin{bmatrix} n_{a_{11}} & 0 & 0 & 0 \\
0 & n_{a_{22}} & 0 & 0 \\
0 & 0 & n_{a_{33}} \end{bmatrix}, \quad N_B = \begin{bmatrix} n_{b_{11}} & n_{b_{12}} & 0 \\
0 & n_{b_{22}} & n_{b_{23}} \\
0 & 0 & n_{b_{33}} \end{bmatrix}, \quad N_K = \begin{bmatrix} n_{k_{11}} & n_{k_{12}} & 0 \\
0 & n_{k_{22}} & n_{k_{23}} \\
0 & 0 & n_{k_{33}} \end{bmatrix}.
\]

So as to collect data for the identification, the SIMULINK model of the canal is simulated. In the simulation, three PRBS are applied to inputs (gate positions) 1, 2 and 3 around the equilibrium values of 5.96 cm, 6.01 cm and 6.09 cm, respectively, while the fourth gate is kept still at the position 28.77 cm. The data obtained is shown in figure 2.4.

However, the identification results obtained were unsatisfactory, in the sense that it was not possible to obtain an ARX model whose response fitted the data collected in more than 50%, no matter the orders used. So as to accomplish a better fit, a change of the input variable is considered: instead of using the gate position, which entails nonlinearities, the flow drawn by the gate can be used. This change of variables is obtained through equation (2.1).

However, instead of the actual flow drawn by each gate, \( q_i \), we used a value proportional to it, \( v_i \), that relates to \( q_i \) by

\[
q_i(t) = C_{ds} v_i(t),
\]

where

\[
v_i(t) = u_i(t)W \sqrt{2g(h_{upstream,i} - h_{downstream,i}).}
\]

(2.7)

The reason for using the value of \( v_i \) instead of \( q_i \) was the fact that at first the discharge coefficient, \( C_{ds} \), was unknown, and afterwards we continued to use the same algorithm. Nevertheless, this does not affect the results obtained.
2.2 Model identification

Figure 2.4: Open-loop response of pools 1, 2 and 3 when a PRBS is applied to each gate position around the equilibrium values 5.96 cm, 6.01 cm and 6.09 cm, for gates 1, 2 and 3, respectively, while the fourth gate is kept at the position 28.77 cm. The SIMULINK model of the canal was used.
2. System Identification

Figure 2.5 shows the time evolution of $v_i$ for each gate $i$, obtained with equation (2.7) from the gate positions, $u_i$, represented in figure 2.4. As we can see, in figure 2.5, the equilibrium point around which variations are being made is $0.05 \text{ m}^3/\text{s}$ for all gates. Since the intake flow is $0.03 \text{ m}^3/\text{s}$, and so must be the equilibrium point of the flow drawn by each gate, we conclude that $C_{ds} = 3/5$.

In fact, after this change of variables, the results obtained are much more satisfying, since the ARX model response fits the validation data in more than 90\%, as shown in figure 2.6. For this reason, henceforward the identification of MIMO systems is to be performed using the flow under the gate instead of its position.

Moreover, the results obtained also prove that the matrix structures assumed for $A(q^{-1})$ and $B(q^{-1})$ are acceptable. The orders and delays of the ARX model whose response is presented in figure 2.6 are:

$$
N_A = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \quad N_B = \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & 4 \\ 0 & 3 & 3 \end{bmatrix}, \quad N_K = \begin{bmatrix} 1 & 2 & 0 \\ 1 & 1 & 2 \\ 0 & 1 & 1 \end{bmatrix}.
$$

After converting the ARX model obtained to a discrete-time state-space model such that

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

$$
y(k) = Cx(k),
$$

matrices $A$, $B$ and $C$ will have the following structure

$$
A = \begin{bmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} & 0 \\ B_{21} & B_{22} & B_{23} \\ 0 & B_{32} & B_{33} \end{bmatrix}, \quad C = \begin{bmatrix} C_{11} & 0 & 0 \\ 0 & C_{22} & 0 \\ 0 & 0 & C_{33} \end{bmatrix}.
$$
2.2 Model identification

Comparison between data and model response

<table>
<thead>
<tr>
<th>J_1 [m]</th>
<th>J_2 [m]</th>
<th>J_3 [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>data; measured</td>
<td>model; fit: 92.56%</td>
<td>data; measured</td>
</tr>
<tr>
<td>data; measured</td>
<td>model; fit: 93.65%</td>
<td>data; measured</td>
</tr>
<tr>
<td>data; measured</td>
<td>model; fit: 94.74%</td>
<td>data; measured</td>
</tr>
</tbody>
</table>

Time [s] x 10^4

Figure 2.6: Comparison between the data collected from the open-loop simulation of the SIMULINK model and the response of the ARX model, after filtering the data.

which suggest that each pool interacts only with its neighbors and only through the inputs.

2.2.3 MIMO system identification assuming actuator faults

For the sake of simplicity, we assume that only gate 2 is susceptible of failing, and that the fault consists in the gate getting stuck at a certain position regardless of the command received. In order to design a control system tolerant to such a fault, two controllers are considered: one that controls the system in its regular operation and another one for the case that a fault in the actuator of gate 2 is detected. The first controller makes use of the model presented in section 2.2.2, which has three inputs and three outputs. The second controller, which deals with faults in gate 2, is responsible only for the water levels of pools 1 and 3, leaving the water level of pool 2 unattended. Therefore, this controller requires a linear model with two inputs, being the positions of gates 1 and 3, and two outputs, which are the water levels of the upstream pools.

Reproducing the procedure followed in section 2.2.2, in order to find a linear model that approximates the behavior of the canal, we begin by identifying using data from the SIMULINK model, this time keeping gate 2 at a chosen position. As in section 2.2.2, we seek an ARX model whose response is as similar as possible to the system response. Once again, we assume that the water level of one pool has no influence on the water level of the other, but since only two pools are taken into account, now both gates affect both water levels. Given this, we can organize the orders and delays of the polynomials (2.5) and (2.6) into 2 x 2 matrices:

\[ N_A = \begin{bmatrix} n_{a_1} & 0 \\ 0 & n_{a_2} \end{bmatrix}, \quad N_B = \begin{bmatrix} n_{b_1} & n_{b_2} \\ n_{b_3} & n_{b_4} \end{bmatrix}, \quad N_K = \begin{bmatrix} n_{k_1} & n_{k_2} \\ n_{k_3} & n_{k_4} \end{bmatrix}. \]

As before, in order to collect data for the identification, we apply a PRBS to each input of the system considered. Figure 2.7 shows the time evolution of the water levels of pools 1 and 3 and the flow drawn by the respective gate, when gate 2 is kept at a height of 6 cm and gate 4 at
2. System Identification

The value of $6\text{ cm}$ was selected as it is the equilibrium value around which this study is being carried out.

![Water levels and Flow drawn by the gates](image)

Figure 2.7: Open-loop response of the SIMULINK model, taking into account pools 1 and 3, when gate 2 is kept at a height of $6\text{ cm}$ and gate 4 at $28.77\text{ cm}$. Water levels above, variables proportional to the flow under gates below.

The ARX model found whose response best fits a validation data set collected from the SIMULINK model makes use of the orders and delays

$$N_A = \begin{bmatrix} 2 & 0 & 4 \\ 0 & 4 & 0 \end{bmatrix}, \quad N_B = \begin{bmatrix} 4 & 3 \\ 4 & 4 \end{bmatrix}, \quad N_K = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$ 

The comparison between the validation data and the model response is shown in figure 2.8.

If we do not consider the approximation of the $N_A$ matrix being diagonal, the best fits we obtain do not improve; on the contrary, they are even slightly worse. The comparison between the validation data and the ARX model response is shown in figure 2.9 and the orders and delays that lead to it are the following:

$$N_A = \begin{bmatrix} 2 & 1 & 4 \\ 2 & 4 & 1 \end{bmatrix}, \quad N_B = \begin{bmatrix} 4 & 3 \\ 0 & 4 \end{bmatrix}, \quad N_K = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$ 

Similar to what we saw in section 2.2.2, after converting the ARX model obtained to a discrete-time state-space model

$$x(k + 1) = A^F x(k) + B^F v(k),$$
2.2 Model identification

Comparison between data and model response

**Figure 2.8:** Comparison between the data collected from the open-loop simulation of the SIMULINK model and the response of the ARX model. The inputs are the positions of gates 1 and 3, the outputs are the water levels of pools 1 and 3 and gate 2 is stuck at a height of 6 cm, while gate 4 is kept at a position of 28.77 cm.

Comparison between data and model response

**Figure 2.9:** Comparison between the data collected from the open-loop simulation of the SIMULINK model and the response of the ARX model, without assuming simplifications in the order matrices. The inputs are the positions of gates 1 and 3, the outputs are the water levels of pools 1 and 3 and gate 2 is stuck at a height of 6 cm, while gate 4 is kept at a position of 28.77 cm.

with output equation

\[ y(k) = C^F x(k), \]

matrices \( A^F, B^F \) and \( C^F \) will have the following structure

\[ A^F = \begin{bmatrix} A_{11}^F & 0 \\ 0 & A_{22}^F \end{bmatrix}, \quad B^F = \begin{bmatrix} B_{11}^F \\ B_{21}^F \\ B_{22}^F \end{bmatrix}, \quad C^F = \begin{bmatrix} C_{11}^F & 0 \\ 0 & C_{22}^F \end{bmatrix}, \]

where the superscript \( F \) suggest the occurrence of a fault. This superscript is used to distinguish these matrices from those in section 2.2.2, since they are not the same.

### 2.2.4 MISO system identification

In opposition to centralized control, where only one controller is considered independently of the number of outputs, distributed control foresees the use of one controller for each output. Therefore, whereas in sections 2.2.2 and 2.2.3 we saw the canal as a whole system, we now regard it as a chain of serially connected subsystems, where each subsystem is a pool of the canal.
2. System Identification

As such, each subsystem has as output the water level of the correspondent pool and as input the position of the respective gate. Furthermore, we now consider the flows of the lateral offtakes as accessible disturbances, which we also introduce as inputs in the subsystems. Each subsystem is assumed to interact with its direct neighbors through the inputs, disregarding however the cross-coupling of states between subsystems. Figure 2.10 shows a scheme of the described chain of subsystems, where \( u_i \) is position of gate \( i \), \( y_i \) is the downstream water level of pool \( i \) and \( \text{off}_i \) is the disturbance derived from the offtake of the same pool. In the present case, where only three pools are considered, subsystem \( \Sigma_i \) corresponds to pool 2 and the extensions implied in figure 2.10 are not needed.

![Figure 2.10: Decomposition of the main system (canal) into subsystems (pools) which interact with each other.](image)

As to the identification process, each pool of the canal is seen as a Multiple-Input Single-Output (MISO) system, which has as output the water level of the corresponding pool and as inputs the respective gate position and offtake flow and those of its neighbors. Thus, depending on the number of neighbors of the pool at stake, the correspondent MISO model will have four or six inputs.

For each subsystem, in a total of three, we now seek a MISO ARX model whose response fits best the data collected from the SIMULINK model of the canal, this time taking into account the offtake flows variations.

Figure 2.11 shows the comparison between the validation data and the responses of the best ARX models found. Up to time instant \( 10^5 \) s only variations in the gate positions were made, after what only offtake flows suffered changes. The orders of the models that lead to these fits are the following

\[
N_{A_1} = 2, \quad N_{B_1} = [4 \ 2 \ 1 \ 4], \quad N_{K_1} = [1 \ 1 \ 1 \ 1],
\]

\[
N_{A_2} = 2, \quad N_{B_2} = [4 \ 4 \ 3 \ 3 \ 4 \ 4], \quad N_{K_2} = [1 \ 1 \ 1 \ 1 \ 1 \ 1],
\]

\[
N_{A_3} = 2, \quad N_{B_3} = [4 \ 4 \ 1 \ 4], \quad N_{K_3} = [1 \ 1 \ 1 \ 1].
\]

Once obtained the three ARX structures, they can be converted to discrete-time state-space...


2.2 Model identification

Figure 2.11: Comparison between the data collected from the open-loop simulation of the SIMULINK model and the response of the ARX models, after filtering the data.

models and for subsystem \( i \) we write it as

\[
x_i(k + 1) = A_{ii} x_i(k) + B_{ii} v_i(k) + \sum_{j=1}^{j=3} \sum_{i-1 \leq j \leq i+1} \Psi_{ij} v_j(k) + \sum_{j=1}^{j=3} \Gamma_{ij} d_{off_j}(k)
\]

(2.8)

\[
y_i(k) = C_{ii} x_i(k),
\]

where \( x_i \in \mathbb{R}^n \), with \( n \) the number of states, \( v_i \in \mathbb{R} \) is the input to subsystem \( i \) (flow under the gate), \( y_i \in \mathbb{R} \) is the output to subsystem \( i \) and \( d_{off_j} \in \mathbb{R} \) is the disturbance caused by the offtake of subsystem \( i \). Matrices \( A_{ii}[n \times n] \), \( B_{ii}[n \times 1] \) and \( C_{ii}[1 \times n] \) are the ones that model subsystem \( i \) as a SISO system with input \( u_i \) and output \( y_i \). Matrix \( \Psi_{ii}[n \times 1] \) models the influence of the inputs to the other subsystems in subsystem \( i \) and \( \Gamma_{ii}[n \times 1] \) models the influence of the offtakes in subsystem \( i \).

If we assemble these equations for all the three subsystems considered, we reach one state equation for the whole canal system, as well as one output equation, which are given by

\[
\begin{bmatrix}
x_1(k + 1) \\
x_2(k + 1) \\
x_3(k + 1)
\end{bmatrix} =
\begin{bmatrix}
A_{11} & 0 & 0 \\
0 & A_{22} & 0 \\
0 & 0 & A_{33}
\end{bmatrix}
\begin{bmatrix}
x_1(k) \\
x_2(k) \\
x_3(k)
\end{bmatrix} +
\begin{bmatrix}
B_{11} & 0 & 0 \\
0 & B_{22} & 0 \\
0 & 0 & B_{33}
\end{bmatrix}
\begin{bmatrix}
v_1(k) \\
v_2(k) \\
v_3(k)
\end{bmatrix} +
\begin{bmatrix}
0 & \Psi_{12} & 0 \\
\Psi_{21} & 0 & \Psi_{23} \\
0 & \Psi_{32} & 0
\end{bmatrix}
\begin{bmatrix}
v_1(k) \\
v_2(k) \\
v_3(k)
\end{bmatrix} +
\begin{bmatrix}
\Gamma_{11} & \Gamma_{12} & 0 \\
\Gamma_{21} & \Gamma_{22} & \Gamma_{23} \\
0 & \Gamma_{32} & \Gamma_{33}
\end{bmatrix}
\begin{bmatrix}
d_{off_1}(k) \\
d_{off_2}(k) \\
d_{off_3}(k)
\end{bmatrix}
\]

(2.9)

\[
\begin{bmatrix}
y_1(k) \\
y_2(k) \\
y_3(k)
\end{bmatrix} =
\begin{bmatrix}
C_{11} & 0 & 0 \\
0 & C_{22} & 0 \\
0 & 0 & C_{33}
\end{bmatrix}
\begin{bmatrix}
x_1(k) \\
x_2(k) \\
x_3(k)
\end{bmatrix} +
\begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}
\begin{bmatrix}
v_1(k) \\
v_2(k) \\
v_3(k)
\end{bmatrix}
\]

(2.10)

If we compare this state equation with the one reached in section 2.2.2 for the MIMO case, not taking the offtakes into account, we notice that the only difference is that we now split the before \( B \) matrix into matrices \( B \) and \( \Psi \). From a functional point of view, we are now dealing with
2. System Identification

SISO systems, with input $v_i$ and output $y_i$, that have accessible disturbances. These disturbances can either derive from the offtakes or from the inputs of the adjacent subsystems. The difference between inputs and accessible disturbances is important since only the inputs can be manipulated by the controllers, whereas the disturbances can either be random variables (offtakes) or dependent on factors external to the controller (inputs of the adjacent subsystems).

2.2.5 MISO system identification assuming actuator faults

Similarly to the procedure followed for the MIMO case, identification is carried out using data from the SIMULINK model of the canal, while keeping gate 2 at a height of $6 \text{ cm}$. The comparison between the validation data set and the ARX model achieved is shown in figure 2.12 and the orders that lead to these fits are the following

\[ N_{A_1} = 2, \quad N_{B_1} = [4 \ 1 \ 4 \ 4], \quad N_{K_1} = [1 \ 1 \ 1 \ 1], \]
\[ N_{A_3} = 3, \quad N_{B_3} = [4 \ 1 \ 4 \ 1], \quad N_{K_3} = [1 \ 1 \ 1 \ 1]. \]

![Comparison between data and model response](image)

Figure 2.12: Comparison between the data collected from the open-loop simulation of the SIMULINK model and the response of the ARX models, after filtering the data.
3 LQG Control

Contents

3.1 LQG controller ................................................................. 25
3.2 Linear-Quadratic Regulator .................................................. 25
3.3 Kalman Filter ................................................................. 27
3.4 Separation principle ......................................................... 28
3.5 Single gate LQG controller .................................................. 29
3. LQG Control
In this chapter the LQG controller is presented and a single gate example is used to study the influence of each of the controller parameters in the water level response.

3.1 LQG controller

The LQG controller consists in the combination of a Kalman filter, i.e., a Linear-Quadratic Estimator (LQE) with a Linear-Quadratic Regulator (LQR). The separation principle ensures that both can be designed and computed independently.

The backbone of both the centralized and distributed controllers is the LQG structure which is now explained. Later, we will consider an extension of LQG to include feedforward from accessible disturbances.

3.2 Linear-Quadratic Regulator

Consider a Linear Time Invariant (LTI) plant described by the state-space representation

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

and output equation

$$ y(k) = Cx(k) , $$

where $k$ is a nonnegative integer that represents discrete-time, $x \in \mathbb{R}^n$ is the state vector, $v \in \mathbb{R}^{n_v}$ is the input vector, with $n_v$ the number of inputs, $y \in \mathbb{R}^{n_y}$ is the output vector, with $n_y$ the number of outputs, and $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times n_v}$, $C \in \mathbb{R}^{n_y \times n}$ are matrices that define the parametrization of the model.

The linear-quadratic regulator problem is solved by calculating the optimal gain matrix $K$ such that the state-feedback law $u(k) = -Kx(k)$ minimizes the steady state quadratic cost

$$ J = \frac{1}{2} \sum_{k=1}^{\infty} \left[ x^T(k)Qx(k) + v^T(k)Rv(k) \right], $$

where $Q \in \mathbb{R}^{n \times n}$ is chosen such that $Q = C^T C$ and $R \in \mathbb{R}^{n_v \times n_v}$ is a positive definite matrix, here assumed to be diagonal, for the multivariable case, or simply a positive scalar constant, $\rho$, for the single gate case.

The optimal state feedback gain $K$ is given by the expression

$$ K = (I + R^{-1}B^T S B)^{-1}R^{-1}B^T S A , $$

in which $S$ is the positive definite solution of the algebraic Riccati equation

$$ S = A^T S[I + BR^{-1}B^T S]^{-1}A + Q. $$
In order to assure that the system response follows a reference, integral action must be taken into account in the controller design. Figure 3.1 shows the schematic of the controlled system including integral action. \( T_s \) stands for the sample time.

The plant in figure 3.1 is described by the dynamics in (3.1) and (3.2). The integrator in figure 3.1 is described by the equation

\[
x_I(k) = T_s q^{-1} e(k) \iff x_I(k + 1) = x_I(k) + T_s e(k),
\]

where \( e(k) = r(k) - y(k) = r(k) - Cx(k) \), with \( r(k) \) being the desired reference. Since we are designing a regulator, we consider \( r(k) = 0 \) and the state-space dynamics of the system composed by the plant and the regulator can be written as

\[
x(k + 1) = \bar{A} \bar{x}(k) + \bar{B}v(k),
\]

in which

\[
\bar{x} = \begin{bmatrix} x \\ x_I \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} A & 0 \\ -T_s C & I \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}
\]

where \( I_{ny \times ny} \), \( ny \) the number of outputs of the process, stands for the identity matrix and \( T_s \) is the sample time. We can also write the output of the process in the augmented form as

\[
y(k) = \bar{C} \bar{x}(k),
\]

with \( \bar{C} = [C \ 0] \), and the output of the regulator as

\[
v(k) = -[K x \ K_I] \begin{bmatrix} x(k) \\ x_I(k) \end{bmatrix},
\]

where

\[
[K x \ K_I] = (I + R^{-1} \bar{B}^\top \bar{S})^{-1} R^{-1} \bar{B}^\top \bar{S} \bar{A},
\]

with

\[
\bar{S} = \bar{A}^\top \bar{S}(I + BR^{-1} \bar{B}^\top \bar{S})^{-1} \bar{A} + Q.
\]

If we replace equation (3.9) in (3.7), we obtain the LQR state equation

\[
\begin{bmatrix} x(k + 1) \\ x_I(k + 1) \end{bmatrix} = \begin{bmatrix} A - BK_x & -BK_I \\ -T_s C & I \end{bmatrix} \begin{bmatrix} x(k) \\ x_I(k) \end{bmatrix},
\]

considering full access to the state.
However, since the observability matrix associated to the pair \((A, C)\), given by
\[
O[A, C] = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} C & 0 \\ CA & 0 \end{bmatrix},
\]
does not have full rank, the model described by equation (3.7) is unobservable, even assuming
\((A, C)\) to be observable. Therefore, the quadratic cost in equation (3.3) no longer applies. The new
cost function has an additional term that depends on the integral state, \(x_I\), and it is given by the
expression
\[
J = \frac{1}{2} \sum_{k=1}^{\infty} \left[ x^T(k)Qx(k) + x_I^T(k)Q_Ix_I(k) + v^T(k)Rv(k) \right],
\]
in which \(Q_I[n_y \times n_y]\) is chosen to be the identity matrix.

If we group the states \(x\) and \(x_I\) in equation (3.14) in one vector, we obtain an expression for
the quadratic cost closer to (3.3), which is
\[
J = \frac{1}{2} \sum_{k=1}^{\infty} \left[ \bar{x}^T(k)Q\bar{x}(k) + v^T(k)Rv(k) \right],
\]
where
\[
\bar{x} = \begin{bmatrix} x \\ x_I \end{bmatrix}, \quad Q = \begin{bmatrix} Q & 0 \\ 0 & I \end{bmatrix}.
\]

### 3.3 Kalman Filter

Most of the times, however, it is not possible to have access to the state. Therefore, there is
the need to estimate it, so that the control law mentioned in section 3.2 can be applied.

The estimator to be used is the Kalman filter, which is an optimal linear estimator, in the sense
that it optimizes the signal-to-noise ratio of the model. This is accomplished by finding an optimal
gain matrix \(M\) that places the estimator poles conveniently.

Consider the discrete plant
\[
x(k+1) = Ax(k) + Bv(k) + Gw(k)
\]
with measurements
\[
y(k) = Cx(k) + v_n(k),
\]
where the process noise \(w(k)\) and measurement noise \(v_n(k)\) are random white noise sequences
with zero mean, i.e., such that
\[
E[w(k)] = E[v_n(k)] = 0,
\]
have no time correlation, that is
\[
E[w(i)w^T(j)] = E[v_n(i)v_n^T(j)] = 0, \quad i \neq j,
\]
and have covariance matrices defined by
\[
Q_n = E[w(k)w^T(k)], \quad R_n = E[v_n(k)v_n^T(k)].
\]
The Kalman filter estimates the state based on the equations

\[
\dot{x}(k|k-1) = Ax(k-1|k-1) + Bu(k-1),
\]

\[
\dot{x}(k|k) = \dot{x}(k|k-1) + M\left[y(k) - C\dot{x}(k|k-1)\right],
\]

where \( M \) is the optimal gain matrix given by

\[
M = PC^T(CPC^T + R)^{-1},
\]

with \( P \) the positive definite solution of the algebraic Riccati equation

\[
P = A^TP[I + BR^{-1}B^TP]^{-1}A + Q_n.
\]

### 3.4 Separation principle

As shown in section 3.2, the system composed by the process and the regulator is not observable. This means that it is not possible to estimate the state vector \( \bar{x} \). Instead, we only estimate \( x \) and use the true value of \( x_I \). In order for us to be able to design the regulator and the kalman filter separately, we need to guarantee that the Separation Principle still holds, even though not all states are being estimated.

We begin by defining the estimation error \( \tilde{x}(k) \) as

\[
\tilde{x}(k) = x(k) - \hat{x}(k|k),
\]

where \( \tilde{x}(k|k) \), by joining equations (3.17) and (3.18), can be written as a function of \( \dot{x}(k-1|k-1) \), \( v(k-1) \) and \( y(k) \) as

\[
\dot{x}(k|k) = (I - MC)A\dot{x}(k-1|k-1) + (I - MC)Bu(k-1) + My(k).
\]

By replacing \( x(k) \) and \( \dot{x}(k|k) \) in equation (3.21) by (3.1) and (3.22), respectively, we obtain

\[
\tilde{x}(k) = (I - MC)A\tilde{x}(k-1).
\]

If we only estimate \( x(k) \) but not \( x_I(k) \), the control law becomes

\[
v(k) = - \begin{bmatrix} K_x & K_I \end{bmatrix} \begin{bmatrix} \dot{x}(k|k) \\ x_I(k) \end{bmatrix},
\]

which we can replace in equation (3.1) to get

\[
x(k+1) = Ax(k) - BK_x\dot{x}(k|k) - BK_Ix_I(k).
\]

So as to write the state \( x(k) \) as a function of the estimation error \( \tilde{x}(k) \), we replace \( \dot{x}(k|k) \) by \( x(k) - \tilde{x}(k) \) to obtain

\[
x(k+1) = (A - BK_x)x(k) + BK_x\tilde{x}(k) - BK_Ix_I(k).
\]
If we compress all three state equations (3.26), (3.6) and (3.23) into one equation in matrix form, we get

\[
\begin{bmatrix}
    x(k+1) \\
    x_1(k+1) \\
    \bar{x}(k+1)
\end{bmatrix} =
\begin{bmatrix}
    A - BK_x & -BK_I & BK_x \\
    -T_x C & I & 0 \\
    0 & 0 & (I - MC)A
\end{bmatrix}
\begin{bmatrix}
    x(k) \\
    x_1(k) \\
    \bar{x}(k)
\end{bmatrix}.
\]

(3.27)

If we compare equation (3.27) with (3.12), we reach the conclusion that the matrix that relates \(\bar{x}(k+1)\) and \(\bar{x}(k)\) is the same. Since

\[
det \left( sI - \begin{bmatrix}
    A - BK_x & -BK_I & BK_x \\
    -T_x C & I & 0 \\
    0 & 0 & (I - MC)A
\end{bmatrix} \right) =
\]

\[
det \left( sI - \begin{bmatrix}
    A - BK_x & -BK_I \\
    -T_x C & I \\
    0 & 0
\end{bmatrix} \right) \cdot \det \left( sI - (I - MC)A \right),
\]

this proves that even though \(x_1\) is not estimated, the Separation Principle is still verified and the controller and the estimator can be designed separately.

3.5 Single gate LQG controller

Assuming full access to the state, in order to design a LQR, i.e., in order to find the optimal gain matrix \(K\), mentioned in section 3.2, the value of the parameter \(\rho\) needs to be chosen.

Figure 3.3 shows the effect of the variation of the parameter \(\rho\) on the SIMULINK nonlinear model response for the case that only pool 1 is taken into account. It is clear from the figure that the higher the value of \(\rho\), the slower the model responds. The value sought is the one that allows the model to respond as fast as possible, without leading to oscillations or significant overshoot. Therefore, the value chosen for this case is \(\rho = 10^3\).

Once the LQR is designed, the estimator becomes the main focus. As seen in section 3.3, the Kalman filter makes use of two covariance matrices that need to be determined: \(R_n\) and \(Q_n\). The first has the same influence on the response of the estimator as \(\rho\) in the LQR response and it
3. LQG Control

Figure 3.3: Closed-loop response of the controlled SIMULINK model, only taking into account the water level of pool 1 and the position of gate 1, for different values of $\rho$, considering full access to the state. Gates 2, 3 and 4 are kept at the positions 6.01 cm, 6.09 cm and 28.77 cm, respectively.

Also becomes a positive constant in the single gate case, designated by $r_n$. As to the covariance matrix $Q_n$, it can be settled as $q^2BB^T$, where $q$ is a positive constant. To study the influence of each of these parameters in the nonlinear model response, the controlled SIMULINK model is simulated, keeping $\rho = 10^3$. First, the value of $q$ is kept at $q = 1$ and the influence of $r_n$ is studied. Figure 3.4 shows the result of the controlled SIMULINK model simulation, in these conditions.

Figure 3.4: Closed-loop response of the controlled SIMULINK model, only taking into account the water level of pool 1 and the position of gate 1, with estimated state, for different values of $r_n$, while keeping $\rho = 10^3$ and $q = 1$. Gates 2, 3 and 4 are kept at the positions 6.01 cm, 6.09 cm and 28.77 cm, respectively.

From figure 3.4 we can see that the lower the value of $r_n$, the higher the overshoot and the more significant the oscillation becomes. Similar to what happened with $\rho$ in the LQR case, $r_n$ determines the speed of the estimator, which means that the lower the value of $r_n$, the faster the estimator, which leads to higher overshoot and oscillation. However, the difference between the curves obtained for $r_n = 10$ and $r_n = 10^2$ is much less evident when compared to the difference obtained with $r_n = 1$ and $r_n = 10$. In fact, for higher values of $r_n$, the difference is barely detectable. Therefore, the value chosen is $r_n = 10^2$.

To study the influence of parameter $q$ in the nonlinear model response, the value of $r_n$ is now kept constant at the previously chosen value, $r_n = 10^2$. The simulation of the SIMULINK model
is shown in figure 3.5. From this figure, we can see that increasing the value of $q$ up to $10^2$, or higher, leads to oscillatory behavior. The value of $q$ to be chosen must be the highest that does not lead to overshoot or oscillations, which from figure 3.5 we conclude to be $q = 1$.

Figure 3.5: Closed-loop response of the controlled SIMULINK model, only taking into account the water level of pool 1 and the position of gate 1, with estimated state, for different values of $q$, while keeping $\rho = 10^3$ and $r_n = 10^2$. Gates 2, 3 and 4 are kept at the positions 6.01 cm, 6.09 cm and 28.77 cm, respectively.

### 3.5.1 Constraining closed-loop poles

In order to try to overcome the problems of overshoot and oscillatory behaviors just described, typical of LQG controllers, it is also possible to constrain the poles to a circle of radius lower than 1. This can be accomplished by minimizing the cost function

$$J = \sum_{k=1}^{\infty} \left[ x^T(k)Qx(k) + v^T(k)Rv(k) \right]e^{2k},$$

in that $1/\alpha$, $\alpha > 1$, is the new circle radius. The corresponding state-space dynamics is now

$$x(k + 1) = \alpha Ax(k) + \alpha Bv(k),$$

as shown in [15].

Figure 3.6 shows the influence of $\alpha$ on the SIMULINK nonlinear model response, while keeping $\rho = 10^3$, $r_n = 10^2$ and $q = 1$. As observed, the model response is very sensitive to and largely affected by $\alpha$ and for values above 1.005, it no longer converges to the reference value. The value of $\alpha$ to be used is $\alpha = 1.005$, since that is the highest value that allows the response to follow a reference.

Figure 3.7 shows the controlled SIMULINK model time response for pool 1, when the controller parameters assume the previously computed values: $\rho = 10^3$, $r_n = 10^2$, $q = 1$ and $\alpha = 1.005$. We see that the model response follows the desired reference and it presents a settling time of approximately 106 s, which is less than 2 min. In the context of the water delivery canal under study this is still a low value. As we will see, actual experimental settling time values lay between 5 min. and 10 min.
3. LQG Control

Figure 3.6: Closed-loop response of the controlled SIMULINK model, only taking into account the water level of pool 1 and the position of gate 1, with estimated state, for different values of $\alpha$, while keeping $\rho = 10^3$, $r_n = 10^2$ and $q = 1$. Gates 2, 3 and 4 are kept at the positions 6.01 cm, 6.09 cm and 28.77 cm, respectively.

Figure 3.7: Closed-loop response of the controlled SIMULINK model, only taking into account the water level of pool 1 and the position of gate 1. The LQG parameters are $\rho = 10^3$, $r_n = 10^2$, $q = 1$ and $\alpha = 1.005$. Gates 2, 3 and 4 are kept at the positions 6.01 cm, 6.09 cm and 28.77 cm, respectively.
4 Actuator Faults

Contents

4.1 Centralized control tolerant to actuator faults ..................... 35
4.2 Distributed control tolerant to actuator faults ..................... 45
4. Actuator Faults
In this chapter the design of both centralized and distributed control systems tolerant to actuator faults is explained, enhancing the issues of fault detection and control reconfiguration. The results of some illustrative simulations of the nonlinear model of the canal are presented.

4.1 Centralized control tolerant to actuator faults

As already mentioned in section 2.2.3, the centralized FTC system to consider consists of two controllers, one for the canal in normal operation, $C_N$, and another one for the case that a fault in the actuator of gate 2 occurs, $C_F$. The first controller has three inputs, which are the variables proportional to the flow drained by each gate 1, 2 and 3, $v_1$, $v_2$ and $v_3$, and three outputs, which are the downstream water levels of pools 1, 2 and 3, $y_1$, $y_2$ and $y_3$. As the actuator of gate 2 stops functioning, the position of this gate remains constant and a fault is detected by an algorithm conceived for that purpose. After that, the control of the canal is ensured by the second controller, which only has two inputs, $v_1$ and $v_3$, since the information of the position of gate 2 was lost, and two outputs, $y_1$ and $y_3$. These two controllers are designed in sections 4.1.1 and 4.1.2.

Figure 4.1 illustrates the control system tolerant to actuator faults described. The fault detector operation is explained in section 4.1.3.

![Figure 4.1: Reconfigurable controller with actuator fault detection. The integrator is included in the blocks that represent the controllers. Block $u = f(v)$ represents equation (2.1), which relates the gate position with the flow passing under it.](image)

4.1.1 Three gate LQG controller

When all gates are operating normally, i.e., assuming no fault has occurred, a multivariable 3 x 3 controller is responsible for controlling the water levels of pools 1 to 3, by manipulation of the respective gate positions. Figure 4.2 shows the interaction of such a controller with the canal.

The three gate LQG controller design is similar to that of the single gate one, with the difference that in the multivariable case $R$ and $R_u$ become $n_y \times n_y$ matrices, with $n_y$ the number of outputs.
Taking into account that three outputs are considered, these matrices can be written as

\[ R = \begin{bmatrix} \rho_1 & 0 & 0 \\ 0 & \rho_2 & 0 \\ 0 & 0 & \rho_3 \end{bmatrix}, \quad R_n = \begin{bmatrix} r_{n_1} & 0 & 0 \\ 0 & r_{n_2} & 0 \\ 0 & 0 & r_{n_3} \end{bmatrix}. \]

For the sake of simplicity, all entries of matrices $R$ and $R_n$ are assumed to have the same value.

The best design option for the three gate LQG controller is the one that makes use of the values reached in section 3.5, i.e., $\rho_i = 10^3$, $r_{n_i} = 10^2$, $i = 1, 2, 3$, $q = 1$ and $\alpha = 1.005$.

Figure 4.3 shows the closed-loop response of the SIMULINK model, for the first three pools. In this simulation variations are made not only in the references but also in the offtake flows, the latter changing between 0 and 5 l/s. The responses observed are similar to the one obtained in section 3.5 and the settling time is also around 2 min.

If we first turn our attention to the gate positions plot in figure 4.3, we notice that whereas the position of gate 1 changes mostly in response to variations in reference 1 and in the flow drawn by offtake 1, the position of gate 3 changes each time any reference or any offtake flow varies. It so happens that each gate position influences mainly the pools upstream of it, and that is why the position of gate 2 also changes mostly in response to variations in the references and offtake flows of pools 1 and 2. In fact, if we look at the time evolution of the position of gate 1, we can only distinguish two set points around which the position varies: one when offtake 1 is not draining water and another one when it is. As to the position of gate 2, we differentiate three levels: when both offtakes 1 and 2 are switched off, when one is on and the other is off and when both are on. Finally, for gate 3 we find a set point for each combination between the offtakes of the three pools. This can be explained intuitively, inasmuch as all gate positions only depend on the flow drained to the pool immediately upstream. Since the flow drained to pool 1 is only determined by the intake flow, which is constant, pool 1 is barely affected by the other pools. Following the same logic, the flow drawn to pool 2 depends not only on the intake flow but also on changes in the position of gate 1 and offtake flow of pool 1, and the same happens to pool 3, which depends on both upstream pools.
4.1 Centralized control tolerant to actuator faults

Water levels

Gate positions

Offtake flows

Figure 4.3: Closed-loop response of the controlled SIMULINK model, using centralized control. Only the water levels of pools 1, 2 and 3 and the positions of the respective gates are taken into account, while gate 4 is kept at the position of $28.77 \text{ cm}$. The controller parameters are $\rho_i = 10^3$, $r_n = 10^2$, $q = 1$ and $\alpha = 1.005$. 
4. Actuator Faults

If we now focus our attention, for instance, on time instant $4000 \text{ s}$, we come across the effect of the centralized control, since all gates react to a change in the reference of pool 1. We also notice that the gates reaction decreases intensity from gate 1 to gate 3, as expected.

By looking at the time evolution of the water levels, we see that they follow the respective reference, meaning that the controller responds to the disturbances caused by the offtakes. It is evident, however, that, especially when offtake $i$ is switched on, the water level of pool $i$ deviates slightly from the desired value, but also that it is rapidly restored.

4.1.2 Two gate LQG controller

After a fault is detected in gate 2, the water level of pool 2 is no longer controlled, since the correspondent gate can not be manipulated. The controller for this faulty condition is a $2 \times 2$ multivariable controller that manipulates gates 1 and 3 in order to control the water levels of pools 1 and 3. The interaction between this controller and the canal is shown in figure 4.4.

![Multivariable controller after a fault in gate 2 is detected.](image)

Since we now consider a two gate controller, matrices $R$ and $R_n$ become

$$R^F = \begin{bmatrix} \rho_1^F & 0 \\ 0 & \rho_2^F \end{bmatrix}, \quad R_n^F = \begin{bmatrix} r_{n_1}^F & 0 \\ 0 & r_{n_2}^F \end{bmatrix},$$

and once again, for the sake of simplicity, we consider $\rho_1^F = \rho_2^F$ and $r_{n_1}^F = r_{n_2}^F$.

The parameters of the LQG controller used to control the canal in case of the occurrence of a fault in gate 2 assume the same values as in section 4.1.1, which are $\rho_i = 10^3$, $r_{n_i} = 10^2$, $i = 1, 3$, $q = 1$ and $\alpha = 1.005$. For lower values of $\rho$ the SIMULINK model response becomes oscillatory.

Figure 4.5 shows the simulation of the controlled SIMULINK model, when gate 2 is kept constant at a height of $6 \text{ cm}$. From figure 4.5 we notice that the water levels of pools 1 and 3 follow the respective reference, even though the same does not happen in pool 2, as expected. As in section 4.1.1, we see that when the flow drawn by an offtake varies, the water level of the correspondent pool deviates slightly from the desired value, just to return to it immediately after. We can again differentiate two
4.1 Centralized control tolerant to actuator faults

Figure 4.5: Closed-loop response of the controlled SIMULINK model, using centralized control, when gate 2 is kept at the position of 6 cm. Only the water levels of pools 1, 2 and 3 and the position of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controllers parameters are $\rho_i = 10^3$, $r_{ni} = 10^2$, $q = 1$ and $\alpha = 1.005$. 
4. Actuator Faults

set points in the evolution of the position of gate 1, around which the position varies, one when offtake 1 is on and the other when it is off. As to gate 3, its position varies around several set points, which switch between each other every time a variation occurs in any offtake flow. Since gate 2 is stuck, it is in the water level of pool 2 that these several set points appear. Whenever a change in the offtake flows of pools 1 and 2 occurs, the water level of pool 2 rises or decreases, depending on whether water is being added to or removed from the pool, respectively. On the other hand, changes in offtake 3 do not interfere with the water level of pool 2. Regardless of the set point, the water level of pool 2 follows the behavior of that of pool 3, since it only responds to changes in the position of gate 3 as well.

We can still notice, when offtakes 1 and 2 are turned off, from time instant \( t = 1.85 \times 10^4 \) s on, that the water level of pool 2 follows the respective reference. This occurs only because gate 2 is at a height of 6 cm, which, in this case, is the set point when none of the offtakes is draining water. If, instead, it is kept at a higher position, the set point of the water level of pool 2 decreases and, if gate 2 is fully opened, pools 2 and 3 are considered to be part of the same pool, with similar water levels. On the other hand, if gate 2 is kept at a position lower than 6 cm, the water level of pool 2 rises above the reference and, for positions below 3 cm, the water overflows. This study can be found in [26].

4.1.3 Actuator Fault Detection

Initially the canal is controlled by the three gate controller designed in section 4.1.1, but as a fault occurs, it is necessary to switch to the two gate controller designed in section 4.1.2. With this in mind, we first need a fault detector, which is able to detect a fault in gate 2 as fast as possible, while ignoring false alarms.

Since in real experiments the command sent to the gate and the actual gate position are not equal most of the time, due to measurements oscillations, the fault detection criterion cannot be based on the fact that the difference between those two values has to be zero. In order for the control system to detect the occurrence of a fault, the following three step algorithm was developed: First, for each gate \( i \), \( i = 1, 2, 3 \), an error, \( \tilde{u}_i \), between the command sent to the gate, \( u_i \), and the actual gate position, \( u_{r,i} \), is defined as

\[
\tilde{u}_i(k) = u_i(k) - u_{r,i}(k).
\]  

(4.1)

After that, a performance index, \( \Pi \), can be obtained from this error by

\[
\Pi(k) = \gamma \Pi(k-1) + (1 - \gamma) |\tilde{u}_i(k)|
\]  

(4.2)

A fault is detected whenever \( \Pi(k) \geq \Pi_{max} \), where \( \Pi_{max} \) is a chosen threshold.

A block diagram that illustrates this three step algorithm can be found in figure 4.1, incorporated in the control system tolerant to actuator faults diagram.
4.1 Centralized control tolerant to actuator faults

Figure 4.6 shows the time evolution of $\Pi$ and the threshold (above) and a fault indicator, $I_F$ (below), for an experimental example (which is presented in the experimental results section, in figure 6.1). The vertical green line indicates that a fault has occurred. The fault indicator equals one if a fault has been detected and equals zero otherwise. In the present experiment, the values of $\Pi_{max}$ and $\gamma$ chosen are $\Pi_{max} = 5$ and $\gamma = 0.95$ and the fault was detected after approximately two minutes.

![Figure 4.6: Time evolution of $\Pi$ (blue), $\Pi_{max}$ (red) and fault occurrence (green) above. Fault detection indicator below.](image)

The values of $\gamma$ and $\Pi_{max}$ chosen must be such that the fault is detected as fast as possible and without the occurrence of false alarms, that is, when a fault is detected but no fault has occurred. Figure 4.7 shows the effect of the variation of $\gamma$ and $\Pi_{max}$ on the number of false alarms and on the time elapsed since a fault occurred until it was detected (hereafter denoted as detection time), for the experiment of figure 6.1. Plots (a) and (b) show the number of false alarms and the detection time as a function of $\gamma$, respectively, while keeping $\Pi_{max} = 5$. In plot (c) we see the number of false alarms as a function of the detection time, when $\gamma$ varies between 0 and 1, while $\Pi_{max} = 5$. When, instead of varying $\gamma$ we vary the threshold $\Pi_{max}$, while keeping $\gamma = 0.95$, we obtain plots (d), (e) and (f).

From plot (a) we conclude that the higher the value of $\gamma$ the fewer the number of false alarms we obtain. The first value of $\gamma$ that ensures that no false alarms occur for this experiment is $\gamma = 0.913$. Now switching our attention to plot (b), we notice that the detection time decreases until $\gamma = 0.913$, after what it starts increasing slowly until $\gamma \approx 0.96$ and afterwards it increases sharply towards infinity. In fact, when $\gamma = 1$, the performance index $\Pi$ equals zero, which means that the fault is never detected and that no false alarms occur. From plot (c) we see that the point with the lowest detection time and the lowest number of false alarms corresponds to $\gamma = 0.913$, which would then be the best choice for this experiment. In order to assure that no false alarms occur for other experiments we need to choose a higher value of $\gamma$, taking into account that it should be lower than 0.96, so that the fault is detected in a reasonable period of time. Therefore, the value chosen was $\gamma = 0.95$. 
4. Actuator Faults

Figure 4.7: Effects of the variation of $\gamma$ on the number of false alarms and the time for detection, while keeping $\Pi_{\text{max}} = 5$ (a, b, c), and effects of the variation of $\Pi_{\text{max}}$ on the number of false alarms and the time for detection, while keeping $\gamma = 0.95$ (d, e, f).

Following the same reasoning, from plot (d) we notice that the minimum threshold that ensures that no false alarms occur for this experiment is 4.58. As to the detection time, in plot (e) we see that it is minimum for that same value of $\gamma$, after what it increases slightly until $\Pi_{\text{max}} \approx 7$ and afterwards it increases sharply from less than 2 min to 47 min. In plot (f) the value of the threshold that corresponds to the lower detection time and no false alarms is 4.58. The chosen value, $\Pi_{\text{max}} = 5$, assures that no false alarms occur, while guaranteeing that the fault is detected in a reasonably short period of time.

4.1.4 Reconfiguration

The reconfiguration process follows the discrete state diagram presented in figure 4.8. State $S_1$ corresponds to the case in which all gates operate normally with the three gate controller designed, $C_N$. Upon the occurrence of a fault, the system state switched to $S_2$, in which gate 2 is blocked but the fault has not been detected yet and the controller in operation is still $C_N$. When the fault is detected, the state switches to $S_3$, where the controller in operation becomes the two-gate controller, $C_F$. When the fault is recovered, i.e., gate 2 returns to its normal operation, the state returns to $S_1$.

In order to avoid instability that might arise due to fast switching, a dwell time condition is imposed. This condition implies that, once a controller is applied to the plant, it will remain so for
at least a minimum time period (dwell time).

![Figure 4.8: Discrete states in controller reconfiguration.](image)

Figure 4.8 shows a simulation of the SIMULINK model controlled by the control system tolerant to actuator faults designed, where a fault occurs at time instant 8900 s. Since in the SIMULINK model the commands sent to the gate actuators become their actual positions, the fault is detected as soon as the actual value of the gate position is different from the command sent to it. This means that a fault is detected immediately after its occurrence.

If we attend to the three plots in figure 4.9, we notice that until time instant $1.4 \times 10^4$ s the responses behave as we expected from what was seen in sections 4.1.1 and 4.1.2: until the occurrence of the fault, all responses react rapidly to the disturbances derived from the offtake; when the fault is detected, the transition between controllers generates an undesirable oscillation, which disappears immediately after, and the water levels are restored to the aimed values; since gate 2 stopped moving at a position lower than the current set point (which at that instant was nearly 4 cm), the water level of pool 2 follows the behavior of that of pool 3 at a higher level than its reference.

After instant $1.4 \times 10^4$ s, offtake flows variations lead to much more significant disturbances. First, the opening of offtake 3 just causes the water level of pool 2 to decrease. After that, however, when offtake 2 is closed, the water level of pool 2 rises continuously and it does not overflow only because offtake 2 is switched on again. The same happens when offtake 1 is closed. On the other hand, at the closure of offtake 3, the water level of pool 2 does not suffer changes, since, as seen before, it does not depend on pool 3.

At time instant $2 \times 10^4$ s, when reference 1 decreases, in order for the water level of pool 1 to continue following it, gate 1 opens completely and the water level follows beneath the desired value. This is due to the fact that, inasmuch as offtake 1 had been switched off short before, the water levels of pools 1 and 2 were both rising at that time and, since the water level of pool 2 was at a higher value than that of pool 1, the water level of pool 1 did not follow the reference immediately. This caused gate 1 to overreact and fully open. We notice that the water level of pool 1 only reaches the respective reference when the water level of pool 2 decreases below that
4. Actuator Faults

Figure 4.9: Closed-loop response of the controlled SIMULINK model, using centralized control, when gate 2 remains at a position of nearly 4 cm after time instant 8900 s. Only the water levels of pools 1, 2 and 3 and the position of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controllers parameters are $\rho_i = 10^3$, $r_{ni} = 10^2$, $q = 1$ and $\alpha = 1.005$. 
4.2 Distributed control tolerant to actuator faults

This means that, as a fault occurs, several situations can lead to water overflow in pools 1 and 2, a risk that arises from gate 2 remaining still at a position lower than the current set point. We should notice that a variation of around 8 mm in a gate position leads to a variation of 10 cm in the water level of the upstream pool and, consequently, gate 2 getting stuck at a position slightly lower than the set point leads to a significant rise in the water level of pool 2. Even if gate 2 stops moving at the set point, if all offtakes are switched on before the fault occurs, switching them back off afterwards still causes an overflow.

Another limitation seen is that the water levels only follow the respective reference inasmuch as the water levels of their neighbors allow it, i.e., as long as the water level of the upstream neighbor is not lower than its own and that of the downstream neighbor is not higher.

Bumpless transfer between controllers

The oscillation caused by the transition between controllers is due to the fact that the two controllers are based on different models and it is, therefore, expected that their outputs at each time instant may not the same. A way to force the output of the controller $C_F$ to be the same as that of $C_N$ consists in initializing the integrator of the controller $C_F$ such as to compensate the difference between the present gate position and the contribution to control given by state feedback. By solving equation (3.24) with respect to $x_I(k)$, we obtain the integrator state as a function of the the real flow under the gate, $v_r(k)$, and the state feedback

$$x_f^F(k) = -K_f^{F-1}v_r(k) + K_f^{F} x_F^F(k|k).$$

(4.3)

If we feedback the real gate position values to the integrator of the fault controller, $C_F$, as shown in figure 4.10, we can write the integrator state as the following piecewise-defined equation

$$x_f^F(k) = \begin{cases} 
-K_f^{F-1}v_r(k) + K_f^{F} x_F^F(k|k) & \text{if } k < k_d, \\
 x_f^F(k-1) + T_s e(k-1) & \text{if } k \geq k_d,
\end{cases}$$

where $k_d$ is the time at which the fault is detected and the superscript $F$ enhances the fact that the integrator belongs to controller $C_F$. The second expression was seen in equation (3.6).

Figure 4.11 shows the same simulation as the one seen in figure 4.9, but now considering the integrator state initialization described. The transition between controllers is no longer noticeable; the water levels follow the respective reference as if the same controller is operating.

4.2 Distributed control tolerant to actuator faults

4.2.1 Distributed LQ control with accessible disturbances

As already mentioned in section 2.2.4, in a distributed framework, a controller is designed for each pool. Each controller receives the information of the water level, real gate position and
4. Actuator Faults

![Block diagram of the canal subject to the fault controller. A bumpless transition is accomplished by initializing the integrator such that the controller has as output the last gate position values read from the canal.](image)

The state space model we now consider has two extra terms when compared to that in equation (3.1), one that concerns the coordination between controller and another one that concerns the offtakes, and it is given by

$$x(k + 1) = Ax(k) + Bv(k) + \Psi d_{\text{gate}}(k) + \Gamma d_{\text{off}}(k),$$

(4.4)

where $x(k) \in \mathbb{R}^n$ is the state, $v(k) \in \mathbb{R}$ is the pool input, $d_{\text{gate}}(k) \in \mathbb{R}^m$, $m$ the number of pools considered, is the vector of the inputs of all pools considered, $d_{\text{off}}(k) \in \mathbb{R}^m$ is the vector of the accessible disturbance derived from the offtakes of all the pools considered, $A[n \times n]$ and $B[n \times 1]$ are matrices that model the input/output dynamics of the pool and $\Psi[n \times m]$ and $\Gamma[n \times m]$ are matrices that model the influence of the neighbors inputs and offtakes, respectively, on the considered pool. If we attend to equation (2.10), each line of the matrices that contain the $\Psi_{ij}$ and $\Gamma_{ij}$ elements correspond to matrices $\Psi$ and $\Gamma$ in equation (4.4).

The output equation is the same as in equation (3.2).

We now associate to this plant the quadratic cost from equation (3.3), which we intend to minimize, assuming knowledge of the accessible disturbances, at each time instant $k$.

Since we now have two additional terms, the solution to the problem is different from the one seen in section 3.2 and it is obtained by applying the Pontryagin’s Minimum Principle in discrete time, presented in appendix B. Both the outputs of the controllers neighbors and the offtake flows are treated as accessible disturbances.
4.2 Distributed control tolerant to actuator faults

Figure 4.11: Closed-loop response of the controlled SIMULINK model, using centralized control, when gate 2 remains at a position of nearly 4 cm after time instant 8900 s. Only the water levels of pools 1, 2 and 3 and the position of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controllers parameters are $\rho_i = 10^3$, $r_n = 10^3$, $q = 1$ and $\alpha = 1.005$. Bumpless transition between controllers.
Comparing equation (3.15) with equation (B.2) we conclude that the Lagrangian function is

$$L(x(k), u(k), k) = -\frac{1}{2} [x^T(k)C^T C x(k) + v^T(k)Rv(k)] \quad (4.5)$$

and the Hamiltonian function in equation (B.3) is written as

$$H(k) = \lambda^T(k+1) [Ax(k) + Bu(k) + \Psi d_{\text{gate}}(k) + \Gamma d(k)]$$

$$- \frac{1}{2} [x^T(k)C^T C x(k) + v^T(k)Rv(k)] , \quad (4.6)$$

where $\lambda$ is the co-state. Therefore, the stationary condition (B.5) is given by

$$\frac{\partial H(k)}{\partial v(k)} = \lambda^T(k+1)B - Rv(k) = 0 , \quad (4.7)$$

which, resolving with respect to $u$, leads to the optimal control expression

$$v(k) = R^{-1}B^T \lambda(k+1) . \quad (4.8)$$

So as to write $v(k)$ only as a function of measured variables, we assume

$$\lambda(k) = -Px(k) + g , \quad (4.9)$$

where $P$ is a $n \times n$ matrix and $g$ a vector that depends on the accessible disturbances. In order for $P$ and $g$ to be constant, we consider that $N \to \infty$ and that the accessible disturbances vary slowly. If we replace $\lambda(k+1)$ in (4.8) by equation (4.9), we get

$$v(k) = -R^{-1}B^T PAx(k) + R^{-1}B^T g . \quad (4.10)$$

Since at time $k$ the value of $x(k+1)$ is unknown, we replace it by the expression in (4.4), which yields

$$v(k) = -R^{-1}B^T P[Ax(k) + Bu(k) + \Psi d_{\text{gate}}(k) + \Gamma d_{\text{off}}(k)] + R^{-1}B^T g \quad (4.11)$$

Solving the previous equation with respect to $v(k)$ results in

$$v(k) = -(I + R^{-1} B^T PB)^{-1} R^{-1} B^T PAx(k)$$

$$+ (I + R^{-1} B^T PB)^{-1} R^{-1} B^T [g - P\Psi d_{\text{gate}}(k) - P\Gamma d_{\text{off}}(k)] \quad (4.12)$$

If we define the optimal LQ gain $K$ by

$$K = (I + R^{-1} B^T PB)^{-1} R^{-1} B^T PA \quad (4.13)$$

Figure 4.12: Schematics of the Distributed LQG control applied to the canal.
4.2 Distributed control tolerant to actuator faults

and the feedforward control action $v_{ff}$ by

$$v_{ff}(k) = (I + R^{-1}B^TPB)^{-1}R^{-1}B^T[g - P\Psi d_{gate}(k) - P\Gamma d_{off}(k)],$$  \hspace{1cm} (4.14)

equation (4.12) can be written as

$$v(k) = -Kx(k) + v_{ff}.$$  \hspace{1cm} (4.15)

Whereas $K$ only depends on known parameters, $v_{ff}$ is expressed as a function of $g$ and $P$, which are still unspecified. So as to determine these parameters, while ensuring that equation (4.9) still holds, we begin by obtaining the co-state and the closed loop state equations. The adjoint equation, which depends on the partial derivative of $L$ in equation (4.5) with respect to $x$

$$\frac{\partial L(k)}{\partial x(k)} = -(Cx(k))^TC,$$  \hspace{1cm} (4.16)

in this case is given by

$$\lambda(k) = A^T\lambda(k+1) - C^TCx(k).$$  \hspace{1cm} (4.17)

If we replace $\lambda(k)$ by the expression given in (4.9), we get

$$-Px(k) + g = -A^TPx(k+1) + A^Tg - C^TCx(k).$$  \hspace{1cm} (4.18)

As to the closed loop state equation, it can be obtained by replacing $u(k)$ in equation (4.4) by the expression reached in equation (4.10), which yields

$$x(k+1) = Ax(k) + BR^{-1}B^T[-Px(k+1) + g] + \Psi d_{gate}(k) + \Gamma d_{off}(k).$$  \hspace{1cm} (4.19)

Solving with respect to $x(k+1)$ we obtain

$$x(k+1) = [I + BR^{-1}B^TP]^{-1}Ax(k) + [I + BR^{-1}B^TP]^{-1}[BR^{-1}B^Tg + \Psi d_{gate}(k) + \Gamma d_{off}(k)]$$  \hspace{1cm} (4.20)

and, returning to equation (4.18), if we replace $x(k+1)$ by this expression and group all the terms in $x(k)$, we get

$$\{ P - A^TP[I + BR^{-1}B^TP]^{-1}A - C^TC \} x(k) =$$

$$\{ I + A^TP[I + BR^{-1}B^TP]^{-1}BR^{-1}B^T - A^T \} g$$

$$+ A^TP[I + BR^{-1}B^TP]^{-1}[\Psi d_{gate}(k) + \Gamma d_{off}(k)].$$  \hspace{1cm} (4.21)

In order for equation (4.21) to hold for all $x$, $v$ and $d_{off}$, $P$ and $g$ must simultaneously satisfy the algebraic Riccati equation

$$P = A^TP[I + BR^{-1}B^TP]^{-1}A + C^TC$$  \hspace{1cm} (4.22)

and the condition on vector $g$

$$\{ I + A^TP[I + BR^{-1}B^TP]^{-1}BR^{-1}B^T - A^T \} g = -A^TP[I + BR^{-1}B^TP]^{-1}[\Psi d_{gate}(k) + \Gamma d_{off}(k)].$$  \hspace{1cm} (4.23)
4. Actuator Faults

This equation can be written in a more simplified form as

\[
g = k_g \begin{bmatrix} d_{\text{gate}}(k) \\ d_{\text{off}}(k) \end{bmatrix}, \tag{4.24}
\]

where

\[
k_g = - \left\{ I + A^T P[I + BR^{-1}B^T P]^{-1}BR^{-1}B^T - A^T \right\}^{-1} A^T P[I + BR^{-1}B^T P]^{-1} \begin{bmatrix} \Psi & \Gamma \end{bmatrix}.
\]

Once the vector \( g \) is specified, we can write \( v_{ff} \) in equation (4.14) as

\[
v_{ff}(k) = k_{ff} \begin{bmatrix} d_{\text{gate}}(k) \\ d_{\text{off}}(k) \end{bmatrix}, \tag{4.25}
\]

where

\[
k_{ff} = (I + R^{-1}B^T PB)^{-1}R^{-1}B^T \left( k_g - P[\Psi \ \Gamma] \right). \tag{4.26}
\]

In order to include integral action, as implemented in section 3.2, the state space model in equation (4.4) becomes

\[
\bar{x}(k + 1) = \bar{A}x(k) + \bar{B}v(k) + \bar{\Psi}d_{\text{gate}}(k) + \bar{\Gamma}d_{\text{off}}(k), \tag{4.27}
\]

where

\[
\bar{\Psi} = \begin{bmatrix} \Psi \\ 0 \end{bmatrix}, \quad \bar{\Gamma} = \begin{bmatrix} \Gamma \\ 0 \end{bmatrix},
\]

and the optimal control is given by

\[
v(k) = -K_x x(k) - K_I x_I(k) + K_{ff} \begin{bmatrix} d_{\text{gate}}(k) \\ d_{\text{off}}(k) \end{bmatrix}, \tag{4.28}
\]

where

\[
[K_x \ K_I] = (I + R^{-1}B^T \bar{S}B)^{-1}R^{-1}B^T \bar{S}A \tag{4.29}
\]

is the state feedback gain, including integral action, and

\[
K_{ff} = (I + R^{-1}B^T \bar{P}B)^{-1}R^{-1}B^T \left( k_g - \bar{P}[\bar{\Psi} \ \bar{\Gamma}] \right) \tag{4.30}
\]

is the feedforward gain, with

\[
K_g = - \left\{ I + \bar{A}^T \bar{P}[I + BR^{-1}B^T \bar{P}]^{-1}BR^{-1}B^T - \bar{A}^T \right\}^{-1} \bar{A}^T P[I + BR^{-1}B^T \bar{P}]^{-1} \begin{bmatrix} \Psi & \Gamma \end{bmatrix},
\]

\[
\bar{P} = \bar{A}^T \bar{P} \left[ I + BR^{-1}B^T \bar{P} \right]^{-1} \bar{A} + \bar{Q}. \tag{4.31}
\]

Parameters \( R \) and \( \bar{Q} \) are the ones defined in section 3.2. Since each controller only computes the position of one gate, parameter \( R \) is not a matrix, but a non-negative real constant, \( \rho \).
4.2 Distributed control tolerant to actuator faults

4.2.2 Coordination between controllers

We now need to guarantee coordination between all three controllers, \textit{i.e.}, that all the manipulated variables of the canal are computed taking the others into account. Otherwise, we would be standing before decentralized control, which may lead to oscillatory behavior.

Coordination between controllers is accomplished through a two-step algorithm that at each sampling time, \( k \), repeats the following actions a specified number of times that assures that convergence is reached:

1. Compute \( v_i(k) \), \( i = 1, \ldots, m \), with \( m \) the number of pools considered, through equation (4.28).
   Before the first iteration, \( d_{gate}(k) \) should be initialized beforehand with the last values of the gate positions read from the canal.

2. Set \( d_{gate}(k) = \begin{bmatrix} v_1(k) \\ \vdots \\ v_m(k) \end{bmatrix} \).

Convergence can be reached as long as
\[ \lambda \triangleq \max |\text{eig}(K_{ffgate})| < 1, \]
where \( K_{ffgate} \) is the matrix that relates the component of \( v_{ff} \) responsible for the accessible disturbances derived from the neighbors manipulated variables, \( v_{ffgate} \), with \( d_{gate} \) such that
\[ v_{ffgate}(k) = K_{ffgate} d_{gate}(k), \tag{4.32} \]
in which \( K_{ffgate} \) is a \( m \times m \) matrix and \( v_{ffgate}(k) \in \mathbb{R}^m \), \( m \) being the number of pools considered.

4.2.3 LQG controllers structure

Each pool is controlled by a LQG controller composed by the LQR in section 4.2.1, of which we need to determine parameter \( \rho \), and the estimator presented in section 4.1.1. When all three gates are operating normally, three controllers are used, each one responsible for controlling the water level of each pool. In figure 4.13 we can see the interaction between these three controllers and the canal.

Each controller, \( C_{N,i} \), interacts with its neighbors and the respective pool as shown in figure 4.14. Variable \( d \) represents both accessible disturbances, \textit{i.e.}, \( d = \begin{bmatrix} d_{gate} \\ d_{off} \end{bmatrix} \).

In order for the design of the LQG controllers to be complete, we need to chose an adequate value of \( \rho \). The lower value of \( \rho \) that does not lead to oscillatory behavior is once again \( 10^3 \), hence that was the value chosen. Figure 4.15 shows the simulation of the SIMULINK model, when the water levels of pools 1, 2 and 3 are subject to distributed control. The simulation is the same as that in figure 4.3, but instead of using one multivariable controller, we now use three single gate controllers that interact with each other.
4. Actuator Faults

In figure 4.15, we can see that all water levels follow the respective reference, apart from slight deviations that occur when the other pools references or offtakes suffer changes. Once again, it is evident that the water level of pool 1 is barely affected by events in the other pools or gates, only responding to variations of gate 1 and offtake 1, whereas pool 2 already depends on changes occurring in pool 1 and both pools 1 and 2 influence pool 3. If we now attend to time instant $8 \times 10^3$ s, we observe the effect of distributed control, since all gates react to the variation of reference 1: gate 1 responds directly to the reference variation; gate 2 moves as a response to the interaction between controllers 1 and 2, in order to ensure that the water level of pool 2 is also kept at the desired value; since gate 2 moved, the interaction of controller 3 with controller 2 leads gate 3 to move as well, in order to ensure that the water level of pool 3 is also kept at the desired value.

Turning our attention to the gate positions plot, once again we distinguish several set points around which variations are carried out. These set points are the same as those seen in figure 4.3. If we compare figures 4.3 and 4.15, we notice that the deviations derived from the changes in the pools neighbors references are slightly more emphasized now than in the centralized control case. Moreover, if we attend to the responses when the respective reference changes, we see that in the centralized control case, it reaches the final value faster. In fact, in that situation, we observe that when the water level slowly tries to accompany the reference, as another change
4.2 Distributed control tolerant to actuator faults

Figure 4.15: Closed-loop response of the controlled SIMULINK model, using distributed control. Only the water levels of pools 1, 2 and 3 and the positions of the respective gates are taken into account, while gate 4 is kept constant at the position of 28.77 cm. The controllers parameters are $\rho_i = 10^4$, $r_n = 10^2$, $q = 1$ and $\alpha = 1.005$. 
4. Actuator Faults

occurs, a sudden variation is verified that takes the water level to the desired value. If it was not for that, the settling time would be much higher, as we can see at the beginning of the time evolution of the water level of pool 3.

As to the model response to disturbances derived from the offtakes, we notice that the feedforward action present in the distributed control system leads to better results: even though the overshoot observed now is not lower than that observed in figure 4.3, the water level is much faster restored to the intended value, since the controller responds to the disturbance at the time of its occurrence. In the evolution of the gate positions, if we attend, for instance, to time $1.85 \times 10^4$ s, we see that, at the variation of the flow drawn by offtake 1, all gate positions change their set points, as it was seen in figure 4.3. However, we now observe a slight overshoot representative of the feedforward action, that was not present in the centralized control case. At time $1.5 \times 10^4$ s, when offtake 2 ceases draining water, the same effect is observed but only in the position of gates 2 and 3, since pool 1 and gate 1 are not affected by the others. One last example, at time $1 \times 10^4$ s, when the flow of offtake 3 drops to 0 l/s, only the position of gate 3 changes, once again accusing the feedforward action.

After a fault is detected, only the water levels of pools 1 and 3 are controlled, but no longer by the same LQG controllers. The three controllers designed for the regular operation case are replaced by two new ones, designed for the case that gate 2 is kept constant at a height of $6 \text{ cm}$. The controllers parameters are, however, the same as before, as they proved to be the best option. Figure 4.16 shows the interaction between the two controllers and the canal. The feedforward action adjacent to the offtake disturbances is held as before: pool 1 is affected by the offtakes of pools 1 and 2; pool 2 is affected by the offtakes of all three pools and pool 3 is affected by the offtakes of pools 2 and 3.

Figure 4.16: Distributed controller after a fault is detected in gate 2.

Figure 4.17 shows the SIMULINK model response, when the water levels of pools 1 and 3 are controlled by those LQG controllers, while gate 2 is kept constant at a height of $6 \text{ cm}$ and gate 4.
4.2 Distributed control tolerant to actuator faults

at 28.77 cm.

In figure 4.17 we notice that the water level of pool 1 follows the reference, only with slight deviations that, as seen before, occur mostly when the flow of offtake 1 varies, but now also when the other offtakes and the reference of pool 3 vary. As to the water level of pool 3, the response presented is much slower than that of the water level of pool 1. That can be easily observed in the last transition of reference 3. In fact, if we attend to the time interval between $1.7 \times 10^4$ and $2.1 \times 10^4$, we see that level 3 never reaches the reference, as the events in pool 1 do not allow it. First, the closure of offtake 1, which would cause all water levels to rise, results in the opening of gate 3, which leads to a decrease in the water level of pool 3. After that, as reference 1 drops, gate 3 opens again, as a result of the interaction between controllers 1 and 3. If instead we turn our attention to the interval $9.2 \times 10^3$ to $1.32 \times 10^4$, we see that the water level of pool 3 gets much closer to the reference. This happens for the closure of offtake 3 causes the water level of pool 3 to rise and, consequently, gate 3 to open.

Even though it may be not so clear in figure 4.17, the offtake flows changes cause a slight oscillation in the water levels responses. This problem could be solved by increasing the value of $\rho$, but that would make the controllers even slower.

By comparing figures 4.15 and 4.17, we observe that the model response subject to centralized control is substantially better than when it is subject to distributed control, especially for the water level of pool 3.

4.2.4 Reconfiguration

The complete distributed control system is composed of five LQG controllers, described in section 4.2.3, and the fault detection algorithm described in section 4.1.3. The reconfiguration using a distributed controller is equal to that using centralized control, seen in section 4.1.4.

The same simulation as the one presented in figure 4.9 for the centralized control case is shown in figure 4.18, but now making use of the distributed control system designed.

The occurrence of the fault, at time $8900$ s causes significant oscillation in the water level of pool 3, but not in those of the other pools, as seen for the centralized control case, in figure 4.9. Also by comparing figures 4.9 and 4.18, we notice that even though the fault occurs at the same time in both cases, gate 2 stops moving at a lower position in the centralized control situation. This happens because, in the distributed control case, the gates responses are faster.

To eliminate the oscillations caused by the transition between controllers, we make use of the same bumpless transfer method described in section 4.1.4. The simulation of figure 4.18 is repeated, now taking into account the initialization of the integrator state, and the result is shown in figure 4.19. As we can see, once again, the transition is no longer noticeable.
4. Actuator Faults

Figure 4.17: Closed-loop response of the controlled SIMULINK model, using distributed control, when gate 2 is kept constant at a position of 6 cm. Only the water levels of pools 1, 2 and 3 and the positions of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controllers parameters are $\rho_i = 10^3$, $r_n = 10^2$, $q = 1$ and $\alpha = 1.005$. 

Water levels

Gate positions

Offtake flows
4.2 Distributed control tolerant to actuator faults

Figure 4.18: Closed-loop response of the controlled SIMULINK model, using distributed control, when gate 2 remains at a position of nearly 4 cm after time instant 8900 s. Only the water levels of pools 1, 2 and 3 and the positions of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controllers parameters are $\rho_i = 10^3$, $r_{ni} = 10^2$, $q = 1$ and $\alpha = 1.005$. 
Figure 4.19: Closed-loop response of the controlled SIMULINK model, using distributed control, when gate 2 remains at a position of nearly 4 cm after time instant $8900 \text{ s}$. Only the water levels of pools 1, 2 and 3 and the positions of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controllers parameters are $\rho_i = 10^3$, $r_{ni} = 10^2$, $q = 1$ and $\alpha = 1.005$. Bumpless transition between controllers.
5 Sensor Faults

Contents

5.1 Sensor fault definition ........................................ 61
5.2 Signal reconstruction ........................................ 61
5.3 Sensor Fault Detection ........................................ 62
5.4 Centralized control tolerant to sensor faults ................ 64
5.5 Distributed control tolerant to sensor faults ............... 64
5. Sensor Faults
This chapter explains how a faulty signal can be reconstructed from the other measurements available and presents a fault detection algorithm. The results of some illustrative simulations of the nonlinear model of the canal are presented.

5.1 Sensor fault definition

We say a sensor fault has occurred whenever the value read by the sensor is not the correct one. Therefore, the main challenges in this subject are to reconstruct the signal lost from the information available of the other sensors and gates and, afterwards, the detection of the fault. These studies are performed in sections 5.2 and 5.3.

As a matter of simplicity, we consider that faults can only occur in the sensor placed downstream of pool 2 and that the fault consists of the sensor reading always a value much higher than the true water level.

5.2 Signal reconstruction

Since the water levels of the several pools are decoupled, i.e., one does not influence the others, the reconstruction of the water level at the end of pool 2 can not be based only on the water levels of the other pools, and the gate positions just by themselves are not enough. Therefore, we exploit the sensor placed at the center of pool 2, insofar as it is the closest one to the faulty sensor.

Once again, we create an ARX model, which has now as input the water level at the center of pool 2, $C_2$, and as output the water level at the end of pool 2, $J_2$. As in chapter 2, the data must be filtered and its mean removed before the identification. The response of the best model found is compared to the validation data set in figure 5.1 and the orders of the polynomials $A(q)$ and $B(q)$ and the delay that lead to this result are $na = 1$, $nb = 4$ and $nk = 1$.

![Comparison between data and model response](image)

Figure 5.1: Reconstruction of the value read by the sensor placed at the end of pool 2, $J_2$, from the value read by the sensor placed at the center of pool 2, $C_2$.

The fit of 99.59\% obtained shows that the central water level, $C_2$, is enough to reconstruct the value of the downstream water level, $J_2$. 
5. Sensor Faults

5.3 Sensor Fault Detection

Initially, the value sent to the controller is the one read by the sensor placed at the end of pool 2, \( J_2 \). As a fault occurs, that signal must be replaced by the reconstructed one, which requires the existence of a fault detector.

The sensor fault detector is similar to the actuator fault one, in the sense that we once again use a performance index, \( \Pi' \), that now uses the error \( \tilde{y}_i \), defined as the absolute value of the difference between the water level at the end of pool \( i \), \( J_i \), and the reconstructed value, \( J_i^* \).

\[
\tilde{y}_i(k) = J_i(k) - J_i^*(k) .
\] (5.1)

With a similar structure as the one in section 4.1.3, the performance index can be obtained from the error \( \tilde{y}_i \) by

\[
\Pi'(k) = \gamma \Pi'(k - 1) + (1 - \gamma) |\tilde{y}_i(k)| .
\] (5.2)

Since only sensor 2 is considered liable to fail, in this case, \( i = 2 \).

As in section 4.1.3, whenever this performance index exceeds a given value, \( \Pi_{\text{max}} \), a fault is detected. Additionally, we now also consider that the fault may be temporary and the sensor may return to its usual behavior without human intervention. The detection of the fault recovery is accomplished through the same performance index and the same boundary condition.

Figure 5.2 shows the effect of the variation of \( \gamma \) and \( \Pi_{\text{max}} \) on the number of false alarms and both the time elapsed since a fault occurs until it is detected and the time elapsed since a recovery from a fault occurs until it is detected, for the experimental example presented in figure 6.7.

Before going any further, a note should be made with respect to the example presented in figure 6.7. When the experiment was done, the reconstructed signal used was achieved not only through the value read by the central sensor, but also through the flows drawn by gates 1 and 2. This is, however, not the best option, since the value of the flow drawn by gate 2 is obtained with \( J_2 \), which after the fault occurs is not the true water level value. The reason the results obtained are not very different is due to the fact that, as seen in section 5.2, the central sensor alone is responsible for 99.67\% of the fit, and the addition of the flows only improves it in 0.3\%.

Now looking at figure 5.2, and beginning with plot (a), we notice that, unlike what was seen in section 4.1.3, from the chosen threshold value, \( \Pi_{\text{max}} = 0.03 m \), the variation of \( \gamma \) does not influence significantly the number of false alarms. For values of \( \gamma \) lower than 0.1, only one false alarm is detected, and for higher values, as the transitions in \( \Pi \) attenuate, no false alarms are detected. As to the fault detection time, in plot (b), we see that until \( \gamma = 0.9 \), the detection is immediate. This only happens because after the fault occurs, the faulty sensor reads always the value 800 mm, which is substantially higher than the true water level. After \( \gamma = 0.9 \), the fault detection time rises abruptly, since, as \( \gamma \to 1 \), \( \Pi \to 0 \), and at the limit the fault is never detected.
5.3 Sensor Fault Detection

Figure 5.2: Effect of the variation of $\gamma$ on the number of false alarms and the time of fault detection and recovery, while keeping $\Pi_{\text{max}} = 0.03 m$ (a, b, c), and effect of the variation of $\Pi_{\text{max}}$ on the number of false alarms and the time of fault detection and recovery, while keeping $\gamma = 0.9$ (d, e, f).

The same applies to the recovery detection time, in plot (c). The maximum acceptable value would, therefore, be 0.9.

If we now attend to plots (d), (e) and (f), in figure 5.2, where the threshold varies, while $\gamma = 0.9$, we notice that, beginning with plot (d), as the threshold rises, the number of false alarms diminishes, until it equals zero, for $\Pi_{\text{max}} \geq 0.021$. Initially only one false alarm is detected, since for $\Pi = 0$, a fault is detected from the start, until the end. From plots (e) and (f), we see that the lower the threshold, the sooner the fault is detected but the later the recovery is detected and that, the higher the threshold, the later the fault is detected but the sooner the recovery is detected. Thus, a compromise must be reached, taking into account that $\Pi_{\text{max}} \geq 0.021$, so that no false alarms occur. By observation of the graphs, the value chosen was 0.03 $m$.

In (e), when the fault detection time is $-1$, it means that a fault was detected right from the beginning.

In section 6.2, we will, however, see that even with the value $\Pi_{\text{max}} = 0.04 m$ chosen, false alarms still occur. This is due to the fact that a different ARX model was used in the reconstruction of the lost signal, as already explained.
5. Sensor Faults

5.4 Centralized control tolerant to sensor faults

The centralized LQG controller designed in section 4.1.1 is the one to be used to control the three first pools of the canal in this section. As before, gate 4 is kept at the position of 28.77 cm.

Figure 5.3 shows the result of a simulation of the SIMULINK model, subject to that controller.

The simulation is in all aspects similar to that seen in figure 4.3, with the exception that after time 4700 s a fault occurs in the sensor placed at the end of pool 2. Upon the occurrence of the fault, which consists in the sensor reading always the same value (in this case, 1 m), the value used to control the water level of pool 2 is no longer the one read by that sensor, but the one reconstructed from the value read by the sensor placed at the center of pool 2.

Before the reconstruction process, in order to obtain a better result, the ARX model input should be filtered. Moreover, the reconstruction process makes use of an ARX model, which requires the input to be unbiased. Therefore the set point value should be subtracted from the value read by the central sensor of pool 2 beforehand. At the end, as the reconstruction obtained is unbiased, an offset should be added to it. This offset can be obtained through the integration of the difference between the value read by the sensor at the end of pool 2 and the reconstructed signal over the first time instants of the simulation, divided by the time interval.

If we compare figures 4.3 and 5.3, we notice that the responses present more significant deviations from the references, when the flows of the offtakes vary, now. Despite that, the water levels still follow the references, showing a very similar behavior to that seen in figure 4.3.

In fact, if we attend to figure 5.4, where the reconstructed value (red) is compared to the real one (green), we see that they are very close. Therefore it is expected that the response obtained through the use of the reconstructed value is similar to the one obtained with the real value.

If we now turn our attention to figure 5.5, which shows the estimation error between the real value and the reconstructed one, we see that it is always below 3 cm, which is the threshold chosen. Comparing figure 5.5 to figure 5.4, we notice that the peaks observed in the latter occur when a reference or the flow of an offtake vary.

In this case, as the value read by the faulty sensor is much higher than the true value of the water level of pool 2, the fault is detected immediately. In figure 5.6 the estimation error between the value read by the sensor placed at the end of pool 2 and the reconstructed one is shown. As we can see, the error is more than ten times higher after the occurrence of the fault, which allows the fault detection to be immediate, in as much as the threshold chosen is close to the estimation error when no fault has occurred.

5.5 Distributed control tolerant to sensor faults

The simulation presented in section 5.4 is repeated, only now using distributed control, and the result is shown in figure 5.7.
5.5 Distributed control tolerant to sensor faults

Figure 5.3: Closed-loop response of the controlled SIMULINK model, when sensor 2 fails at time 4700 s, using centralized control. Only the water levels of pools 1, 2 and 3 and the positions of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controller parameters are $\rho_i = 10^3$, $\nu_i = 10^2$, $q = 1$ and $\alpha = 1.005$. 
5. Sensor Faults

Figure 5.4: Comparison between the real value of the water level at the end of pool 2 (green) and the reconstructed value (red). The reference is presented in blue.

Figure 5.5: Estimation error between the real value of the water level downstream of pool 2 and the one reconstructed through the value read by the sensor placed at the center of pool 2. The threshold is represented in red.

Figure 5.6: Estimation error between the value read by the sensor placed at the end of pool 2 and the value reconstructed through the readings of the sensor placed at the center of pool 2. The threshold is represented in red and the vertical orange line signals the fault occurrence.
5.5 Distributed control tolerant to sensor faults

Figure 5.7: Closed-loop response of the controlled SIMULINK model, when sensor 2 fails at time 4700 s, using distributed control. Only the water levels of pools 1, 2 and 3 and the positions of the respective gates are taken into account, while gate 4 is kept at the position of 28.77 cm. The controller parameters are $\rho_i = 10^3$, $r_{ni} = 10^2$, $q = 1$ and $\alpha = 1.005$. 
5. Sensor Faults

If we compare figure 5.7 with figure 4.15, we see that the time evolution of the water level of pool 1, as well as the position of the respective gate, is almost the same, the differences are barely detectable. As to the other pools, the differences in the water levels and gate positions time evolutions are detectable, but we can see they are still very similar. The water level of pool 3 is the one that presents more significant deviations from the reference, which were not so evident in figure 4.15. Nevertheless, all three responses follow the respective references with slight deviations, from which they always retrieve.
6

Experimental Results

Contents

6.1 Actuator faults ................................................. 71
6.2 Sensor faults ................................................. 74
In this chapter some of the experimental results obtained are exposed, regarding both centralized and distributed controllers and for both actuator and sensor faults situations.

6.1 Actuator faults

Figure 6.1 shows the closed-loop response of the canal, when the water levels of pools 1 to 3 are controlled with the centralized control system designed in section 4.1. In fact, experiments in the canal proved the controller parameters used in the SIMULINK model simulations to be the most suitable.

Figure 6.1: Closed-loop response of the canal subject to centralized control. Reconfiguration after a fault in gate 2 is detected (yellow mark). Both controllers parameters are $\rho_i = 1000$, $r_n = 100$, $q = 1$ and $\alpha = 1.004$. Water levels above, with the respective references, and the gate positions below, their true value in green and the command sent in blue. The red vertical line signals the fault occurrence.

Initially, the three gates and the three water levels are controlled. We can see, around time $800$ s, when reference 1 rises, the effect of centralized control in the coordination between gates...
1 and 2. As the reference rises, gate 1 closes, so as to retain more water in pool 1, and so does gate 2, so that the level of pool 2 does not decrease. We can still notice that gate 3 opens, even if very slightly, which reflects the coordination with gate 2. This is in accordance to what was seen in sections 2.2.2 and 4.1.1. This coordination assures that the water levels of all three pools follow the respective references, only presenting slight oscillations.

At time 1500 s a fault in the actuator of gate 2 occurs (red mark) and the gate remains at the same position henceforward. After approximately 2 min, when the gate position no longer responds to the water level oscillations, the fault is detected (yellow mark) and we can notice a slight oscillation, especially in the water level of pool 1. This is due to the abrupt transition from one controller to the other, which is evident in the plot of the position of gate 1. When this experiment was done, the bumpless transfer problem had not yet been solved.

Once the fault is detected, the controller is reconfigured, as explained before, to use only the still operating gates. Immediately after the fault is detected, the water levels of all pools continue following the respective references, until the reference of pool 3 decreases. After that, the water levels of pools 1 and 3 keep on following the references, while the water level of pool 2 decreases under the aimed value, in response to the opening of gate 3. Also, when reference 3 varies, we can see that the position of gate 1 also varies, even if very slightly. This reflects the coordination between gates 1 and 3, at the same time that it shows that the pools upstream are much less affected by the ones downstream than the other way round. The latter is evident when, at time 5400 s, reference 1 decreases and both gates 1 and 3 open almost equally, with the difference that gate 3 closes again afterwards, while gate 1 remains at the same position. We can also notice that, while this transition caused the water level of pool 3 to deviate from its reference, the transition in reference 3 did not do the same with respect to the water level of pool 1. In relation to the water level of pool 2, this last transition causes it to rise, since, as gate 1 opens, more water is drained to pool 2. As gate 3 opens, the excess of water is drained from pool 2 to pool 3 and from this one to pool 4, and the water level of pool 2 returns to its previous value.

Figure 6.2 shows the same simulation but now using the distributed control system. Opposite to what happened in the centralized case, the experiments performed in the canal using distributed control presented better results for $\rho_i = 300$.

When comparing figure 6.2 to figure 6.1, we notice that the distributed control system leads to a more oscillatory behavior, which is particularly evident in the water level of pool 3 and correspondent gate. As before, in the first transition of gate 1, the coordination between the controllers is evident, since all gates close, so as to cause the water level of pool 1 to rise. As the fault occurs, we notice that it is detected sooner (within 45 s) than in the centralized case, which is also due to the more oscillatory behavior now observed. Once again, we witness an abrupt transition from one controller to the other, especially evident in the plot of the position of gate 1. After the
fault is detected, the behavior observed is very similar to the one witnessed in figure 6.1, and as
the references 1 and 3 vary we can once again see the coordination between controllers 1 and
3, but always in the presence of a more significant oscillation. Apart from that, we note that the
water levels follow the respective references, except for the water level of pool 2 after the fault
occurs, as expected. However, at the end, when reference 1 drops, we witness a decrease in the
performance of distributed control compared to centralized control, as the water level of pool 3
presents an almost constant, although not very significant, offset in relation to the reference.

![Water levels](image1)

![Gate positions](image2)

Figure 6.2: Closed-loop response of the canal subject to distributed control. Reconfiguration after
a fault in gate 2 is detected (yellow mark). Both controllers parameters are $\rho_i = 300$, $r_{ni} = 100$,
$q = 1$ and $\alpha = 1.004$. Water levels above, with the respective references, and the gate positions
below, their true values in green and the command sent in blue. The red vertical line signals the
fault occurrence.
6. Experimental Results

6.2 Sensor faults

Figure 6.3 shows the closed-loop response of pools 1 to 3 of the canal, when the centralized LQG controller designed in section 4.1.1 is used to control the water levels. At time 1200 s a fault occurs in the sensor placed at the end of pool 2 (first red line), which, after the fault is detected (first yellow line), keeps on reading the value 800 mm, regardless of the true value of the water level in pool 2. Therefore, after that, the value used by the controller is no longer the one read by that sensor, but the one reconstructed from the sensor placed at the center of pool 2, as explained in section 5.2. At time 5400 s the fault ends (second red line), i.e., the sensor is again able to read the true water level values and, as the fault is detected (second yellow line), the controller receives again the value read by the sensor.

![Water levels graph]

![Gate positions graph]

Figure 6.3: Closed-loop response of the canal subject to centralized control. Signal reconstruction used after a fault is detected in sensor 2 (first yellow line) and until the fault recovery is detected (second yellow line). The controller parameters are $\rho = 1000$, $r_n = 100$, $q = 1$ and $\alpha = 1.004$. Water levels above, with the respective references, and the gate positions below, their true value in green and the command sent in blue. The red vertical line signals the fault occurrence.
6.2 Sensor faults

First of all, we notice that the red and yellow lines overlap, when the fault occurs. That only happens, because the faulty sensor reads a value considerably higher than the true water level value, which causes the fault detection to be immediate. In fact, if we attend to figure 6.4, which shows the performance index defined in equation (5.2) (above) and the fault indicator (below), we see that the performance index is significantly higher during the time the sensor is not correctly operating. The threshold chosen in this case is $\Pi_{\text{max}} = 4 \text{ cm}$, which here is high enough to assure that no false alarms are detected.

![Figure 6.4: Performance index (above) and fault detection indicator (below), when centralized control is used.](image)

Again in reference to figure 6.3, after the fault occurs, we witness some oscillation in the position of gate 1, which reverberates throughout all pools, but only until the rise of reference 2. When reference 2 varies, the water level of pool 2 is able to accompany it, however not succeeding as it would if the fault had not occurred.

Here, once again, we observe the effect of centralized control in the coordination between the gates, for instance when reference 2 decreases. We can see that at the same time that gate 2 opens, so as to draw more water out of pool 2, so does gate 3, in order not to accumulate it in pool 3, and gate 1 closes, so that the water level of pool 1 do not decrease as a consequence of gravity. Also, at the last transition of reference 1, we notice that all gates respond to it, although with decreasing intensity from gate 1 to gate 3. On the other hand, as reference 3 varies, gate 1 does not respond to it. This is all in accordance to what was seen in previous sections.

Even though the results obtained are satisfactory, it may be possible to improve them by reconstructing the signal lost only from the value read by the sensor placed at the center of pool 2. As already mentioned in section 5.3, these results were obtained for the case that the reconstructed signal is built also with the flows drawn by gates 1 and 2. In fact, if we attend to figures 6.5 and 6.6, we see the comparison between the real value of the water level at the end of pool 2 and the reconstructed value, using, respectively, both the central sensor and the flows drawn by gates 1 and 2 and only the central sensor.

Although we can still witness a slight delay, which arises from the signal filtering, the reconstruction presented in figure 6.6 is considerably better than that presented in figure 6.5. This
6. Experimental Results

Figure 6.5: Comparison between the real value of the water level at the end of pool 2 (blue) and the reconstructed signal (green), using centralized control. Reconstructed signal built from the central sensor and flows drawn by gates 1 and 2.

Figure 6.6: Comparison between the real value of the water level at the end of pool 2 (blue) and the reconstructed signal (green), using centralized control. Only the value read by the sensor placed at the center of pool 2 was used in the reconstruction.

explains why the threshold chosen in section 5.3, where the reconstruction was considered to be based only on the central sensor, was 3 cm and in this simulation was set at 4 cm.

Still regarding the reconstruction process, a note should be made. As the ARX model based on which the reconstruction is carried out was obtained from data collected from the SIMULINK model, when applied to the actual canal, the reconstructed value has to be multiplied by a factor of 0.8575 so as to match the real water level value. An offset was also observed, which was figured through the method already explained in section 5.3. Moreover, the reconstructed value, only resulting from the linear ARX model applied to the value read by the central sensor, presented highly significant noise, that needed to be removed. The filter used for that purpose was no longer the butterworth one used until now, but a second order filter with transfer function

\[ H(z) = \frac{B(z)}{A(z)}, \]

where

\[ B(z) = 0.0064, \quad A(z) = 1 - 1.84z^{-1} + 0.8464z^{-2}. \]

Concerning now distributed control, if we attend to figure 6.7, we find the exact same simulation, this time using the distributed control system used in section 6.1, when no actuator faults were detected.
If we compare figure 6.7 to figure 6.3, we notice the very similar behaviors, except that with distributed control, the responses obtained are more oscillatory, as already seen in the centralized control case. Furthermore, it is more evident here the effect that the references variations have on pool 3, whose water level in those situations deviates considerably from the desired value.

With respect to the time instant at which a fault is detected, we see that once again the fault occurrence and the fault detection are simultaneous. In fact, if we attend to the performance index in figure 6.8, we observe a plot similar to the one seen for the centralized control case.

As mentioned in section 5.3, even though the threshold used in this simulation was 4 cm, false alarms were still detected. In fact, when in figure 6.7 we observe a thick yellow line, it corresponds to two successive false alarms (four yellow lines). By observation of figure 6.8, we find that the
6. Experimental Results

Figure 6.8: Performance index (above) and fault detection indicator (below), when distributed control is used.

The performance index exceeds the threshold shortly after \(6000\) s. Due to an oscillation that makes the performance index decreases and rise again, the threshold line is crossed four times, which corresponds to two false alarms. Below, the fault detection indicator also presents a thick vertical line, once again corresponding to four vertical lines.

By comparing figures 6.9 and 6.10, it becomes more clear why false alarms are still detected, even though a threshold of \(4\, cm\) was chosen, when in section 5.3 it had been seen that \(3\, cm\) would include sufficient margin to apply to other cases.

Figure 6.9: Comparison between the real value of the water level at the end of pool 2 (blue) and the reconstructed signal (green), using distributed control. Reconstructed signal built from the central sensor and flows drawn by gates 1 and 2.

Figure 6.10: Comparison between the real value of the water level at the end of pool 2 (blue) and the reconstructed signal (green), using distributed control. Only the value read by the sensor placed at the center of pool 2 was used in the reconstruction.
As it was seen for the centralized control case, using the signal reconstructed only from the central sensor leads to a considerably better result. Therefore, it is expected that the response of the water level of pool 2 will improve by the usage of the reconstructed signal obtained this way.
6. Experimental Results
7 Conclusions

Contents

7.1 Main points .................................................. 83
7.2 Future work .................................................. 84
7. Conclusions
This chapter draws the main conclusions that can be extracted from the work produced and enumerates some points that can be improved or further developed as future work on this subject.

7.1 Main points

In this dissertation, actuator and sensor fault tolerant controllers are designed to control the downstream water level of each pool of a water delivery canal. The controllers considered are LQG, based on linear models obtained from data of a nonlinear model that matches the canal. Both centralized and distributed controllers are considered.

Fault tolerance stems from reconfiguration of the controllers structures, upon the detection of a fault. As to actuator faults, when a fault is detected the controller switches to a different structure that uses the still operating gate actuators, in order to assure that the water levels continue to follow the respective references. A bumpless transition is accomplished by initializing the integrator of the controller for the faulty condition such that its output before the fault is detected matches the real gates positions. As to sensor faults, when a fault occurs, the value read by the sensor is replaced by the one reconstructed from the central sensor of the same pool.

In the system identification section, where the linear models on which the controllers design is based are obtained, it was seen that using the flow drawn by the gates instead of the respective gate positions improves the identification result, in as much as a nonlinearity in a Hammerstein system is removed by a change of the manipulated variable.

Regarding the actuator faults, using centralized control, the experimental results obtained show that the controllers designed are able to force the water levels to follow the respective references, except for pool 2 when a fault in gate 2 occurs. The responses show a permanent oscillation of small amplitude, that is increased with a change in a reference or in the offtake flows. With respect to distributed control, when compared to centralized control, a decrease in performance is noticed, mainly due to a more intense oscillation and sharper deviations from the reference, observed when the other references are varied.

The solution presented to deal with actuator faults exhibit, however, some limitations. On the one hand, although we manage to maintain the water levels of pool 1 and 3 at the desired values, the same does not happen with pool 2. On the other hand, and this entailing greater concern, it was seen that some situations might lead to water overflow. For instance, in the event that gate 2 stops moving at a position lower than $3\, \text{cm}$, or when, before a fault occurs, some of the offtakes are draining water and stop afterwards.

In relation to sensor faults, although the results obtained were satisfying, they can be improved; when a fault in a sensor is detected and its readings can not be used by the control system, the reconstructed signal should be built only from the value read by the central sensor, instead of also using the flows drawn by gates 1 and 2.
7.2 Future work

With regard to actuator faults, to begin with, although the bumpless transition method has already successfully been applied to the SIMULINK model, it still needs to be tested in the actual canal. Afterwards, so that this control system can be useful controlling the canal, it must be prepared for the occurrence of faults in any of the gate actuators, and not only in gate 2.

Further work can be developed aimed at dealing with the limitations found. For example, the overflow issue can be handled by also controlling the intake flow. Moreover, although the responses obtained were satisfying, as they followed the references, efforts should be made in the sense of removing the permanent oscillation observed.

Concerning the sensor faults, the signal reconstruction using only the readings of the central sensor needs experimental testing, even though very good results were obtained in simulations with the SIMULINK model. Furthermore, although the transition between the signals sent to the controller does not cause significant oscillations, a bumpless transition would still be preferable. Afterwards, as it was said in relation to the actuator faults, the algorithm should be prepared to deal with faults in any of the sensors, and not only in sensor 2.

Opposite to actuator faults, sensor faults can be of several types. In the present study we only consider the one that needs the simplest fault detection algorithm, since the faulty sensor reads always a value much higher than the true value of the water level. However, a fault in a sensor should be detected whenever it is not reading the correct value of the water level, and therefore, the fault and fault recovery detection algorithms need to be further developed.
Bibliography


Bibliography

Butterworth filter coefficients
A. Butterworth filter coefficients
The butterworth filter used has the following transfer function

\[ H(z) = \frac{B(z)}{A(z)}, \]

where

\[
B(z) = 10^{-4} \times (0.0735 + 0.3676 z^{-1} + 0.7352 z^{-2} + 0.7352 z^{-3} + 0.3676 z^{-4} + 0.0735 z^{-5})
\]

and

\[
A(z) = 1 - 4.3531 z^{-1} + 7.6167 z^{-2} - 6.6928 z^{-3} + 2.9522 z^{-4} - 0.5228 z^{-5}.
\]
A. Butterworth filter coefficients
Pontryagin’s Minimum Principle in discrete time
B. Pontryagin’s Minimum Principle in discrete time
Consider the discrete time plant modeled by the nonlinear equation
\[ x(k + 1) = f(x(k), u(k), k), \]  
(B.1)

where \( k \) is a non-negative integer that represents discrete time, \( x \in \mathbb{R}^n \) the state, \( u \in \mathbb{R}^{n_u} \) is the plant input and \( f : \mathbb{R}^n \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^n \) is a vector function which describes the plant dynamics. The plant initial condition \( x(0) \) is specified. To this plant we associate the performance index
\[ J(u) = \Phi(N, x(N)) + \sum_{k=0}^{N-1} L(x(k), u(k), k), \]  
(B.2)

which we intend to minimize, where \( L \) is the Lagrangian function. It is assumed that \( N \rightarrow \infty \), so that there are no constraints either on the final state \( x(N) \) or on \( u \).

The Hamiltonian function is defined by
\[ H(k) = \lambda^\top(k + 1) f(x(k), u(k), k) + L(x(k), u(k), k), \]  
(B.3)

where \( \lambda \) represents the co-state.

According to the Pontryagin Minimum Principle in discrete time, the optimal state trajectory \( x \), the co-state \( \lambda \) and the optimal control \( u \) satisfy the following set of conditions:

a) State equation (B.1) with the specified initial condition;

b) Adjoint equation
\[ \lambda(k) = \left( \frac{\partial f(x(k), u(k), k)}{\partial x(k)} \right)^\top \lambda(k + 1) + \left( \frac{\partial L(x(k), u(k), k)}{\partial x(k)} \right)^\top \]  
(B.4)

c) Stationarity condition
\[ \frac{\partial H(k)}{\partial u(k)} = 0 \]  
(B.5)

d) Co-state terminal condition
\[ \lambda(N) = \frac{\partial \phi}{\partial x(N)} \]  
(B.6)
B. Pontryagin’s Minimum Principle in discrete time