

ADVERTIMENT. La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del servei TDX (www.tesisenxarxa.net) ha estat autoritzada pels titulars dels drets de propietat intel·lectual únicament per a usos privats emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

ADVERTENCIA. La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del servicio TDR (www.tesisenred.net) ha sido autorizada por los titulares de los derechos de propiedad intelectual únicamente para usos privados enmarcados en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio TDR. No se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

WARNING. On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the TDX (www.tesisenxarxa.net) service has been authorized by the titular of the intellectual property rights only for private uses placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading and availability from a site foreign to the TDX service. Introducing its content in a window or frame foreign to the TDX service is not authorized (framing). This rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author



UNIVERSITAT POLITÈCNICA DE CATALUNYA
E.T.S. d'Enginyers de Camins, Canals i Ports de Barcelona



Doctorate Program in Civil Engineering

Doctoral Thesis

**EXPERIMENTAL DESIGN AND VERIFICATION
OF A CENTRALIZED CONTROLLER FOR
IRRIGATION CANALS**

Author:

Enric Bonet Gil

Advisors:

Dr. Manuel Gómez Valentín

Dr. Joan Soler Guitart

Dissertation submitted to obtain the degree of

**DOCTOR OF PHILOSOPHY
BY THE UNIVERSITAT POLITÈCNICA DE CATALUNYA**

FLUMEN INSTITUTE

January 2015, Barcelona



Our reward is in the effort, not the outcome. A full effort is full victory.
(Nuestra recompensa se encuentra en el esfuerzo y no en el resultado.
Un esfuerzo total es una victoria completa)

Mahatma Gandhi

To my parents, Antonio, Angeles and Tere
My strength and my life,



Abstract

This thesis aims to develop a predictive controller for irrigation canals to improve the management of water resources.

Water is necessary for life and it is a scarce good that we need for drinking, in the agriculture, etc. At the same time, it can constitute a serious threat in particular areas due to the difficulty to grow foods by the increasing of prolonged droughts.

The agriculture holds an important part of the food chain and the water resources for agriculture are important, the problem is the water transport systems present low efficiencies in practice. The yield agriculture has to be optimized, because the goal of an operational water manager is to deliver the water to the irrigation sites accurately and efficiently. To improve the efficiency of the water transport systems is necessary to invest in automating the operation of irrigation canals.

The management of a canal starts from setting the demand delivery accurately taking into account the crops necessities during the irrigation cycle and establishing the gate trajectories for controlling the canal in each time step. In an ideal case, the system would be controlled but someone could introduce a disturbance in the canal which could deviated the real canal state from the desired canal state. In that circumstance, it would be necessary a feedback controller which could aid the watermaster to restore the desired canal state. In order to fulfill this objective, we define an overall control diagrams scheme in chapter 5 which splits the management of the canal control in different blocks and each of these blocks is represented by a particular algorithm. The algorithms developed and tested for us in this thesis are the CSI and GoRoSoBo algorithms.

The first one (CSI) defines a powerful tool in the management of a canal. The Watermaster establishes the gates positions and fixes the desired water level at checkpoints to fulfill a scheduled demand. In that sense, when someone introduce a disturbance into the canal perturbs the water level at checkpoints, so the scheduled deliveries cannot fulfilled by the watermaster. In such case, the water level

measurements at these checkpoints could be sent to the CSI algorithms which calculates the real extracted flow and the current canal state along the canal, that is, the water level and velocity in all cross-sections of the canal. This task is performed by the CSI algorithm which has been designed in this thesis and tested in numerous numerical examples (chapter 7) and experimentally in the laboratory canal of the Technical University of Catalonia (chapter 8).

The last one (GoRoSoBo) is the essential tool in the management of a canal, that is, a control algorithm operating in real-time. The GoRoSoBo algorithm (Gómez, Rodellar, Soler, Bonet) is a feedback control algorithm which calculates the optimum gates trajectories for a predictive horizon taking into account the current canal state and the real extracted flow obtained by CSI as well as the scheduled demands and the previous gate trajectories. GoRoSoBo has been developed in this thesis and tested in several numerical examples (chapter 10) as well as with the Test-Cases proposed by the ASCE Task Committee on Canal Automation Algorithms (chapter 11). In that sense, we propose a centralized control performance to manage the canal control.

In addition to these two main contributions, many other smaller developments, minor results and practical recommendations for irrigation canal automation are presented throughout this thesis.

Acknowledgments

First at all, I'm sorry but I will not write the acknowledgments in English, because it is not easy to express my emotions in a foreign language.

Nunca pude entender porque en los acknowledgments de una tesis se escribiesen tantos agradecimientos, ahora que depósito mi tesis lo puedo llegar a entender. Demasiado tiempo para no pedir favores y no recibir ayudas.

Agradezco a mis tutores de tesis Manuel Gómez y Joan Soler por el tiempo dedicado, la paciencia y apoyarme en este largo camino que hemos recorrido juntos. Os aprecio enormemente, mil gracias. Joan encara que no puguis ser a la defensa, et tinc molt present!!!

Ivan et trobem a faltar mai t'oblidaré company!!!. Cesca moltes gracias sempre m'has recolçat.

Gracias Javi por la ayuda para programar el displayer en Java fuiste un gran profe y una mejor persona!!

Agradecer a mis tíos Emilio y Olga que siempre haya podido contar con ellos y eso nunca se olvida, lástima que estéis tan lejos, os quiero. A Ana y a mi primo Emiliet porque además de ser sangre de mi sangre estoy muy orgulloso de él y espero que siga luchando como los flojos sabemos.

A Isidro y Felicidad porque además de ser buenas personas siempre que me habéis podido ayudar allí habéis estado, a Javi y Juani, y a los peques.

Bueno, llega el turno de agradecer a la gente que más cuesta no emocionarse supongo. A Tere por ser mi compañera, pareja, amiga y animarme aunque a veces debería haberlo hecho yo, gracias por estos años que no han sido fáciles pero estando juntos no hay nada imposible.

A mis padres, mi vida, gracias por regalarme vuestros principios, defectos, virtudes y enfermedades, me gustan todas y las aprecio todas porque no podían venir de gente más pura. Os quiero y os querré siempre, estáis por encima del bien y del mal.

Gracias yayos siempre me quisisteis muchísimo y yo también, espero veros algún día nunca os dije adiós. Dicen que una persona nunca muere hasta que vive la última persona que lo quería, conmigo no moriréis nunca porque siempre os recordaré.

Hasta la victoria siempre!!!

Contents

Abstract	v
Acknowledgments	vii
Contents	ix
List of Figures	xvii
List of Tables	xxv
Nomenclature	xxvii
Abbreviations	xxxiii
CHAPTER 1	1
LAYOUT OF THESIS	1
CHAPTER 2	5
INTRODUCTION AND GENERAL STATEMENTS	5
2.1 DEFINITION OF THE IRRIGATION PROBLEM	6
2.1.1 <i>Conditions involved in reducing the efficiency</i>	7
2.1.1.1 Water distribution: rotational irrigation scheduling or on-demand irrigation scheduling8	
2.1.1.2 Increase of the water price.....	13
2.1.1.3 Productive structure	14
2.1.2 <i>Associated structural problems in canals</i>	15
2.2 SOLUTIONS TO THE IRRIGATION PROBLEM	16
2.3 DIAGNOSIS AND PROGNOSIS OF THE IRRIGATION PROBLEM	18

2.4	OBJECTIVES.....	20
CHAPTER 3		21
LITERATURE REVIEW		21
3.1.	INTRODUCTION.....	21
3.2.	CONSIDERED VARIABLES.....	22
3.2.1.	<i>Controlled variables</i>	23
3.2.1.1.	Controlled variable: Discharge.....	23
3.2.1.2.	Controlled variable: Water level.....	23
3.2.1.3.	Controlled variable: Volume.....	30
3.2.2.	<i>Measured variables</i>	31
3.2.3.	<i>Control action variables</i>	31
3.3.	LOGIC OF CONTROL	32
3.3.1.	<i>Open-loop control</i>	32
3.3.2.	<i>Closed-loop control</i>	33
3.4.	METHODS IN CONTROL SYSTEM DESIGN	34
3.4.1.	<i>Controllers design: Definition of inputs / outputs</i>	34
3.4.2.	<i>Monovaryable methods</i>	35
3.4.3.	<i>Multivariable methods</i>	36
3.5.	FIELD IMPLEMENTATION	37
CHAPTER 4.....		41
CHARACTERIZATION OF A PREDICTIVE CONTROL SCHEME.....		41
4.1.	INTRODUCTION.....	42
4.2.	COMPUTER MODEL	42
4.2.1.	<i>Linear model</i>	44
4.2.2.	<i>Non-Linear model</i>	48
4.3.	OPTIMIZATION PROBLEM	50
4.3.1.	<i>Introduction</i>	50
4.3.2.	<i>Desired output vector</i>	53
4.3.3.	<i>Objective function</i>	54

CHAPTER 5.....	57
OVERALL CONTROL DIAGRAM.....	57
5.1. INTRODUCTION	57
5.2. STRUCTURE DIAGRAM	57
5.3. CONCLUSIONS.....	64
CHAPTER 6.....	67
CANAL SURVEY INFORMATION: CSI ALGORITHM	67
6.1. INTRODUCTION	68
6.2. PROBLEM STATEMENT	71
6.2.1. <i>Definition of the optimization problem.....</i>	<i>72</i>
6.2.2. <i>Optimization problem: Objective function.....</i>	<i>76</i>
6.2.3. <i>Optimization problem statement.....</i>	<i>78</i>
6.2.4. <i>Conclusions of the problem statement</i>	<i>78</i>
6.2.5. <i>System output / inputs variables of CSI.....</i>	<i>80</i>
6.3. ESTIMATION ALGORITHM FROM UNCONSTRAINED OPTIMIZATION	81
6.3.1. <i>First-order and second-order conditions.....</i>	<i>81</i>
6.3.2. <i>Implemented algorithm</i>	<i>84</i>
6.4. CONCLUSION	88
6.5. APPENDIX I	91
6.5.1. <i>Free surface flow equations.....</i>	<i>91</i>
6.5.2. <i>The discretization of the characteristic equations</i>	<i>96</i>
6.5.3. <i>Applying to a structured grid</i>	<i>99</i>
6.5.4. <i>Control structures.....</i>	<i>102</i>
6.5.5. <i>The discretization of the checkpoint equations</i>	<i>104</i>
6.5.6. <i>Pump flow trajectories.....</i>	<i>107</i>
6.5.7. <i>The boundary conditions.....</i>	<i>108</i>
6.5.8. <i>Hydraulic influence of a pump flow trajectory parameter on the state vector</i> <i>112</i>	
6.5.9. <i>Compilation of the Hydraulic influence matrix</i>	<i>117</i>

6.5.9.1.	Definition	117
6.5.9.2.	The discrete observer	118
6.5.9.3.	Features of the hydraulic influence matrix	119
6.6.	APPENDIX II.....	121
CHAPTER 7		125
CSI PERFORMANCE: NUMERICAL EXAMPLES		125
7.1.	INTRODUCTION.....	126
7.2.	GEOMETRY	127
7.3.	DISCRETIZATION OF A PROBLEM	127
7.4.	BOUNDARY AND INITIAL CONDITIONS	128
7.5.	TEST CASES ANALYZED	130
7.5.1.	<i>First example: flow disturbance</i>	<i>130</i>
7.5.2.	<i>Second example: small disturbance</i>	<i>132</i>
7.5.3.	<i>Third example: flow disturbance not localized at a control structure.....</i>	<i>134</i>
7.6.	ASSOCIATED ERRORS DUE TO THE LOSS OF WATER LEVEL MEASUREMENTS	141
7.7.	ASSOCIATED ERRORS DUE TO MEASUREMENT ERRORS IN DEPTH GAGES	144
7.8.	HYDROYNAMIC CANAL STATE	150
7.9.	CONCLUSIONS	154
7.10.	APPENDIX I:	156
CHAPTER 8.....		163
CSI TESTS ON THE EXPERIMENTAL CANAL PAC-UPC.....		163
8.1.	GENERAL DESCRIPTION	163
8.1.1.	<i>Gates</i>	<i>170</i>
8.1.2.	<i>Weirs</i>	<i>170</i>
8.1.3.	<i>Water level sensors</i>	<i>171</i>
8.2.	INITIAL AND BOUNDARY CONDITIONS.....	171
8.3.	DISCRETIZATION OF A PROBLEM	174
8.4.	SCENARIO.....	175
8.5.	TEST RESULTS	179

8.6.	SENSITIVITY OF THE EXTRACTED FLOW IN FRONT OF MANNING COEFFICIENT	181
8.7.	CONCLUSIONS.....	182
CHAPTER 9.....		183
ONLINE PREDICTIVE CONTROL: GOROSOBO.....		183
9.1.	INTRODUCTION	183
9.2.	PROBLEM STATEMENT	191
9.2.1.	<i>Definition of the optimization problem.....</i>	<i>191</i>
9.2.2.	<i>Control problem: Objective function.....</i>	<i>193</i>
9.2.3.	<i>Predictive control</i>	<i>195</i>
9.2.4.	<i>Inputs/outputs variables of a controlled system.....</i>	<i>197</i>
9.2.5.	<i>Analysis of the Hydraulic Influence Matrix: HIM(U).....</i>	<i>197</i>
9.3.	ALGORITHM CONTROLLER FROM UNCONSTRAINED OPTIMIZATION	199
9.3.1	<i>First-order and second-order conditions.....</i>	<i>200</i>
9.3.2	<i>Optimized problem theory</i>	<i>201</i>
9.3.3	<i>Implemented algorithm</i>	<i>201</i>
9.3.4	<i>Analysis of the unconstrained optimization problem</i>	<i>203</i>
9.4.	ALGORITHM CONTROLLER FROM CONSTRAINED OPTIMIZATION.....	205
9.4.1.	<i>Optimized problem theory</i>	<i>209</i>
9.4.2.	<i>Implemented algorithm</i>	<i>209</i>
9.4.3.	<i>Analysis of the Future State Computation</i>	<i>211</i>
9.4.4.	<i>Conclusions of the constrained optimization problem.....</i>	<i>216</i>
9.5.	APPENDIX I	217
9.6.	APPENDIX II	220
9.7.	APPENDIX III	223
9.8.	APPENDIX IV	225
CHAPTER 10.....		231
GOROSOBO APPLICATION: CANAL WITH ONLY ONE POOL		231
10.1.	INTRODUCTION	231

10.2.	CANAL GEOMETRY	232
10.3.	CANAL FEATURES	232
10.4.	PROBLEM SETTING	234
10.5.	THE TARGET VALUE	236
10.6.	CONSTRAINTS AND TRAJECTORIES	237
10.7.	TEST RESULTS	237
10.8.	CONCLUSIONS	257
CHAPTER 11		259
ASCE TEST CASES.....		259
11.1.	CANAL FEATURES	260
11.1.1.	<i>Maricopa Stanfield canal</i>	260
11.1.2.	<i>Corning Canal</i>	262
11.2.	TARGET VALUES	265
11.2.1	<i>Maricopa Stanfield</i>	265
11.2.2	<i>Corning Canal</i>	265
11.3.	UNSCHEDULED FLOW CHANGES	266
11.3.1	<i>Maricopa Stanfield</i>	267
11.3.2	<i>Corning Canal</i>	268
11.4.	CONSTRAINS AND TRAJECTORIES.....	270
11.5.	TEST RESULTS	271
11.5.1	<i>Test-Case 1-1 (Maricopa Stanfield)</i>	271
11.5.2	<i>Test Case 1-2 (Maricopa Stanfield)</i>	275
11.5.3	<i>Test-Case 2-1 (Corning canal)</i>	280
11.5.4	<i>Test Case 2-2 (Corning canal)</i>	284
11.6.	PERFORMANCE INDICATORS.....	288
11.6.1.	<i>Maximum Absolute Error (MAE)</i>	289
11.6.2.	<i>Integral of Absolute Magnitude of Error (IAE)</i>	289
11.6.3.	<i>Steady-State Error (StE)</i>	289
11.6.4.	<i>Integrated Absolute Discharge Change (IAQ)</i>	290
11.7.	COMPARISON BETWEEN CONTROLLERS	290

11.7.1	<i>Test-Case 1-1 (Maricopa Stanfield)</i>	290
11.7.2	<i>Test-Case 1-2 (Maricopa Stanfield)</i>	292
11.7.3	<i>Test-Case 2-1 (Corning canal)</i>	293
11.7.4	<i>Test-Case 2-2 (Corning canal)</i>	293
11.8.	CONCLUSIONS.....	295
CHAPTER 12.....		299
CONCLUSIONS AND FUTURE WORK		299
12.1	SUMMARY OF CONCLUSIONS	299
12.2	CONCLUSIONS OF CSI.....	299
12.3	CONCLUSIONS OF GOROSOBO	300
12.4	FUTURE WORK.....	302
REFERENCES.....		305

List of Figures

Figure 2.1 Evolution of the percent decrease on the overall water resources in Spain (MIMAM,2000).	6
Figure 2.2 The Valencia Water Court (this picture was painted by Bernardo Ferrándiz in 1865, and it is preserved in the Salón Dorado Grande del Palacio de la Generalidad Valenciana (Spain)).....	9
Figure 2.3: Basin irrigation of wheat.	11
Figure 2.4: Center pivot with drop sprinklers.	12
Figure 2.5: Drip irrigation.....	12
Figure 2.6 Survey about productive structure of farms during 2007 in Catalonia (INE).	15
Figure 2.7: Breakage of the cross section of a canal by changes in gradient of pressures (Chimbote canal (Peru)).....	16
Figure 2.8: The ICT office of the master station of Navarra canal (Spain).	18
Figure 3.1: Representation of a system.	22
Figure 3.2: Downstream control concept.	24
Figure 3.3: Downstream control concept.	26
Figure 3.4: Upstream control concept.	26
Figure 3.5: Control of the downstream level of a pool.	28
Figure 3.6: Upstream flow change with the constant downstream depth method of operation.	28
Figure 3.7: Control of the upstream level of a pool.	29
Figure 3.8: Control of an intermediate water level.	30
Figure 3.9: Open-loop control (feedforward).....	32
Figure 3.10: Closed-loop control (feedback).....	33
Figure 3.11: Inputs / Outputs of a controlled system.	35
Figure 3.12: Local controller.....	38
Figure 3.13: Semi-local controller.	38
Figure 3.14: Centralized configuration.	39
Figure 4.1: Display of the values obtained by a computer model of a canal (this displayer was programed in Java code and it was developed for us to check the results obtained with the predictive control).....	44

Figure 4.2: A gate trajectory.	46
Figure 4.3: Display of the values obtained by a computer model of a canal with eight pools with a total length of 28 Km during a predictive horizon of 6000 seconds. The canal is in steady state, therefore the gate position, changes in flow rate, demand deliveries are fixed over the time. The displayer used in this case is the Surfer 8.	48
Figure 4.4: Optimum point in a constraint optimization problem.	52
Figure 4.5: Orifice offtake in a cross section.	54
Figure 5.1: Overall control diagram of irrigation canal.	58
Figure 5.2: The hydrographs at the lateral diversion points of the main canal.	59
Figure 5.3: A DAC used to convert a digital signal to analog.	60
Figure 5.4: The first figure is a ideally sampled signal and the second one is a piecewise constant output of an idealized DAC lacking a reconstruction filter.	60
Figure 5.5: A simple view of an on-line predictive control on a canal.	63
Figure 5.6: A PID controller Block Diagram.	65
Figure 6.1: Flow control structures in the irrigation ditch of Manresa.	67
Figure 6.2: Pumping in a cross-section i.	69
Figure 6.3: Influence of an extraction flow in the canal state.	70
Figure 6.4: Pumping in a cross section i of a computational grid. The extracted flow influences the hydrodynamic variables in blue dots sections.	71
Figure 6.5: Sketch of a numerical grid of a canal with two pools that is controlled by two checkpoints downstream of each pool. There are pump stations close to each checkpoints. Pump flow trajectories are defined with four operation periods. Also, it shows the x/t-dots where the flow behavior is defined. Notice that "K" with capital letter denotes time interval of control and "k" with small letter denotes time instant of simulation.	73
Figure 6.6: Mathematical representation of a pump flow trajectory.	76
Figure 6.7: Minimum value (Q_b^*) for an objective function.	77
Figure 6.8: Direct relation between water level error and extracted flow.	79
Figure 6.9: Input/Output variables in CSI algorithm.	80
Figure 6.10: Types of stationary point.	83
Figure 6.11: The dependence domain of point R.	94
Figure 6.12: The influence domains of points P, Q and R.	95
Figure 6.13: The steps for the interpolation onto a structured grid.	99
Figure 6.14: Interpolation functions.	100
Figure 6.15: A pair of systems of characteristic curves passing through point R.	101

Figure 6.16: Diagram of a checkpoint with gate, lateral weir and pumping.	103
Figure 6.17: Graph with discretization of the control point equations.	104
Figure 6.18: Graphical representation on two canal pools of the time discretization.	107
Figure 6.19: Graph showing the formulation of the second theory of uniqueness for second order sets of hyperbolic equations.	109
Figure 6.20: Graph representing the application of the two uniqueness theorem to the boundaries.	109
Figure 6.21: The evolution of the influence of the pump flow through time and space. The grey dots correspond to the points where there is no influence of the pump flow: $\partial \mathbf{X}_i \mathbf{K} \partial \mathbf{q}_b(\mathbf{K}) = \mathbf{0}$ and the blue dots where there is influence of the pump flow: $\partial \mathbf{X}_i \mathbf{K} \partial \mathbf{q}_b(\mathbf{K}) \neq \mathbf{0}$	114
Figure 6.22: An example of a pump parameter which has no influence on any point.	120
Figure 7.1: Canal profile.	126
Figure 7.2: Backwater profile of the canal in the initial steady state.	128
Figure 7.3: Water level measured at checkpoint 1 and 2 during the past time horizon in the first example.	131
Figure 7.4: Pump flow at checkpoint 1 and 2 during the past time horizon in the first example.	132
Figure 7.5: Water level measured in checkpoint 1 and 2 during a past time horizon in the second example.	133
Figure 7.6: Pump flow in checkpoint 1 and 2 during a past time horizon in the second example.	134
Figure 7.7: Water level measured in checkpoint 1 and 2 during a past time horizon in the third example.	135
Figure 7.8: Pump flow in checkpoint 1 and 2 during a past time horizon in the third example.	137
Figure 7.9: Pump flow in checkpoint 1 and 2 during a past time horizon in example 3a.	138
Figure 7.10: The wave generated by the disturbance traveling upstream and downstream from P.K. 3+750.	140
Figure 7.11: Pump flow in checkpoint 1 and 2 during a past time horizon.	142
Figure 7.12: Water level measurements V.S. simulated water levels by CSI at checkpoint 1 and 2 during the past time horizon.	143
Figure 7.13: Water level measurements at checkpoint 2 V.S. simulated water level by CSI at checkpoint 2 during the past time horizon.	144
Figure 7.14: Water level measured at checkpoint 1 and 2 during a past time horizon (First example: Case: +2 cm).	145

Figure 7.15: Water level measured at checkpoint 1 and 2 during a past time horizon (First example: Case:-2cm).	145
Figure 7.16: Water level measured at checkpoint 1 and 2 during a past time horizon (First example: Case:-/+2cm).	146
Figure 7.17: Pump flow at checkpoint 1 and 2 during the past time horizon (First example: Case:+2cm).	147
Figure 7.18: Pump flow at checkpoint 1 and 2 during the past time horizon (First example: Case:-2cm).	147
Figure 7.19: Pump flow at checkpoint 1 and 2 during the past time horizon (First example: Case:-/+2cm).	148
Figure 7.20: Water level measured at checkpoint 1 and 2 during the past time horizon (Third example: Case:+2cm).	149
Figure 7.21: Pump flow at checkpoint 1 and 2 during the past time horizon (Third example: Case:-/+2cm).	150
Figure 7.22: Water level measurements in the first example (pointed at 2700 s and 3900s).	151
Figure 7.23: Water profile obtained by a computer model V.S. Water profile obtained by CSI at 2700s.	152
Figure 7.24: Water profile obtained by the V.S. Water profile obtained by CSI at 3900s.	153
Figure 8.1: Schematic diagram of the top view of the Canal PAC-UPC.	164
Figure 8.2: Schematic diagram of the profile view of the Canal PAC-UPC.	165
Figure 8.3: General view of the Canal PAC-UPC.	166
Figure 8.4: Profile view for the geometric configuration of the Canal PAC-UPC in the test.	168
Figure 8.5: Schematic diagram of the top view of a canal bend of the Canal PAC-UPC.	169
Figure 8.6: Photo of a weir of the canal PAC.	171
Figure 8.7: A submerged hydraulic jump.	172
Figure 8.8: A sharp crested weir.	173
Figure 8.9: Backwater profile for the canal from initial conditions (steady state).	174
Figure 8.10: Water level measures in sensor L7.	176
Figure 8.11: Flow through the weir 2 using the weir equation and the discharge coefficient calibrated by Horváth (2013).	177
Figure 8.12: Water level measures in sensors L6, L10 y L11. The measures were obtained every second and calculating the average value per 10 seconds.	177
Figure 8.13: Water level measures in sensor L1 (upstream reservoir). The measures were obtained every second and calculating the average value per 10 seconds.	178

Figure 8.14: The extracted flow rate through the weir 2 by CSI using 10-period simple moving average V.S. the extracted flow rate through the weir 2 by the weir equation using the discharge coefficient calibrated by Horváth (2013).....	179
Figure 8.15: The extracted flow rate at weir (W2) modifying the Manning roughness coefficients.	181
Figure 9.1: Developed algorithms in this thesis in an overall control diagram of irrigation canal. This cycle is done in each operation period and GoRoSoBo calculates the optimum gate trajectories in each operation period for a predictive horizon.	184
Figure 9.2: Ideal case for a predictive control which maintains a water level to satisfy a scheduled demand of 4 m ³ /s.....	186
Figure 9.3: Real case for a predictive control. The controller calculates a gate trajectories to maintain a water level and supply the scheduled demand of 4 m ³ /s. The problem is when someone introduce a disturbance of 2 m ³ /s. The discharge through offtakes would be reduced to 2 m ³ /s.	187
Figure 9.4: Logic of control in our overall control diagram of irrigation canal.	188
Figure 9.5: General view of the influence of a gate movement in the hydrodynamic state of a canal.....	190
Figure 9.6: Influence of a gate movement in several points of the space/time domain in the canal disposed in a computational grid. Notice that "K" with capital letter denotes time interval of control and "k" with small letter denotes time instant of simulation.	190
Figure 9.7: Mathematical representation of a gate trajectory.	192
Figure 9.8: Sketch of a numerical grid in a canal with two pools controlled by two checkpoints downstream each pool. There are two sluice gates. Gate trajectories are defined with four operation periods. The x/t-dots where the flow behavior is defined are shown.....	193
Figure 9.9: Minimum value (U*) for an objective function.	194
Figure 9.10: A model predictive control scheme.....	195
Figure 9.11: Logic of control of the feedback algorithm.....	196
Figure 9.12: Representation of the feedback algorithm.....	197
Figure 9.13: Flow disturbance during a regulation period V.S. water level disturbance at a particular time step.	203
Figure 9.14: Maximum and minimum gate position for the constrained optimization problem.	206
Figure 9.15: Maximum gate movement between consecutive regulation periods for the constrained optimization problem.	207

Figure 9.16: Maximum gate movement between consecutive regulation periods for different predictive horizons for the constrained optimization problem.	208
Figure 9.17: Logic of control of the GoRoSoBo algorithm.	212
Figure 9.18: Associated problems to disturbances with long duration in GoRoSoBo not using future prediction.....	213
Figure 9.19: Variation of the canal state taking into account different predictive horizons...	215
Figure 10.1: Simple canal with only one pool.....	232
Figure 10.2: Backwater curve of the canal (Initial condition at the first regulation period of the irrigation cycle).	234
Figure 10.3: Time discretization of a test in the canal.....	236
Figure 10.4: Scheduled and unscheduled deliveries introduced in the System (Case 1).....	238
Figure 10.5: Disturbance calculated by CSI (Case 1).....	238
Figure 10.6: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 1.	240
Figure 10.7: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 1.....	240
Figure 10.8: Scheduled and unscheduled deliveries introduced in the System (Case 2).....	241
Figure 10.9: Disturbance calculated by CSI (Case 2).....	241
Figure 10.10: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 2.	243
Figure 10.11: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 2.....	243
Figure 10.12: Scheduled and unscheduled deliveries introduced in the System (Case 3)...	244
Figure 10.13: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 3.	245
Figure 10.14: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 3.....	245
Figure 10.15: Scheduled and unscheduled deliveries introduced in the System (Case 4)...	246
Figure 10.16: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 4.	247
Figure 10.17: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 4.....	247
Figure 10.18: Scheduled and unscheduled deliveries introduced in the System (Case 5)...	248
Figure 10.19: Disturbance calculated by CSI (Case 5).....	248

Figure 10.20: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 5.	249
Figure 10.21: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 5.	250
Figure 10.22: Scheduled and unscheduled deliveries introduced in the System (Case 6) ..	250
Figure 10.23: Disturbance calculated by CSI (Case 6).....	251
Figure 10.24: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 6.	252
Figure 10.25: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 6.	252
Figure 10.26: Scheduled and unscheduled deliveries introduced in the System (Case 7) ..	253
Figure 10.27: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 7.	254
Figure 10.28: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 7.	254
Figure 10.29: Enclosed area V.S. Water level at the checkpoint obtained by GoRoSoBo (orange continuous line) in Case 7.	254
Figure 10.30: Scheduled and unscheduled deliveries introduced in the System (Case 8) ..	255
Figure 10.31: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 8.	256
Figure 10.32: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 8.	256
Figure 11.1: Maricopa Stanfield profile. The red lines mark the position of checkpoints. The first pool is number I and the first checkpoint is number 1.....	261
Figure 11.2: Corning Canal profile. The red lines mark the position of checkpoints. The first pool is number I and the first checkpoint is number 1.	263
Figure 11.3: Demanding extracted flow, extracted flow by CSI and real extracted flow in Test-Case 1-1 (Maricopa Stanfield).	272
Figure 11.4: Gate trajectories (1, 3, 5 and 8) in Test-Case 1-1 (Maricopa Stanfield).....	273
Figure 11.5: Gate trajectories (2, 4, 6 and 7) in Test-Case 1-1 (Maricopa Stanfield).....	274
Figure 11.6: Water level at checkpoints 1, 3, 5 and 8 in Test-Case 1-1 (Maricopa Stanfield).	274
Figure 11.7: Water level at checkpoints 2, 4, 6 and 7 in Test-Case 1-1 (Maricopa Stanfield).	275

Figure 11.8: Demanded extracted flow, extracted flow by CSI and real extracted flow in Test-Case 1-2 (Maricopa Stanfield).....	277
Figure 11.9: Gate trajectories (1, 3, 5 and 8) in Test-Case 1-2 (Maricopa Stanfield).	278
Figure 11.10: Gate trajectories (2, 4, 6 and 7) in Test-Case 1-2 (Maricopa Stanfield).	278
Figure 11.11: Water level at checkpoints 1, 3, 5 and 8 in Test-Case 1-2 (Maricopa Stanfield).	279
Figure 11.12: Water level at checkpoints 2, 4, 6 and 7 in Test-Case 1-2 (Maricopa Stanfield).	279
Figure 11.13: Demanded extracted flow, extracted flow by CSI and the real extracted flow in Test-Case 2-1 (Corning canal).	281
Figure 11.14: Gate trajectories (1, 2, 4 and 6) in Test-Case 2-1 (Corning canal).....	282
Figure 11.15: Gate trajectories (3, 5, 7 and 8) in Test-Case 2-1 (Corning canal).....	282
Figure 11.16: Water level at checkpoints 1, 2, 4 and 6 in Test-Case 2-1 (Corning canal). ...	283
Figure 11.17: Water level at checkpoints 3, 5, 7 and 8 in Test-Case 2-1 (Corning canal). ...	283
Figure 11.18: Demanded extracted flow, extracted flow by CSI and the real extracted flow in Test-Case 2-2 (Corning canal).	285
Figure 11.19: Gate trajectories (1, 2, 4 and 6) in Test-Case 2-2 (Corning canal).....	286
Figure 11.20: Gate trajectories (3, 5, 7 and 8) in Test-Case 2-2 (Corning canal).....	287
Figure 11.21: Water level at checkpoints 1, 2, 4 and 6 in Test-Case 2-2 (Corning canal). ...	287
Figure 11.22: Water level at checkpoints 3, 5, 7 and 8 in Test-Case 2-2 (Corning canal). ...	288

List of Tables

Table 6.1: summarize of the whole procedure.....	112
Table 7.1: General canal features.....	127
Table 7.2: Checkpoints and Sluice gate/ pump station/ orifice offtake features (control structure).....	127
Table 7.3: Initial conditions in the canal.....	128
Table 7.4: Flow delivered by an orifice offtake at the initial time step.....	128
Table 7.5: Water volume error between the simulated and real flow hydrographs in the third example.....	139
Table 7.6: Water volume error between the simulated and real flow hydrographs.....	142
Table 7.7: Water volume error between the simulated and real flow hydrographs from the minute 65 to minute 160 (Third example: Case +2cm).....	150
Table 7.8: Water velocity obtained by the computer model (real disturbance) V.S. Water velocity obtained by CSI along the canal in the first example at 2700 s and 3900 s.....	156
Table 8.1: Features of the canal pools.....	167
Table 8.2: Physical parameters on each canal bend.....	170
Table 8.3: features of the gate.....	172
Table 8.4: Features of the weir.....	173
Table 8.5: Initial conditions (water level at particular points).....	174
Table 8.6: The disturbance characteristics introduced to the system.....	176
Table 10.1: Features of the canal control structures 1.....	233
Table 10.2: Features of the canal control structures 2.....	233
Table 10.3: The scheduled and unscheduled deliveries introduced in the cases tested.....	235
Table 10.4: Overall and functional constraints values.....	237
Table 11.1: Features of Maricopa Stanfield pools.....	261
Table 11.2: Maricopa Stanfield control structures.....	261
Table 11.3: Initial conditions Test Case 1-1.....	262
Table 11.4: Initial conditions Test Case 1-2.....	262

Table 11.5: Features of Corning canal pools.	263
Table 11.6: Corning Canal control structures.....	263
Table 11.7: Initial conditions Test Case 2-1.	264
Table 11.8: Initial conditions Test Case 2-2.	264
Table 11.9: Desired water level Test Case 1-1/1-2.....	265
Table 11.10: Desired water level Test Case 2-1/2-2.....	265
Table 11.11: Unscheduled offtake changes on Test Case 1-1.	267
Table 11.12: Unscheduled offtake changes on Test Case 1-2.	268
Table 11.13: Unscheduled offtake changes on Test Case 2-1.	269
Table 11.14: Unscheduled offtake changes on Test Case 2-1.	269
Table 11.15: Overall and functional constraints values.	270
Table 11.16: Unscheduled offtake changes in test-case 1-1 (Maricopa Stanfield).....	271
Table 11.17: Unscheduled offtake changes in test-case 1-2 (Maricopa Stanfield).....	275
Table 11.18: Unscheduled offtake changes in test-case 2-1 (Corning canal).	280
Table 11.19: Unscheduled offtake changes in test-case 2-1 (Corning canal).	284
Table 11.20: The performance indicators obtained in Test-Case 1-1 (Maricopa Stanfield).	292
Table 11.21: The performance indicators obtained in Test-Case 1-2 (Maricopa Stanfield).	292
Table 11.22: The performance indicators obtained in Test-Case 2-1 (Corning canal).	293
Table 11.23: The performance indicators obtained in Test-Case 2-2 (Corning canal).	295

Nomenclature

α	Index defined in the active set method;
Δ	The divergence of a vector field's source or sink;
λ_k	Lagrange multipliers of active constraints of the quadratic problem at iteration k ;
λ	Lagrange's multipliers vector at iteration k ;
Λ	Diagonal matrix with the eigenvalues of the G matrix;
ν	Marquardt coefficient;
γ	Prediction horizon in CSI;
ϕ	Prediction horizon in GoRoSoBo;
ψ	Parameter to denote variables of flow description such as depth, velocity;
A	Matrix of dimension $(2 \times n_s) \times (2 \times n_s)$, obtained defining the HIM(Q);
$A(y)$	The area of the wet section which depends on the water level “ y ”;
a_c	The gate width;
b	Vector of dimension $(2 \times n_s)$, $b[x^{k+1}, q(K)]$ obtained defining the HIM(Q);
b_j	The j -gate width;
$c(y)$	Water wave celerity which is dependent on the water level “ y ”;
C	Matrix of dimension $n_y \times n_x$ called discrete observer matrix by Malaterre;
C_c	Contraction coefficient of the gate;
C_d	Discharge coefficient of the gate;
c_T	Local losses coefficient in the canal;
C_w	Discharge coefficient of a weir;
d_j	Drop at j -gate;
$d_0 U_{\max}$	Maximum gate movement allowed between two consecutive operation periods;

dU_{\max}	Maximum gate movement allowed between two consecutive operation periods of different prediction horizons;
ΔU	Perturbed gate trajectory vector;
ΔQ_b	Perturbed input/output flow vector;
Δt	Numerical discretization time according to CFL condition;
ΔT	Operation period defined by the watermaster;
ΔY	Water level perturbation;
$\Delta \Delta U$	Auxiliary solution-vector (gate) of the quadratic problem;
Δx	Numerical discretization space cell length;
ΔX	Perturbed state vector obtained when a perturbation is introduced into the system;
$error_{k_i}^{k_f}$	Error between the computed and desired water level at a checkpoint;
$f(k)$	Input function at time step k ;
g	Gravity;
G_{it}	Performance index Hessian approximation at iteration it ;
$g_i(X)$	Inequality constraints;
$G(Q_b)$	A set of feasible directions for the extracted flow vector;
G	Matrix obtained multiplying the transposed "J" matrix by the "J" matrix;
$G(U)$	A set of feasible directions for the gate trajectories vector;
g_{it}	Auxiliary Jacobian of the Lagrange's function at iteration it ;
H_{it}	Auxiliary Hessian approximation of the Lagrange's function at iteration it ;
$HIM(Q_b)$	Hydraulic influence matrix (derivative parameter Q_b);
$HIM(U)$	Hydraulic influence matrix (derivative parameter U);
H_{up}	Constant upstream water level at upstream boundary condition (canal header);
I	Identity matrix;
$[I_M(Q_b)]_j$	The hydraulic influence on the state vector of a certain pump station "j";
$[I_M(U)]_j$	The hydraulic influence on the state vector of a certain gate "j";

$J(Q_b)$	Performance criterion (input/output flow trajectory) (objective function);
$J(U)$	Performance criterion (gate trajectory) (objective function);
J	Matrix obtained deriving the output water level vector by a variable;
k_F	Number of sections in which the prediction horizon is divided depending on the CFL condition/ time frequency of measuring data;
K_F	Number of operation periods defined by the watermaster in which the simulating horizon is divided;
K_p	Proportional gain, a tuning PID parameter;
K_i	Integral gain, a tuning PID parameter;
K_d	Derivative gain, a tuning PID parameter;
$k_o(k)$	Orifice coefficient depending on its overture of the gravity i-offtake at time instant k ;
L_n	Water level from sensor called L_n ;
L	Lower triangular matrix set to Cholesky factorization
L'	Matrix with the direct influence from inlet-outlet flow discharge over the canal
$l_j(X)$	Equality constraints;
\mathcal{L}	Lagrangian function;
M	Matrix obtained from the Saint Venant equation to represent the influence between parameters;
N	Matrix obtained from the Saint Venant equation to represent the influence between parameters;
n	Manning coefficient;
n_c	Number of checkpoints;
n_g	Number of gates;
n_{HX}	Number of rows of the hydraulic influence on the state vector of a certain gate “j”;
n_{HY}	Number of columns of the hydraulic influence on the state vector of a certain gate “j”;
n_p	Number of pump stations;
n_q	Dimension of inlet-outlet flow vector ($n_q = n_p \times K_F$);
n_s	Number of sections in which the canal is discretized;
n_u	Dimension of control trajectory vector ($n_u = n_g \times K_F$);
n_x	Dimension of prediction vector ($n_x = (2 \times n_s) \times k_F$);

n_y	Dimension of prediction output vector ($n_y=k_F \times n_c$);
$O_{k_1}^{k_F}$	Error between the computed and desired flow rate at a checkpoint;
$P(y)$	The wet perimeter which is function of the water level “y”;
Q_b	Input/output flow vector;
Q'	Weighting matrix;
Q_b^*	Solution of the input/output flow vector;
\bar{Q}_b	The desired input/output flow vector;
$q_b(K)$	inlet-outlet flow at the control structure during the operation period K;
Q_j	Undershot j-gate flow;
Q_{wn}	Discharge over the weir n;
Q_{offtake}	Discharge through the orifice offtake;
q_s	Discharge through the lateral spillway;
R_H	The hydraulic radius is a measure of a channel flow efficiency;
$r_k(u)$	Constraint function “k” apply to the gate trajectory;
$r_{k_1}^{k_F}$	Residual vector between the desired water level and the computed water level;
S	Matrix obtained from the Saint Venant equation to represent the influence between parameters;
$S(y)$	Horizontal surface of the reception area in the checkpoint;
S_0	Bottom slope;
S_f	Friction slope;
$s_{k_1}^{k_F}$	Error between the computed and the minimum operating cost gate trajectory at a checkpoint;
$s(k)$	Input data vector at time step k;
T^K	Operation instant when the inlet-outlet flow are changed;
$T(y)$	The maximum width dependent on the water level “y”;
$u_{ij}(K)$	Gate-opening of the gate i during the operation period K at the prediction horizon j;

U	Gate trajectory vector;
U^*	Solution of the gate trajectory vector;
\bar{U}	The desired gate trajectory vector;
U_0	Vector with the gate positions at the first regulation period for the previous predictive horizon;
U_{\max}	Maximum gate position allowed;
U_{\min}	Minimum gate position allowed;
$U_{ij}(k)$	Gate-opening of the gate i at the predictive horizon j for the regulation period k ;
$v_i(k)$	Mean velocity at time instant k at canal section i ;
V	Orthonormal matrix defined by the eigenvectors of G ;
$W(Q_b)$	Hessian matrix obtained deriving the variable Q_b (pump flow vector);
$W(U)$	Hessian matrix obtained deriving the variable U (gate trajectories vector);
W_n	Weir number “ n ”;
$x(k)$	State vector at time instant k ;
$\mathbf{x}_1^{k_F}$	Prediction vector from 1 to k_F ;
$y(k)$	Subset of water depths of the state vector at time instant k at checkpoints;
$\mathbf{y}_1^{k_F}$	Prediction output vector from 1 to k_F ;
$y_M(k)$	Subset of measurement water levels at checkpoints at time instant k ;
$\mathbf{Y}_{M_1}^{k_F}$	Measurement output vector;
y_{dw}	Downstream j -gate water depth;
$y_i(k)$	Water depth at time instant k at canal section i ;
y_o	Height of the center of the orifice from bottom of the gravity offtake;
y_s	Downstream water level of the reservoir;
y_{up}	Upstream j -gate water depth;
z	Variable used to interpolate values with the Lagrange factors;

Abbreviations

ASCE	American Society of Civil Engineers;
AEMET	Agencia Estatal de Meteorología;
AR4	Rossby centre regional atmospheric model;
BIVAL	Constant downstream volume system;
Canal PAC-UPC	Canal de Prueba de Algoritmos de Control - Universitat Politècnica de Catalunya;
CACG	Compagnie d'Aménagement des Coteaux de Gascogne;
CARA	Compagnie d'Aménagement Rural d'Aquitaine;
CARDD	Canal Automation for Rapid Demand Deliveries;
CEMAGREF	Centre National du Machinisme Agricole, du Génie Rural, des Eaux et des Forêts;
CFL	Courant Friedrichs Levi;
CLIS	Constant Level control method based on the Inverse Solution of the Saint Venant equations;
CNABRL	Compagnie Nationale d'Aménagement du Bas-Rhône Languedoc;
CPU	Central Processing Unit;
CSI	Canal Survey Information;
ELFLO	Electronic Filter and Level Offset;
FAO	Food and Agriculture Organization of United Nations;
HEC-RAS	Hydrologic Engineering Center - River Analysis System;
HIM	Hydraulic Influence Matrix;
HIRLAM	High Resolution Limited Area Model;
GN	Gauss Newton;

GoRoSo	Gómez, Rodellar and Soler (feedforward algorithm);
GoRoSoBo	Gómez, Rodellar, Soler and Bonet (feedback algorithm);
GPC	Generalized predictive control method;
IAE	Integral of Absolute Magnitude of Error;
IAQ	Integrated Absolute Discharge Change;
ICT	Information and Communications Technology;
ID	Integrator Delay;
INE	Instituto Nacional de Estadística;
LQ	Linear Quadratic;
LQG	Linear Quadratic Gaussian;
LQR	Linear Quadratic Regulator;
LM	Levenberg Marquardt;
LN	Lagrange Newton;
MAE	Maximum Absolute Error;
MCG	Modelos de circulación General;
MCR	Modelos climáticos Regionales;
MIMO	Multiple-Input Multiple-Output;
MISO	Single-Input Single-Output;
MMA/MIMAM	Ministerio de Medio Ambiente;
MPC	Model Predictive Control;
PC	Predictive Control;
PILOTE	Preissmann Implicit scheme, Linear Optimal control, Tracking of variables and Estimation of perturbations;
PI	Proportional Integral;
PID	Proportional Integral Derivative;
PIR	Proportional Integral Retard;
P.K.	The distance along a axis from the origin point;

PNACC	Plan Nacional de Adaptación al Cambio Climático;
PRECOM	Multivariable Predictive Controller;
PREDAPTROL	Predictive & Adaptive control;
PREDTROLLER	Predictive controller;
QP	Quadratic Programming;
SACMAN	Software for Automated Canal Management;
SCADA	Supervisory Control And Data Acquisition;
QP	Quadratic Programming;
SIC	Simulation of Irrigation Canals;
SIMO	Single-Input Multiple-Output;
SISO	Single-Input Single-Output;
STE	Steady state error;
UMA	PID developed by UMA engineering;
UNDESA	United Nations Department of Economic and Social Affairs;

Chapter 1

Layout of thesis

An irrigation canal is a hydraulic system whose main objective is to convey water from a source (dam, river) to different users. Such systems can be very large (several tens or hundreds of kilometers), characterized by time delays and non-linear dynamics, strong unknown perturbations and interactions among subsystems. Varying operational objectives are assigned to their managers. The main general one is to provide water to the different users at the right moment and in the right quantity, reducing losses as much as possible to guarantee the safety of the infrastructure. In particular, a major concern is to prevent the canals from overtopping, but also from having water levels inside the pools below the supply depths of the gravity offtakes.

One of the most important problems for an on-demand irrigation system designer is the calculation of the discharges flowing into the network. Such discharges strongly vary over time depending on the cropping pattern, meteorological conditions, on-farm irrigation efficiency and farmers' behavior. The amount of water to be supplied during an irrigation event, referred to as the target or required depth of application, is a major design consideration.

The unknown changes in water demand are quite usual in a canal during an irrigation cycle. The result is that the real extracted flow is unknown by the watermaster. In that case, it is quite difficult to recompute a new gate trajectories to satisfy the scheduled deliveries when the watermaster does not know the real extracted flow and the current canal state. For this reason, the motivation of this thesis and the main objective are two. On one hand, developing an algorithm to establish the disturbances and the current canal state (CSI algorithm), that is, an algorithm that calculates the velocity and water level in each cross-section of the canal from water depth lectures at some canal sections during a past time horizon. On the other hand, a predictive control algorithm (GoRoSoBo algorithm) to establish the gate trajectories in real time from the current canal state. In this way, the watermaster could fulfil the real water demands in the canal.

The thesis is divided into 13 chapters:

In the **second chapter**, we first talk about estimations about agricultural water consumption, and define the main problems generated by a low efficiency in water uses. To fix the problems related with the low efficiency, it is necessary to point out the main guiltest as they are the water distribution infrastructure (the transport and the techniques of application of water in the parcel), the water prices and the structural productive, that is, the economic dimensions that irrigated farms have. To finalize the chapter, we define the concept of diagnosis and prognosis in canal control, we will check the efficiency in water uses obtained by control algorithms in the diagnosis, and we propose mechanisms to improve the results of these algorithms in the prognosis.

In the **third chapter**, we introduce the literature review about the canal control algorithms. We define important control concepts which aid us to introduce a classification of control algorithms in irrigation canals. Each of these algorithms have been characterized according to several criteria. We follow the criteria established by Malaterre (1998). We classify these algorithms by the considered variables (controlled, measured, control action variables), the logic of control (type and direction), the design technique and the field implementation.

In the **fourth chapter**, we make a general introduction about the main characteristics of control algorithms and we define their main elements. We give our particular point of view of predictive control and introduce our computer model and our optimization problem, as well as, the main elements of these ones.

In the **fifth chapter**, the modernization of irrigation projects is achieved with a general improvement in the water transport and distribution networks and the field application techniques with a view to enhancing the efficiency of water use, which is extremely low. Such an overall improvement in efficiency can occur at three different levels: transport through the main canal, distribution through the secondary network and field application. In our opinion, improving water use at these three levels must be tackled at each level individually and at all three together. In line with predictive control theory (Martín-Sánchez and Rodellar, 1996), we define in this chapter, the different calculation steps that should be included in our overall irrigation canal control process, as the crop needs and outlet hydrographs, parameter identification, off-line reference trajectory, canal current state and on-line predictive control.

In the **sixth chapter**, we introduce the CSI algorithm (Canal Survey Information algorithm) which sets the real extracted flow and the hydrodynamic state in a canal at any moment from water depth lectures at some canal sections. We develop this algorithm for several reasons: it is an useful tool to get information about the disturbances and it is an useful supplementary tool for a feedback controller, because that algorithm establishes the past and current hydrodynamic state of a canal, that is, the water level and flow rate at every cross-section of the canal from the past to the present time. To develop this algorithm, we also had to develop the hydraulic influence matrix, which establishes the influence of flow changes (for instance, one pump running) on the canal state vector.

In the **seventh chapter**, the CSI algorithm is tested through some numerical examples. We define a main canal system consisting of two canal pools and we test the CSI algorithm in this canal in three different tests. We also did a sensitivity analysis to evaluate CSI when we obtain errors in water depth measurements from the canal sensors.

In the **eighth chapter**, we test CSI algorithm in a laboratory canal. The laboratory canal used is the Canal PAC-UPC designed to develop basic and applied research in irrigation canals control area and in all subjacent areas like canal instrumentation, canal modelling and water measurements. The CSI algorithm is tested in the canal and we present the results. We introduce several tests to study the sensitivity of the algorithm versus changes in physical parameters as the Manning coefficient.

In the **ninth chapter**, we introduce and develop the theory of the proposed feedback algorithm, GoRoSoBo (Gómez, Rodellar, Soler and Bonet). GoRoSoBo is a predictive control algorithm that keeps the target level at several sections of the canal, so that the flow through the offtakes are the scheduled demands, in case of unknown flow changes in the canal. Any water level deviation from the desired setpoint is sent to the CSI algorithm which computes the real extracted flow and the current canal state and this data is sent to GoRoSoBo in order to produce a corrective action (new gate trajectories). To develop this algorithm, we used the hydraulic influence matrix, which establishes the influence of a gate movement on the canal state vector, introduced by Soler (2003). On the other hand, we define the algorithms developed for us in each block of our overall control scheme introduced at chapter 5.

In the **tenth chapter**, The GoRoSoBo algorithm is tested by several numerical examples. We define a main canal system consisting of one canal pool and we test the GoRoSoBo algorithm in this canal in eight different tests.

In the **eleventh chapter**, the GoRoSoBo algorithm is applied to the “test cases” examples defined by the ASCE Task Committee on Canal Automation Algorithm. We analyze the results obtained with GoRoSoBo and compute the performance indexes proposed by the ASCE. We compare the performance indicators obtained with GoRoSoBo with the performance indicators estimated by others centralized predictive controllers as CLIS (Liu et al. 1998) and Pilote (Malaterre et al. 1995).

In the **last chapter**, we introduce the conclusions of the thesis and the future works.

Chapter 2

Introduction and general statements

The world population is predicted to grow from 6.9 billion in 2010 to 8.3 billion in 2030 and 9.1 billion in 2050 (UNDESA, 2009) and the food demand is predicted to increase by 50% in 2030 and by 70% in 2050 (Bruinsma, 2009).

The most recent estimates for irrigated agriculture is an increase in comparison with the 2008, from 2,743 km³ in 2008 to 3,858 km³ in 2050 (FAO, 2011a,b). Much of the increase in irrigation water will be in regions already suffering from water scarcity. To study the impact of water scarcity, there are available several simulation models, for instance MCG (General circulation model) or MCR (regional climate model). In case of Spain, some of these models (AR4 (Rossby Centre regional atmospheric model) and HIRLAM (High Resolution Limited Area Model)) predict an increase of 1 °C in temperature and a decrease of 5% in precipitation in 2020, and as result a decline in water resources from these areas of 10% according to AEMET (2009)(Agencia Estatal de Meteorología). These values could increase significantly according to these studies which calculate a decrease of 20% in precipitation and increase of 3 °C in temperature in 2050 according to the PNACC (2011).

In the next Figure 2.1, we can show the consequences of water scarcity in two scenarios for several catchments in Spain. The first scenario represents simulations with an increase of 1°C in temperature, in one of those scenarios without change in precipitation and in the other scenario with a decrease of 5% in precipitation.

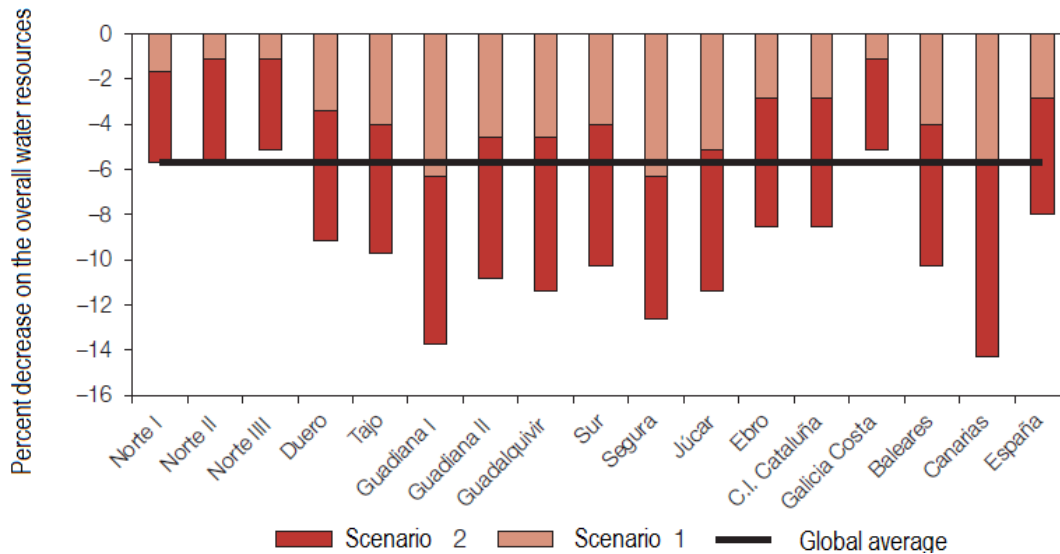


Figure 2.1 Evolution of the percent decrease on the overall water resources in Spain (MIMAM,2000).

Taking into account these estimations, the main challenge facing the agricultural sector is making a 70% more food available with less water resources. More water resources or a more efficient use of the resources will be needed to fit this objective. We cannot waste the water resources and therefore we are forced to improve the current irrigation technics.

2.1 Definition of the irrigation problem

The operational aspects of farm irrigation and water supply systems, in areas still dominated largely by tradition, do not usually reflect a high degree of water use efficiency as a primary objective. This efficiency, expressed as the ratio between the quantities of irrigation water effectively used by the crops and the total quantities supplied, has only been considered as an important factor in irrigation during the last 60 to 65 years. This is not really surprising because up to 80 years ago our knowledge of the water requirements of crops, more specifically those of evapotranspiration, was only vague and water resources investigations of irrigated areas were not yet receiving as much attention as today. Moreover water was available and the price of the water was low or even free.

There are three physical characteristics which govern any irrigation operation, in terms of both quantity and time:

- The evapotranspiration of the various crops cultivated and this changes during the growing season.
- The moisture content of the soils between field capacity and a preselected depletion limit (that is, the lowest acceptable moisture content that does not significantly affect yields)
- The infiltration rate of the relevant soils.

Other physical factors such as rainfall distribution, topography, and canal seepage may, of course, play a role, but the above three characteristics must be considered under all circumstances.

Together, all these factors must serve as a basis for defining such operational features as depth, duration, and interval of irrigation for different crops and soils. But even with this information available, it is only possible to predict the overall irrigation efficiency within an accuracy of 15 per cent (Bos et al., 1990 and 2005).

The lack of basic knowledge of water use efficiencies has several serious drawbacks:

- Due to the low efficiency in the planning and design of irrigation systems, a large safety margin is applied.
- Investments are thus considerably higher than would otherwise be necessary.
- The limited water resources are not optimally distributed and used, much water is wasted and less area can be irrigated.
- The low overall irrigation efficiency creates harmful side-effects such as rising groundwater tables and soil salinization. To control the groundwater table a costly subsurface drainage system may be necessary and this will seriously affect the economy of the project.

These irrigation problems are related with the efficiency in water uses but the irrigation problems are also related with an old infrastructure which is responsible of water losses due to cracks of the concrete of the canal side walls.

2.1.1 Conditions involved in reducing the efficiency

There are three ways to make better the efficiency of water used in the crops according to Oca (2009).

The first one is to improve the efficiency in the application of water to the crop by the farmer, the infrastructure and the modes of **water distribution** used by the irrigation communities to convey and distribute the water. The second one is acting on **water price**, once analyzed the socio-economic effects of increasing or decreasing the water price, in order to encourage the farmers to save water. A third area that we cannot ignore, it is the possibility to actuate on the **structure of agricultural production** in irrigated areas, because if the profit margin is low, the farmer does not have any chance to adopt new technologies which improve the efficiency of water use.

These three areas of actuation are not independent at all, because we can build infrastructures to improve the efficiency of water distribution but the farmers must adopt new technologies to make a good use of water resources and save money, and all of these things are strongly conditioned by the productive structure.

2.1.1.1 Water distribution: rotational irrigation scheduling or on-demand irrigation scheduling

The water distribution from turnout structures requires infrastructure for conveying the water, but the final water distribution to the farmer depends on complex organizational structures that can adopt different forms of distribution. The water distribution system can influence the technique of irrigation and can be used to encourage water saving.

The irrigation communities are usually responsible for water distribution and they also manage the maintenance of canal networks, the main and secondary canals. The irrigation communities are institutions with a long tradition in Spain and in other countries, and their role is distribution of irrigation water. The oldest democratic governance institution in Europe designed to guarantee the correct functioning of a complex network of irrigation canals and safeguard the interests of irrigation communities is the Valencia Water Court (Tribunal de las Aguas de Valencia). This court settles conflicts between irrigation users on an impartial basis and is made up of democratically elected farmers experts in uses and customs, and fair in their proceedings.

The Valencia Water Court (Figure 2.2) has survived over the centuries to the present day and it was integrated into the Spanish legal system, with the same guarantees and legal value as any civil court. This is due to its effective contribution to the maintenance of the vast and complex system of irrigation canals used for Valencia's

fertile plains, built in the Andalusí age (9th-13th Century). That speaks of the great importance of the water distribution in Spain due to the dry climate of the country.



Figure 2.2 The Valencia Water Court (this picture was painted by Bernardo Ferrándiz in 1865, and it is preserved in the Salón Dorado Grande del Palacio de la Generalidad Valenciana (Spain)).

In a general way, we could classify the modes of water distribution depending on the degree of flexibility in water use by farmers.

The **rotational irrigation scheduling** is the most restrictive of all irrigation systems. The rate, frequency, and duration are fixed by policy of the central water authority and remain fixed for the entire irrigation season. There are a number of possible variations of a rotational irrigation scheduling that allow the farmer fulfill the crop needs better. Several methods are presented herein:

- Continuous flow schedules are a special case of rotation systems, where the duration is a fixed period, the rate is constant and the frequency remains fixed during the growing season. Rotation and continuous flow systems are used in places with plenty of water, the growing season is short, and the economics of efficient irrigation for maximum yield is not taken into account. In many

cases, while a constant rate is delivered to a farm, the water is rotated between fields or spilled when not needed for irrigation.

- Varied-amount rotation schedules are one way of adjusting the volume of water delivered over different parts of the growing season. In general, the frequency remains fixed, while the duration and/or rate is varied to apply more or less water to a particular area.
- Varied-frequency rotation schedules are another way of adjusting for variations in crop water use over the irrigation season. Under these schedules, the frequency of delivery is varied.

The rotational irrigation scheduling is the most common mode of water irrigation distribution. It is usually linked with an old irrigation infrastructure with a low frequency in water deliveries and a low efficiency in irrigation crops, where irrigation shift is supervised by the irrigation community not by the user.

The pricing is usually based on the irrigated area of each user, which does not encourage the efficient use of water.

The **on-demand delivery schedule** is the most flexible distribution system. The farmer or the watermaster decides at what time and what volume of water should be applied according to the crop needs, once analyzed several parameters measured in each crop or from the knowledge of the farmer. That is, in on-demand oriented systems, we set the irrigation scheduling from determining the timing and amount of water irrigation most suitable according to a series of technical procedures. The quantification of the water balance is based from the retention of water in the soil, that is, we can estimate the crop water consumption from the monitoring of water status of the soil or plant.

Regarding the monitoring of crop water demand, it should be noted that there is a new generation of sensors to detect the water necessities of the plant with much more accuracy than until now. There are new sensors which use capacitance measurements that allow the continuous recording of soil moisture at different depths. In that sense, these sensors allow us to detect which are the best periods to irrigate the plant. It has also advanced in the early detection of water stress of the plant by remote sensing (Fereres et al. 2000).

These different modes of irrigation scheduling take a great weight in the consumption of water for irrigation and define the developing degree of an irrigation area. In general,

the goal is to supply the entire field uniformly with water, so that each plant has the amount of water it needs, neither too much nor too little. There are various types of irrigation techniques which differ in how the water obtained from the source is distributed within the field and these are related with the modes of water distribution.

- In surface irrigation systems, water moves across the surface in order to wet and infiltrate into the soil. Surface irrigation can be subdivided into furrow irrigation (small parallel channels along the field length and the water is applied through each furrow) and basin irrigation (small areas having level surfaces that are surrounded by earth banks, see Figure 2.3). The surface irrigation is often called flood irrigation, as the irrigation results is the flooding of the cultivated land. Historically, this has been the most common irrigation technique in water distribution and this type of irrigation technique is usually used in rotational irrigation scheduling. Among the different modes of rotational irrigation, the surface irrigation is the least efficient especially when it comes to shallow soils with low storage water capacity, which usually requires the application of large volumes of water at crop.
- Localized irrigation is a system where water is distributed under low pressure through a piped network, in a pre-determined pattern, and applied as a small discharge to each plant. Drip irrigation (Figure 2.5), spray or micro-sprinkler irrigation and bubbler irrigation belong to this category of irrigation systems. This type of irrigation techniques is usually used on-demand irrigation scheduling.



Figure 2.3: Basin irrigation of wheat.



Figure 2.4: Center pivot with drop sprinklers.



Figure 2.5: Drip irrigation.

The most important advantage of sprinkler and drip system is the ability to modify and adjust very accurately the rate and frequency of irrigation, that is, these systems can be adjusted easily to the characteristics of moisture retention soil and crop specific needs.

The efficiency of water distribution is not only dependent on the modes of distribution, one of the most relevant causes in loss of efficiency is due to an old infrastructure. The waterproofing coating may be damaged, therefore it would be not in the same

conditions that the first day, so water leaks can be significant in many cases or even become excessive.

The on-demand irrigation system needs an infrastructure more expensive and bigger than a rotational irrigation system, because an on-demand system has to deliver water as long as the farmer requires water for irrigating the crop. In case of crops with similar water needs or in particular periods of water scarcity, the water is delivered to the crops at the same time increasing the peak flow in the canal. But on the other hand, the volume of water delivered through on-demand irrigation systems is lower than in rotational irrigation systems, because the application efficiency* is the maximum in on-demand irrigation systems and the farm irrigation efficiency is improved..

* The application efficiency is the ratio of the water that is stored in the root zone for later use by the plants to the total water applied.

2.1.1.2 Increase of the water price

The water price is an important tool to improve the efficiency in use of irrigation water, as we can use this tool to provide economic advantages to the users who make a good water use. It is necessary to adjust the water rate to the real water price to raise awareness to the users of the water value. In case of irrigation water, it is extremely important to analyze some important peculiarities such as the geographic and climatic conditions of the region and analyze the socio-economic effects of increasing or decreasing water prices. We can use this tool in different ways:

- Application of a general increase in water prices as a way of saving water.
- Establishment of economic incentives to reduce the use of irrigation water, especially in situations of scarcity.

Increases in water prices must be introduced jointly with a particular water model distribution, that is, the irrigation distribution system should be an on-demand irrigation system, as the farmer could be more accurate in water consumption increasing the application efficiency and could save water and money.

The majority of pricing services set by the irrigation community is based, as we have already shown in the previous sections, in fixing a price per hectare irrigated regardless of the volume of water supplied. In Spain, the main obstacle to implement

new fee structures is the difficulty to install flow meters in each parcel. In this sense, developing methods to measure the flow rate delivered would be useful.

However, before the possibility of increasing water prices for agricultural use, we must bear in mind that for most farms, the irrigation water is a large percentage of the average costs of cultivation as the seeds, fertilizers, salary of farmers..., in case that the farmers pay the real cost. Many times, these costs are subsidized by the government. This problem linked with the structural characteristics of the farms make that the social and economic impact of such measures can be extremely important for the farm agriculture.

2.1.1.3 Productive structure

To review the future prospects of irrigation is fundamental to know the structure of productive irrigated farms and more specifically the economic dimensions that they have. One way to characterize them is through the gross margin of farms, defined as the balance between the monetary value of gross output and the value of direct costs inherent in production. We show in the next figure the graph of the productive structure of farms in Catalonia-Spain (Figure 2.6) obtained by the National Institute of Statistics of Spain (INE). According to the criteria of INE, among these direct costs are included seeds, fertilizers, pesticides, water, etc., but does not include labor neither maintenance of machinery and facilities.

Note that 82% of farms have a gross income less than 48,000 euros per year and these farms occupy 46.5% of the irrigated area. It is also remarkable that over 50% of farms, which occupy about 15% of the irrigated area, can only reach 10,000 euros of gross margin.

This production structure is a serious problem for adopting new technologies which would improve the efficiency of water use by the agricultural sector. The small economic size of the most irrigation farms prevents the use of new technology because the capital investment that the modernization process requires is high.

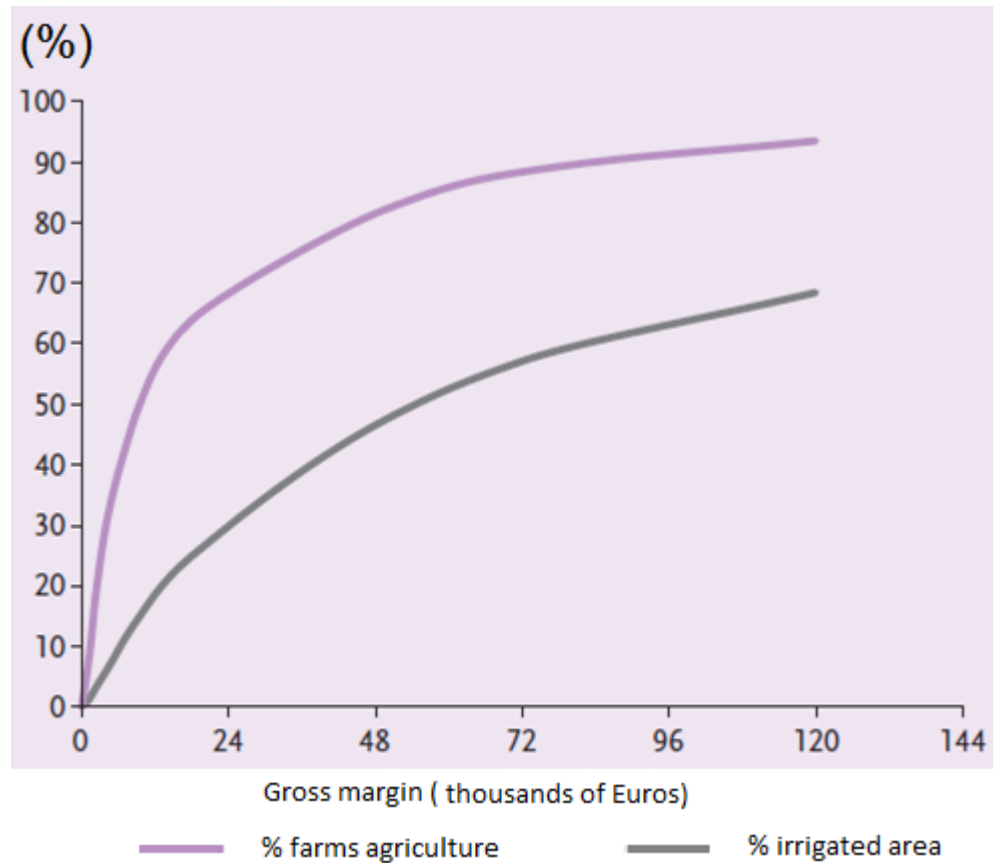


Figure 2.6 Survey about productive structure of farms during 2007 in Catalonia (INE).

2.1.2 Associated structural problems in canals

The structural problems in canals are frequently associated a bad management of these which are also related with a bad efficiency in water uses. Structural problems in a canal are sometimes consequence of unscheduled water deliveries or unknown extracted flows by the farmers. Small oscillations in water level at a cross section are not relevant, but if the water level variations are quick and higher than a meter, these oscillations could cause serious structural problems in a canal depending on the water level fall at the cross section, the height and angle of the side walls and the hydrologic features of the adjacent land.

For instance, in a case that the soil in contact to the cross-section is a non-porous soil, the soil cannot release quickly the pressure in his pores. In case of big changes in water level at a cross section, the gradient of pressures behind the side walls would

be so high than the water pressure would be able to crack the concrete (Figure 2.7) or even the structure could fail. Anyway, the fissures increase water losses and they also reduce the system efficiency.

Frequently, there are unscheduled water deliveries in a canal and it is not usual to take data records about increments and decrements in water level. So, it is quite difficult to know the water level variation in each cross-section along the canal. In this case, it would be useful to have a control element able to quantify the flow rate changes in a canal, from water level measurements at a particular cross-sections, to prevent important water level oscillations in the canal.



Figure 2.7: Breakage of the cross section of a canal by changes in gradient of pressures (Chimbote canal (Peru)).

2.2 Solutions to the irrigation problem

There are three areas in which we can act to increase the irrigation efficiency:

- Changing the water distribution from rotational irrigation scheduling to on-demand irrigation scheduling.
- Adapt the water prices to the real cost and establish economic incentives.
- Improve the productive structure.

The first and the third point are targeted introducing new technology as canal controllers and investing in irrigation infrastructure. If we look at the first point, the use of local manual control to manage the demand deliveries in a canal, rather than canal

controllers, causes more disadvantages than advantages. It is quite difficult globally to control a canal with local manual control. Simultaneous gate operations cannot be accomplished by local manual control unless irrigation system operators are stationed at every check structure so they can communicate with each other when flow changes occurs. On the other hand, we cannot expect that the operating staff control the interaction between all the parameters of a complex system.

Other shortcoming of manual control is the availability of operating staff, as they cannot be continually in attendance at every structure, their degree of dedication and motivation is not unlimited as their ability to resist possible pressures from the farmers. As a consequence, control structures are frequently set incorrectly, and operation is irregular and/or inappropriate, similarly measurements and readings may be imprecise. These shortcomings as the resulting lack of reliability, flexibility and equity in the deliveries are responsible to reduce the water distribution efficiency and appear to be the main cause of the very low performance of most irrigation projects.

In spite of these difficulties, we can achieve a satisfactory result between water requirement and demand deliveries, as it has been demonstrated using appropriate engineering techniques, and more recently of electronics and computer science (ICT) to reinforce traditional practice (Figure 2.8). The use of electronic means is becoming more and more common in all areas because it is cheaper. For this reason, nowadays it is easier to adapt them to the irrigation crops.

The key elements of any effective control system will include:

- New techniques to estimate the crop water consumption.
- The method of water distribution, such as on-demand irrigation scheduling.
- Equipment such as sluice gates, gages to measure water levels and canal controllers allowing us to control and modify the flow rate or the flow through the delivery turnouts.

The degree of satisfaction of farmer will depend on the productive structure which can be optimized by irrigation techniques and the water distribution system.



Figure 2.8: The ICT office of the master station of Navarra canal (Spain).

2.3 Diagnosis and Prognosis of the irrigation problem

The stage of diagnosis and prognosis consists of the analysis of the irrigation system to identify shortcomings (diagnosis) and to propose corrective actions (prognosis).

Many algorithms, in particular open loop algorithms, have difficulties and failures in the implementation in real canals (Rogers and Goussard, 1998). These difficulties are due to deviations between the predictive/control model and the reality, and these deviations are even more important when the factors introduced in the next paragraph are more relevant:

- Conceptual errors in the internal model: the model is based on the Saint-Venant equations, and these equations are only applicable in particular cases, when the pressure distribution along a vertical axis is the same as in hydrostatic conditions, the vertical accelerations are negligible, the curvature of the free-surface is small,...
- Errors in values of physical parameters of canal: Manning roughness coefficient, discharge coefficients in weirs and gates...
- Disturbances introduced into the canal: infiltration losses by cracks in the coating, unknown withdrawal flow rate...

The conceptual errors associated to the model (the first factor) are not important in canals, because the distribution of pressure in a cross section may be assumed

hydrostatic, the vertical accelerations are negligible and the curvature of the free-surface is usually small.

Instead, the errors in values of physical parameters (the second factor) are common in canals, because it is quite difficult to obtain an accurate value of the Manning coefficient or the discharge coefficient in weirs or gate. The model is very sensitive to these physical parameter; for that reason, we have to calibrate them accurately.

The third factor (disturbances introduced into the canal) leads important deviations between the model and the reality. The disturbances caused by infiltration can be controlled by a good coating and these disturbances are not usually important and they have a constant value. Due to the difficulty to distinguish between disturbances as small perturbation caused by conceptual errors of the model or water infiltration losses, all of them are considered the same type of disturbance which are not important. But the disturbances introduced by changes in demand deliveries are more dangerous because the canal could overflow or could dry depending on the disturbance. These disturbances are usually unscheduled demand deliveries which are caused by flow rate extractions during the irrigation cycle.

As we introduced before, if the farmers can adjust the supplies obtained by the irrigation community before the irrigation cycle begin, these changes on demand deliveries are known disturbances. These disturbances are known in advance, and these can be mitigated by an open loop controller (feedforward controller) as GoRoSo (Soler, 2003).

The main problem in a canal are the disturbances caused by climatic variations (rainfalls and associated runoff) or unscheduled demands by farmers, that extract more or less flow that has been demanded by them, because these disturbances are more difficult to mitigate by a controller. One way to protect the canal from these disturbances could be with ponds built by the irrigation community or farmers themselves. A reservoir is able to store water according to the crop requirements and regulates the volume of water provided by the canal, but it is not always possible build large reservoirs to regulate each pool of the canal. Another option to control the disturbances in canals, which operate in steady state, would be to use the wedge storage of every pool, but this only works with very low disturbances. In all other cases, you need a closed loop controller (feedback controller) to modify the sluice gate

trajectory or a control variable of the system to return to the desired canal state on real time, as the control algorithm proposed in this thesis.

2.4 Objectives

In this chapter, we have identified the majority of associated problems in irrigation crops and we have proposed some actions to avoid these problems. In that sense, we are going to focus in a certain action (introduction of control algorithms) because we have contributed to the development of control algorithms in this thesis.

Once we have introduced the irrigation problems in transport and water distribution in a canal, we can introduce the necessity of adopting control algorithms in canals to increase the efficiency in water supply and solve some of these problems. This is one of the objectives of this thesis.

The main objective in this thesis will be develop a feedback control algorithm to control the unknown demand deliveries in real time.

Chapter 3

Literature review

3.1. Introduction

P.O. Malaterre presented a working collection of all the documentation on canal control algorithms existing at the time when published in the Journal of Irrigation and Drainage of the American Society of Civil Engineers (ASCE), see Malaterre et al. (1998). The great work developed by Malaterre, defining a complete classification of the algorithms involved in canal control helps me to structure this chapter. The documentation of the literature review focuses on defining several classifications for control algorithms from different points of view and presents most of the control algorithms that have been developed so far.

We define a control system, as an arrangement of algorithms, electronic, electrical, and mechanical components that commands or directs the regulation of a canal system. The control system is responsible for controlling structures in a canal based on information extracted from canal measurements, in addition to certain variable operating conditions which are defined as targets, usually constant water levels. The term canal control describes those steps necessary to ensure the required pool water level and flow along the canal. The flow conditions are controlled by adjusting the gates positions.

Canal conditions depend upon the adjustment of action variables (gate openings, discharges) that provide some control of the canal system. Some variables are easily adjusted while others are more complicated or cannot be adjusted. A variable easily adjusted is the gate position. Other variables such as the Manning roughness, and the geometry of the canal cannot be readily controlled. Canal control is achieved by manipulation of the variables to obtain the desired canal system conditions.

The control algorithm, which contains the set of rules or set of instructions, is one of the primary element in a controller, and it calculates the desired value (position or

speed) that the gate must fulfill. An algorithm is a prescribed set of well defined rules or processes for the solution of a problem in a finite number of steps. In canal control, algorithms are the procedures used to implement the various canal methods of operation. The algorithm is designed to process the input information from the sensors, perform the comparator function, and calculate the proper output to the actuator. The input are the quantities that are observed, measured, or predicted; the output is a control action.

There are many criteria to characterize the canal control algorithms, such as control variables (flow, depth, gate opening), interactions between the local controllers (local, centralized...), communications requirements, technical design, alarms, and localization of measurements along the canal.

Different control algorithms for the regulation of irrigation canals have been developed and applied throughout the world. Each of them can be characterized according to several criteria, among which are: the considered variables (controlled, measured and control action variables), the logic of control (type and direction), the control system design and field implementation.

3.2. Considered variables

In control theory, a system is usually represented as in Figure 3.1. "U" are the control action variables (e.g.: gate openings, increments of gate opening, discharges) to modify the state of the system and "Y" represents the system controlled variables (e.g.: water level, discharge, volume) or measured variables (e.g. water level, discharge). There are three kind of variables in a control algorithm.

- Controlled variables
- Measured variables
- Control action variables

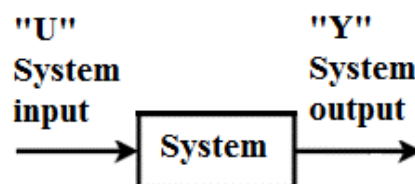


Figure 3.1: Representation of a system.

3.2.1. Controlled variables

Controlled variables are system variables to which targets are assigned. The controlled variables are those describing the desired behavior of the canal. Controlled variables on an irrigation canal can be of three types: flows, water levels or volumes, according to the terminology used in the physical system modelling. The controlled variables are localized at certain cross-sections in the canal.

3.2.1.1. Controlled variable: Discharge

The needs of irrigation canal users are defined mainly in terms of discharge. For example, agricultural needs are expressed in terms of given discharges delivered to a parcel, to a secondary canal, or to a pumping station, environmental needs as tailend discharge, or minimal discharge. Users' needs can then be defined in a more flexible way, in terms of volume distributed over a time period. In this case, the controlled variable is no longer a given value of discharge, but a volume.

Discharge fluctuations are then authorized, but these disturbances can be managed by controllers or storage reservoir which is not the frequent option due to reservoirs are expensive and the space available next to the canal could be limited as we introduced before.

A hydraulic system has to be managed, directly or indirectly, in order to satisfy users' demands in discharge. Considering the nature of the physical phenomenon (gravity surface flow from upstream to downstream), these demands in discharge can initially be satisfied from the volume of water in the canal, but these demands are usually satisfied from a source situated at the upstream end of the system, frequently a reservoir. Rodellar (Rodellar et al., 1993), CACG (Piquereau A. et al., 1982), CARAMBA (De Leon, 1986), Davis University (Balogun O.S., 1985), SIMBAK (Cheverreau, 1991), are examples of regulation methods controlling discharges.

3.2.1.2. Controlled variable: Water level

The water levels can be easily measured in canals, so it is usually the controlled variable. Controlled water levels "y" can be upstream (Figure 3.2 A), downstream (Figure 3.2 B), or intermediate inside the pool (Figure 3.2 C) and operational characteristics are very different depending on the location of "y". The methods of operation should not be confused with canal operation and control concepts. These

methods of operation are based upon the location of a pivot point within the canal pool, where the water level remains constant while the water surface slope varies. We have introduced the methods of operation as follow (Buyalski et al. 1991):

- Constant upstream depth- The pivot point is located at the upstream end of the canal pool (Figure 3.2 A).
- Constant downstream depth- The pivot point is located at the downstream end of the canal pool (Figure 3.2 B).
- Constant volume- The pivot point is located near the midpoint of the canal pool (Figure 3.2 C).

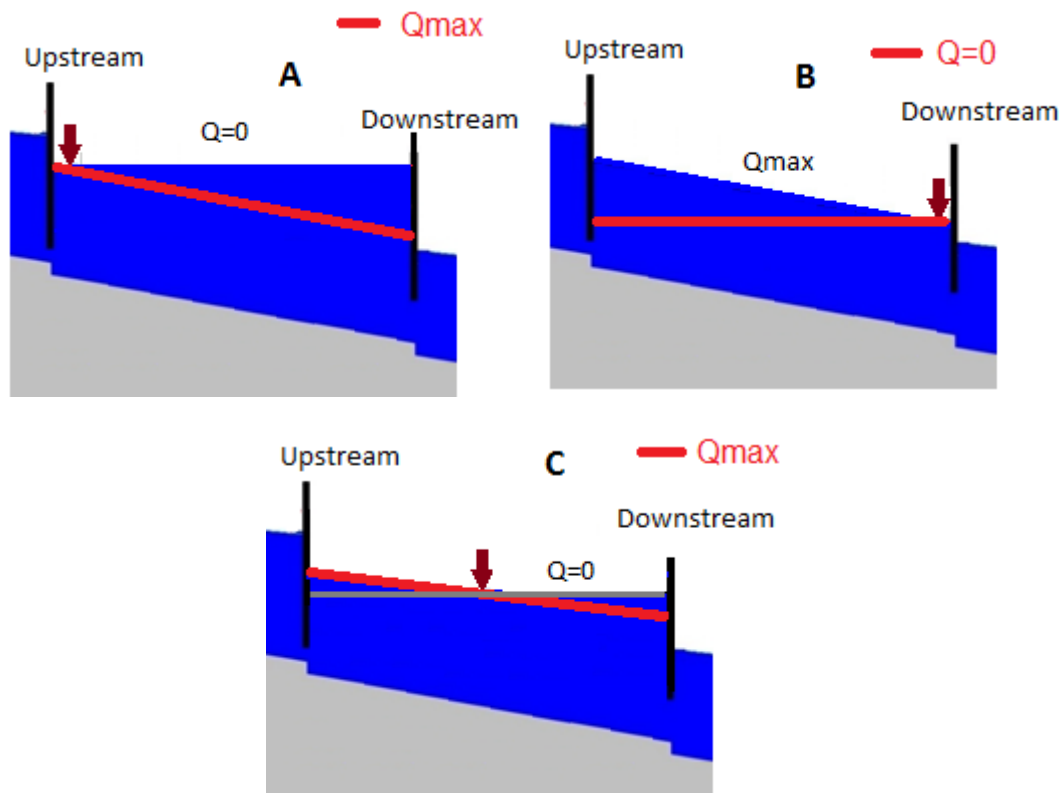


Figure 3.2: Downstream control concept.

Before defining the methods of operation, we have to introduce several concept: canal operation and canal control concept and, in this way, make more understandable the methods of operation.

The canal operation concept, established by the canal system operating criteria, determines the flow schedule. We can define the canal operation concept as:

- Downstream operation concept applies to canal systems that are primarily demand-oriented and usually is associated with delivery systems. These systems convey water from a single source such as a storage reservoir to a number of individual points of use, and they are associated with irrigation. The delivery systems convey initially the maximum flow and it decreases downstream.
- Upstream operation concept applies to canal systems that are primarily supply-oriented and usually the concept is associated with collector systems. These systems convey water from several individual sources such as surface inlet drains for rain storm located upstream to a single point of diversion located downstream. The collector system is supply-oriented because it must convey a variable water supply from any number of sources to a downstream single point of diversion. The collector systems convey initially the minimum flow and it increases downstream.

The control concept determines how the canal control structures are adjusted to satisfy the canal operation concept. The control concepts used in a canal system are defined by the location of information needed to operate the control structure:

- Downstream control. Control structure adjustments are based upon information from downstream pool. The required information could be measured by a sensor located downstream or by the irrigation system operators, and the watermaster establishes the water schedule from the required information. Downstream control transfers the downstream canalside turnout demands to the upstream water supply source and is compatible with the downstream operation concept.

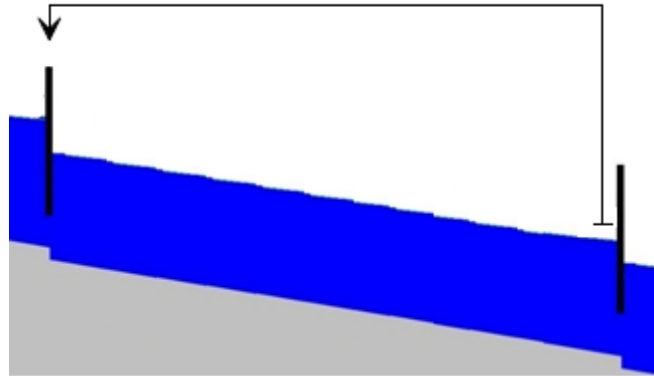


Figure 3.3: Downstream control concept.

- Upstream control. Control structure adjustments are based upon information from upstream pool. The required information could be measured by a sensor located upstream of the position of the gate/regulator, and the watermaster establishes the upstream water schedule from the required information. Upstream control transfers the upstream water supply information (or inflow) to downstream regulators and is compatible with the upstream operation concept.

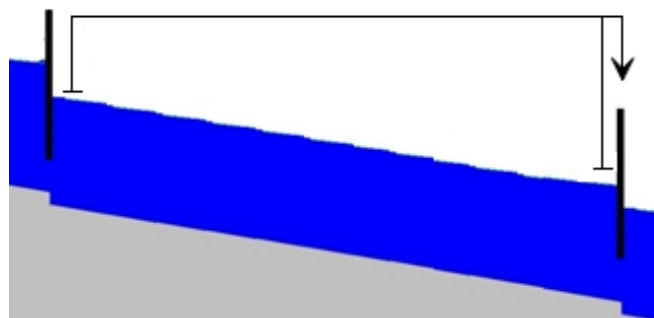


Figure 3.4: Upstream control concept.

Once we have defined the control concept and the operation concept, we can define the methods of operation:

Constant downstream depth

The constant downstream depth method of operation wherein the water depth at the downstream end of each canal pool remains relatively constant is used in most canal systems. The reason why this method is so prevalent is that a canal can be sized to convey the maximum steady flow (Figure 3.5); steady-state water depths should never exceed the normal depth for the design flow rate. The cross-section size and freeboard can be minimized, thus reducing construction costs.

When a constant depth is maintained at the downstream end of canal pool. The water surface profile will essentially pivot about this point as the canal flow as shown on Figure 3.5. A storage wedge between different steady-state flow profiles is created. When flow increases, the water surface gradient and storage volume must also increase. Conversely, storage volume must decrease for a reduction in steady-state flow.

Because of these storage considerations, a natural tendency exists for a flow change that originates at the upstream end of a pool to create the change in storage that is needed to keep the downstream pool depth constant (Figure 3.6).

Examples of regulation methods controlling downstream water level are GoRoSo (Soler, 2003), PI controller as Litrico X. et al. (2006) and Aguilar et al. (2012), LQ control (Weyer, 2003), Model Predictive control (Van Overloop, P. J., 2006), KAPTROLLER (Akouz K. et al., 1995), IMTA-Cemagref tested in Yaqui canal (Mexico) (Chavez A.A et al., 1994), PRECOM (Malaterre et al., 1996), Buyalski and Serfozo (1979), Deltour (1992), Chevereau (1991). There are other regulation methods which combine several controlled variables (discharge and downstream water level), as CLIS (Liu F. et al., 1998) or a distributed model predictive controller as Alvarez et al. (2013) or Rodellar et al. (1993).

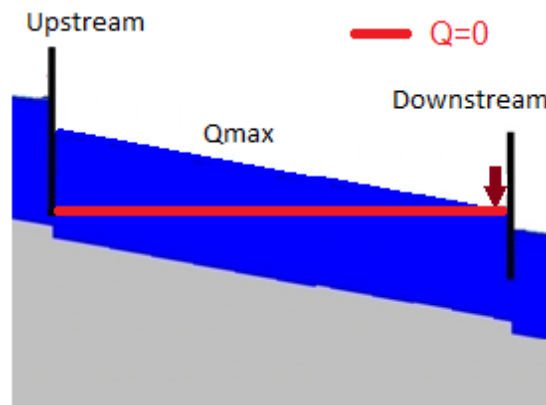


Figure 3.5: Control of the downstream level of a pool.

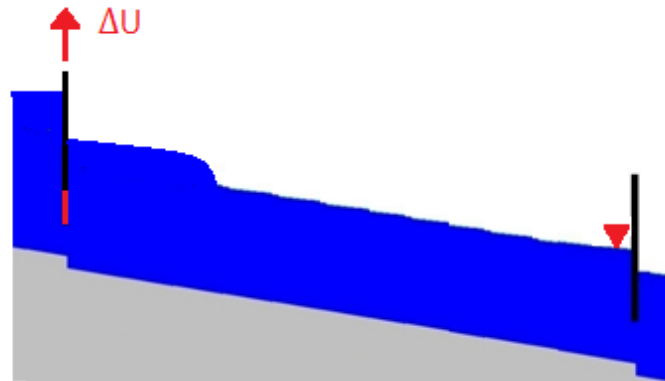


Figure 3.6: Upstream flow change with the constant downstream depth method of operation.

Constant upstream depth

With this method of operation, a constant upstream depth is maintained constant by pivoting the water surface at the upstream end of the canal pool as shown on Figure 3.7. The constant upstream depth method is sometimes called "level bank" operation, because canal banks must be horizontal to accommodate the zero-flow profile (Figure 3.7). The construction of a level bank canal is the main problem to this method. A level canal bank increases the cost of construction considerably, especially for concrete-lined canals because this cross section is not constant along the canal. Most existing canals could not use level bank operation unless additional canal bank and lining were

added to the downstream portion of each pool. Exceptions to this would be canals with little elevation drop between checks, or those operating at flows well below maximum flow capacity.

Flow changes originating at the downstream end of the pool cause canal water depths to change in the direction needed to achieve new steady-state profiles. AVIS and AVIO gates (Goussard, 1993), PIR (Deltour, 1992) are examples of regulation methods controlling upstream water level.

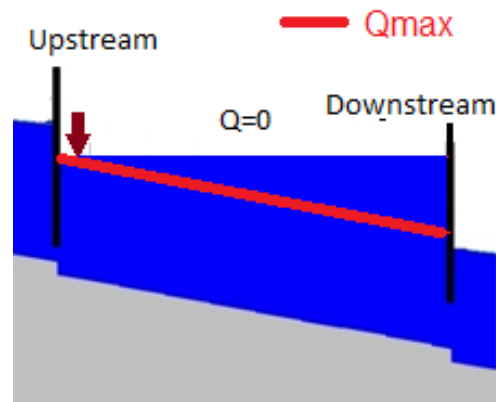


Figure 3.7: Control of the upstream level of a pool.

Constant intermediate water level

This method of operation is based upon maintaining a relatively constant water volume in each canal pool at all times. The water surface will pivot about a point near midpool as the flow changes from one steady-state to another.

Storage wedges will exist on either side of the midpool pivot point as shown on Figure 3.8. For any given flow changes, volume changes in each of these wedges is equal and opposite. When flow decreases, volume of water in the upstream wedge decreases and volume increases in the downstream wedge. When flow increases, the opposite occurs.

With the constant upstream depth and constant downstream depth methods, excessive time is required to either build up or deplete the storage in the entire canal system when changing the steady-state rate of flow. The constant intermediate water

level method of operation avoids lengthy delays, because total volume of water in the canal system does not change significantly. BIVAL (Zimbelman D.D., 1987) is an examples of regulation methods controlling constant volume.

On the other side, canal dimensions must be adapted to the desired demands, and as a consequence oversized canals can be finally necessary, in comparison with other procedures. It is a method of operation used infrequently.

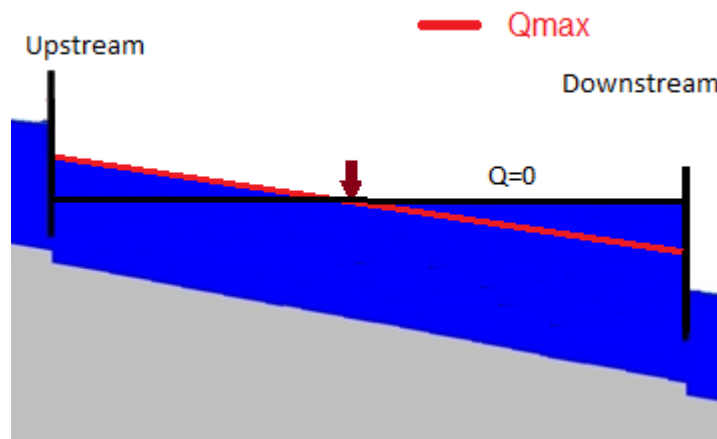


Figure 3.8: Control of an intermediate water level.

3.2.1.3. Controlled variable: Volume

The methods which use the volume as controlled variable offer the advantage of drastically reducing transients by making the flow rate variations in each reach independent of the stored volumes. These methods are applicable to irrigation canals with important storage volumes, and equipped with turnouts whose feeding is not dependent on water levels in the main canal (e.g.: pumping stations). Examples of these methods are the Dynamic Regulation (Coeuret ,1977), Controlled Volumes (Buyalski, 1991), BIVAL (Zimbelman D.D., 1987), or a multivariable approach for the command of Canal de Provence Aix Nord water supply subsystem (MIMO) (Viala Y. et al., 2004).

3.2.2. Measured variables

Measured variables on irrigation canals are generally water levels (e.g.: ELFLO (Shand, 1971), PIR (Deltour, 1992), PREDAPTRON (Rodellar et al., 1993), PRECOM (Malaterre et al., 1996), KAPTROLLER (Akouz K. et al., 1995), Corrigan (1982), GoRoSo (Soler, 2003). In some cases, measured variables can be discharges (e.g.: CACG (Piquereau et al., 1982), CARAMBA (De Leon, 1986)). A discharge can be measured with specific equipment (based in general on the measure of one or several flow velocities, with a propeller, an ultrasonic or electromagnetic device), through a cross structure equation or a local control section rating curve with a sufficient precision. A depth can be measured installing pressure transducers, bubbler pipes or gages directly in the cross-section.

Automatically controlled canal operation depends on water level data collection. The accurate water level information is critical as the actions introduced to the canal are based on these water level measurements.

3.2.3. Control action variables

Control a variable requires mechanical equipment such as gates, valves, and pumps.

Control action variables “U” are generally either gate openings, increments of gate opening and “Q” are generally discharges, or increments of discharge. Gate openings have the advantage to take into account the complex dynamics linking this opening with the local discharge and upstream and downstream water levels. Examples of controllers which used these control variables are:

These variables are the gate positions (G) and the discharge (Q):

- The discharge: the control action is the flow rate (Q), it is required a convert function to translate it into gate position (G) so that it can be applied to the system. Examples of this cases would be CACG (Piquereau et al., 1982) and KAPTROLLER (Akouz K. et al., 1995).
- The gate position: it has the advantage to take into account the complex gate dynamics. Controllers that use the gate position as control action variable are CLIS (Liu et al., 1998), Sepúlveda (2007).

3.3. Logic of control

To define the logic of control concept depending on the type of information used to calculate the control action variables of the system, we will talk about "closed loop", also called "feedback control", or "open-loop", also called "feedforward control".

3.3.1. Open-loop control

Open-loop control is when the controlled variable is adjusted with no comparisons to actual response or to actual desired conditions. An example of open-loop control is a traffic light signal. The traffic lights respond at specific time intervals regardless of the actual traffic conditions.

In open-loop, the control action "U" is calculated knowing the dynamics of the system (using a model), the setpoint output "Yc" (for instant water demands by farmers, known in advance) and possibly an estimation "P" of perturbations (Figure 3.9). The open-loop can compensate inherent system time delays by anticipating users' needs. These needs have to be known as precisely as possible. They should take into account climatic, agronomic, and sociological data, as well as recordings of the water consumption of previous weeks or seasons (Perrin 1989). An open-loop is generally insufficient, due to model errors, perturbation estimation errors, and unknown perturbations. Open-loops can be applied to all the controlled variables: discharge, water level and volume. Examples of such methods are Bautista et al. (2003), Baume (1993), Wahlin & Bautista (2003), Soler (2003), Sawadogo (1991), Sabet et al. (1985), Tomicic (1989), Huising (2004).

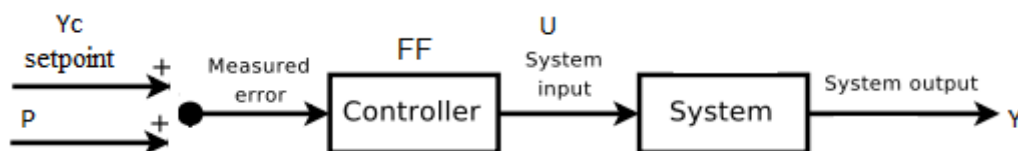


Figure 3.9: Open-loop control (feedforward).

3.3.2. Closed-loop control

Closed-loop control is when the controlled variable is measured and compared with a setpoint representing the desired performance. Any deviation from the setpoint is send to the control system so that it sets the control actions to reduce the deviation of the controlled variable from the reference.

In closed-loop, the control action variable “U” is calculated from the error measured between the real controlled variable “Y” and its corresponding setpoint “Yc” (Figure 3.10). Perturbations “P”, even if unknown, are taken into account indirectly, through their effects on the output “Y” of the system. In control theory, this concept is essential since it links a control action “U” to a controlled variable “Y”. Closed-loops can be applied to all the controlled variables: discharge, water level and volume.

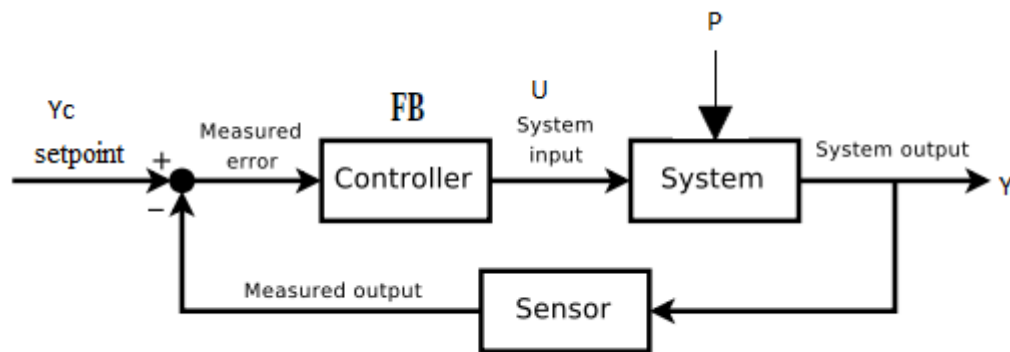


Figure 3.10: Closed-loop control (feedback).

Examples of closed-loops in discharge are GPC (Sawadogo, 1992), Rodellar et al. (1993), CACG (Piquereau et al., 1982), CARA (Marzouki, 1989), CLIS (Liu et al., 1998) and Isapoor et al. (2011).

We can introduce two points of view about the closed-loops in water level depending on the relative locations of the control action and controlled variable, as it was defined in canal operation and canal control concept.

Water level downstream control generates indirectly a discharge closed-loop control, since it is obtained from the modification of the upstream discharge. This characteristic

is an essential property of the water level downstream control. Examples of such methods are AMIL, AVIS and AVIO gates (Goussard, 1987), LittleMan downstream, ELFLO (Shand, 1971), CARDD (Burt, 1983), Zimbelman (1987), PID (Chevereau, 1991), PIR (Deltour, 1992), MPC (van Overloop, P. J., 2006), Clemmens & Schuurmans, J. (2004), Gómez et al. (2002).

Some water level control methods combine upstream and downstream control logics. They are called feedback mixed controls. Examples of such methods are Balogun (1985 and 1988), Filipovic et al. (1989).

3.4. Methods in control system design

The design technique is the algorithm or methodology used within the control algorithm in order to generate the control action variables from the measured variables (i.e. the determination of control action "U" leading to a desired dynamic behavior of the controlled system).

3.4.1. Controllers design: Definition of inputs / outputs

In canals, discharge is normally the parameter that is to be controlled, although discharge is not a quantity which can readily be determined. Instead, some other quantity must be measured; then, discharge is related to these quantities through algorithms. In canals, the parameter most frequently used is water surface elevation or the difference in water surface elevation between two points. Unless otherwise noted, water surface elevation is the controlled parameter (input). The location and number of sensors used as output depend on the objectives of the conveyance system. Only one water level sensor per reach is used if the canal is a delivery system consisting of gravity turnouts. Usually, the location of the sensor is near the downstream end of a canal reach near the turnouts. Multiple sensors in a canal pool are used to implement various control methods.

Control action variables "U" and controlled variables "Y" of a given controlled system have been defined in section 3.1. Variables "U" are particular inputs of the system and "Y" particulars outputs of the system. Since our classification concerns controllers and not systems, we choose to define "inputs" and "outputs" in reference to the controller and not to the system. The different types of inputs and outputs are illustrated in Figure 3.11, where:

- U' are variables acting on the system, and not generated by the controller (e.g.: non controllable inputs, perturbations).
- U variables generated by the controller, and acting on the system (e.g.: control action variables).
- Y' variables generated by the system, and not directly used by the controller.
- Y variables used by the controller, and generated by the system (e.g.: measured variables used by the controller).

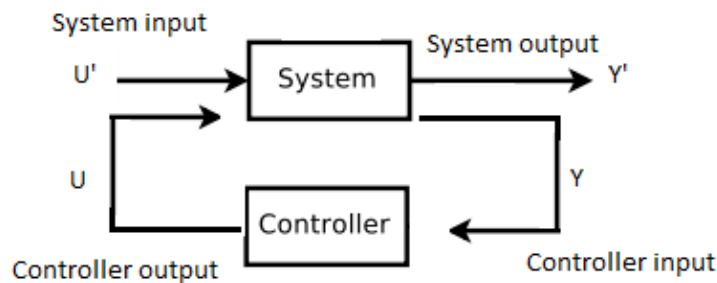


Figure 3.11: Inputs / Outputs of a controlled system.

A monovariate controller (in input and output) has only one input and one output (variables being considered as scalars and not as vectors). Monovariate methods are also called SISO (Single Input, Single Output). A multivariate controller (in input and in output) considers several inputs and several outputs. The multivariate methods are also called MIMO (Multiple Inputs, Multiple Outputs).

3.4.2. Monovariate methods

These methods have traditionally been used in irrigation canal control. Monovariate methods are applied a single pool of the canal, these methods require to split the overall system into several subsystems without explicitly taking into account interactions between them. An irrigation canal is a multivariate system presenting strong interactions between subsystems. For example, the operation of a gate influences several upstream and downstream pools.

Most of the irrigation canal control methods based on control theory use the well-known linear monovariate PID controller. Examples of PID related methods are:

- P: AMIL, AVIS, AVIO (Goussard, 1987).
- PI: ELFLO, BIVAL, Litrico et al. (2003).
- PID: PID UMA Engineering.

Although very efficient in most cases, PID controllers do not explicitly take into account the characteristic time delays in a canal. Examples of these methods are IMTA (Cemagref, 1994), BIVAL, ELFLO, PID UMA Engineering).

There are methods as, the generalized predictive control method (GPC), a monovariate optimization method, based on the minimization of a criterion "J", which calculates the control action variable and the error between the controlled variable and its targeted value. This method has been developed by Sawadogo (1992), Rodellar (1993) and Camacho and Bordons (2004).

GPC method uses transfer function models: Chan et al. (1990), Soeterboek (1990), Lee et al. (1990), Linkens et al. (1992) and it incorporates an open-loop and a closed-loop.

There are more algorithms which use fuzzy control methods as CNABRL on the T2 canal, a method based on fuzzy control expert systems used in Marrakech (Morocco), or methods based on neural networks are (e.g.: Schaalje et al. (1993) and Toudeft (1994)).

The little-man is another controller that uses monovariate method which was implemented in The Columbia River Basin Project (near Ephrata, Washington). The little-man control technique is applied to canal systems using the local automatic control method. The application of the Little-Man algorithm is most successful in implementing the constant upstream control method of operation.

Monovariate methods take into account just one variable of the system, without taking explicitly into account interactions between them. Instead, an irrigation canal is a multivariate system which presents strong interactions between them.

3.4.3. Multivariate methods

A multivariate controller is designed to control several pools, therefore the controller operate globally in the canal, instead the monovariate controllers which are designed to control a single pool.

Multivariable controllers for canal automation have been designed based on optimal control techniques: linear (Reddy, 1996) or non-linear optimization and model inversion (Liu et al., 1995).

Different model inversion methods are described in the literature, leading generally to open-loop controllers as Chevereau (1991), Liu et al. (1992), and more rarely to closed-loop controllers (Liu et al., 1994).

Optimization methods have been developed too. These methods are, in essence, multi-variable. Different methods exist: linear optimization (Sabet 1985), non-linear optimization as Tomicic (1989), Khaladi (1992), Lin (1992), Soler (2003), van Overloop, P. J. (2006), Álvarez A. et al. (2013), LQR (Corriga, 1983), Garcia (1992), Filipovic (1989), Reddy (1992), Malaterre (1994), Sawadogo et al. (1994), LQG (Lemos J.M. et al., 2013), Wahlin (2004), Montazar et al. (2005), PI control (Litrico et al., 2006), Begovich et al. (2005). The classical non-linear optimization only leads to an open-loop, sensitive to errors and perturbations. In order to introduce a closed-loop, the optimization has to be processed periodically. This complicates the method and limits its applications due to real-time constraints and computer time. Furthermore, the determination of real initial conditions, required for the optimization, is not easy. On the other hand, LQR methods, based on a state space representation, incorporate, in essence, an open-loop and a closed-loop.

The implementation of multivariable methods is far more complex than for monovariable methods.

3.5. Field implementation

In the history of canal control, several multivariable methods have been developed. However, very few of them have been finally implemented on canals.

Different aspects of field implementation of regulation methods can be distinguished. They are the configuration (e.g. localized, semi-local, centralized), the devices (e.g. automatic gates), the instrumentation (e.g. water level sensor, discharge measurement device), communications (e.g. radio transmission), calculation and data processing.

In local configuration (Figure 3.12) each device is controlled by an independent controller. The device (e.g.: PID UMA Engineering, LittleMan, GEC Alsthom Gates)

only uses local information (measured variables). A local automatic controller consists of all the equipment required to execute the control algorithm and the necessary inputs and outputs in a stand-alone unit usually located at each check structure. Examples of this method of implementation would be Gómez et al. (1998), Cardona et al. (1997), Begovich et al. (2007) and Van Overloop et al. (2005).

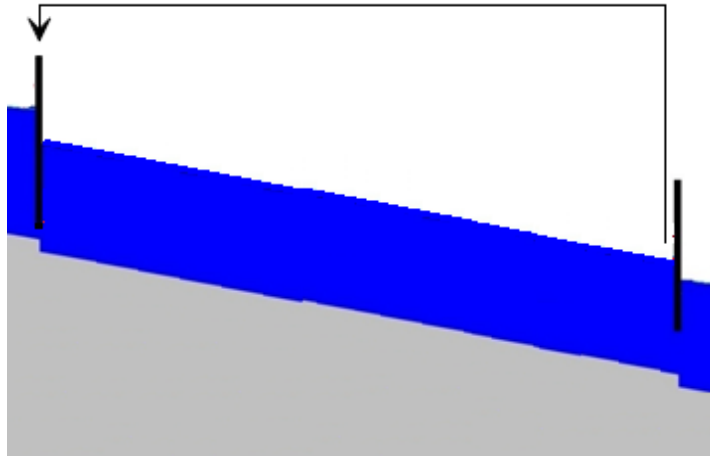


Figure 3.12: Local controller.

In semi-local configuration (Figure 3.13) controllers are no longer independent. They explicitly use data generated by adjacent controllers or setpoints. For example, a controller can use the control action variable of the next downstream controller (e.g.: PIR, ELFLO + decoupler, semi-decentralized Predictive Control (Sepúlveda, 2007)).



Figure 3.13: Semi-local controller.

In centralized configuration (Figure 3.14) all control actions are generated by a central controller (human being or computer) in a headquarter where information from the whole canal is available and used in the controller, distant from the different actuators. Control actions are gate openings (e.g.: Salt River Project) or discharges. Centralized control allows for supervision and remote control of the system. However, the implementation is more complex and more sensitive to hardware and communications breakdowns than localized configurations. Examples of this implementation would be GoRoSoBo (Soler, 2003) and Montazar et al. (2005).

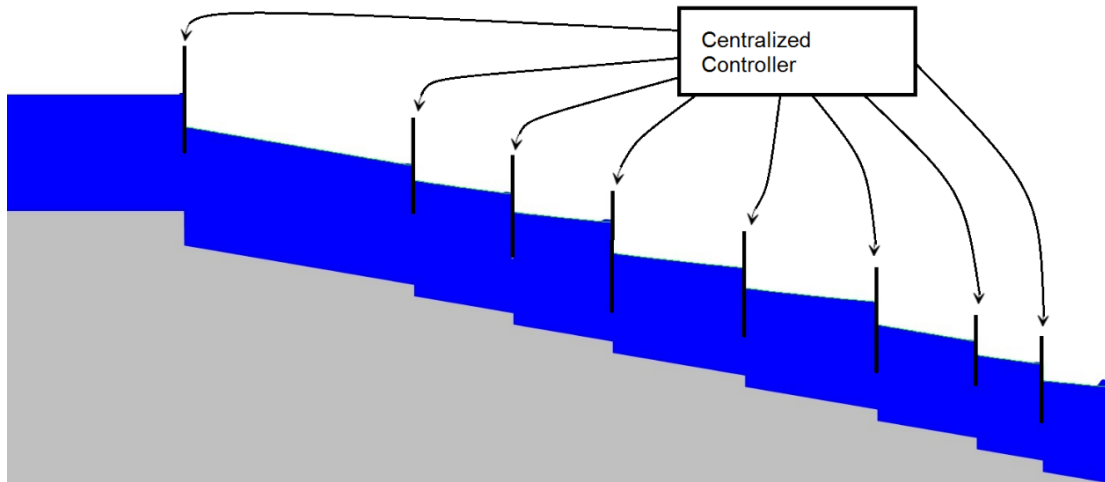


Figure 3.14: Centralized configuration.

Chapter 4

Characterization of a predictive control scheme

The term canal control describes those steps needed to ensure the required water level and flow along the canal. The control of the canal is achieved by manipulation of variables (sluiceway openings, pump flows) to obtain the desired conditions in the canal. These conditions are controlled by adjusting the volume of water pumped in/out the canal, adjusting the positions of gates, and by regulating flows through the delivery turnouts. When canal is in steady flow, the control of the canal is easy, but problems appear due to flow changes at setpoints. The demand deliveries are not constant in space neither in time. The canal state may vary unpredictably due to disturbances, which are considered to belong to the external environment of the system (runoff produced by rain events, canal losses due to infiltration, unknown turnouts operations). So if we want to keep under control the canal in these circumstances, it is necessary to use controllers, as for instance, proportional integrated derivative controllers (Montazar et al., 2005), predictive controllers (Gómez et al., 2002), optimal controllers (Malaterre, 1998) or Heuristic controllers (Durdu, 2004).

If the canal system is operated by automation, the efficiency, flexibility, and responsiveness of the diversion project will be improved, and the water resource loss will also be reduced during the diversion project operation process (Buyalski et al., 1991).

Many times, the global state of a canal is governed by predictive controllers, which modifies the gate position depending on the demand or on-demand delivery changes, and these controllers have to take into account a lot of variables which affect the canal flow behavior, for instance, parameters as the Manning or energy loss coefficient, or the past and current canal state. Before considering a dynamic system (the canal state) governed by an overall control diagram as we will show in chapter 5, we will characterize how our predictive controller should work.

4.1. Introduction

The main focus of a predictive control (PC) is to define control actions using input data from a computer model, which defines the dynamics of a canal system that can be correctly approximated by Saint-Venant's equations or other hydraulic approach and the hydraulic model of a check structure (as a sluice-gate, a weir,...) (Buyalski et al., 1991; Yevjevich, 1975). On the other hand, physical parameters of the computer model (as the Manning coefficient) must be calibrated before using the predictive control.

We should follow the next steps in the process of a predictive control algorithm:

- Estimate the canal flow response during a predictive horizon.
- Determine the sequence of control actions during a predictive horizon to control the canal.

Really, the canal control is strongly associated with taking more actions depending on the reference defined by the Watermaster; normally this response is to increase or decrease the water level at a setpoint or checkpoint, that is, at a certain cross section in the canal in which we can measure the water level with pressure transducers or staff gages directly in the canal. Typically, these setpoints are associated with a certain constant value (desired water level) during the irrigation cycle.

A control action to minimize the variation between the desired and simulated/measured behaviour, is usually made by optimization problems which seeks to minimize an objective function. Minimizing the objective function is equivalent to minimize the absolute deviation between simulated values and desired values. Before introducing the optimization problem of our predictive control, we are going to introduce a general definition of computer models and objective functions which are also involved on predictive controllers.

4.2. Computer model

An irrigation canal is a hydraulic system, whose objective is mainly to convey water from its source (reservoir) down to the users (farmers). This system is controlled by cross structures as sluiceways, weirs, orifice offtake which are operated in order to control the water levels or discharges in several cross sections along the canal.

The physical dynamics of this hydraulic system can be correctly approximated by Saint-Venant equations which are a nonlinear partial derivative hyperbolic equations, combined with non-linear algebraic cross structure equations depending on the cross structure (weir equation, sluicgate equation). These must be completed by initial conditions and boundary conditions at cross structures, and the whole canal.

In irrigation canals, we have an advantage over other fields that use hydraulic modeling, due to the regularity and shape of the cross-sections. The flow in the canal can be simulated by 1D flow models, although it is true in certain sections as cross structures, the flow behaviour is 2-D or 3-D. Using 1-D flow models is very advantageous because these models decrease the computational time in front of 2D or 3D models, the computational time is a crucial factor to take into account for a predictive model. An unsteady 1D flow model is used to analyze the system behavior in this thesis and the model results are visualized with our displayer (Figure 4.1).

In the references, there are several models describing the flow in canals. Some of these models are established from the full Saint-Venant equations and are called non-linear models, and other models are established from approximations of the Saint-Venant equations and are called linear-models.

Simplifications and approximations of the Saint Venant equations can be separated into four categories: empirical, linearized, hydrological, and hydraulic (Fread, 1985).

- Empirical methods are based on a large collection of observed data for a discrete reach of a river or canal and are only applicable for the reach.
- Linearized methods simplify the Saint Venant equations by neglecting nonlinear terms.
- Hydrological methods utilize the mass conservation equation and a relationship between storage and discharge.
- Hydraulic methods add the momentum conservation equation.

On the other hand, the linear models are commonly used in predictive control. These models simplify the Saint-Venant's equations by linearizing them around a steady state, called reference state (Brogan, 1985). The advantage of a linear model is the simplicity. However, the same characteristic is responsible for the disadvantages. These models do not consider the non-linear terms, which have an important influence in the system dynamic when the flow changes are important in time.

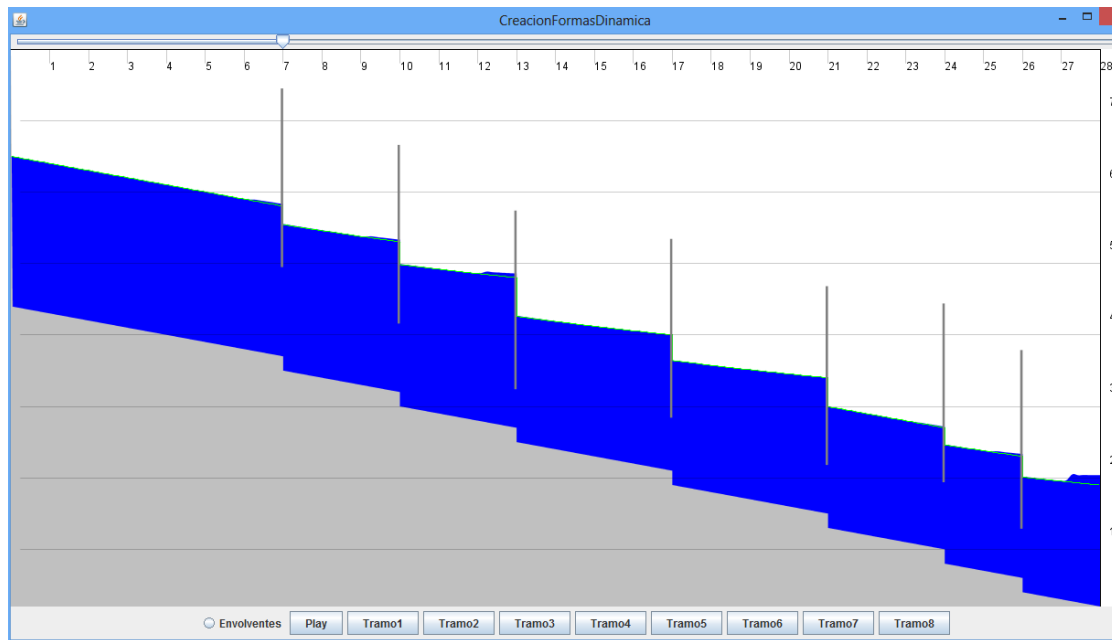


Figure 4.1: Display of the values obtained by a computer model of a canal (this displayer was programmed in Java code and it was developed for us to check the results obtained with the predictive control).

4.2.1. Linear model

There are several linear models in the bibliography, and we are going to introduce some of the more common:

- The Muskingum model is a frequently used hydrologic linear model for flood routing (Cunge (1969); Dooge et al., 1982; Viessman et al., 1995). It contains two equations, a continuity and a storage equation. The parameters of the model are K (the travel time of the canal reach) and χ (dimensionless coefficient weighing the relative effects of inflow and outflow on the reach storage) containing all the information about the river or canal reach.
- The first or second order Hayami model is derived from the diffusive wave equation (Alvarez B. X. (2004), Chentouf, B. (2001)), a simplified form of the full Saint-Venant equations.

- A linearized model was used by Malaterre (1994), based in a linearization of the full Saint Venant equations (a finite difference approach of the Saint-Venant equations) around a steady state.

We can generalize all these models, and give a global point of view of them, as all these models have to represent the state vector, which describes the time evolution of the hydrodynamic variables of a canal (water level and velocity) in terms of control actions. An example of the equations that represents a linear model can be written as follow:

$$x(k + 1) = [a_1]x(k) + s(k + 1) \quad (4.1)$$

$$s(k + 1) = [a_2]u(k + 1) + \dots + [a_n]q(k + 1)$$

The state vector $x(k+1)$ is the state vector at time step $(k + 1)$ and it contains a set of n_x elements of water level and velocity at every one cross sections. The vector $s(k+1)$ is the input data vector which contains the gate trajectories, extracted flows, demand deliveries,... at time step $k+1$. The matrixes $[a_1] \dots [a_n]$ are constant and contain the linearization of Saint-Venant equations. The vector $u(k+1)$ contains the gate position trajectories from the time step k to $k+1$ (see Figure 4.2). The vector $q(k+1)$ contains the changes in flow rate from the time step k to $k+1$ which is called the extracted flow vector.

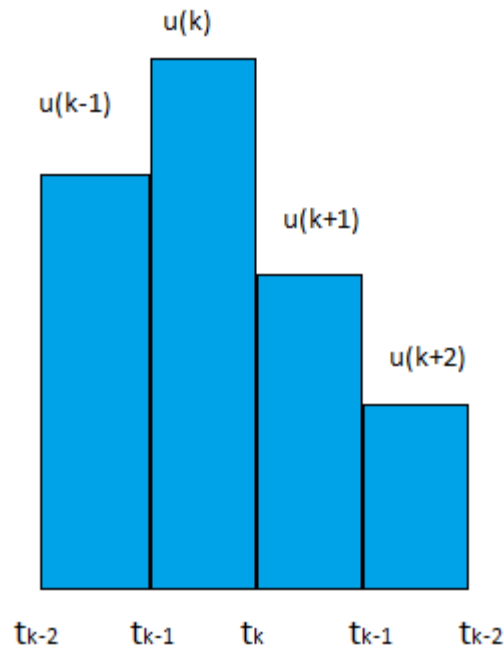


Figure 4.2: A gate trajectory.

We can define this state vector during a predictive horizon, for instance, between the time step k_{I+1} to k_F . The vector, which compiles all terms, is called prediction vector, that is, the output data of the model:

$$X_{k_{I+1}}^{k_F} = [x(k_I + 1)^T + \dots + x(k_F)^T]^T \quad (4.2)$$

The same way with the gate position vector ($U_{k_{I+1}}^{k_F}$) and the extracted flow vector ($Q_{k_{I+1}}^{k_F}$):

$$U_{k_{I+1}}^{k_F} = [u(k_I + 1)^T + \dots + u(k_F)^T]^T \quad (4.3)$$

$$Q_{k_{I+1}}^{k_F} = [q(k_I + 1)^T + \dots + q(k_F)^T]^T$$

We can define the prediction vector as a sum at every term at every time step.

$$X_{k_{I+1}}^{k_F} = [A_1]x(k_I) + [A_2]U_{k_1}^{k_F} + \dots + [A_n]Q_{k_1}^{k_F} \quad (4.4)$$

For the numerical analysis, the grid used in the numerical method is divided in a huge number of nodes, where every node coincides with a cross section. The canal is divided in many cross sections, but the water level is only measured in some of them, called checkpoints. The values of water level at checkpoints are contained in the water level state vector $y(k+1)$ including n_y elements of the state vector $x(k+1)$.

On the other hand, we can compile the water level state vector for all time instants in the so called prediction output vector, as:

$$Y_{k_I+1}^{k_F} = [y(k_I + 1)^T + \dots + y(k_F)^T]^T \quad (4.5)$$

The $Y_{k_I+1}^{k_F}$ vector contains some of the elements of the prediction vector $X_{k_I+1}^{k_F}$. We write this vector as “Y”.

Now, let us consider a new vector, which will be essential when we introduce the optimization problem. This vector contains the desired values for the prediction output vector (desired water levels at checkpoints) during the predictive horizon. This vector defines the target values in our optimization problem.

$$\bar{Y}_{k_I+1}^{k_F} = [\bar{y}(k_I + 1)^T + \dots + \bar{y}(k_F)^T]^T \quad (4.6)$$

The desired output vector is called \bar{Y} .

The most important features of a linear model, which is defined in equation (4.1), are the variables which define the model. The prediction vector during a predictive horizon (k_I+1, k_F) is defined for the state vector in the time step k_I at every cross sections and the control actions implemented during the predictive horizon. The more cross sections are defined by the state vector, the more accurate will be the predictive vector comparing this to the reality. We could come to the same conclusion talking about the prediction output vector.

The matrixes [A] and [B] are built by constant elements, representing the physical parameters of the canal. This is not completely correct because these parameters represent coefficients as the Manning roughness coefficient that could change over time.

However the elements of those matrixes [A] and [B] can be evaluated off-line, and get these elements in certain periods of time.

4.2.2. Non-Linear model

The non-linear models solve the full Saint Venant equations simultaneously for unsteady flow along the canal. They provide the most accurate solutions available for calculating an outflow hydrograph while considering the effects of canal storage and wave shape (Bedient and Huber, 2002). The models are categorized by their numerical solution schemes which include the method of the characteristics, finite differences, finite element methods, finite volumes or others.

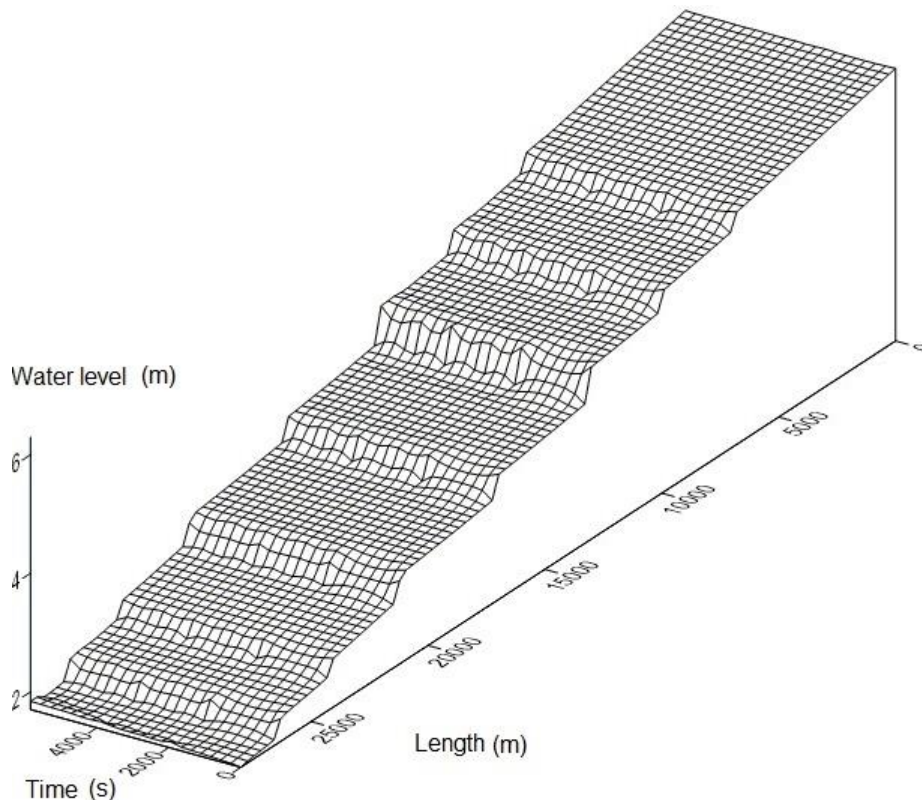


Figure 4.3: Display of the values obtained by a computer model of a canal with eight pools with a total length of 28 Km during a predictive horizon of 6000 seconds. The canal is in steady state, therefore the gate position, changes in flow rate, demand deliveries are fixed over the time. The displayer used in this case is the Surfer 8.

The method of characteristics is commonly used and it is based on the characteristic form of the governing equations. The two partial differential equations (Saint-Venant equations) are replaced with four total differential equations and solved in the x, t

domain. The four equations are commonly solved using explicit or implicit finite difference techniques (Amein, 1966; Liggett and Woolhiser, 1967; Baltzer and Lai, 1968; Ellis, 1970; Strelkoff, 1970; Gómez, 1988).

Numerical solution schemes are often referred to as being explicit or implicit. When a direct computation of the dependent variables can be made in terms of known quantities, the computation is said to be explicit. Much research has been performed on this topic (Garrison et al., 1969; Liggett and Woolhiser, 1967). When the dependent variables are defined by coupled sets of equations, and either a matrix or iterative technique is needed to obtain the solution, the numerical method is said to be implicit. Again, this topic has been well researched by Amein and Chu (1975), Amein and Fang (1970), among others. The implicit method has fewer stability problems and can use larger time steps than the explicit method.

Using nonlinear models in control algorithms is motivated by the possibility to improve the accuracy of the model that represents the system by introducing the nonlinearities of the process. The applicability of nonlinear models in forecasts is a point of discussion still open, although it seems that less, as the current computers solve the full Saint-Venant equations very fast. In this way, transport and canal networks where the watermaster wants to keep close of a steady state without huge disturbances, the potential improvement of using nonlinear models may seem small. However in canals where the state vector has a significant variation of water levels and flows, mainly due to external disturbances, the advantages of using nonlinear models can be significant.

We can define a nonlinear model as follows:

$$x(k + 1) = f[x(k), s(k + 1)] \quad (4.7)$$

$$s(k + 1) = u(k + 1) + \dots + q(k + 1)$$

Where f is a vector function which depends on the state vector of a previous time step (k), the sluice gate trajectories at time step $k+1$, the changes in flow rate at time step $k+1$, the scheduled demands at time step $k+1$,....

The vector function is also dependent on the same physical parameters as in a linear model: Manning coefficient, local losses coefficients.

We have chosen a non-linear model to solve the full Saint-Venant equations in this thesis which uses the characteristic curves method, due to the fact that the computation time for solving these equations is not a problem for the current computers.

4.3. Optimization problem

4.3.1. Introduction

In a simplest case, an optimization problem consists of maximizing or minimizing a function by systematically choosing input values from an allowed set, and computing the value of the function. This function is the so called objective function.

Basic components of an optimization problem are defined as:

- **The objective function.** It expresses the aim of the model which is either to be minimized or maximized. For example, in a manufacturing process, the aim may be to maximize the profit or minimize the cost. Comparing the data prescribed by a user-defined model with the observed data, the aim is minimizing the total deviation of the predictions based on the model from the observed data. In designing a predictive control for a canal, the goal is to minimize the error between the measured and the desired water level minimizing the gate movements.
- **The control action variables** of the objective function. In the manufacturing problem, the variables may include the amounts of different resources used or the time spent on each activity. In the design of a predictive control for a canal, the variable could be the sluice gate trajectory during a predictive horizon.
- A **set of constraints** are those which allow the unknowns to take on certain values but exclude others. In the manufacturing problem, one cannot spend negative amount of time on any activity, so one constraint is that the "time" variable which cannot be non-negative. A set of constraints for a predictive control algorithm for canals could be to limit the gate movements between time steps, or the minimum and maximum sluice gate position.

The objective of an optimization problem is to find values of the variables that minimize or maximize the objective function while satisfying the constraints. In optimization

problems, the data prescribed by a model does not match with the target data, and the main goal is to make these values matching by modifying a set of unknowns or control variables.

Which would it be the objective function in our optimization problem? The objective function is defined from the deviation between the model data (i.e. the prediction output vector) and the desired output vector. On the other hand, the control action variables could be the gate trajectory vector or the extracted flow vector.

We can classify the optimization methods as:

- The classic methods are the common methods used in optimization, among them, the lineal optimization, non-lineal optimization, dynamic optimization.
- The other optimization methods are the metaheuristic optimization methods intimately linked to artificial intelligent, for instance, the evolutionary algorithms or the simulated annealing method (Theodore W. M. et. al., 1999).

The main difference between both optimization methods is that the classical methods calculate a local minimum and guarantee a local minimum, while the metaheuristic optimization methods calculate a global minimum but they do not guarantee to reach a global minimum.

The classic methods are robust and the computational cost is very low. These methods have been used in control algorithms giving excellent results. An optimization problem subject to constraints can be stated as follows:

$$\begin{aligned}
 \text{To find } X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \text{ which minimizes } f(X) \text{ given} & \quad (4.8) \\
 g_i(X) \leq 0, \quad i = 1, 2, \dots, m & \\
 l_j(X) = 0, \quad i = 1, 2, \dots, p &
 \end{aligned}$$

Where X is an n -dimensional vector called the design vector, $f(X)$ is the objective function, and $g_i(X)$ and $l_j(X)$ are the inequality and equality constraints, respectively. When the constraints $g_i(x)$ and $l_j(x)$ are active, the optimization problem is constrained.

The number of variables “n” and the number of constraints “m” and/or “p” need not be related in any way. This type of problem is called a constrained optimization problem.

If the locus of all points satisfying $f(x_i)$ is equal to a constant “ C_i ” (Figure 4.4), it can form a family of surfaces for several control action variables (x_i) in the design space called the objective function surfaces. When we draw the objective function surfaces with the constraint surfaces, see Figure 4.4, we can identify the optimum point. When we have four or more design variables, we have to solve the problem as a mathematical problem and this visualization is not possible. It is possible graphically only when the number of design variables is three or lower.

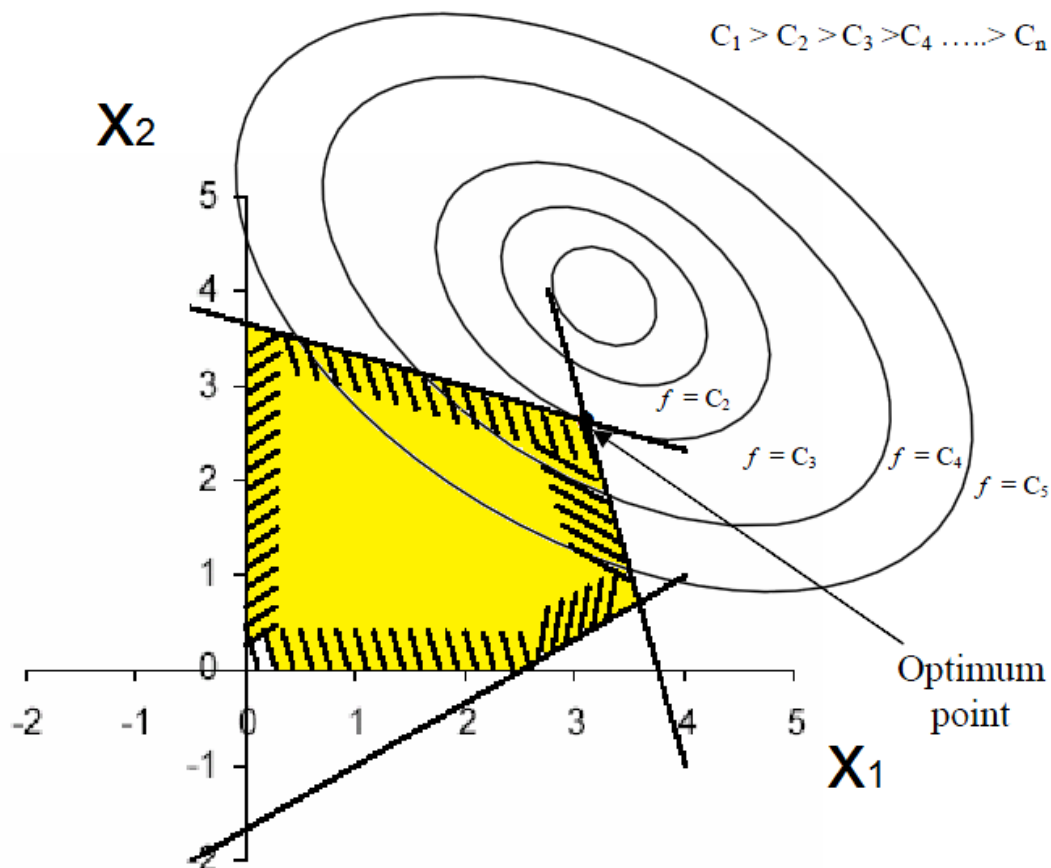


Figure 4.4: Optimum point in a constraint optimization problem.

Optimization problems can be defined without any constraints and they are called unconstrained optimization problems. The field of unconstrained optimization is quite large, and there are a lot of algorithms and software available. In this thesis we will use both constrained and unconstrained optimization problems.

4.3.2. Desired output vector

There are many ways of defining the water offtake along a canal during a given forecasting horizon, for instance from side exits by gravity through orifices as in Figure 4.5, probably the most used, although the pump station is used too. The gravity offtake is very common, because small deviations of the water level do not greatly affect the flow rate deliveries. This kind of hydraulic structure is not able to dry up the canal because the orifice is usually set above the canal bottom. In contrast, the pump is able to dry the canal depending of the extracted flow. In order to maintain operational the water offtake, water level at the target must be higher than y_0 (Figure 4.5), and the flow rate delivered by the orifice is related with the square root of the water level above the y_0 . The control of delivered flow rate is kept by controlling the water level.

On the other hand, we could set a certain desired water level value at every orifice offtake to achieve a particular scheduled delivery during a predictive horizon, as the flow rate depends on the water level at the orifice offtake. The watermaster keep the desired water level at the checkpoints to fulfill the supplies scheduled and the farmers obtain a hydrograph that meets their needs. Sometimes a farmer ignores the irrigation scheduling and disturbance caused by him will affect the scheduled deliveries from other farmers. Although it is impossible that an external disturbance does not affect the scheduled deliveries from others farmers, the feedback controller must recompute a new sluice gate trajectory to reach quickly the desired water level at all checkpoints, for restoring as soon as possible the scheduled demand deliveries.

The external disturbances are unknown in the process, for that reason it is not an easy way to estimate the desired water level. The desired water level is the value in order to meet the needs of the farmers, and the deviations between the desired water level and the real is proportional to the deviations between the delivered flow rate and the scheduled delivered flow rate. There is a direct link between the disturbance and the deviations of demand deliveries. The delivery schedule is fulfilled as well as the desired water level at particular cross sections are equal to the water level measurements.

The water level measurements are done in particular cross sections or checkpoints usually located next to a sluice gate or pump station. As we will check in next chapters, the best mathematical approximation for an optimization problem would be to use the

water level and velocity at every cross sections in the canal and calculate the deviation between the desired and measured water level at each cross section. But it would be impossible to get measures at every cross section of the canal.

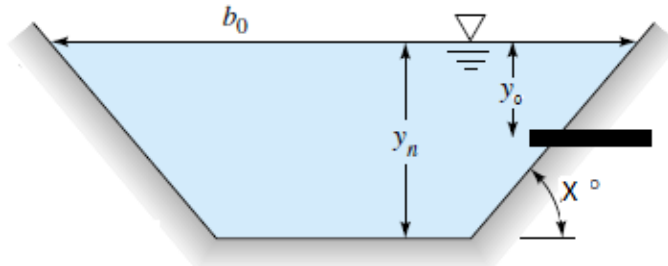


Figure 4.5: Orifice offtake in a cross section.

The aim of the objective function is minimize the error between the measured and the desired water level and this function may be chosen to meet a variety of conditions, as we will see in the next section 4.3.3.

4.3.3. Objective function

The goal of the optimization process is to find the parameter values that result in a maximum or minimum of a function called the objective function. We can classify the objective functions as follow:

- The type of objective function: linear or quadratic.
- The variables involved in the objective function.
- The checkpoints or target points evaluated by the objective function.

The objective function defined in equation (4.8) should be minimized and the complexity to derive this function is linked with the type of objective function and the constraints.

In predictive control in canals, the variables involved in the objective function could be the water level, and the variability between the computed and desired water level at the checkpoint would be the error which should be minimized. This error can be established in the objective function in different forms, quadratic (4.9) or linear.

$$J_{k_1}^{k_F} = \frac{1}{2} \left\| r'_{k_1}{}^{k_F} \right\|_T \quad \text{quadratic objective function} \quad (4.9)$$

$$r'_{k_1}{}^{k_F} = Y_{k_1}^{k_F} - \bar{Y}_{k_1}^{k_F} \quad \|r'\|_T = r'^T [Q'] r'$$

T is a weight matrix

From the type of objective function and constraints, we could also classify the optimization problem as follow:

- If the objective function and the constraints are linear, the optimization problem is linear.
- If the objective function is quadratic and the constraints are linear, the optimization problem is quadratic.
- If the objective function is quadratic and the constraints are nonlinear, the optimization problem is nonlinear.

The variables involved in the objective function could be the water level or the flow rate, and the variability between the computed and desired water level or the computed and desired flow-rate ($\bar{Q}_{k_1}^{k_F}$) at the checkpoint is the error which should be minimized. The next equation shows this kind of objective function:

$$J_{k_1}^{k_F} = \frac{1}{2} \left\| r'_{k_1}{}^{k_F} \right\|_T \quad \text{or} \quad \frac{1}{2} \left\| o_{k_1}^{k_F} \right\|_T, \quad (4.10)$$

$$r'_{k_1}{}^{k_F} = Y_{k_1}^{k_F} - \bar{Y}_{k_1}^{k_F}; \quad \|r'\|_T = r'^T [Q'] r'$$

$$o_{k_1}^{k_F} = Q_{k_1}^{k_F} - \bar{Q}_{k_1}^{k_F}; \quad \|q\|_T = q^T [Q'] q$$

We have introduced the water level and the flow rate, both controlled variables, to define the objective function, but we could use a control action variable as the gate openings (U) to evaluate the objective function. The main goal is to minimize the deviation between the desired and measured water level, but it is also important to minimize the sluice gates movement, because an excessive gate movement causes an engine wear of the gate and a waste of energy for every gate movement which we could be avoided. The objective function can be dependent of both. The next equation shows this objective function:

$$J_{k_1}^{k_F} = +\frac{1}{2} \|r_{k_1}^{k_F}\|_T + \frac{1}{2} \|s_{k_1}^{k_F}\|_{T'} \quad (4.11)$$

$$s_{k_1}^{k_F} = U_{k_1}^{k_F} - \bar{U}_{k_1}^{k_F}$$

Where $\bar{U}_{k_1}^{k_F}$ is the gate trajectory with minimum operating cost.

In predictive control, the gate position could be a variable involved in objective function too, but we would have to define the gate trajectory with minimum operating cost which we call the desired gate trajectory.

On the other hand, it would be excellent to have a big number of checkpoint to establish a more effective control of the canal state but the checkpoint or target points evaluated by the objective function are usually limited by the number of sensors.

Chapter 5

Overall Control Diagram

5.1. Introduction

The main objective of this thesis is to develop an on-line predictive control able to operate the canal in real time rectifying the gate trajectory in case of disturbances, establishing the desired behavior of the canal by the watermaster. In our opinion, an on-line predictive controller, operating alone and only using as input data the water levels measured at the checkpoints, is not an accurate predictive controller. There are factors that largely affect the canal, as disturbances or changes in Manning coefficient that the on-line predictive controller needs to know to recalculate the optimal sluice gate trajectory. For this reason, we define a set of algorithms that operate all together in parallel but not all of them necessarily operate in real time, as the period of time that each one is working depends on the variability of the process in time. All the information, obtained by these algorithms, is supplied to the on-line predictive controller to meets its objective.

5.2. Structure Diagram

Although there are multiple forms to define an overall scheme for canal control, we show our ideal solution of an overall control diagram of a canal in figure 5.1. This figure will be commented in several chapters in this thesis, because the algorithms developed for us follow the structure of this diagram. Every process as well as the task developed for each algorithm is introduced in the next paragraph (Figure 5.1).

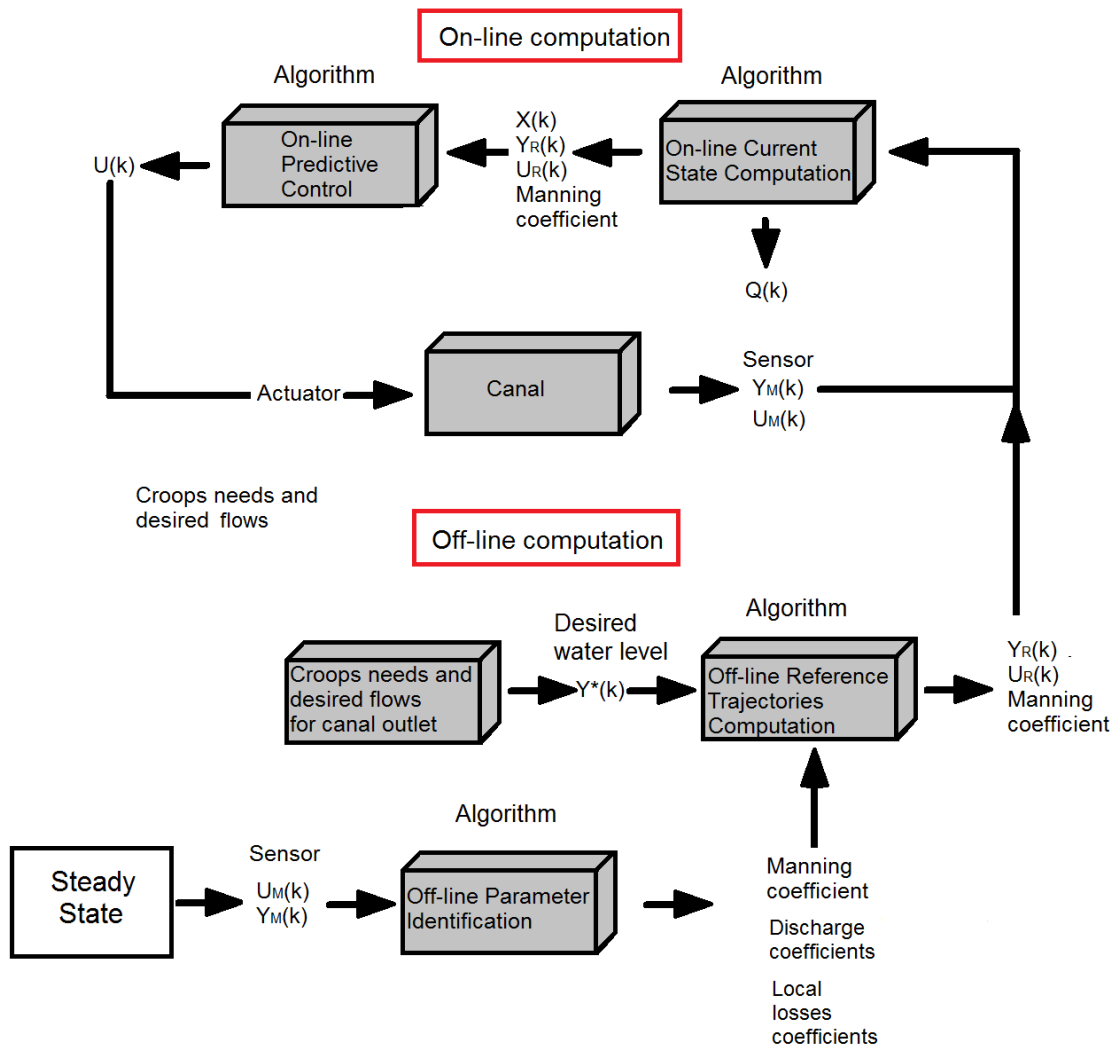


Figure 5.1: Overall control diagram of irrigation canal.

1. «**Crop needs and desired hydrographs for canal outlets**»: The hydrographs at the lateral diversion points of the main canal are calculated on the basis of the water demands. They are fixed considering the farmer requirements and others demands accepted by the watermaster. The behaviour of the canal supplying these hydrographs determines the “desired behavior” (Y^*) at several cross sections (Figure 5.2).

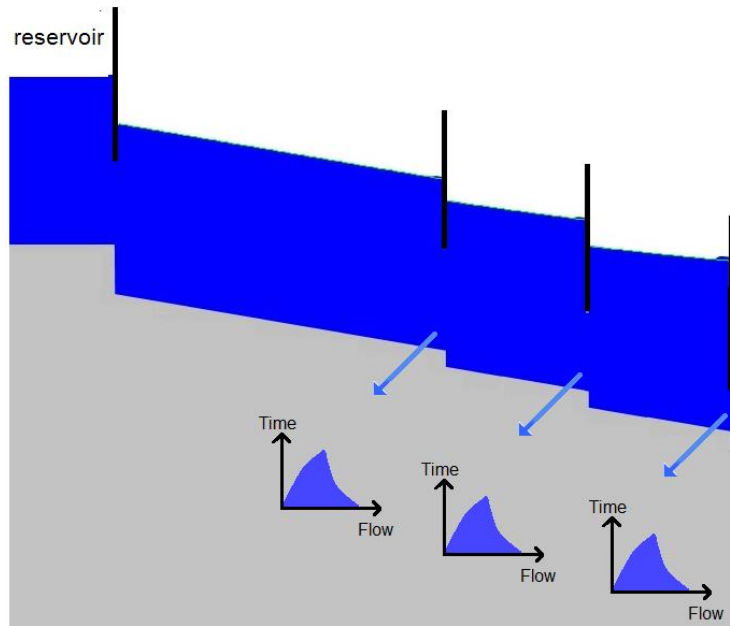


Figure 5.2: The hydrographs at the lateral diversion points of the main canal.

2. **«Off-line Computation of the Reference Trajectories»:** The desired behaviour (Y^*) must be transmitted to the “Reference Trajectories Calculation” algorithm that determines the positions of each gate. This algorithm calculates the optimum behavior (Y_R (reference water level)), which is the one most similar to the desired behavior that is physically possible. We call “ U_R ” the optimum gate trajectories calculated to obtain the optimum behavior (Y_R). They must be calculated off-line (e.g. with an anticipated irrigation cycle). There is an extensive bibliography of feedforward control algorithms which compute the reference trajectories as GoRoSo (Soler, 2003), Bautista et al. (2003), Wahlin et al. (2003).

3. **«Off-line Parameter Identification»:** In case of variations of empirical parameters as the Manning coefficient. The water level measurements obtained during a previous irrigation cycle in a steady state could be transmitted to the “Off-line Parameter Identification” algorithm for estimating these empirical parameters (Figure 5.1). The values of these parameters could be sent to the “predictive controller” and the “on-line current state computation” to be more accurate in their respective processes. This algorithm could also operate on-line and updates the empirical parameters during the irrigation cycle.

4. «**Digital to analog converter —D/A and A/D**»: A computer or control system based on a microprocessor cannot interpret analog signals, and uses only digital signals. Need to be translated, or converted into binary signals.

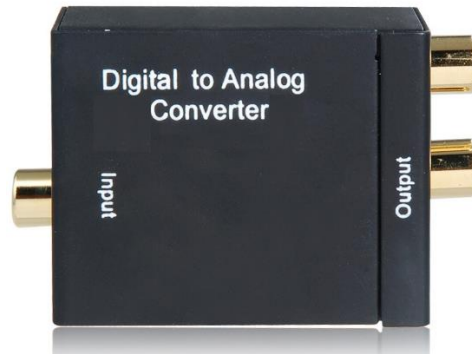


Figure 5.3: A DAC used to convert a digital signal to analog.

A DAC converts an abstract finite-precision number (usually a fixed-point binary number) into a physical quantity (e.g., a voltage or an intensity) (Figure 5.4). Control actions on the canal are implemented by means of digital-analog transformations (D/A) and average measurements obtained for each sampling period ($y_M(k)$ and $u_M(k)$) are obtained by means of analog-digital transformations (A/D). The water level measurements are converted and transmitted to the “On-line Current State Computation” algorithm (A/D converter). On the other hand, the gate trajectories from the “on-line predictive control” are converted and transmitted to the sluice gate actuator (D/A converter).

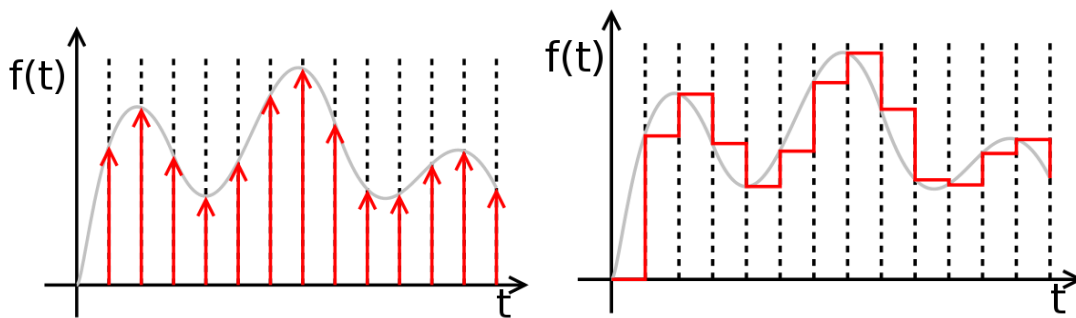


Figure 5.4: The first figure is a ideally sampled signal and the second one is a piecewise constant output of an idealized DAC lacking a reconstruction filter.

5. **«On-line Current State Computation»:** This module uses all values stored in the database as $y_M(k+i)$ and $u_M(k+i)$, which have been measured at the checkpoints in the canal during a past time horizon ($i = 0, \dots, -\lambda$). If the water level measured at checkpoints is different from the desired water level, it is probably due to a disturbance introduced into the system or erroneous roughness coefficients used in the calculation. That is why it is very important to update Manning's coefficients periodically. One of the algorithms developed in this module is able to estimate the disturbances which have modified the measured water level at checkpoints from the desired water level, the algorithm can obtain the current hydrodynamic state of the canal, which is very useful in any case. The hydrodynamic state of a canal is defined as the velocity and water level at each cross-section of the canal. This process has not been developed by any author and for that reason there is not literature review about that kind of algorithm.

6. **«On-line Predictive Control»:** These reference trajectories are transmitted to the next algorithm, called "Predictive Control", which must re-act on-line in case of deviations between the observed data and the desired behavior at the checkpoints in each regulation period (e.g. every 5 min), due to unknown perturbations. The predictive control recalculates new gate positions ($u(k)$) to come back to the reference behavior. There is an extensive bibliography of feedback control algorithms which recalculate the gate trajectories as CARA (Marzouki 1989), Wahlin et al. (2006) and Clemmens et al. (2004).

The diagram (Figure 5.1) shows a cycle of processes, where some of these blocks represent processes in real time, and other blocks represent processes off-line. For instance the blocks that represent: the crop needs and desired hydrographs for canal outlets, off-line computation of the reference trajectories and off-line parameter identification are processes that can be solved off-line, that is, they are not solved in every sampling period K . The remaining blocks should be solved on-line, using data collected at every sampling period. The bigger restriction in real time problems is the computational time which must be as short as possible, and always lower than the regulation period, because the sluice gate actuator receives new gate trajectories from the control algorithm at every regulation period.

The blocks that calculate the reference trajectories are updated periodically, and the update frequency is determined by the irrigation cycle. The reference trajectories are calculated every irrigation cycle according to the scheduled deliveries and the changes in Manning coefficient.

The data recorded from the beginning of the irrigation cycle, as the measured water level at checkpoints, can be used to estimate the current state of the canal; the output data obtained in solving this problem become the input data for the “on-line predictive control”. In this last block (on-line predictive control), the predictive control recalculates a new sluice gate trajectory if the water level at the checkpoints have changed from the desired water level, which depend on the current canal state at each regulation period. To complete each cycle of the overall control diagram, the control algorithm sends a set of sluice gates positions to the D/A converter and this sends a signal to the sluice gate actuator.

From a theoretical point of view, the data input used in an on-line predictive control is obtained in different processes; therefore not all input data used in the on-line predictive control are obtained at the same time. The overall control diagram contains several processes, and data obtained in every of them are used in different blocks, and every of them is defined in a particular step during the overall diagram.

For instance, the off-line parameter identification is not a process estimated on-line, because the main objective is probably to estimate the Manning coefficient, and this parameter is not susceptible to change every regulation period. This process is evaluated at the beginning of the irrigation cycle. The off-line parameter identification is a part quite important in the overall diagram because the canal state is very sensitive to Manning coefficient changes, even to the point that good control algorithm could be useless calibrated with wrong Manning coefficients, as we show in chapter 8. In that sense, the ASCE Task Committee on Canal Automation Algorithms developed a series of test cases to test the suitability of canal control, some of these tests consist in changing the canal conditions as the Manning coefficient to demonstrate the sensitivity of the algorithm to changes in canal conditions over time. We will assume in this thesis that all physical parameter as the Manning coefficient or the energy loss coefficient are set at the beginning of the irrigation cycle, and they are constant during the irrigation cycle.

In the next paragraph, we are going to summarize the rest of processes involved in the overall diagram.

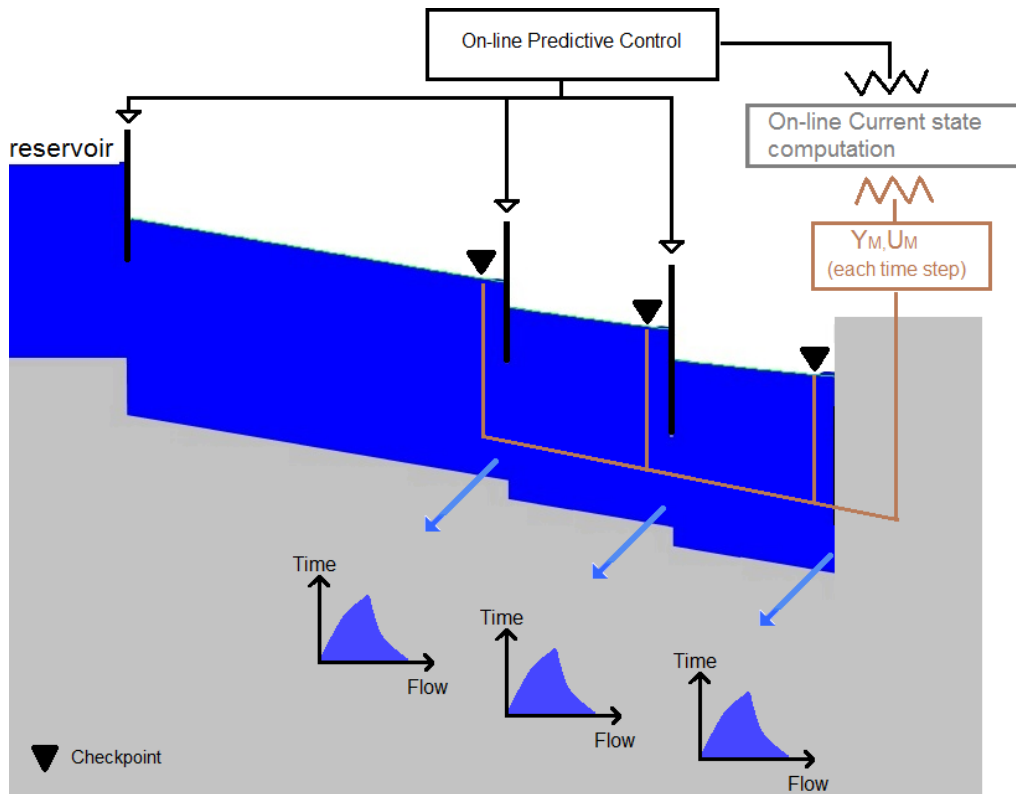


Figure 5.5: A simple view of an on-line predictive control on a canal.

- The process defined as “off-line computation of reference trajectories” runs off-line. The watermaster knows all scheduled deliveries during the irrigation cycle and this process calculates the optimal gate trajectories for a predictive horizon taking into account the scheduled information. This process is estimated by a feedforward controller as GoRoSo.
- The process “on-line current state computation” runs on-line because it estimates the canal state at each time step. This process is developed in this thesis. The algorithm uses data obtained from the beginning of the irrigation cycle to the present time. The data used by this process are the reference water levels, gate trajectories, demand deliveries and the water level measurements during the predictive past horizon. The solution of this process are:
 - The disturbances.
 - The water level and velocity along the canal at each period of time.

- The process "on-line predictive control" runs in real-time, because it recalculates the sluice gate trajectories if the water levels at the checkpoints have changed from the desired water level values in each regulation period (Figure 5.5). This process is dependent on the on-line current state process, as the predictive control needs data input about the disturbances in the canal to set the water level error during the predictive horizon. The input data used by this process are the sluice gate trajectories, the pump trajectories, current canal state and demand deliveries. The on-line predictive control is defined as a feedback controller and this term is also used to describe closed loop controllers introduced before. The output data of this process is a new gate trajectories. This process is developed in this thesis.

5.3. Conclusions

Once defined all the process involved in the overall-diagram, we have identified different ways in which all these process are connected. Are all these processes easy to include in an overall control diagram? Not at all, for the moment we will not take into account the off-line parameter identification, due to the fact that the Manning coefficient could be estimated in advance, as this parameter is almost constant during an irrigation cycle. The off-line computation of reference trajectories is not an obstacle as there are several algorithms in the bibliography which are able to calculate the reference gate trajectory in a predictive horizon as GoRoSo (Soler, 2003) or Tomicic (1989). But the process which defines the current state of the canal is not one performed by algorithms found in the literature review, as it is an algorithm introduced in this thesis and it has not even been proposed before. On the other hand, the process defined as the on-line predictive controller is the main objective of this thesis and this process needs the prediction of the current canal state.

As we will explain in chapter 6, the first idea was not to design an algorithm to identify disturbances and the current canal state, but we must to include this algorithm, because the optimal sluice gate trajectory obtained by the on-line predictive control is dependent on the flow state. The on-line predictive control needs the more accurate estimation of the disturbances and the hydrodynamic canal state.

If there is not a good prediction of the current canal state, the gate trajectories solution will be bad because the estimation of the future error will also be bad, although the on-line predictive controller had been designed from an accurate and robust algorithm.

Some type of controllers, as the PID, whose algorithm consists of three basic coefficients (proportional, integral and derivative), predicts an error based in these three terms. The proportional term produces an output value that is proportional to the current error value, the contribution from the integral term is proportional to both the magnitude of the error and the duration of the error and the derivative of the process error is calculated by determining the slope of the error over time (Figure 5.6). Although this type of controller are able to predict an error for the predictive horizon without knowing the real external disturbance introduced into the system, it would be more accurate establish the real disturbance.

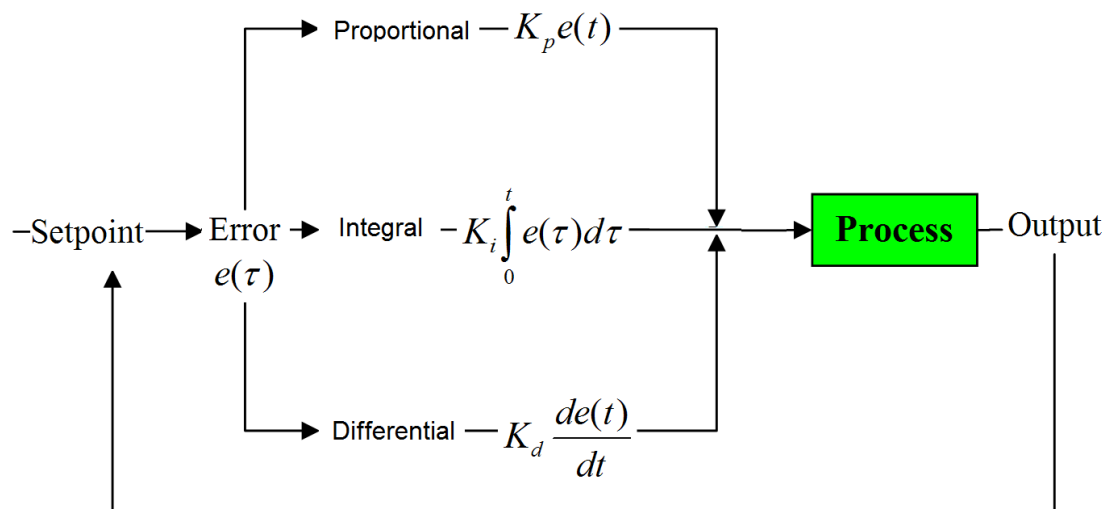


Figure 5.6: A PID controller Block Diagram.

Despite the independence between the different processes, all of them have in common a similar mathematical approach: they can be formulated as an optimization problem, they use a computer model, and all of these processes must modify a certain control action variable for setting a measured variable (simulated by the computer model) as close as possible to a particular desired value defined by the watermaster.

Chapter 6

Canal Survey Information: CSI algorithm

One of the main problems in water management of irrigation systems is the control of the equitable distribution of water among different orifice offtakes. The difficulty of managing a canal is partly caused by the lack of knowledge of the canal state because the scheduled demand is not fulfilled in many times. The farmers extract more water than it is was scheduled; for that reason it is impossible to determine the canal state by the watermaster. This was one of the reasons to develop an algorithm (CSI) which is able to estimate the real extracted flow in certain cross sections during a past time horizon and the hydrodynamic canal state, that is, the water level and velocity along the irrigation canal during the past time horizon.

Both results obtained with CSI could be valuable for a feedback controller. For instance, there are some controller which operate with computer model that does not know the unscheduled deliveries introduced in the canal. If the disturbance is significant, the real hydrodynamic state of the canal is quite different to the canal state obtained by the computer model, so the data introduced to the feedback controller is not good enough and the controller finally fails.



Figure 6.1: Flow control structures in the irrigation ditch of Manresa.

6.1. Introduction

In case that we introduce a flow change in a particular section of a canal, we could estimate the perturbations associated to this flow change from a computer model based on the Saint-Venant equations. CSI does the inverse problem, that is, from perturbations (water level measurements at checkpoints) it calculates the flow change associated to these.

CSI algorithm uses the water level measurements at certain points of the canal, the scheduled demands and the gate trajectories during a past time horizon to obtain the extracted flow in that time horizon and the hydrodynamic canal state at the current time. This mathematical algorithm is introduced in this chapter.

This algorithm was not our priority at the beginning of this thesis, because the main objective was to develop a feedback controller. But several problems relating to the predictive controller forced us to change our point of view. These problems were identified testing our feedback controller, introduced in chapter 9 (GoRoSoBo). Some of these tests (Test Cases) were proposed by the ASCE (Clemmens et al.,1998) and the results were not satisfactory. One of the main reason about the unsatisfactory results was a lack of knowledge of the canal state, and not predict quite well the future behavior of the canal state. This is the reason why the knowledge of the canal state at every moment is so important, as we introduced in chapter 5.

The disturbances that usually modify the canal state are changes in flow deliveries to the farmers, and many times these flow changes are unknown. In that sense, CSI algorithm defines the flow water changes in the canal during a past time horizon, see Figure 6.2.

If a disturbance is introduced to the system, the free surface of the canal will be different to the water surface obtained without flow changes. In that sense, the smaller is the disturbance, the more similar will be the water surfaces. Flow changes can be expressed as a sequence of depths and velocities obtained at some cross-sections in a canal at discrete times. There is only a set of flow changes to define a particular canal state.

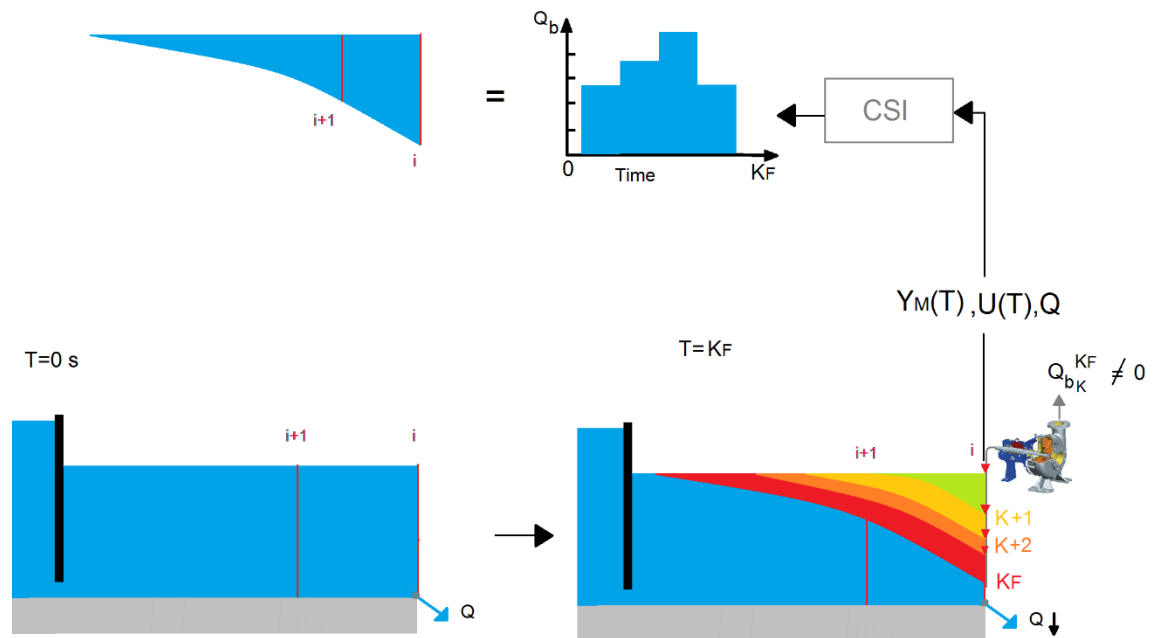


Figure 6.2: Pumping in a cross-section i .

When some hydrodynamic parameters (as the water level) at target points are measured during a past time horizon, it is possible to determine the flow changes introduced in the canal with the aid of CSI. This is the objective in this chapter.

In order to achieve this objective, we introduce the HIM matrix (Hydraulic Influence Matrix) which defines the influence of the flow changes on the canal state. This matrix was defined by Soler (2003), although it was redefined by us in this chapter. Soler introduced this matrix in such a way that the identity of this matrix is to give us a value of the influence of gate movements on the canal state. The purpose of the matrix is to obtain a value of magnitude of water level and velocity on the canal due to sluice gate movements. Instead, we introduce a different point of view of the HIM which establishes the influence of an extracted flow on the water depth and velocity in all points of the canal during a past time horizon. The HIM theory is shown in Appendix I.

In order to make more understandable the meaning of the HIM, we introduce the next example. When we pump in a section of the canal, we modify the canal state, that is, the water level and velocity in particular points at a particular time (Figure 6.3). For instance at $t=10s$, we modify the water level and velocity in cross-sections close to the

pump (cross-section i), but we do not modify the hydrodynamic variables in a far cross section ($i+1$). Instead at time= $40s$, we modify the hydrodynamic variables in all cross-sections from i to $i+1$. The HIM matrix defines how the extracted flow modify the hydrodynamic variables in all sections of the canal during a past time horizon, for instance, from $t=0$ s to 40 s in Figure 6.3. In that sense, the HIM evaluates the range of influence of a flow change along the canal during a period of time.

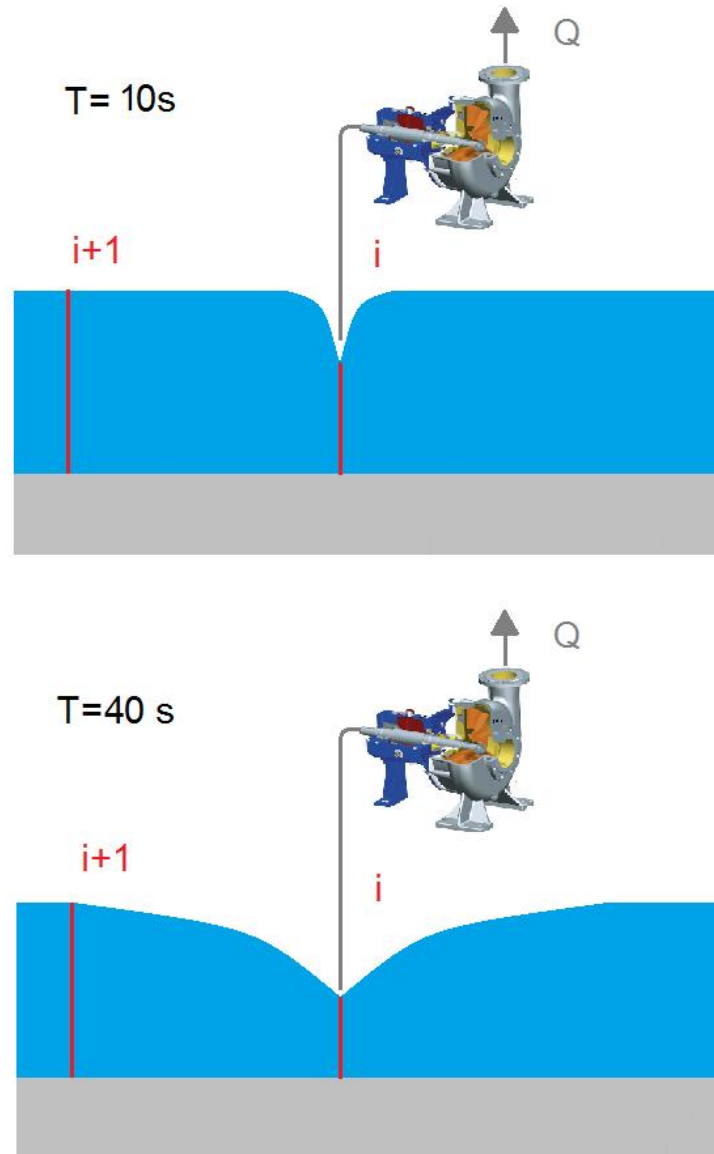


Figure 6.3: Influence of an extraction flow in the canal state.

We choose a set of points of the space/time domain of the canal (Figure 6.3), where we can establish the influence of flow change on the water level and velocity at every

point, see Figure 6.4. Each element of the HIM matrix establishes the relation between an extracted flow at a particular section at a particular time step on the velocity/water level of a particular section at a particular time in the canal.

A big percentage of disturbances that introduce changes in a canal state are caused by flow changes, for instance when a farmer hold a pump to get more supplies or when the farmer increase his deliveries from the orifice offtakes cause disturbances, these are the kind of disturbances considered when we defined the HIM. This matrix gives us the relation between these disturbances and the canal state.

The hydraulic influence matrix is the main tool to the CSI algorithm which calculates the history of flow rate changes during a past time horizon in a canal.

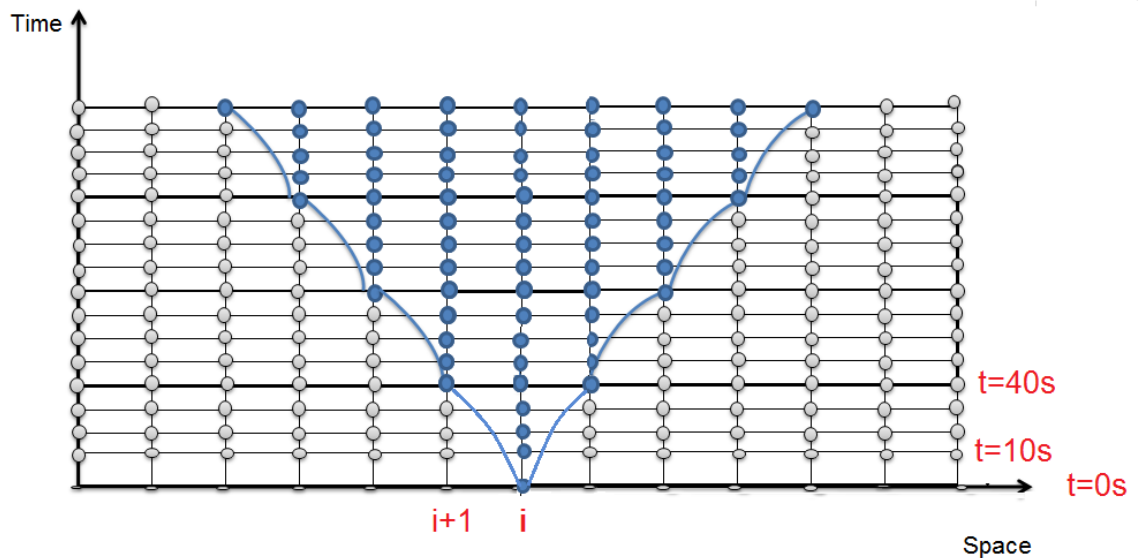


Figure 6.4: Pumping in a cross section i of a computational grid. The extracted flow influences the hydrodynamic variables in blue dots sections.

6.2. Problem statement

In many cases, the objective in an irrigation canal is to keep constant the water level in certain points of the canal where there are orifice offtakes. In that sense, the watermaster fulfils the scheduled demand previously established with the farmers.

Keeping constant the water level is difficult in a canal because there are many outlets along the canal and it is not easy to control the discharges in all of these points. For that reason, we would like to estimate the unknown flow changes which modify the desired water level at the checkpoints. More specifically, our objective is to obtain the flow changes (control inputs) over a scenario (past time horizon) in which we know the initial condition of the canal, the gate trajectories, the water level evolution at specific cross-sections (checkpoints) and the offtake scheduling demand. This objective is achieved by the CSI algorithm.

This algorithm establishes an objective function defined from the total deviation between the measured data (water level measurement) and the data obtained by a computer model (predicted water level) considering an extracted flow. This objective function is minimized from an optimization procedure with the objective to obtain the optimal extracted flow. In that sense, CSI solves an inverse problem as an unconstrained optimization procedure.

Some variables and parameters that CSI uses to solve the optimization problem, are obtained by a computer model describing the free-surface flow in canals, which can be defined from the Saint-Venant equations. These are constituted by a pair of equations that forms a quasilinear partial differential system. Moreover, we need to use some other equations to describe the flow along longitudinal structures like siphons, jumps, etc. Also, there are cross structures for control of flow like gates, weirs, off-takes, etc. Since the flow equations do not have analytical solution, we choose a set of points of the space/time domain to form a computational grid where the solution is computed by using a numerical procedure (Figure 6.5).

6.2.1. Definition of the optimization problem

As we explained before, the CSI algorithm needs as input data, the water level measurements at certain points (checkpoints). Now, let us consider a vector, which contains the water level measurements in the checkpoints at time instant k :

$$y_M(k) = [y_{m1}(k), \dots, y_{mi}(k), \dots, y_{mnc}(k)]^T \quad (6.1)$$

Finally, all vectors (6.1) for each time instant of the past time horizon are joined to define the measured water level vector, whose dimension is n_c , where n_c is the number of checkpoints. We define this vector as:

$$Y_M = [y_M(1), y_M(2), \dots, y_M(k_F - 1), y_M(k_F)]^T \tag{6.2}$$

We can check the measured water level vector values in a computational grid in Figure 6.5 (yellow dots) whose dimensions is n_y , where $n_y = k_F \times n_c$.

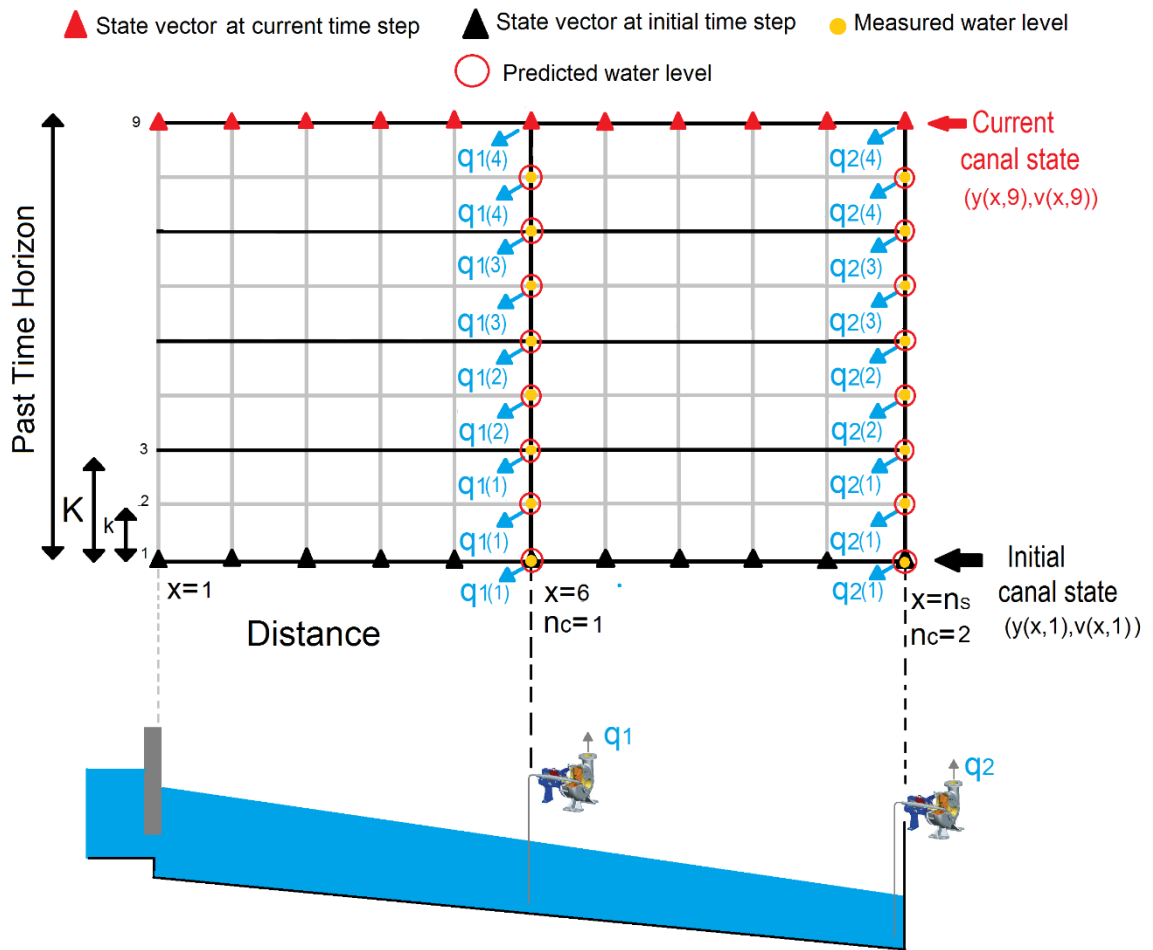


Figure 6.5: Sketch of a numerical grid of a canal with two pools that is controlled by two checkpoints downstream of each pool. There are pump stations close to each checkpoints. Pump flow trajectories are defined with four operation periods. Also, it shows the x/t -dots where the flow behavior is defined. Notice that "K" with capital letter denotes time interval of control and "k" with small letter denotes time instant of simulation.

We could obtain the predicted water level from the state vector $x(k)$ which is defined as the vector containing the numerical solution at the time instant k of all the discretization points:

$$x(k) = \left[y_1(k), v_1(k), \dots, y_i(k), v_i(k), \dots, y_{n_s}(k), v_{n_s}(k) \right]^T \quad (6.3)$$

where $y_i(k)$ and $v_i(k)$ = water depth and mean velocity at point i ; and n_s = number of cross sections in which the canal is discretized. In this way, the vector $x(0)$ is the known initial condition.

We can check the state vector at the current time which defines the current hydrodynamic state in a computational grid in Figure 6.5 (red triangles) and the state vector at initial time step (black triangles).

We may include all state vectors (6.3) for each k -instant of the past time horizon into a single vector that is called prediction vector:

$$X_1^{k_F} = \left[x(1)^T, x(2)^T, \dots, x(k_F - 1)^T, x(k_F)^T \right]^T \quad (6.4)$$

where k_F = final instant of the past time horizon (Figure 6.5). The dimension of this vector is $n_x = (2 \times n_s) \times k_F$.

Given the initial conditions $x(0)$ and the boundary conditions needed for simulation, the prediction vector is uniquely determined by the extracted flow vector.

Taking into account that in our optimization problem, the objective function only evaluates the error in particular points. We are only interested in the water level at target points where we also obtain the water level measurements. We define a new vector that contains the water depth values given at a prescribed number of points (n_c) for $k=1, \dots, k_F-1$:

$$y(k) = \left[y_1(k), \dots, y_i(k), \dots, y_{n_c}(k) \right]^T \quad (6.5)$$

This vector is constituted by a subset of values of the state-vector (6.3). All vectors shown in the equation (6.5) for all the time instants during the past time horizon are

lumped in the so called prediction output vector. The position of the elements of this vector in the grid domain (Figure 6.5) coincides with the elements of the measured water level vector:

$$Y_1^{k_F} = \left[y(1)^T, y(2)^T, \dots, y(k_F - 1)^T, y(k_F)^T \right]^T \quad (6.6)$$

The dimension of the prediction output vector is $n_Y = k_F \times n_C$. The vector (6.6) contains all water depth values and it coincides with the yellow dots (x/t-dots) shown in Figure 6.5.

The prediction output vector is clearly related to the prediction vector (6.4) in the form:

$$Y_1^{k_F} = C X_1^{k_F} \quad (6.7)$$

where C is a matrix, called discrete observer matrix by Malaterre (1994), matrix of dimension $n_Y \times n_X$ and their components are only "zeros" or "ones". This matrix defines the direction of the control logics along a canal pool: downstream level control, upstream level control or control of intermediate water levels.

As we introduced before, CSI calculates the extracted flow at several points (for instance, pump stations) during a past time horizon. In that case, as it is illustrated in Figure 6.6, the pump stations are pumping with an operation period K. Then, the extracted flow trajectories are piecewise functions. The extracted flow vector is defined by lumping together all the extracted flows during the past time horizon, as follow:

$$Q_b = \left[q_1(1), \dots, q_{n_p}(1), \dots, q_1(K_F), \dots, q_{n_p}(K_F) \right]^T \quad (6.8)$$

where the dimension of this vector is $n_Q = n_P \times K_F$, n_P is the number of pump stations and K_F is the number of control operation periods in which the past time horizon is divided.

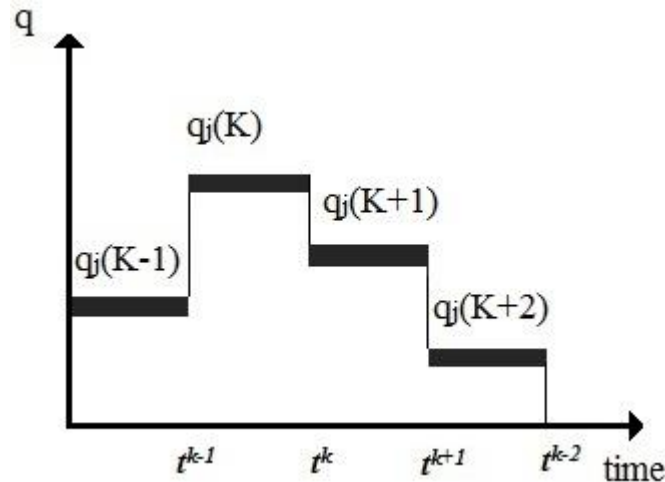


Figure 6.6: Mathematical representation of a pump flow trajectory.

In this way, only Q_b determines the behavior of the canal along the past time horizon. When the extracted flow trajectories are implemented in the canal, the flow response given by the model will be unique. Inversely, one flow behavior is caused by only one set of extracted flow vector.

We can check the extracted flow vector values in a computational grid in Figure 6.5 (blue arrows).

Once CSI has estimated the extracted flow vector, the algorithm can also estimate the state vector at the current time, that is, the current hydrodynamic canal state (water level and velocity in all cross sections of the canal) from the real extracted flow, scheduled demands, the initial conditions, the gate trajectories during the past time horizon.

6.2.2. Optimization problem: Objective function

As we introduced in chapter 4, an objective function is an equation to be optimized given certain variables that need to be minimized or maximized using nonlinear programming techniques.

The main objective consists to set the flow changes that modify the water level at checkpoints from the measured water levels in canal. The CSI algorithm calculates

the extracted flow trajectories (Q_b) which explain the changes between the measured and predicted water levels.

The objective is to make the prediction output vector more similar to the measured water level vector by manipulating the extracted flow vector. In mathematical terms, the objective is to obtain the extracted flow vector that minimizes the following performance criterion:

$$J(Q_b) = \frac{1}{2} (Y_1^{KF}(Q_b) - Y_M)^T [Q'] (Y_1^{KF}(Q_b) - Y_M) \quad (6.9)$$

$$\text{Minimum } J(Q_b^*) = F[Y_1^{KF}(Q_b^*)]$$

where Q' matrix is a weighing matrix and the dimension of the matrix are $n_Y \times n_Y$. This matrix could be used to define the level of importance of the water level error in a particular checkpoint. We define this matrix as the identity matrix in CSI. The vector Q_b contains the extracted flow trajectories (6.9).

The minimum of the objective function is also the minimum deviation between the prediction output vector and the measured water level vector at checkpoints. We obtain minimizing this function (6.9) a set of actions (Q_b^*) which establish the best possible approximation between measured and computed water level. This is the goal of the objective function and Q_b^* is the solution, see Figure 6.7.

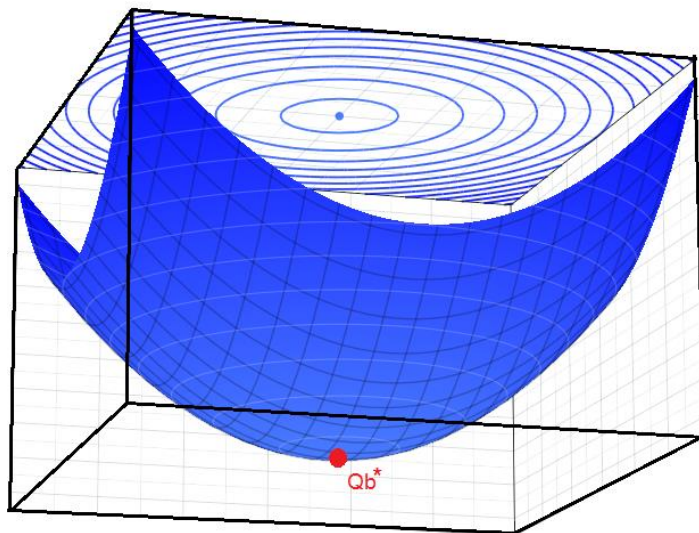


Figure 6.7: Minimum value (Q_b^*) for an objective function.

6.2.3. Optimization problem statement

To summarize this section, the inverse problem is formulated as an unconstrained optimization problem. The optimization problem posed in (6.9) is the classical non-linear problem without constraints. There are many methods to solve an optimization problem in optimization literature. In this thesis, the method used to solve this problem is the Levenberg-Marquardt method, and it is introduced in the next section 6.3. There is an extensive bibliography about the unconstrained optimized problems, see Fletcher (1987) and Numerical Recipes in FORTRAN 77 (1992).

6.2.4. Conclusions of the problem statement

In this thesis, the input variables for CSI will be directly obtained from a computer code. So system input variable as water level measurements are obtained from the computer model. As we can know the water level error at checkpoints, the next step it is to calculate the extracted flow trajectories that better explain the water level changes at checkpoints. For this purpose, we will use the Hydraulic Influence Matrix ($HIM(Q_b)$), although the HIM matrix was introduced by Soler (2003), we have modified the HIM matrix as we are interested in obtaining the influence of a extracted flow on the canal state. In this chapter, every element of the HIM matrix represents the influence of a extracted flow at a particular point on the canal state (water level, velocity in a particular section at a particular time step), so an easy way to find the changes on the canal state when you introduce a change in flow is from the hydraulic influence matrix.

If all flow changes are lumped in an extraction flow trajectories vector ΔQ_b , then multiplying the hydraulic influence matrix by this vector, we would obtain the variation of the hydrodynamic variables associated to these flow changes Figure 6.8, as follow.

$$\Delta X = [HIM(Q_b)]\Delta Q_b \quad (6.10)$$

where ΔX represent the changes in hydrodynamic variable at checkpoints.

In the appendix I at the end of the chapter, we introduce the hydraulic influence matrix where more details can be found.

The name of Hydraulic Influence Matrix is originated because their hydraulic components represent the influence of a flow change over the state vector in all cross-sections at several time steps. The question is, how can I obtain this matrix?

Based on the full Saint-Venant equations, which describes the free surface flow in canals, it is established a set of discretized finite difference equations to calculate the flow behavior, from the extracted flow. The numerical part consists of the application of the numerical method of characteristics to the Saint-Venant equations in order to obtain algebraic equations, and all the influences are lumped together in a global matrix, which is referred to as HIM(Q_b). Based on this system of equations, and using the first derivative $\left(\frac{\partial X}{\partial Q}\right)$ on an analytical process, it can be established the changes in flow behavior by a flow change at a point at a certain time instant.

This matrix makes possible to implement the CSI algorithm as a direct relation between the water level error and the flow changes (Figure 6.8). A detailed description of the method to compile the Hydraulic Influence Matrix is defined in appendix I.

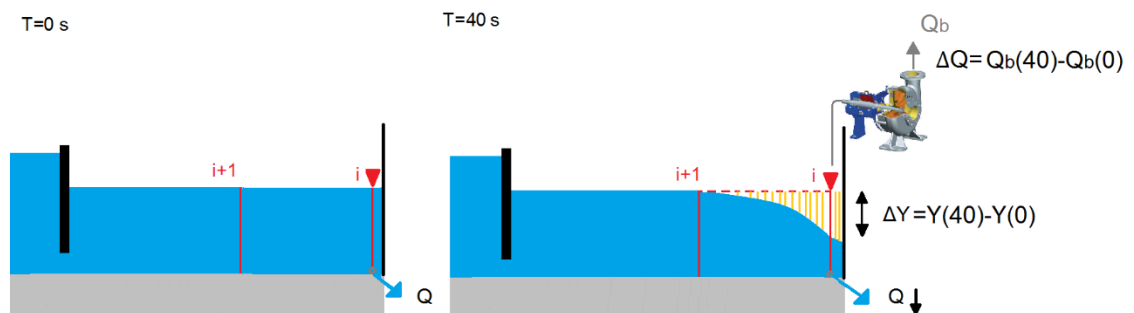


Figure 6.8: Direct relation between water level error and extracted flow.

One of the singularities of this matrix is its condition number which initially caused numerical problems in CSI. This matrix is an ill conditioned matrix because some of them have large influence on the canal state, while other components have small influence on the canal state. In that way, it is possible that some rows or columns are close to zero which results in instabilities in the numerical method. In this case, we had no other choice that improve the condition number of the matrix as CSI has to inverse the HIM matrix.

6.2.5. System output / inputs variables of CSI

To test the numerical examples, we use a computer model to replace the “measurements”. An advantage of using predictive models is that the numerical examples that anyone can test are limitless. The disadvantage is that measurements obtained by a predictive model are not as accurate as the measurements taken in a canal, as we introduced in chapter 4.

The inputs are applied to a control method to predict a control action and influence the output variables of the system. In the Figure 6.9, we show the input and output variables in CSI algorithm taking into account the considered variables in the process, as follow:

- Procedure: CSI algorithm.
- Input variables: Scheduled demands, gate trajectories and water level measurements.
- Output variables: extracted flow vector and state vector at the current time step.

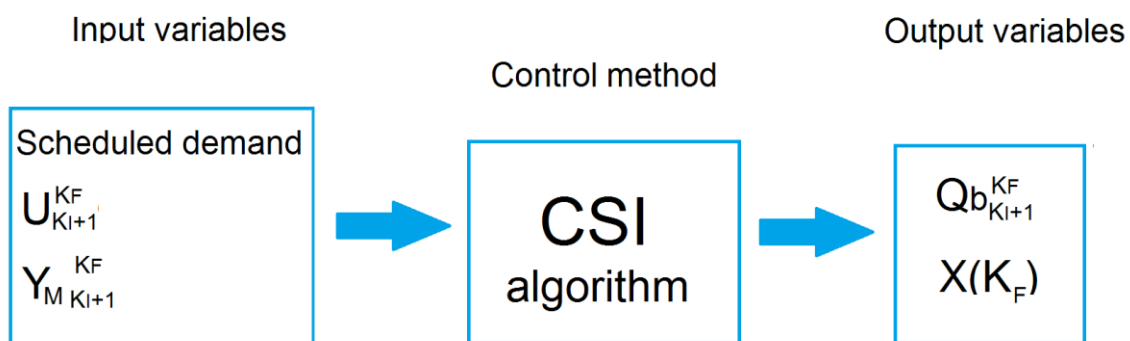


Figure 6.9: Input/Output variables in CSI algorithm.

6.3. Estimation algorithm from unconstrained optimization

CSI algorithm solves an unconstrained optimization problem to obtain the extracted flow vector solution, minimizing the objective function. There are several methods to solve an optimization problem, all of them have to reach the solution (Q_b^*) and fulfil two conditions to be sure that Q_b^* is the better solution. The first condition (6.11) demonstrates that the solution is a maximum, minimum or a saddle, so the first derivative of function in the solution (Q_b^*) is equal to zero. The second condition (6.13) demonstrates that the solution (Q_b^*) is a minimum. These conditions have to be achieved by all optimized methods (constrained and unconstrained). In this way, it is interesting to introduce these order conditions.

6.3.1. First-order and second-order conditions

There is not any auxiliary information about the proposition and lemmas mentioned in this paragraph, but there is an extensive bibliography about the conditions in optimized problems, see Fletcher (1987) and Luenberger (1984).

First-order conditions:

The first-order conditions are necessary to find an optimal solution of an unconstrained optimization problem as:

$$\nabla J_{Q_b}(Q_b^*) = 0 \quad (6.11)$$

where $\nabla J_{Q_b}(Q_b^*)$ is the first derivative of the objective function around the solution (Q_b^*).

We can introduce a definition of a strict local minimizer in which $J(Q_b) > J(Q_b^*)$ for all $Q_b \neq Q_b^*$ sufficiently close to Q_b^* . A stronger definition is set for an isolated local minimizer in which Q^* is the only local minimizer in a neighborhood of Q_b^* .

So developing the first-order condition, we obtain:

$$\nabla_{Q_b} J(Q_b^*) = (\nabla_{Q_b} X_1^{KF}(Q_b^*)^T - Y^*)^T [C]^T [Q'] ([C] X_1^{KF}(Q_b^*) - Y^*) = 0 \quad (6.12)$$

The vector $\nabla_{Q_b} J(Q_b^*)$ is the first derivative of the objective function. The term $\nabla_{Q_b} X_{k_j+1}^{K_F}(Q_b^*)^T$ contains the first derivative of the extracted flow on the state vector. We can express the first derivatives from the water level and velocity variables $(\frac{\partial v_i}{\partial q_{b_i}}, \frac{\partial y_i}{\partial q_{b_i}})$.

Dimension of the matrix $\nabla_{Q_b} J(Q_b^*)$ is $n_Q \times n_X$.

The gradient of the tangent line must be null over $J(Q_b)$ surface at a stationary point or solution point. Although this is a necessary condition, this is not the only one condition as we will see in the next paragraph.

Second-order conditions:

In fact, all minimization methods are only based on locating a point Q_b^* such that $\nabla_{Q_b} J(Q_b^*) = 0$. This may not be a strict local minimum and in general it is referred to as a stationary point (Figure 6.10). We need a second condition to confirm that Q_b^* is a strict local minimum point.

$$s^T \{W(Q_b^*, \lambda^*)\} s > 0; \forall s \in G(Q_b^*) \quad (6.13)$$

Where $G(Q_b^*)$ is a set of the feasible directions. In an unconstrained optimization problem, all directions are feasible because there is not any constraint which prevents any direction, so any "s" is a feasible direction to get Q_b^* .

$$W(Q^*) = \nabla_{Q_b Q_b}^2 J(Q^*) \quad (6.14)$$

Where $W(Q_b^*)$ is the Hessian matrix.

$$\nabla_{Q_b Q_b}^2 = \begin{bmatrix} \frac{\partial^2}{\partial q_{b_1}^2} & \cdots & \frac{\partial^2}{\partial q_{b_{n_Q}} \partial q_{b_1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial q_{b_1} \partial q_{b_{n_Q}}} & \cdots & \frac{\partial^2}{\partial q_{b_{n_Q}}^2} \end{bmatrix} \quad (6.15)$$

These conditions are necessary and enough to demonstrate that Q_b^* is a strict local minimum.

The equation (6.12) refers to the first order necessary condition, as it is based on the first order variations in $J(Q_b^*)$ and therefore first derivative.

The equation (6.13) is the second order necessary condition, and it represents the condition that the Hessian matrix is a positive definite matrix. By definition of this property, if the Hessian matrix was an indefinite matrix, it would be impossible to find a feasible direction ∇Q_b . That is, although Q_b^* complies with one of the conditions requirements (the first condition), we could obtain a value of $J(Q_b^* + \nabla Q_b) < J(Q_b^*)$, and Q_b^* would not be a strict local minimum. The last condition would be able rewritten as:

$$\Delta Q_b^T [\nabla_{Q_b Q_b}^2 J(Q_b^*)] \Delta Q_b < 0 \quad (6.16)$$

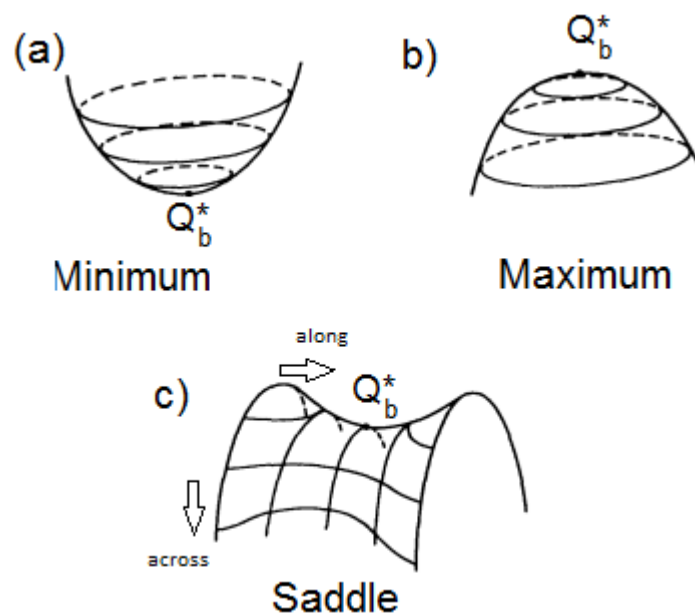


Figure 6.10: Types of stationary point.

Meanwhile all the graphs have zero slope at Q_b^* (Figure 6.10), for (a) there is positive curvature in every direction, for (b) negative curvature in every direction, and for (c) there is negative curvature across the saddle and positive curvature along the saddle.

As it has been written previously, a minimum of the function corresponds to a positive definite Hessian matrix, a maximum to a negative definite matrix, and a saddle point to an indefinite Hessian matrix.

We can write the Hessian matrix from the first derivative of the objective function (6.12):

$$\begin{aligned} \nabla_{Q_b Q_b}^2 J(Q_b^*) &= (\nabla_{Q_b} X_1^{KF}(Q_b^*)^T)[C]^T [Q'] [C] (\nabla_{Q_b} X_1^{KF}(Q_b^*)^T)^T \\ &+ ([C] X_1^{KF}(Q_b^*)^T - Y^*)^T (\nabla_{Q_b Q_b}^2 X_1^{KF}(Q_b^*)^T) \end{aligned} \quad (6.17)$$

The problem is the second derivative parameter $(\nabla_{Q_b Q_b}^2 X_1^{KF}(Q_b^*)^T)$ which is quite difficult to evaluate but it is necessary to do it according to the second order condition to guarantee the solution. To evaluate this second derivative, the algorithm wastes a lot of computational time, so we use another way to evaluate $\nabla_{Q_b Q_b}^2 J(Q_b^*)$ without calculating $(\nabla_{Q_b Q_b}^2 X_1^{KF}(Q_b^*)^T)$ taking into account a simplification for the equation.

For Q_b values sufficiently close to Q_b^* (i.e. for all Q_b in some neighbourhood of Q_b^*), the value of $([C] X_1^{KF}(Q_b^*)^T - Y^*)^T$ is close to zero. The equation (6.17) can be simplified omitting the second derivative term:

$$\nabla_{Q_b Q_b}^2 J(Q_b^*) \approx (\nabla_{Q_b} X_1^{KF}(Q_b^*)^T)[C]^T [Q'] [C] (\nabla_{Q_b} X_1^{KF}(Q_b^*)^T)^T \quad (6.18)$$

When $\nabla_{Q_b Q_b}^2 J(Q_b^*)$ is solved in the simplified form, we use the Gauss-Newton method or Least Squares method to solve (6.18). When $\nabla_{Q_b Q_b}^2 J(Q_b^*)$ is solved in the full form, we use the Newton method to solve it (6.17).

6.3.2. Implemented algorithm

Many techniques for solving this type of problem are available in the optimization literature. The method used to solve the nonlinear optimization problem is the Levenberg-Marquardt method, which is a robust method of easy implementation and this is a special method to solve ill conditioned matrix as the HIM matrix.

The Levenberg-Marquardt (LM) algorithm is an iterative technique that locates the minimum of a function that is expressed as the sum of squares of non-linear functions. It has become a standard technique for non-linear least-squares problems, widely adopted in a broad spectrum of disciplines. LM can be thought of as a combination of steepest descent and the Gauss-Newton method. When the current solution is far from the correct one, the algorithm behaves like a steepest descent method: slow, but

guaranteed to converge. When the current solution is close to the correct solution, it becomes a Gauss-Newton method. The references by Fletcher (1987), Gill et al. (1981) or Luenberger (1984) are useful to solve optimization problems. The theory associated to this method is developed in the appendix II at the end of the chapter.

We choose the characteristics curves to describe the temporal discretization of second order equations of Saint-Venant because it has advantage over other methods evaluated. This method allows the user to understand easily the physics of wave and the algorithm can use large time steps without significant loss of accuracy, as the condition of Courant - Friedrichs - Levy makes possible the use of much larger time steps with consequent saving CPU time.

In the context of simulation, this advantage can be considered as theoretical, although the method uses larger time steps. The solution is also more difficult to find for a large time step than for a short time step due to we need to use numerical techniques for solving nonlinear equations. However, if you consider that each time step the algorithm has to solve a system of equations for each parameter evaluated and build the hydraulic influence matrix, taking into account that each element of this matrix contains a first derivative of the objective function $(\nabla_{Q_b} X_1^{KF}(Q_b^*))^T$, the reduction in time discretization is an advantage.

As it was mentioned, the HIM matrix is an ill conditioned matrix. In that sense, LM give us a certain degree of stability in the algorithm because this method introduces a value in the diagonal of the Hessian matrix to improve the condition of the matrix.

$$H = [\nabla_{Q_b}^2 J(Q_b^*)] + V[I] \quad (6.19)$$

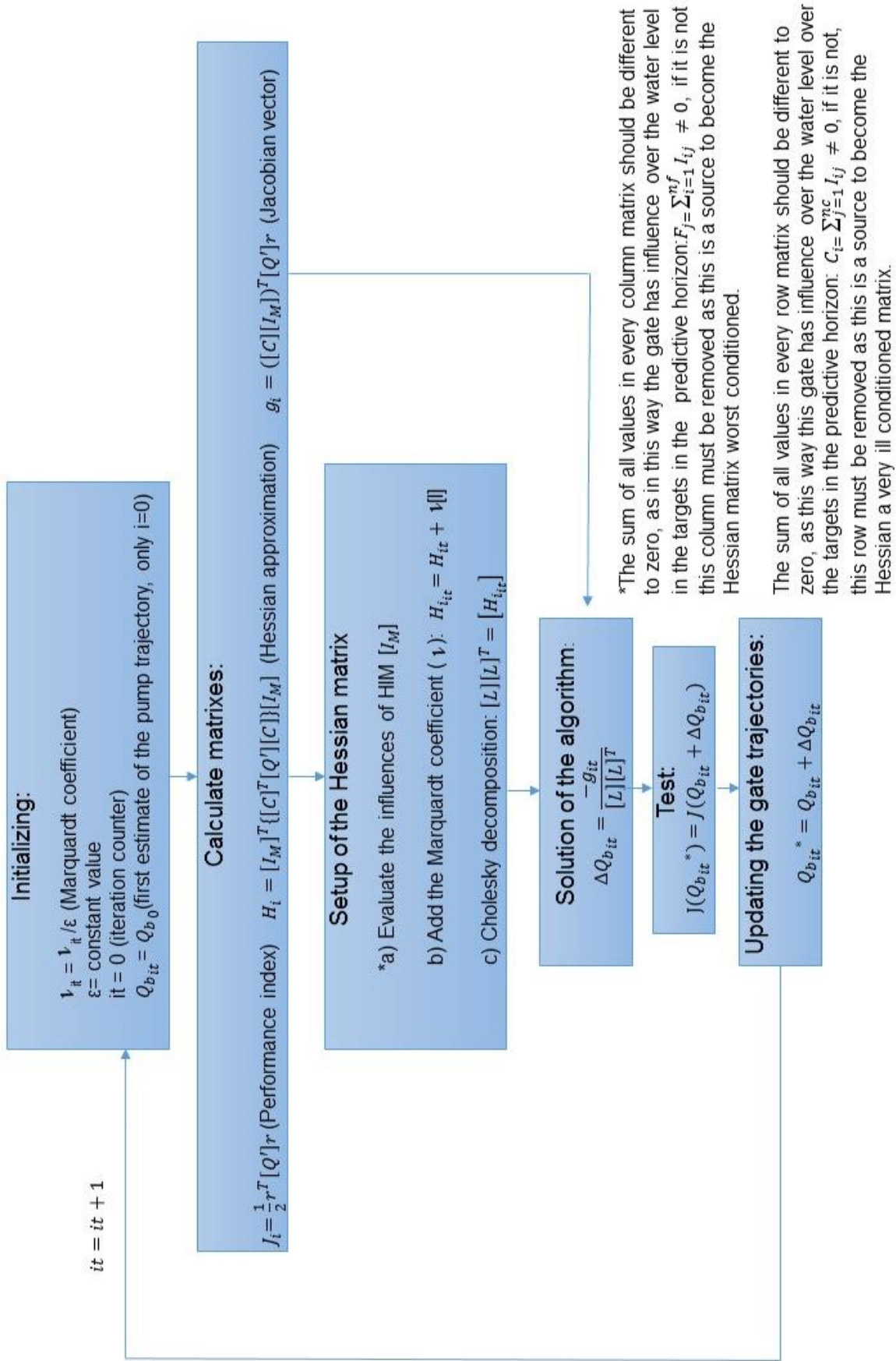
Where $[I]$ is the identity matrix and V is a quantity named Marquardt coefficient.

The steps defined in the implemented algorithm are shown in the next flowchart which is a type of diagram that represents all the process done by the algorithm. Process operations are represented in boxes, and arrows define the sequencing of operations.

We have to define some matrixes and variables that are used in the flowchart:

- $[Q']$ is a weighting matrix.
- $[C]$ is the discrete observer matrix , a matrix of zero and ones.
- $r' = [C]X_1^{KF}(Q_{bit}) - Y^*$ is the residual vector.
- $X_1^{KF}(Q_{bit})$ is the state vector.
- $[I_M] = I_M[X_1^{KF}(Q_{bit})]$ is the Hydraulic Influence Matrix, all evaluated in Q_{bit} .

All the process of the algorithm are defined in each iterative loop, where the solution Q_b^* is calculated in each iteration (it) for a past time horizon (γ).



The $HIM(Q_{bit})$ matrix is an ill conditioned matrix. In that sense, the matrix obtained by the product matrix of $HIM(Q_{bit})^T \times HIM(Q_{bit})$ is also ill defined and for this reason the matrix determinant is close to be zero or negative, and the matrix makes unstable to the CSI algorithm. To overcome this trouble, we can add to the diagonal matrix H_{it} , the Marquardt coefficient equal to the minimal absolute value of the eigenvalue (H_{it}). In this way, we are sure that the matrix is positive definite and the algorithm is stable.

The choice of the Marquardt coefficient may be done in different ways:

- Selected heuristically and kept constant for a particular canal.
- Selected heuristically and variable at each iteration. The coefficient is divided by a constant value (ϵ), so this decreases at each iteration ($\gamma = \gamma / \epsilon$).
- Variable at each iteration with a minimal value in order to ensure that the matrix product $HIM(Q_{bit})^T \times HIM(Q_{bit})$ is positive definite. To achieve this objective, we have to obtain the eigenvalues of this matrix and identify the minimal eigenvalue. In the case of getting a negative definite matrix, the Marquardt coefficient has to be equal to the absolute value of the minimal eigenvalue. If the matrix is definite positive, the Marquardt coefficient is equal to zero.

We chose the second option, the Marquardt coefficient is variable at each iteration.

6.4. Conclusion

The CSI algorithm calculates the disturbances introduced in a system, and the hydrodynamic canal state during a past time horizon. In that sense, the CSI is an excellent tool for a watermaster for two reasons. The first one, the watermaster can know the extracted flow in the canal and the localization of the extraction points. The second one, CSI is a useful tool for a feedback controller which has to know the hydrodynamic canal state in real time.

The CSI algorithm solves an unconstrained optimization problem because it does not need constraints to converge to the solution and the computational time used to solve an unconstrained optimized problem is lower than a constrained optimization problem.

The CSI uses the Hessian matrix in the optimization problem. This matrix is ill conditioned due to variability of the influence of the extracted flow on the state vector, because every extraction point has different value of influence on the checkpoint depending on the distance between them. The influence of a flow change on a checkpoint also depends on the discrete time between the action and the response. To solve the optimization problem, we had to improve the Hessian matrix. This was one of the reasons to use the Levenberg-Marquardt method.

CSI solves an unconstrained optimization problem without any problem, but we had some doubt about the reliability of the algorithm. For instance, the GoRoSoBo algorithm (feedback controller) initially solved an unconstrained optimization problem too, but the solutions obtained by the algorithm were not feasible in some tests, because the solutions were not physically possible, so GoRoSoBo algorithm was modified in a constrained optimization problem.

Taking into account, the optimization problem solved by CSI algorithm and the practical examples performed with it, we are sure that CSI algorithm always calculates a feasible solution. We have introduced all the reasons in the following points by which we are sure about the reliability of the algorithm:

- The CSI algorithm calculates an extraction flow trajectory which is in accordance with the water level variation between the computed water level and the measured water level at checkpoints. Both water levels are physically possible, because the computed water level at checkpoints is the water level at checkpoints without unknown deliveries, whereas the water level measured is directly obtained from measures at checkpoints in the canal. The aim of CSI algorithm is to find out the disturbances caused by unscheduled deliveries. This process is quite different to the process solved by GoRoSoBo, which has to calculate a sluice gate trajectory to reduce the difference between the measured water level and desired water level at checkpoints, and this sluice gate trajectory could be a feasible solution or not depending on the error between both water levels and the period of time in which the sluice gates have to reach the desired water level.
- The CSI has not the same problems associated to the HIM matrix as GoRoSoBo (without constraints). This feedback algorithm was tested with the

Test-Cases proposed by the ASCE (Clemmens et al., 1998) and the gate trajectories obtained by the algorithm in these tests had not a physical sense. In the Test-Cases not all sluice gates are close to a checkpoint. In that sense, the water level obtained at checkpoints allows to the algorithm controls the error between the computed water level and the desired water level, but in case that the sluice gates are close to checkpoints also controlling the excessive sluice gate movements introduced by the algorithm. In the Test-Cases, there is not any checkpoint close to the first gate and this is the most important gate in the canal, because this takes the control on the flow between the reservoir and the canal. If the sluice gate only influences the water level at distant checkpoints and the wave generated by the sluice gate movement decrease a lot with the distance, this sluice gate has a low influence on the water level at the checkpoints. If the algorithm tries to modify the flow conditions with aid of this gate, then the algorithm has not other choice but to introduce excessive sluice gate movements in the system. In these specific cases, the optimization problem would require constraints. This is not a situation that affects to CSI algorithm, because the extraction points are close to a checkpoint in these tests and the difference between the measured and computed water level are due to specific flow changes.

As we will explained in next chapters, the CSI algorithm was not a main objective of this thesis, until we realized the importance to define the current state of a canal for a feedback controller. The results of the CSI algorithm were excellent as we will show in the next chapters 7 and 8.

6.5. APPENDIX I

HIM (Hydraulic Influence Matrix)

In this appendix we introduce the HIM matrix. We introduce the equations which define the hydraulic influence matrix and the influence concept of every parameter of the matrix.

The hydraulic influence matrix concept was introduced by Soler (2003), but the theory of this previous work was modified, as we introduced before. CSI algorithm needs the influence of a state variable on the canal state and this state variable is the flow change, for instance a pump flow. A summary of the HIM matrix, introduced by Soler (2003), was published in his thesis.

6.5.1. Free surface flow equations

The equations of Barré de Saint-Venant (1871) describe the free-surface flow in prismatic canals and are the result of the application of the principles of mass conservation and of the quantity of movement in a controlled volume of short length in the direction of flow. A rigorous deduction of these equations for prismatic canals can be found in Walker (1987), resulting in:

$$\left. \begin{aligned} \frac{\partial y}{\partial t} + v \frac{\partial y}{\partial x} + \frac{A(y)}{T(y)} \frac{\partial v}{\partial x} &= 0 \\ \frac{\partial v}{\partial t} + v \frac{\partial v}{\partial x} + g \frac{\partial y}{\partial x} &= g [s_0 - s_f(y, v)] \end{aligned} \right\} \quad (6.20)$$

where x and t are the independent space and time variables, y is the level of the free-surface from canal bottom, v is the average velocity of all particles of a cross section of the flow, $A(y)$ is the area of wet section which depends on the water level, $T(y)$ is the maximum width also dependent on the water level, S_0 is the canal bottom slope, and finally, $S_f(y, v)$ is the slope friction.

The pair of equations (6.20) will be applicable to reality under the following assumptions:

- The curvature of the free-surface is small.

- The vertical accelerations are disregarded.
- The pressure distribution along an axis vertical to the liquid is the same as in hydrostatic conditions.
- The canal slope is supposed sufficiently small such that its sin is practically null.
- The energy dissipation term is specified through Manning's equation which is used in stationary regime.

$$S_f(y, v) = n^2 \frac{v|v|}{R_H^{4/3}} \quad (6.21)$$

with $R_H = \frac{A(y)}{P(y)}$

where n is Manning's coefficient and $P(y)$ is the wet perimeter which is a function of the water level, and the changes in flow conditions are not fast enough to generate significant wave fronts.

These equations cannot be solved analytically, only numerically. Thus, a variety of numerical methods exist, which can be found, among others, in Gómez (1988). According to Wylie (1969) all the numerical methods of resolution, whether they are explicit, implicit or characteristics methods, present results which are similar when compared to reality, depending more on the accuracy of the starting data than on the particular methodologies. Bearing this key fact in mind, we use the method of characteristics as it helps physical comprehension of the underlying wave phenomenon in free surface flow.

Usually, equations (6.20) are expressed in the classical space and time (x/t) axes, but the so called characteristic curves are used, expressed parametrically with $x^+(t)$ and $x^-(t)$ which locally satisfy the two following differential equations:

$$\left. \begin{aligned} \frac{dx^+}{dt} &= v + c(y) \\ \frac{dx^-}{dt} &= v - c(y) \end{aligned} \right\} \quad (6.22)$$

Thus, the system (6.20) is transformed into the following two ordinary differential equations:

$$\left. \begin{aligned} \frac{dv}{dt} + \frac{g}{c(y)} \frac{dy}{dt} &= g[s_0 - s_f(y, v)] \\ \frac{dv}{dt} - \frac{g}{c(y)} \frac{dy}{dt} &= g[s_0 - s_f(y, v)] \end{aligned} \right\} \quad (6.23)$$

where the first is valid along the curve $x^+(t)$ and the second along $x^-(t)$ and $c(y)$ is the wave celerity.

$$c(y) = \sqrt{\frac{gA(y)}{T(y)}} \quad (6.24)$$

The difficulty of the method lies in the fact that the equations (6.22) have to be solved along the characteristic curves or the local axes that are the solution of (6.23). As this last one is a set of non-linear equations, it obliges us to solve the four equations simultaneously. Fortunately, the curves $x^+(t)$ and $x^-(t)$ always intersect, although they are not orthogonal, and therefore assure hyperbolicity.

In short, solving the system of two partial derivative equations (6.20) is the same as solving the following set of four ordinary differential equations:

$$\left. \begin{aligned} (a) \quad \frac{dv}{dt} + \frac{g}{c(y)} \frac{dy}{dt} &= g[s_0 - s_f(y, v)] \\ (b) \quad \frac{dx^+}{dt} &= v + c(y) \\ (c) \quad \frac{dv}{dt} - \frac{g}{c(y)} \frac{dy}{dt} &= g[s_0 - s_f(y, v)] \\ (d) \quad \frac{dx^-}{dt} &= v - c(y) \end{aligned} \right\} \quad (6.25)$$

The mathematical process of transformation of system (6.20) to the equivalent (6.25) is found in many references such as Gómez (1988), Soler (1996), and Ames (1977) so will not be discussed further here.

The system (6.25) describes the conditions of flow in a canal in the same way as the set of equations (6.20) as it adds no new hypothesis in the transformation. However,

the system (6.25) is limited in the way it is applied. The variable x , which was initially independent, is now dependent on time t , as it is understood in (6.22); then, ((6.25) - (a)) will be true only along the curves that satisfy the equation ((6.25) - (b)) and, in the same way ((6.25) - (c)) will be true along the solution curves of ((6.25) - (d)).

The system (6.25) can be represented in the graph x/t as in Figure 6.11, where the four equations are satisfied at the point of intersection R and therefore the four unknown variables x , t , y and v can be found theoretically. This way, if flow conditions at points P and Q are known, the position of point R can be found and integrated numerically, along with the flow conditions.

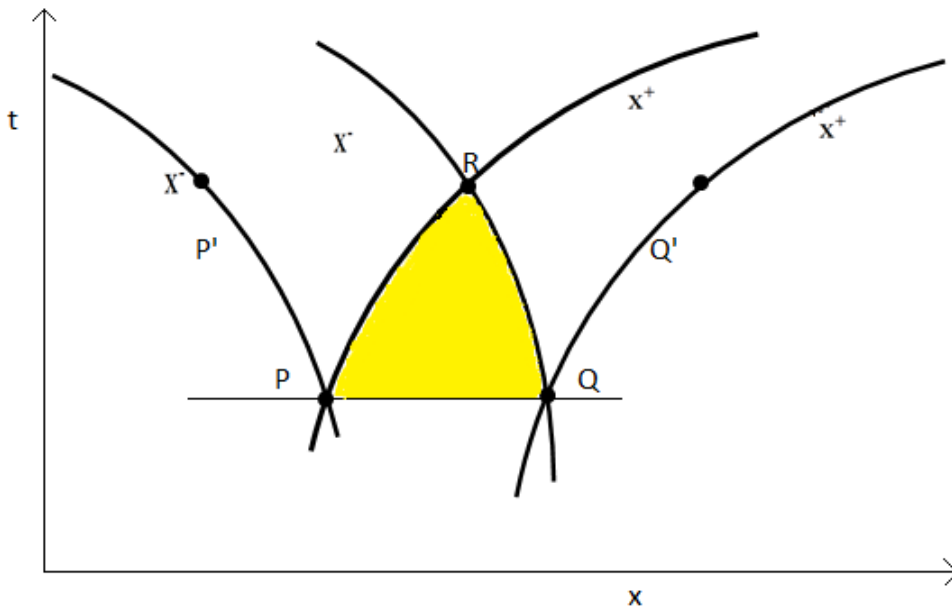


Figure 6.11: The dependence domain of point R .

This can be proved thanks to the first of the uniqueness theorems developed by Crandall (1956) in which he shows that if on a curve on the graph x/t which is not a characteristic curve, as the line PQ in Figure 6.11, the conditions of flow y and v are known, then the set of equations (6.25) determines its solution with uniqueness in the zone marked as PQR . This zone is called the dependence domain of point R because the solution at this point is determined exclusively by the conditions of flow produced

at any point of this domain. That means that any disturbance introduced at any point of the dependence domain will affect the position of point R and the flow conditions.

A complementary concept to the dependence domain is the influence domain. We show in Figure 6.12, the influence domain of point P, that is the set of points of the graph x/t (the area shaded with red vertical lines) which are seen to be affected by the present conditions of flow (y_P, v_P) at this point. In the same way, the area shaded with yellow horizontal lines is the influence domain of point Q and the area shaded with both vertical and horizontal lines is the influence domain of point R.

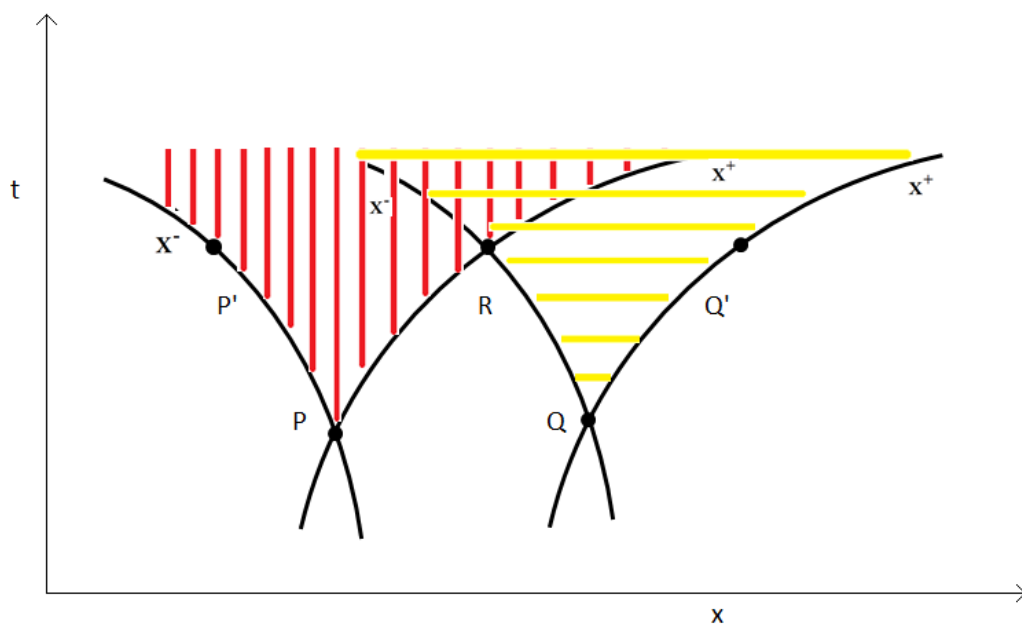


Figure 6.12: The influence domains of points P, Q and R.

Once the concepts of the influence and dependence domains have been introduced, we are able to present the basic objective of the following part. When a flow change is made by a pump, a series of changes to the flow condition are produced. Firstly, near the pump itself and then further away. If the flow change is considered as a disturbance, then we can speak of the disturbance influence domain.

The objective of the next section is to present a form of calculating and quantifying the variation in the conditions of point R that we will call Δy_R and Δv_R when disturbances are introduced to points P and Q which we call Δy_P and Δv_P and Δy_Q and Δv_Q , respectively.

6.5.2. The discretization of the characteristic equations

As previously mentioned, the system (6.20) and the equivalent (6.25) have no known analytical solution, so the use of numerical techniques has, until present, been compulsory. There are many numerical methods that can be used, at least in a theoretical manner. In this thesis we have preferred to use a specific discretization and make the appropriate mathematical developments based on the result of this discretization. In order to have the longest possible integration time period without loss of precision, we have adopted a discretization in finite differences of second order, called in Gómez (1988) as "the method of characteristic curves". If this method is applied to equations (6.25) and we take into account the characteristics curves that contain the points P-R and Q-R (Figure 6.12), respectively, we will obtain the next equations:

$$\left. \begin{aligned} \frac{v_R - v_P}{t_R - t_P} + \left[\theta \frac{g}{c_R} + (1-\theta) \frac{g}{c_P} \right] \frac{y_R - y_P}{t_R - t_P} &= gS_0 - \left[\theta S_{f_R} + (1-\theta) S_{f_P} \right] \\ \frac{x_R - x_P}{t_R - t_P} &= \theta [v_R + c_R] + (1-\theta) [v_P + c_P] \\ \frac{v_R - v_Q}{t_R - t_Q} - \left[\theta \frac{g}{c_R} + (1-\theta) \frac{g}{c_Q} \right] \frac{y_R - y_Q}{t_R - t_Q} &= gS_0 - \left[\theta S_{f_R} + (1-\theta) S_{f_Q} \right] \\ \frac{x_R - x_Q}{t_R - t_Q} &= \theta [v_R - c_R] + (1-\theta) [v_Q - c_Q] \end{aligned} \right\} \quad (6.26)$$

Where $S_{fR}=S_f(y_R, v_R)$, $S_{fP}=S_f(y_P, v_P)$, $S_{fQ}=S_f(y_Q, v_Q)$ and $0 \leq \theta \leq 1$ is the coefficient of average time that indicates the type of numerical scheme used. This is to say, when $\theta=1$ the numerical scheme is implicit, when $\theta=0$ is explicit, and when $\theta=1/2$ the numerical scheme is in central differences or method of the characteristic curves.

If the conditions of flow at points P and Q are known, then from the four equations (6.26) x_P , t_P , y_P , v_P and x_Q , t_Q , y_Q , v_Q are also known and x_R , t_R , y_R , v_R remain as unknowns variables, which can be found by using any of the methods of solving non-linear equations, such as the Newton-Raphson method. The four equations of (6.26) can therefore be re-written in the form:

$$\left. \begin{aligned}
 f_1 &\equiv x_R - x_P - \frac{1}{2}(t_R - t_P)[v_R + c_R + v_P + c_P] = 0 \\
 f_2 &\equiv (v_R - v_P) + \frac{g}{2} \frac{c_R + c_P}{c_R c_P} (y_R - y_P) - g(t_R - t_P) \left(\frac{S_{f_R} + S_{f_P}}{2} - S_0 \right) = 0 \\
 f_3 &\equiv (v_R - v_Q) - \frac{g}{2} \frac{c_R + c_Q}{c_R c_Q} (y_R - y_Q) - g(t_R - t_Q) \left(\frac{S_{f_R} + S_{f_Q}}{2} - S_0 \right) = 0 \\
 f_4 &\equiv x_R - x_Q - \frac{1}{2}(t_R - t_Q)[v_R - c_R + v_Q - c_Q] = 0
 \end{aligned} \right\} \quad (6.27)$$

Once the system (6.27) has been solved, one may ask the following question: What would be the solution if, instead of the conditions (y_P, v_P) at point P, we had, $(y_P + \Delta y_P, v_P)$, that is, when a change in water level is introduced. As this flow change perturbs the flow conditions of R, we can linearize to the proximity of y_P and expand a first order Taylor approximation series as:

$$y_R(y_P + \Delta y_P) = y_R(y_P) + \frac{\partial y_R}{\partial y_P} \Delta y_P + O(\Delta y_P^2) \quad (6.28)$$

To answer this question, we introduce the concept of hydraulic influence.

We define hydraulic influence of the conditions of water level and velocity at a point at a given time instant (as for example point P in Figure 6.11 or Figure 6.12) over the conditions at another point at another time instant (as for example point R in the same Figures) as the disturbance effect that is produced at point R caused by a small modification to the conditions of point P.

In order to answer the given question, suppose that all the variables of the system (6.27) depend implicitly on y_P , that is, $y_R(y_P)$, $v_R(y_P)$, $t_R(y_P)$ and $x_R(y_P)$ and also consider that the implicit function theorem is applied. Then by solving the 4x4 system:

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_R} & \frac{\partial f_1}{\partial y_R} & \frac{\partial f_1}{\partial v_R} & \frac{\partial f_1}{\partial t_R} \\ \frac{\partial f_2}{\partial x_R} & \frac{\partial f_2}{\partial y_R} & \frac{\partial f_2}{\partial v_R} & \frac{\partial f_2}{\partial t_R} \\ \frac{\partial f_3}{\partial x_R} & \frac{\partial f_3}{\partial y_R} & \frac{\partial f_3}{\partial v_R} & \frac{\partial f_3}{\partial t_R} \\ \frac{\partial f_4}{\partial x_R} & \frac{\partial f_4}{\partial y_R} & \frac{\partial f_4}{\partial v_R} & \frac{\partial f_4}{\partial t_R} \end{bmatrix} \begin{pmatrix} \frac{\partial x_R}{\partial y_P} \\ \frac{\partial y_R}{\partial y_P} \\ \frac{\partial v_R}{\partial y_P} \\ \frac{\partial t_R}{\partial y_P} \end{pmatrix} = - \begin{pmatrix} \frac{\partial f_1}{\partial y_P} \\ \frac{\partial f_2}{\partial y_P} \\ 0 \\ 0 \end{pmatrix} \quad (6.29)$$

The values $\frac{\partial x_R}{\partial y_P}$, $\frac{\partial y_R}{\partial y_P}$, $\frac{\partial v_R}{\partial y_P}$ and $\frac{\partial t_R}{\partial y_P}$ can be found.

It should be said that as (6.29) is the result of the application of the implicit function theorem on (6.27) it needs to fulfill the condition that the matrix in (6.29) can be inverted. That is to say, that its determinant is different from zero.

It can be shown that this condition is fulfilled and therefore the values of the implicit derivatives $\frac{\partial x_R}{\partial y_P}$, $\frac{\partial y_R}{\partial y_P}$, $\frac{\partial v_R}{\partial y_P}$ and $\frac{\partial t_R}{\partial y_P}$ always exist and are unique. The matrix structure in (6.29) can be formed for the remaining "disturbable" variables v_P , y_Q and v_Q . In general, and for future developments, we will refer to a generic variable ψ to denote variables of flow description such as depth, velocity, physical coefficients, gate position, pump flow etc. Then, the following general expression will be used:

$$\mathbf{M} \frac{\partial}{\partial \psi} \begin{pmatrix} x_R \\ y_R \\ v_R \\ t_R \end{pmatrix} = -\mathbf{N} \frac{\partial}{\partial \psi} \begin{pmatrix} y_P \\ v_P \\ y_Q \\ v_Q \end{pmatrix} \quad (6.30)$$

Where:

$$\mathbf{M} = \frac{\partial(f_1, f_2, f_3, f_4)}{\partial(x_P, y_R, v_R, t_R)} ; \quad \mathbf{N} = \frac{\partial(f_1, f_2, f_3, f_4)}{\partial(y_P, v_P, y_Q, v_Q)}$$

Note that (6.29) is equal to (6.30) when $\psi = y_P$ and thus it has a more general character. From the physical point of view (6.30) "moves" or "modifies" the influence of a variable ψ from the points P and Q to point R.

The way of calculating the influences shown in this section is closely linked to the numerical scheme of characteristic curves. However, usually this scheme is not exactly used because it gives the solution at a point R whose coordinates (x_R, t_R) are unknown a priori. These coordinates are part of the solution and normally it is more important to know the solution of the flow conditions at specific point and at specific time instants. To solve this problem, there are two possibilities: first solve and then interpolate, or first interpolate and then solve. The second option will be the one used in this thesis to solve the problem and is presented in the next section.

6.5.3. Applying to a structured grid

In Figure 6.13 you can see how by placing the characteristic curves net (Figure 6.13 a)) on top of a structured mesh (Figure 6.13 b)) a new scheme is obtained where the variables for points P and Q are interpolated (Figure 6.13 c)). In this way we can obtain the flow conditions for the fixed point R .

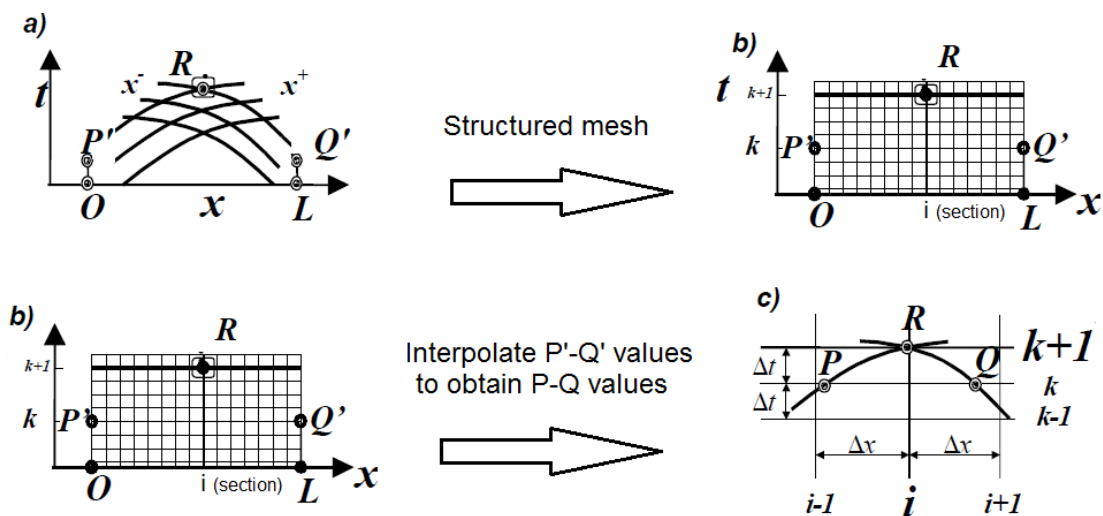


Figure 6.13: The steps for the interpolation onto a structured grid.

Obviously, the same system (6.27) is solved, but now with the new unknown's x_P , y_R , v_R and x_Q . A structured grid like this one creates a new nomenclature. Indeed, every variable will have a double index, where k refers to time and i to space. So y_{ik} and v_{ik} represent the values for water level and average velocity at the co-ordinates $x_i=i\Delta x$ and $t^k=k\Delta t$ where Δx and Δt are appropriately selected. As interpolation has to be also of second order (in order to be coherent with the numerical procedure used) we have

used the Lagrange factors (a way of representing quadratic splines). For a dummy variable z the result is (Figure 6.14):

$$z(x) = s^k(x, z_{i-1}^k, z_i^k, z_{i+1}^k) = \left(\frac{x-x_i}{\Delta x}\right)\left(\frac{x-x_{i-1}}{2\Delta x}\right)z_{i+1}^k + \left(\frac{x-x_{i-1}}{\Delta x}\right)\left(\frac{x-x_{i+1}}{-\Delta x}\right)z_i^k + \left(\frac{x-x_i}{-\Delta x}\right)\left(\frac{x-x_{i+1}}{-2\Delta x}\right)z_{i-1}^k \quad (6.31)$$

In this way the variables y_P , v_P , y_Q and v_Q become functions of x_P and x_Q , as follows:

$$\begin{aligned} y_P(x_P) &= s^k(x_P, y_{i-1}^k, y_i^k, y_{i+1}^k) ; & v_P(x_P) &= s^k(x_P, v_{i-1}^k, v_i^k, v_{i+1}^k) \\ y_Q(x_Q) &= s^k(x_Q, y_{i-1}^k, y_i^k, y_{i+1}^k) ; & v_Q(x_Q) &= s^k(x_Q, v_{i-1}^k, v_i^k, v_{i+1}^k) \end{aligned} \quad (6.32)$$

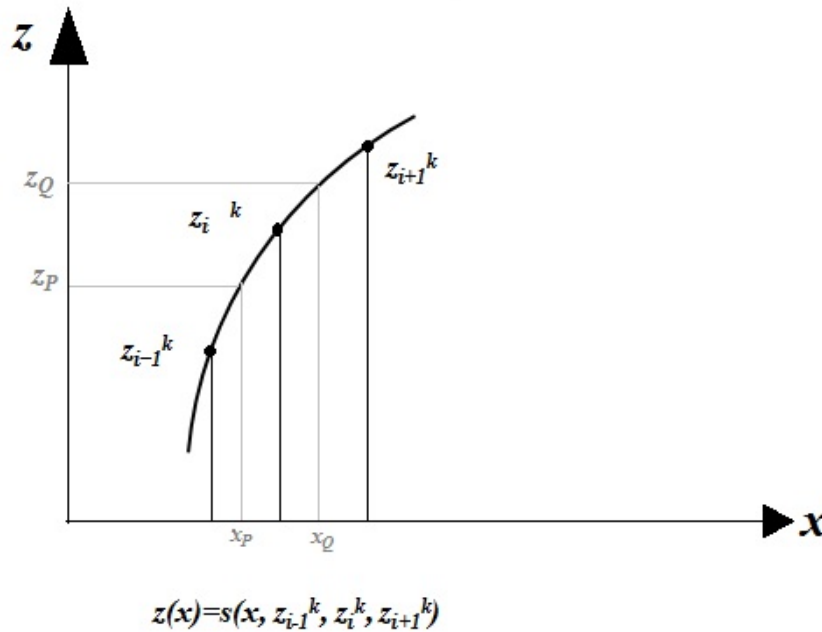


Figure 6.14: Interpolation functions.

From now on it is necessary to determine the system to be solved in order to find the evolution of the influence of a parameter through a structured grid as it has been done for finding (6.30) from the characteristic curves. Contrary to what might be expected, finding the "transfer" of the influences (or evolution of the influence in time or space) in an interpolation grid simplifies the problem. The reason for this is that the influence of the general parameter on the position and at time instant $\frac{\partial x_R}{\partial \psi} = \frac{\partial x_i}{\partial \psi'} \frac{\partial t_R}{\partial \psi} = \frac{\partial t^{k+1}}{\partial \psi}$ loses all meaning, because we want to find the solution to specific x and y axes (Figure 6.15).

If we introduce two different disturbances at time K , then by solving the four equations (6.27) with $x_P, y_i^{k+1}, v_i^{k+1}$ and x_Q as unknowns, two systems of characteristics are obtained x^+, x'^+ and x^-, x'^- , two solutions for point $R (y_i^{k+1}, v_i^{k+1})$ and $(y_i^{k+1}, v_i^{k+1})'$ and two sets of interpolated x axes x_P, x_Q and $x_{P'}, x_{Q'}$ are obtained, whereas the position of point R remains unaltered.

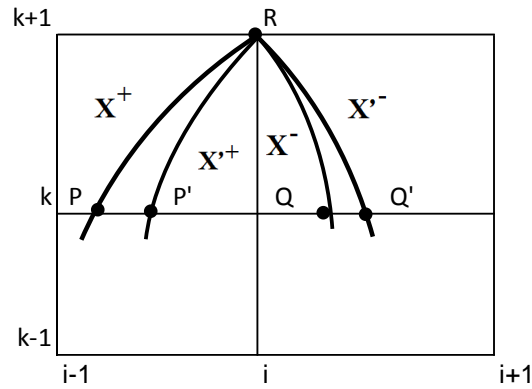


Figure 6.15: A pair of systems of characteristic curves passing through point R .

Applying once more the implicit function theorem to (6.27) with the assumption that $y_{i-1}^k, v_{i-1}^k, y_i^k, v_i^k, y_{i+1}^k, v_{i+1}^k, y_{i-1}^{k+1}$ and v_{i+1}^{k+1} now depend on a general parameter ψ , gives a system similar to (6.30):

$$\mathbf{M} \frac{\partial}{\partial \psi} \begin{pmatrix} x_P \\ y_i^{k+1} \\ v_i^{k+1} \\ x_Q \end{pmatrix} = -\mathbf{NS} \frac{\partial}{\partial \psi} \begin{pmatrix} y_{i-1}^k \\ v_{i-1}^k \\ y_i^k \\ v_i^k \\ y_{i+1}^k \\ v_{i+1}^k \end{pmatrix} \quad (6.33)$$

where

$$\mathbf{M} = \frac{\partial(f_1, f_2, f_3, f_4)}{\partial(x_P, y_i^{k+1}, v_i^{k+1}, x_Q)} ; \quad \mathbf{N} = \frac{\partial(f_1, f_2, f_3, f_4)}{\partial(y_P, v_P, y_Q, v_Q)}$$

$$\mathbf{S} = \frac{\partial(y_P, v_P, y_Q, v_Q)}{\partial(y_{i-1}^k, v_{i-1}^k, y_i^k, v_i^k, y_{i+1}^k, v_{i+1}^k)}$$

It should be said that of the four values obtained $\frac{\partial x_P}{\partial \psi}$, $\frac{\partial y_i^{k+1}}{\partial \psi}$, $\frac{\partial v_i^{k+1}}{\partial \psi}$ and $\frac{\partial x_Q}{\partial \psi}$ we are only interested in keeping $\frac{\partial y_i^{k+1}}{\partial \psi}$ and $\frac{\partial v_i^{k+1}}{\partial \psi}$ to find the solution to k+1, as the two other values never intervene in the right part of system (6.33).

About the differences between (6.33) and (6.30), we should underline the presence of the matrix $[S]_i^k$, the values of which can be obtained from (6.32). Note also that from the known values $\frac{\partial y_{i-1}^k}{\partial \psi}$, $\frac{\partial v_{i-1}^k}{\partial \psi}$, $\frac{\partial y_i^k}{\partial \psi}$, $\frac{\partial v_i^k}{\partial \psi}$, $\frac{\partial y_{i+1}^k}{\partial \psi}$ and $\frac{\partial v_{i+1}^k}{\partial \psi}$ at time k, the values of $\frac{\partial y_i^{k+1}}{\partial \psi}$ and $\frac{\partial v_i^{k+1}}{\partial \psi}$ at time k+1 are obtained, which shows the concept of influence and dependence domain.

For each point of the structured grid, a set of equations of the type (6.33) can be solved except for nodes that correspond to checkpoints or target points with pump, gates, and boundary conditions. These cases are analysed in the two following sections.

6.5.4. Control structures

There are many flow control structures in canals which allow flow modelling according to the specification of the watermaster. The individual study of each of these structures is impossible in this work, for this reason we will introduce the most common structures. A commonly found structure is a checkpoint with a sluice-gate, a lateral weir outlet, offtake orifice and a pumping, as seen in Figure 6.16. The interaction of this control structure with the flow can be described according to the principles of mass and energy conservation. These principles establish two mathematical relations between the flow conditions just upstream and downstream of the checkpoint, see Figure 6.16:

$$\left. \begin{aligned} S(y_e) \frac{dy_e}{dt} &= A(y_e)v_e - q_b - q_s(y_e) - A(y_s)v_s - q_{\text{offtake}}(y_e) \\ A(y_s)v_s &= k_c u \sqrt{y_e - y_s + d} \end{aligned} \right\} \quad (6.34)$$

- $S(y_e)$ is the horizontal surface of the reception area in the checkpoint.
- $A(y_e)v_e$ is the incoming flow to checkpoint, defined in terms of water level and velocity.
- $A(y_s)v_s$ is the outgoing flow to checkpoint which continues along the canal, described in terms of water level and velocity.
- $k_c = \sqrt{2g} * C_c * a_c$ where C_c is the discharge coefficient of the sluice-gate and a_c is the

sluice-gate width.

- d is the checkpoint drop, and u is the gate opening.
- q_b is the pumping offtake.
- $q_s(y_e) = C_s \cdot a_s \cdot (y_e - y_0)^{3/2}$ is the outgoing lateral flow through the weir where C_s is the discharge coefficient, a_s is the width of the lip and y_0 is the height of the lip measured from the bottom.
- $q_{\text{offtake}}(y_e) = C_d \cdot A_0 \cdot \sqrt{2g \cdot y_e}$ is the outgoing offtake orifice flow where C_d is the discharge coefficient, A_0 is the area of the offtake orifice.

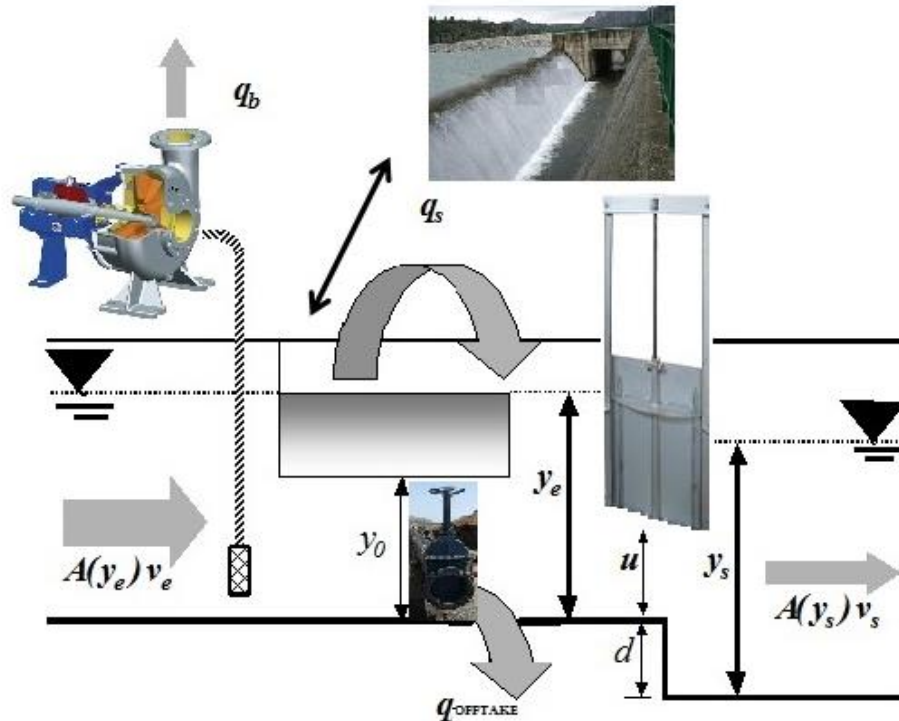


Figure 6.16: Diagram of a checkpoint with gate, lateral weir and pumping.

From the control point of view, it is necessary to note the important difference between several types of lateral outflow: the pumping is predetermined by the operator, whereas the lateral weir outlet and the offtake orifice, represented by the overflow, depend on the existing water level upstream from the control point and so it is controlled by this. The difference lies in the fact that, by pumping, the extracted flow can be produced with the only condition being having enough water, whereas with the

overflow the water obtained will depend on the water level maintained. So in this second case the control of the extracted flow will be much more difficult. Apart from the cases in which pumping is compulsory (due to level limitations) the overflow system is preferred because it has lower energy costs even though it is more difficult to control, although this latter case depends on the irrigation canal, and the importance of the demand deliveries.

6.5.5. The discretization of the checkpoint equations

The presence of checkpoints or control structures in the middle of a canal leads to the sub-division of this canal into canal pools, in a way that there is a canal pool between two checkpoints, and there is a checkpoint between two pools. So y_n^{k+1} represents the water level at node n in the section upstream of the control point at time $k+1$, that is, the incoming water level y_e . In the same way y_1^{k+1} is defined as the existing water level at the first node of the downstream pool from the checkpoint at the same time $k+1$, and y_s the outgoing water level at the control point (Figure 6.17). The same can be said for the velocities v_n^{k+1} and v_1^{k+1} .

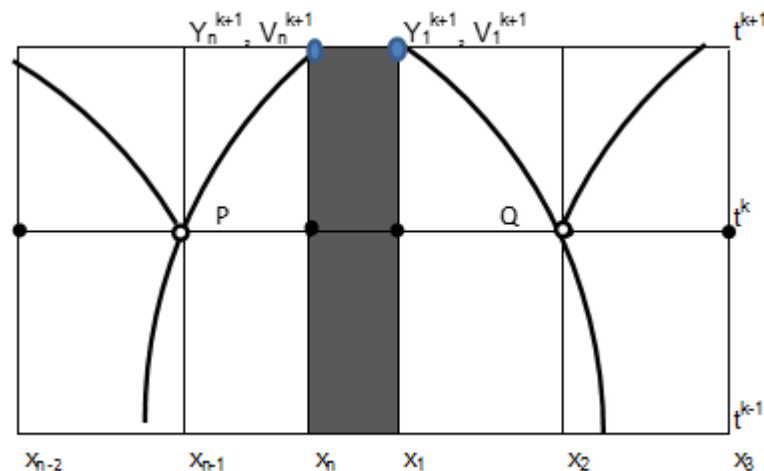


Figure 6.17: Graph with discretization of the control point equations.

If discretization is carried out with time and we rewrite the control point equations (6.34), join them with the characteristics of (6.27) and then change the nomenclature, we arrive at the following system of six equations:

$$\left. \begin{aligned}
 f_1 &\equiv x_n - x_p - \frac{1}{2} \Delta t \left[v_n^{k+1} + c_n^{k+1} + v_p + c_p \right] = 0 \\
 f_2 &\equiv \left(v_n^{k+1} - v_p \right) + \frac{g}{2} \frac{c_n^{k+1} + c_p}{c_n^{k+1} c_p} \left(y_n^{k+1} - y_p \right) - g \Delta t \left(\frac{S_{f_n}^{k+1} + S_{f_p}}{2} - S_0 \right) = 0 \\
 f_3 &\equiv \left(v_1^{k+1} - v_Q \right) - \frac{g}{2} \frac{c_1^{k+1} + c_Q}{c_1^{k+1} c_Q} \left(y_1^{k+1} - y_Q \right) - g \Delta t \left(\frac{S_{f_1}^{k+1} + S_{f_Q}}{2} - S_0 \right) = 0 \\
 f_4 &\equiv x_1^{k+1} - x_Q - \frac{1}{2} \Delta t \left[v_1^{k+1} - c_1^{k+1} + v_Q - c_Q \right] = 0 \\
 f_5 &\equiv A \left(y_n^{k+1} \right) v_n^{k+1} - q_b - q_s \left(y_n^{k+1} \right) - A \left(y_1^{k+1} \right) v_1^{k+1} - q_{\text{offtake}} \left(y_n^{k+1} \right) = 0 \\
 f_6 &\equiv A \left(y_1^{k+1} \right) v_1^{k+1} - k_c u \sqrt{y_n^{k+1} - y_1^{k+1} + d} = 0
 \end{aligned} \right\} \quad (6.35)$$

Where

- $\Delta t = t^{k+1} - t_p = t^{k+1} - t_Q$
- $y_P(x_P) = S(x_P, y_{n-2}^k, y_{n-1}^k, y_n^k)$
- $y_Q(x_Q) = S(x_Q, y_1^k, y_2^k, y_3^k)$
- $v_P(x_P) = S(x_P, v_{n-2}^k, v_{n-1}^k, v_n^k)$
- $v_Q(x_Q) = S(x_Q, v_1^k, v_2^k, v_3^k)$
- $c_n^{k+1} = c(y_n^{k+1})$
- $c_1^{k+1} = c(y_1^{k+1})$
- $S_{f_n}^{k+1} = S_f(y_n^{k+1}, v_n^{k+1})$
- $S_{f_1}^{k+1} = S_f(y_1^{k+1}, v_1^{k+1})$

Where the unknowns are $x_p, y_n^{k+1}, v_n^{k+1}, y_1^{k+1}, v_1^{k+1}$ and x_Q .

In order to continue with the calculation of the influences of a general parameter ψ , it is necessary to assume, in our case, that this parameter defines the pumping flow $q_b(\psi)$, and the unknowns variables $x_p, y_n^{k+1}, v_n^{k+1}, y_1^{k+1}, v_1^{k+1}$ and x_Q depend on ψ . So applying once more the implicit function theorem to (6.35), we obtain

$$\mathbf{M} \frac{\partial}{\partial \psi} \begin{pmatrix} x_p \\ y_1^{k+1} \\ v_1^{k+1} \\ y_n^{k+1} \\ v_n^{k+1} \\ x_Q \end{pmatrix} = \mathbf{NS} \frac{\partial}{\partial \psi} \begin{pmatrix} y_{n-2}^k \\ v_{n-2}^k \\ y_{n-1}^k \\ v_{n-1}^k \\ y_n^k \\ v_n^k \\ y_1^k \\ v_1^k \\ y_2^k \\ v_2^k \\ y_3^k \\ v_3^k \end{pmatrix} + \mathbf{L} \frac{\partial q_b}{\partial \psi} \quad (6.36)$$

Where

$$\mathbf{M} = \frac{\partial(f_1, f_2, f_3, f_4, f_5, f_6)}{\partial(x_p, y_n^{k+1}, v_n^{k+1}, y_1^{k+1}, v_1^{k+1}, x_Q)}$$

$$\mathbf{N} = -\frac{\partial(f_1, f_2, f_3, f_4, f_5, f_6)}{\partial(x_p, y_p, v_p, y_Q, v_Q, x_Q)}$$

$$\mathbf{L} = \left(0 \quad 0 \quad 0 \quad 0 \quad \frac{\partial f_5}{\partial q_b} \quad 0 \right)^T$$

$$\mathbf{S} = \frac{\partial(x_p, y_p, v_p, y_Q, v_Q, x_Q)}{\partial(x_p, y_{n-2}^k, v_{n-2}^k, y_{n-1}^k, v_{n-1}^k, y_n^k, v_n^k, y_1^k, v_1^k, y_2^k, v_2^k, y_3^k, v_3^k, x_Q)}$$

So far we have dealt with the influence of a generic variable ψ on the solution without entering into details about what the variable could represent. In (6.36) for the first time, it appears the pump flow $q_b(\psi)$ explicitly in the description. Despite the fact that the specific form of this function is still unknown, (6.36) shows that the influence of the parameter ψ on flow conditions at time $k+1$ is the sum of the indirect influence of the conditions at instant k and the direct influence at instant $k+1$ through the term $\left| \frac{\partial q_b}{\partial \psi} \right|_n^{k+1}$, which represents the variation in the pump flow when the parameter ψ changes.

6.5.6. Pump flow trajectories

In irrigation canal control, the period of time that the pump need to reach full-power operation can be considered negligible if compared to the rest of the time that the pump remains working.

This is the reason why an usual form of mathematically representing the pump flow trajectory is with a piece-wise function as seen in Figure 6.6. This representation allows the parametrization of the problem, which is the process of identifying the unknowns. By simply identifying the parameter ψ as the pump flow value $q_b(K)$ at the interval K between the time instants t^{K-1} and t^K , the process is finished. A pump flow trajectory is defined as the following vector of variables:

$$Q_b = [q_b(1) \ \cdots \ q_b(K) \ \cdots \ q_b(K_F)]^T \tag{6.37}$$

where K_F is the last interval in which the function of the pump flow trajectory is defined.

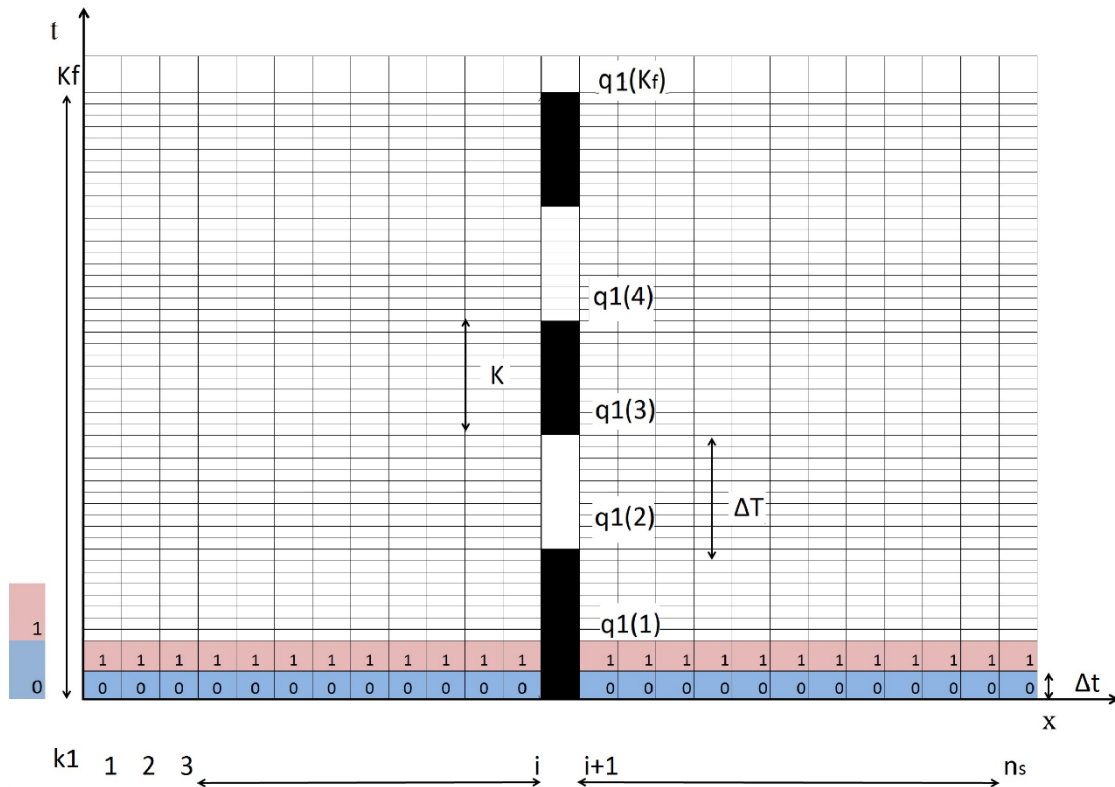


Figure 6.18: Graphical representation on two canal pools of the time discretization.

A pump flow change implies some very important consequences which are described here:

- In order to solve the problem it will be necessary to use a set of Saint-Venant equations, for example those described in (6.27). It will be necessary to limit the maximum time step Δt for stability reasons, as described in the stability condition of Courant-Friedrichs-Levy (CFL). This condition says that the maximum step allowed is the time it takes a wave to travel the incremental space Δx . If a pump parameter were defined for each discretization time instant, then the total number of variables would depend on CFL, that is, on the existing flow conditions for each instant. Given that, it would be very difficult, or even impossible, to solve the problem, so it is necessary to establish a time discretization of the pump flow trajectory independent of that of the simulation. To show this difference the superscript K (as a capital letter) will be used for the time discretization of the pump flow trajectory and k (as a small letter) will be used for the discretization of the simulation called time step, as shown in Figure 6.18. The time discretization K is usually named regulation period.

- The piece-wise form of function in Figure 6.6 implies $\frac{\partial q_b}{\partial \psi} = 1$ in (6.36), so we will talk about the pumping flow $q_b(K)$ during the K -interval of time (Figure 6.18).

- Taking into account the pump flow trajectory vector (6.37), the variables give a time nature, which allows us to state: $\frac{\partial y_i^k}{\partial q_b(K)} = 0$ if $t^k < T^K$.

6.5.7. The boundary conditions

To conclude the study of the influence that a pump flow has on the flow conditions at a point in the canal at a specific time instant, it is necessary to describe the boundary conditions as well as to find out how the influences evolve when they arrive at the edge of the canal and bounce. Crandall (1956) demonstrates another uniqueness theorem for second order systems like the one studied in this thesis, with the help of the graph in Figure 6.19, the theorem states the following: If the flow conditions at a characteristic intersection S are known and if only one variable is known from the two non-characteristic curves SP and SQ , then the solution uniqueness is guaranteed in the shaded area $SPRQ$.

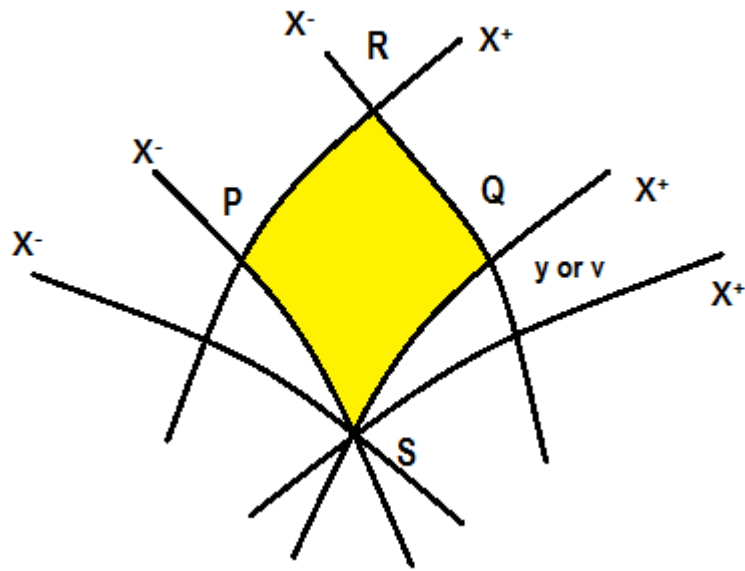


Figure 6.19: Graph showing the formulation of the second theory of uniqueness for second order sets of hyperbolic equations.

The joint application of this theory, and the one mentioned previously (Figure 6.11) entitles us to state that by establishing two conditions, one at each extreme of the solution interval, uniqueness can be guaranteed in the shaded areas of Figure 6.20. That is, if flow conditions at points P and Q on two characteristic curves and any condition of the y axis at both extremes are known, then the solution for points R and R' can be obtained.



Figure 6.20: Graph representing the application of the two uniqueness theorem to the boundaries.

It should be said that in a subcritical regime (the usual working regime in irrigation canals, and the only considered in this thesis) a condition for each canal boundary must be established (as stated in the previous paragraph) because the slopes of the characteristic curves x^+ and x^- have opposite signs since the velocity of the wave is higher than the velocity of fluid media. That is, from (6.22) we have:

$$\frac{dx^+}{dt} \equiv v + c(y) > 0 \quad ; \quad \frac{dx^-}{dt} \equiv v - c(y) < 0$$

There are a large number of boundary conditions that can be set at the extreme ends of a canal. Continuing with the nomenclature of the structured grid (Figure 6.18), the boundary conditions are established as two expressions of the following generic types:

$$\begin{aligned} f_7 &\equiv (y_1^{k+1} \quad v_1^{k+1}) = 0 \quad \text{upstream boundary condition} \\ f_8 &\equiv (y_{n_s}^{k+1} \quad v_{n_s}^{k+1}) = 0 \quad \text{downstream boundary condition} \end{aligned} \quad (6.38)$$

Taking the third and fourth equations of (6.35) and the first from (6.38) gives a new system that needs to be solved in order to find the flow conditions at the upstream boundary:

$$\left. \begin{aligned} f_7 &\equiv (y_1^{k+1} \quad v_1^{k+1}) = 0 \\ f_3 &\equiv (v_1^{k+1} - v_Q) - \frac{g}{2} \frac{c_1^{k+1} + c_Q}{c_1^{k+1} c_Q} (y_1^{k+1} - y_Q) - g \Delta t \left(\frac{S_{f_1}^{k+1} + S_{f_Q}}{2} - S_0 \right) = 0 \\ f_4 &\equiv x_1^{k+1} - x_Q - \frac{1}{2} \Delta t [v_1^{k+1} - c_1^{k+1} + v_Q - c_Q] = 0 \end{aligned} \right\} \quad (6.39)$$

By doing the same we can find the set of equations corresponding to the downstream ends if we take the first and the second equations of (6.35) and the second from (6.38), which gives:

$$\left. \begin{aligned} f_1 &\equiv x_{n_s} - x_P - \frac{1}{2} \Delta t [v_{n_s}^{k+1} + c_{n_s}^{k+1} + v_P + c_P] = 0 \\ f_2 &\equiv (v_{n_s}^{k+1} - v_P) + \frac{g}{2} \frac{c_{n_s}^{k+1} + c_P}{c_{n_s}^{k+1} c_P} (y_{n_s}^{k+1} - y_P) - g \Delta t \left(\frac{S_{f_{n_s}}^{k+1} + S_{f_P}}{2} - S_0 \right) = 0 \\ f_8 &\equiv (y_{n_s}^{k+1} \quad v_{n_s}^{k+1}) = 0 \end{aligned} \right\} \quad (6.40)$$

Applying once more the hypothesis that all variables are dependent on a pump flow $q_b(K)$ as we did in order to find (6.33) and (6.36) for the upstream point we can write:

$$\mathbf{M} \frac{\partial}{\partial q_b(K)} \begin{pmatrix} x_Q \\ y_1^{k+1} \\ v_1^{k+1} \end{pmatrix} = \mathbf{NS} \frac{\partial}{\partial q_b(K)} \begin{pmatrix} y_1^k \\ v_1^k \\ y_2^k \\ v_2^k \\ y_3^k \\ v_3^k \end{pmatrix} + \mathbf{L} \quad (6.41)$$

Where

$$\mathbf{M} = \frac{\partial(f_7, f_3, f_4)}{\partial(y_1^{k+1}, v_1^{k+1}, x_Q)}$$

$$\mathbf{N} = -\frac{\partial(f_7, f_3, f_4)}{\partial(y_Q, v_Q, x_Q)}$$

$$\mathbf{L} = \left(\frac{\partial f_7}{\partial q_b(K)} \quad 0 \quad 0 \right)^T$$

$$\mathbf{S} = \frac{\partial(y_Q, v_Q, x_Q)}{\partial(y_1^k, v_1^k, y_2^k, v_2^k, y_3^k, v_3^k)}$$

and for the downstream end:

$$\mathbf{M} \frac{\partial}{\partial q_b(K)} \begin{pmatrix} y_1^{k+1} \\ v_1^{k+1} \\ x_P \end{pmatrix} = \mathbf{NS} \frac{\partial}{\partial q_b(K)} \begin{pmatrix} y_{n_s-2}^k \\ v_{n_s-2}^k \\ y_{n_s-1}^k \\ v_{n_s-1}^k \\ y_{n_s}^k \\ v_{n_s}^k \end{pmatrix} + \mathbf{L} \quad (6.42)$$

Where

$$\mathbf{M} = \frac{\partial(f_1, f_2, f_8)}{\partial(y_{n_s}^{k+1}, v_{n_s}^{k+1}, x_P)} ; \quad \mathbf{N} = -\frac{\partial(f_1, f_2, f_8)}{\partial(y_P, v_P, x_P)}$$

$$\mathbf{L} = \left(0 \quad 0 \quad \frac{\partial f_8}{\partial q_b(K)} \right)^T ; \quad \mathbf{S} = \frac{\partial(y_P, v_P, x_P)}{\partial(y_{n_s-2}^k, v_{n_s-2}^k, y_{n_s-1}^k, v_{n_s-1}^k, y_{n_s}^k, v_{n_s}^k)}$$

6.5.8. Hydraulic influence of a pump flow trajectory parameter on the state vector

The structured grid (Figure 6.18) shows a time-space discretization of a canal with two pools, Pool I subdivided into $i-1$ cells and Pool II subdivided into n_s-1 cells. The grid also shows the flow solution at time k and at the subsequent time $k+1$ (every Δt) for all the n_s sections. The algorithm allows us to find sequentially the solution for all the computational nodes at $k+1$, provided the type of each of these nodes is identified, the algorithm can solve the corresponding set of equations. Table 6.1 summarizes the whole procedure.

	Pool	Node	Simulation
Upstream boundary	I	1	(6.41)
Computational nodes	I	2, . . . ,i-1	(6.33)
Checkpoints	I and II	i ,i+1	(6.36)
Computational nodes	II	i+2, . . . ,n _s -1	(6.33)
Downstream boundary	II	n _s	(6.42)

Table 6.1: summarize of the whole procedure

So far we have only discussed about the solution of the equations for simulation purposes. However our final goal is the study of the time-space evolution of the influence of any variable in the trajectory (6.37). For each time instant, the new value of the influence of any pump flow $q_b(K)$ is calculated at each computational node using some of the previously mentioned systems of equations: (6.33), (6.36), (6.41) or (6.42).

Adequately compiled into a single set that can be represented in the following manner:

$$\frac{\partial \mathbf{x}^{k+1}}{\partial q_b(K)} = \mathbf{A} \frac{\partial \mathbf{x}^k}{\partial q_b(K)} + \mathbf{b} \left[\mathbf{x}^{k+1}, q_b(K) \right] \quad (6.43)$$

where $A = A[x^{k+1}, x^k, q_b(K)]$ is a $(2n_s) \times (2n_s)$ square matrix where n_s is the total number of cross sections of the simulation according to the nomenclature showed in Figure 6.18, $b[x^{k+1}, q_b(K)]$ is the direct influence vector of $2n_s$ components and x is the state vector of dimension $2n_s$, which at instant k and $k+1$ is defined as:

$$\mathbf{x}^k = \left(y_1^k \quad v_1^k \quad \dots \quad y_i^k \quad v_i^k \quad \dots \quad y_{n_s}^k \quad v_{n_s}^k \right)^T \quad (6.44)$$

$$\mathbf{x}^{k+1} = \left(y_1^{k+1} \quad v_1^{k+1} \quad \dots \quad y_i^{k+1} \quad v_i^{k+1} \quad \dots \quad y_{n_s}^{k+1} \quad v_{n_s}^{k+1} \right)^T$$

The set of equations (6.43) shows that the influence of the pump parameter $q_b(K)$ on the state vector at time $k+1$ is equal to the direct influence of $q_b(K)$ in instant $k+1$ plus the updating of the influence in time k .

The Figure 6.21 shows the influence of a pump flow change during a several regulation period ($T^k - T^{k+1}$), as the disturbance extends upstream and the downstream in the canal. Grey colours represent points that the disturbance has no influence and blue colour tones represent points that the disturbance decreases or increases water levels due to the pump flow. The plot can be also seen as the effect of a perturbation made by $q_b(K)$ over the predictive vector. If we look at the diagram in Figure 6.21, we need to make the following considerations:

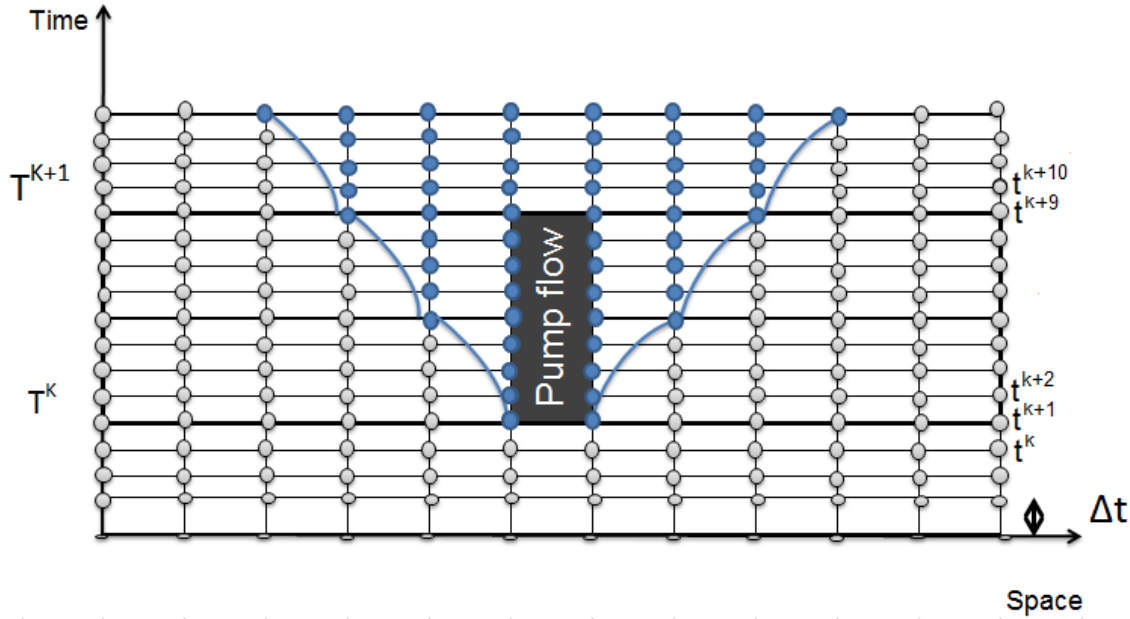


Figure 6.21: The evolution of the influence of the pump flow through time and space. The grey dots correspond to the points where there is no influence of the pump flow: $\frac{\partial x_i^k}{\partial q_b(K)} = 0$ and the blue dots where there is influence of the pump flow: $\frac{\partial x_i^k}{\partial q_b(K)} \neq 0$.

- a) In case that we study the influence of $q_b(K)$ at time step k , as $t^k < T^k$, then $\frac{\partial x_i^k}{\partial q_b(K)} = 0$ there is no need to set the influence of the pump flow because the influence domain of a parameter backwards in time is null.
- b) In case that we study the influence of $q_b(K)$ at time step $k+1$, as $t^{k+1} = T^k$, the set of equations (6.43) becomes:

$$\frac{\partial x^{k+1}}{\partial q_b(K)} = [A] \frac{\partial x^k}{\partial q_b(K)} + b[x^{k+1}, q_b(K)] = b[x^{k+1}, q_b(K)] \quad (6.45)$$

where $[A] = A[x^{k+1}, x^k, q_b(K)]$ represents the updating of the influence from x^k to x^{k+1} , and $b[x^{k+1}, q_b(K)]$ is the direct influence vector and x is the state vector.

This last term is useful to find the initial value of the influence of the parameter $q_b(K)$ on the flow conditions x^{k+1} , that is to say, the direct influence.

- c) In case that we study the influence of $q_b(K)$ at time step $k+2$. The complete set of equations (6.43) is used when $T^k < t^{k+2} \leq T^{k+1}$:

$$\frac{\partial x^{k+2}}{\partial q(K)} = A[x^{k+1}, q_b(K)] \frac{\partial x^{k+1}}{\partial q(K)} + b[x^{k+2}, q_b(K)] \quad (6.46)$$

d) Finally, when $t^{k+10} > T^{k+1}$ the equation (6.43) lose the direct influence of the parameter and becomes:

$$\frac{\partial x^{k+10}}{\partial q(K)} = A[x^{k+9}, q_b(K)] \frac{\partial x^{k+9}}{\partial q(K)} \quad (6.47)$$

After determining the values of the state vector by applying the corresponding sets of equations, the influence domain of the parameter $q_b(K)$ can be described and quantified, and it's influence on the system is showed by the blue dots of the Figure 6.21.

The hydraulic influence matrix is composed by the influence of the pump flow over the canal state lumped in a matrix, which it is composed by all the terms:

$$[HIM(Q_{b_j})] = \left[\frac{\partial x_{k_1}^{k_F}}{\partial q_{b_j}(1)} \dots \frac{\partial x_{k_1}^{k_F}}{\partial q_{b_j}(K)} \dots \frac{\partial x_{k_1}^{k_F}}{\partial q_{b_j}(K_F)} \right]. \quad (6.48)$$

A forecast of the perturbed flow can be calculated by the product:

$$\Delta X_1^{k_F} = \left(\frac{\partial X_1^{k_F}}{\partial q_b(K)} \right) \Delta q_b(K) \quad (6.49)$$

If we want to know a forecast of the predictive vector when the pump flow trajectories are perturbed a little, we only have to consider:

$$X_1^{k_F}(Q_b^* + \Delta Q_b) = X_1^{k_F}(Q_b^*) + \Delta X_1^{k_F} \quad (6.50)$$

where Q_b is the vector of the compilation of all the pump flow trajectories:

$$Q_b = \left(q_{b_1}(1) \dots q_{b_i}(1) \dots q_{b_{n_p}}(1) \dots q_{b_1}(K_F) \dots q_{b_i}(K_F) \dots q_{b_{n_p}}(K_F) \right) \quad (6.51)$$

where $\Delta Q_b = (0, \dots, \Delta q_b(K), \dots)$ is the disturbance of the pump flow trajectories, Q_b^* is the reference pump flow trajectories, and the $X_1^{k_F}(Q_b^* + \Delta Q_b)$ is the simulated state vector when the pump flow trajectories are disturbed a little.

The $HIM(Q_b)$ allows to analyse the effect that a flow change has on the system (that we could call analysis of the "influence domain") and also which pump station has an effect on a cross-section of the canal at a certain instant (that we could call analysis of the "dependence domain" of the point).

The $HIM(Q_b)$ can be considered as a linearization of the Saint-Venant model as the model used by Malaterre (1994), but the $HIM(Q_b)$ is a linearization model around of an unsteady state solution. The HIM is a helpful tool to calculate the state vector when the pump flow trajectories are disturbed a little:

$$X_1^{k_F}(Q_q^* + \Delta Q_q) - X_1^{k_F}(Q_q^*) = HIM(Q_q^*)\Delta Q_q + O(\Delta Q_q^2) \quad (6.52)$$

Neglecting the second order and upper terms, (6.52) can be rewritten in the form:

$$\Delta X = HIM(Q_q^*)\Delta Q_q \quad (6.53)$$

where ΔQ_b = the perturbed pump flow vector; and ΔX = the predicted error from the state vector.

We way include all the state vectors (6.44) for each k-instant on the simulated past time horizon into the following single vector named predicted vector:

$$\mathbf{X}_1^{k_F}(Q) = \left[\mathbf{x}^1(Q)^T \quad \mathbf{x}^2(Q)^T \quad \dots \quad \mathbf{x}^{k_F-1}(Q)^T \quad \mathbf{x}^{k_F}(Q)^T \right]^T \quad (6.54)$$

Where k_F = final instant of the past time horizon of a period of time. We can now compile into only one vector, in a similar way to before, all the values of the influence of any pump flow trajectory $q_{b_j}(K)$ on the state vector. This new vector we will call the vector of influence of the pump flow on the predictive vector:

$$\frac{\partial \mathbf{X}_1^{k_F}(Q)}{\partial q_{b_j}(K)} = \left[\frac{\partial \mathbf{x}^1(Q)^T}{\partial q_{b_j}(K)} \quad \frac{\partial \mathbf{x}^2(Q)^T}{\partial q_{b_j}(K)} \quad \dots \quad \frac{\partial \mathbf{x}^{k_F-1}(Q)^T}{\partial q_{b_j}(K)} \quad \frac{\partial \mathbf{x}^{k_F}(Q)^T}{\partial q_{b_j}(K)} \right]^T \quad (6.55)$$

6.5.9. Compilation of the Hydraulic influence matrix

6.5.9.1. Definition

Once we have introduced the hydraulic influence of a pump flow on the canal state, it is possible to compile all these elements in a matrix which we call "the hydraulic influence matrix". If we define the hydraulic influence on the state vector (6.54) of a certain pump station "j" (6.37) during a past time horizon (γ) in each cross-section, the following hydraulic influence matrix, denoted by $[I_M(q_b)]_j$, is obtained:

$$[I_M(q_b)]_j = \left[\begin{array}{ccc} \frac{\partial X_{k_1}^{k_F}}{\partial q_{b_j}(1)} & \cdots & \frac{\partial X_{k_1}^{k_F}}{\partial q_{b_j}(K)} & \cdots & \frac{\partial X_{k_1}^{k_F}}{\partial q_{b_j}(\gamma)} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial x^{k_i}}{\partial q_{b_j}(1)} & \cdots & \frac{\partial x^{k_i}}{\partial q_{b_j}(K)} & \cdots & \frac{\partial x^{k_i}}{\partial q_{b_j}(\gamma)} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial x^{k_F}}{\partial q_{b_j}(1)} & \cdots & \frac{\partial x^{k_F}}{\partial q_{b_j}(K)} & \cdots & \frac{\partial x^{k_F}}{\partial q_{b_j}(\gamma)} \end{array} \right] = \quad (6.56)$$

Where $2 \times n_s$ is the dimension of the state vector at time step k equal to the number of cross sections plus the number of parameters (water level and velocity), j is "jth" pump station in the canal, and γ is the final regulation period.

A feature of this matrix is the number of columns and the rows. There are more rows than columns because every term $\frac{\partial x^k}{\partial q_{b_j}(K)}$ has $2 \times n_s$ rows, and the temporal discretization of the simulation Δt^k is smallest than the regulation period ΔT^K .

In the case that the matrix only defines the hydraulic influence of the pump flow on the water level (not considering the velocity), the number of elements by each term $\frac{\partial x^k}{\partial q_{b_j}(K)}$ is reduced to n_s , and if the number of cross-sections is also reduced to the number of

checkpoints so $n_s = n_c$. In case that the temporal discretization of the simulation introduced into the matrix is equal to the number of regulation period and the number of checkpoints is equal to the number of pumps, the HIM matrix is a square matrix.

Another one feature of the HIM is that there are some elements with a zero value, because there is not any influence of the pumps on certain points at a certain time step.

When there are more than one pump station in the canal, the matrix has to compile the pump stations all together. In that case, the HIM is compiled as:

$$[I_M(Q_b)]_X = \left[\left[\frac{\partial X_1^{k_F}(q_b)}{\partial q_{b_1}(1)} \dots \frac{\partial X_1^{k_F}(q_b)}{\partial q_{b_{n_p}}(1)} \right] \dots \left[\frac{\partial X_1^{k_F}(q_b)}{\partial q_{b_1}(\gamma)} \dots \frac{\partial X_1^{k_F}(q_b)}{\partial q_{b_{n_p}}(\gamma)} \right] \right] \quad (6.57)$$

where sub index X is used to denote the hydraulic influence matrix on the state vector, n_p is the total number of the pump stations.

6.5.9.2. The discrete observer

In most cases we only need to know some values of the state vector in specific points of the canal in particular time steps. It is possible to see an example of two canal pools with one pump in the Figure 6.18, in this spatial discretization the number of sections is n_s and the number of time steps (remember that the duration of the time step depend on the Courant condition in implicit schemes) is $n_T = k_F - k_I$. The discretization of the pump flow trajectories depends on the number of pumps $n_p = 1$ and the regulation periods, in this case $\gamma = 4$. As a result the total number of parameters becomes $n_Q = \gamma \times n_p = 4$.

So the dimensions of the vectors and matrices shown in this spatial discretization are the following:

The dimension of the simulation vector $X_{k_{I+1}}^{k_F}$: $n_x = 2 \times n_T \times n_s$

The dimension of the pump flow vector Q_b : $n_Q = \gamma \times n_p = 4$

The dimension of the hydraulic influence matrix on the simulation vector $[I_M(Q_b)]_x$: $n_x \times n_Q = 2 \times n_T \times n_s \times n_Q$

The result of the simulation at checkpoints $Y_{k_{l+1}}^{k_F}(Q_b)$: $n_Y = n_T \times n_s$

- The dimension of the hydraulic influence matrix on the discrete observer set $[I_M(Q_b)]_Y$: $n_Y \times n_Q$

In order to express this information mathematically, as we introduced before, a new matrix needs to be introduced called "discrete observer matrix", denoted as [C] in the control literature. This matrix is made up exclusively of "zeros" and "ones", and it represents respectively in the next equation (6.58), the state vector at certain time steps and at certain points of the canal, and the influence matrix of the trajectories at these points.

$$\begin{aligned} Y_1^{k_F}(Q_b) &= [C]X_1^{k_F}(Q_b) \\ [I_M(Q_b)]_Y &= [C][I_M(Q_b)]_x \end{aligned} \quad (6.58)$$

In the case, that $n_T = \gamma, n_s = \text{number of checkpoints} = n_c$ and the only one hydrodynamic variable represented is the water level. The HIM is a square matrix, conclusion reached anteriorly.

6.5.9.3. Features of the hydraulic influence matrix

One of the most important characteristic of the hydraulic influence matrix ($[I_M(Q_b)]_Y$) must be that this matrix is not a non-singular matrix.. This requisite is justified by the fact that in the optimisation process, the matrix has to be inverted through its pseudo-inversion (6.59) (see algorithms from sections 6.3.2). In other words, if we want to know what pump flow change (ΔQ_b) has to be introduced into the pump flow trajectory Q_b to obtain a change of the state vector (ΔY), it is necessary to solve the pseudo-inversion, as we introduce in Appendix 2.

$$\Delta Q_b = \left\{ [I_M(Q_b)]_Y^T [I_M(Q_b)]_Y \right\}^{-1} [I_M(Q_b)]_Y^T \Delta Y \quad (6.59)$$

where $\left\{ [I_M(Q_b)]_Y^T [I_M(Q_b)]_Y \right\}^{-1}$ is the so-called pseudo-inverse matrix, $\Delta Y = Y_1^{k_F}(Q_b + \Delta Q_b) - Y_{k_1}^{k_F}(Q_b)$ is a deviation on the water level. So if the hydraulic influence matrix is not singular, then the pseudo-inversion will be able to be inverted and the set of equations (6.59) can be solved.

The problem is when a bad definition of the discrete observer matrix makes that the hydraulic influence matrix is become a singular matrix, for instance when a pump

station "j" ($q_{jb}(K_F-1)$) extracted flow at the regulation period K_F-1 , as we show at the Figure 6.22, this has not influence on section "i" of the canal from time step 1 to K_F , so a whole column of $[I_M(Q)]_Y$ is full of "zeros" (6.56).

$$\frac{\partial Y_1^{K_F}(Q_b)}{\partial q_{bj}(K)} = 0 \quad (6.60)$$

As we can look at the Figure 6.22, the parameter $q_{bj}(K_F)$ has no influence on the levels $Y_i(1), \dots, Y_i(K_F-2), \dots, Y_i(K_F-1), \dots, Y_i(K_F)$. As a final conclusion, the study of the hydraulic influence matrix could allow the establishment of the control parameters, such as the determination of the testing period.

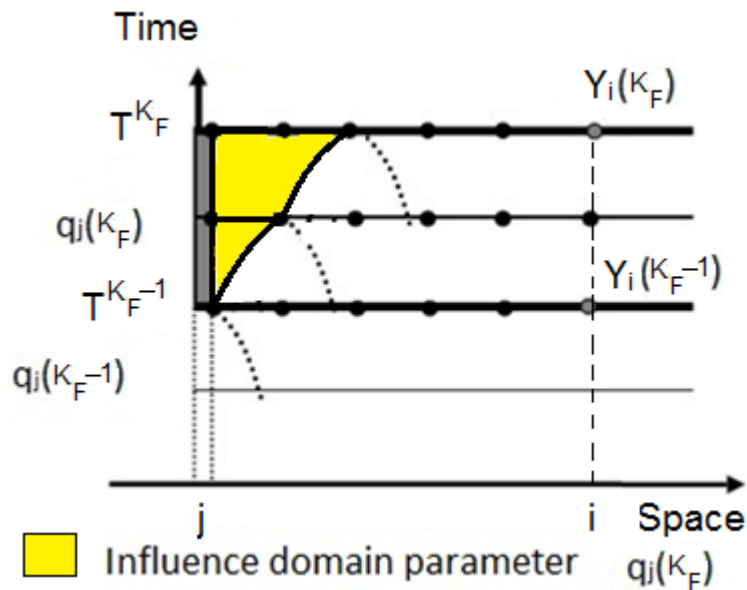


Figure 6.22: An example of a pump parameter which has no influence on any point.

6.6. APPENDIX II

Unconstrained optimized problem (Levenberg-Marquardt)

As shown in the "perturbation theory" by Brogan (1985) applied in the predictive output vector, we can introduce Taylor series modifying this vector around a pump flow reference trajectory by means:

$$Y = Y(Q_b^* + \Delta Q_b) = Y^*(Q_b^*) + \left[\frac{\partial Y}{\partial Q_b} \right]_{Q_b^*} \cdot \Delta Q_b + \frac{1}{2} \left[\frac{\partial^2 Y}{\partial Q_b^2} \right]_{Q_b^*} \cdot \Delta Q_b^2 + \dots \quad (6.61)$$

Assuming that the disturbance caused is sufficiently small, we can neglect the terms after of the first term (6.62) obtaining:

$$\Delta Q_b = Y(Q_b^* + \Delta Q_b) - Y^* = \left[\frac{\partial Y}{\partial Q_b} \right]_{Q_b^*} \Delta Q_b \quad (6.62)$$

On the other hand, it is possible to use the hydraulic influence matrix (HIM), defined by Soler (2003).

$$[J^*] = \left[\frac{\partial Y}{\partial Q_b} \right]_{Q_b^*} \quad (6.63)$$

Taking account all of these terms, we can write:

$$\Delta Q_b = [J^{*T} \cdot J^*]^{-1} \cdot [J^{*T}] \cdot \Delta Y \quad (6.64)$$

We can set the ΔQ pump flow changes to reach the desired water level in the targets as these have been modified for a perturbation ΔY . The equation system can be solved by Jacobi method, Cholesky method, singular value decomposition method or LU decomposition. There are an extensive bibliography about these methods, as Numerical Recipes in FORTRAN 77 (1992).

The problem is to solve the equation (6.64) at each time step. We can define the equation on a compact form:

$$[G] = [J^{*T}] \cdot [J^*] \quad (6.65)$$

$$b' = [J^{*T}] \cdot \Delta Y \quad (6.66)$$

So we can replace the equation (6.65) and equation (6.66) in equation (6.64):

$$b' = [G] \cdot \Delta Q_b \quad (6.67)$$

Although the system matrix is positive definite by definition, is not well conditioned (Soler, 2003). This implies that solving the system, numerical errors can be very important. To avoid this problem we add a positive term to the diagonal of the system matrix, this parameter is called Marquardt coefficient (V). It has been used several times, in all problems with the Hessian matrix. It exists a lot of theory about it, for instance in Fletcher (1987).

The system of equations can be written then as:

$$b' = [G + vI] \cdot \Delta Q_b \quad (6.68)$$

Where [I] is the identity matrix of the same dimensions as [G]. To solve the system (6.68) can use different methods. In our case we use a process optimization method based on the Levenberg-Marquardt method.

Supposing we have a diagonal matrix [Λ] with the eigenvalues of the matrix [G], and a matrix orthonormal ([V]: $[V \times V^T] = [I]$) defined by the eigenvectors associated with each eigenvalue of [G]. Multiplying both sides of equality (6.68) for the matrix $[V^T]$, we obtain:

$$[V^T \cdot G + V^T \cdot vI][V \cdot V^T] \Delta Q_b = [V^T] \cdot b' \quad (6.69)$$

Knowing that the [V] matrix is orthonormal, we can write (6.69) as follow:

$$\left[V^T \cdot G \cdot V + V^T \cdot \nu I \cdot V \right] \left[V^T \right] \Delta Q_b = \left[V^T \right] \cdot b' \quad (6.70)$$

And according to the property $[\Lambda] = [V^T \times G \times V]$, it can be transformed (6.70) as follows:

$$\left[\Lambda + \nu I \right] \cdot \left[V^T \right] \cdot \Delta Q_b = \left[V^T \right] \cdot b' \quad (6.71)$$

Furthermore, the fact of using a matrix as orthonormal matrix $[V]$ allows us to ensure that:

$$\left\| \left[\Lambda + \nu I \right] \cdot \left[V^T \right] \cdot \Delta Q_b \right\| = \left\| \left[V^T \right] \cdot b' \right\| \quad (6.72)$$

To simplify (6.71) we can define the vectors Z and B as follow:

$$Z = \left[V^T \right] \cdot \Delta Q_b \quad (6.73)$$

$$B = \left[V^T \right] \cdot b' \quad (6.74)$$

So Z is an unknown quantity, and (6.71) can be transformed into:

$$\left[\Lambda + \nu I \right] Z = B \quad (6.75)$$

From equation (6.75) the Z components can be separated easily because the new system matrix is now diagonal. Each of the Z components can be calculated as follow:

$$Z(i) = \frac{B(i)}{\Lambda(i, i) + \nu[I]} \quad (6.76)$$

Although the matrix $[G]$ is positive definite, it is ill conditioned so it is possible that some eigenvalue of $[G]$ are negative, although very close to absolute zero. To prevent this problem is introduced the Marquardt coefficient whose value is necessarily greater than the absolute value of the smallest eigenvalue of $[G]$, so that $(\Lambda + \nu[I])$ is always a positive number. Depending on the value you chose for the coefficient, the solution will be different. In that sense, we solve the system of equation in a iteration procedure,

the algorithm reduces the Marquardt coefficient in each iteration in order to reduce the influence of the coefficient in the final solution.

Solving the equation (6.76), we obtain the solution (the changes in extracted flow vector).

$$\Delta Q_b = [V] \cdot Z \quad (6.77)$$

So an extracted flow vector is calculated at time step K for a past time horizon ($Q_0^K = Q_0^K + \Delta Q_0^K$). The optimization problem must be solved again in the next time instant K+1, so we could obtain a new extracted flow vector at time step at time step K+1 for a past time horizon ($Q_0^{K+1} = Q_0^{K+1} + \Delta Q_0^{K+1}$).

Chapter 7

CSI performance: Numerical examples

In this chapter we test the CSI algorithm in several numerical examples to establish the disturbances introduced in the system and the hydrodynamic state of a canal during a past time horizon.

The CSI is an useful algorithm, as we have introduced in the last chapter, and can be used in different ways. For instance, if the demand deliveries have been increased because a farmer took a greater flow by the orifice offtake that it was scheduled, CSI can obtain the unknown flow withdrawn by the farmer. If a farmer pumps water from an unknown point in the canal, CSI can also obtain the approximate location of this point and the volume of water withdrawn by the farmer.

CSI estimates the hydrodynamic state of a canal during a past time horizon and the current canal state which is very useful for a feedback controller which needs information about the canal state in real time.

In this chapter, we introduce some numerical examples considering an irrigation canal in subcritical flow regime during a past time horizon, in which the unknown flow changes are associated with changes in demand deliveries at the control structures. The objective is to define with the aid of the algorithm, the past and current hydrodynamic state of the canal and the unknown flow diversions.

To introduce this chapter we have to define the problem statement specifying, the geometry of the canal, the initial and boundary conditions, and the measured water levels at the checkpoints. We analyze the results obtained in each example, that is, the extracted flow water in space and time and the hydrodynamic canal state

7.1. Introduction

The first objective of CSI is estimate the offtake changes and the second objective is obtain the hydrodynamic canal state from present moment to a past time by two reasons. On one side, we want to quantify the wrong deliveries and on the other side define the past and the current canal state to send to the feedback algorithm. CSI calculates the extracted flow (through lateral weirs, pumps, orifice offtake,...) from the water level measurements at checkpoints, the gate trajectories and the scheduled demands at the orifice offtakes during a past time horizon in these examples. The purpose is difficult to achieve because the farmer can extract flow in every pump station or orifice offtake in the canal at any time.

We proposed several scenarios to test CSI algorithm in a canal which has two pools separated by sluice gates (Figure 7.1). The flow is controlled by a gate downstream from the reservoir. Water is delivered through gravity outlets at the downstream end of each pool, where the check-points are located. There are pumping stations at the end of each pool which can introduce disturbances on the system in space and time.

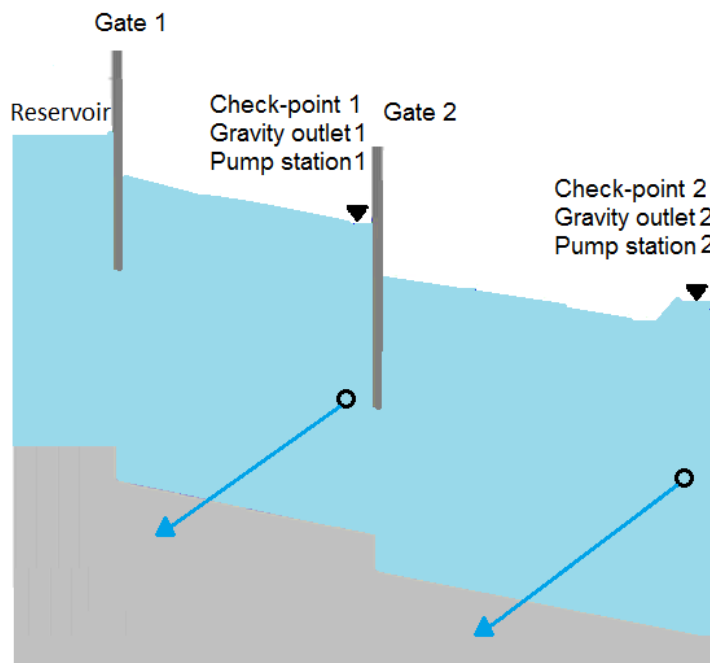


Figure 7.1: Canal profile.

7.2. Geometry

The canal with a trapezoidal section is represented in the Figure 7.1, and the general data is shown in Table 7.1. Checkpoints, sluice gates, pump stations and orifice offtakes are shown in Table 7.2:

Pool number	Pool length (Km)	Bottom slope (%)	Side slopes (H:V)	Manning's coefficient (n)	Bottom width (m)	Canal Depth (m)
I	2.5	0.1	1.5:1	0.025	1	2.5
II	2.5	0.1	1.5:1	0.025	1	2.5

Table 7.1: General canal features.

Number of control structure or checkpoint	Gate discharge coefficient	Gate width (m)	Gate height (m)	Step (m)	Discharge coef./diameter orifice offtake (m)	Orifice offtake height (m)	Lateral spillway height (m)	Lateral spillway width (m)/discharge coefficient
0*	0.61	5.0	2.5	0.6	-	-	-	-
1	0.61	5.0	2.5	0.6	0.6/0.77	1.0	2.3	500/1.99
2	-	-	-	-	0.6/0.77	1.0	2.3	500/1.99

Table 7.2: Checkpoints and Sluice gate/ pump station/ orifice offtake features (control structure).

*There is only a gate in section 0, there is not a checkpoint.

7.3. Discretization of a problem

The past time horizon length is defined equal to $T^{KF} = 14400$ seconds = 240 minutes = 4 hours. The time interval between successive control actions is $\Delta T = 300$ s = 5 min. The number of regulation periods during the past time horizon is $\gamma = 48$. The canal has been spatially discretized (Δx) every 20 meters, so the numbers of computational cross-sections in each pool are 125 and along the canal are 250.

7.4. Boundary and initial conditions

In these examples is considered an upstream large reservoir, whose water level $H_{\text{reservoir}}$ is 3 m constant throughout the test. This is the upstream boundary condition. At the end of the last pool, there is a control structure with orifice offtake and a pump station. The flow through the orifice offtake depends on the upstream water level of the orifice and the flow pumped is variable at every example. This is the downstream boundary condition. The examples start from an initial steady state with the demand delivery at the end of the pools constant ($5\text{m}^3/\text{s}$ through the orifice offtake), and the flow is not pumped at the beginning of those examples. The initial conditions are the same in all examples.

Control structure	Initial Flow rate (m ³ /s)	Control structure	Initial water level (m)
Gate 1	10.0	Checkpoint 1	2.0
Gate 2	5.0	Checkpoint 2	2.0

Table 7.3: Initial conditions in the canal.

Control structure	Flow delivered by an orifice offtake (m ³ /s)
Gravity outlet 1	5.0
Gravity outlet 2	5.0

Table 7.4: Flow delivered by an orifice offtake at the initial time step.

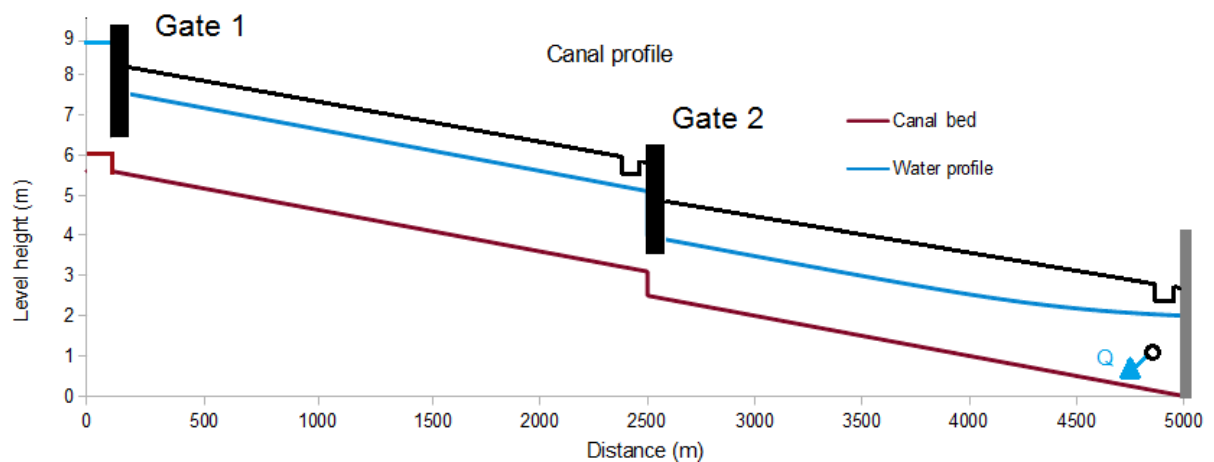


Figure 7.2: Backwater profile of the canal in the initial steady state.

CSI computes the unscheduled offtake changes introduced to the canal and write these offtake changes as a pump flow trajectory (temporal function) associated at a certain pump station. The algorithm assigns to the pump stations all flow changes between the initial steady state to the current state.

Once we have introduced input parameters used in CSI algorithm as the boundary and initial conditions, we have to define important variables in flow as the scheduled deliveries, the gates trajectories, pump flow trajectories and the measured water levels at the checkpoints during the past time horizon.

- Scheduled Deliveries (q_{offtake}): There are orifices offtake to convey flow rate to the farmers at the end of every pool. The water demand in all scenarios (shown in Table 7.4) was estimated at every regulation time step during the past time horizon and this information is known by CSI.

$$Q_{\text{offtake}} = C_d \times \frac{\pi \times \phi^2}{4} \sqrt{2g(y - y_0)} \quad (7.1)$$

where y is the water level in canal at offtake, y_0 is the orifice offtake height equal to $y_{\text{target}}/2$, C_d is the discharge coefficient and ϕ is the orifice offtake diameter.

- Measured water levels (Y^*): The water level is measured at every checkpoint at every time step during the past time horizon. We have to introduce the water level measurements to CSI in each of these examples.
- The gate trajectories (U): We have to define the sluice gate trajectories during the simulating horizon. Gate position is defined at every regulation period during the past time horizon. The gate position remains constant in these examples.
- The pump flow trajectories (q_{pump}): CSI has to compute the pump flow trajectories during the past time horizon. The pump flow trajectory vector is composed by two elements (two pumps stations) in a simulated time horizon of 48 regulation period, so the vector size is of 96 parameters.

CSI solves an optimization problem to set the pump flow trajectories. The algorithm uses an iterative method (Levenberg-Marquardt) that provides successive

approximations to the solution, and it is necessary to establish a first pump flow trajectory. In this respect, the Levenberg-Marquardt is a derived method of Newton methods used in optimization problems, which find the solution although the first approximation starts very far from the final minimum, obtaining an accurate solution. If the first approximation to the solution is close to the optimum solution, the method converges faster.

7.5. Test cases analyzed

In order to test the algorithm, we introduce different disturbances in the canal in each example, which are unknown for CSI. We will consequently get variations between the measured water level and the expected water level at every checkpoint during the past time horizon. In that sense, if we introduce a pump flow change in a pump station during a period of time, the initial steady state conditions of the canal is disturbed and the flow conditions becomes unsteady. At the end of the test, the canal comes back to the steady state because the duration of the pump flow disturbance is short.

7.5.1. First example: flow disturbance

The water level measurements (Figure 7.3) are obtained after introducing a flow disturbance of $2 \text{ m}^3/\text{s}$ (on the pump station 1 for 15 minutes, from the minute 40 to 55). These disturbances are introduced to the computer model as a pump flow change and we obtain the water level measurements from this model. Once the water level measurements are introduced in the CSI algorithm, the algorithm will propose the pump flow trajectories that describe with best accuracy the variation of water level at the checkpoints during the past time horizon which lasts 4 hours.

The flow disturbance reduces the water level at the checkpoint 1, from 2 m to 1.60 m, and at the checkpoint 2, from 2 m to 1.92 m, see Figure 7.3. A flow change of $2 \text{ m}^3/\text{s}$ at the pump station 1 has a significant impact on the canal profile of the canal, and the water level at the checkpoint 1 and 2 is recovered to the desired water level (2 m) in 160 minutes and 150 minutes, respectively. As soon as the water level at checkpoints recover the desired water level at this points, the flow through the orifice offtake come back to be $5 \text{ m}^3/\text{s}$.

The flow disturbance introduced in this test is always an outlet pump flow (positive value) but the algorithm also works well when the pump flow is an inflow or external runoff (negative value), as we will show in chapter 10.

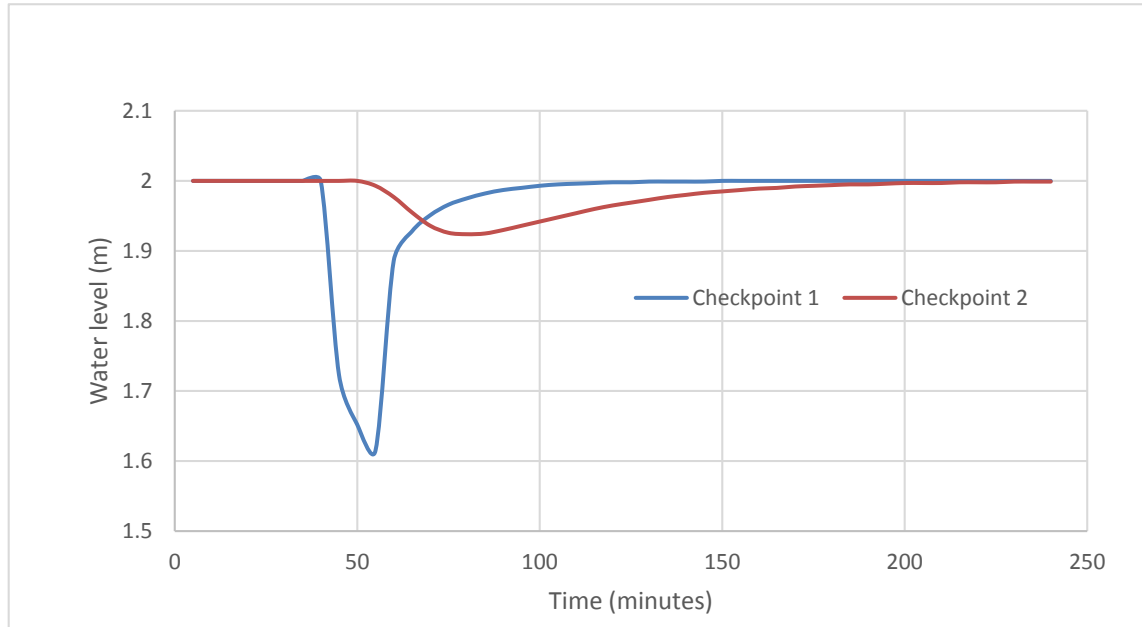


Figure 7.3: Water level measured at checkpoint 1 and 2 during the past time horizon in the first example.

The results of the pump flow obtained by CSI are shown in Figure 7.4. We can compare the result between the real pump flow (disturbance) and the simulating pump flow calculated by CSI. Both curves match exactly, differences between them are practically non-existent. The maximum error is around $0.005 \text{ m}^3/\text{s}$, taking into account that the real pump flow is $2 \text{ m}^3/\text{s}$, the percentage maximum error between the real and simulated result is 0.25%.

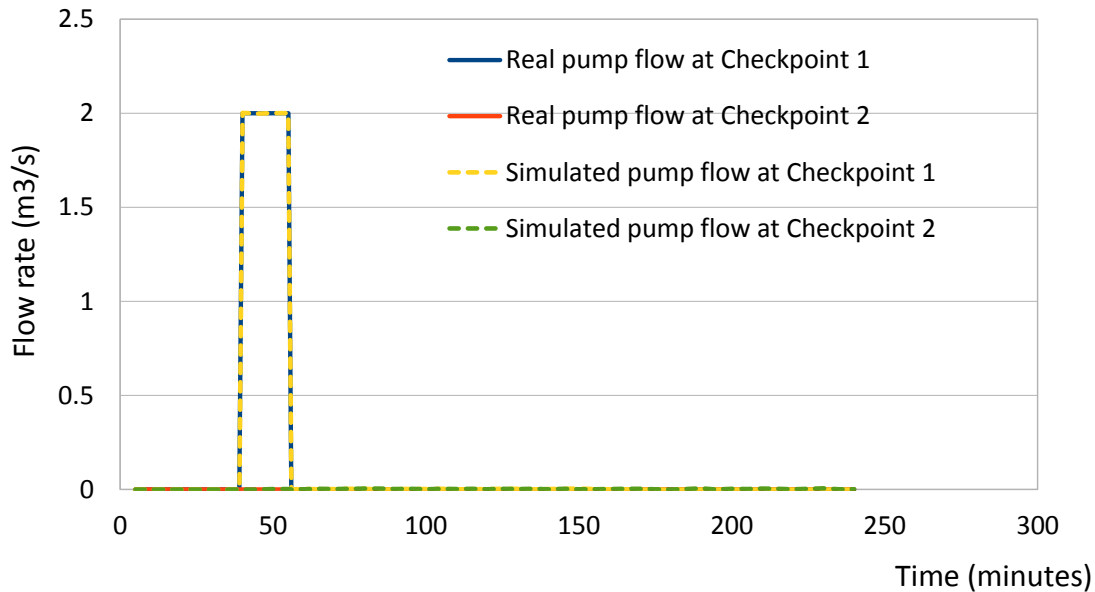


Figure 7.4: Pump flow at checkpoint 1 and 2 during the past time horizon in the first example.

7.5.2. Second example: small disturbance

The objective of this test is to set the accuracy of CSI algorithm in case of small water offtakes. We analyse CSI in a test in which the disturbance is very small. The difference between the real disturbance and the results obtained by CSI should be very small, much more in this example because the disturbance is reduced. It is possible in case of small disturbances that we could not differentiate between the real disturbance and the CSI error. If the error committed by CSI algorithm is much lower than the value of the real disturbance, CSI is an accurate algorithm.

We introduce a disturbance much lower than the disturbance introduced at the last example and we compare the similarity between the CSI's error respect to the value of the disturbance introduced. In this scenario, the disturbance is 50 l/s (a 1% of the total flow rate in the canal). This disturbance is introduced at the pump station 1 during 15 minutes from the minute 40 to 55.

In order to illustrate this example, if the error in the result is close to 25 l/s, the 0,5% of the total flow rate in the canal, the error would be similar to the disturbance, and the accuracy of CSI would be low in this test.

The disturbance reduces the water level at the checkpoint 1, from 2 m to 1.990 m, and at checkpoint 2, from 2 m to 1.998 m, see Figure 7.5. A flow change of 50 l/s at the pump station 1 will have a little impact on the water profile of the canal, and the water level at checkpoint 1 and 2 is recovered to the expected water level in 50 minutes and 100 minutes, respectively.

The water level measurements used by the CSI algorithm are shown in the Figure 7.5.

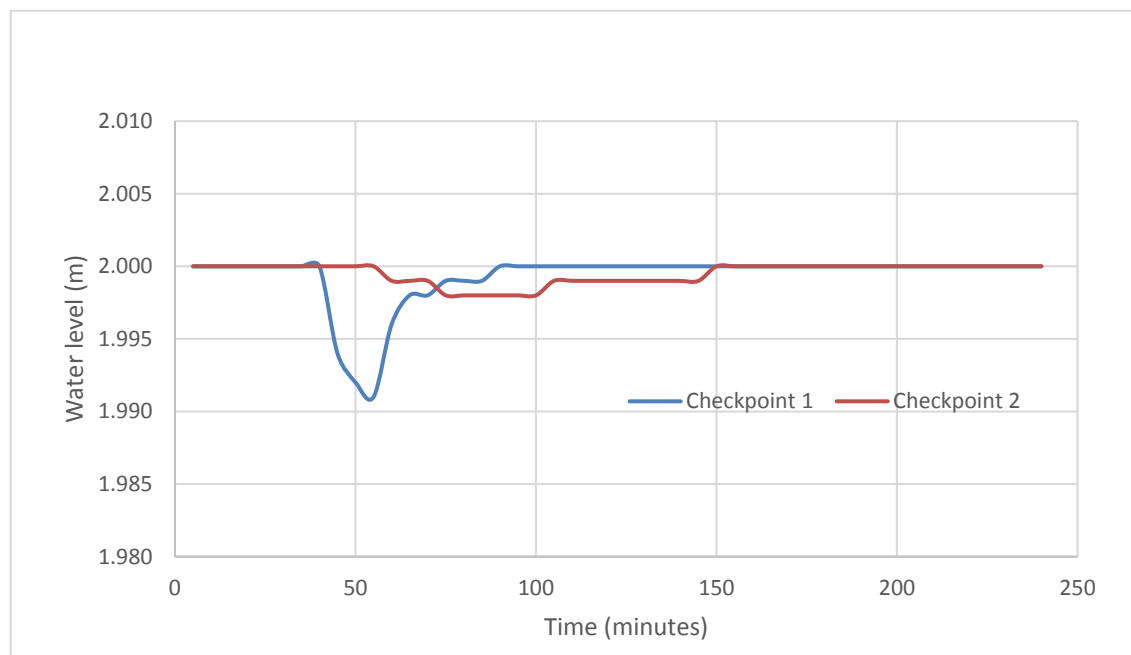


Figure 7.5: Water level measured in checkpoint 1 and 2 during a past time horizon in the second example.

The results of pump flow obtained by CSI are shown in the Figure 7.6. In all cases, the maximum error between the real and simulated solution is around 0.004 m³/s, as the real pump flow is 50 l/s, the percentage of maximum error is close to 8%. In that sense, the result obtained by CSI algorithm is a good approach to the real pump flow.

In this case, the only problem is to have a water level gage with such accuracy, although the gage should be more or less accurate depending on the geometry and features of the canal and the disturbance. In that example, an external disturbance value of 50 l/s represents a water level disturbance of few millimeters, and this water level variation can be too low to be measured accurately for a water gage.

Taking into account the maximum error obtained by CSI and the accuracy of a gage, CSI is enough accurate for the gages that we usually use.

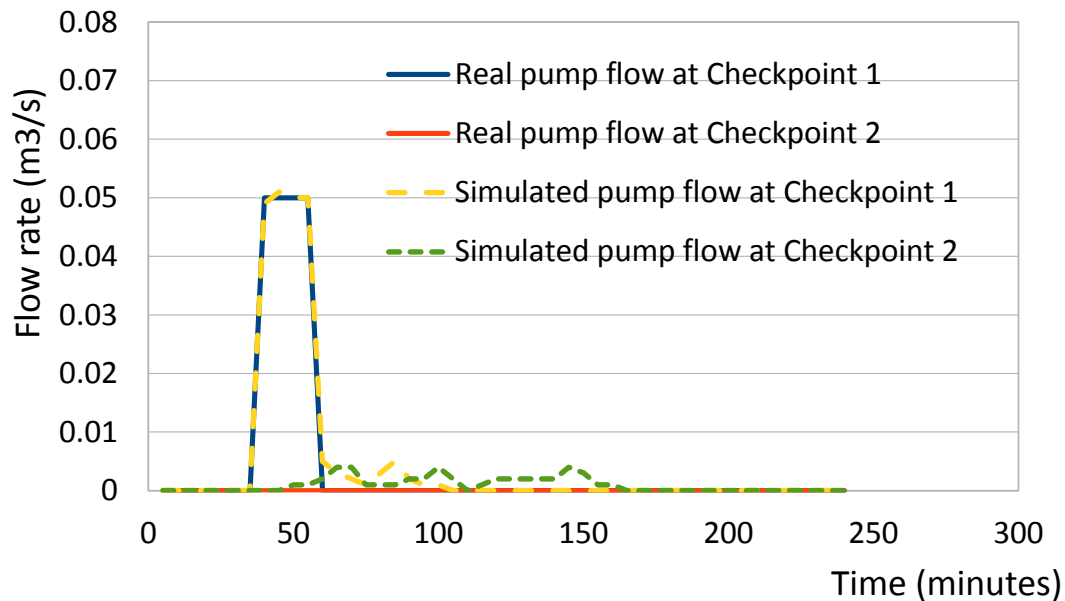


Figure 7.6: Pump flow in checkpoint 1 and 2 during a past time horizon in the second example.

7.5.3. Third example: flow disturbance not localized at a control structure

The CSI algorithm calculates pump flow changes at every control structure, but sometimes the pump unit can be located in other places, instead of in a known control structure. For this reason we establish this scenario to check the CSI behavior when the disturbance is introduced out of a control structure.

In this example, we introduce a disturbance at a certain cross-section of a canal (PK 3+750), where the pump flow change is located at the middle of the second pool, and this cross-section is **not a control structure**. The CSI has not any water level measures at PK 3+750, as this cross-section is not a node set by the algorithm as a checkpoint, so it is a “normal” node for the algorithm.

The water level measurements (Figure 7.7), which are used as input data in the CSI algorithm, are obtained after introducing in the computer model a flow disturbance of $2 \text{ m}^3/\text{s}$ (at the P.K. 3+750 for 60 minutes, from the minute 60 to 120).

The flow disturbance reduces the water level at the checkpoint 2 from 2 m to 1.375 m, and at checkpoint 1 is kept constant, due to the canal features and the flow conditions, see Figure 7.7. In that sense, a flow pumped of $2 \text{ m}^3/\text{s}$ at P.K. 3+750 will have a significant impact on the checkpoint 2 and little impact at checkpoint 1. The water level at the checkpoint 2 is recovered to the desired water level (2 m) in 130 minutes. The flow through the second orifice offtake come back to be $5 \text{ m}^3/\text{s}$ at the end of the test.

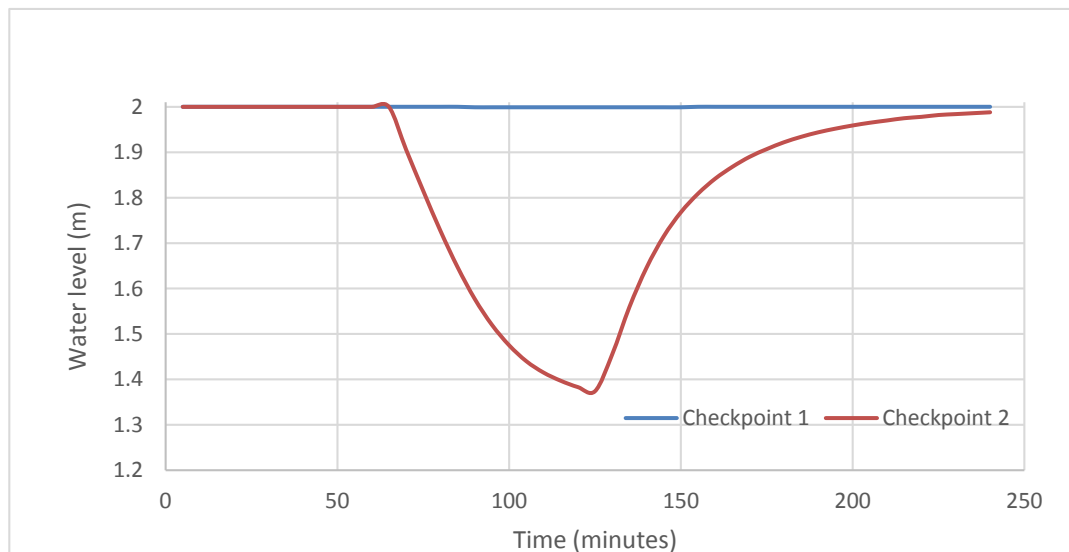


Figure 7.7: Water level measured in checkpoint 1 and 2 during a past time horizon in the third example.

In this example the CSI algorithm has to adapt a new situation, because CSI is only able to calculate disturbances at a control structure, but in this case the flow rate

pumping is not localized at a control structure, but in an unknown cross-section in the canal, between two checkpoints.

The results obtained by CSI are shown in Figure 7.8. The CSI algorithm distributes the pumped flow in an unknown cross-section to the upstream and downstream pump stations.

The simulated pump flow at "checkpoint 2" (P.K. 5+000) starts 5 minutes after that the real pump flow at PK 3+750 starts to pumping. This is because, until the depressive translatory wave does not reach to checkpoint 2, 1250 meter downstream from the PK 3+750, there is not any change on water levels measurements at checkpoint 2, so the CSI algorithm does not have any notice about the pump flow change. This 5 minutes of difference correspond to the time that the wave travels from P.K 3+750 to P.K 5+000.

The travel time was also computed by the equation (7.2), and the value was approximately 287 seconds, with an average velocity of 0.54 m/s and an average celerity of 3.80 m/s obtained from the computer model.

$$Time = \frac{(X_{P.K.5+000} - X_{P.K.3+750})}{(Velocity + Celerity)} \approx 287 \text{ seconds} \quad (7.2)$$

Depending upon the value of the hydraulic depth, the wave velocity may be 10 times faster than the mean water velocity. In reality, flow will begin to decrease at cross-sections downstream as soon as the leading edge of the translatory wave front arrives, which is much sooner than would be predicted from the average flow velocity in the canal. The total amount of the flow change will reach later at points downstream, due to the fact that the total flow travels at the average flow velocity in the canal. Once the depressive translatory wave arrives at checkpoint 2, the water level starts to decrease, but the checkpoint 2 does not have any notice of the true value of this disturbance (2 m³/s) until 45 minutes later, once the total amount of the flow change reaches at checkpoint 2.

We can check that CSI assigns completely the flow change at P.K. 3+750 to the pump station at checkpoint 2, and this is a logical solution because the water level variation at checkpoint 1 is almost null.

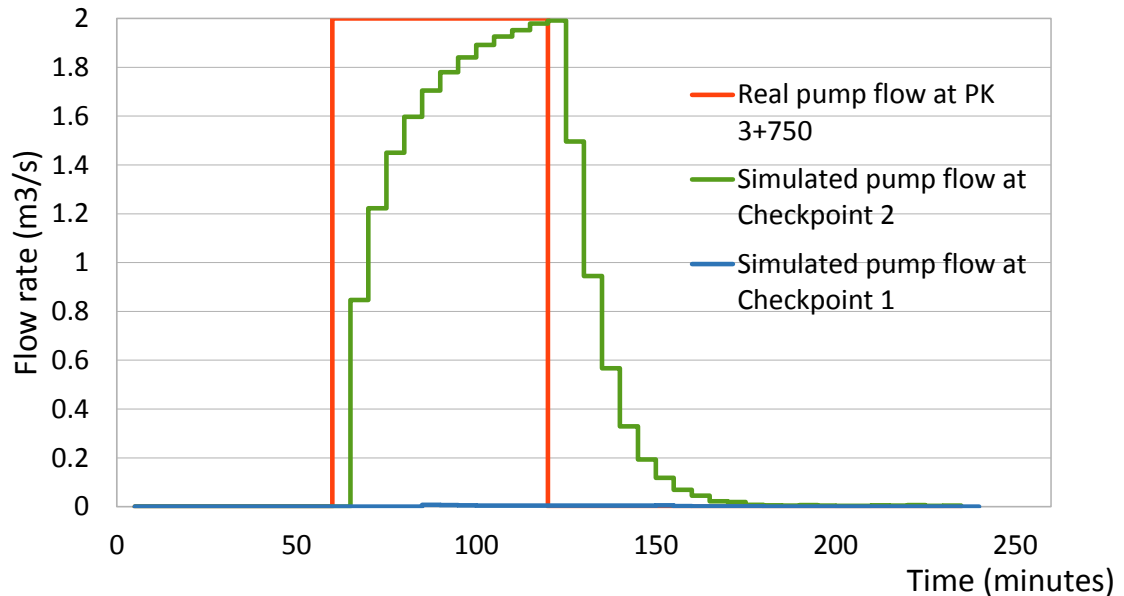


Figure 7.8: Pump flow in checkpoint 1 and 2 during a past time horizon in the third example.

There are two remarkable things:

- If the pump flow change value had been lower, for instance 50 l/s, and the period of time pumping is much lower, CSI would have never known the maximum flow change value due to the hydrograph routing.
- The CSI algorithm assigns entirely the pump flow change introduced at P.K. 3+750 to the pump station close to the checkpoint 2. The pump station close to the checkpoint 1 does not pump because the water level measures at checkpoint 1 are very close to 2 m (the water level at checkpoint 1 in steady state). In case that the depressive translatory wave* reaches checkpoint 1, the friction forces have attenuated a lot the height of the wave front. For this reason the water level changes at checkpoint 1 are close to 0. This happens because the canal is very steep and the height of the wave front is not sufficiently high to change a lot the water level at checkpoint 1. For this reason, this example was tested with the same flow conditions but modifying the canal geometry (example 3a), the canal slope was reduced to 0.01% in order to get a higher

depressive translatory wave at checkpoint 1. The results obtained by CSI are shown at Figure 7.9. The results were as we expected because the pump flow changes were distributed between the pump station at checkpoint 1 and 2, although the pump flow values at checkpoint 2 was higher than 1.

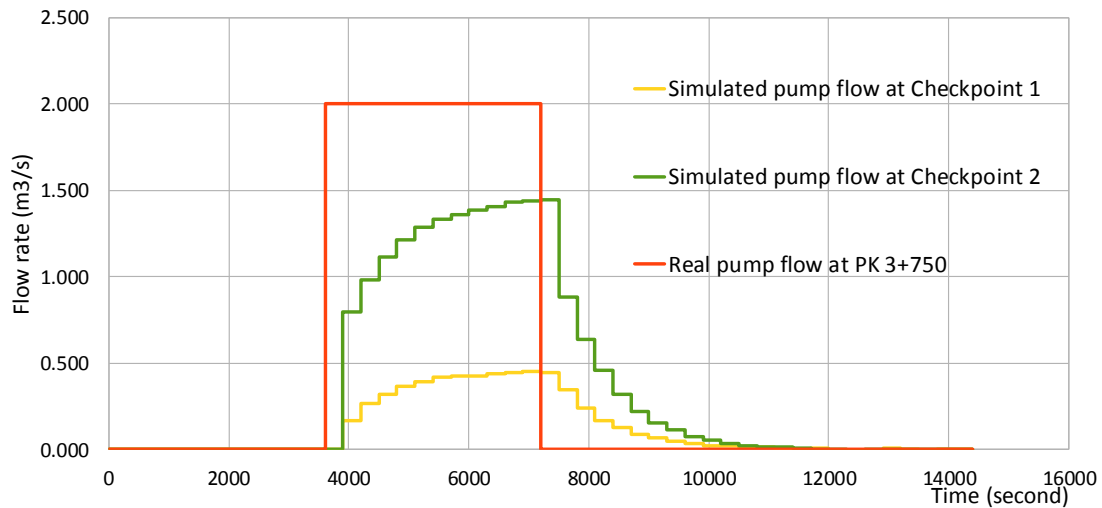


Figure 7.9: Pump flow in checkpoint 1 and 2 during a past time horizon in example 3a.

But the more interesting thing about this scenario was:

As we show in Table 7.5, the water volume extracted by the real pump flow at P.K. 3+750 (Figure 7.8) is similar to the water volume extracted by the proposed pump flows at checkpoint 1 and 2 in the third example, with an error of 0.28%. This error value is the same in the scenario 3a with a canal slope of 0.01 %.

*A translatory wave is a gravity wave that propagates in an open canal and results in displacement of water particles in a direction parallel to the flow.

Water volume extracted by the real pump flow hydrograph at P.K. 3+750 (m3)	Water volume extracted by the simulating pump flow hydrograph at checkpoint 1 and 2 (m3)	Error (%)
7200	7180,4	0,283

Table 7.5: Water volume error between the simulated and real flow hydrographs in the third example.

The watermaster can determine how much water has been extracted in the canal and it is possible to determine the location too, as we can use the input and output data of the algorithm to calculate it.

In this case, the disturbance is introduced in the P.K. 3+750 at $T=3600$ seconds. The wave generated by the disturbance in the example 3a travels upstream with a velocity ($v+c$) of 4.21 m/s and downstream with a velocity of 5.20 m/s. The sensor 1 and 2 measures the water level every 5 min (300 seconds), when the sensor measures the water level at $T=3900$ seconds the perturbation has arrived to the checkpoint 1 and 2 according to the output data of CSI, so the wave could arrive to the sensor at checkpoint 1 just 1 second later that the sensor measures the water level ($T=3601$ seconds) or 299 seconds later ($T=3899$ seconds). To obtain the location of the disturbance could draw in the computational grid of the canal four red lines (Figure 7.10), two lines from upstream ($v-c$) and two lines from downstream ($v+c$), each of them defining a wave that arrives 1 second later than the sensor measures the water level (3601 seconds) and 299 seconds later (3899 seconds). The intersection of these lines define an area (A-B-C-D) in the space-time diagram and every intersection point could be the solution for this disturbance. It is possible to make a good estimation of the pump location depending of the period of time between water level measurements.

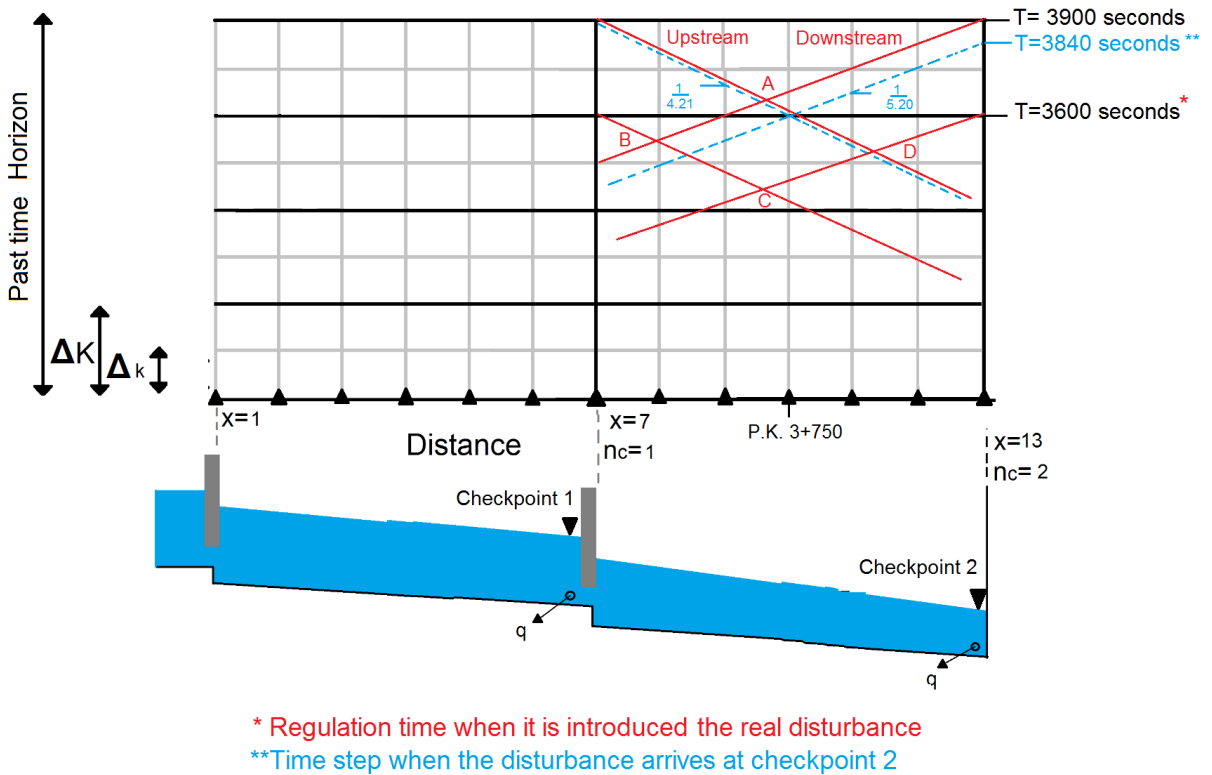


Figure 7.10: The wave generated by the disturbance traveling upstream and downstream from P.K. 3+750.

In this case as we know the location, velocity and celerity of the disturbance introduced in the canal, we can determine exactly the time that the wave needs to arrive downstream (checkpoint 2) and upstream (checkpoint 1) and we can check easily if this point is inside of the area (A-B-C-D).

$$Time (checkpoint 2) = \frac{(X_{P.K.5+000} - X_{P.K.3+750})}{(Velocity + Celerity)} \approx 240 \text{ seconds} \quad (7.3)$$

$$Time (checkpoint 1) = \frac{(X_{P.K.3+750} - X_{P.K.2+500})}{(Velocity + Celerity)} \approx 296 \text{ seconds}$$

7.6. Associated errors due to the loss of water level measurements

We try to set the behaviour of the algorithm when the water level measures at a checkpoint have been lost or simply the water level sensor is not working properly. To represent this case, we introduce a new checkpoint (checkpoint X) and a new pump station respect to the first example, both located at P.K. 3+750. This pump introduces an flow disturbance of $2 \text{ m}^3/\text{s}$ (at the P.K. 3+750 during 60 minutes, from the minute 60 to 120). In case the CSI algorithm knows this new pump station at point P.K. 3+750, the algorithm assigns a new control structure at node P.K. 3+750. But if the water level sensor at checkpoint X (P.K. 3 750) is not working, we do not have water level measures at this cross-section.

In that case, the geometry, the discretization of the problem, the initial condition and the disturbance is equal to the third example, so we can use the water level measurements at the checkpoints 1 and 2 of the third example. To consult this information see the section 7.5.3.

The pump flow trajectory simulated by CSI is quite different to the pump flow trajectory introduced to the canal, as we can see in Figure 7.11. Because the disturbance was introduced at P.K. 3+750 of the canal, the only pump station that must pump for CSI should be the pump station at P.K. 3+750. However the pump station at P.K. 5+000 is also pumping for the algorithm. CSI divides the total flow rate extracted at P.K. 3+750. Most of the flow rate is assigned to the pump station at P.K 3+750 and the lower flow rate is assigned to the pump station at P.K 5+000, as we check at Figure 7.11.

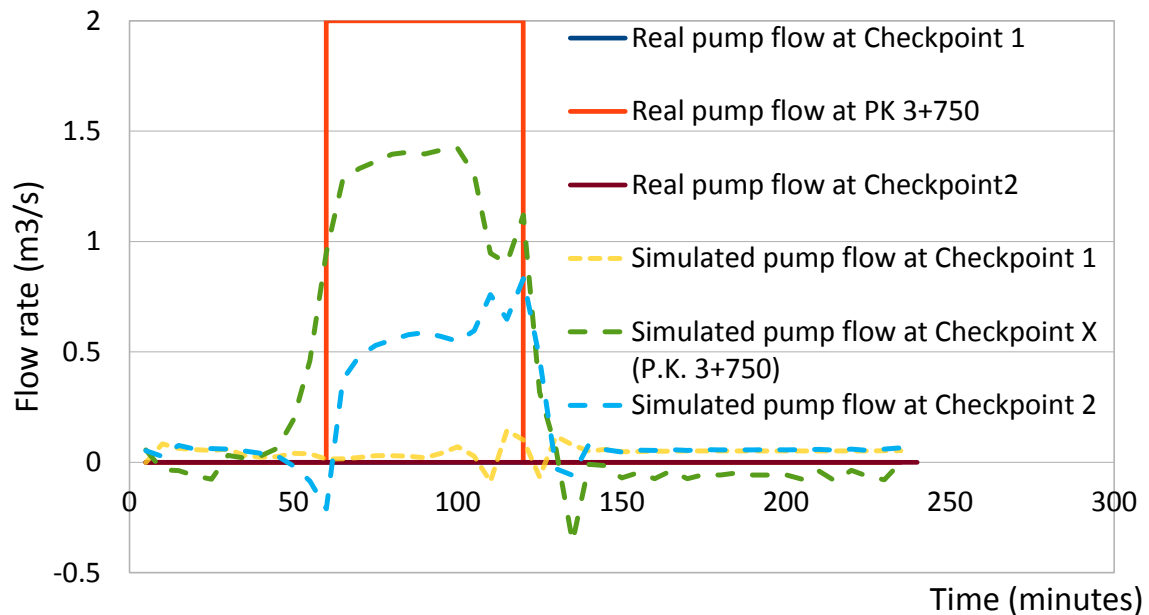


Figure 7.11: Pump flow in checkpoint 1 and 2 during a past time horizon.

The water volume extracted by the real pump flow hydrograph at P.K. 3+750 (Figure 7.11) is slightly different to the water volume extracted by the sum of the simulated hydrographs at checkpoint 1, 2 and X, with an error of 11% (see Table 7.6).

Water volume extracted by the real pump flow hydrograph in P.K. 3+750 (m ³)	Water volume extracted by the simulated pump flow hydrograph at checkpoint 1 and 2 (m ³)	Error (%)
7200	7885.78	10.95

Table 7.6: Water volume error between the simulated and real flow hydrographs.

The expected result in this example would be to assign the pump flow change to the new pump station at P.K. 3+750, as this is what really happens. But the question is: if the algorithm does not get water level measures at P.K. 3+750, only at checkpoints 1 and 2; can the algorithm obtain the real pump flow trajectories at checkpoint 1, X and 2? The first answer would be no, but we are going to explain this statement.

In the next figure (Figure 7.12), we can observe the water level measures at the checkpoints introduced to CSI, and the water level simulated by the pump flow trajectories calculated by CSI. Both water level trajectories are similar but the pump flow trajectories are different. The maximum difference between the water level measured and simulated at checkpoint 2 is 2 centimeters at a time step 113 min, and the other values are similar with differences of 2-5 mm during the past time horizon. In the next figure (Figure 7.13) is only analyzed the results at the checkpoint 2, as it is the checkpoint with bigger deviation of results.

CSI has calculated an extracted flow vector which reproduces quite well the water level error introduced in the algorithm, but the pump trajectories are not the expected result.

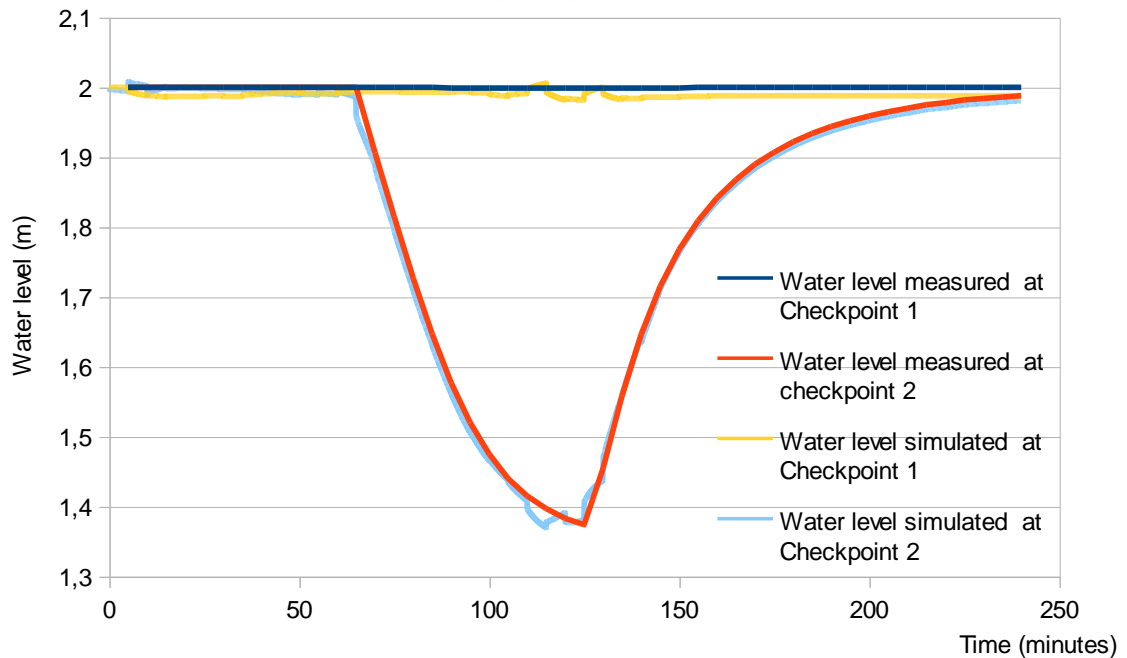


Figure 7.12: Water level measurements V.S. simulated water levels by CSI at checkpoint 1 and 2 during the past time horizon.

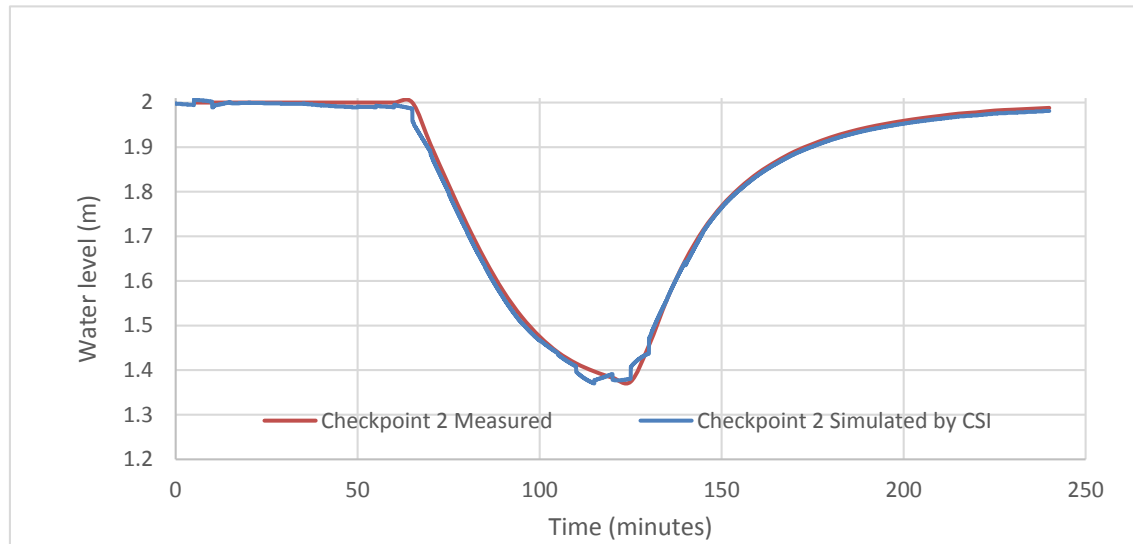


Figure 7.13: Water level measurements at checkpoint 2 V.S. simulated water level by CSI at checkpoint 2 during the past time horizon.

7.7. Associated errors due to measurement errors in depth gages

The water level at the checkpoints is measured by sensors which set the water level from an electric signal. Occasionally, this electric signal has an error due to electrical noise or simply due to slight variations of water level at the checkpoint. This error can be significant in some case, and although the signal is filtered, the water level measurements provided by the sensor are not fully precise.

In this section, we have the purpose to check the sensivity of the CSI algorithm to measurement errors. We introduce an error in the water level measurements and we test again the first and third example with CSI algorithm. In this way, we introduce three new cases for the first example:

- A) We have an error of +2 cm in water level measurements at checkpoint 1 and 2 of the first example (flow disturbance), so the water level function at checkpoint 1 and 2 (Figure 7.3) is moved.

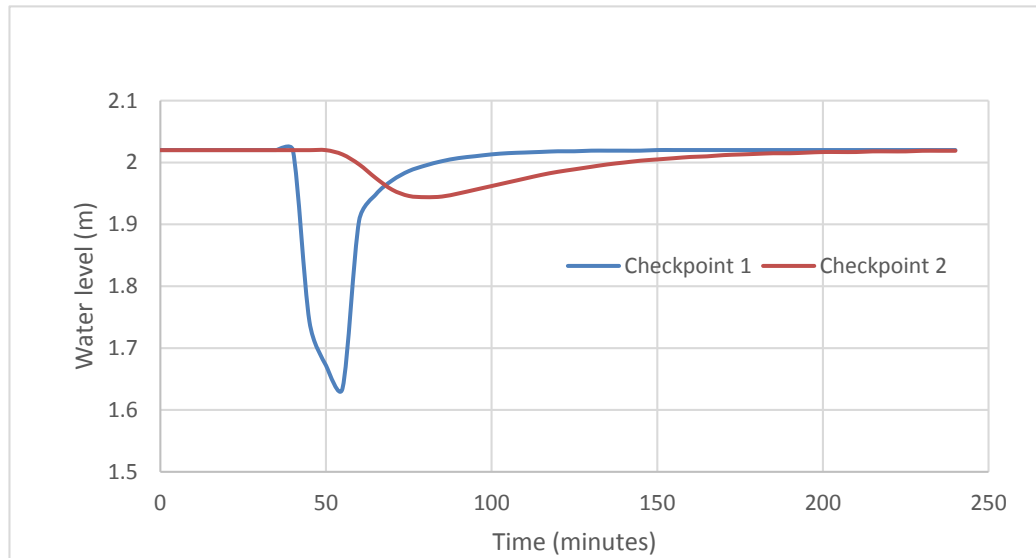


Figure 7.14: Water level measured at checkpoint 1 and 2 during a past time horizon (First example: Case:+2 cm).

- B) We introduce an error of -2 cm in water level measurements at checkpoint 1 and 2 of the first example (flow disturbance). The water level function at checkpoint 1 and 2 (Figure 7.3) is moved.

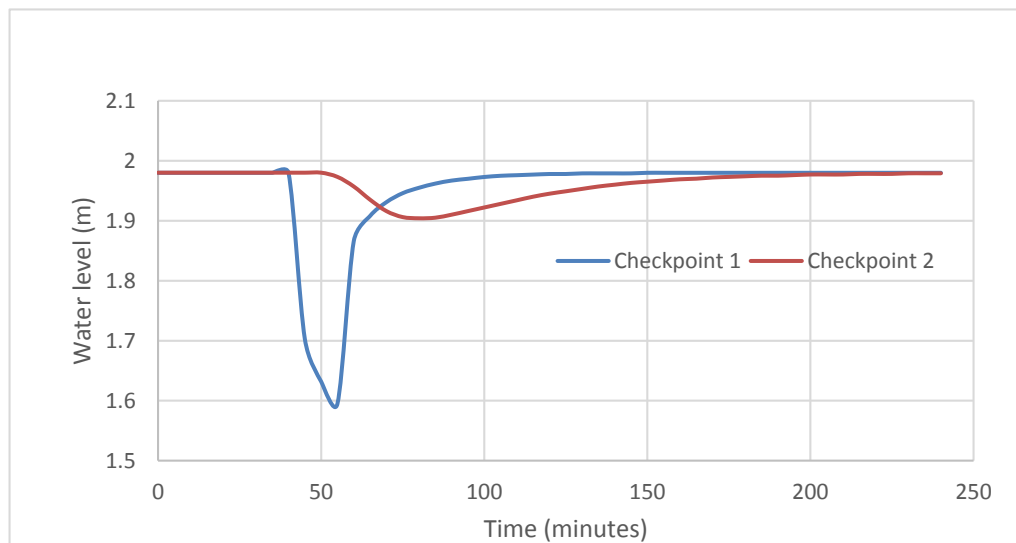


Figure 7.15: Water level measured at checkpoint 1 and 2 during a past time horizon (First example: Case:-2cm).

- C) We introduce an error of +2 cm in the water level measurements at the first time step, and – 2cm in the water level measurement at the second time step, and so on, at checkpoint 1 and 2 in the first example. In that sense, the water level function at checkpoint 1 and 2 (Figure 7.3) has a saw-tooth shape oscillating with +/- 2 cm variation.

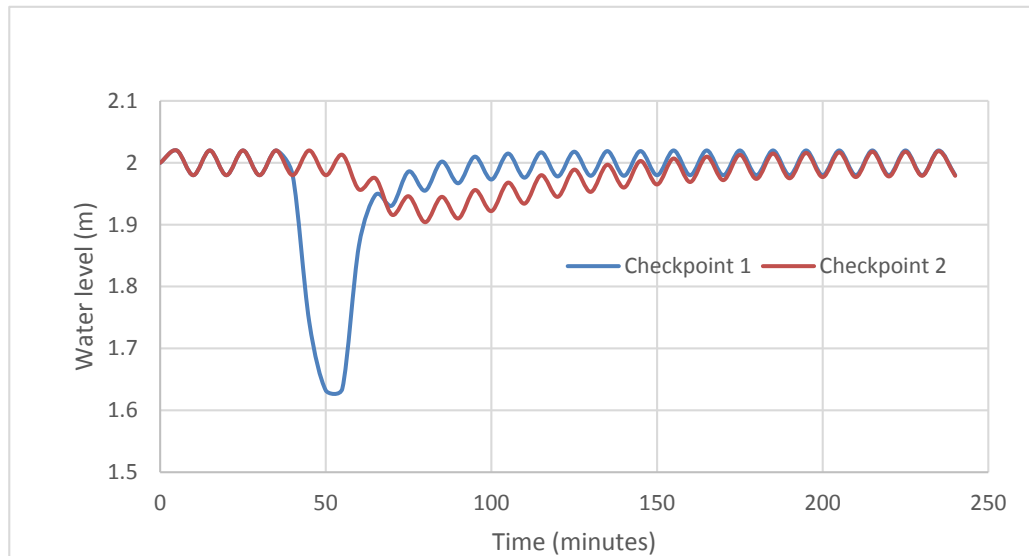


Figure 7.16: Water level measured at checkpoint 1 and 2 during a past time horizon (First example: Case:-/+2cm).

The results obtained by CSI from this data are shown in Figure 7.17, Figure 7.18 and Figure 7.19.

We got an error in pump flow associated with the error in water level measurements.

- A) The error in pump flow result is almost constant (Figure 7.17), and the error is easier to define at the end of the test (steady state). We can define in general that the pump flow error at checkpoint 1 is approximately -0.084 m³/s and at checkpoint 2 is -0.02 m³/s. As the depth error increases the real water level in two centimeters, the CSI computes the necessary pump flow to increase the water level at checkpoints in two centimeters. The pump has to introduce a flow of -0.084 m³/s at checkpoint 1 and -0.02 m³/s at checkpoint 2. The percentage of maximum pump flow error is 4%.

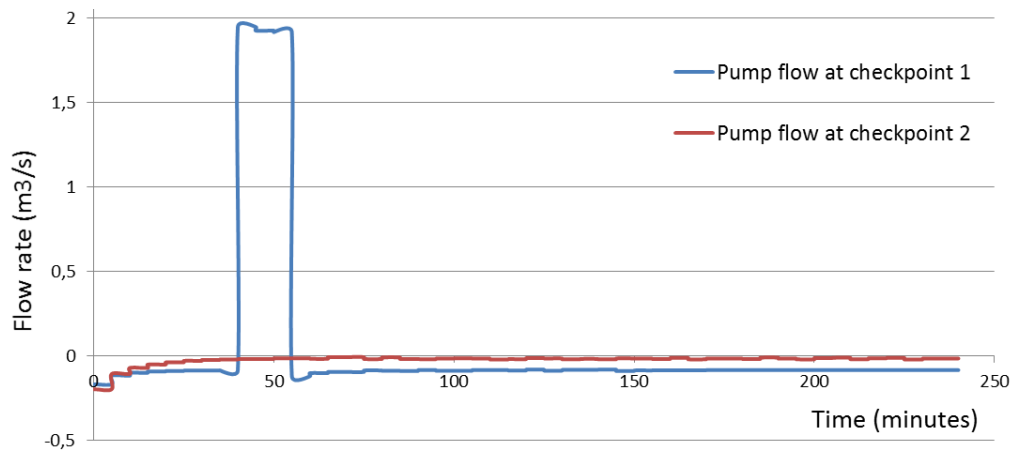


Figure 7.17: Pump flow at checkpoint 1 and 2 during the past time horizon (First example: Case:+2cm).

- B) The error in pump flow result is almost constant in Figure 7.18, and the error is easier to define at the end of the test (steady state) too. We can define in general that the pump flow error at checkpoint 1 is approximately $0.085 \text{ m}^3/\text{s}$ and at checkpoint 2 is $0.016 \text{ m}^3/\text{s}$. As the depth error decreases the real water level in two centimeters, the CSI computes the necessary pump flow to reduce the water level at checkpoints in two centimeters. That is, the pump has to extract more flow. The percentage of pump flow error is 4%.

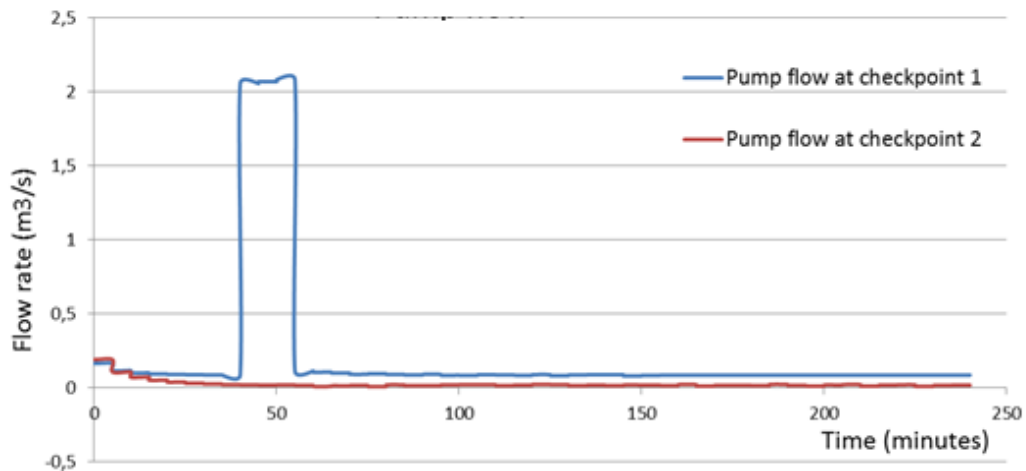


Figure 7.18: Pump flow at checkpoint 1 and 2 during the past time horizon (First example: Case:-2cm).

- C) In that case, the error in pump flow changes in each regulation time from the depth error (Figure 7.19), with an absolute error of 4 cm. For that reason, the pump flow error at checkpoint 1 changes from $0.203 \text{ m}^3/\text{s}$ to $-0.204 \text{ m}^3/\text{s}$ and at the checkpoint 2 from $0.242 \text{ m}^3/\text{s}$ to $-0.245 \text{ m}^3/\text{s}$. The percentage of maximum flow error is 12%.

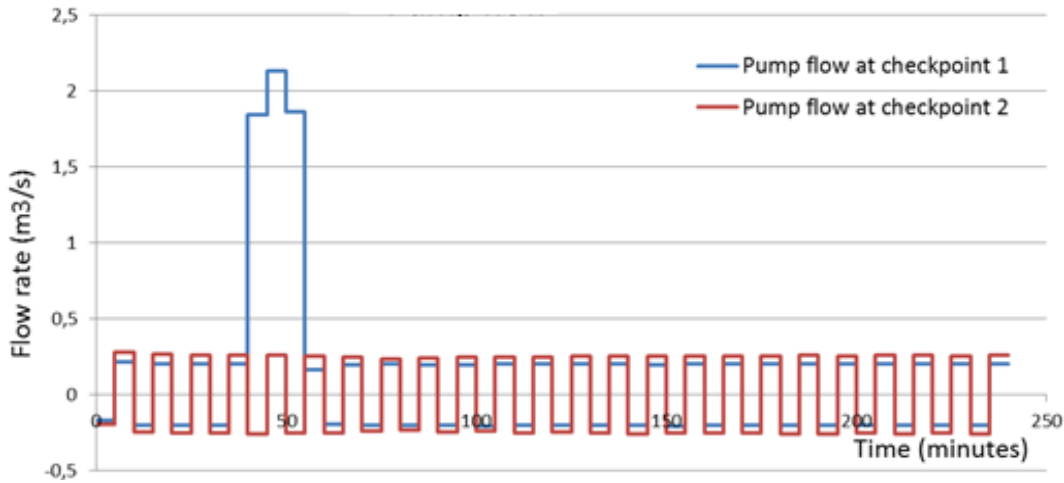


Figure 7.19: Pump flow at checkpoint 1 and 2 during the past time horizon (First example: Case:-/+2cm).

There is a proportional relation between the depth gage error and the pump flow error for this canal.

We have checked the first example taking into account errors in the water level measurements. In the next example, we will test the third example taking into account an error in the water level measurements at checkpoint 1 and 2 of +2cm. We want to remember that in the third example the disturbance was introduced at an unknown cross-section for the CSI which was not defined as a pump station.

We introduced an error of +2 cm in all water level measurements at checkpoint 1 and 2, see Figure 7.20.

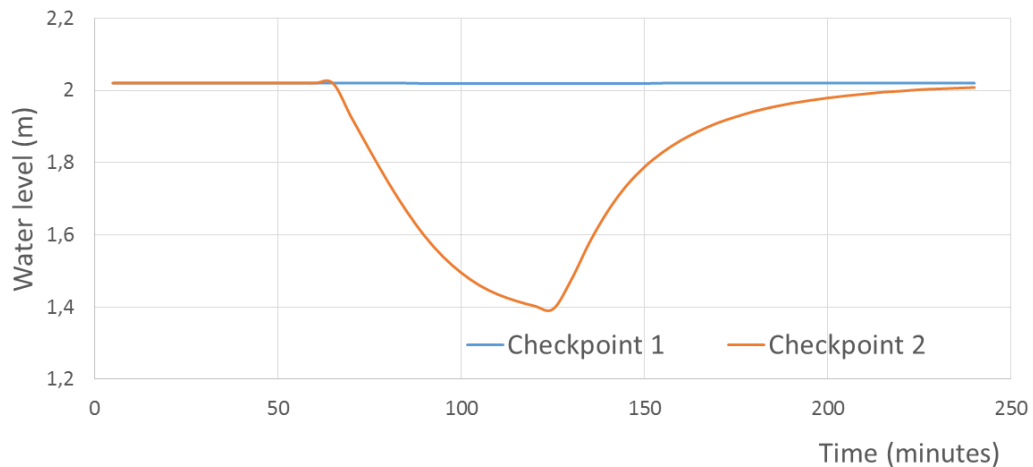


Figure 7.20: Water level measured at checkpoint 1 and 2 during the past time horizon (Third example: Case:+2cm).

We got an error in pump flow result associated with the depth gage error in water level measurements as in the other examples.

The error in pump flow result is almost constant comparing with the pump flow in the third example, as we show in Figure 7.21. The flow error is similar to the error obtained in the first example with a depth gage error variation of +2cm. We can define in general that the pump flow error at checkpoint 1 is approximately $-0.084 \text{ m}^3/\text{s}$ and at checkpoint 2 is $-0.02 \text{ m}^3/\text{s}$. The percentage of maximum pump flow error is 4%. So the pump flow trajectory obtained in this case is similar to the pump flow trajectory obtained in the third example without depth gage error (Figure 7.8).

As we show in Table 7.7, the water volume extracted by the real pump flow hydrograph at P.K. 3+750 (Figure 7.6) and the simulating pump by CSI with depth gage error are quite similar, in case of an error of +2cm in water level measurements. The water volume error between both hydrographs is 6.5 % (Table 7.7). The difference in water volume between both hydrographs is due to the flow rate error of $0.085 \text{ m}^3/\text{s}$ at checkpoint 1 and $0.02 \text{ m}^3/\text{s}$ at checkpoint 2, although the flow rate error is low, if the duration of the disturbance is around 95 min (Figure 7.21), the water volume is not a negligible amount.

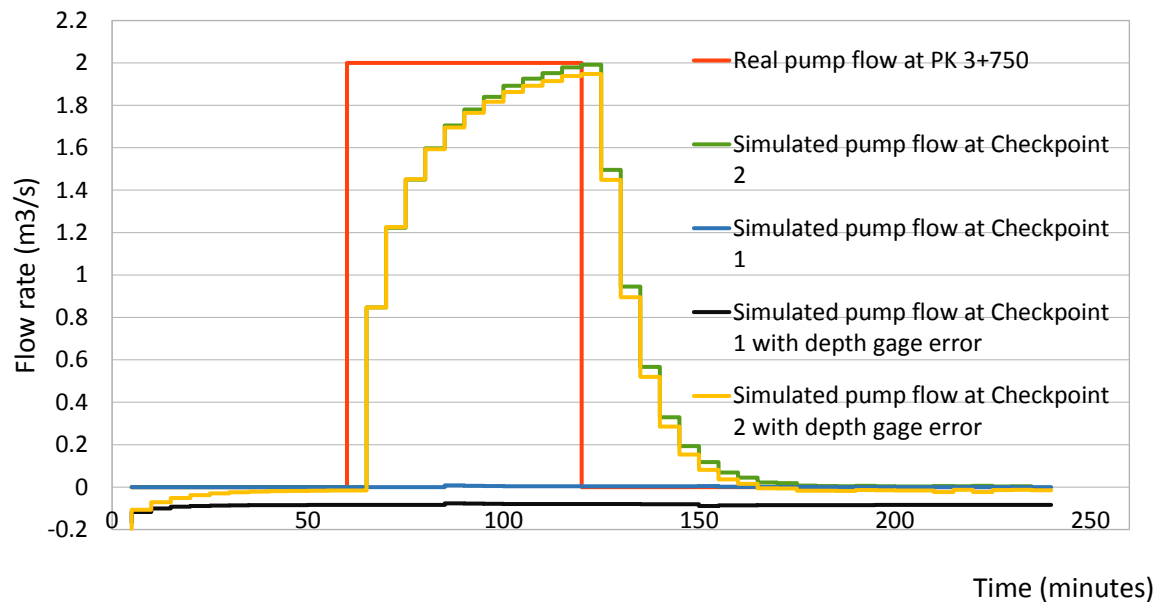


Figure 7.21: Pump flow at checkpoint 1 and 2 during the past time horizon (Third example: Case: -/+2cm).

Water volume by the real pump flow hydrograph in P.K. 3+750 (m3)	Water volume by the simulated pump flow hydrograph at checkpoint 1 and 2 (m3) in case of an error noise of +2cm	Error (%)
7200	6725.5	6.5

Table 7.7: Water volume error between the simulated and real flow hydrographs from the minute 65 to minute 160 (Third example: Case +2cm).

7.8. Hydrodynamic canal state

We show in this section that the CSI can also define the hydrodynamic state of a canal during a past time horizon and the current canal state, that is, the velocity and water level in each cross-section of the canal. This data is very useful for a feedback controller which has to know the hydrodynamic canal state in real time to calculate the optimum gate trajectories.

We have to show that CSI is a useful tool for calculating the hydrodynamic canal state, for that reason, we show the hydrodynamic canal state calculated by CSI in the first example at 2700 seconds (5 minutes after introducing the disturbance) and at 3900 seconds (10 minutes after disappearing the disturbance). We can check the water level measurements at checkpoint 1 and 2 at Figure 7.22.

In the next figures (Figure 7.23 and Figure 7.24), we show the water profile calculated by CSI and the water profile calculated by a computer model (taking into account the real extracted flow). The error between the real extracted flow and the extracted flow calculated by CSI is around 5 l/s in that case, as we introduced in the section 7.5.1. Considering that the dimension of the cross sections in the canal, an error of 5 l/s at the checkpoint represents a water level variation around 1 mm in some cross-sections (see Figure 7.23 and Figure 7.24).

In the Appendix 1, we show the values of water velocity along the canal obtained by CSI and a computer model at 2700 seconds and 3900 seconds.

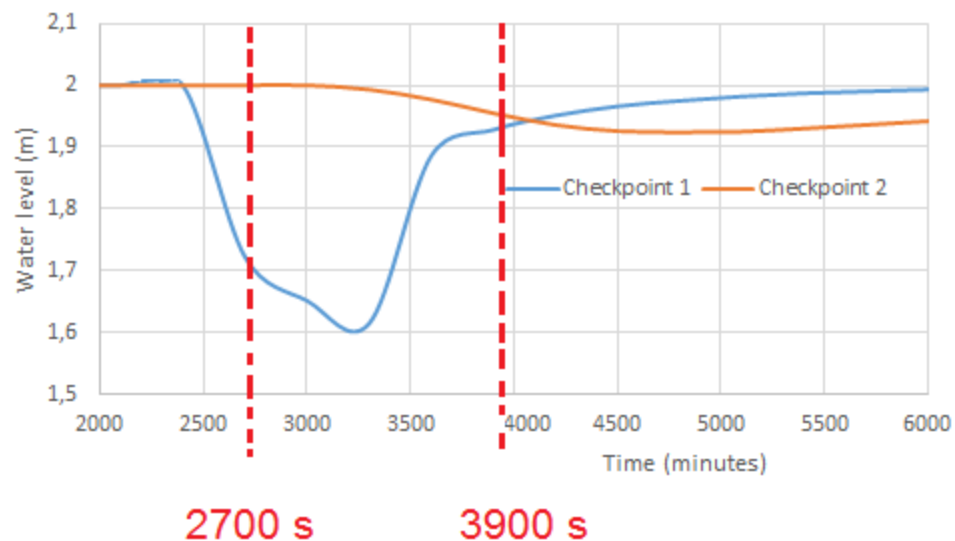


Figure 7.22: Water level measurements in the first example (pointed at 2700 s and 3900s).

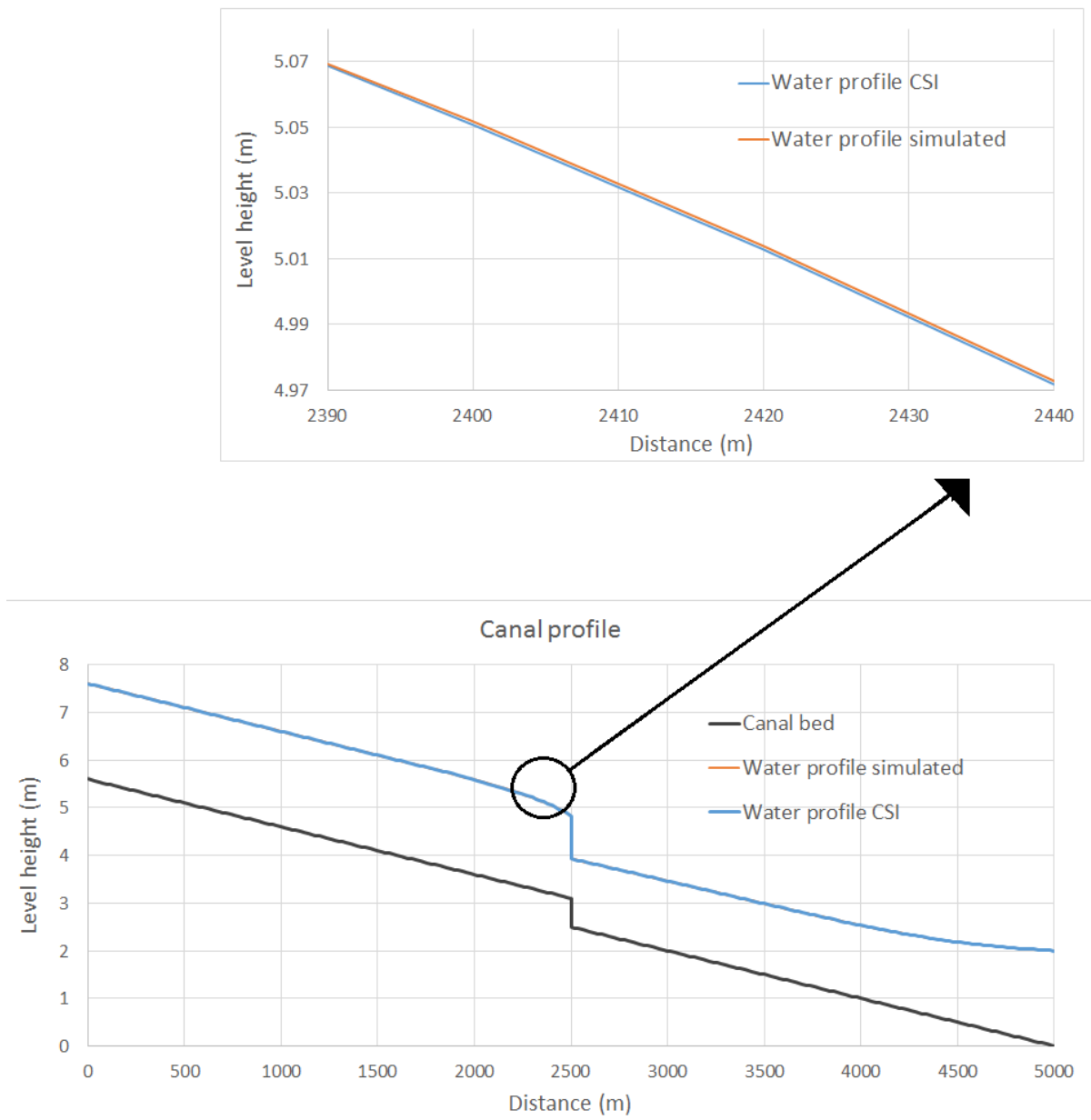


Figure 7.23: Water profile obtained by a computer model V.S. Water profile obtained by CSI at 2700s.

Both water profiles are similar, as the accuracy of CSI calculating the extracted flow vector is so high. An error of 5 l/s in the extracted flow is equivalent to an error in water level of 1 mm and in water velocity of 0.001 m/s in cross-sections close to the extracted point.

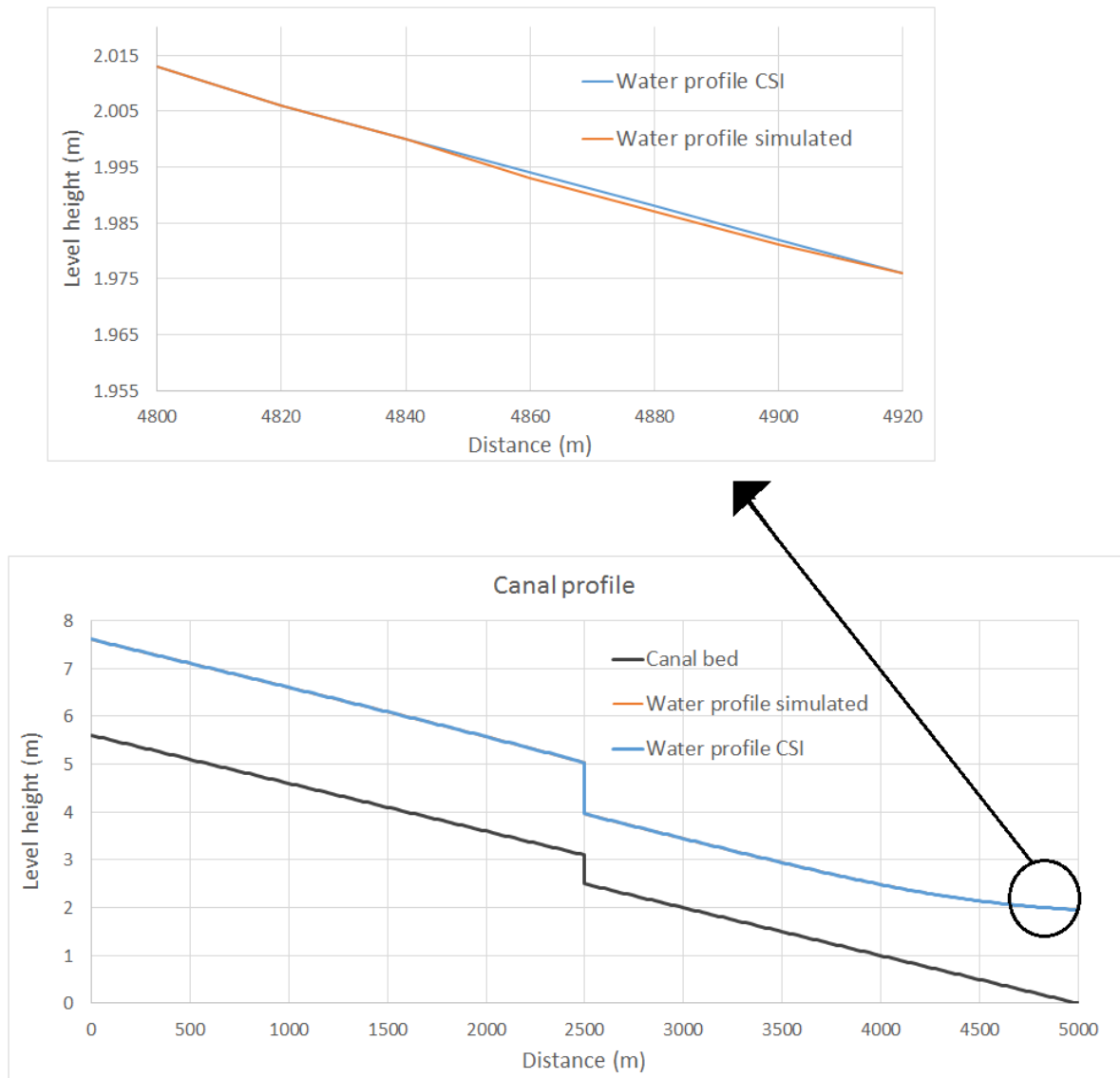


Figure 7.24: Water profile obtained by the V.S. Water profile obtained by CSI at 3900s.

The second case at 3900 s is equivalent to the other case. Here the cross sections with a water level error around of 1mm are close to the second extraction point, in which the extracted flow error is around 5 l/s.

The hydrodynamic canal state obtained in both cases is similar using a computer model operating with the real disturbance or using CSI which previously calculates the disturbance and then the hydrodynamic canal state.

7.9. Conclusions

All these tests have shown excellent results, because the algorithm finds a solution of the problem and reproduces the pump flow trajectories correctly in most cases. These examples have afforded us to know the limits of the algorithm, which is quite important, because it is necessary to know the limits of the algorithm for not demanding things which it is not able to do.

If we look at examples 1 and 2, the real pump flow changes introduced at the checkpoint 1 and the pump flow obtained by CSI are equal, with a maximum error between them lower than $0.005 \text{ m}^3/\text{s}$ (0.25 %) and $0.004 \text{ m}^3/\text{s}$ (8%), respectively. If the pump flow change is introduced at a node, which CSI recognizes as a pump station and there is a checkpoint associated to this pump station, the results are good.

The results obtained in example 3 are not so good but these results are valuable. In the example 3 although it is impossible to estimate correctly the pump flow hydrograph, it is possible to estimate how much water is extracted from the canal with a maximum error of 0.28 %. It is also possible to locate quite well the cross-section where the pump extracts the flow rate, with more or less accuracy depending on the time frequency between measurements and the value of the pump flow change. This is not an unusual scenario in a real canal, because we cannot always count on having checkpoints close to the pump which is operating, as the disturbances are usually introduced without permission and without knowledge of the watermaster.

In case that a pump extracts water at an unknown cross-section in the canal, and the algorithm has not assigned any checkpoint in this localization, CSI can obtain the total volume of water extracted by the pump.

In the case of associated errors due to the loss of water level measurements at checkpoints, we try to show that it is not a good idea to have more pump stations or variables than checkpoints or measurement points. In cases that there are more pumps stations than measurement points, there are multiple solutions or pump flow trajectories for reaching to similar water level measures at the checkpoints. Instead, there is only one solution when we know which pump extracts water and which not. The pump flow results are brilliant in this example 3, when the algorithm knows that the only operative pump station is the pump station at P.K. 3+750.

In case of loss of accuracy in water level sensors, we have to be careful with the data obtained from the gages. In those cases, CSI has obtained good results taking into account an error of 2 cm in example 1, the pump flow result is only deviated a maximum of 0.085 m³/s, from 2 m³/s to 2.085 m³/s. We have also repeated the third example considering an error of 2 cm. The pump flow trajectories calculated by CSI with error and without error are similar to the real pump flow trajectory, and the water volume error is not significant respect the real extracted water volume (6.5 %).

We have also calculated the hydrodynamic canal state at two time instant, 2700 seconds and 3900 seconds at the first example. The accuracy of CSI in calculating the extracted flow vector is essential for calculating with accuracy the hydrodynamic canal state with CSI. The CSI calculates the hydrodynamic canal state from the scheduled demands, gate trajectories, initial conditions and the extracted flow vector. The hydrodynamic canal state obtained by a computer model using the real disturbances and the result obtained using CSI (previously computing the disturbance) is very similar.

The algorithm is as good as the results show; the algorithm works quite well with small disturbances and also in case that the flow rate pumping is not located in a control structure.

7.10. APPENDIX I:

Table 7.8: Water velocity obtained by the computer model (real disturbance) V.S. Water velocity obtained by CSI along the canal in the first example at 2700 s and 3900 s.							
Time step (s)	Distance (m)	Water velocity by CSI (m/s)	Water velocity by the model (m/s)	Time step (s)	Distance (m)	Water velocity by CSI (m/s)	Water velocity by the model (m/s)
2.700	0	1,245	1,245	3900	0	1,245	1,245
2.700	20	1,245	1,245	3901	20	1,245	1,245
2.700	40	1,245	1,245	3902	40	1,245	1,245
2.700	60	1,245	1,245	3903	60	1,245	1,245
2.700	80	1,245	1,245	3904	80	1,245	1,245
2.700	100	1,245	1,245	3905	100	1,245	1,245
2.700	120	1,245	1,245	3906	120	1,245	1,245
2.700	140	1,245	1,245	3907	140	1,245	1,245
2.700	160	1,245	1,245	3908	160	1,245	1,245
2.700	180	1,245	1,245	3909	180	1,245	1,245
2.700	200	1,245	1,245	3910	200	1,245	1,245
2.700	220	1,245	1,245	3911	220	1,245	1,245
2.700	240	1,245	1,245	3912	240	1,245	1,245
2.700	260	1,245	1,245	3913	260	1,245	1,245
2.700	280	1,245	1,245	3914	280	1,245	1,245
2.700	300	1,245	1,245	3915	300	1,245	1,245
2.700	320	1,245	1,245	3916	320	1,245	1,245
2.700	340	1,245	1,245	3917	340	1,245	1,245
2.700	360	1,245	1,245	3918	360	1,245	1,245
2.700	380	1,245	1,245	3919	380	1,245	1,245
2.700	400	1,245	1,245	3920	400	1,245	1,245
2.700	420	1,245	1,245	3921	420	1,245	1,245
2.700	440	1,245	1,245	3922	440	1,245	1,245
2.700	460	1,245	1,245	3923	460	1,245	1,245
2.700	480	1,245	1,245	3924	480	1,245	1,245
2.700	500	1,245	1,245	3925	500	1,245	1,245
2.700	520	1,245	1,245	3926	520	1,245	1,245
2.700	540	1,245	1,245	3927	540	1,245	1,245
2.700	560	1,245	1,245	3928	560	1,246	1,246
2.700	580	1,245	1,245	3929	580	1,246	1,246
2.700	600	1,245	1,245	3930	600	1,246	1,246
2.700	620	1,245	1,245	3931	620	1,246	1,246
2.700	640	1,245	1,245	3932	640	1,246	1,246

Time step (s)	Distance (m)	Water velocity by CSI (m/s)	Water velocity by the model (m/s)	Time step (s)	Distance (m)	Water velocity by CSI (m/s)	Water velocity by the model (m/s)
2.700	660	1,245	1,245	3933	660	1,246	1,246
2.700	680	1,245	1,245	3934	680	1,246	1,246
2.700	700	1,245	1,245	3935	700	1,246	1,246
2.700	720	1,245	1,245	3936	720	1,246	1,246
2.700	740	1,245	1,245	3937	740	1,246	1,246
2.700	760	1,245	1,245	3938	760	1,246	1,246
2.700	780	1,245	1,245	3939	780	1,246	1,246
2.700	800	1,245	1,245	3940	800	1,246	1,246
2.700	820	1,245	1,245	3941	820	1,247	1,247
2.700	840	1,245	1,245	3942	840	1,247	1,247
2.700	860	1,245	1,245	3943	860	1,247	1,247
2.700	880	1,245	1,245	3944	880	1,247	1,247
2.700	900	1,245	1,245	3945	900	1,247	1,247
2.700	920	1,245	1,245	3946	920	1,247	1,247
2.700	940	1,245	1,245	3947	940	1,247	1,247
2.700	960	1,245	1,245	3948	960	1,248	1,248
2.700	980	1,245	1,245	3949	980	1,248	1,248
2.700	1000	1,245	1,245	3950	1000	1,248	1,248
2.700	1020	1,245	1,245	3951	1020	1,248	1,248
2.700	1040	1,245	1,245	3952	1040	1,248	1,248
2.700	1060	1,245	1,245	3953	1060	1,249	1,249
2.700	1080	1,245	1,245	3954	1080	1,249	1,249
2.700	1100	1,245	1,245	3955	1100	1,249	1,249
2.700	1120	1,245	1,245	3956	1120	1,25	1,25
2.700	1140	1,245	1,245	3957	1140	1,25	1,25
2.700	1160	1,245	1,245	3958	1160	1,25	1,25
2.700	1180	1,245	1,245	3959	1180	1,25	1,25
2.700	1200	1,245	1,245	3960	1200	1,251	1,251
2.700	1220	1,245	1,245	3961	1220	1,251	1,251
2.700	1240	1,245	1,245	3962	1240	1,252	1,252
2.700	1260	1,245	1,245	3963	1260	1,252	1,252
2.700	1280	1,245	1,245	3964	1280	1,252	1,252
2.700	1300	1,245	1,245	3965	1300	1,253	1,253
2.700	1320	1,245	1,245	3966	1320	1,253	1,253
2.700	1340	1,245	1,245	3967	1340	1,253	1,253
2.700	1360	1,245	1,245	3968	1360	1,254	1,254
2.700	1380	1,245	1,245	3969	1380	1,254	1,254
2.700	1400	1,245	1,245	3970	1400	1,254	1,254

Time step (s)	Distance (m)	Water velocity by CSI (m/s)	Water velocity by the model (m/s)	Time step (s)	Distance (m)	Water velocity by CSI (m/s)	Water velocity by the model (m/s)
2.700	1420	1,245	1,245	3971	1420	1,255	1,255
2.700	1440	1,245	1,245	3972	1440	1,255	1,255
2.700	1460	1,245	1,245	3973	1460	1,255	1,255
2.700	1480	1,245	1,245	3974	1480	1,256	1,256
2.700	1500	1,245	1,245	3975	1500	1,256	1,256
2.700	1520	1,245	1,245	3976	1520	1,257	1,257
2.700	1540	1,245	1,245	3977	1540	1,257	1,257
2.700	1560	1,245	1,245	3978	1560	1,258	1,258
2.700	1580	1,245	1,245	3979	1580	1,258	1,258
2.700	1600	1,245	1,245	3980	1600	1,259	1,259
2.700	1620	1,245	1,245	3981	1620	1,259	1,259
2.700	1640	1,245	1,245	3982	1640	1,26	1,26
2.700	1660	1,245	1,245	3983	1660	1,26	1,26
2.700	1680	1,245	1,245	3984	1680	1,261	1,261
2.700	1700	1,245	1,245	3985	1700	1,261	1,261
2.700	1720	1,245	1,245	3986	1720	1,262	1,262
2.700	1740	1,246	1,246	3987	1740	1,262	1,262
2.700	1760	1,246	1,246	3988	1760	1,263	1,263
2.700	1780	1,246	1,246	3989	1780	1,263	1,263
2.700	1800	1,246	1,246	3990	1800	1,264	1,264
2.700	1820	1,247	1,247	3991	1820	1,264	1,264
2.700	1840	1,249	1,249	3992	1840	1,265	1,265
2.700	1860	1,251	1,251	3993	1860	1,266	1,266
2.700	1880	1,253	1,253	3994	1880	1,266	1,266
2.700	1900	1,257	1,257	3995	1900	1,267	1,267
2.700	1920	1,262	1,262	3996	1920	1,268	1,268
2.700	1940	1,267	1,267	3997	1940	1,268	1,268
2.700	1960	1,273	1,273	3998	1960	1,269	1,269
2.700	1980	1,278	1,278	3999	1980	1,27	1,27
2.700	2000	1,283	1,283	4000	2000	1,27	1,27
2.700	2020	1,288	1,288	4001	2020	1,271	1,271
2.700	2040	1,293	1,292	4002	2040	1,272	1,272
2.700	2060	1,297	1,297	4003	2060	1,272	1,272
2.700	2080	1,302	1,302	4004	2080	1,273	1,273
2.700	2100	1,308	1,308	4005	2100	1,274	1,274
2.700	2120	1,314	1,314	4006	2120	1,275	1,275
2.700	2140	1,321	1,321	4007	2140	1,276	1,276
2.700	2160	1,329	1,329	4008	2160	1,276	1,276

Time step (s)	Distance (m)	Water velocity by CSI (m/s)	Water velocity by the model (m/s)	Time step (s)	Distance (m)	Water velocity by CSI (m/s)	Water velocity by the model (m/s)
2.700	2180	1,338	1,337	4009	2180	1,277	1,277
2.700	2200	1,347	1,346	4010	2200	1,278	1,278
2.700	2220	1,356	1,356	4011	2220	1,279	1,279
2.700	2240	1,367	1,367	4012	2240	1,28	1,28
2.700	2260	1,379	1,379	4013	2260	1,281	1,281
2.700	2280	1,392	1,392	4014	2280	1,282	1,282
2.700	2300	1,407	1,406	4015	2300	1,283	1,283
2.700	2320	1,423	1,422	4016	2320	1,284	1,284
2.700	2340	1,441	1,44	4017	2340	1,285	1,285
2.700	2360	1,461	1,46	4018	2360	1,286	1,286
2.700	2380	1,484	1,483	4019	2380	1,287	1,287
2.700	2400	1,51	1,509	4020	2400	1,288	1,288
2.700	2420	1,54	1,539	4021	2420	1,289	1,289
2.700	2440	1,575	1,575	4022	2440	1,29	1,29
2.700	2460	1,618	1,617	4023	2460	1,291	1,291
2.700	2480	1,67	1,669	4024	2480	1,293	1,292
2.700	2500	1,736	1,735	4025	2500	1,294	1,294
2.700	2500	0,99	0,99	4026	2500	1,052	1,052
2.700	2520	0,991	0,991	4027	2520	1,052	1,052
2.700	2540	0,991	0,992	4028	2540	1,052	1,052
2.700	2560	0,992	0,992	4029	2560	1,052	1,052
2.700	2580	0,993	0,993	4030	2580	1,052	1,052
2.700	2600	0,994	0,994	4031	2600	1,051	1,051
2.700	2620	0,995	0,995	4032	2620	1,051	1,051
2.700	2640	0,996	0,996	4033	2640	1,051	1,051
2.700	2660	0,996	0,997	4034	2660	1,051	1,051
2.700	2680	0,997	0,997	4035	2680	1,051	1,051
2.700	2700	0,998	0,998	4036	2700	1,05	1,05
2.700	2720	0,999	0,999	4037	2720	1,05	1,05
2.700	2740	1	1	4038	2740	1,05	1,05
2.700	2760	1,001	1,001	4039	2760	1,049	1,049
2.700	2780	1,002	1,002	4040	2780	1,049	1,049
2.700	2800	1,003	1,003	4041	2800	1,049	1,049
2.700	2820	1,004	1,004	4042	2820	1,048	1,048
2.700	2840	1,005	1,005	4043	2840	1,048	1,048
2.700	2860	1,006	1,006	4044	2860	1,048	1,048
2.700	2880	1,007	1,007	4045	2880	1,047	1,047
2.700	2900	1,008	1,008	4046	2900	1,047	1,047

Time step (s)	Distance (m)	Water velocity by CSI (m/s)	Water velocity by the model (m/s)	Time step (s)	Distance (m)	Water velocity by CSI (m/s)	Water velocity by the model (m/s)
2.700	2920	1,009	1,009	4047	2920	1,046	1,046
2.700	2940	1,01	1,01	4048	2940	1,046	1,046
2.700	2960	1,011	1,011	4049	2960	1,045	1,045
2.700	2980	1,012	1,012	4050	2980	1,045	1,045
2.700	3000	1,013	1,013	4051	3000	1,044	1,044
2.700	3020	1,014	1,014	4052	3020	1,043	1,043
2.700	3040	1,015	1,015	4053	3040	1,043	1,043
2.700	3060	1,015	1,015	4054	3060	1,042	1,042
2.700	3080	1,016	1,016	4055	3080	1,042	1,042
2.700	3100	1,017	1,017	4056	3100	1,041	1,041
2.700	3120	1,018	1,018	4057	3120	1,04	1,04
2.700	3140	1,019	1,019	4058	3140	1,039	1,039
2.700	3160	1,02	1,02	4059	3160	1,039	1,039
2.700	3180	1,02	1,02	4060	3180	1,038	1,038
2.700	3200	1,021	1,021	4061	3200	1,037	1,037
2.700	3220	1,022	1,022	4062	3220	1,036	1,036
2.700	3240	1,023	1,023	4063	3240	1,035	1,035
2.700	3260	1,023	1,023	4064	3260	1,034	1,034
2.700	3280	1,024	1,024	4065	3280	1,034	1,034
2.700	3300	1,024	1,024	4066	3300	1,033	1,033
2.700	3320	1,025	1,025	4067	3320	1,032	1,032
2.700	3340	1,025	1,025	4068	3340	1,031	1,03
2.700	3360	1,026	1,026	4069	3360	1,029	1,029
2.700	3380	1,026	1,026	4070	3380	1,028	1,028
2.700	3400	1,026	1,026	4071	3400	1,027	1,027
2.700	3420	1,027	1,027	4072	3420	1,026	1,026
2.700	3440	1,027	1,027	4073	3440	1,025	1,025
2.700	3460	1,027	1,027	4074	3460	1,023	1,023
2.700	3480	1,027	1,027	4075	3480	1,022	1,022
2.700	3500	1,027	1,027	4076	3500	1,021	1,021
2.700	3520	1,027	1,027	4077	3520	1,019	1,019
2.700	3540	1,026	1,026	4078	3540	1,018	1,018
2.700	3560	1,026	1,026	4079	3560	1,016	1,016
2.700	3580	1,026	1,026	4080	3580	1,015	1,015
2.700	3600	1,026	1,026	4081	3600	1,013	1,013
2.700	3620	1,026	1,026	4082	3620	1,011	1,011
2.700	3640	1,025	1,025	4083	3640	1,01	1,01
2.700	3660	1,025	1,025	4084	3660	1,008	1,008

Time step (s)	Distance (m)	Water velocity by CSI (m/s)	Water velocity by the model (m/s)	Time step (s)	Distance (m)	Water velocity by CSI (m/s)	Water velocity by the model (m/s)
2.700	3680	1,025	1,025	4085	3680	1,006	1,006
2.700	3700	1,024	1,024	4086	3700	1,004	1,004
2.700	3720	1,023	1,023	4087	3720	1,002	1,002
2.700	3740	1,022	1,022	4088	3740	1	1
2.700	3760	1,02	1,02	4089	3760	0,997	0,997
2.700	3780	1,018	1,018	4090	3780	0,995	0,995
2.700	3800	1,016	1,016	4091	3800	0,993	0,993
2.700	3820	1,014	1,014	4092	3820	0,99	0,99
2.700	3840	1,012	1,012	4093	3840	0,987	0,987
2.700	3860	1,01	1,01	4094	3860	0,985	0,985
2.700	3880	1,008	1,008	4095	3880	0,982	0,982
2.700	3900	1,005	1,005	4096	3900	0,979	0,979
2.700	3920	1,002	1,002	4097	3920	0,976	0,976
2.700	3940	1	1	4098	3940	0,973	0,972
2.700	3960	0,997	0,997	4099	3960	0,969	0,969
2.700	3980	0,994	0,994	4100	3980	0,966	0,966
2.700	4000	0,99	0,99	4101	4000	0,962	0,962
2.700	4020	0,987	0,987	4102	4020	0,958	0,958
2.700	4040	0,983	0,983	4103	4040	0,954	0,954
2.700	4060	0,979	0,979	4104	4060	0,95	0,95
2.700	4080	0,975	0,975	4105	4080	0,946	0,946
2.700	4100	0,971	0,971	4106	4100	0,942	0,942
2.700	4120	0,966	0,966	4107	4120	0,937	0,937
2.700	4140	0,962	0,962	4108	4140	0,933	0,932
2.700	4160	0,957	0,957	4109	4160	0,928	0,928
2.700	4180	0,952	0,952	4110	4180	0,923	0,923
2.700	4200	0,947	0,947	4111	4200	0,918	0,918
2.700	4220	0,941	0,941	4112	4220	0,912	0,912
2.700	4240	0,936	0,936	4113	4240	0,907	0,907
2.700	4260	0,93	0,93	4114	4260	0,901	0,901
2.700	4280	0,924	0,924	4115	4280	0,895	0,895
2.700	4300	0,917	0,917	4116	4300	0,889	0,889
2.700	4320	0,911	0,911	4117	4320	0,883	0,883
2.700	4340	0,904	0,904	4118	4340	0,877	0,877
2.700	4360	0,897	0,897	4119	4360	0,871	0,871
2.700	4380	0,89	0,89	4120	4380	0,864	0,864
2.700	4400	0,883	0,883	4121	4400	0,858	0,858
2.700	4420	0,875	0,875	4122	4420	0,851	0,851

Time step (s)	Distance (m)	Water velocity by CSI (m/s)	Water velocity by the model (m/s)	Time step (s)	Distance (m)	Water velocity by CSI (m/s)	Water velocity by the model (m/s)
2.700	4440	0,868	0,868	4123	4440	0,844	0,844
2.700	4460	0,86	0,86	4124	4460	0,837	0,837
2.700	4480	0,852	0,852	4125	4480	0,83	0,83
2.700	4500	0,844	0,844	4126	4500	0,823	0,823
2.700	4520	0,836	0,836	4127	4520	0,816	0,816
2.700	4540	0,828	0,828	4128	4540	0,808	0,808
2.700	4560	0,819	0,819	4129	4560	0,801	0,801
2.700	4580	0,811	0,811	4130	4580	0,793	0,794
2.700	4600	0,802	0,802	4131	4600	0,786	0,786
2.700	4620	0,793	0,793	4132	4620	0,778	0,779
2.700	4640	0,785	0,785	4133	4640	0,771	0,771
2.700	4660	0,776	0,776	4134	4660	0,763	0,763
2.700	4680	0,767	0,767	4135	4680	0,755	0,756
2.700	4700	0,758	0,758	4136	4700	0,748	0,748
2.700	4720	0,749	0,749	4137	4720	0,74	0,74
2.700	4740	0,74	0,74	4138	4740	0,732	0,733
2.700	4760	0,731	0,731	4139	4760	0,724	0,725
2.700	4780	0,722	0,722	4140	4780	0,717	0,717
2.700	4800	0,713	0,713	4141	4800	0,709	0,709
2.700	4820	0,704	0,704	4142	4820	0,701	0,702
2.700	4840	0,695	0,695	4143	4840	0,694	0,694
2.700	4860	0,686	0,686	4144	4860	0,686	0,687
2.700	4880	0,677	0,677	4145	4880	0,679	0,679
2.700	4900	0,668	0,668	4146	4900	0,671	0,672
2.700	4920	0,659	0,659	4147	4920	0,664	0,664
2.700	4940	0,651	0,651	4148	4940	0,657	0,657
2.700	4960	0,642	0,642	4149	4960	0,649	0,65
2.700	4980	0,634	0,634	4150	4980	0,642	0,643
2.700	5000	0,625	0,625	4151	5000	0,635	0,636

Chapter 8

CSI tests on the experimental canal PAC-UPC

In this chapter, we have tested the CSI algorithm in a real canal. We want to prove that CSI is able to calculate with accuracy a disturbance introduced in the canal, and in this way, some tests were done in a laboratory canal to check the good results obtained by CSI in chapter 7.

Canal PAC-UPC is a laboratory canal specially designed to develop basic and applied research in irrigation canal control area and in all subjacent areas like irrigation canal instrumentation, irrigation canal modelling, water measurements, etc.

The canal is located in the Laboratory of Physical Models inside the North Campus of the UPC. This laboratory occupies a 2000m² surface area.

Most of the design, acquisition and final implementation of the instrumentation and of the motorization of the canal were responsibility of Carlos Alberto Sepúlveda Toepfer, and all this work was reflected in his thesis, see Sepúlveda (2007).

8.1. General description

The original idea was to build a canal that could exhibit notorious transport delays in order to resemble real irrigation canal control problems. For this reason, it was decided to construct a zero slope rectangular cross-section canal, enough long and due to the lack of space inside the laboratory, the canal was designed with a serpentine shape to achieve the maximum canal length in the available space. With this particular design, the result was a 220m long, 44 cm width and 1m depth canal. A detailed scheme of this laboratory canal is presented in Figure 8.1 and Figure 8.2.

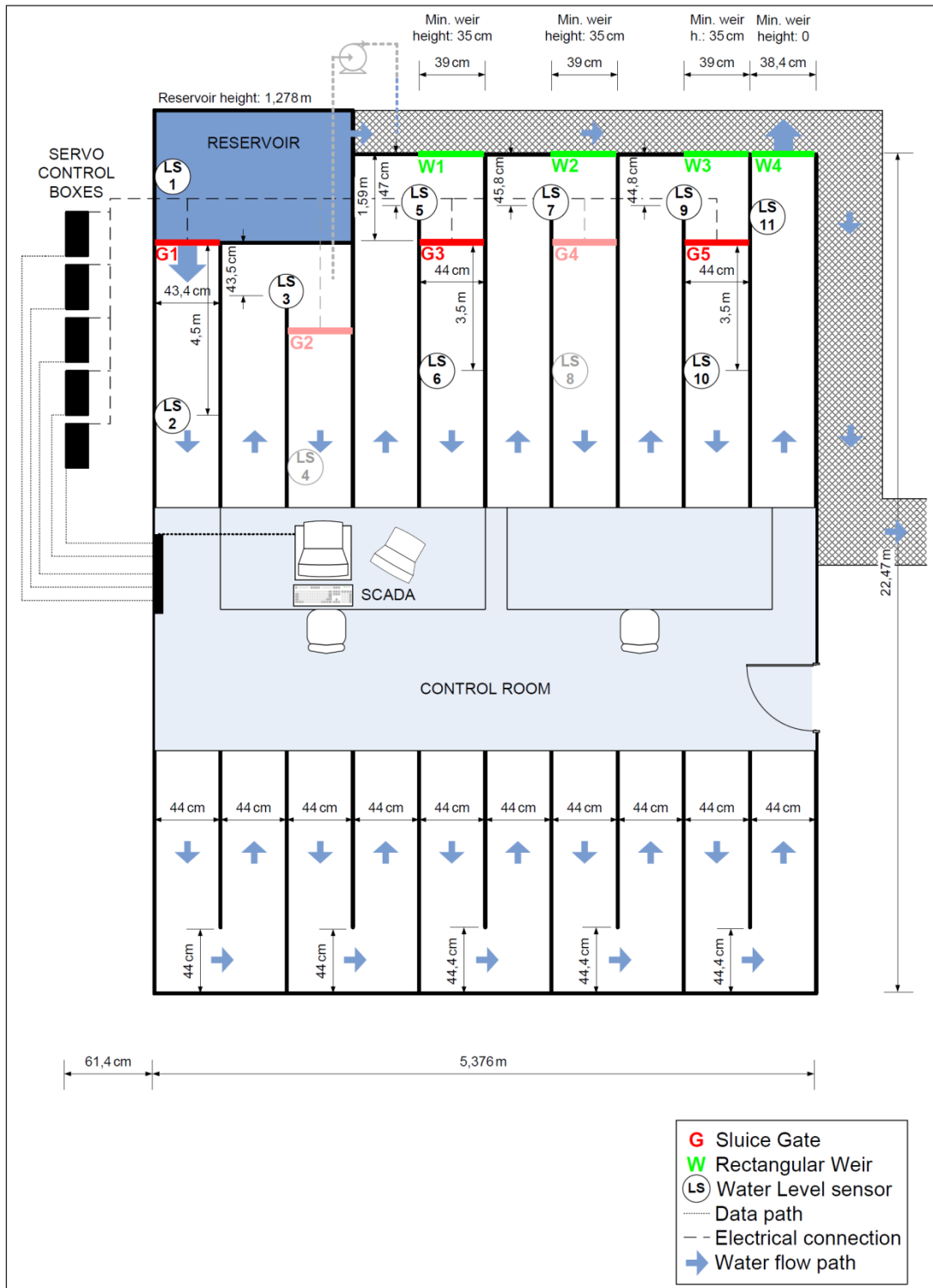


Figure 8.1: Schematic diagram of the top view of the Canal PAC-UPC.

The canal is supplied by a small reservoir at its head. The objective of this element is to provide the canal with enough and virtually unlimited water.

The canal takes water from this reservoir through Gate1 (G1), which is normally under submerged conditions. This gate can regulate the canal inflow by adjusting the gate opening. The water that is not used is recirculated to the laboratory pumping system.

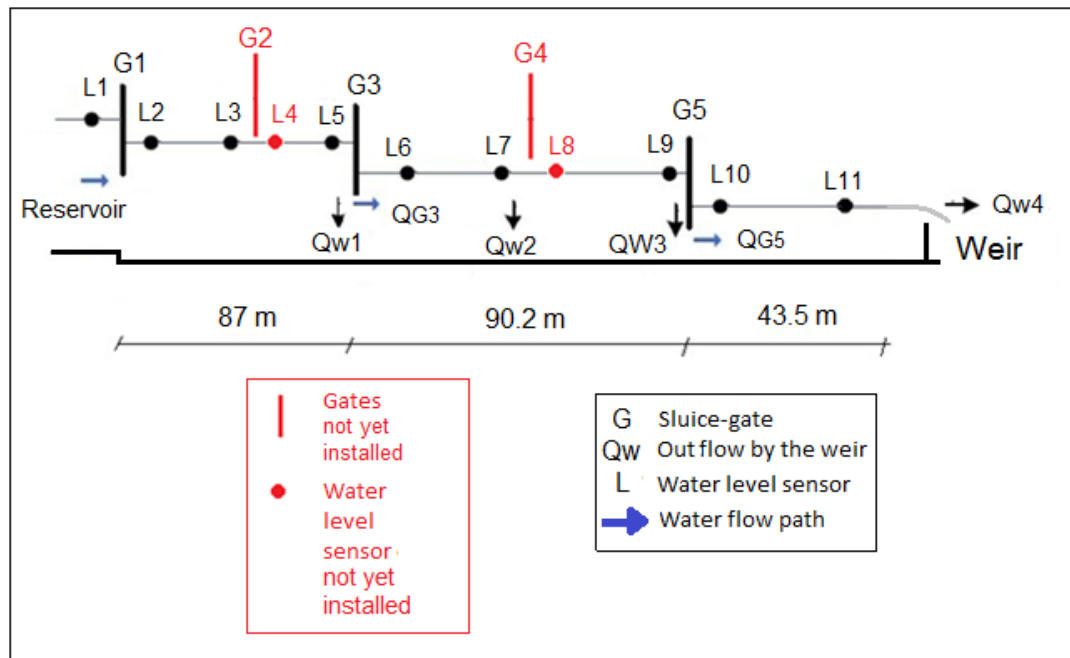


Figure 8.2: Schematic diagram of the profile view of the Canal PAC-UPC.

As illustrated by Figure 8.1 and Figure 8.2, the canal has currently:

- 1 head reservoir
- 3 vertical sluice gates
- 4 rectangular weirs
- 9 water level sensors
- 1 control room

In the present condition, it is possible to arrange the canal with several pool configurations, i.e. a canal with only one very long pool, a canal with one long pool and one short pool, etc.

In the Figure 8.3 , we show a general view of the canal.



Figure 8.3: General view of the Canal PAC-UPC.

Test geometry canal

In this test the geometric configuration of the canal is the following: the sluice gate 1 is the only sluice gate operating in this test because the gates 3 and 5 are out of the water. Only the weirs W2 and W4 are operating, and the algorithm only uses the water level measured at the sensors L1, L6, L10 and L11, although we will also use the data of the sensor L7 to check some results. The canal geometry configuration used in CSI for the canal is shown in the next Table 8.1 and the Figure 8.4:

Pool	Numerical node upstream	Numerical node downstream	Pool length (m)	Canal Depth (m)	Manning's coefficient (n)	Width (m)
1 (from G1 to G2)	1	43	42	1,00	0,016	0,44
2 (from G2 to G3)	44	89	45	1,00	0,016	0,44
3 (from G3 to G4)	90	135	45	1,00	0,016	0,44
4 (from G4 to G5)	136	181	45	1,00	0,015	0,44
5 (from G5 to W4)	182	225	43	1,00	0,016	0,44

Table 8.1: Features of the canal pools.

We could not use all the sensors because some of them were out of order and others sensors were located very close to the canal bend. In every bend of the canal, the water surface has some variations of more or less 1 cm due to the local flow pattern.

CSI is very sensitive to the water level variation at the checkpoints; even more if these water level variations are equivalent to the water level changes introduced to the system by the unknown discharge. In that case, CSI cannot recognize if the water level changes at the checkpoints are introduced by the unknown discharge or otherwise. In this circumstance, CSI would calculate an extracted flow change considering the absolute value of the water level changes, and therefore the extracted flow simulated by CSI would be different to the real unknown discharge introduced into the canal.

We can filter the signal, but if the water level changes by the unknown disturbance is similar to the noise generated by the canal bend, it would be impossible to distinguish between them.

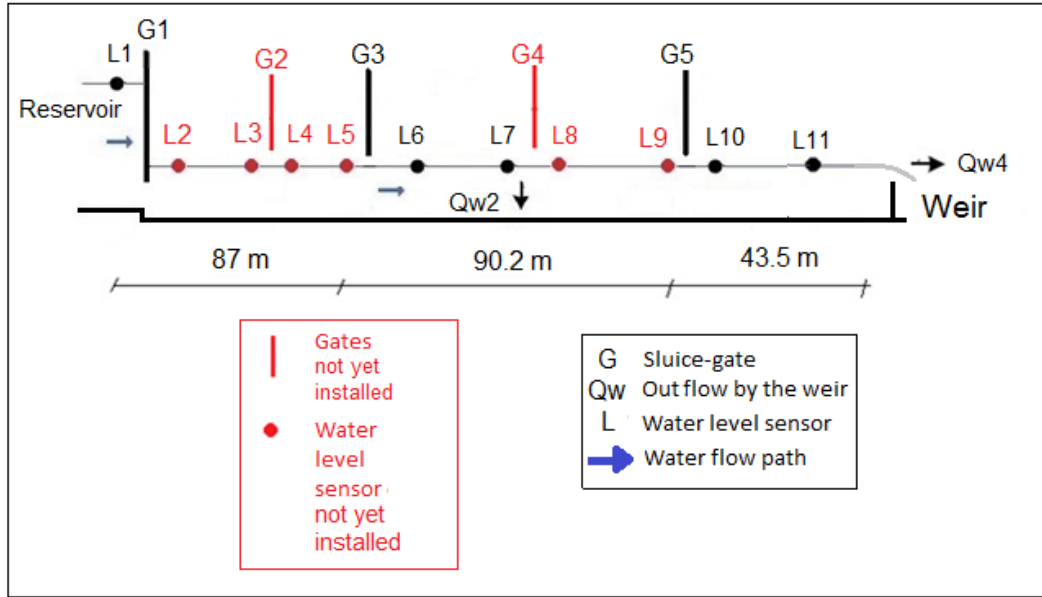


Figure 8.4: Profile view for the geometric configuration of the Canal PAC-UPC in the test.

CSI does not simulate the weirs of the canal although it could be possible, that is, the CSI only obtains the outflows. The advantage of working with the flow is that the control action variable is directly the flow rate and the algorithm does not need any coefficient, as the discharge coefficient. We tested the CSI algorithm in several examples and the results were not good. We realized that we had to modify the specific nodes which represent the control structure to simulate correctly the flow in the canal, because the control structures coincide with the bends of the canal (Figure 8.5) and we had to introduce a loss coefficient to simulate the energy losses in the canal bend. For this reason, a new kind of node was built for this test, and the equations involved in this specific node are defined as:

$$\left. \begin{aligned} A(y_{R1}) \times v_{R1} - A(y_{R2}) \times v_{R2} - q(K) &= 0 \\ y_{R1} + \frac{v_{R1}^2}{2g} - y_{R2} - \frac{v_{R2}^2}{2g} - Loss1 - Loss2 &= 0 \end{aligned} \right\} \quad (8.1)$$

$$\begin{aligned}
 Loss1 &= c_T \times \frac{|v_{R1}^2 - v_{R2}^2|}{2g} \\
 Loss2 &= L_T \times \frac{[S_f(b_1, y_{R1}, v_{R1}) + S_f(b_2, y_{R2}, v_{R2})]}{2} \\
 S_f(b, y, v) &= n^2 \frac{v^2}{\left[\frac{by}{b+2y}\right]^{\frac{4}{3}}}
 \end{aligned} \tag{8.2}$$

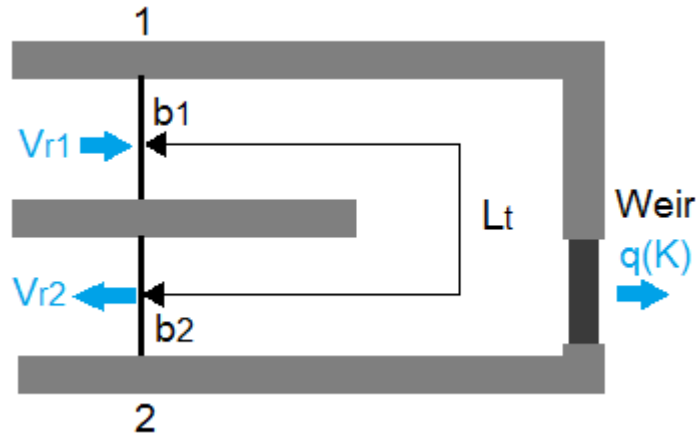


Figure 8.5: Schematic diagram of the top view of a canal bend of the Canal PAC-UPC.

where y_{R1} , v_{R1} , y_{R2} y v_{R2} are the water level and velocity upstream and downstream of the bend, respectively and the transition length is 4 m. The bottom widths are b_1 and b_2 (upstream and downstream of the bend, respectively), in our case $b_1 = b_2$. The coefficient of energy lost in transitions is c_T . n is the Manning coefficient, S_f is the Manning's equation and $q(K)$ is the water withdrawals or extracted flow during the period K .

The extracted flow $q(K)$ is calculated by CSI in each regulation period and every term is compiled in the extracted flow vector.

We calibrated the physical parameter (c_T) doing several test. We simulated the water profile of the canal with a computer model from the initial conditions introduced in section 8.4, so we did these tests in the real canal in the same flow conditions and we

took water level measurements in several sections of the canal. We adjusted the physical parameter of the model until we obtain the same water levels in these sections of the canal in both cases (simulated and real).

The physical parameter adjusted in each canal bend are shown in Table 8.2:

Specific loss node	Upstream pool	Downstream pool	c_T
1 (from L3 to L4)	1	2	0,2
2 (from L5 to L6)	2	3	0,2
3 (from L7 to L8)	3	4	0,2
4 (from L9 to L10)	4	5	0,2

Table 8.2: Physical parameters on each canal bend.

8.1.1. Gates

Canal has vertical sluice gates. They are made of methacrylate reinforced with a metal skeleton in order to provide enough stiffness and a low weight. The vertical movement of the gates is guided by metal frameworks embedded in the canal and is executed by three-phase servomotors. These servomotors are located on top of the gates and are commanded by control boxes situated next to the canal. This particular gate motorization enables only constant speed movements of about 1 cm/s. In this test, all gates except the first have been kept out of the water during the whole test.

8.1.2. Weirs

Rectangular weirs are used to extract water in order to emulate the effect of offtake discharges in real irrigation canals. It is possible to obtain the flow rate extracted measuring the water level upstream of the weir. The Figure 8.6 shows a photo of a weir in the canal.



Figure 8.6: Photo of a weir of the canal PAC.

These weirs have a width of approximately 39 cm and were constructed starting from a 35 cm canal height, except for the end weir (Weir 4) that starts from the canal invert. From this minimum height of 35 cm, it is possible to increase the height of the weir by placing PVC pieces in metal rails, one on top of the other. There are pieces of 5 cm, 10 cm, 20 cm and 35 cm. With different combinations, it is possible to cover a broad range of weir heights, and thereby to achieve a broad range of output flow through the weir.

8.1.3. Water level sensors

The canal has currently nine level sensors located in strategic places, upstream and downstream of each sluice gate and close to every rectangular weir (see Figure 8.1/ Figure 8.2 for details). Their mission is twofold: sensors take measurements of water level to supply data to CSI and the measurements upstream of the weir are used to calculate the discharge through the weirs.

8.2. Initial and boundary conditions

We introduced the initial conditions from the water level measurements at the checkpoints, the flow downstream of every pool (Table 8.5) and the position of the gate 1. The upstream boundary condition is determined by the gate 1 and the water level at the reservoir. The gate equation is represented for the next system of equation:

$$A(y_s)v_s = c_G \times a_G \times u \times \sqrt{L_1(t) - y_s} \quad (\text{submerged hydraulic jump})$$

$$A(y_s)v_s = c_G \times a_G \times u \times \sqrt{L_1(t) - Cc \times u} \quad (\text{unsubmerged hydraulic jump}) \quad (8.3)$$

where y_s is the downstream water level of the reservoir and v_s is the velocity in that cross section, $L_1(t)$ is the water level measurement upstream of the reservoir, Cc is the contraction coefficient, C_G is the discharge coefficient which depends on different parameters such as upstream water level, gate opening, contraction coefficient and the flow condition, a_G is the gate width and u is the gate opening.

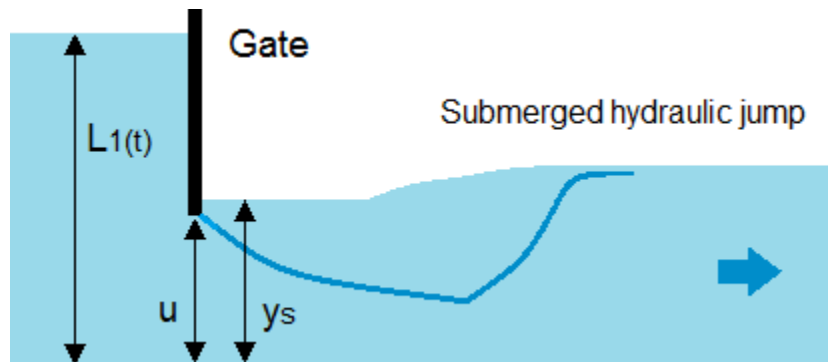


Figure 8.7: A submerged hydraulic jump.

The gate features are shown in the next table:

Gate	Numerical node upstream	Numerical node downstream	Gate discharge coefficient	Contraction coefficient	L_1 water level reservoir	Gate width (m)	Height of the gate opening (m)	Steep (m)
1	0	1	0.68	0.60	1.257	0,434	0,122	0,0

Table 8.3: features of the gate.

Finally, the downstream boundary condition is associated to the weir 4 where the level of water reaches a critical depth. Weir equation is represented by the next discharge equations of a sharp crested weir:

$$A(y_{R1}) \times v_{R1} - c_w \times B \times (H)^{\frac{3}{2}} = 0 \quad (8.4)$$

$$c_w = \frac{2}{3} \times \sqrt{2g} \times c_d$$

where y_{R1} and v_{R1} are the water level and velocity in the sharp crested weir, H is the measured head above the crest, excluding the velocity head, C_d is the discharge coefficient and B is the weir width.

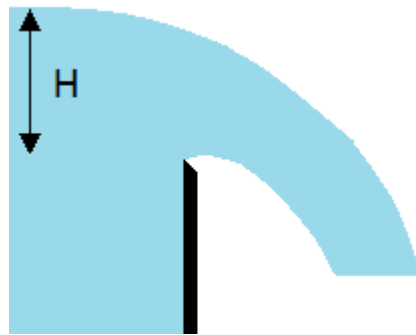


Figure 8.8: A sharp crested weir.

The features of the sharp crested weir are shown in the next table:

Weir	Numerical node	Weir discharge coefficient (C_w)	Weir height (y_w) (m)	Weir width (m)
4	341	0.5776	0,35	0,39

Table 8.4: Features of the weir.

The steady state is the initial condition for the canal. The total flow is 110 l/s and the weir 1, 2 and 3 are not operative. In the Table 8.5, we show the water level measurements which were measured manually and the flow rate at particular points at initial time:

Checkpoint	Initial water level (m)	Flow (m ³ /s)
1 (L4)	0.758	0.110
2 (L6)	0.730	0.110
3 (L8)	0.689	0.110
4 (L10)	0.644	0.110
5 (L11)	0.604	0.110

Table 8.5: Initial conditions (water level at particular points).

The backwater profile for the canal from initial conditions can be seen in Figure 8.9.

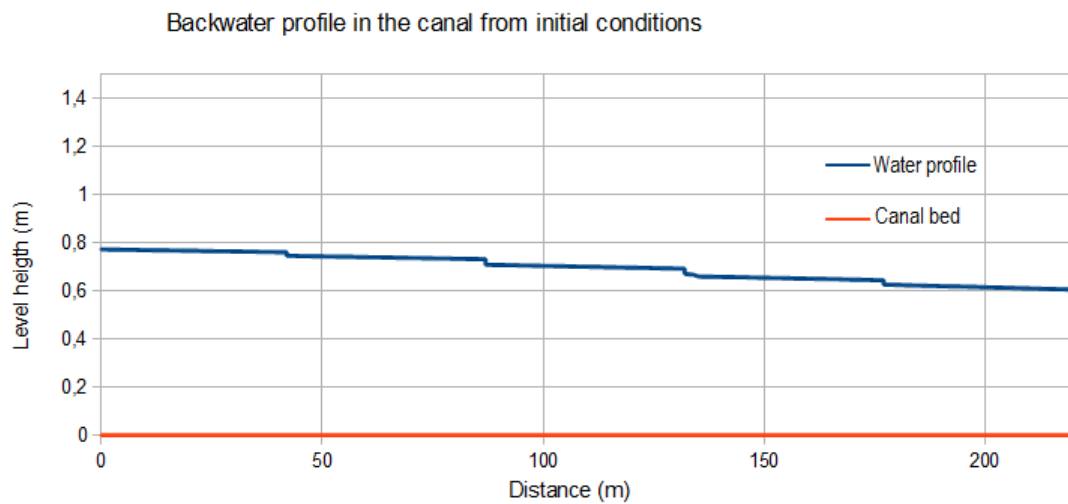


Figure 8.9: Backwater profile for the canal from initial conditions (steady state).

8.3. Discretization of a problem

The past time horizon is defined equal to $T^{KF} = 3410$ seconds = 57 minutes. The regulation period or the time interval between successive control actions for a controller is $\Delta T = 10$ s, so the number of regulation periods during the predictive horizon is $\gamma = 341$. The canal has been spatially discretized (Δx) every meter, so the number of computational cross-sections along the canal is 220.

8.4. Scenario

The canal state has been perturbed by a disturbance with the aid of the weirs. The main objective for CSI is to find the flow trajectory which describes better the flow through the weir.

Several tests were done before and we conclude that, although water level sensors should be close to the crest weir, as the big changes in water levels are associated to these extraction points. The crest weirs are also close to the canal bends, and we have to take into account the local phenomenon introduced by the canal bends which affects the backwater profile. The local phenomenon has a significant influence on the value of the water level measurements, as it causes over-elevations in the backwater profile which is a big problem for CSI. This is the reason why we only use the sensors L6, L10 and L11 in CSI.

As we have introduced in the last chapter, we use a 1D computer model but the flow is clearly 2D in the canal bend. For that reason, we had to introduce in the 1D computer model a new specific node in the location of the canal bend to simulate the energy losses occurred in this short reach.

The output data of the sensors must be accurate as much as possible because the noise associated to the water level measures can invalidate the test.

At the beginning of the test, the flow rate in the canal is 110 l/s. In a particular time (250 s after starting the test), some of the pieces that make up the lateral weir (W2) were removed, so the weir height changes to 55 cm. Later, at time 1950 s, the weir was closed again.

We had not any flow meter in the canal and for this reason the exact value of the flow through the weir 2 was not measured directly. However we could estimate the flow through the weir 2, see Figure 8.11, because we obtained the water level measurements of L7 sensor (Figure 8.10), which is the closest sensor to the weir 2 (W2), and the discharge coefficient of the weir 2 was calibrated in previous works, see Horváth (2013).

We show in the Table 8.6, as we change the weir height to introduce the disturbance in the canal:

Time (s)	G1 Gate opening (m)	W1 Weir height (m)	W2 Weir height (m)	W3 Weir height (m)	W4 Weir height (m)
0	0.122	0.90	0.90	0.90	0.35
250	0.122	0.90	0.55	0.90	0.35
1950	0.122	0.90	0.90	0.90	0.35
3410	0.122	0.90	0.90	0.90	0.35

Table 8.6: The disturbance characteristics introduced to the system.

We can recognize at Figure 8.10, the exactly time when we introduce the disturbance in the canal because the water level at sensor L7 decreases quickly 3 cm.

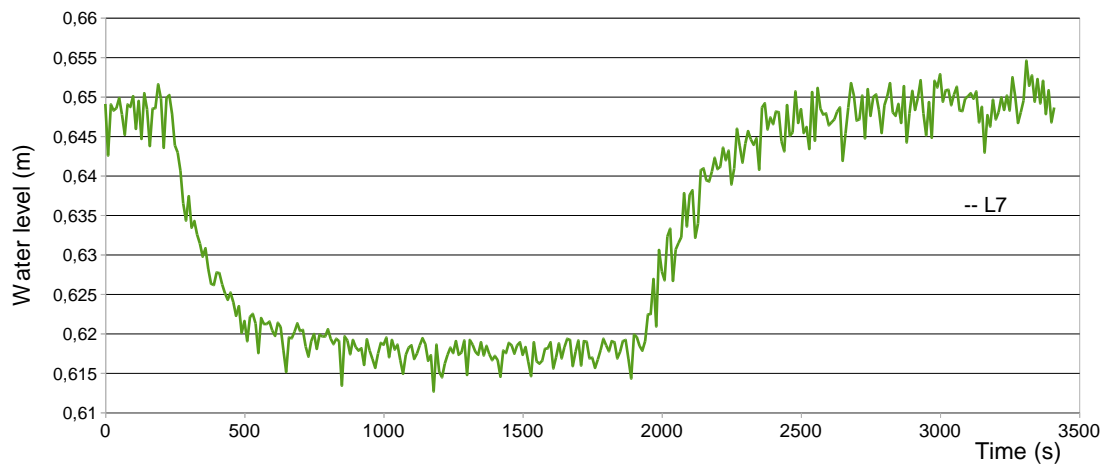


Figure 8.10: Water level measures in sensor L7.

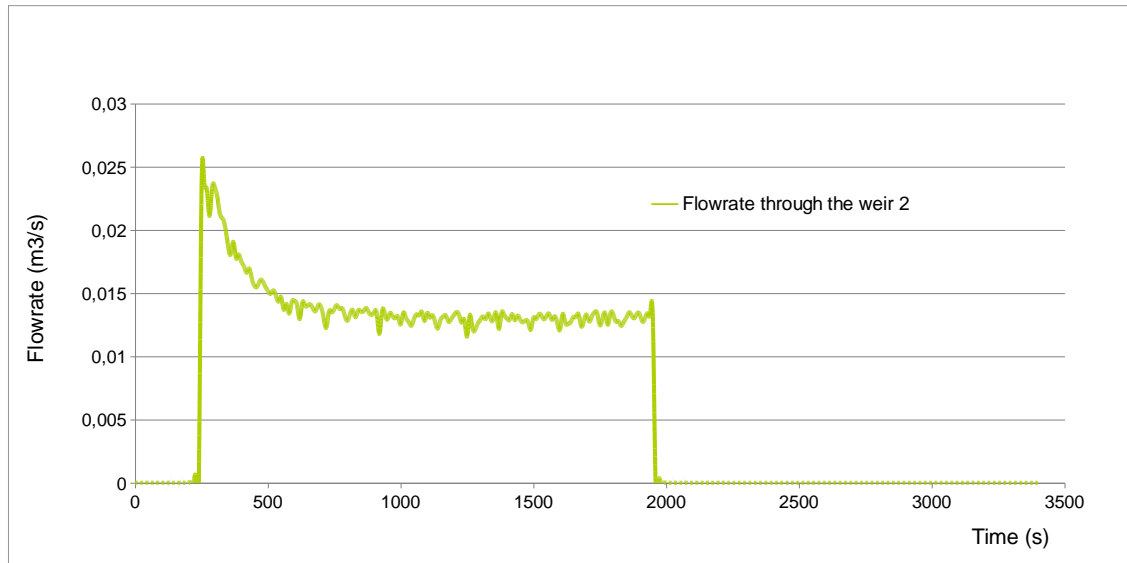


Figure 8.11: Flow through the weir 2 using the weir equation and the discharge coefficient calibrated by Horváth (2013).

Obviously, the flow extracted through the weir modified the water level surface and the flow along the canal. From the sensors L6, L10 and L11 were obtained the water level measurements shown in the next Figure 8.12.

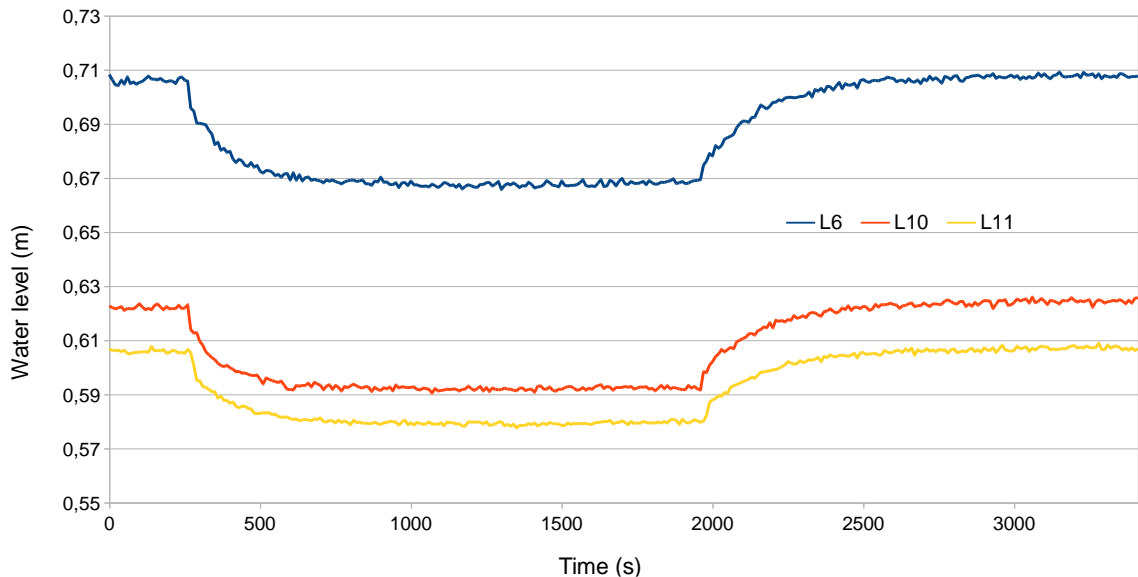


Figure 8.12: Water level measures in sensors L6, L10 y L11. The measures were obtained every second and calculating the average value per 10 seconds.

The water level measurements at sensor L6, L10 and L11 have an error of depth gages of 1/2 mm, as we can check at Figure 8.12, but the water level measurements at sensor L3, L5 or L7 (Figure 8.10) have an error of depth gages of 7/10 mm. These last measurements are not quite accurate for CSI in this test because the water level variation in the canal caused by the flow through the weir 2 is around 2.5-4 cm (Figure 8.12).

The water level measures at the reservoir (L1) are shown in Figure 8.13. This data series collected by the sensor L1 every second are used by CSI for the establishment of the upstream boundary condition. Although the water level at the reservoir should be constant in time, due to limitations of laboratory installations, the water level at the reservoir is variable in time, as it depends on the water level downstream of the gate, for this reason the water level in the reservoir decreases when the flow is extracted through the weir, as it is shown in Figure 8.13.

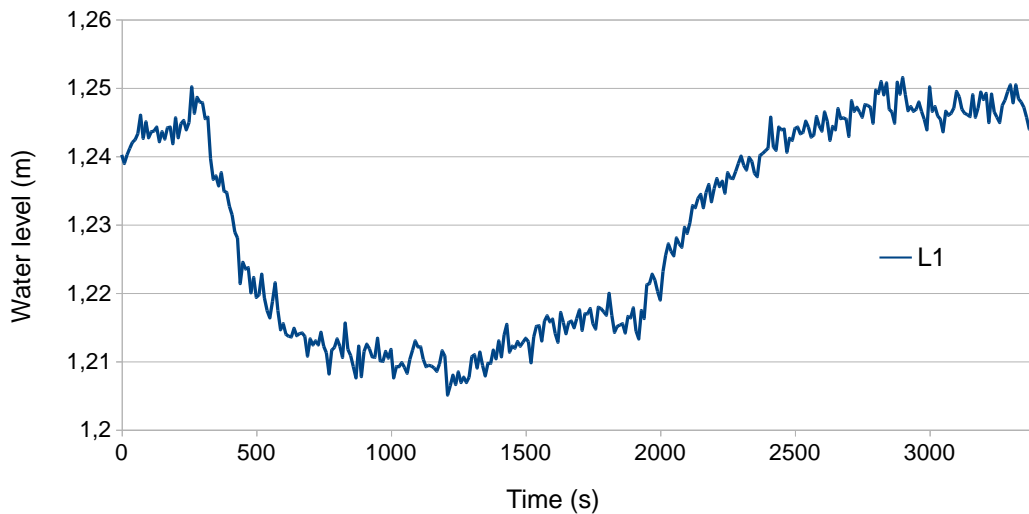


Figure 8.13: Water level measures in sensor L1 (upstream reservoir). The measures were obtained every second and calculating the average value per 10 seconds.

8.5. Test results

The disturbances are introduced to the system by modifying the weir height. The sensors L6, L10, L11 take the water level measurements and these values are introduced to CSI algorithm, which calculates a discharge through the weir that generates a variation in the water levels at the checkpoints equal to the water level measured at the sensors L6, L10 and L11. The extracted hydrograph explains the evolution of the water level measures at the sensors during the past time horizon.

The hydrograph obtained by CSI was filtered using a 10 time steps moving average (SMA), see Figure 8.14. We also show in Figure 8.14, the flow through the weir (W2) obtained with the equation of a sharp crested weir from the water level measures at sensor L7 and the discharge coefficient calibrated by Hórváth (2013).

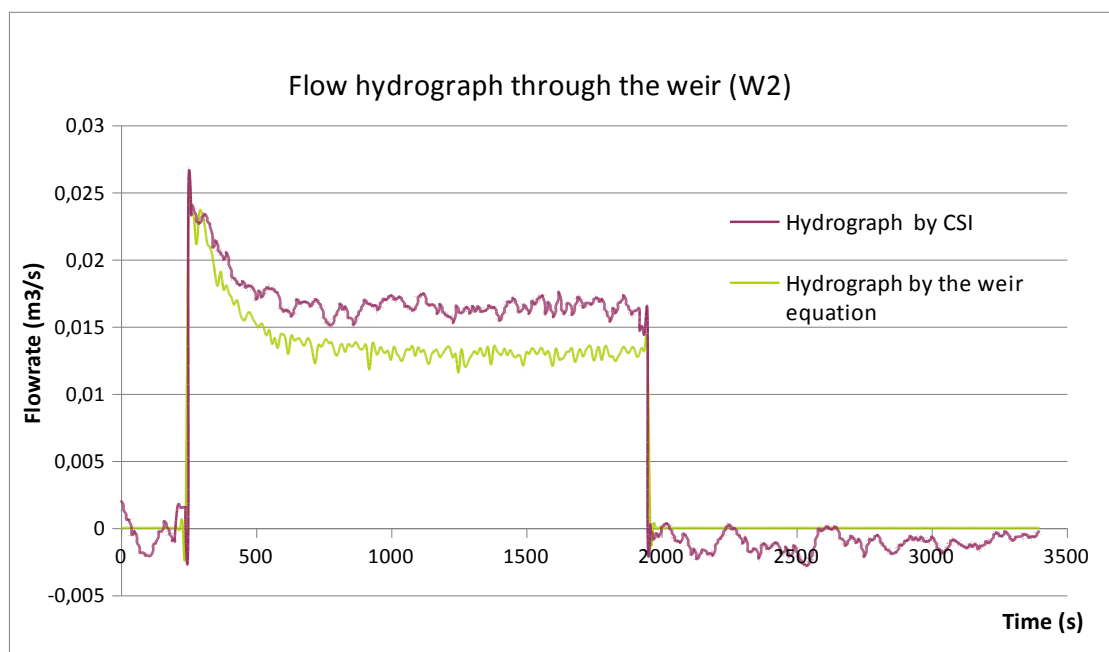


Figure 8.14: The extracted flow rate through the weir 2 by CSI using 10-period simple moving average V.S. the extracted flow rate through the weir 2 by the weir equation using the discharge coefficient calibrated by Horváth (2013).

After analysing the Figure 8.14, we can add:

- The algorithm obtains an extracted hydrograph similar to the real flow extracted through the weir (W2), especially at the initial moment to introduce the disturbance.
- Although the real extracted flow was obtained by the weir equation using not accurate water level measures (sensor L7), these measurements are quite accurate for using this equation. As it is shown at the Figure 8.14, the noise error of this curve is not significant.
- The water level measurements of sensors L6, L10 and L11 were chosen to introduce in CSI because the variations of water level measures are lower than other sensors, as these sensors are far away of the canal bend.
- The flow rate difference between the hydrograph obtained by CSI and the hydrograph using the discharge coefficient calibrated by Hórvarth was around 2.5 l/s.
- Probably, the differences between both hydrographs may be the result of deviations in water level measurements at sensors L6, L7, L10 and L11. But, it is also possible, that these differences are due to a bad calibration of the equivalent Manning coefficient or the energy loss transition coefficient, although we tried to adjust them as good as possible. We have to point that the differences between both hydrographs are not significant, only 2.5 l/s over 110 l/s along the canal.
- The initial conditions are input data for the algorithm, but sometimes there is not any possibility to start the simulating horizon knowing the initial conditions. For that reason, we test this example for different flow initial conditions and the algorithm is sensitive to this initials conditions, which affects the flow rate extracted through the weir calculated by CSI. However if you choose a past time horizon of a reasonably long period, the influence of initial condition on the canal flow disappears. Duration of this period depends on the geometry and the features of the canal.

8.6. Sensitivity of the extracted flow in front of Manning coefficient

This test was performed to study the sensitivity of the flow through the weir versus the Manning roughness coefficient of the canal. With this objective in mind, the same test was done three times more modifying at each one the Manning coefficient increasing the real value in 0.003 and 0.006 and decreasing the real value in 0.003. The solution obtained by the CSI algorithm is shown in Figure 8.15. This type of test was called “tuning test” in Test Cases (Clemmens et. al. 1998).

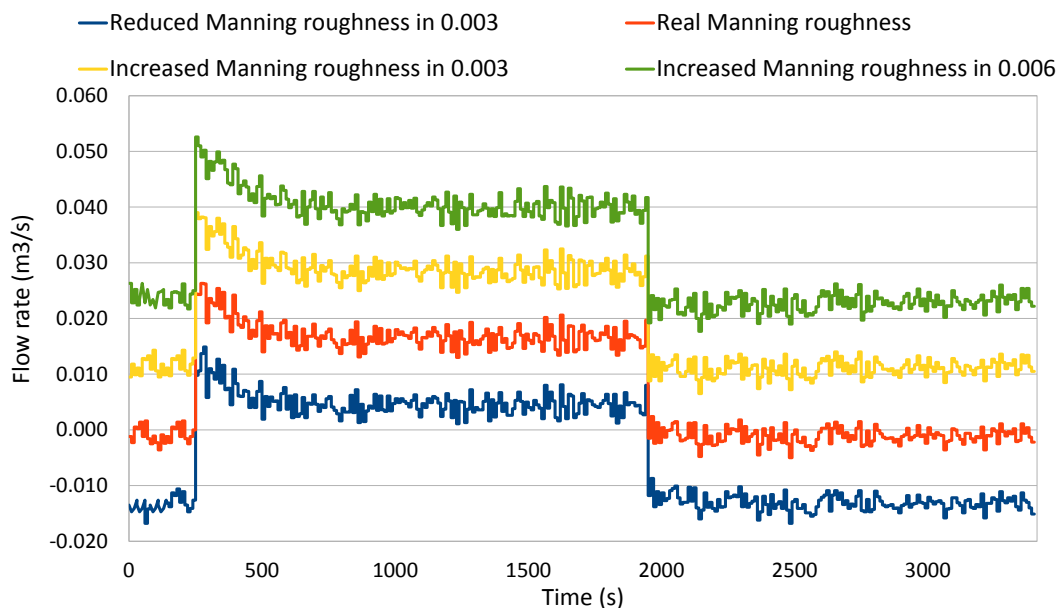


Figure 8.15: The extracted flow rate at weir (W2) modifying the Manning roughness coefficients.

We show in Figure 8.15 how the Manning roughness coefficient has a large effect on the solution obtained by the algorithm, so we have to be very careful with the Manning roughness coefficient value introduced to the CSI algorithm.

The accuracy of Manning coefficient is more important when the flow through the weir is a low flow rate, because a little variation in Manning coefficient can introduce a big variation in flow rates similar to the flow extracted through the weir. There is a linear relationship between the increment of Manning coefficient and the flow extracted through the weir.

8.7. Conclusions

Conclusions are listed below:

- We have obtained a fast convergence and accurate results with CSI which has been possible thanks to the Levenberg-Marquardt method.
- The CSI algorithm is very stable because it calculates the extracted hydrograph more similar to the real hydrograph that has been extracted by the weir. That is, the result is physically feasible and CSI does not look for incoherent solutions which could also define the canal state. For this reason, it is not necessary to introduce constraints to stabilize the algorithm.
- The water levels measurements must be as accurate as possible, as the result (extracted hydrograph) obtained by CSI will be as accurate as the input data (water level measurements) is.
- The CSI algorithm is highly sensitive to some physical parameters as the Manning roughness coefficient and by extension, other parameters as the local energy losses in the canal bend. For this reason, the algorithm should not be used in canals whose physical parameters are not well known.

Chapter 9

ONLINE PREDICTIVE CONTROL: GoRoSoBo

One of the main objectives in this thesis is manage and drive the canal to the desired state during the irrigation periods. There are different types of algorithms able to manage a canal, for instance feedforward algorithms, feedback algorithms or heuristic algorithms.

The feedforward algorithms are used in many canals but they present a particular problem. The feedforward algorithms calculate the optimal gate trajectories taking into account a particular scheduled demand established for an irrigation cycle. The problem is when someone introduces a disturbance and the algorithm fails to manage the canal.

The heuristic algorithms are not based on physical laws, so the system is for them a black box. These kind of algorithms are usually used when large time series of measurements are available and the system is too complex to model. The problem of these algorithms is similar to the feedforward algorithms, when someone introduces a disturbance and the algorithm operates out of the range which was calibrated, the controller fails.

We have developed a feedback algorithm (GoRoSoBo) which gets the measured values and compare them with a reference representing the desired performance. Any deviation from the reference is fed back into the control system so that this reduces the deviation of the controlled quantity from the reference. We introduce GoRoSoBo in this chapter.

9.1. Introduction

To fulfil our objective, we are going to introduce our predictive control which is part of the overall control diagram of an irrigation canal, introduced in Chapter 5. If we take a look to the Figure 9.1, we have introduced in this diagram our developed algorithms.

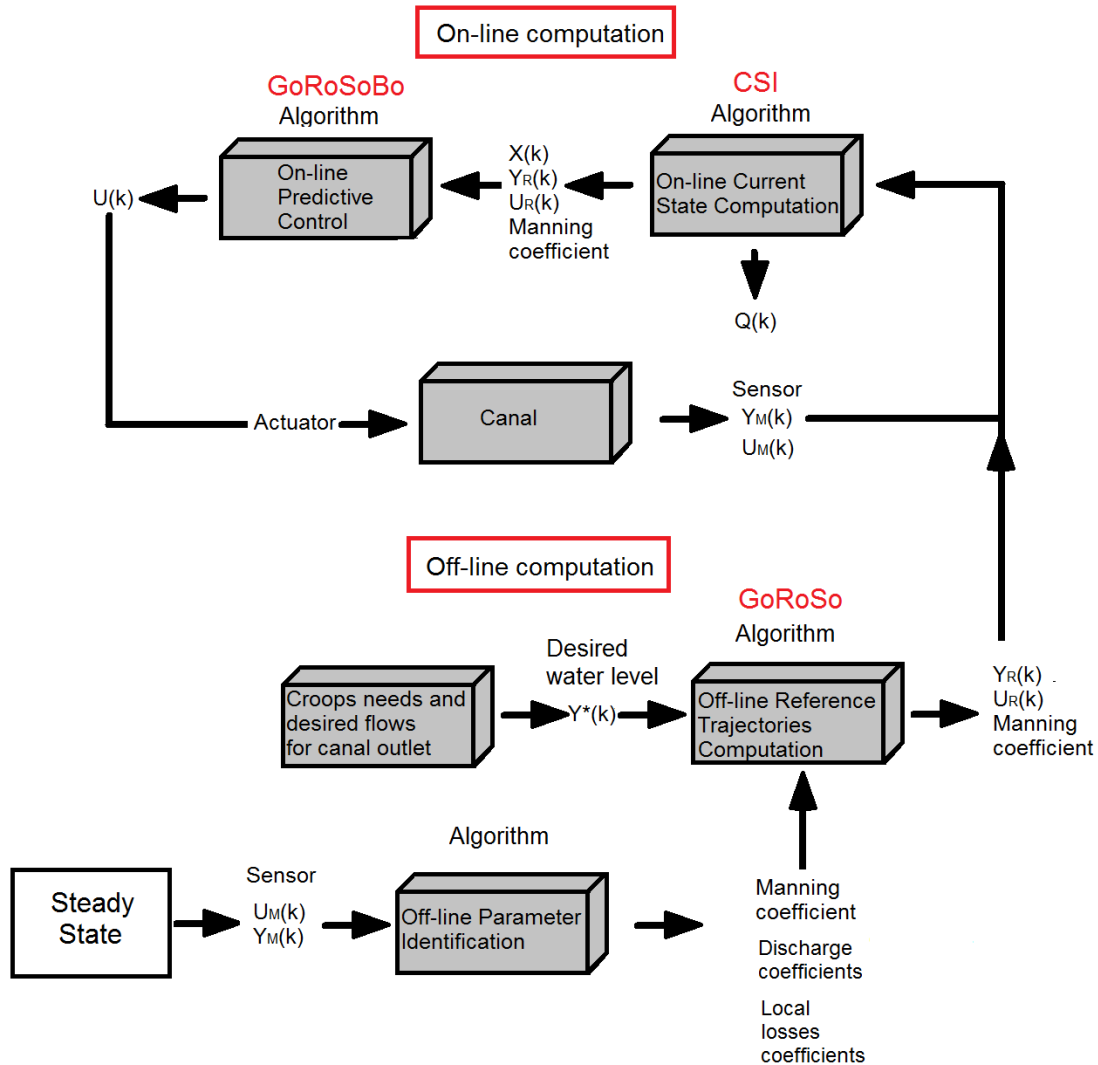


Figure 9.1: Developed algorithms in this thesis in an overall control diagram of irrigation canal. This cycle is done in each operation period and GoRoSoBo calculates the optimum gate trajectories in each operation period for a predictive horizon.

The primary objective for a control algorithm is to maintain the water level at several points of the canal, where the flow is extracted by control structures (lateral weirs, pumps, offtake orifices). Typically, almost all of these structures work by gravity so the flow control depends on the water level. Maintaining the water level at the checkpoints allows controlling the demand deliveries and protects the canal from significant variations of water level which could damage the canal.

The objective of maintaining a constant water level at checkpoints is rarely achieved because the water deliveries along the canal are not always well implemented and

sometimes these deliveries are unknown. The disturbances produce transients in the canal and consequently, changes in water level and discharge in orifice offtakes. The flow rate delivered differs from the scheduled delivery established by the watermaster, and as a consequence the efficiency of irrigation canal decreases. It is important to develop a feedback controller which comes back the canal to the desired water level at the checkpoints as soon as possible.

There are some feedback controllers which works in parallel with computer models and the results obtained are good when the disturbance is small because it only affects at the neighborhood of a section. As these controllers do not know the unscheduled deliveries introduced in the canal, if the disturbance is significant, the data introduced to the feedback controller, obtained previously by the computer model, is not good enough and the controller finally fails.

These controllers do not work correctly when the water level measurements at a checkpoint are far away from the desired water level. The gate trajectories are far away to the real solution because the disturbance has modified the canal state entirely and the controller only takes into account some measured values in several checkpoints in the canal. These controllers only consider the perturbation indirectly, through their effects on the water level measurements of the system.

Instead, GoRoSoBo has not this problem because we developed CSI to estimate these perturbations and establishes the hydrodynamic canal state at every cross-section, as we have seen in chapter 6, 7 and 8. All these data could be sent to the feedback controller (GoRoSoBo) to calculate the optimum gate trajectories taking into account a particular target which is usually to maintain the water level at the checkpoints.

There are other authors as Delgoda et al. (2013) or Van Overloop et al. (2008) which have also developed predictive controllers based in predictions of unscheduled flows.

In an ideal case where there are not disturbances in the canal, a feedforward algorithm could manage the canal, as we show in Figure 9.2. In this case, the gate positions are fixed by a feedforward algorithm taking into account the scheduled demand (a particular desired water level at the orifice offtake), which is $4 \text{ m}^3/\text{s}$ in Figure 9.2, during the irrigation cycle.

Ideal case

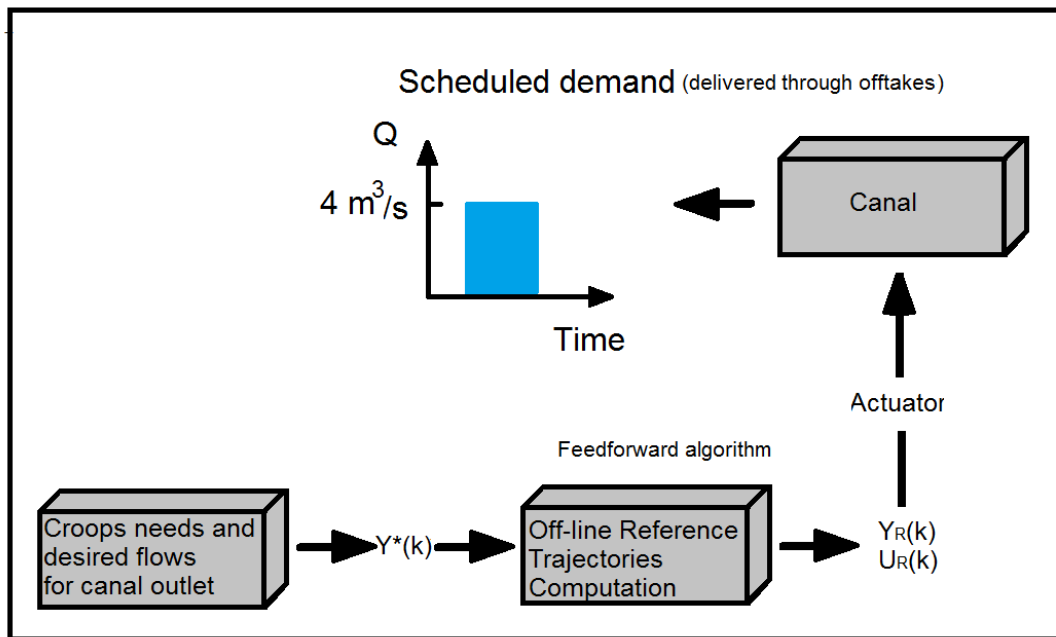


Figure 9.2: Ideal case for a predictive control which maintains a water level to satisfy a scheduled demand of $4 \text{ m}^3/\text{s}$.

The ideal case is quite far from reality frequently, because someone could pump an unknown flow so the water level above the offtakes would decrease and the flow delivered to the farmers too, see Figure 9.3. In this case, someone introduces a disturbance ($2 \text{ m}^3/\text{s}$) in the canal and the water level at the checkpoints decreases. The feedforward algorithm only considers the scheduled demand previously fixed, so the water level at the checkpoints is not recovered and the flow delivered to the farmers is reduced too.

In this particular case we could use our overall control diagram of an irrigation canal. Initially, the canal shows a steady state (Figure 9.4 (a)), but someone introduces a disturbance at $T=40\text{s}$ (Figure 9.4 (b)). The water level measurements are sent first to the CSI which calculates the disturbances and the current canal state, and then GoRoSoBo calculates a new gate trajectory to fulfill the objective (come back to the desired water level at the offtake), (Figure 9.4 (b)). The new gate trajectory is sent to the actuator which modifies the gate position (Figure 9.4 (c)). The canal state comes back to the desired state, (Figure 9.4 (d)).

Real case

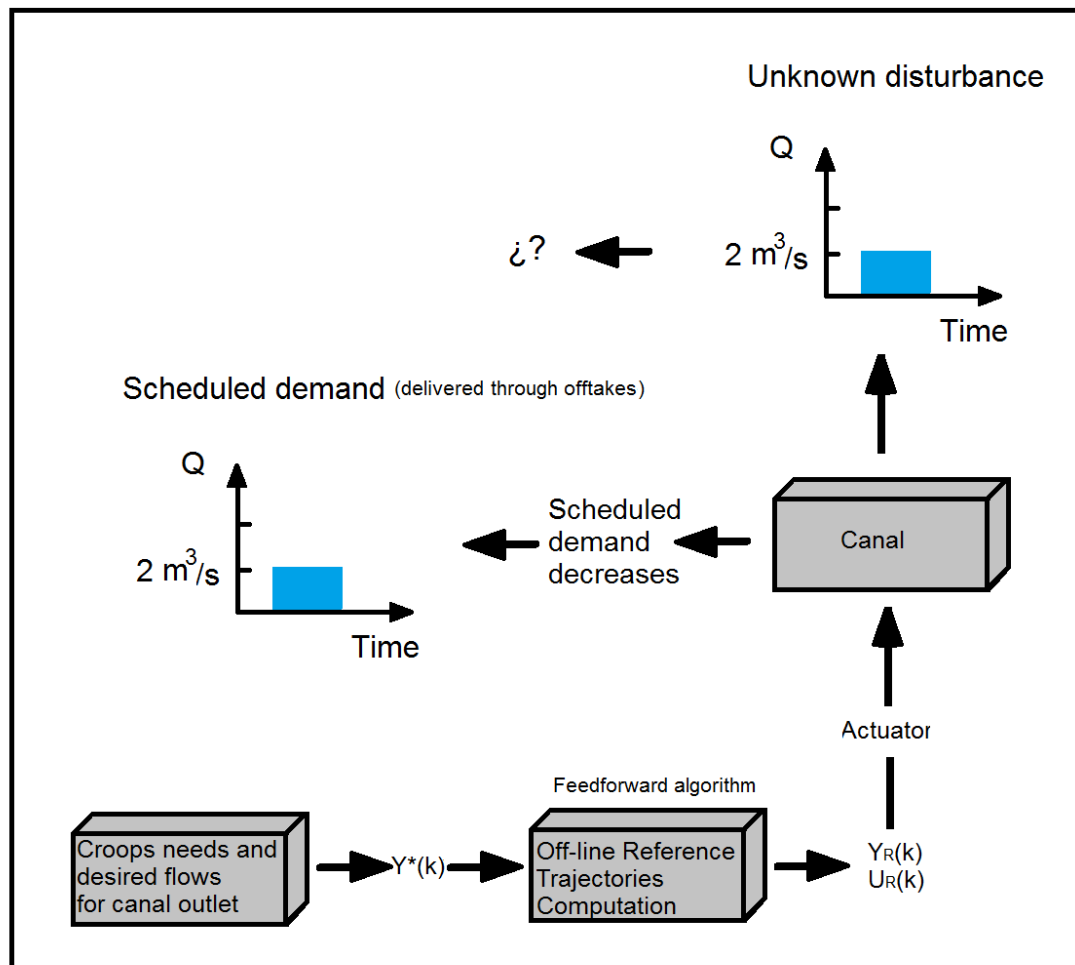


Figure 9.3: Real case for a predictive control. The controller calculates a gate trajectories to maintain a water level and supply the scheduled demand of $4 \text{ m}^3/\text{s}$. The problem is when someone introduce a disturbance of $2 \text{ m}^3/\text{s}$. The discharge through offtakes would be reduced to $2 \text{ m}^3/\text{s}$.

The question is, why do we develop a new feedback algorithm when there are a lot of them in the bibliography review? Because some of them, as CLIS (Liu et al., 1998) or PILOTE (Malaterre et al., 1995), operate quite well and obtain good performance indicators in the Test Cases proposed by ASCE (Clements et. al., 1998) but they do not operate as we want, and other algorithms not achieve our objectives, as we introduce as follow:

- Many feedback algorithms calculate the optimal gate trajectories from empiric relations between the gate trajectory and water level error at checkpoints obtained

in specific canals. But these algorithms are not based on physical laws and the empirical relation only operate in a particular range.

- There are other feedback algorithms which calculate the optimal gate trajectories from relations between the gate trajectories and the water level error in a particular steady state. In case that a small disturbance is introduced in the canal, these algorithms operate quite well. In case that the disturbance is significant, the initial steady state is quite different to the current canal state, so the gate trajectories calculated with these algorithms will not be the optimal trajectories.
- There are feedback control algorithms which need two or more water level and flow measurements for each pool to calculate the gate trajectories.

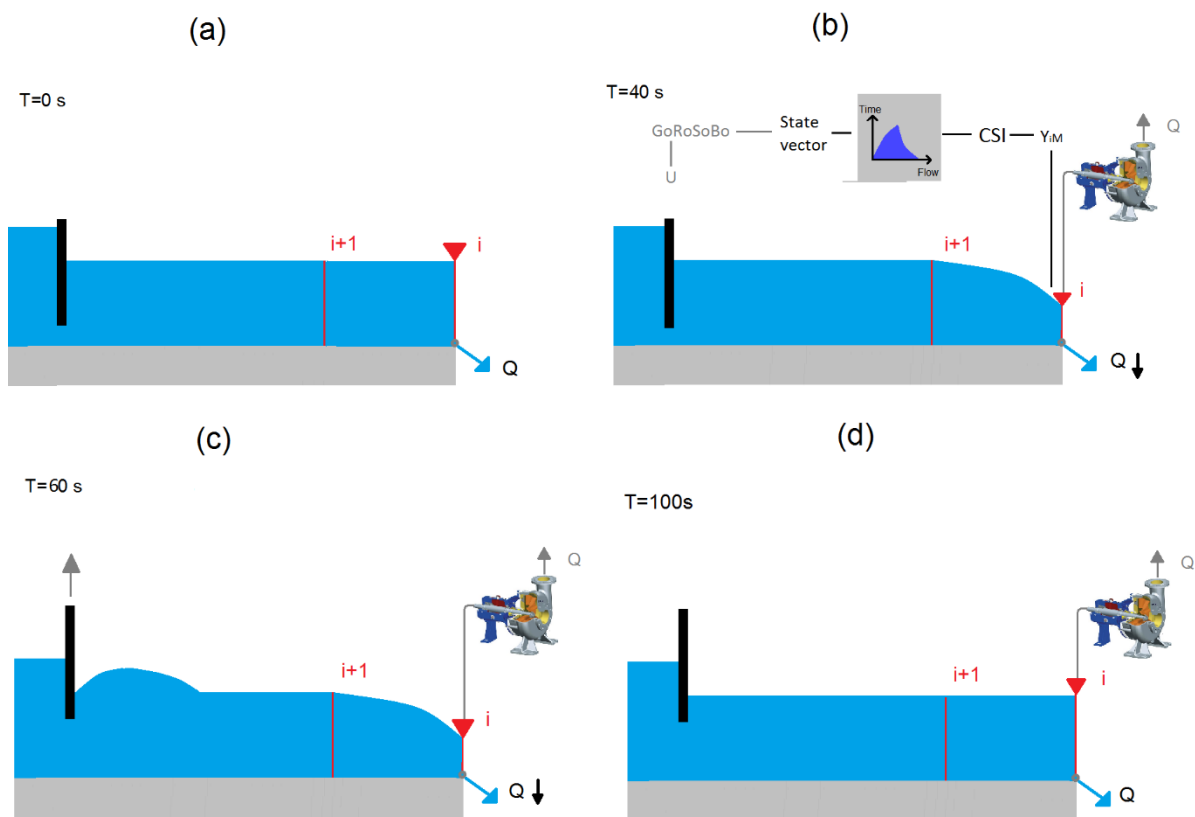


Figure 9.4: Logic of control in our overall control diagram of irrigation canal.

In this thesis, our feedback algorithm (GoRoSoBo) calculates the optimum gate trajectories from a water level error with aid of the HIM (Hydraulic Influence Matrix, Soler (2003)). This matrix establishes the influence of a gate movement on the free surface. Every element of

the HIM matrix establish the variation of the variable X (water level and velocity along the canal) associated to any change in gate positions (9.1). The HIM matrix could be recalculated every time step, so we know the influence of a gate position on the water level at the current canal state.

$$[HIM(U)] = \begin{bmatrix} \frac{\partial X}{\partial U} \end{bmatrix} \quad (9.1)$$

The name of Hydraulic Influence Matrix (HIM) is originated because their hydraulic components represent the influence of gate trajectories over the state vector (see chapter 6) at all sections during the predictive horizon. Based on the full Saint-Venant equations, and using the first derivative $\left(\frac{\partial X}{\partial U}\right)$ on an analytical process, it can be established the changes in flow behavior from a sluice gate movement at any section in a certain time step.

To show a general view of the HIM, we are going to introduce the Figure 9.5 . When we open a gate in a canal, we modify the canal state, the water level and velocity in particular points at a particular time. For instance at $t=10s$, we modify the water level and velocity in cross-sections close to the gate (cross-section i), but we do not modify the hydrodynamic variables in a far cross section (" $i+2$ " or " $i+3$ "). At $t=30s$, the hydrodynamic variables in all cross-sections from " i " to " $i+3$ " have been modified. The HIM matrix establishes how a gate movement modify the hydrodynamic variables in all sections of the canal during a prediction horizon.

The HIM matrix is ill-conditioned, as we introduced in chapter 6, partly explained by the disparity of influence between their elements values, as some of them are zero. The influence of a sluice gate movement on a particular section depends on the distance between them, and the influence decreases after several regulation periods from the gate movement.

For instance in the Figure 9.6, the gate at " i " changes its position (ΔU_i) at $t=0$, so the gate generates a perturbation which travels upstream and downstream of the canal. In this case, the section " $i+3$ " at $t=0s$ is not influenced by the gate movement (ΔU_i), but at $t=30s$ is influenced by the gate. The magnitude of the influence of ΔU_i is not equal in section " i " at $t=0s$ and in section " $i+3$ " at $t=30s$, because the influence of the perturbation decreases in space and time.

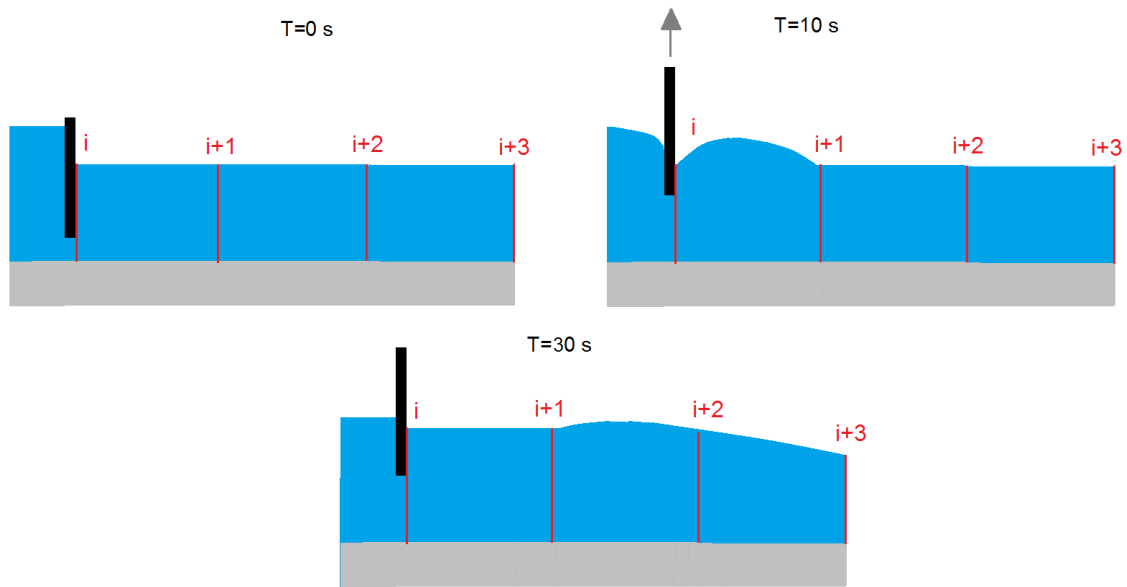


Figure 9.5: General view of the influence of a gate movement in the hydrodynamic state of a canal.

We can choose a set of points of the space/time domain from the canal (Figure 9.5) as many as we want to form a computational grid where we can establish the influence of a gate movement on the water level and velocity in each of these points (Figure 9.6).

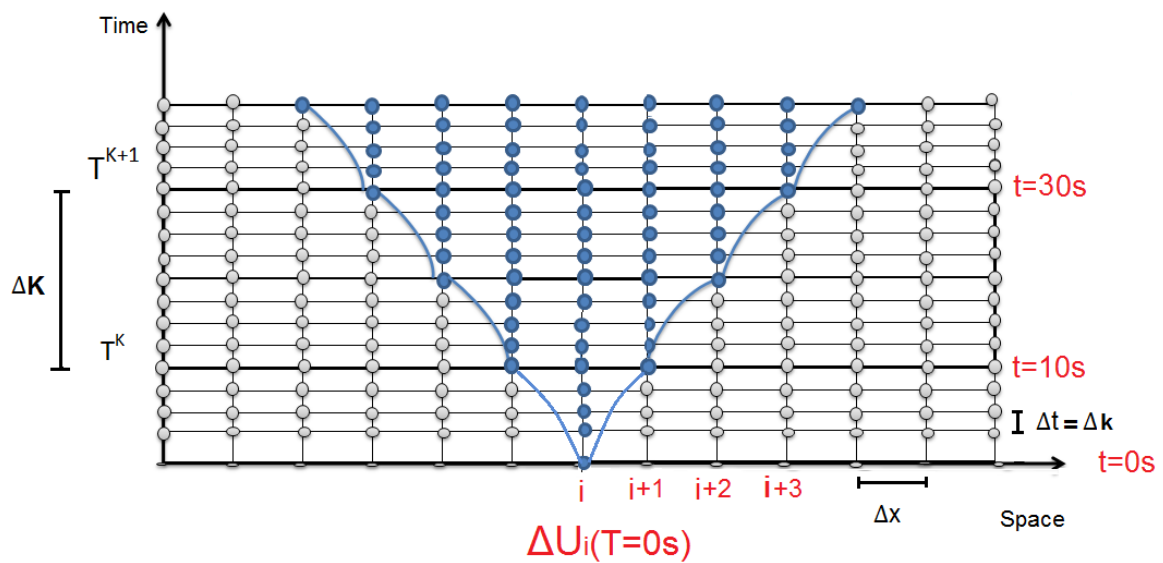


Figure 9.6: Influence of a gate movement in several points of the space/time domain in the canal disposed in a computational grid. Notice that "K" with capital letter denotes time interval of control and "k" with small letter denotes time instant of simulation.

9.2. Problem statement

We could calculate the perturbations associated a particular gate movement in a canal from a computer model based on the Saint-Venant equations. GoRoSoBo does the inverse process, that is, it calculates the gate trajectories that most closely approximate a particular perturbation (water level error at checkpoint).

The GoRoSoBo algorithm is a feedback controller where the desired output (variable with desired setpoint) is used to adjust the water level measurements to drive the system to the desired water demand from control actions (the optimal gate trajectories).

There are multiple forms to obtain the optimal gate trajectories, for instance there are controllers which are based on the inversion of the equations describing the movement of the water, other controllers are based on the optimization of a given criterion for the present and a given time horizon, other controllers are based on the use of the proportional/derivative/integral of the measured error to correct the control action, etc.

Our proposal defines an objective function from the total deviation between the data prescribed by the user (target water level) with the measured data (water level measurement). This objective function is minimized through an optimization procedure with the objective to obtain the optimal gate trajectories. In that sense, GoRoSoBo solves an inverse sequential problem as an optimization procedure.

9.2.1. Definition of the optimization problem

In this optimization problem, we replace the water level measurements with the prediction output vector ($Y_1^{k_F}$) obtained from a computer model (more details in chapter 6). This vector contains the values of water level at certain cross-sections of the canal for a prediction horizon $(1-k_F)$, from particular gate trajectories. The dimension of the prediction output vector is $n_Y = k_F \times n_c$, where k_F is the final time step of the prediction horizon and n_c is the number of checkpoints.

GoRoSoBo calculates the optimal gate trajectories during a prediction horizon. As it is illustrated in Figure 9.7, the gates are operated with a period ΔK . Gate trajectories are assumed as piecewise functions. The gate position remains fixed during the regulation

period and the gate position could change between regulation periods. The gate trajectories vector is defined by lumping together all the gate positions as follows:

$$U = \left[U_1(1), \dots, U_{n_g}(1), \dots, U_1(K_F), \dots, U_{n_g}(K_F) \right]^T \quad (9.2)$$

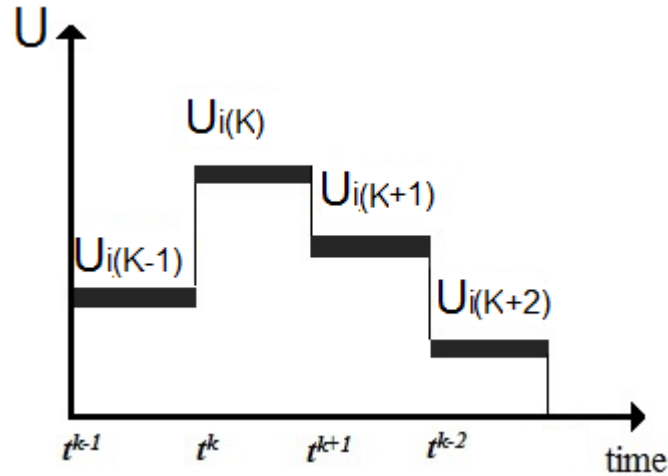


Figure 9.7: Mathematical representation of a gate trajectory.

where the dimension of this vector is $n_U = K_F \times n_g$, n_g is the number of gates and K_F is the final regulation period. In each operation period, GoRoSoBo calculates new gate trajectories for a prediction horizon.

The data prescribed by the user in this optimization problem is the water level at checkpoints, where there are gravity offtakes and the algorithm has to maintain the water level.

We can write the target water level vector at a particular time step k as:

$$y^*(k) = \left[y_1^*(k), \dots, y_i^*(k), \dots, y_{n_c}^*(k) \right]^T \quad (9.3)$$

Finally, all vectors (9.3) for each time instant of the predictive horizon are joined to define the desired water level vector, the dimension of the vector is n_Y . We define this vector as:

$$Y^* = \left[y^*(1), y^*(2), \dots, y^*(k_F - 1), y^*(k_F) \right]^T \quad (9.4)$$

We choose a set of points of the space/time domain to form a computational grid where the solution is computed by using a numerical procedure (Figure 9.8). We can check the desired water level vector values at the computational grid in Figure 9.8 (yellow dots), we can also check the measured water level vector at the computational grid (red circle).

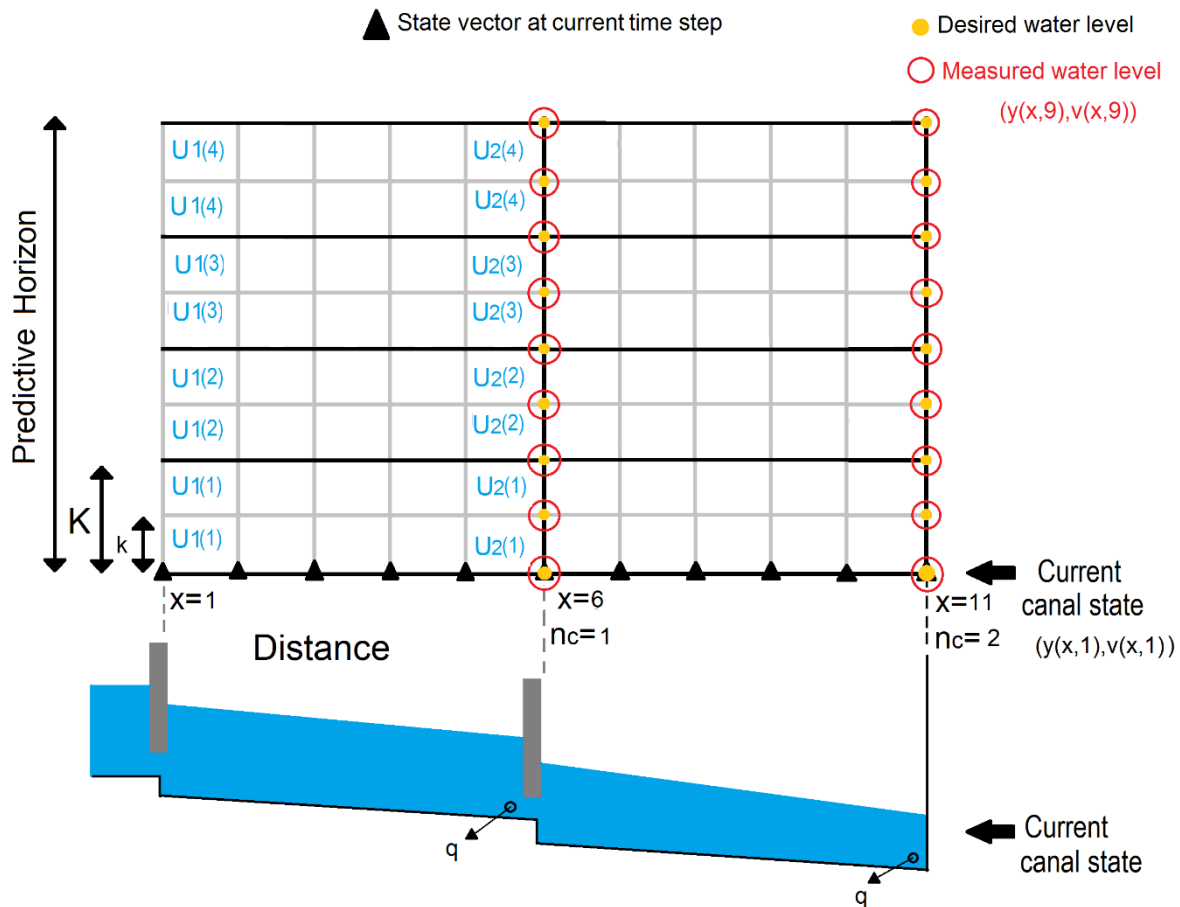


Figure 9.8: Sketch of a numerical grid in a canal with two pools controlled by two checkpoints downstream each pool. There are two sluice gates. Gate trajectories are defined with four operation periods. The x/t -dots where the flow behavior is defined are shown.

9.2.2. Control problem: Objective function

As we introduced in chapter 4, an objective function expresses the aim of the control model to minimize or maximize some variables to obtain an optimal response. An objective function can be the result of an attempt to express a control action (gate movement) in mathematical terms to drive the system to the desired output (variable with desired setpoint).

In that way, we thus define ideal flow behavior as a sequence of desired water level values for given sections (checkpoints) at several time steps. In order to continue with the mathematical treatment, these values are defined by the desired water level vector.

In reality, the sequence of water level measurements could be not close to the desired water level vector due to disturbances, and this variation between both values is defined by the objective function.

The objective function expresses the residual value between the prediction output vector and the desired water level at the target points. Minimizing this function, we obtain a set of control actions (U^*) and the best possible approximation to the desired value (Y^*). This is the goal of the objective function and U^* is the solution.

$$J(U) = \frac{1}{2} (Y_1^{K_F}(U) - Y^*)^T [Q'] (Y_1^{K_F}(U) - Y^*) \quad (9.5)$$

$$\text{Minimum } J(U^*) = F[Y_1^{K_F}(U^*)] \quad (9.6)$$

where Q' matrix is a weighing matrix with dimension $n_y \times n_y$. This matrix could be used to define the level of importance of the water level error in a particular checkpoint weighting some values more than others depending on their importance. We define this matrix as the identity matrix in GoRoSoBo. The vector U contains the gate trajectories, and the dimension of this vector is $n_U = K_F \times n_g$, where $Y_1^{K_F}(U)$ is the water level at the checkpoints from K_1 to K_F due to gate trajectories U .

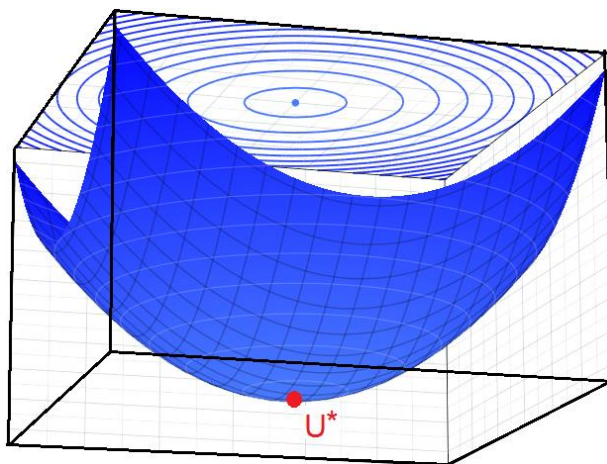


Figure 9.9: Minimum value (U^*) for an objective function.

9.2.3. Predictive control

Once we know the water level/velocity along the canal at time step K with the aid of CSI (Figure 9.10), we could predict the water level/velocity along the canal for a predictive horizon $(k-K+3)$ as we know the canal state and the real extracted flow at K , the gate trajectories and the scheduled demand from K to $K+3$. In that sense, we could obtain the water level error at the checkpoints between target value and the predicted water level, so the predictive control could calculate some actions at K (present time) for the prediction horizon to remove the water level error. At the end of this process, we would obtain a predicted output of the system at K for the prediction horizon, see Martín-Sánchez and Rodellar (1996).

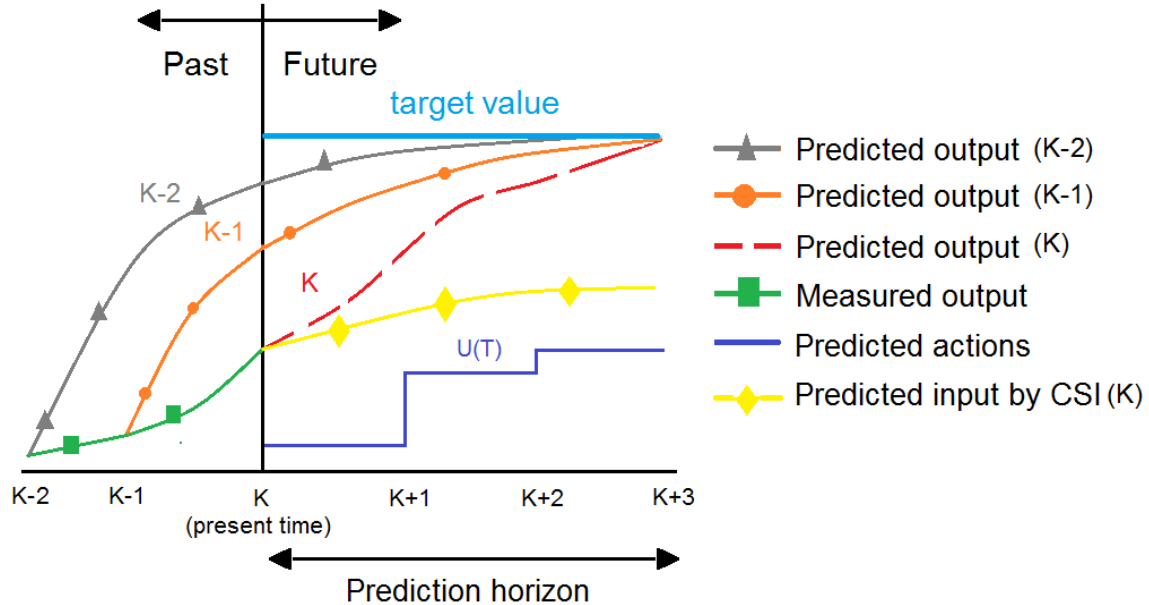


Figure 9.10: A model predictive control scheme.

In that way, we could establish the perturbation of the gate trajectories vector (ΔU) (9.7) to correct the water level error vector at the checkpoints (ΔY) for a prediction horizon from a simplification of the Hydraulic Influence Matrix $(HIM(U)')$ which represents the influence of the gates on the **water level vector** at checkpoints.

$$\Delta Y = [HIM(U)'] \Delta U$$

$$[HIM(U)'] = \left[\frac{\partial Y}{\partial U} \right] \quad (9.7)$$

The HIM matrix makes possible to implement the GoRoSoBo algorithm as this matrix establishes a direct relation between the water level error and the new gate trajectories vector. In that sense, the relation between the gate movements (ΔU) to restore the volume of water (yellow area) in the canal of Figure 9.11, is established by the HIM.

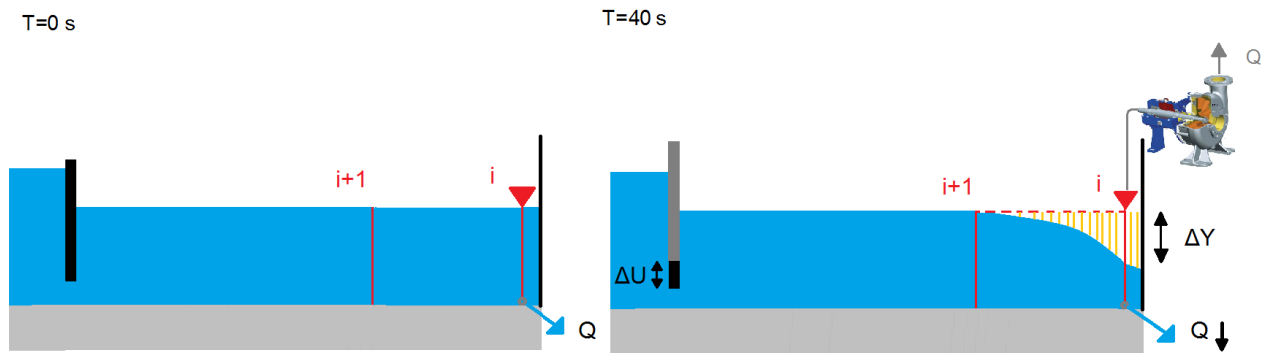


Figure 9.11: Logic of control of the feedback algorithm.

The analytic process to obtain $\left(\frac{\partial Y}{\partial U}\right)$ is analyzing the influence of a perturbation on the free surface described by the Saint Venant equations in its characteristic form. If we look at Figure 9.6, a perturbation (ΔU) introduced in the space/time domain generates an upstream wave and a downstream wave which are defined by the characteristic curves, we could analyze the variation of water level from the characteristic curves generated by the perturbation. Instead, the numerical process to obtain this matrix consist in the discretization of the analytic term calculating the state vector X (water level and velocity along the canal) considering a perturbation of the gate trajectories ($U+\Delta U$) and the state vector X obtained from the gate trajectories (U) (9.8). The analytic and numerical processes were analyzed by Soler (2003), and both results were similar.

$$\frac{\partial Y_1^{kF}}{\partial U(K)} (\text{analytic process}) \approx \frac{Y_1^{kF}(U(K) + \Delta U) - Y_1^{kF}(U(K))}{\|\Delta U\|} (\text{numerical process}) \quad (9.8)$$

9.2.4. Inputs/outputs variables of a controlled system

The system input variables are applied to a control method to produce a control action and modify the output variables of the system. In our problem, the system input variables are the state vector at current time step, scheduled demands, the desired water level vector and the gate trajectories in a previous predictive horizon. The system output variables would be the gate trajectories at the current time step for a prediction horizon with the objective to keep the desired water level at the checkpoints. In Figure 9.12 we show the input and output variables in GoRoSoBo algorithm taking into account the considered variables in the process, as follow:

- Control method: GoRoSoBo algorithm.
- Input variables: Scheduled demands, gate trajectories (previous prediction horizon), state vector at current time step and the desired water level at checkpoints.
- Output variables: gate trajectories at the current time step.

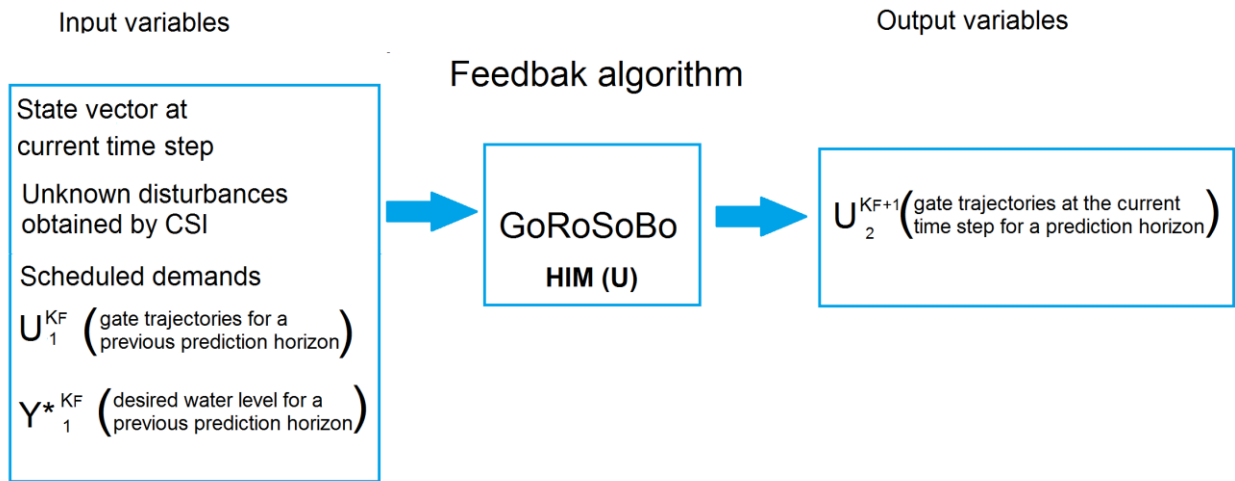


Figure 9.12: Representation of the feedback algorithm.

9.2.5. Analysis of the Hydraulic Influence Matrix: HIM(U)

The HIM matrix defines the influence of the gate on the canal state. The compiled HIM matrix ($[I_M(u)]_i$) for a gate (i) for a particular predictive horizon (j) is shown in the next equation:

$$[I_M(U)]_{ij} = \left[\frac{\partial X_1^k}{\partial U_i(1)} \quad \dots \quad \frac{\partial X_1^{k_F}}{\partial U_i(K)} \quad \dots \quad \frac{\partial X_1^{k_F}}{\partial U_i(\varphi)} \right]_j = \quad (9.9)$$

$$\begin{bmatrix} \frac{\partial x^1}{\partial U_i(1)} & \dots & \frac{\partial x^1}{\partial U_i(K)} & \dots & \frac{\partial x^1}{\partial U_i(\varphi)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial x^k}{\partial U_i(1)} & \dots & \frac{\partial x^k}{\partial U_i(K)} & \dots & \frac{\partial x^k}{\partial U_i(\varphi)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial x^{k_F}}{\partial U_i(1)} & \dots & \frac{\partial x^{k_F}}{\partial U_i(K)} & \dots & \frac{\partial x^{k_F}}{\partial U_i(\varphi)} \end{bmatrix}$$

where the state vector at time step k is defined respectively as:

$$x^k = (y_1^k \ v_1^k \ \dots \ y_i^k \ v_i^k \ \dots \ y_{n_s}^k \ v_{n_s}^k) \quad (9.10)$$

When there is more than one gate to control the canal, the matrix has to compile the gates all together. In that case, the HIM is compiled as:

$$[I_M(U)]_{Xj} = \left[\left[\frac{\partial X_1^{k_F}(U)}{\partial U_1(1)} \quad \dots \quad \frac{\partial X_1^{k_F}(U)}{\partial U_{n_g}(1)} \right] \dots \left[\frac{\partial X_1^{k_F}(U)}{\partial U_1(\varphi)} \quad \dots \quad \frac{\partial X_1^{k_F}(U)}{\partial U_{n_g}(\varphi)} \right] \right] \quad (9.11)$$

where sub index X is used to denote the HIM on the state vector.

The dimension of $[I_M(U)]_{Xj} = n_{HX} \times n_{HY}$, $n_{HX} = 2 \times n_s \times (k_F - k_i)$, $n_{HY} = \varphi \times n_g$, n_s is the total number of cross sections which has been discretized the canal, φ is the number of regulation times of the prediction horizon, n_g is the number of gates, i is the "ith" gate, j is the "jth" prediction horizon and the dimension of the state vector (x) in the time step k is $2 \times n_s$.

A feature of this matrix is about the number of columns and rows. There are more rows than columns because every term $\frac{\partial x^k}{\partial U_i(k)}$ is a vector with $2 \times n_s$ rows, and the temporal discretization of the simulation Δt^k is smallest than the regulation period ΔT^k .

In case that the matrix only defines the influence of a gate on the water level (not considering the velocity), the number of elements of the vector is reduced to n_s .

Another one feature about the HIM shows some elements with a zero value, because there is no influence of the gates over the state vector in particular points during the prediction horizon which could destabilize the algorithm because the matrix is ill conditioned. We could obtain null values in some elements of the matrix, because these elements express the influence of a disturbance at the present time when the disturbance is introduced in future, or these elements express the influence of a disturbance in a faraway section of the canal where the disturbance has not influence.

One of main problems of the HIM is the time used to calculate the matrix which depends on the pc features and the total parameters: gates, checkpoints and the predictive horizon. We tried to reduce the calculation time using the HIM around a steady state obtained previously because GoRoSoBo operates in real time and there is not so much time to calculate the gate trajectories between time steps. This option was discarded at the beginning because we tried to solve some hard tests with GoRoSoBo, as the Test-Cases proposed by the ASCE committee (Clemmens, et al. 1998), and it was impossible to solve the problems and get reasonable performance indicators using the HIM in steady state. The unscheduled water changes are quite huge in these tests. For that reason the free surface after several time steps is quite different of the free surface at the initial steady state. In that sense, the HIM would not represent the influence of the gate trajectories on the current canal state.

The HIM around a steady state would be acceptable for a test without huge flow changes, although it is also important the time duration of the flow changes. If the external disturbance is smooth, but its time duration is long, the gate trajectory obtained using a HIM matrix around a steady state is getting worst in each regulation period. For that reason, it will be impossible to control the canal.

We could do a simile between the problems associated to the HIM around a steady state with a loop problem, as the steady state is not the current canal state, the HIM around a steady state does not reflect the real canal state, and the gate trajectories obtained with this matrix will not be accurate. As the gate trajectories are not enough accurate, the real water level will never reach the desired water level at target points.

9.3. Algorithm controller from unconstrained optimization

Initially, GoRoSoBo solved an unconstrained optimization problem minimizing an objective function to reach the solution. There are several methods to solve an unconstrained

optimization problem. All of them have to reach the solution (U^*) and verify two conditions to be sure that U^* is the better solution. The first condition demonstrates that the solution is a maximum, minimum or a saddle and the second condition demonstrates that the solution (U^*) is a minimum. These conditions have to be achieved by all optimized methods (**constrained** and **unconstrained**). In this way, it is interesting to introduce these order conditions.

9.3.1 First-order and second-order conditions

It exists an extensive bibliography about the order conditions in optimization problems, for instance Fletcher (1987) and Luenberger (1984). Before solving an optimization problem, we have to be sure that the optimization problem fulfills the first order and second order conditions.

The first-order conditions are necessary to find an optimal solution of an unconstrained optimization problem as:

$$\nabla_{J_U}(U^*) = 0 \quad (9.12)$$

$$\nabla_{UJ}(U^*) = (\nabla_U X_1^{KF}(U^*)^T - Y^*)^T [C]^T [Q] ([C] X_1^{KF}(U^*) - Y^*) = 0 \quad (9.13)$$

where $\nabla_{J_U}(U^*)$ is the first derivative of the objective function around the solution (U^*).

In fact many minimization methods are based only upon trying to locate a solution U^* . This may not be a strict local minimum and in general is referred as a stationary point. It is necessary a second condition to confirm that U^* is a strict local minimum point. As we introduced previously, a minimum corresponds to a positive definite Hessian matrix.

Where the Hessian matrix can be defined as:

$$\nabla_{UU}^2 J(U^*) \approx (\nabla_U X_1^{KF}(U^*)^T) [C]^T [Q] [C] (\nabla_U X_1^{KF}(U^*)^T)^T \quad (9.14)$$

When $\nabla_{UU}^2 J(U^*)$ is used in the simplified form, this equation is called Gauss-Newton method or Least Squares method (9.14).

9.3.2 Optimized problem theory

As noted, GoRoSoBo was initially developed to solve an unconstrained optimization problem using the Levenberg-Marquardt method. The theory associated to this method was explained in chapter 6, when it was introduced the CSI algorithm, although we have developed this numerical method as the approach of Brogan perturbation theory (see Appendix 1) in which we use the Taylor series to develop the perturbation equation around a gate trajectory.

9.3.3 Implemented algorithm

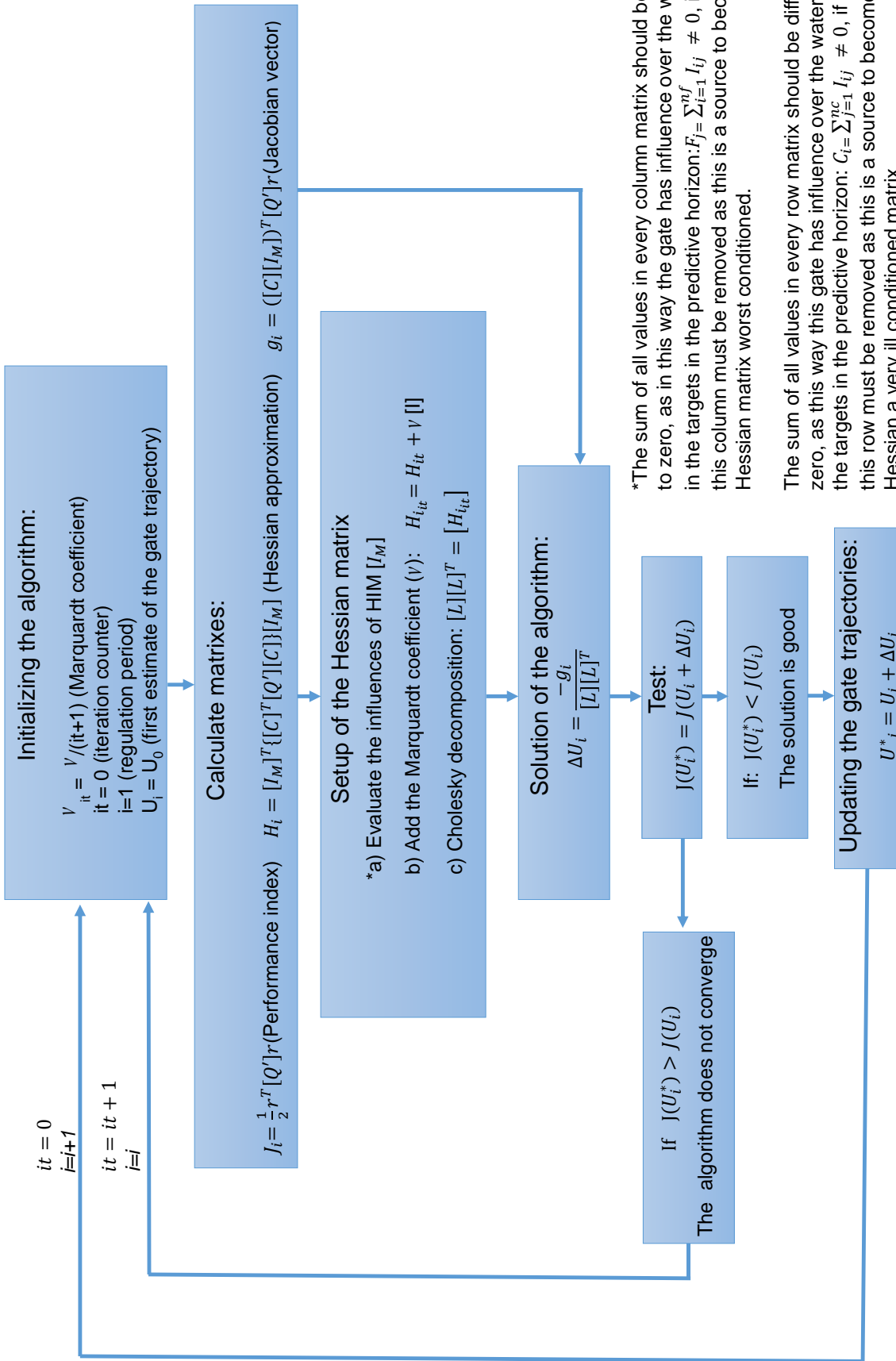
We used Levenberg-Marquardt method, because this method is robust and applicable to ill-conditioned matrix as the Hessian matrix. The HIM matrix is a positive definite matrix but it is also an ill conditioned matrix. The approximation of the Hessian matrix, obtained from multiplying the HIM matrix by its transposed, has a worst condition number than the HIM matrix. As noted, we use the Levenberg-Marquardt method which introduces a value (ν) in the diagonal of the Hessian matrix to improve the condition number of the matrix.

$$H = [\nabla_{UU}^2 J(U^*) + \nu [I]] \quad (9.15)$$

where $[I]$ is the identity matrix and $\nu > 0$ is a real value named Marquardt coefficient. The steps defined in the implemented algorithm are shown in the next flowchart which is a type of diagram that represents every process, showing the steps as boxes, and their order by connecting them with arrows. This representation illustrates a solution to the optimization problem. Process operations are represented in these boxes, and the sequencing of operations by the arrows. Before to show the flowchart, we have to define some matrixes and variables that are used in the flowchart:

- $[Q']$ is the weight matrix.
- $[C]$ is the discrete observer matrix (a matrix of zero and ones).
- $r' = [C]X_{ki}^{kF}(U_i) - Y^*$ is the residual vector.
- $X_{ki}^{kF}(U_i)$ is the state vector.
- $[I_M] = I_M[X_{ki}^{kF}(U_i)]$ is the Hydraulic Influence Matrix, all evaluated in U_i .

All the process of the algorithm are defined in each loop, where the solution U^* is calculated in each regulation period (i) for a predictive horizon.



9.3.4 Analysis of the unconstrained optimization problem

As we introduced in the last section (9.3.1), our control algorithm was implemented from the HIM matrix around a steady state solving an unconstrained optimization problem but this method did not function properly in several tests in which the disturbances were significant.

This part of the thesis was also important; because we realized that it was impossible to develop a control algorithm without information about the current canal state. A water level error (difference between the computed water level and measured water level) at a checkpoint in an instant of time modifies the free surface of a canal for a short time, whereas a flow change at a checkpoint during a regulation period modifies the free surface for a long time (Figure 9.13). Both situations are quite different and the control algorithms should take into account one case or another depending the situation, because the gate trajectories will be different in both cases. For that reason, it is so important obtain the disturbances with CSI, because we know as the free surface is modified in any time.

If a disturbance is introduced into the canal, the algorithm cannot control the canal only measuring the water level change and making a prediction of the future error, without the information about the disturbance. We realized in that moment that the controller needed the CSI algorithm, as the control algorithm needs to know the current canal state and disturbances at every regulation time.

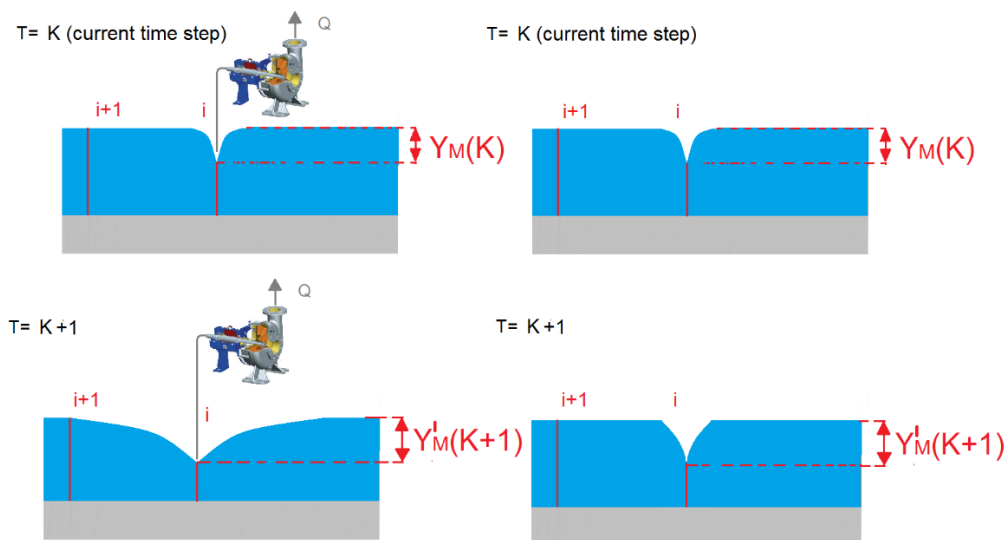


Figure 9.13: Flow disturbance during a regulation period V.S. water level disturbance at a particular time step.

As we introduced before, our controller solved an unconstrained optimization problem based on the system of equations (9.7). We implemented an unconstrained optimization problem in our predictive control, because it is easier to implement the code and reduces the computation time in front of algorithms based on constrained optimization but the results with this algorithm were not good.

Sometimes one gets more information from the failures than from successful tests. In case that a significant unscheduled flow change is introduced in the canal, the predicted water level error obtained by CSI would be significant too. So the predictive controller would introduce important changes in gate trajectories to drive the system to the desired level such changes could be physically impossible and the algorithm could fail. This was one of the reasons to introduce constraints in the optimization problem.

We realized about this problem when we implemented the Test-Cases proposed by the ASCE (Clemmens et al., 1998). The first part of the Test-Cases (scheduled offtake changes) has been solved by several feedforward controllers, one of them was GoRoSo (Soler, 2003). Initially, this algorithm also solved an unconstrained optimization problem but GoRoSo was tested with these Test-Cases and the results were not good due to the important scheduled offtake changes. This was one of the reasons to modify the algorithm and GoRoSo implemented a constrained optimization problem which solved successfully the Test-Cases obtaining good performance indicators.

We had the same problems with GoRoSoBo when we tried to solve the second part of the Test-Cases (unscheduled offtake changes). Although the control problems solving the second part were more important than the first part because the feedback algorithm does not have any notice about the disturbance until the water level at the checkpoints has been perturbed, that is, there is not any possibility to change the gate trajectory to reduce the future water level errors before the disturbance perturbs the water level at the checkpoints. Instead, the scheduled offtake changes of the first part are known in advance by the feedforward algorithm at the beginning of the irrigation cycle.

It is absolutely necessary to introduce constraints in our optimization problem. A constrained optimization problem is solved by GoRoSoBo.

Before changing the code of our feedback algorithm to solve a constrained optimization problem, we studied other options before developing a constrained optimization problem. We

tried to introduce “artificial constraints” in an unconstrained optimization problem with the aid of the Marquardt coefficient. Although the real objective of using Marquardt coefficient is to improve an ill conditioned matrix, we tried to use the Marquardt coefficient to restrict the gate movements. The Marquardt coefficient was computed with the aim that the minimum gate position is equal or greater than the lowest gate position allowed. The problem is that all the gate trajectories calculated by the algorithm are penalized for this Marquardt coefficient and the coefficient would only be able to restrict the minimum gate position. In some way, applying the Marquardt coefficient is the same that modify the influences of the elements of the HIM, because we apply this coefficient in the diagonal of the Hessian matrix.

We dismissed this option and we implemented in our algorithm a constrained optimization problem solved by the Lagrange-Newton method.

9.4. Algorithm controller from constrained optimization

As we discussed in previous sections of this chapter, we need to include constraints on the gate movements. In case of disturbances, the feedback algorithm computes new gate trajectories, taking into account the constraints, to maintain the water level as close as possible to the target water level at the checkpoints, so the gate trajectories computed by the algorithm are physically possible and the best approximation to the optimal gate trajectories. Establishing constraints in a predictive control has been made by many authors, for instance by Soler (2003) in GoRoSo (feedforward predictive control) or Zagona (1992) in MPC (model predictive control).

We can classify the different types of constraints as:

- Overall constraints
- Functional constraints

Taking into account the experience of these authors about constraints in predictive controllers, we introduced a set of overall constraints to avoid gate positions lower than the minimum gate opening and greater than the maximum gate opening (Figure 9.14):

$$U_{ij}(K) - U_{min} \geq 0 \tag{9.16}$$

$$U_{max} - U_{ij}(K) \geq 0 \tag{9.17}$$

where U_{\min} is the minimum gate position, U_{\max} is the maximum gate position, K is the number of the regulation period in a predictive horizon, “ j ” is the “ j th” predictive horizon, and “ i ” is the “ i th” gate. That is, $U_{ij}(K)$ is the gate position in the “ j th” predictive horizon, for the “ i th” gate in the regulation period K .

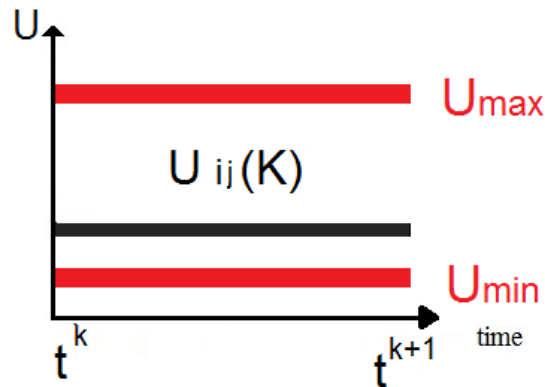


Figure 9.14: Maximum and minimum gate position for the constrained optimization problem.

We also use functional constraints which restrict the gate movements between time steps. The gate position between consecutive time steps should not be very different, because the free surface would change a lot between regulation periods and it would be difficult to control the canal. When does the control algorithm compute big changes in gate position for consecutive time steps?

This happens in the following situations:

- The first one is by significant flow changes in the canal.
- The second one is by small flow changes but the gate has a low influence on the target points. It is possible that a gate has a low or null influence on several target points because these points are far away of this gate and the wave height generated by the gate is attenuated a lot due to the friction forces, or because the wave generated by the gate arrives at the target points later than the end of the prediction horizon. If we want to fix the desired water level at the checkpoints with this gate, as the influence is low or null, the gate movement would be significant and functional constraints would be necessary (Figure 9.15):

$$|u_{ij}(K + 1) - u_{ij}(K)| \leq dU_{max} \quad (9.18)$$

where dU_{max} is the maximum gate movement acceptable between consecutive regulation periods. This value can be associated with the characteristics of the servo-motors used to move the gate.

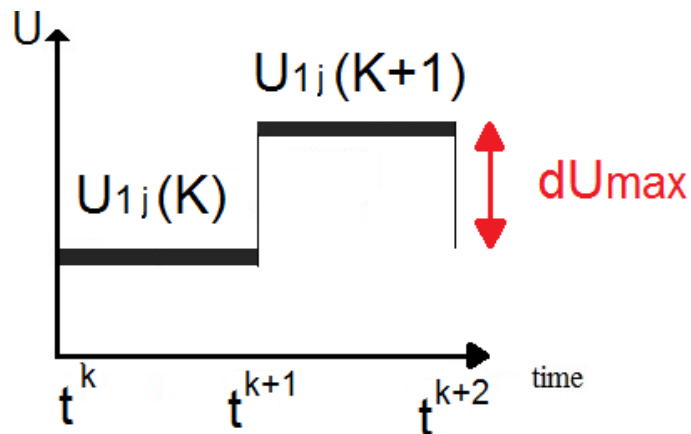


Figure 9.15: Maximum gate movement between consecutive regulation periods for the constrained optimization problem.

After numerous tests using overall and functional constraints, it did not work. All of these kinds of constraints were not enough. We thought that a new one constraint was necessary to control the system, because some gate movements destabilized the system.

The predictive controller computes new gate trajectories for a predictive horizon in every regulation period, but only the gate positions during the first regulation period for the prediction horizon are sent to the actuator ($U_{ij}^*(K)$) (Figure 9.16). In the next regulation period, new gate trajectories will be computed by the control algorithm and only the gate positions at the first regulation period are sent to the actuator again ($U_{ij}^*(K)$). The gate trajectories used in the control structures contain the gate position in the first regulation period of each predictive horizon. For that reason, we need to restrict the gate movement between consecutive regulation periods of different predictive horizons (Figure 9.16). A feedback controller, as GoRoSoBo, needs new constraints between gate positions for consecutive regulation periods obtained in different predictive horizons. The equations associated to this new constraints are written as:

$$d_0 U \max - |u_{ij}(K+1) - U_o| \geq 0 \quad (9.19)$$

$$U_o = u_{i(j-1)}(K)$$

where $d_0 U \max$ is the maximum gate movement acceptable between consecutive regulation periods of different prediction horizons. U_o is a vector with the gate positions at “K” regulation period for the prediction horizon (j-1), so U_o is constant in the “jth” prediction horizon. A range of values to fix $d_0 U \max$ is similar to the $dU \max$ which could be estimated from empirical methods based on the experience of engineers in real cases, or from the features of the servo-motors which drive the gates.

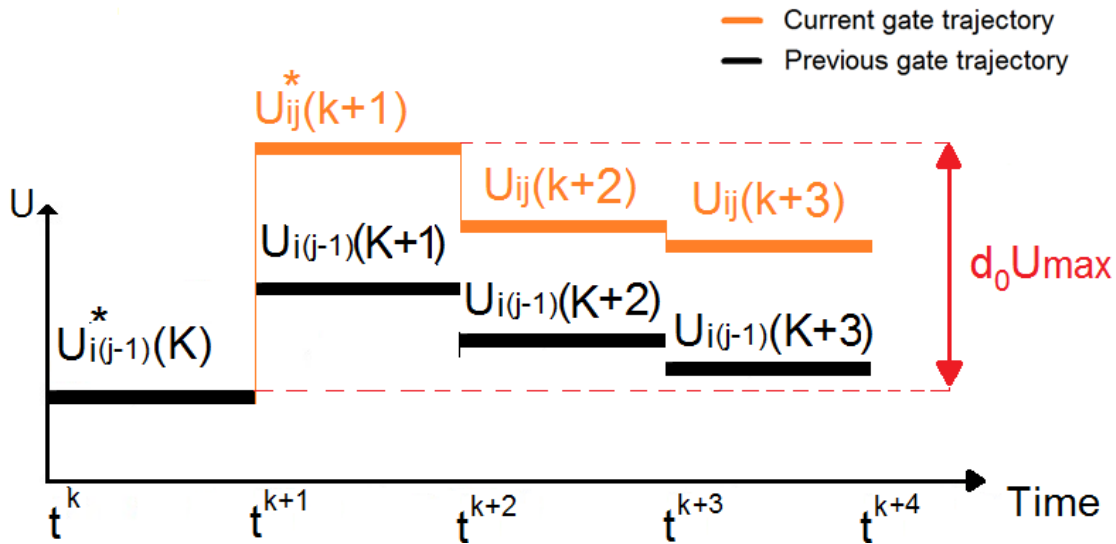


Figure 9.16: Maximum gate movement between consecutive regulation periods for different predictive horizons for the constrained optimization problem.

Previously to discuss about the first and second order conditions, introduced in Appendix 2, which must be fulfilled by the constrained optimization problem, it is necessary to analyze the overall and functional constraint functions proposed in this constraint optimization problem. The Kuhn-Tucker conditions (first order condition) are necessary only if a particular condition is satisfied. That condition, called the constraint qualification, imposes certain restrictions on the constraint functions of a nonlinear programming problem, for the specific purpose of ruling out certain irregularities on the boundary of the feasible set, which would invalidate the Kuhn-Tucker conditions, as show Fletcher (1987) or Luenberger (1984). The constraint functions must fulfill the constraint qualification and this condition is fulfilled in our optimization problem, because the active constraints functions are linear, and the first

derivative of these functions are linearly independent, see in Soler (2003). We have to note, that only active restrictions will be taken under consideration to evaluate the constraint qualification. The discussion about the constraint qualification is shown in the Appendix 3.

9.4.1. Optimized problem theory

The methods used to solve constrained optimization problem were two: the Lagrange-Newton method and the Active Set Method.

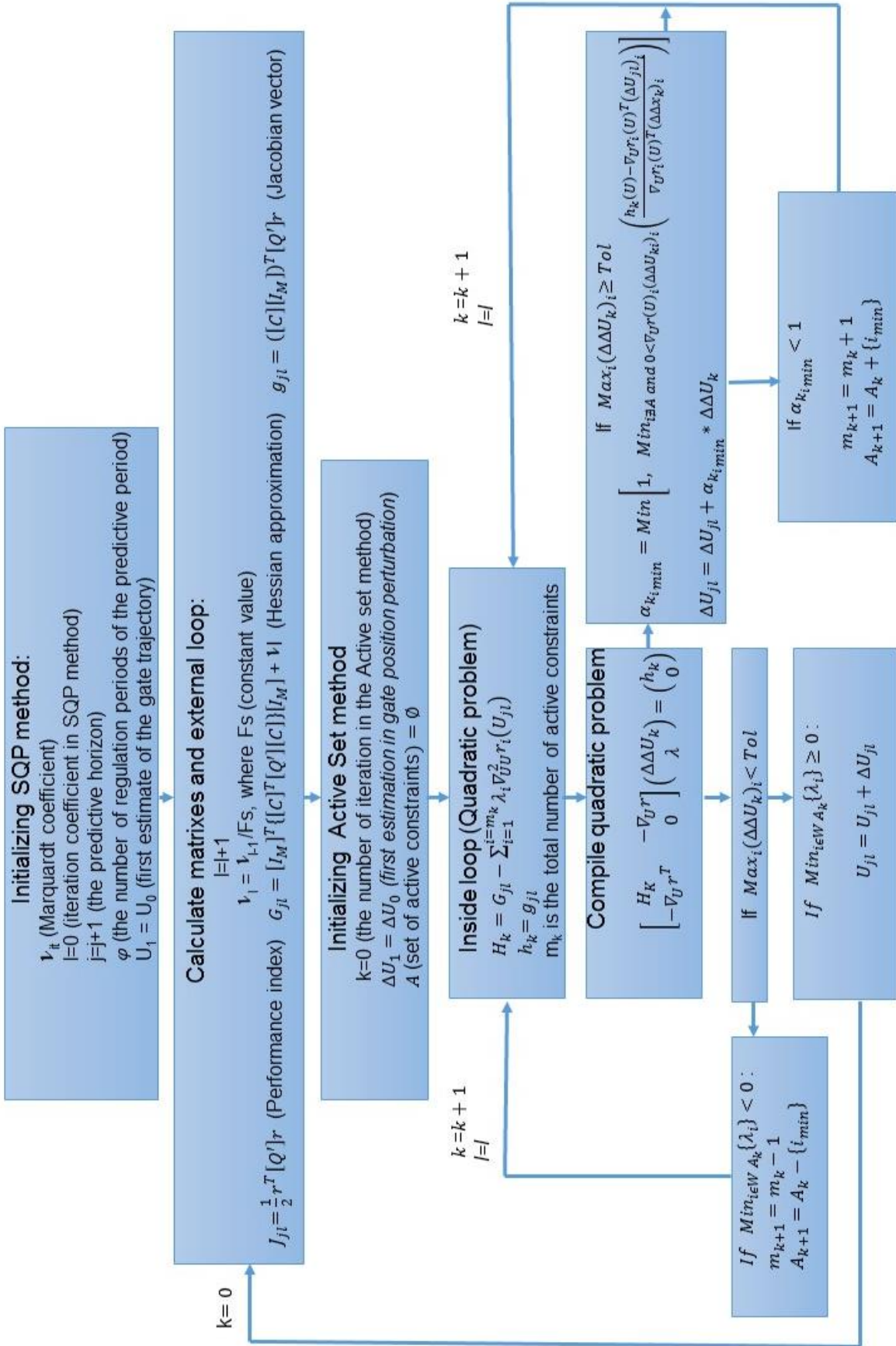
The placing of one loop inside the body of another loop is called nesting. We use the Active Set method nested inside of the body of Lagrange-Newton method. These methods were chosen taking into account the good results obtained when they were implemented in GoRoSo control algorithm by Soler (2003).

The Lagrange-Newton method, also called SQP (Sequential Quadratic Problem) by Fletcher (1987), is an iterative method and each iteration sets a quadratic problem and every quadratic problem is solved by the Active set method. We use these methods because they are robust and fast convergence methods in optimization problem. The SQP method is usually used in constraint optimization problems where the constraints are nonlinear. We use this method although the constraints introduced in our optimization problem are linear.

The SQP method is based on applying the Newton's method to find the stationary point from the Lagrange function, for that reason this method is called Lagrange-Newton method, see Fletcher (1987). Both methods are used to solve the constrained optimization problem which is defined in each regulation period. A detailed explanation of the Lagrange-Newton and Active Set methods is done in the Appendix 4.

9.4.2. Implemented algorithm

The implemented algorithm consists of two iterative loops: an **interior loop** solves the quadratic problem by the Active set method nested within the **external loop** which solves the SQP (Sequential Quadratic problem). The steps defined in the implemented algorithm are shown in the next flowchart which is a type of diagram that represents every process, showing the steps as boxes, and their order by connecting them with arrows. As we did in the unconstrained optimization problem, we also added the Marquardt coefficient to the diagonal of the Hessian matrix to improve its condition number.



9.4.3. Analysis of the Future State Computation

If we want to define accurately the future canal state; we should know the future disturbances. The problem is that we do not know the future changes in flow, because these changes can be caused by unscheduled flow extractions and this will happen at the future, not right now.

At this point we have two options:

- Establishing the current canal state and not make future predictions.
- Establishing the current canal state and make future predictions.

So the question once CSI has established the disturbances in the canal would be: Which is the future forecast in flow changes? Our future predictions could be that the disturbance will only exist during a regulation period, but we could also think that the disturbance will exist during several regulation periods. It is not an easy question because the control action calculated by GoRoSoBo will be different in both options. In that sense, in the second option the controller response will be stronger throughout the predictive horizon because the duration of the disturbance is longer than the first option. The canal state, that is, the velocity and the water levels along the canal are different in both options (Figure 9.17) as well as the gates openings because the flow through the gates must be different, although the water level at the checkpoint ($Y=Y^*$) is equal.

The problem is that the feedback controller modifies the gate opening according to the current canal's hydrodynamic state and the future predictions. In case that we do not make future predictions about the disturbance, we suppose that the disturbance were introduced from the past to the present time, but if the disturbance has a long time duration in future, our future predictions will be wrong respect to the reality. In that case, the feedback controller cannot calculate the optimum gate trajectories because it does not have a good prediction of the future canal state. In that sense, if our current predictions do not assume a disturbance with a long time for the future the gate trajectories solution is bad, because the water level error at the checkpoints is significant at the end of the simulation. The solution in gate trajectories obtained by GoRoSoBo always maintains a stationary water level error at the end of the simulation, and the sluiceway trajectory never corrects this error, see Figure 9.18. This is a generalized problem in all predictive controllers, nobody knows which value and how is distributed the error during the prediction horizon, see Figure 9.10.

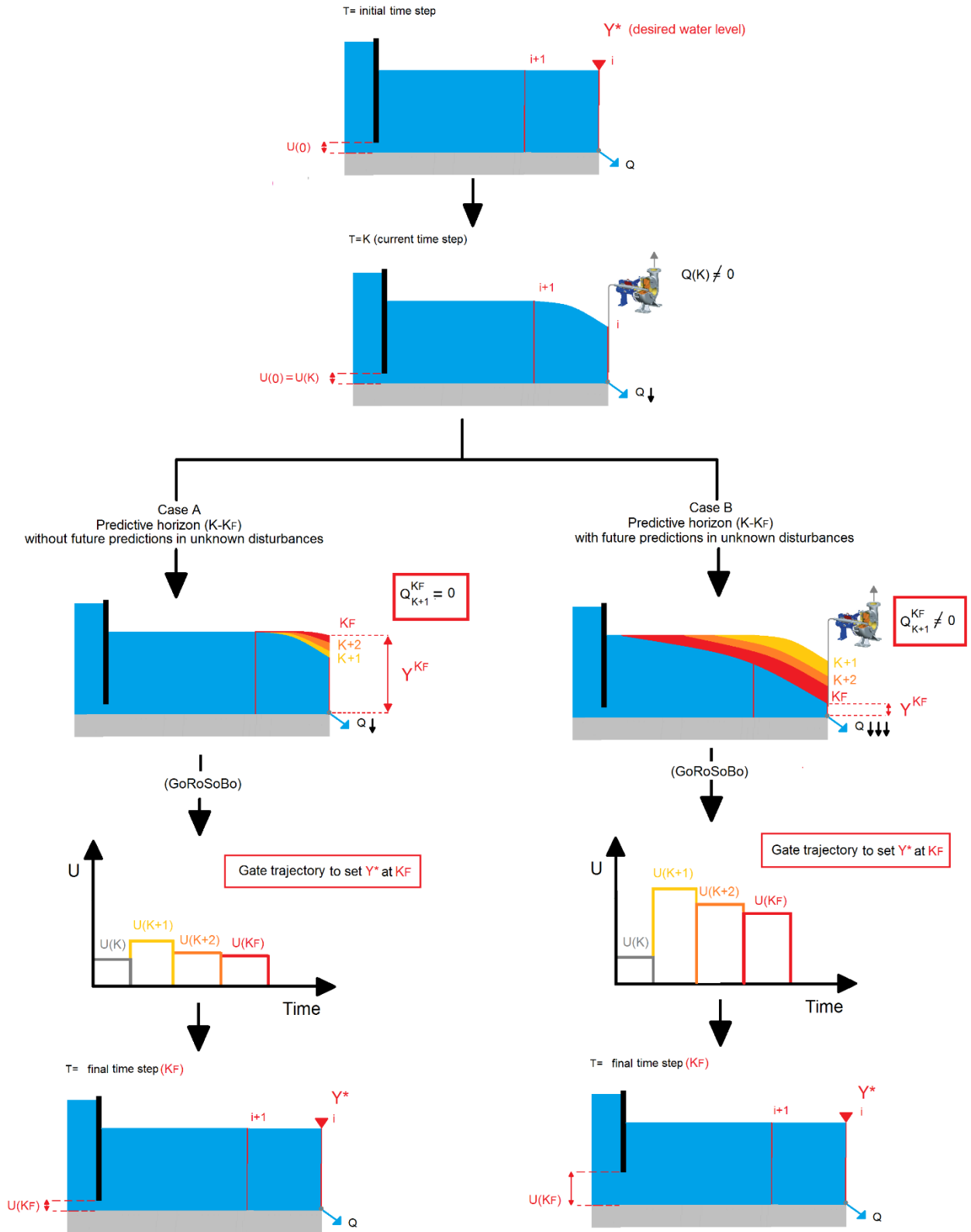


Figure 9.17: Logic of control of the GoRoSoBo algorithm.

There seems not to be a direct cause, but our predictions of the canal state in future take more relevance depending on the kind of the disturbance introduced into the canal. After performing several tests with different kind of disturbances using GoRoSoBo, we concluded that if the disturbance is significant and especially for long time duration, although GoRoSoBo knows the current canal state, it is not enough to find the right solution of gate trajectories. This problems associated to this kind of disturbance is related with the future prediction of the disturbance.

Summarizing, the feedback controller cannot manage the canal for that kind of disturbance, only considering the current canal state.

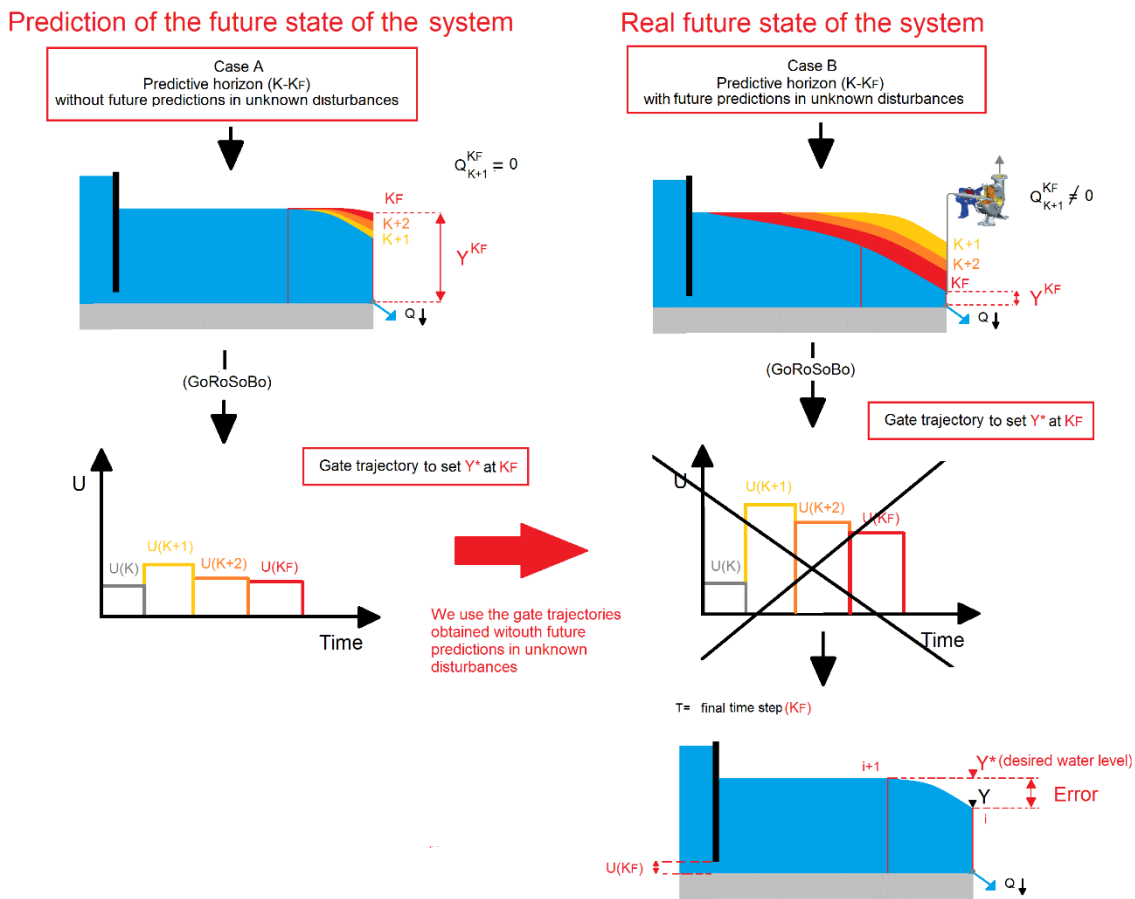


Figure 9.18: Associated problems to disturbances with long duration in GoRoSoBo not using future prediction.

In case that the real disturbance is only introduced during a regulation time period, when the disturbance disappears the water level is recovered quickly at checkpoints and the water level error in future will be low. In this way, the current changes in gate position to solve a

minor error in the future would be small, as once the disturbance disappears the water levels at checkpoints are recovered quickly by themselves introducing little changes in gate trajectories. The algorithm will not introduce big perturbations in the system at present time, taking into account that the error is reducing in the future, and the perturbations introduced by the gate could be more important than perturbations for the disturbance.

But in cases that the disturbance has a long time duration, it is necessary establish future prediction, because the changes in gate trajectory introduced by GoRoSoBo at the present time to correct the error in future are fundamental, due to the control actions taken at the present time are reflected in future, see Figure 9.19. Not considering that the disturbance will continue over a long period of time, it is the cause of not driving the canal to the desired state and maintains a stationary error between the desired and real water level in future.

For this reason, once the extracted flow vector is established by CSI, the changes in flow at present time will be the forecast for the future in chapter 11 and 12. In other words, the changes on the extracted flow vector obtained currently, keep on being the changes in this vector during the prediction horizon. Although it is impossible to predict the future, there is no any way to drive the canal to the desired state in case of this kind of disturbances; so we will make this future prediction.

Someone could think that this solution probably is the best when the disturbance has a long time duration, but what will happen if the flow disturbance disappears in a short period of time and by contrast our future predictions suppose the opposite?. In this case, the algorithm would calculate a gate trajectory taking into account a significant disturbance in future. There is no simple answer to this question.

The result depends on the duration of the regulation period, as CSI receives water level measurements every regulation period, and CSI recomputes the extracted flow vector every regulation period. If this period is short, CSI would determine sooner that the disturbance has disappeared and GoRoSoBo would recompute quickly a new gate trajectories to correct the last one.

In GoRoSoBo algorithm, the gate movements are restricted because we try to avoid instability in the numerical model. For these reasons, the changes in water level introduced by excessive gate movements, due to wrong future predictions, are not so important.

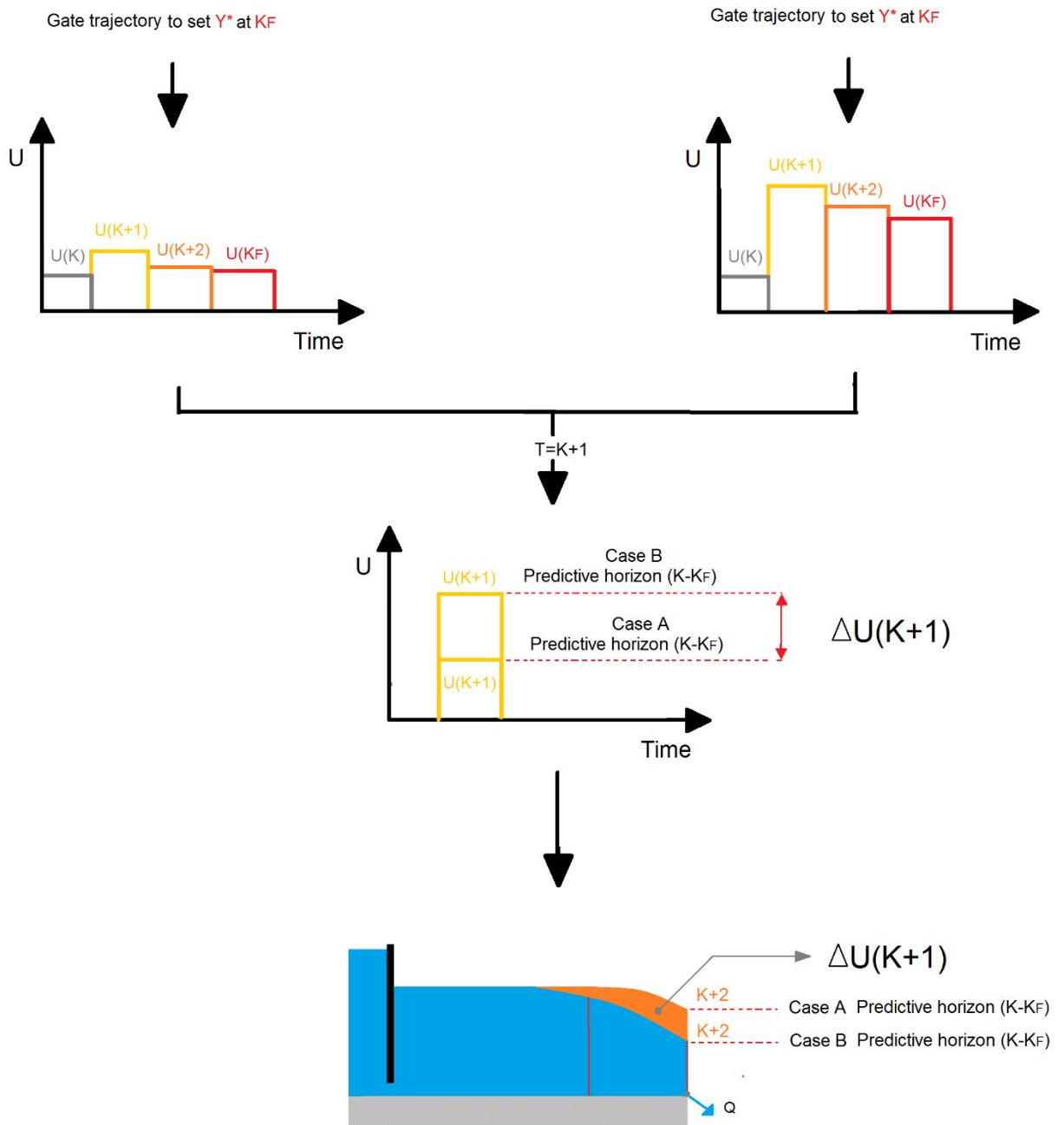


Figure 9.19: Variation of the canal state taking into account different predictive horizons.

9.4.4. Conclusions of the constrained optimization problem

As a summary, we can conclude that:

1. The GoRoSoBo obtains information from CSI about the current canal state and the extracted flow, so GoRoSoBo knows the water level error along the canal at the present time and it can make a prediction of the water level error for the prediction horizon from this information. The only way to get the current state in a canal, without CSI, would be measuring in every cross section of the canal and using a calibrated predictive model and the only way to establish the extracted flow in a canal would be implying the installation of flow meters in every extraction point.
2. The HIM matrix, that represents the influence of a sluice gate on the canal state, is an excellent tool for GoRoSoBo which computes a new set of gate trajectories due to variations between the measured and the desired water level.
3. The HIM matrix must be recomputed every regulation period in case of important disturbances.
4. In front of significant disturbances, GoRoSoBo should use constraints, and mostly feedback controllers.
5. The Lagrange-Newton method used to solve a constrained optimization problem is a robust and fast convergence method.
6. GoRoSoBo is a feedback algorithm so the gate trajectories are computed at every regulation period during the irrigation cycle. For that reason, the algorithm needs a new overall constraint to restrict the gate movements at consecutive regulation periods from different prediction horizons.
7. It is necessary that GoRoSoBo introduces forecasted predictions in disturbances to drive the system to the desired state.
8. The GoRoSoBo algorithm shows promising results in all tests done, as we will show in the next chapters eleven and twelve.

9.5. APPENDIX I

Unconstrained optimized problem (Levenberg-Manquardt)

In the next equation, we show the "perturbation theory" by Brogan (1985) developed (eq.(9.20)) in Taylor series around the gate reference trajectory:

$$Y = Y(U^* + \Delta U) = Y^*(U^*) + \left[\frac{\partial Y}{\partial U} \right]_{U^*} \cdot \Delta U + \frac{1}{2} \left[\frac{\partial^2 Y}{\partial U^2} \right]_{U^*} \cdot \Delta U^2 + \dots \quad (9.20)$$

Assuming that the disturbance caused is sufficiently small, we can neglect the terms after the first term (eq.(9.21)), obtaining:

$$\Delta Y = Y(U^* + \Delta U) - Y^* = \left[\frac{\partial Y}{\partial U} \right]_{U^*} \Delta U \quad (9.21)$$

On the other hand, it is possible to use the hydraulic influence matrix (HIM), defined by Soler (2003) as:

$$[J^*] = \left[\frac{\partial Y}{\partial U} \right]_{U^*} \quad (9.22)$$

Taking account all of these terms, we can write:

$$\Delta U = [J^{*T} \cdot J^*]^{-1} \cdot [J^{*T}] \cdot \Delta Y \quad (9.23)$$

We can compute a set of ΔU gate movements to reach the desired water level ($Y=Y+\Delta Y$) at the checkpoints. The equation system can be solved by Jacobi method, Cholesky method, singular value decomposition method or LU decomposition. There is an extensive bibliography about these methods, as Numerical Recipes in FORTRAN 77 (1992), where it is also illustrated the Levenberg-Marquardt method.

The problem is to solve the equation (9.23) at each time step. We can define the equation on a compact form:

$$[G] = [J^{*T}] \cdot [J^*] \quad (9.24)$$

$$b' = [J^{*T}] \cdot \Delta Y \quad (9.25)$$

So replacing the equation (9.24) and equation (9.25) in equation (9.23), we obtain:

$$b' = [G] \cdot \Delta U \quad (9.26)$$

Although the system matrix is positive definite by definition, is not well conditioned (Soler, 2003). This implies that solving the system, we can get numerical errors. To avoid this problem we add a positive term to the diagonal of the Hessian matrix, this parameter is called Marquardt coefficient (ν). It exists a lot of theory about this coefficient, for instance in Fletcher (1987).

The system of equations can be written then as:

$$b' = [G + \nu I] \cdot \Delta U \quad (9.27)$$

where $[I]$ is the identity matrix of the same dimensions as $[G]$.

To solve the system (eq.(9.27)), we have to make some hypothesis. We have a diagonal matrix $[\Lambda]$ with the eigenvalues of the matrix $[G]$, and a matrix orthonormal ($[V]$: $[V \times V^T] = [I]$) defined by the eigenvectors associated with each eigenvalue of $[G]$. Multiplying both sides of equality (eq.(9.27)) for the matrix $[V^T]$, we obtain:

$$[V^T \cdot G + V^T \cdot \nu I][V \cdot V^T] \Delta U = [V^T] \cdot b' \quad (9.28)$$

Knowing that the $[V]$ matrix is orthonormal, we can write the system as:

$$[V^T \cdot G \cdot V + V^T \cdot \nu I \cdot V][V^T] \Delta U = [V^T] \cdot b' \quad (9.29)$$

And according to the property $[\Lambda] = [V^T \times G \times V]$, it can be transformed (eq.(9.29)) as follows:

$$[\Lambda + \nu I] \cdot [V^T] \cdot \Delta U = [V^T] \cdot b' \quad (9.30)$$

Furthermore, the fact of using a matrix as orthonormal matrix $[V]$ allows us to ensure that:

$$\|[\Lambda + \nu I] \cdot [V^T] \cdot \Delta U\| = \|[V^T] \cdot b'\| \quad (9.31)$$

To simplify (eq.(9.30)), we can define the vectors Z and B as follow:

$$Z = [V^T] \cdot \Delta U \quad (9.32)$$

$$B = [V^T] \cdot b' \quad (9.33)$$

So Z is an unknown quantity, and (eq.(9.30)) can be transformed into:

$$[\Lambda + \nu I]Z = B \quad (9.34)$$

From (eq.(9.34)), the Z components can be separated easily because the new system matrix is now diagonal. Each of the Z components can be calculated as (eq.(9.35)):

$$Z(i) = \frac{B(i)}{\Lambda(i,i) + \nu} \quad (9.35)$$

Although the matrix [G] is positive definite, it is ill conditioned too, and it is possible that some eigenvalue of [G] are negative, although very close to absolute zero. To prevent this problem, we introduced the Marquardt coefficient. If the Marquardt coefficient value is significant ΔU will tend to zero and the water level in the targets cannot be recovered. However, if V is very close to the minimum you can get excessively large ΔU , which cause large fluctuations in level. So you are looking for a range of values V that provides a magnitude of a vector ΔU minor or equal to a certain value ε_{\max} . Depending on the value you chose for it, the solution will be different, for this reason the value of this coefficient is getting smaller in each iteration of the algorithm, in order to achieve that the solution is getting closer of the optimum solution.

Once you know the initial value for V , the system (eq. (9.35)) is resolved and consequently the system (eq.(9.32)), obtaining the desired values of ΔU (eq.(9.36)).

$$\Delta U = [V] \cdot Z \quad (9.36)$$

In that sense, the algorithm computes a new gate trajectory ($U_a = U + \Delta U$) in the time step K for a prediction horizon.

9.6. APPENDIX II

The Lagrange multipliers is the method used to solve the constrained optimization problem. The method of the Lagrange multipliers is explained in Fletcher (1987), Luenberger (1984) and Gill (1981), although we will make a brief introduction of the method in this appendix. We must prove that our constrained optimization problem fulfils the order conditions, as we show in this appendix 2, to be sure that we obtain the optimal solution.

First order condition

We could introduce the constrained optimization problem as follow:

$$\text{Minimize } J(U) = \frac{1}{2} [[C]X_{ki}^{kF}(U) - Y^*]^T [Q] [[C]X_{ki}^{kF}(U) - Y^*] \quad (9.37)$$

$$r_k(U) = 0, \quad i \in I(U)$$

$$r_k(U) \geq 0, \quad i \in NI(U)$$

where $J(U)$ is the objective function, $X_{ki}^{kF}(U)$ is the state vector from the regulation time step K_i to K_F , Y^* is the desired water level vector, Q is a weighting matrix, C is the discrete observer matrix and $r_k(U)$ is the “kth” constraint function, $I(U)$ is a set of equalities constraints and $NI(U)$ is a set of inequalities constraints.

Minimizing the objective function, we obtain the gate solution U^* . A feasible point U' fulfills all constraints, see Fletcher (1987). When a feasible point fulfills an inequality constraint as an equality is called active constraint. A set of constraints (equality or inequality) which are active in U' are called active set.

We could define the constrained optimization problem from the Lagrange function as follow:

If U^* is a local minimum of the constrained optimization problem and λ_i are the Lagrange multipliers associated to the restrictions $r_k(U^*)$, we can establish the following system:

$$\mathcal{L}(U^*, \lambda^*) = J(U^*) - \sum_{i=1}^{nt} \lambda_k^* r_k(U^*) \quad (9.38)$$

In that case, the first order conditions or Khun-Tucker are written as:

$$\begin{aligned}
\nabla_U \mathcal{L}(U^*, \lambda^*) &= 0 \\
r_k(U^*) &= 0, \quad i \in I \\
r_k(U^*) &\geq 0, \quad i \in NI \\
\lambda_k^* &\geq 0, \quad i \in NI \\
\lambda_k^* r_k(U^*) &= 0, \quad \forall i
\end{aligned} \tag{9.39}$$

If we develop the first derivative, the constraints and the multipliers, we will obtain:

$$\begin{aligned}
\nabla_U \mathcal{L}(U^*) &= (\nabla_U X_1^{KF}(U^*))^T \{ [C]^T [Q] ([C] X_1^{KF}(U^*) - Y^*) + \nabla_x r^T \lambda^* \} \\
&\quad + \nabla_U r^T \lambda^* = 0
\end{aligned} \tag{9.40}$$

where,

$$\begin{aligned}
\lambda &= (\lambda_1, \dots, \lambda_{nt})^T \\
r &= (r_1(X(U^*), U^*), \dots, r_{nt}(X(U^*), U^*))
\end{aligned}$$

If the solution U^* accomplishes this system (equation (9.40)), so the solution accomplishes the first order condition.

So far, we have only considered the first order conditions (that is the first derivate). It is also possible to state second order conditions which give information about the curvature of the objective function at a local minimum.

The second order conditions

In this paragraph we analyse the second order condition, that is, we analyse the function in the neighbourhood of the local solution, where $G(U^*)$ is a set of feasible directions. The full definition of this second order condition is shown in Fletcher (1987), where it is demonstrated how we reach to the next expression:

$$s^T W(U, \lambda)^* s > 0 \quad \forall s \in G(U^*) \tag{9.41}$$

$$W(U^*, \lambda^*) = \nabla_{UU}^2 J(U^*) - \sum_{i=1}^{nt} \lambda_k \nabla_{UU}^2 r_k(U^*) \tag{9.42}$$

where $W(U^*, \lambda^*)$ must be a definite positive matrix and “s” is a vector of any fixed length, the limiting vector “s” is referred to as a feasible direction to get U^* .

That means, the Lagrange function has not negative curvature for all feasible directions at U^* .

So we need to derive the equation (9.40) to get $\nabla_{UU}^2 \mathcal{L}$:

$$\begin{aligned} \nabla_{UU}^2 \mathcal{L}(U^*, \lambda^*) &= (\nabla_{UU}^2 X_{K_{l+1}}^{K_F}(U^*)^T) \{ [C]^T [Q] ([C] X_{K_{l+1}}^{K_F}(U^*) - Y^*) + \nabla_x r^T \lambda^* \} \\ &+ [\nabla_U X_{K_{l+1}}^{K_F}(U^*)^T] \{ [C]^T [Q] [C] + [\nabla_{XX}^T r] \lambda \} [\nabla_U X_{K_{l+1}}^{K_F}(U^*)^T]^T \\ &+ [\nabla_U X_{K_{l+1}}^{K_F}(U^*)^T] \sum_{i=1}^{i=n_T} \lambda_i [\nabla_{UX}^2 r_i] \\ &+ [\nabla_U X_{K_{l+1}}^{K_F}(U^*)^T] \sum_{i=1}^{i=n_T} \lambda_i [\nabla_{XU}^2 r_i] + \sum_{i=1}^{i=n_T} \lambda_i [\nabla_{UU}^2 r_i] \end{aligned} \quad (9.43)$$

where $\nabla_{\lambda U}^2 \mathcal{L} = \nabla_{U\lambda}^2 \mathcal{L} = -\nabla_U r$.

As the constraint functions are linear, some terms of the system (9.43) are null: $\nabla_{UX}^2 r_i = 0$, $\nabla_{UU}^2 r_i = 0$, $\nabla_{XX}^2 r_i = 0$ and $\nabla_U^2 r_i = 0$ for $i=1, \dots, n_t$, so:

$$\begin{aligned} \nabla_{UU}^2 \mathcal{L}(U^*, \lambda^*) &= (\nabla_{UU}^2 X_{K_{l+1}}^{K_F}(U^*)^T) \{ [C]^T [Q] ([C] X_{K_{l+1}}^{K_F}(U^*) - Y^*) \} \\ &+ [I_M]^T_Y [Q] [C] [I_M]_Y \end{aligned} \quad (9.44)$$

where:

$$[I_M]_Y = [C] (\nabla_U X_{K_{l+1}}^{K_F}(U^*)^T) \quad (9.45)$$

When the solution U^* is the optimal solution, so $([C] X_{K_{l+1}}^{K_F}(U^*)^T - Y^*)^T$ is close to zero. This means that the predicted behavior of the model and the desired behavior are very similar, so in this case, it is easy to demonstrate that $\nabla_{UU}^2 \mathcal{L}(U^*, \lambda^*)$ becomes positive definite as $[C] (\nabla_U X_{K_{l+1}}^{K_F}(U^*)^T)$ matrix is not singular. Moreover, the lineal constraints functions verify the constraint qualification for any value of U , **then the $\nabla_{UU}^2 \mathcal{L}(U^*)$ matrix is unfailingly positive defined for any values of U^* .**

Note: in this case the Hessian matrix with linear constraints is equivalent to the Hessian matrix obtained from unconstrained optimization problems.

9.7. APPENDIX III

Constraint qualification

To verify the constraint qualification is obviously necessary to analyses the constraint functions defined in (9.16), (9.17), (9.18) and (9.19):

$$r_{k-1}(U) = u_{ij}(K) - U_{min} = 0 \quad (9.46)$$

$$r_k(U) = U_{max} - u_{ij}(K) = 0 \quad (9.47)$$

$$r_{k+1}(U) = dU_{max} - (u_{ij}(K + 1) - u_{ij}(K)) = 0 \quad (9.48)$$

$$r_{k+2}(U) = dU_{max} - (u_{ij}(K) - u_{ij}(K + 1)) = 0 \quad (9.49)$$

$$r_{k+3}(U) = d_0U_{max} - (u_{ij}(K) - u_{(j-1)i}(K - 1)) = 0 \quad (9.50)$$

$$r_{k+4}(U) = d_0U_{max} - (-u_{ij}(K) + u_{(j-1)i}(K - 1)) = 0 \quad (9.51)$$

Where $r_k(U)$ is the “kth” constraint function, U_{min} and U_{max} is the minimum and maximum gate position allowed, dU_{max} is the maximum gate movement between consecutive time steps, the d_0U_{max} is the maximum gate movement between consecutive time steps for different prediction horizons.

Before analyzing the constraint function, we have to point that the theory for constrained optimization problem only consider the active constraints to calculate the solution U , and other restrictions are essentially ignored.

The first two overall constraints (9.46) and (9.47) exclude each other because, if a gate is totally open, so the constraint (9.47) is active, it cannot be totally close, so the constraint (9.46) cannot active.

The functional restrictions (9.48) and (9.49) exclude each other too because, these restrictions make reference to the gate movements between consecutive regulation time steps of a particular gate, and these constraints are completely opposed. The tendency of a gate movement should be to close or open, not all of them at the same time, so if the constraint function (9.49) is active to restrict the maximum gate opening so the constraint function (9.48) would not be active.

The overall constraints (9.50) and (9.51) is the same case that the last. These constraint functions are completely opposed. These constraints only can be active in the first regulation period for a prediction horizon.

In summary, you can ensure that the constraint qualification is verified for whatever combination of position values of the gates. For two reasons, the first one is that all constraints are linear and the second one is that the active constraints are linearly independent.

9.8. APPENDIX IV

Lagrange Newton Method

We can define the constrained optimization problem as:

$$\text{Minimize } J(U) = \frac{1}{2} [[C]X_{ki}^{kF}(U) - Y^*]^T [Q] [[C]X_{ki}^{kF}(U) - Y^*] \quad (9.52)$$

$$r_k(U) = u_{ij}(K) - U_{min} \geq 0$$

$$r_k(U) = U_{max} - u_{ij}(K) \geq 0$$

$$r_k(U) = dU_{max} - [u_{ij}(K+1) - u_{ij}(K)] \geq 0$$

$$r_k(U) = dU_{max} - [u_{ij}(K) - u_{ij}(K+1)] \geq 0$$

$$r_k(U) = d_0U_{max} - [u_{ij}(K) - u_{i(j-1)}(K-1)] \geq 0$$

$$r_k(U) = d_0U_{max} - [u_{i(j-1)}(K-1) - u_{ij}(K)] \geq 0$$

where U_{min} and U_{max} is the minimum and maximum gate position, dU_{max} is the maximum gate movement between consecutive time steps, the d_0U_{max} is the maximum gate movement between consecutive time steps for different prediction horizons.

The Lagrange-Newton method solves a constrained optimization problem from the Lagrange function as follow:

$$\mathcal{L}(U, \lambda) = J(U) - \sum_{i=1}^{nt} \lambda_k r_k(U) \quad (9.53)$$

where $r_k(U)$ is the “kth” constraint, λ_k is the “kth” Lagrange multiplier and n_i is the number of constraints.

We could establish the minimum of the Lagrange function as follow:

$$\nabla \mathcal{L}(U, \lambda) = 0 \quad (9.54)$$

Taking into account the Kuhn-Tucker conditions, it is possible to rewrite the optimization problem defined by the Lagrange function from Taylor series for U , λ , and the result of this method is a sequence of approximations U , λ (9.55) to get the solution vector U^* and the vector of optimum Lagrange multipliers λ^* :

$$\nabla \mathcal{L}(U + \Delta U, \lambda + \Delta \lambda) = \nabla \mathcal{L} + [\nabla^2 \mathcal{L}] \begin{pmatrix} \Delta U \\ \Delta \lambda \end{pmatrix} + \dots \quad (9.55)$$

where $\nabla \mathcal{L} = \nabla \mathcal{L}(U, \lambda)$. Neglecting higher order terms and setting the left hand size to zero, that is, $\nabla \mathcal{L}(U + \Delta U, \lambda + \Delta \lambda) = 0$, because $U + \Delta U$ and $\lambda + \Delta \lambda$ is the solution of the problem, gives the iteration procedure:

$$[\nabla^2 \mathcal{L}] \begin{pmatrix} \Delta U \\ \Delta \lambda \end{pmatrix} = -\nabla \mathcal{L} \quad (9.56)$$

where $\nabla^2 \mathcal{L}$ is the Hessian matrix of the Lagrange function, and $\nabla \mathcal{L}$ is the first derivative of the Lagrange function.

where:

$$[\nabla^2 \mathcal{L}] = \begin{bmatrix} W & -\nabla_U r^T \\ -\nabla_U r^T & 0 \end{bmatrix} \quad (9.57)$$

$$W = \nabla_{UU}^2 J(U) - \sum_{i=1}^{nt} \lambda_i [\nabla_{UU}^2 r_i] \quad (9.58)$$

$$\nabla \mathcal{L} = \begin{pmatrix} \nabla_U J(U) - \sum_{i=1}^{nt} \lambda_i [\nabla_U r_i] \\ r \end{pmatrix} \quad (9.59)$$

So the system is written as follow:

$$\begin{bmatrix} W & -\nabla_U r^T \\ -\nabla_U r^T & 0 \end{bmatrix} \begin{pmatrix} \Delta U \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} \nabla_U J(U) - \sum_{i=1}^{nt} \lambda_i [\nabla_U r_i] \\ r \end{pmatrix} \quad (9.60)$$

In fact it is more convenient to write $\lambda^{k+1} = \lambda^k + \Delta \lambda$, and so to solve the equivalent system:

$$\begin{bmatrix} W & -\nabla_U r \\ -\nabla_U r^T & 0 \end{bmatrix} \begin{pmatrix} \Delta U \\ \lambda \end{pmatrix} = \begin{pmatrix} \nabla_U J(U) \\ r \end{pmatrix} \quad (9.61)$$

Once solved the system and found the updated solution, $U = U + \Delta U$ and λ , the solution must be checked so it must be fulfilled that $\nabla \mathcal{L}(U, \lambda) = 0$, and if this is not accomplished, the Hessian matrix and the gradient vector must be recalculated.

We can do a simplification in matrix W , because all constraints are linear in this problem, and the element $\nabla_{UU}^2 r_i = 0$, and the matrix W can be written in the form $W = \nabla_{UU}^2 J(U)$ which is the Hessian matrix for unconstrained optimized problem.

The system (9.61) has to be solved to get the solution ΔU . The problem is that we do not know initially which constraints are active, and it is not possible to limit δU . We cannot be sure that the solution $U^{k+1} = U^k + \Delta U$ is in a feasible region and the constraint optimization problem could become an unconstrained optimization problem. For this reason, we use the Active Set Method to keep the solution in the feasible region defined by constraints.

Active Set Method

The Active set method was introduced to the algorithm code nested into the Lagrange-Newton method. The aim of the Active Set method is to compute a ΔU solution into the feasible region. The active set method never computes directly ΔU , if not an approximation ($\Delta \Delta U$), where $\Delta U = \Delta U + \alpha \Delta \Delta U$, in this way the method does never compute a ΔU solution outside the feasible region. The active method is very useful when we want to solve a constrained optimization problem.

The idea underlying in the Active set method is to set inequality constraints in two groups which are treated as active, while all other variables are treated as inactive. The inactive constraints are essentially ignored, as the solution ΔU is not restricted by these constraints. We can define the constrained optimization problem as follow:

$$\begin{aligned} \text{Minimize } & J(U + \delta U) \\ & r(U + \Delta U) \geq 0 \end{aligned} \quad (9.62)$$

The system (9.61) can be seen established as a quadratic problem, which consists in replacing the functions dependent on U by the respective linearization around U , establishing the following quadratic optimization problem:

$$J(U + \Delta U) = J(U) + \nabla_U J(U) \Delta U + \frac{1}{2} \Delta U^T [\nabla_{UU}^2 J(U)] \Delta U \quad (9.63)$$

$$r(U + \Delta U) = r(U) + [\nabla_U r(U)]^T \Delta U$$

If we establish the Lagrange function, we can write the system as follow:

$$\mathcal{L}(\Delta U, \lambda) = J(U) + \nabla_U J(U) \Delta U + \frac{1}{2} \Delta U^T [\nabla_{UU}^2 J(U)] \Delta U \quad (9.64)$$

$$- (r(U) + [\nabla_U r(U)]^T \Delta U) \lambda$$

We must remember that $r(U) = 0$, because the solution (U) is always inside the feasible region, so only the active constraints are evaluated and these are only equality constraints.

If we establishes that $\nabla \mathcal{L}(\Delta U, \lambda) = 0$, we can write the system as follow:

$$\nabla_U J(U) + [\nabla_{UU}^2 J(U)] \Delta U - [\nabla_U r(U)] \lambda = 0 \quad (9.65)$$

$$- [\nabla_U r(U)]^T \Delta U = 0$$

The system of equation (9.65) is nonlinear, for this reason, we have to solve it iteratively as follow:

$$\begin{bmatrix} \nabla_{UU}^2 J(U) & -\nabla_U r \\ -\nabla_U r^T & 0 \end{bmatrix} \begin{pmatrix} \Delta \Delta U \\ \Delta \lambda \end{pmatrix} = - \begin{pmatrix} \nabla_U J(U) + \nabla_U J(U) \Delta U + [\nabla_U r(U)] \lambda \\ 0 \end{pmatrix} \quad (9.66)$$

$$\Delta U = \Delta U + \Delta \Delta U$$

$$\lambda = \lambda + \Delta \lambda$$

In fact it is more convenient to write $\lambda^{k+1} = \lambda^k + \Delta \lambda$, and so to solve the equivalent system:

$$\begin{bmatrix} \nabla_{UU}^2 J(U) & -\nabla_U r \\ -\nabla_U r^T & 0 \end{bmatrix} \begin{pmatrix} \Delta \Delta U \\ \lambda \end{pmatrix} = - \begin{pmatrix} \nabla_U J(U) + \nabla_U J(U) \Delta U \\ 0 \end{pmatrix} \quad (9.67)$$

$$\Delta U = \Delta U + \Delta \Delta U$$

It is possible, that for any $\Delta U = \Delta U + \Delta\Delta U$, the solution $U = U + \Delta U$ was outside of the feasible set. For this reason, it is introduced the parameter α .

$$\Delta U = \Delta U + \alpha\Delta\Delta U$$

(9.68)

$$\alpha = \text{Min} \left[1, \text{Min}_{k \notin A \text{ and } 0 < \nabla_U r_k(U) \Delta\Delta U} \left(\frac{r_k(U) - \nabla_U r_k(U)^T \Delta U}{\nabla_U r_k(U)^T \Delta\Delta U} \right) \right]$$

So that $\Delta U = \Delta U + \alpha\Delta\Delta U$. If $\alpha < 1$ then a new constraint becomes active, defined by the index which achieves the minimum in the equation (9.68), and this index is added to the active set A. Then if the other constraints were satisfied and the Lagrange multipliers turned out to non-negative, the solution would be correct.

Chapter 10

GoRoSoBo application: canal with only one pool

In this chapter we present several numerical test-cases on a canal with GoRoSoBo algorithm which works together with CSI. GoRoSoBo is a feedback algorithm which operates in real time and the objective of this algorithm is to maintain the desired water level at checkpoints. The algorithm modifies the gate trajectories to fulfill this objective.

Under routine conditions, canal operations consist to control the scheduled delivery at each control structure. In case that a small disturbance for a short period of time is introduced into the canal during the irrigation cycle, the control strategy could be feedforward, as the flow is recovered quickly to the operative steady state. We are not worried about those disturbances. However, when an important disturbance (bigger than 10% or 20 % of the total flow rate) is introduced in the canal, we will need to use a feedback strategy. GoRoSoBo algorithm is tested in a canal with only one pool. We choose that kind of canal because it is easier to understand and evaluate the solution (gate trajectory) obtained by the algorithm and as a simple test to show the capacities of the proposed algorithm.

10.1. Introduction

We introduce different cases to test GoRoSoBo in this chapter. The objective is to check the accuracy of the control algorithm considering several kind of disturbances with different magnitude and duration under subcritical flow conditions.

The initial canal state is set from a known steady state, but during the irrigation cycle we introduce the scheduled deliveries and disturbances. The canal state will initially change to an unsteady state and later it will reach another steady state different than the original case. The control algorithm has to maintain the water level at the checkpoint to fulfill the forecasted demand. These easy examples test the accuracy and the good behaviour of GoRoSoBo and the associated CSI algorithm working all together.

10.2. Canal geometry

The geometry of the canal proposed in this chapter is based from Bautista's work (Bautista et al. 1997). This canal was used by different authors as Whyte (1969), Liu (1992), Chevereau (1991) and Soler (2003) and all of them tested feedforward controllers. The canal geometry adopted in our examples is based in the Liu's example as well as the scheduled demand which was introduced in some cases of this chapter. There are other authors who have also tested their feedback controllers in canals with only one pool similar to our canal, as Durdu (2009) or Litrico (2006). The canal geometry is shown at the Figure 10.1.

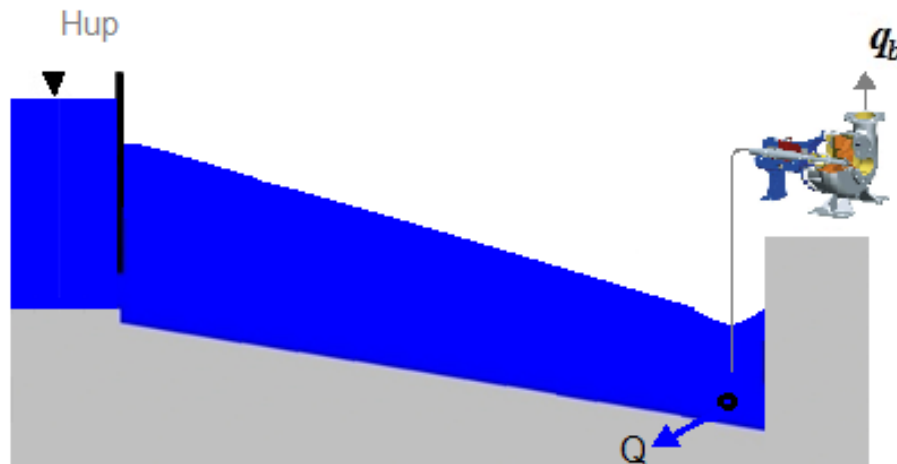


Figure 10.1: Simple canal with only one pool.

10.3. Canal features

According to the canal tested by Liu (1992), we show the features of this canal as follow:

- Geometry and canal parameters: one pool of 2500 meters length, and one sluice gate upstream the canal ($n_g=1$), the bottom slope is 0.001, width is 5 meters, bank slope (H:V) is 1,5:1 and the Manning coefficient is 0.025.

- Discretization: the spatial discretization of the canal is divided in cells of 25 meters, so there are 101 nodes ($n_s = 101$). The regulation period is 5 minutes and the duration of the predictive horizon is 8 hours, so there are 96 regulation periods ($K_F = 96$).
- Initial and boundary conditions: the upstream condition is fixed by a sluice gate that controls the water introduced into the canal from an upstream reservoir with a constant water level (3 meters). Other boundary conditions depend on the flow extracted by the orifice offtake and the disturbance which are introduced in the next section. At the end of the pool there is an orifice offtake, a lateral spillway and a pump station. The steady state is the initial condition of the canal at the first regulation period. Flow rate in the canal is $5\text{m}^3/\text{s}$ and the desired water level at the checkpoint is 1.6 meters (Figure 10.2).

Control structure	Numerical node upstream	Numerical node downstream	Gate discharge coefficient	Gate width (m)	Gate height (m)	Distance from the gate 1 (Km)
0	0	1	0.61	5	2.0	0
1	101	101	-	-	-	2.5

Table 10.1: Features of the canal control structures 1.

Control structure	Orifice offtake height (m)	Discharge coefficient/Orifice offtake diameter (m)	Lateral spillway height (m)	Lateral spillway discharge coefficient	Lateral spillway width (m)
0	-	-	-	-	-
1	0.8	2/0.85	2.0	1.99	500

Table 10.2: Features of the canal control structures 2.

There is a checkpoint at the last downstream node, the water level is measured at the checkpoint in each regulation period.

The orifice offtake are located downstream end of each pool where the extracted flow through the offtake (Q_{offtake}) is described as follow:

$$Q_{\text{offtake}} = K\sqrt{y - y_0} \quad (10.1)$$

$$K = Cd \times \frac{\pi \times \phi^2}{4} \times \sqrt{2g}$$

where y is the water level in canal at offtake, y_0 is the orifice offtake height equal to $y_{\text{target}}/2$, C_d is a discharge coefficient and Φ is the orifice offtake diameter.

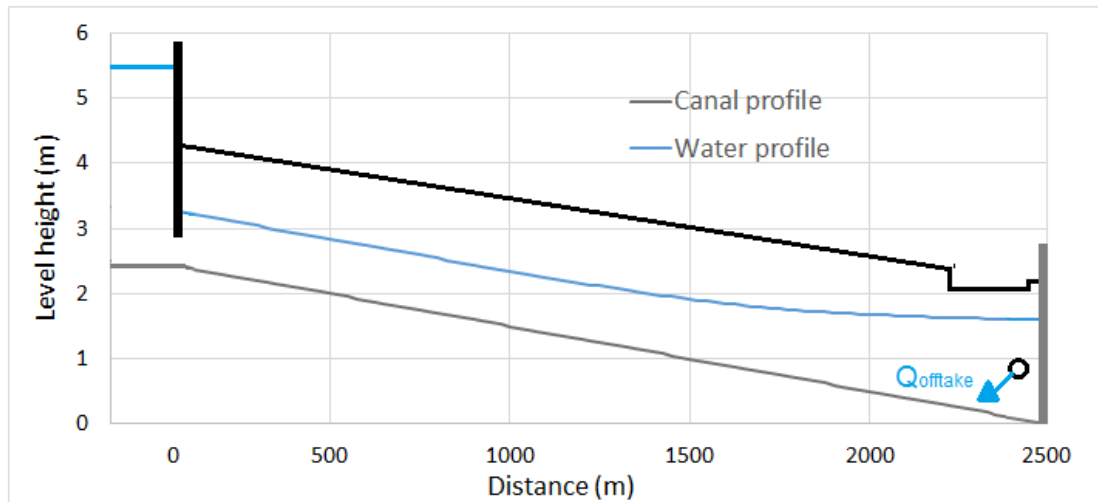


Figure 10.2: Backwater curve of the canal (Initial condition at the first regulation period of the irrigation cycle).

10.4. Problem setting

We test several cases to check the algorithm's behavior in front of different disturbance.

There are two different scheduled deliveries:

- The forecasted demand is constant with a value of $5 \text{ m}^3/\text{s}$ during the predictive horizon.
- The forecasted demand is variable in time, with a value of $5 \text{ m}^3/\text{s}$ for the first two hours, and $10 \text{ m}^3/\text{s}$ for the next two hours, and finally the forecasted demand returns to $5 \text{ m}^3/\text{s}$ during the last four hours of the test.

There are two different kinds of disturbances:

- The first one is called "variable disturbance" which is introduced thirty minutes after the test started and lasts 1 hour, with a value of $2 \text{ m}^3/\text{s}$.
- The second one is called the "step disturbance" which is introduced thirty minutes after the test started and lasts until the end of the test, with a value of $2 \text{ m}^3/\text{s}$.

There are four cases more, but these cases are the same of these last ones, the only difference lies in the value of the disturbance ($-2 \text{ m}^3/\text{s}$).

	Forecast demand	Disturbance
Case 1	constant	“step disturbance” (Figure 10.4)
Case 2	constant	“variable disturbance” (Figure 10.8)
Case 3	variable	“variable disturbance” (Figure 10.12)
Case 4	variable	“step disturbance” (Figure 10.15)
Case 5	constant	“negative step disturbance” (Figure 10.18)
Case 6	constant	“negative variable disturbance” (Figure 10.22)
Case 7	variable	“negative variable disturbance” (Figure 10.26)
Case 8	variable	“negative step disturbance” (Figure 10.30)

Table 10.3: The scheduled and unscheduled deliveries introduced in the cases tested

The cases, with a forecasted constant demand, are perfect to check how the algorithm is operating, as the only change in flow is due to the disturbance. It is easier to check the solution (the gate trajectory) obtained with GoRoSoBo and we can also check how the water level at the checkpoints recovers to the desired value. Possibly, the easiest case to check the gate trajectory solution is test case 1.

The test case 2 produces two big flow changes in the canal: when we introduce the “variable disturbance” and once it disappears. The algorithm has to readjust the gate position in a short period of time.

In test cases 3 and 4, the forecasted demand is not constant in time which in combination with the disturbances test the capacity of the GoRoSoBo algorithm to drive the canal to the desired state because the algorithm has to consider both flow changes.

In the rest of cases, we evaluate the same situation but the disturbance is negative. Depending on the value and the duration of the forecasted demand and the disturbance, some test cases are more difficult to solve for the algorithm than others. For instance in test case 7, the flow rate in the canal changes from $3 \text{ m}^3/\text{s}$ to $10 \text{ m}^3/\text{s}$ in only thirty minutes and this is a big problem for the feedback algorithm.

10.5. The target value

The control algorithm has to establish the target value for the canal maintaining a certain water level at checkpoint which is 1.6 meters. This water level target must be constant for the irrigation cycle. We show at Figure 10.3, the desired water level vector (yellow dots) defined at the checkpoint in every regulation period in a computational grid.

The duration of the test is 8 hour ($K_F=8h$). There is only one checkpoint downstream of the pool, so the numbers of water level targets are 96 ($n_T = n_t * K_F$).

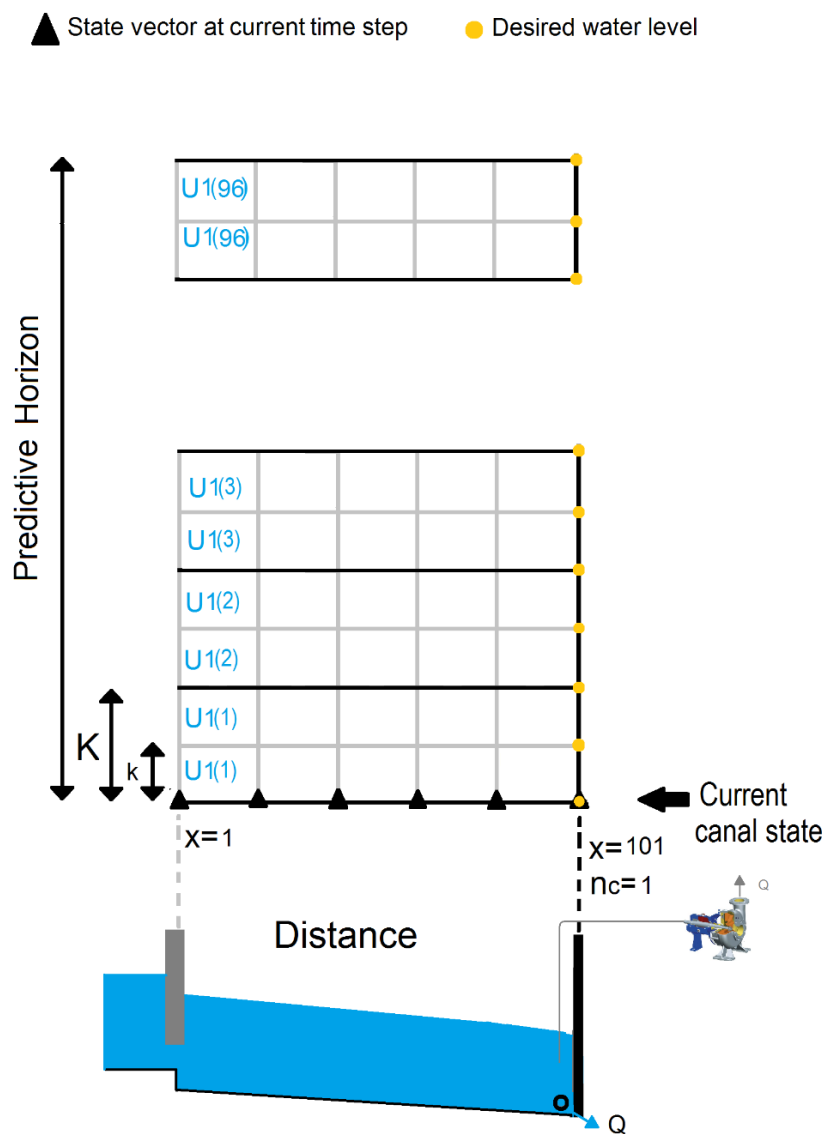


Figure 10.3: Time discretization of a test in the canal.

10.6. Constraints and trajectories

The sluice gate opening must not be greater or smaller than U_{\max} or U_{\min} , respectively and the gate movements between successive regulation periods (dU_{\max}) should be physically acceptable, as we introduced in chapter 9.

Taking into account the results obtained in the analysis of constrains did by Soler (2003) in these problems, we have imposed the overall and functional constraints on Table 10.4. The constraints are determined as a certain percentage of the gate height and this constraints are imposed to the gate position:

U_{\min} (%)	U_{\max} (%)	dU_{\max} (%)
0.05	90	2.5

Table 10.4: Overall and functional constraints values.

10.7. Test results

We represent the test results in several graphs. First graph represents the scheduled delivery and the disturbance during the irrigation cycle, the second one the disturbance calculated by CSI, the third one shows the gate trajectory in each regulation period during the irrigation cycle and the fourth one shows the water level at the checkpoint during the irrigation cycle.

We only show the disturbance calculated by CSI in cases that the disturbance is different than the others, because there are cases where we use the same kind of disturbance. If CSI is able to estimate the disturbance in one of them it is able to do it in the others.

In the first graph, we represent with a blue discontinuous line the scheduled delivery, and the disturbance represented by an orange continuous line. In the third and fourth graph, we represent with a blue discontinuous line the results obtained using a feedforward algorithm (GoRoSo) and with an orange continuous line the results obtained by GoRoSoBo algorithm. We can compare the sluice gate trajectories obtained by GoRoSoBo V.S. the sluice gate trajectories obtained by GoRoSo.

The gate trajectory is defined by the gate position in time, and the gate position is defined as follow:

$$\text{Gate position} = \frac{\text{Gate opening}}{\text{Gate height}} \quad (10. 2)$$

Case 1

We show in the next graph the scheduled and unscheduled deliveries introduced in this case:

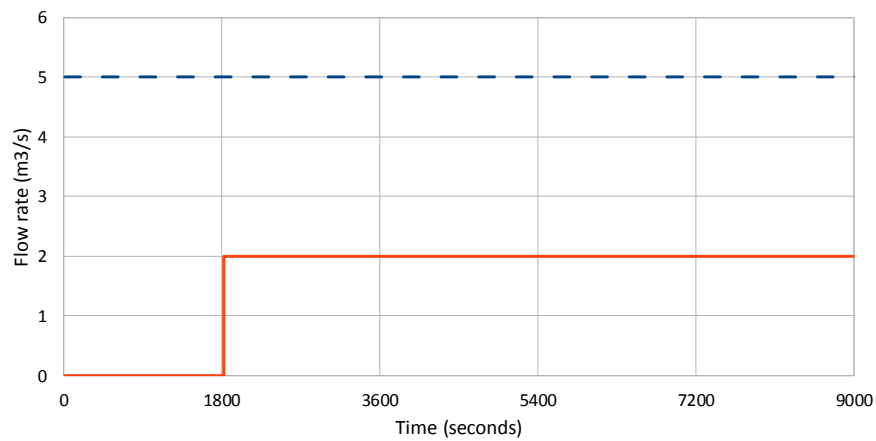


Figure 10.4: Scheduled and unscheduled deliveries introduced in the System (Case 1).

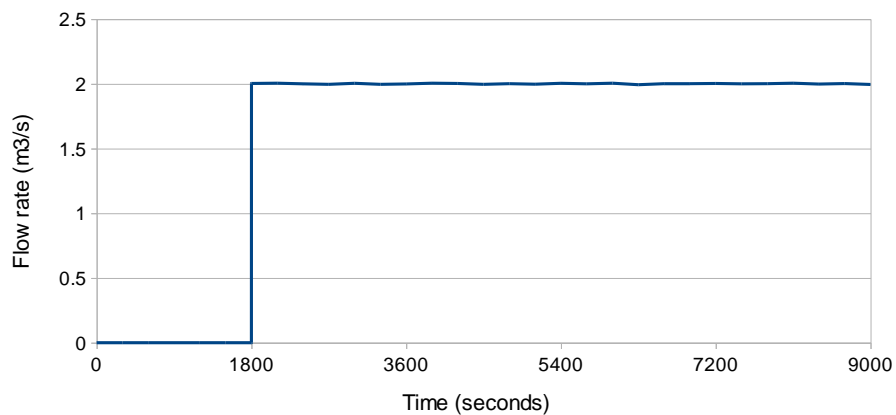


Figure 10.5: Disturbance calculated by CSI (Case 1).

At the beginning of the test, the flow in the canal is steady and the scheduled delivery is constant for all demand period, and the sluice gate position remains fixed. After the first thirty minutes (1800 s), a disturbance is introduced into the system, although the algorithm has not any notice until the next regulation period (2100 s) once the water level is measured at the checkpoint, and the water level has been already reduced in more than 10 cm, from 1.6 cm to 1.50 cm (Figure 10.7).

Once the water level measurements is send to CSI, it calculates the extracted flow vector for a past time horizon, in that sense CSI establishes the disturbance introduced in the system (Figure 10.5). All these information are sent to GoRoSoBo which modifies the sluice gate trajectory to keep constant the water level at the checkpoint which is not going to increase until three regulation periods later (2700 s). Because once the sluice gate generates a wave to increase the water level at the checkpoint, the wave has to travel at the checkpoint (time delay). Once the wave generated by the sluice gate arrives at the checkpoint, the water level increases quickly recovering the target water level of 1.6 meters at 3700 s (Figure 10.7).

The maximum deviation between the water level measured and desired is around 27.5 cm, from 1.6 cm to 1.325 cm, so the sluice gate movement must be quite important to reduce quickly this deviation (Figure 10.6). Gate position changes from 0.125 to 0.34 during a regulation period (1800-2100 s).

Instead the feedforward algorithm does not modify the gate trajectory during the irrigation cycle because the scheduled demand is constant and the algorithm does not know that someone has introduced a disturbance in the canal and the water level is decreasing at checkpoint. The water level decreases to 1.1 m so the flow extracted by the offtake is close to 3 m³/s (Figure 10.7). In that sense, the total extracted flow is 5m³/s, but 3 m³/s are extracted by the offtake and 2 m³/s are extracted by someone who introduced the disturbance. The difference in gate opening between the feedforward and the feedback algorithms at 9000s is due to the disturbance. A difference of 0.06 in gate opening represents a flow of 2 m³/s in this canal with this flow conditions.

Once the algorithm recovers the desired water level, the gate movements should be completely zero in this test. Although the water level is close to the desired value, it exists deviation between them of +/- 1 mm, these deviations between both water levels are due to

residual errors in the computations of both algorithms (CSI and GoRoSoBo), for that reason the gate movements are not completely zero.

This case 1 is the easiest case for GoRoSoBo and it is the most suitable to check the algorithm behaviour because the demand deliveries are constant in time and the disturbance too. We can check at Figure 10.7, as the control algorithm keeps constant the water level at checkpoint (1.6 m), in only twenty five minutes, and it recovers a water level error of 27.5 cm at the checkpoint.

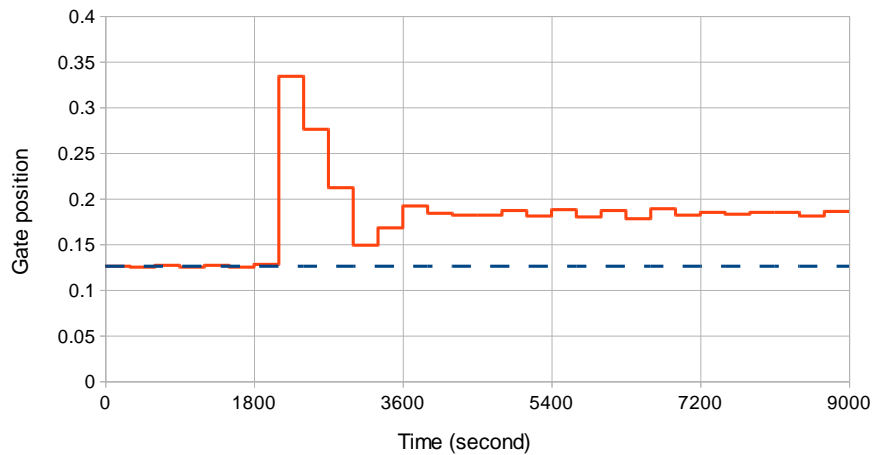


Figure 10.6: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 1.

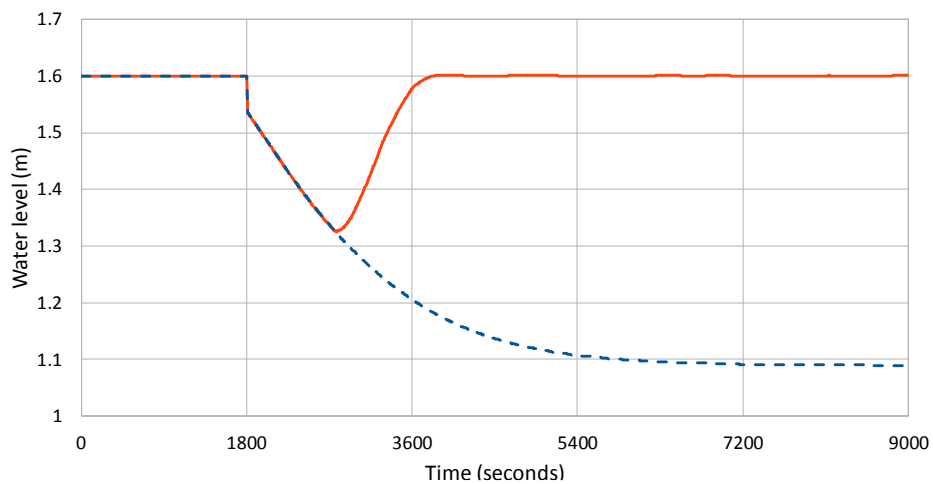


Figure 10.7: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 1.

Case 2

We show in the next graph the scheduled and unscheduled deliveries introduced in this case:

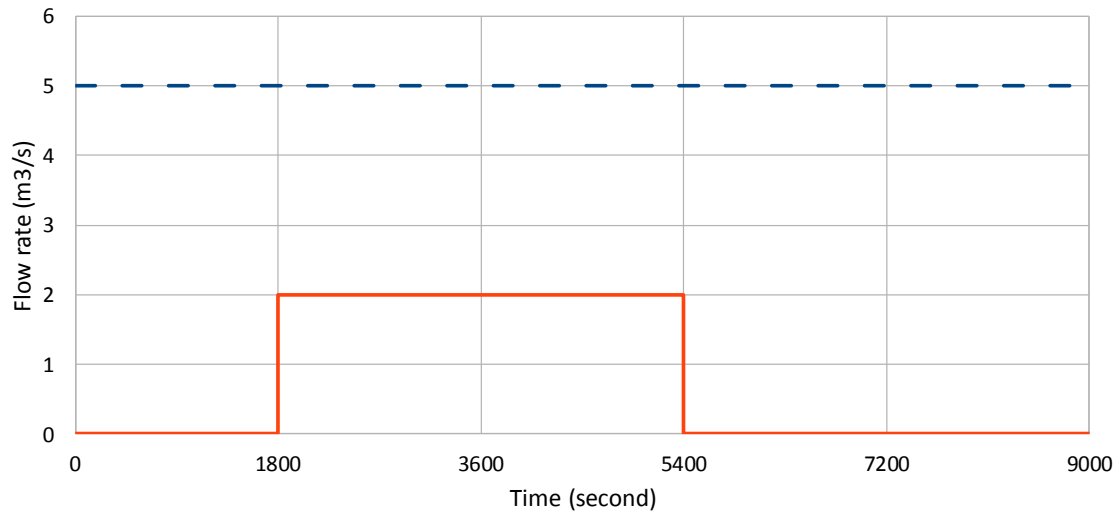


Figure 10.8: Scheduled and unscheduled deliveries introduced in the System (Case 2).

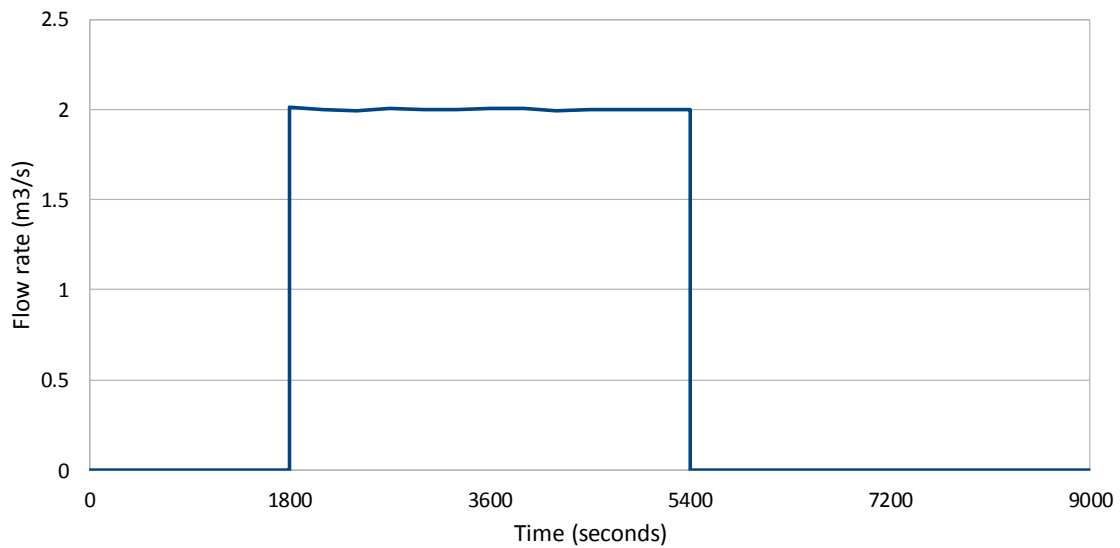


Figure 10.9: Disturbance calculated by CSI (Case 2).

This case is similar to the last case during the first 5400 s because we introduce a disturbance which only lasts one hour, as it is variable in time. The water level at the checkpoint decreases quickly from 1.6 cm to 1.32 cm (28 cm) in just 15 minutes as in the last test, see Figure 10.11.

When the disturbance is introduced to the canal (Figure 10.8), the sensor measures the water level at the checkpoint and this data is sent to CSI which calculates the disturbances considering the scheduled demand and the gate trajectory in the past time horizon (Figure 10.9).

All these data is sent to GoRoSoBo which calculates the gate position around of 0.34 at 2100 s (Figure 10.10) to increase the water level at the checkpoint. Once the disturbance disappears the water level at checkpoint increases too much at 5400 s (Figure 10.11) and CSI comes back to calculate the disturbances and send the information to GoRoSoBo which also calculates a new gate trajectory. The gate closes to reduce the flow rate introduced from the reservoir to the canal, and gate position is reduced from 0.34 to 0.0015 at 5700 s (Figure 10.10). The gate position comes back to 0.125 at 7500 s, because the flow conditions of the system come back to the initial conditions.

Instead the feedforward algorithm does not modify the gate trajectory during the irrigation cycle because the algorithm establishes an optimal gate trajectory at the beginning of the irrigation cycle considering just the scheduled deliveries. As there is not a feedback between the feedforward controller and the canal state, the algorithm does not know that the water level is decreasing at the checkpoint. The water level decreases until the flow extracted by the offtake is close to 3 m³/s (Figure 10.11), once the disturbance disappears the water level recovers its target value, because the flow conditions are equal to the initial conditions. In that sense, if the disturbance is constant, as in the last test, the feedforward algorithms is not able to return to the target value therefore the farmers only get from the orifice offtake 3 m³/s during the irrigation cycle.

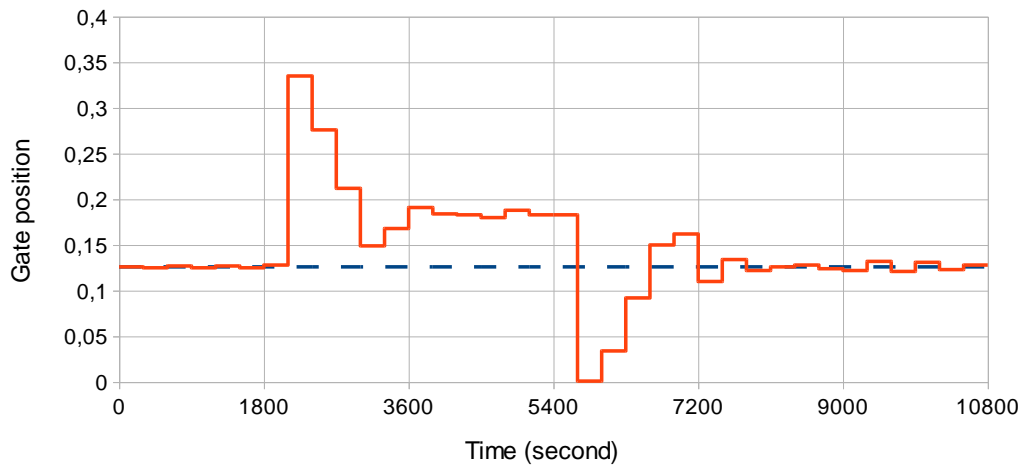


Figure 10.10: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 2.

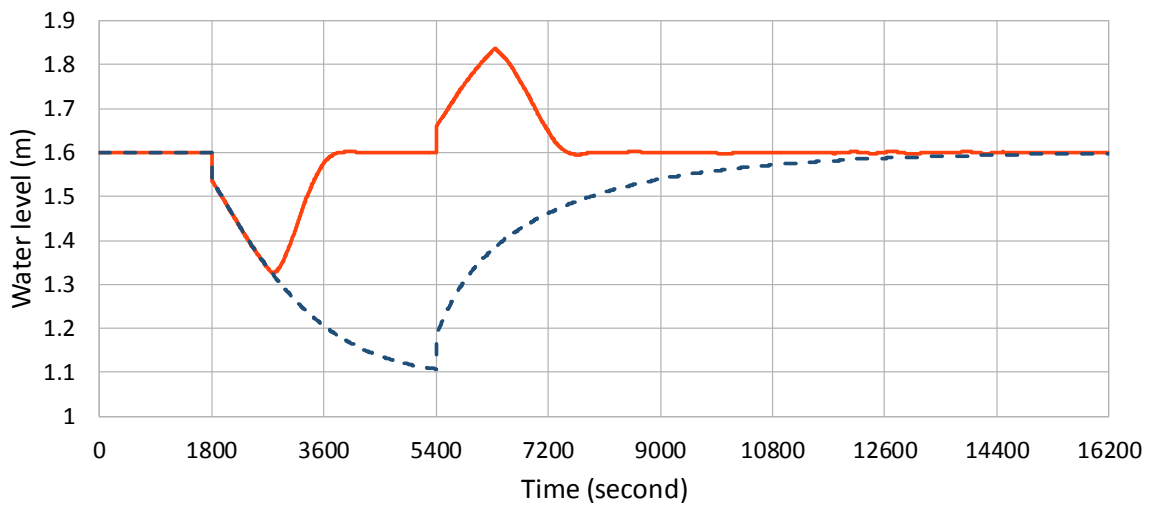


Figure 10.11: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 2.

Case 3

We show in the next graph the scheduled and unscheduled deliveries introduced in this case:

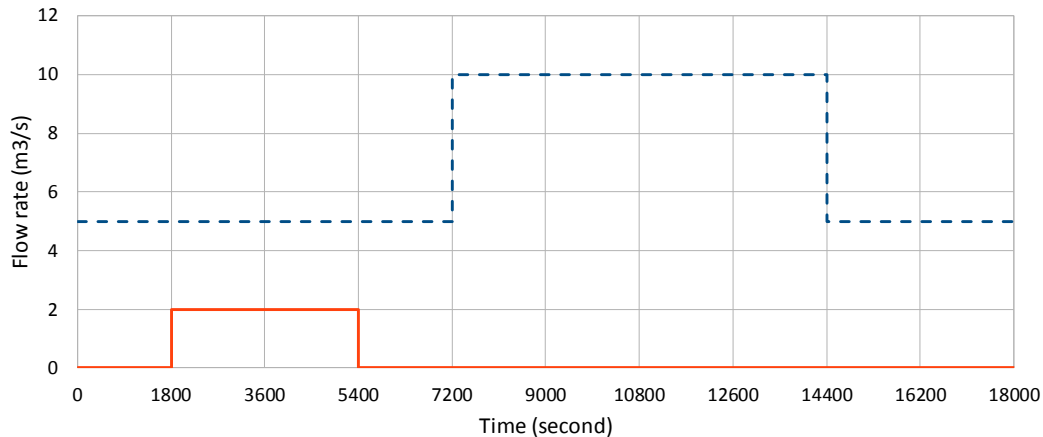


Figure 10.12: Scheduled and unscheduled deliveries introduced in the System (Case 3).

This case is similar to the last the first two hours so we will obtain the same sluice gate trajectory to reduce the deviation between the measured and desired water level.

As we have shown in the last cases, CSI is able to calculate the disturbance introduced into the system and that information is sent to GoRoSoBo which calculates a new gate position at 2100 s to drive the system to the desired state (Figure 10.13).

The feedforward algorithm does not change the gate trajectory once the disturbance is introduced in the canal and the water level at checkpoint decreases to 1.1 m at 5400 s. The water level starts to increase when the disturbance disappears at 5400 s. If we look at Figure 10.12, we can check that the scheduled delivery is not constant in time. For this reason, thirty minutes before changing the forecasted demand from 5 m³/s to 10 m³/s, the feedforward algorithm modifies the gate trajectory to reduce the future deviation between the measured and desired water level (Figure 10.13). The free surface has to be prepared to an important change in water demand, as the forecasted demand changes from 5 m³/s to 10 m³/s. As this change in flow has been scheduled, the feedforward algorithm opens the gate to provide more flow to the canal. As anybody introduces more disturbances in the canal during the irrigation cycle, the water level reaches the target value at 10800 s.

In a feedback strategy, GoRoSoBo has to consider the scheduled demands in addition to the disturbance. GoRoSoBo has to close the gate at time 5700 s (see, Figure 10.13), because the disturbance has disappeared at 5400 s, but at the same time GoRoSoBo has to open the gate because the forecasted demand will change from $5 \text{ m}^3/\text{s}$ to $10 \text{ m}^3/\text{s}$ at time 7200 s. The algorithm must take into account all these flow changes and maintains the desired water level (1.6 meters) at the checkpoint (Figure 10.14).

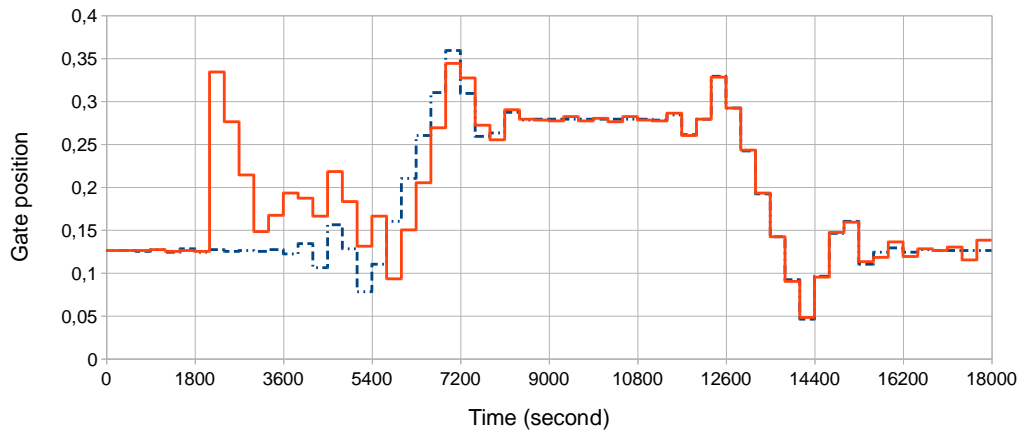


Figure 10.13: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 3.

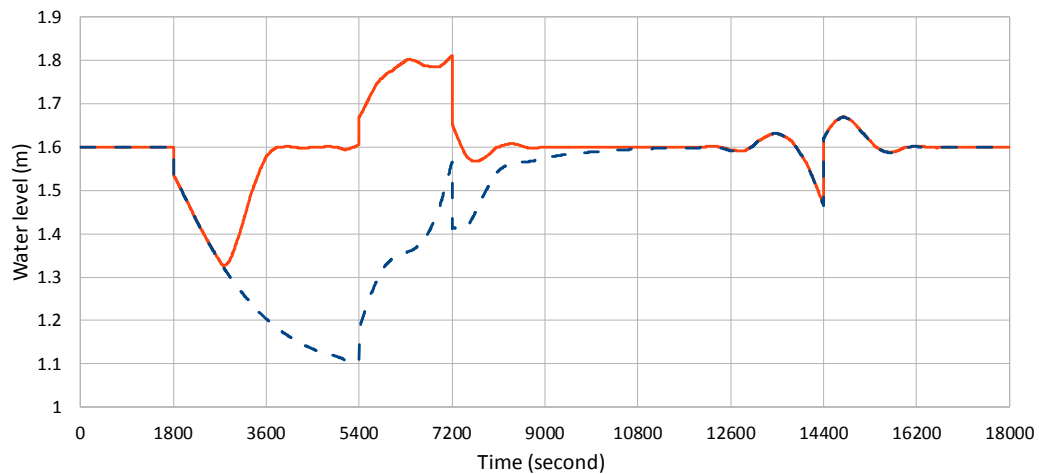


Figure 10.14: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 3.

Case 4

We show in the next graph the scheduled and unscheduled deliveries introduced in this case:

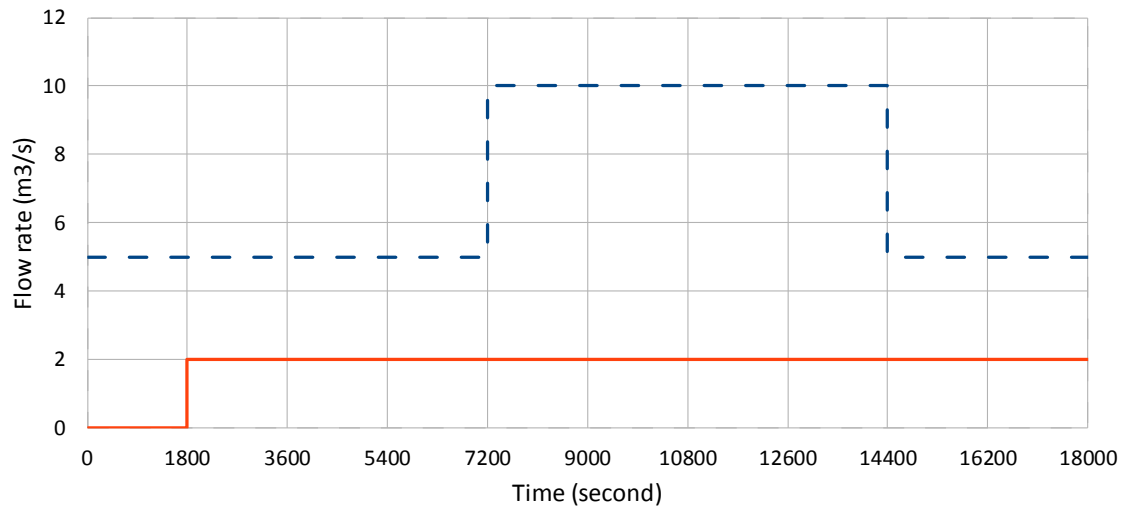


Figure 10.15: Scheduled and unscheduled deliveries introduced in the System (Case 4).

Here the disturbance is constant in time from the first thirty minutes and the forecasted demand is variable (Figure 10.15). The gate trajectory of GoRoSo is proportional to the gate trajectory of GoRoSoBo after the first hour (Figure 10.16), and the difference of both trajectories is due to the constant disturbance of 2 m³/s. Both water levels at checkpoint are proportional from the first four, see Figure 10.17.

The feedforward algorithm does not change the gate trajectory once the disturbance is introduced in the canal at 1800 s and the water level decreases to 1.1 m because the disturbance does not disappear until the end of the test. There is an increase in water demand from 5 m³/s to 10 m³/s which is considered by feedforward algorithm because it is scheduled but the water level never recovers the target value because the disturbance does not disappear during irrigation cycle.

We can check at Figure 10.16 that the difference in gate position of 0.06 between the feedforward and the feedback algorithms represents in the canal a flow of 2 m³/s with this flow conditions. In that sense, a difference in water level at checkpoint of 0.5 m also represents a flow of 2 m³/s (Figure 10.17).

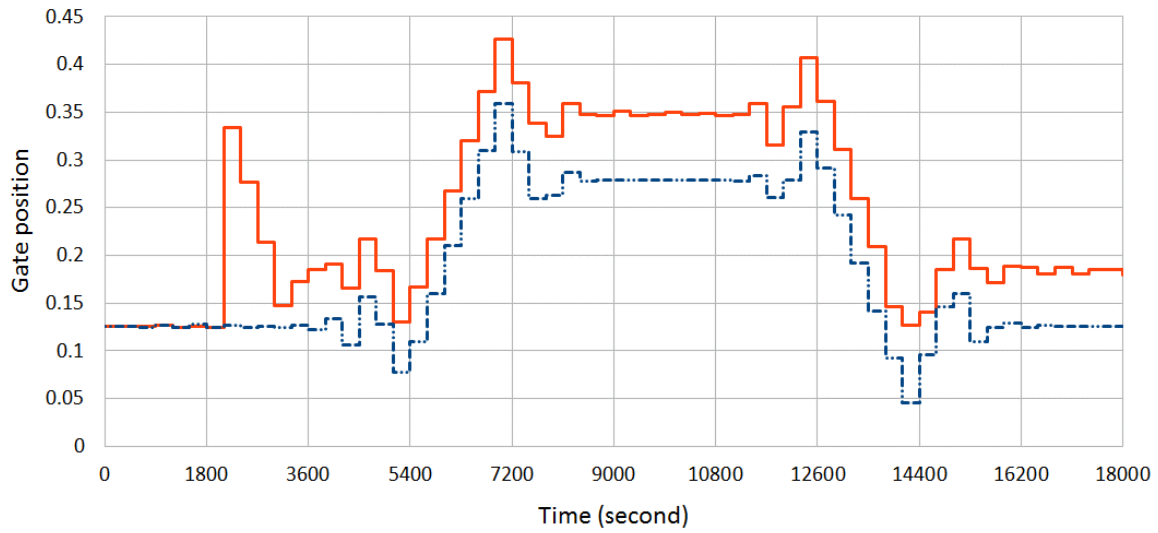


Figure 10.16: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 4.

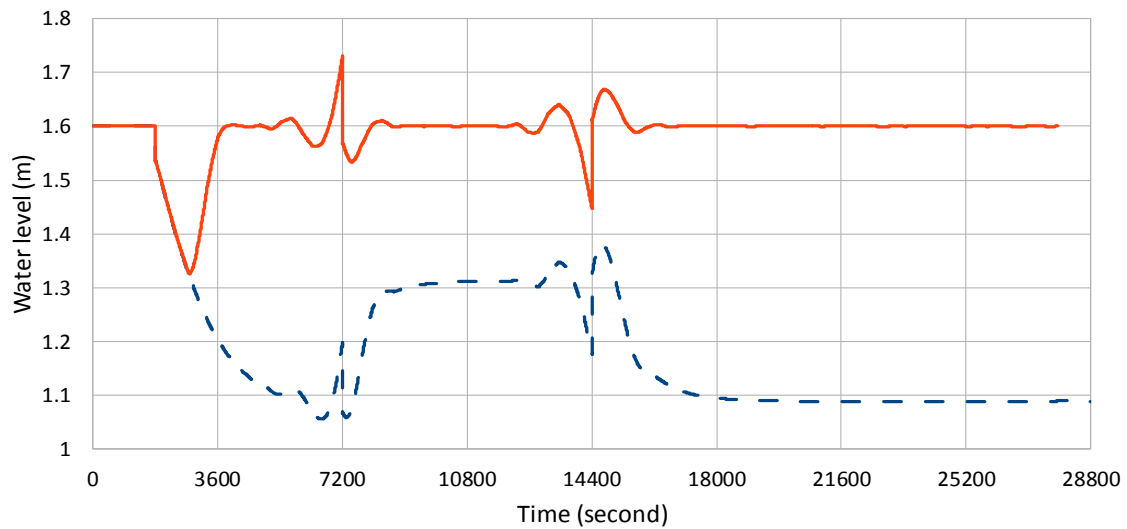


Figure 10.17: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 4.

Case 5

We show in the next graph the scheduled and unscheduled deliveries introduced in this case:

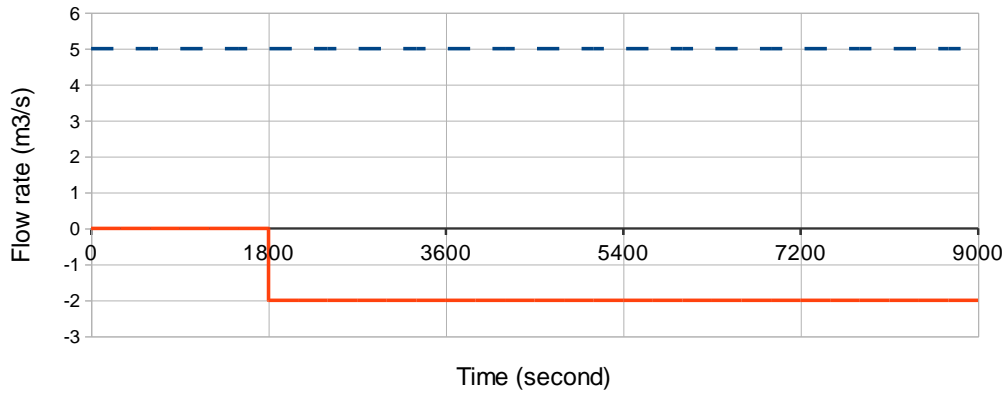


Figure 10.18: Scheduled and unscheduled deliveries introduced in the System (Case 5).

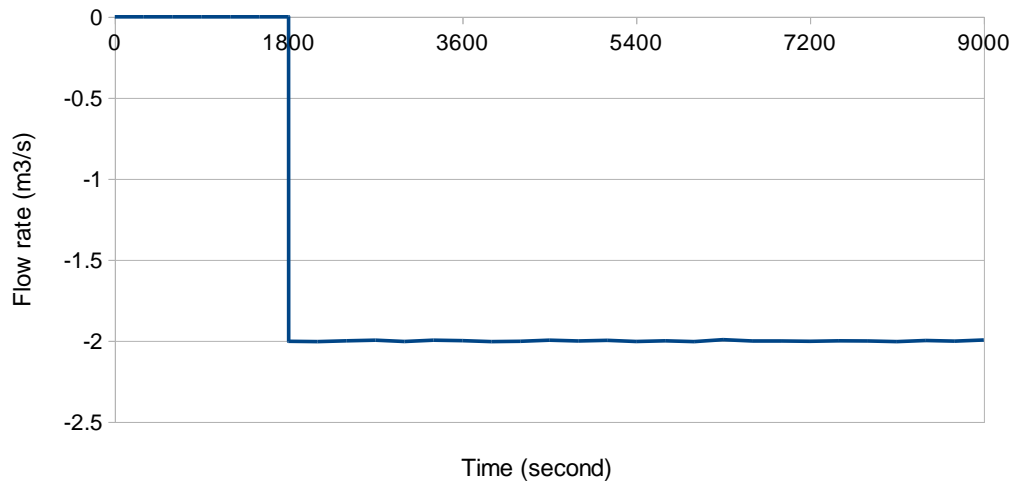


Figure 10.19: Disturbance calculated by CSI (Case 5).

This case is the same than the first one. The only difference between them is about the disturbance which has a negative value (Figure 10.18), so it is not an output flow but it is an input flow to the canal.

As we show at Figure 10.49, CSI is also able to estimate the disturbance when it has a negative value. So this information is sent to GoRoSoBo which modifies the sluice gate trajectory to keep constant the water level at the checkpoint which is not going to decrease until three regulation periods later at time 2700 s (Figure 10.21).

The gate trajectories obtained in case 5 are completely the opposite of the gates trajectories obtained in case 1, see Figure 10.20 and Figure 10.6, respectively. In this case, the gate position from 2100 s to 3000 s is the minimum allowed (U_{\min}), so the flow through the gate is the minimum flow allowed to reduce the water level at checkpoint. The water level trajectory at checkpoint is also completely opposite to the case 1, see Figure 10.21 and Figure 10.7, respectively.

As the feedforward algorithm has not any information about the negative disturbance, the algorithm does not calculate a new gate trajectory and the water level at the checkpoint increases over the canal, so there is canal overflow at 4614 s. As the disturbance does not disappear during the irrigation cycle, the lateral spillway operates until the end of the test.

The difference in gate position between the feedforward and the feedback algorithms at 9000s is due to the negative disturbance. A difference of 0.06 in gate position represents a flow of $-2 \text{ m}^3/\text{s}$ in the canal, as the disturbance has the same absolute value than in case 1, the difference in gate position is the same.

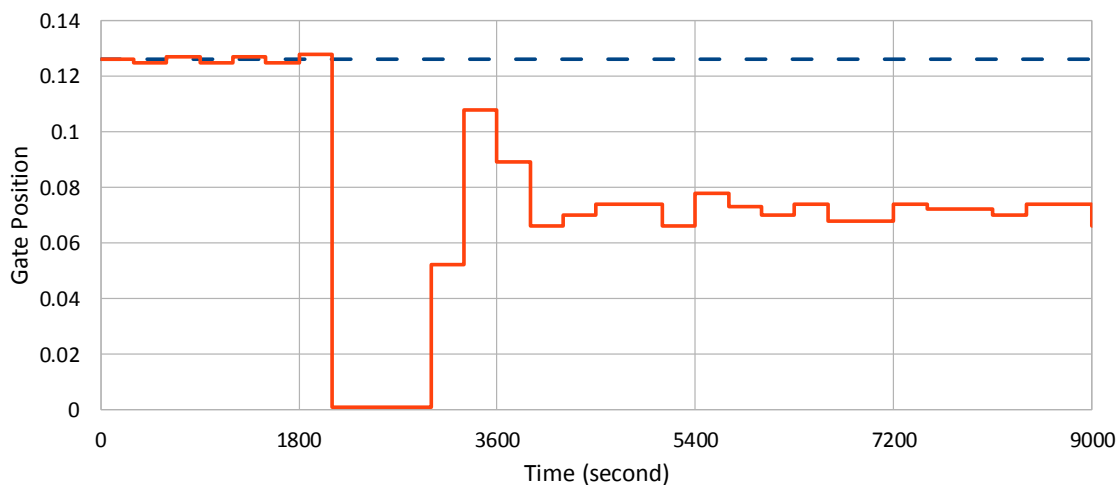


Figure 10.20: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 5.

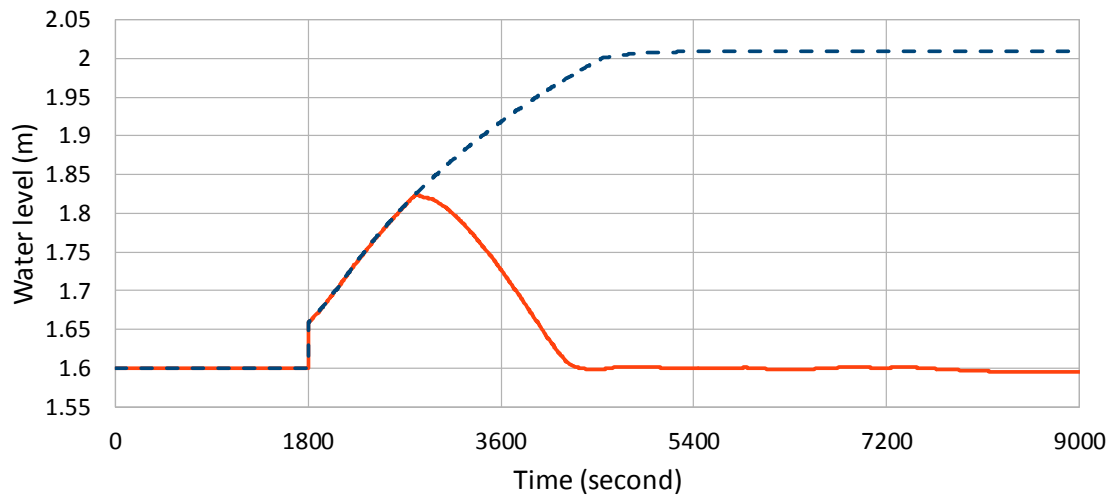


Figure 10.21: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 5.

Case 6

We show in the next graph the scheduled and unscheduled deliveries introduced in this case:

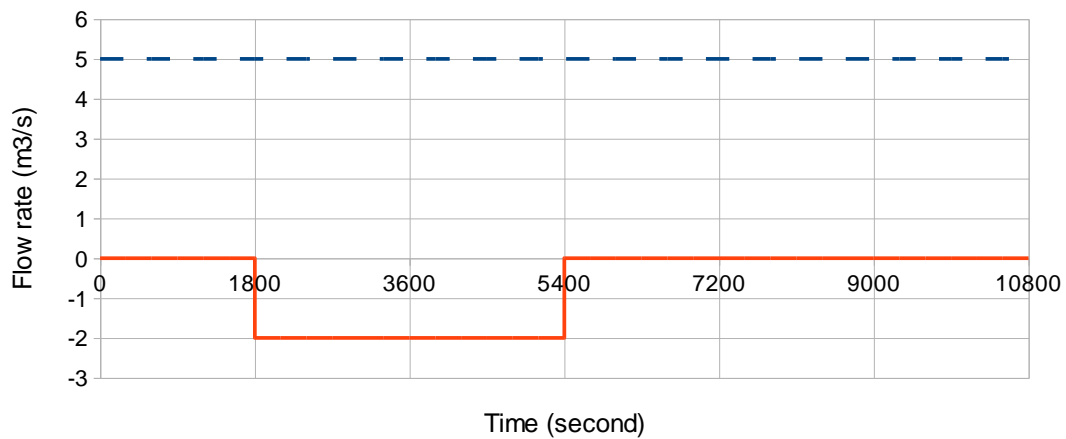


Figure 10.22: Scheduled and unscheduled deliveries introduced in the System (Case 6)

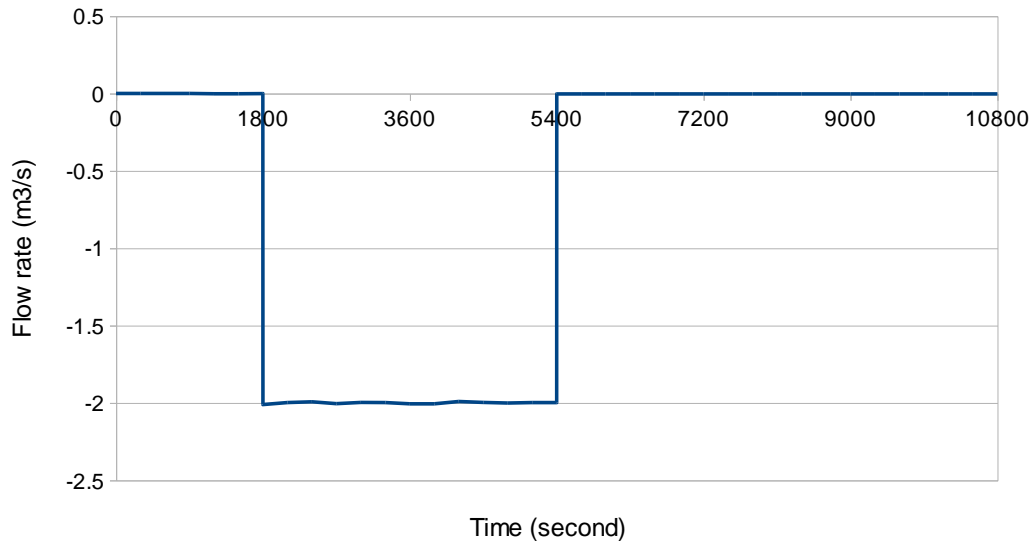


Figure 10.23: Disturbance calculated by CSI (Case 6).

This case is the same than the second case, the only difference between them is about the disturbance which has a negative value (Figure 10.22).

As we show at Figure 10.23, CSI is also able to estimate the disturbance when it has a negative and variable value. The water level increases at 1800 s due to the negative disturbance and GoRoSoBo closes the gate to reduce the water level at checkpoint, when the disturbance disappears at 5400 s, so GoRoSoBo has to open the gate because the flow through the gate is too small to deliver 5 m³/s through the offtake.

The gate trajectories obtained in case 6 are opposite to the gates trajectories obtained in case 2, see Figure 10.24 and Figure 10.10, respectively. In this case, the gate position reaches the minimum allowed (U_{\min}) from 2100 s to 3000 s for that reason this gate trajectory is not equal to the inverse gate trajectory of case 2. The same analogy could be made about the water level trajectories at checkpoint in both cases.

As the feedforward algorithm has not any information about the negative, variable disturbance, there is not any change in gate position so water level at the checkpoint increases to 2 m at 4614 s (Figure 10.25) so there is a canal overflow and the lateral spillway operates from 4614 s to 5400 s when the disturbance disappears and the water level decreases at checkpoint.

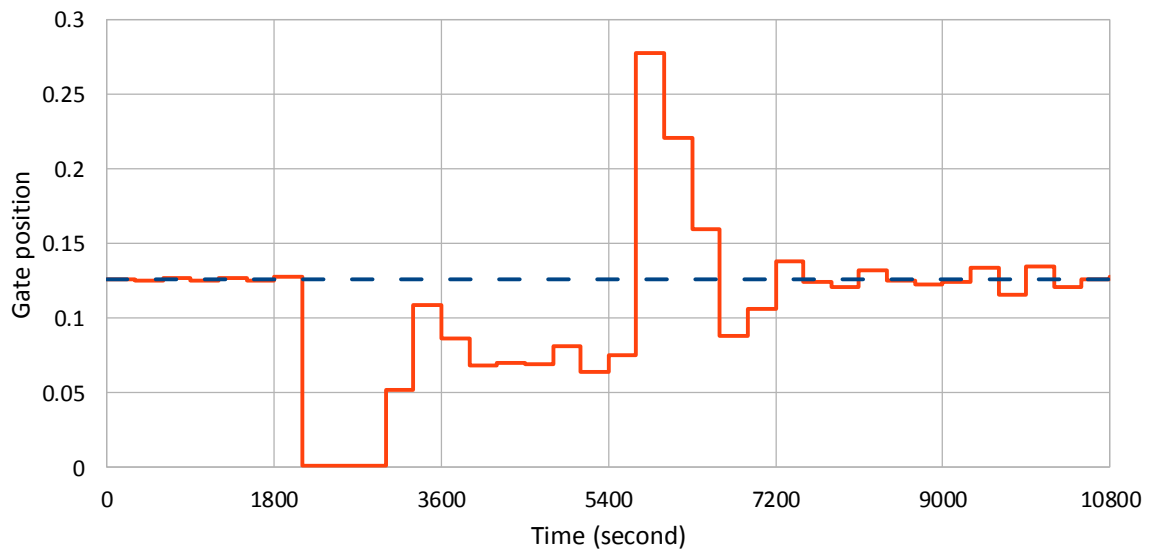


Figure 10.24: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 6.

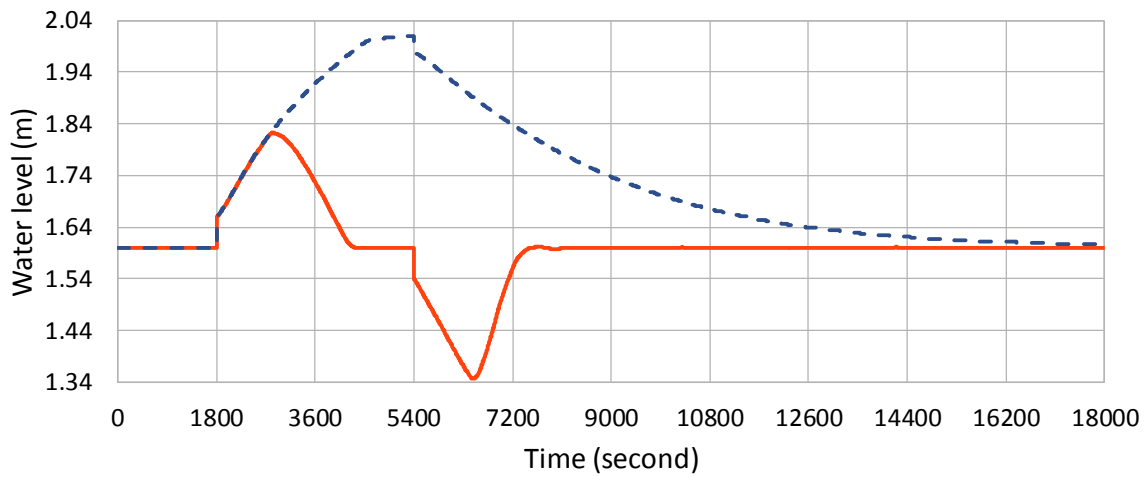


Figure 10.25: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 6.

Case 7

We show in the next graph the scheduled and unscheduled deliveries introduced in this case:

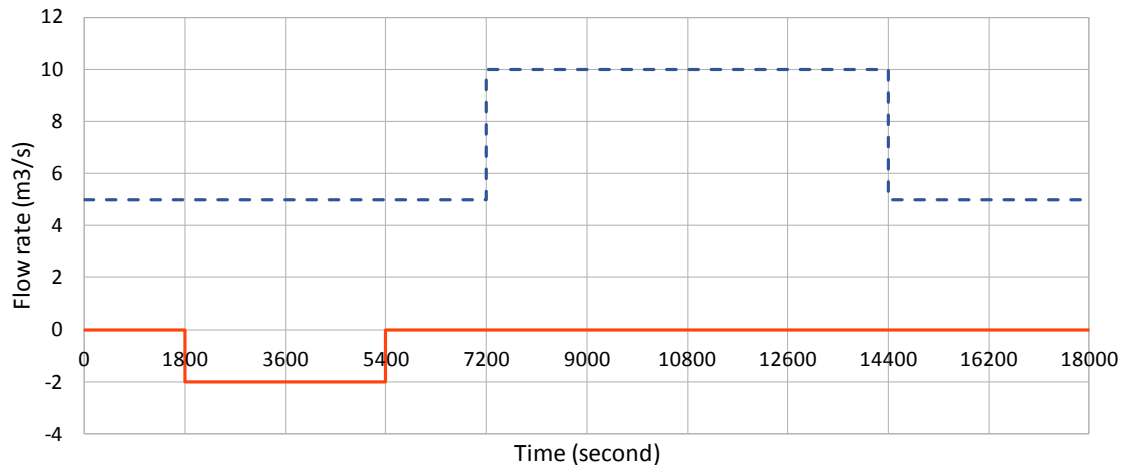


Figure 10.26: Scheduled and unscheduled deliveries introduced in the System (Case 7)

This case is the same than the third case, the only difference between them is about the disturbance (Figure 10.26). The gate trajectories are similar in the feedback and feedforward strategy after the first two hours, see Figure 10.27 and Figure 10.13, because the disturbance disappears at 5400 s and the only flow change is due to the forecasted demand.

As we show in Figure 10.28, the water level at checkpoint obtained using feedforward algorithm is not recovered until time 12000 s, that is, 6600 s after the disturbance disappears, instead the feedback algorithm recovers the water level at checkpoint before the disturbance disappears at 4000 s. We have to point that the enclosed area (A) by the water level trajectory (Figure 10.29) from 1800 s to 4000 s is similar to the enclosed area (B) from 5400 s to 6900s. We can conclude that the volume of water delivered to the farmers from 1800 s to 4000 s over the scheduled demand, because the water level is over the target value (1.6 m), it is similar to the volume of water not delivered from 5400 s to 6900 s, because the water level is under the target value.

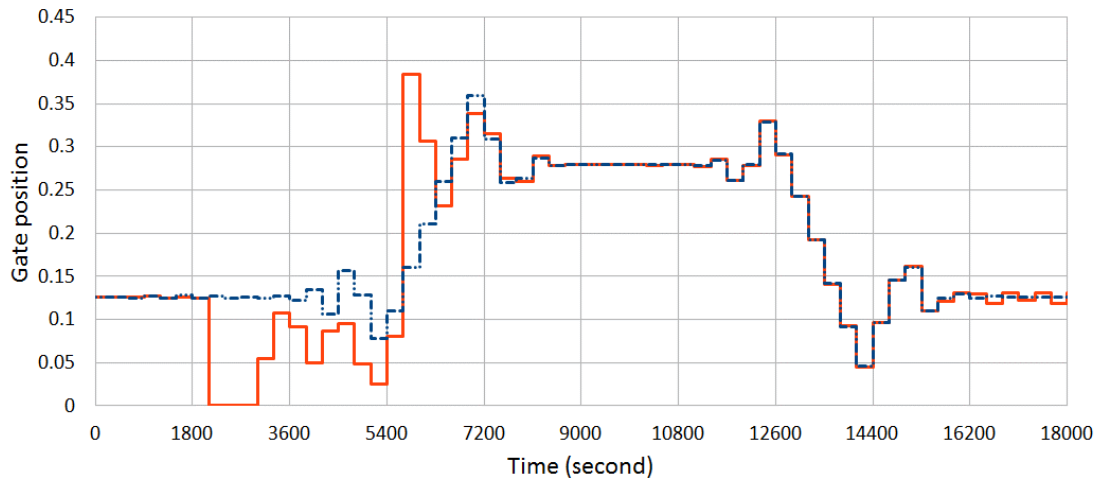


Figure 10.27: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 7.

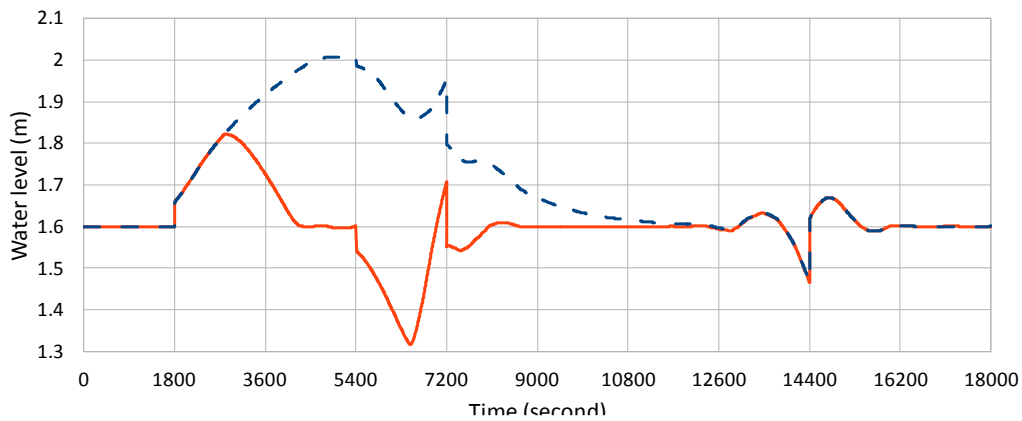


Figure 10.28: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 7.

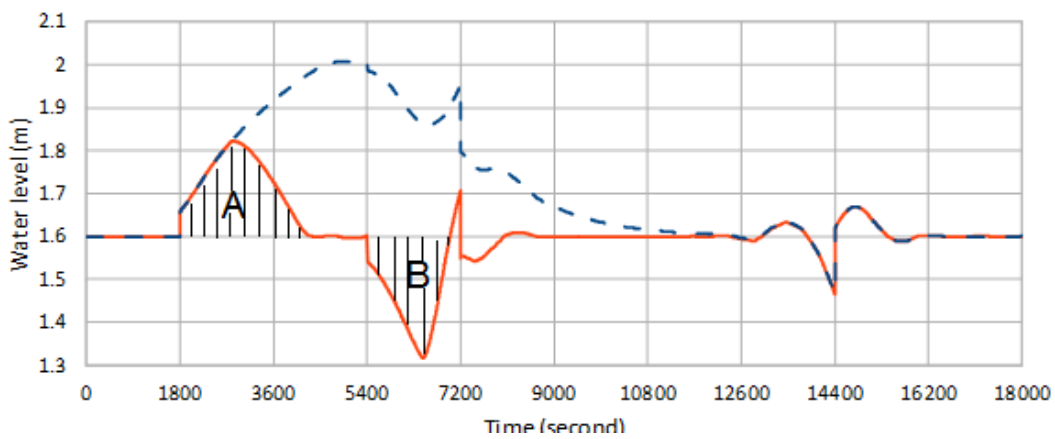


Figure 10.29: Enclosed area V.S. Water level at the checkpoint obtained by GoRoSoBo (orange continuous line) in Case 7.

Case 8

We show in the next graph the scheduled and unscheduled deliveries introduced in this case:

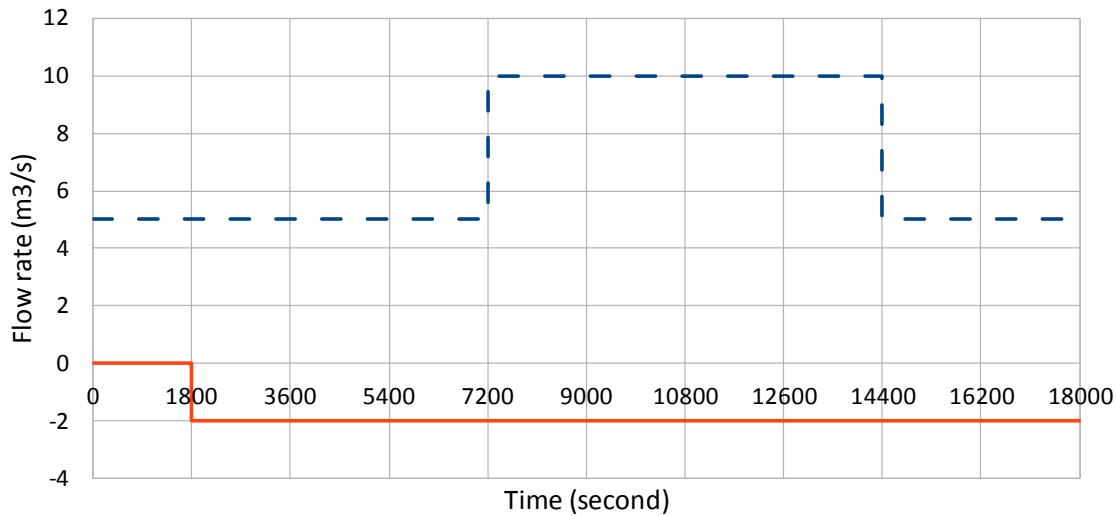


Figure 10.30: Scheduled and unscheduled deliveries introduced in the System (Case 8)

This case is the same than the fourth case, the only difference between them is about the disturbance which has a negative value (Figure 10.30). The gate trajectories are completely different in both cases the first 5400 seconds, and after they are proportional until the end of the test, see Figure 10.31 and Figure 10.16. The difference of both gate trajectories is due to the constant disturbance that increases the total flow rate in $2 \text{ m}^3/\text{s}$ in this case and it reduces the total flow rate in $2 \text{ m}^3/\text{s}$ in case 4.

As we show in Figure 10.31, there is an equivalence in gate position between the feedback and feedforward algorithm from 3900 s to 18000 s. The feedback algorithm calculates a gate position of 0.072 at 18000 s instead the feedforward algorithm calculates a gate position of 0.126 at 18000 s. The feedforward does not know that there is an inlet flow of $2 \text{ m}^3/\text{s}$ in the canal and the water level is 0.409 m over the target value (1.6 m), for this reason the flow delivered by the offtake is $6.147 \text{ m}^3/\text{s}$ when the scheduled demand is $5 \text{ m}^3/\text{s}$.

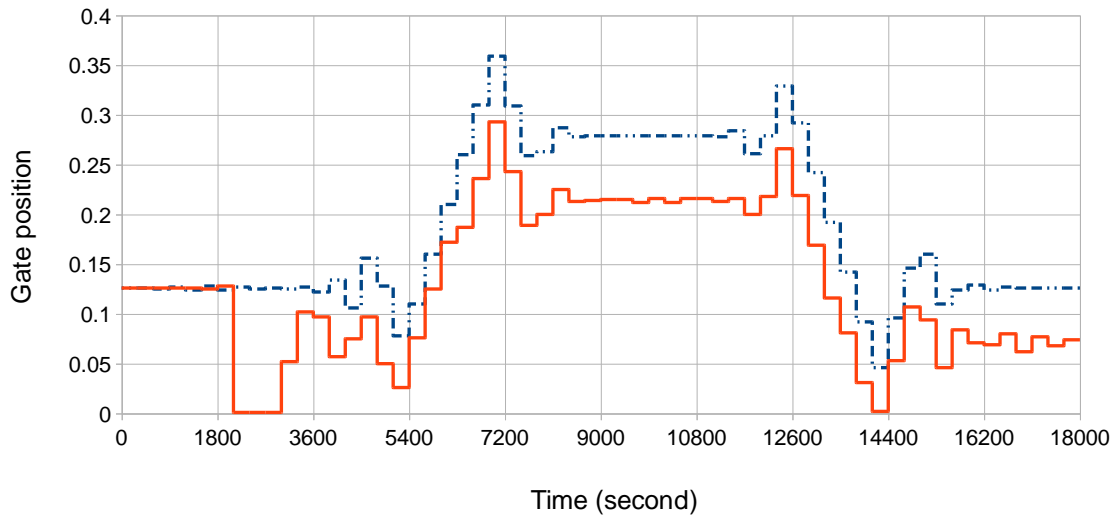


Figure 10.31: Gate trajectory obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 8.

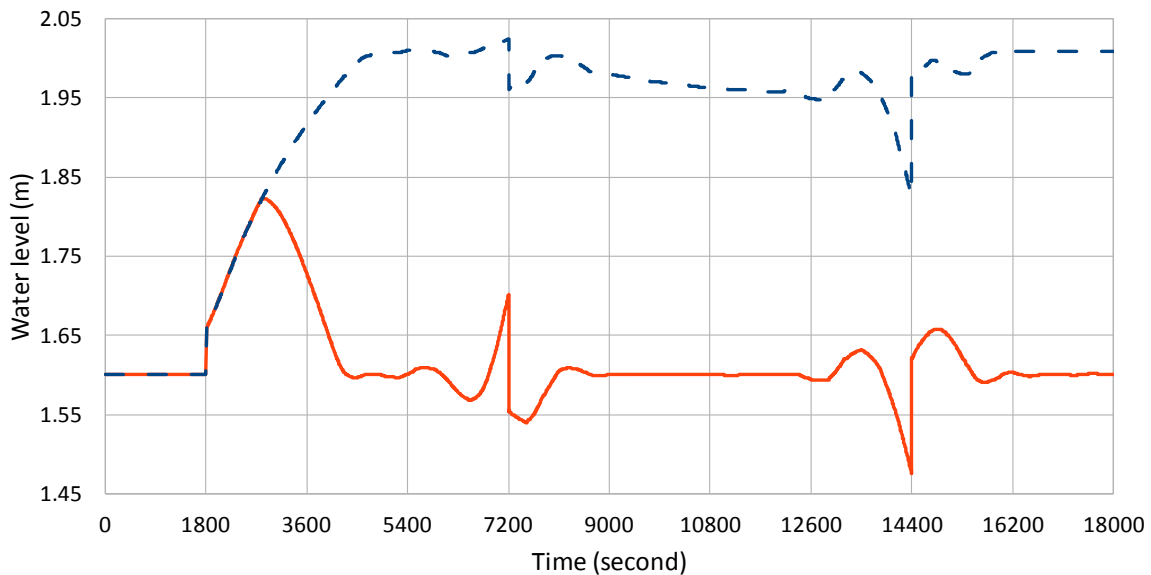


Figure 10.32: Water level at the checkpoint obtained by GoRoSoBo (orange continuous line)/GoRoSo (blue discontinuous line) in Case 8.

These last four test cases can be compared with the first four. We can check in cases where the forecasted demand is constant that the gate trajectories are opposite when disturbance is negative or positive.

10.8. Conclusions

Once the results of the different cases have been shown, some conclusions can be drawn:

1. Some test cases, introduced in this chapter, are hard to solve by a control algorithm, because the flow changes in forecasted demand and disturbance are 100 % and 40 %, respectively, over the total flow rate in the canal.
2. The algorithm computes the best sluice gate trajectory to maintain the water level at the checkpoint. In that sense, the algorithm maintains the forecasted demand at the delivery point although a disturbance perturbs the free surface.
3. To control the canal state in real time. The algorithm must calculate a new gate trajectory, in case that a disturbance modifies the water level at the checkpoint from the target. The changes in gate trajectories will be more important depending of the magnitude of this disturbance. For this reason, it is necessary to restrict the sluice gate movements which could be excessive in these cases.
4. CSI and GoRoSoBo operate together in all those test cases. The water level measurements, scheduled demand and gate trajectory are sent to CSI, in case that a disturbance is introduced into the system, CSI is able to establish this disturbance. All these data is sent to GoRoSoBo which calculates the best sluice gate trajectory to reach the desired behaviour at the checkpoint.
5. In the next chapter, more examples are going to be introduced to test the GoRoSoBo algorithm.

Chapter 11

ASCE Test Cases

We show in this chapter the application of the control algorithm in the test-cases proposed by the ASCE (Clemmens et al., 1998). The ASCE Task Committee on Canal Automation Algorithms proposed test-cases on two real canals for comparison of control algorithms. The test cases cover a set of conditions that may be found in practice. Two test schedules were adopted for each canal; during the first part in every of these four cases are introduced scheduled offtake changes and in the second part are introduced unscheduled offtake changes which are unknown for the algorithm. In this chapter as we want to evaluate GoRoSoBo, we only test the second part of every case.

The two canals proposed by the ASCE were chosen because they are completely different. Whereas the Corning canal is large with a mild slope, the lateral canal WM within Maricopa Stanfield district is small and steep with short response times. The aim in these tests is to check the robustness and accuracy of the control algorithm for different canals. The response of the algorithm should be good in both cases.

Two cases of the Test-Cases, proposed by the ASCE for feedback controllers, introduce huge changes on unscheduled flow deliveries in a short period of time. The aim is to check the capacity of the feedback controller to drive the canal state and remove the disturbance effect introduced by the unscheduled flow change. The test schedules strategies in both canals are similar, starting from steady state. The canal conveys a high flow-rate to deliver water to the farmers due to an important demand delivery. Two hours later of the beginning of the irrigation cycle, the demand delivery decreases sharply because the water demand is reduced to a minimum flow rate in all canal without knowledge of the watermaster. These Test-Cases are quite restrictive in two of them for most of the feedback controllers due to the changes on water demand are very important, for instance in Corning canal there are unscheduled offtake changes around of 4 m³/s. In the lateral canal WM, as it is a steep canal characterized by little storage, feedback controllers have even more problems to control the canal state due to the little storage capacity and the important changes on water demand.

The lateral outlet flow in the test is produced by orifice offtakes at the downstream end of each pool and by pump stations.

Detailed canal features are introduced in this chapter as well as the unscheduled and the scheduled deliveries for every test. The behaviour of GoRoSoBo is evaluated in each test and the results are shown in the next section compared with the results obtained by other feedback control algorithms as CLIS or Pilote. More information about CLIS and Pilote control algorithms can be found in Liu et al. (1998) and Malaterre et al. (1995).

11.1. Canal features

Test canal 1 is based on lateral canal WM within the Maricopa Stanfield Irrigation and Drainage District in central Arizona. Details on canal 1 can be found in Clemmens et al. (1994). This canal is a steep canal characterized by higher values of Froude number than canal 2, although both canals operate in subcritical flow. The canal 1 has a little storage capacity due to its small cross sections. Instead, the canal 2 is based on the upstream portion of the Corning Canal in California, and is much flatter with significant storage (Buyalski and Serforzo 1979). In both canals, the target depth is close to the normal depth for each pool, at maximum flow.

11.1.1. Maricopa Stanfield canal

The Maricopa Stanfield canal has 9.5 Km. It is divided in eight pools by rectangular gates. The orifice offtakes, emergency lateral spillways, pump stations and checkpoints or target points are located at downstream end of each pool. The canal geometry is represented in Figure 11.1 and the canal features are shown in Table 11.1 and Table 11.2.

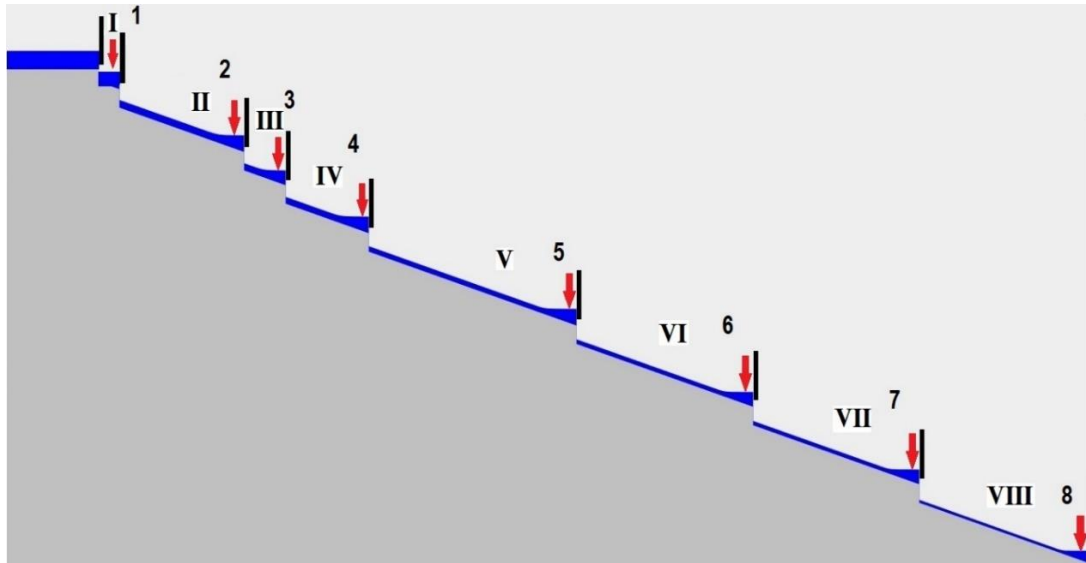


Figure 11.1: Maricopa Stanfield profile. The red lines mark the position of checkpoints. The first pool is number I and the first checkpoint is number 1.

Pool number	Numerical node upstream	Numerical node downstream	Pool length (Km)	Bottom slope	Side Slopes (H:V)	Manning's coefficient (n)	Bottom width (m)	Canal depth (m)
I	1	6	0.1	$2 \cdot 10^{-3}$	1.5:1	0.014	1	1.1
II	7	67	1.2	$2 \cdot 10^{-3}$	1.5:1	0.014	1	1.1
III	68	88	0.4	$2 \cdot 10^{-3}$	1.5:1	0.014	1	1.0
IV	89	129	0.8	$2 \cdot 10^{-3}$	1.5:1	0.014	0.8	1.1
V	130	230	2	$2 \cdot 10^{-3}$	1.5:1	0.014	0.8	1.1
VI	231	316	1.7	$2 \cdot 10^{-3}$	1.5:1	0.014	0.8	1.0
VII	317	397	1.6	$2 \cdot 10^{-3}$	1.5:1	0.014	0.6	1.0
VIII	398	483	1.7	$2 \cdot 10^{-3}$	1.5:1	0.014	0.6	1.0

Table 11.1: Features of Maricopa Stanfield pools.

Target points	Numerical node upstream	Numerical node downstream	Gate discharge coefficient	Gate width (m)	Gate height (m)	Step (m)	Distance from the gate 1 (Km)	Orifice offtake height (m)
0	0	1	0.61	1.5	1.0	1.0	0	-
1	6	7	0.61	1.5	1.1	1.0	0.1	0.45
2	67	68	0.61	1.5	1.1	1.0	1.3	0.45
3	88	89	0.61	1.5	1.0	1.0	1.7	0.40
4	129	130	0.61	1.2	1.1	1.0	2.5	0.45
5	230	231	0.61	1.2	1.1	1.0	4.5	0.45
6	316	317	0.61	1.2	1.0	1.0	6.2	0.40
7	397	398	0.61	1.0	1.0	1.0	7.8	0.40
8	483	483	-	-	-	-	9.5	0.40

Table 11.2: Maricopa Stanfield control structures.

Discretization:

The prediction horizon is 12 hours (43200 seconds). The regulation time step is 300 seconds (5 minutes); consequently there are 144 regulation periods. There are eight sluice gates to control eight pools. As every gate position is estimated every regulation period, we need 1152 gate trajectories parameters (144 regulation period by 8 sluice gates). The canal is spatially discretized every 20 meters.

Initial and boundary conditions:

Upstream the first gate, in Maricopa Stanfield canal exists a reservoir so the boundary condition is a constant water level of 1.0 meters, and this value remains constant over the irrigation cycle. The steady state is the initial condition at the beginning of the Test-Cases, the initial condition is shown in Table 11.3 (Test 1-1) and Table 11.4 (Test 1-2) and the downstream water level of every pool is defined by the target value.

Pool number	Offtake initial flow (m ³ /s)	Resulting initial check flow (m ³ /s)
Heading	-	1.0
1	0.1	0.9
2	0.1	0.8
3	0.2	0.6
4	0.2	0.4
5	0.1	0.3
6	0.1	0.2
7	0.1	0.1
8	0.1	0.0

Table 11.3: Initial conditions Test Case 1-1.

Pool number	Offtake initial flow (m ³ /s)	Resulting initial check flow (m ³ /s)
Heading	-	2.0
1	0.2	1.8
2	0.0	1.8
3	0.4	1.4
4	0.0	1.4
5	0.0	1.4
6	0.3	1.1
7	0.2	0.9
8	0.9	0.0

Table 11.4: Initial conditions Test Case 1-2.

11.1.2. Corning Canal

The Corning canal of California has a significant storage capacity. The canal length is 28 Km and the cross sections are trapezoidal. It is divided in eight pools by rectangular gates. There are orifice offtakes, emergency lateral spillways, pump stations and checkpoints or targets points located at downstream end of each pool. The canal geometry is shown in the Figure 11.2 and other features are shown in the Table 11.5 and Table 11.6.

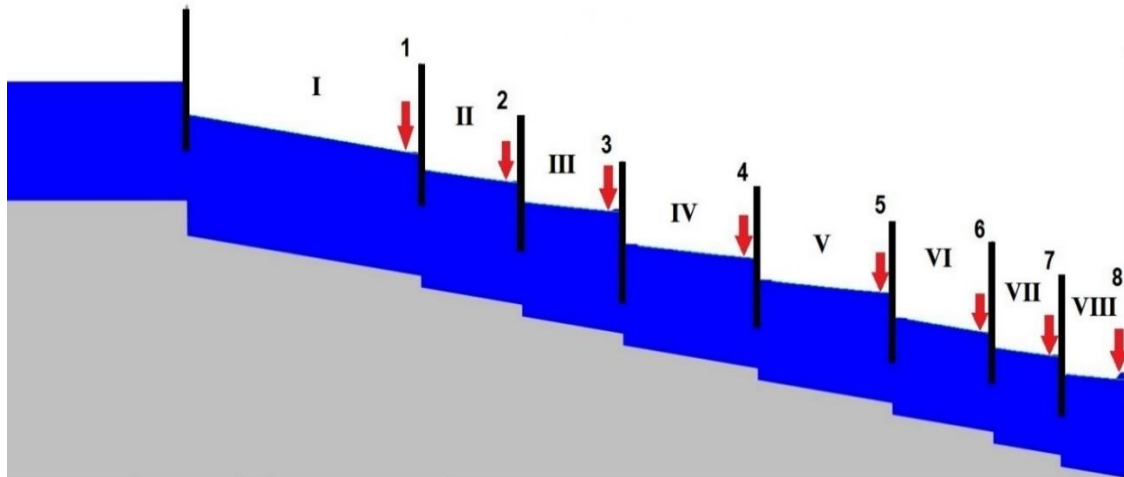


Figure 11.2: Corning Canal profile. The red lines mark the position of checkpoints. The first pool is number I and the first checkpoint is number 1.

Pool number	Numerical node upstream	Numerical node downstream	Pool length (Km)	Bottom slope	Side Slopes (H:V)	Manning's coefficient (n)	Bottom width (m)	Canal Depth (m)
I	1	141	7	10^{-4}	1.5:1	0.02	7	2.5
II	142	202	3	10^{-4}	1.5:1	0.02	7	2.5
III	203	263	3	10^{-4}	1.5:1	0.02	7	2.5
IV	264	344	4	10^{-4}	1.5:1	0.02	6	2.3
V	345	425	4	10^{-4}	1.5:1	0.02	6	2.3
VI	426	486	3	10^{-4}	1.5:1	0.02	5	2.3
VII	487	527	2	10^{-4}	1.5:1	0.02	5	1.9
VIII	528	568	2	10^{-4}	1.5:1	0.02	5	1.9

Table 11.5: Features of Corning canal pools.

Target points	Numerical node upstream	Numerical node downstream	Gate discharge coefficient	Gate width (m)	Gate height (m)	Step (m)	Distance from the gate 1 (Km)	Orifice offtake height (m)	Lateral spillway height (m)
0	0	1	0.61	7	2.3	0.2	0	-	3
1	141	142	0.61	7	2.3	0.2	7	1.05	2.5
2	202	203	0.61	7	2.3	0.2	10	1.05	2.5
3	263	264	0.61	7	2.3	0.2	13	1.05	2.5
4	344	345	0.61	6	2.1	0.2	17	0.95	2.3
5	425	426	0.61	6	2.1	0.2	21	0.95	2.3
6	486	487	0.61	5	1.8	0.2	24	0.85	1.9
7	527	528	0.61	5	1.8	0.2	26	0.85	1.9
8	568	568	-	-	-	-	28	0.85	1.9

Table 11.6: Corning Canal control structures.

Discretization:

The predictive horizon is 12 hours (43200 seconds). Regulation time step is 900 seconds (15 minutes) with 48 regulation periods. There are eight sluice gates to control canal pools; 384 sluice gate trajectories parameters are needed (48 regulation period by 8 sluice gates). The canal is spatially discretized every 50 meters.

Initial and boundary conditions:

Upstream the first gate in Corning canal, there is a reservoir, so the boundary condition is a constant water level of 3.0 meters and this value remains constant over the predictive horizon. The steady state is the initial condition at the beginning of the Test-Cases; the initial conditions are shown in Table 11.7 (Test 2-1) and Table 11.8 (Test 2-2).

Pool number	Offtake initial flows (m ³ /s)	Resulting initial check flows (m ³ /s)
Heading	-	13.5
1	1.0	12.5
2	1.0	11.5
3	1.0	10.5
4	1.0	9.5
5	2.5	7.0
6	2.0	5.0
7	1.0	4.0
8	1.0	3.0

Table 11.7: Initial conditions Test Case 2-1.

Pool number	Offtake initial flows (m ³ /s)	Resulting initial check flows (m ³ /s)
Heading	-	13.7
1	1.7	12.0
2	1.8	10.2
3	2.7	7.5
4	0.3	7.2
5	0.2	7.0
6	0.8	6.2
7	1.2	5.0
8	0.3 +2.0*	2.7

Table 11.8: Initial conditions Test Case 2-2.

*Changes in downstream pump discharge with no change in offtake flow.

The offtake flows are obtained by gravity offtakes located downstream end of each pool where the extracted flow through the offtake (Q_{offtake}) is described as follow:

$$Q_{\text{offtake}} = K\sqrt{y - y_0} \quad (11.1)$$

$$K = C_d \times \frac{\pi \times \phi^2}{4} \times \sqrt{2g}$$

where y is the water level in canal at offtake, y_0 is the orifice offtake height equal to $y_{\text{target}}/2$, C_d the discharge coefficient calculated considering the extracted flow through the offtake for the target value so it gets a variable value during the irrigation cycle and ϕ is the orifice offtake diameter.

11.2. Target values

The target values which we have to maintain at the orifice offtake during the irrigation cycle are fixed in the whole test. We introduce these values of every test case in this section.

11.2.1 Maricopa Stanfield

As we introduced before, the control algorithm has to maintain a desired water level at the checkpoints during a predictive horizon. The desired water levels are shown in Table 11.9.

Target point	Water level desired (m)
1	0.9
2	0.9
3	0.8
4	0.9
5	0.9
6	0.8
7	0.8
8	0.8

Table 11.9: Desired water level Test Case 1-1/1-2.

11.2.2 Corning Canal

The desired water levels at checkpoints are shown in the next table.

Target point	Water level desired (m)
1	2.1
2	2.1
3	2.1
4	1.9
5	1.9
6	1.7
7	1.7
8	1.7

Table 11.10: Desired water level Test Case 2-1/2-2.

11.3. Unscheduled flow changes

The duration of unscheduled deliveries is ten hours, almost all test duration. To control the canal is not easy because these disturbances cause hydraulic and numerical problems in the control algorithms.

CSI algorithm calculates the unscheduled water deliveries and it sends the output data to the control algorithm which responds modifying the gate trajectories. However nobody knows the total duration of the unscheduled water deliveries. For this reason, GoRoSoBo algorithm calculates the sluice gate trajectories for a predictive horizon, taking into account the unscheduled water deliveries obtained by CSI algorithm in a present time and GoRoSoBo extrapolates the unscheduled water deliveries in all prediction horizon to calculate the sluice gate trajectories, as we introduced in chapter 9. In the next regulation period, new water level measurements are taken at the checkpoints and CSI algorithm comes back to calculate the new unscheduled water deliveries and GoRoSoBo algorithm comes back to calculate the new gate trajectories too. In the case that the unscheduled water deliveries only exist at the present time, not during the next regulation periods, the GoRoSoBo will modify the gate trajectories considering a wrong water level error in the prediction horizon. The errors caused by that hypothesis (extrapolate the unscheduled water deliveries in the whole prediction horizon) are small, although these errors depend on the duration of the regulation period and the value of the unscheduled water deliveries.

We had to consider this hypothesis because the sluice gate movements at present time are essentials to reduce the disturbance in future. The disturbance wave generated by the sluice gate movement does not travel quickly in free surface, so the control actions must be taken at the present time to obtain a response in future. If we consider that the unknown delivery does not exist in future, and the unscheduled offtake changes are permanent, the control algorithm can never obtain a gate trajectory to reduce the deviation between the desired and measured water level.

This happens because once the disturbance disappears, the water level at checkpoints returns to the desired water level by itself quickly, if we make this supposition, the control algorithm does not have to make actions for the future, but if this supposition is wrong and the disturbance has a long duration, the water level error will not be removed. For this

reason, as we introduced in chapter 9, the unscheduled offtake obtained by CSI algorithm at the present time is extrapolated during the predictive horizon.

We have to note that there are gates which have not any influences on some target points at the end of the prediction horizon, for instance, a gate movement did by the first gate at last regulation period has not any influences on target 8 at last regulation period because the wave generated by the first sluice gate at last regulation period cannot reach to the target 8. There are some elements in the HIM matrix with zero values, and these zeroes can produce numerical problems, even more on important unscheduled delivery, as there are long periods during the prediction horizon that the sluice gate positions have changed a lot from the initial gate position, instead at the end of the prediction horizon as the sluice gates have not any influences on the checkpoints, the sluice gate remains in their initial position. These changes in gate trajectory so significant generate numerical problems. To solve these numerical problems, we increased the duration of the last regulation period, four times, as in this way all sluice gates have influences on any checkpoint during the last regulation period.

11.3.1 Maricopa Stanfield

The unscheduled water deliveries are more important in test 1-2 than in test 1-1, as it is shown in Table 11.11 and Table 11.12. These unknown deliveries are introduced from the second hour of the test until the end of the test.

Pool number	Offtake initial flow (m ³ /s)	Check initial flow (m ³ /s)	Unscheduled offtake changes at 2 hours (m ³ /s)	Check final flow (m ³ /s)
Heading	-	1.0	-	0.8
1	0.1	0.9	-	0.7
2	0.1	0.8	-	0.6
3	0.2	0.6	-	0.4
4	0.2	0.4	-0.1	0.3
5	0.1	0.3	-0.1	0.3
6	0.1	0.2	-	0.2
7	0.1	0.1	-	0.1
8	0.1	0.0	-	0.0

Table 11.11: Unscheduled offtake changes on Test Case 1-1.

Pool number	Offtake initial flow (m ³ /s)	Check initial flow (m ³ /s)	Unscheduled offtake changes at 2 hours (m ³ /s)	Check final flow (m ³ /s)
Heading	-	2.0	-	2.0
1	0.2	1.8	-	1.8
2	0.0	1.8	0.2	1.6
3	0.4	1.4	-0.2	1.4
4	0.0	1.4	0.2	1.2
5	0.0	1.4	0.2	1.0
6	0.3	1.1	-0.1	0.8
7	0.2	0.9	-	0.6
8	0.9	0.0	-0.3	0.0

Table 11.12: Unscheduled offtake changes on Test Case 1-2.

The unscheduled water deliveries in test 1-2 are important although the changes on the total flow rate is null, because there are pools where the unscheduled deliveries are positive and others are negative.

The unscheduled deliveries are more significant at target 8 in test-case 1-2, where the flow rate changes from 0.9 m³/s to 0.6 m³/s (33%). For this reason, the control algorithm has to modify quickly the gate trajectories, because it exists a possibility to canal overflow in some pools downstream due to the low storage capacity. This is more usual in canals like Maricopa Stanfield than the Corning canal which has an important storage capacity.

11.3.2 Corning Canal

In test 2-1, we find that the unscheduled water deliveries are not very significant, as it is shown in the Table 11.13 and Table 11.14. The only flow change is in the orifice offtake of pool six. It is a flow change on the total flow in the canal of 14.8 % and on the total flow in pool six of 28.5%. These unknown deliveries are introduced from the second hour of the test until the test ending. These unscheduled deliveries are flow rate changes and the control algorithm realizes of these flow changes once the water levels increase or decrease at target points or checkpoints.

Pool number	Offtake initial flow (m ³ /s)	Check initial flow (m ³ /s)	Unscheduled offtake changes at 2 hours (m ³ /s)	Resulting check flow (m ³ /s)
Heading	-	13.5	-	11.5
1	1.0	12.5	-	10.5
2	1.0	11.5	-	9.5
3	1.0	10.5	-	8.5
4	1.0	9.5	-	7.5
5	2.5	7.0	-	5.0
6	2.0	5.0	-2	5.0
7	1.0	4.0	-	4.0
8	1.0	3.0	-	3.0

Table 11.13: Unscheduled offtake changes on Test Case 2-1.

Pool number	Offtake initial flow (m ³ /s)	Check initial flow (m ³ /s)	Unscheduled offtake changes at 2 hours (m ³ /s)	Resulting check flow (m ³ /s)
Heading	-	13.7	-	2.7
1	1.7	12.0	-1.5	2.5
2	1.8	10.2	-1.5	2.2
3	2.7	7.5	-2.5	2.0
4	0.3	7.2	-	1.7
5	0.2	7.0	-	1.5
6	0.8	6.2	-0.5	1.2
7	1.2	5.0	-1.0	1.0
8	0.3 +2.0*	2.7	-2.0*	0.7

Table 11.14: Unscheduled offtake changes on Test Case 2-1.

*Changes in downstream pump discharge with no change in offtake flow.

The unscheduled water deliveries in test 2-2 are very relevant, as these water deliveries have an important percentage over the total flow. In this case, the unscheduled water deliveries are of 81% on the total flow and these water changes are more significant at target 8, where the flow rate turns from 5 m³/s to 1 m³/s. For this reason, it should be pointed out that 15 minutes after introducing the unscheduled water deliveries in the canal, the water level in target 8 is 1.93 meters, so the lateral spillway must operate. This is the maximum flow rate change in all Test-Cases.

11.4. Constrains and trajectories

According with Soler (2003), there are different ways to reach the final solution depending on the maximum or minimum gate position, the maximum sluice gate movement permitted between consecutives regulation periods or the time-frequency that actuator can move the gates. The larger the maximum sluice gate movement permitted, the faster is to drive the canal state to the desired canal state in case of significant flow changes, but on the other hand it is easier to damage the servomechanism of the sluice gate, because the gate trajectories oscillate more. In such case, an optimization problem which the constraints are too lax can destabilize the model, because the change of the free surface due to the gate trajectories can be very abrupt.

ASCE restricts the minimum regulation period in 15 minutes for the test-case 2-1/2-2. This was the regulation period chosen in test 2-1 and 2-2, and the time-frequency on gate movement is also 15 min. The ASCE restricts the minimum regulation period in 5 minutes for the test 1-1 and 1-2, so this was the regulation period chosen in test 1-1 and 1-2 and the time-frequency on sluice gate movement.

Considering the results obtained in the analysis of constrains did by Soler (2003) and Soler et al. (2013), we have imposed the overall and functional constraints of Table 11.15. The constraints are determined as a certain percentage of the gate height, these constraints are imposed to the gate positions.

	U_{\min} (%)	U_{\max} (%)	$dU_{\max}/dU_{0\max}$ (%)
Maricopa Stanfield	2	90	5
Corning Canal	0.5	90	5

Table 11.15: Overall and functional constraints values.

The value of minimum gate position is different in both cases. The gate width of the Corning canal is almost five times bigger than the Maricopa Stanfield so the minimum gate position is fixed by a combination of the gate width and the flow through the gate. For instance, we cannot fix a minimum gate position of 2 % for the Corning canal, because this gate position is equivalent to a significant flow through the gate, in situations that the total flow in the canal decreases by important reductions in water demand, the gates position should be lower than 2 %.

11.5. Test results

We show the results in three graphs for every test case:

- The first one shows the scheduled offtakes and unscheduled offtake changes (demanded extracted flow), the extracted flow calculated by CSI and the real value delivered by the offtake after introducing the new gate trajectories by GoRoSoBo during the irrigation cycle. As we introduced before, GoRoSoBo needs the CSI information to calculate the gates trajectories so CSI has to calculate the extracted flow in the canal considering the scheduled and unscheduled offtake flow.
- The second one* shows the gate trajectories during the irrigation cycle, where the gate trajectory is defined by the gate position in time, and the gate position is defined as follow:

$$\text{Gate position} = \frac{\text{Gate opening}}{\text{Gate height}} \quad (11.2)$$

- The third one* shows the water level at the checkpoints which is proportional to the real flow-rate delivered to the farmer by the offtake.

11.5.1 Test-Case 1-1 (Maricopa Stanfield)

In the next table, we show the scheduled offtakes and the unscheduled water changes which are introduced 2 hours after starting the test:

Pool number	Scheduled offtake flow (m ³ /s)	Unscheduled offtake changes at 2 hours (m ³ /s)
Heading	-	-
1	0.1	-
2	0.1	-
3	0.2	-
4	0.2	-0.1
5	0.1	-0.1
6	0.1	-
7	0.1	-
8	0.1	-

Table 11.16: Unscheduled offtake changes in test-case 1-1 (Maricopa Stanfield).

*We have to note that the gate trajectories and water level trajectories has been divided in two graphs for a better comprehension of the results.

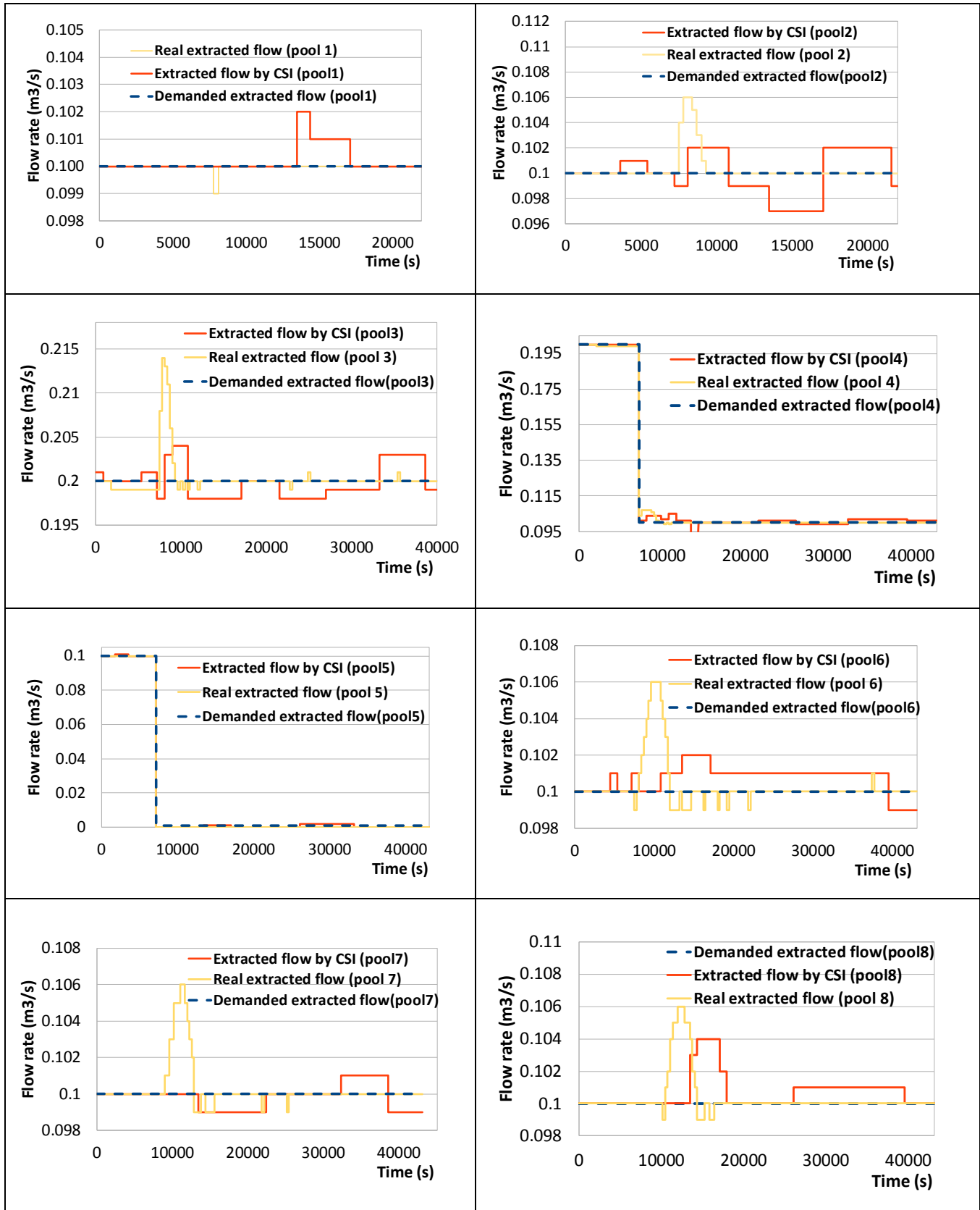


Figure 11.3: Demanded extracted flow, extracted flow by CSI and real extracted flow in Test-Case 1-1 (Maricopa Stanfield).

The only flow changes are introduced in the fourth and fifth pool (Table 11.16), and CSI is able to establish perfectly the extracted flow in the canal with a maximum numerical error of 4% between the demanded extracted flow and the value by CSI, see Figure 11.3. This small error is due to numerical errors in CSI and we obtain a similar maximum error in every pump station.

Once the GoRoSoBo algorithm realizes that the water level increases at checkpoints 4 and 5 at 7500 s (Figure 11.4 and Figure 11.5.), the sluice gates 1 to 5 start to close to reduce the water level at downstream checkpoints (Figure 11.6 and Figure 11.7) and the sluice gates 6, 7 and 8 start to open to reduce the water level at upstream checkpoints. After several regulation periods, the sluice gates 6, 7 and 8 must close because the water level at checkpoints 6, 7 and 8 also increase, see

GoRoSoBo starts to modify the gate trajectories at 7500 s, one regulation period after the disturbance is introduced in the canal. GoRoSoBo drives the canal to the desired water level in every checkpoint at 18000 s.

At the end, the controller response is smoother and gate movement is damped to prevent oscillations that would cause excessive control actions, because the water levels are almost equal to the desired levels, with a maximum error between them around of 0.26%.

The results obtained in this test case are quite good as the performance indicators demonstrate, see section 11.6. Four hours later introducing the unscheduled offtake changes, GoRoSoBo drives the canal to the target value.

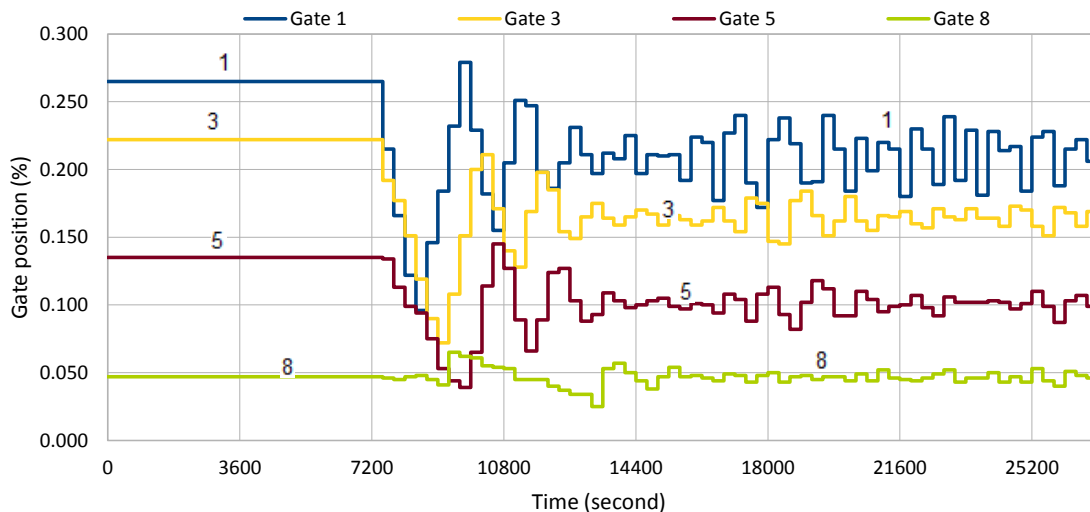


Figure 11.4: Gate trajectories (1, 3, 5 and 8) in Test-Case 1-1 (Maricopa Stanfield).

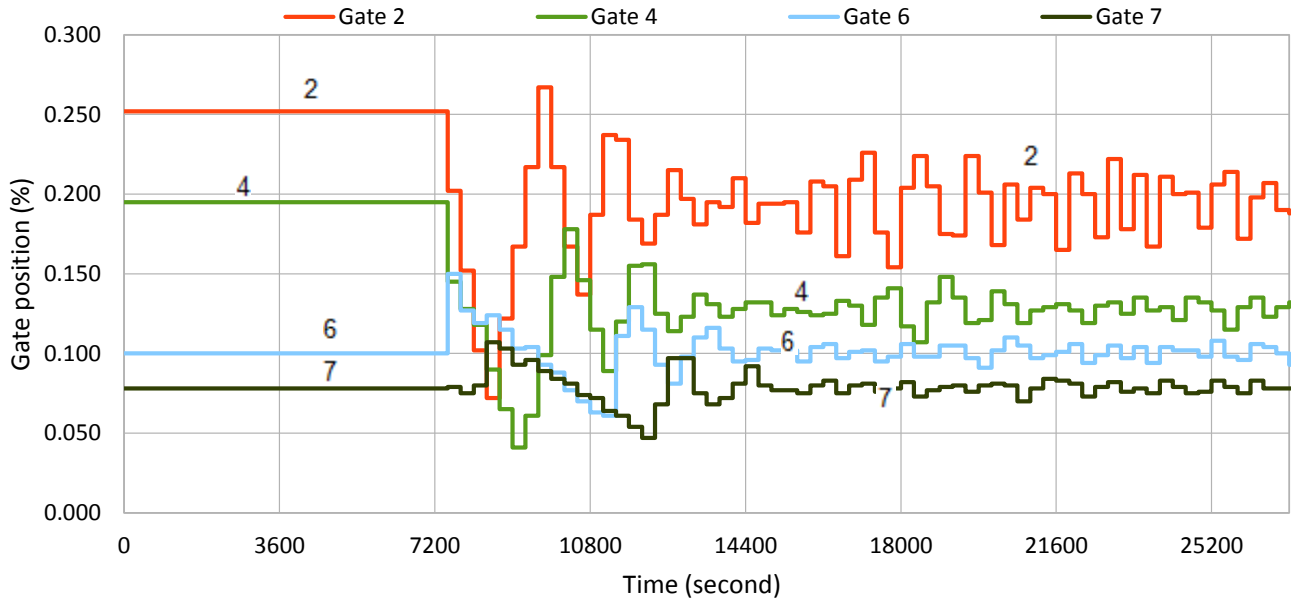


Figure 11.5: Gate trajectories (2, 4, 6 and 7) in Test-Case 1-1 (Maricopa Stanfield).

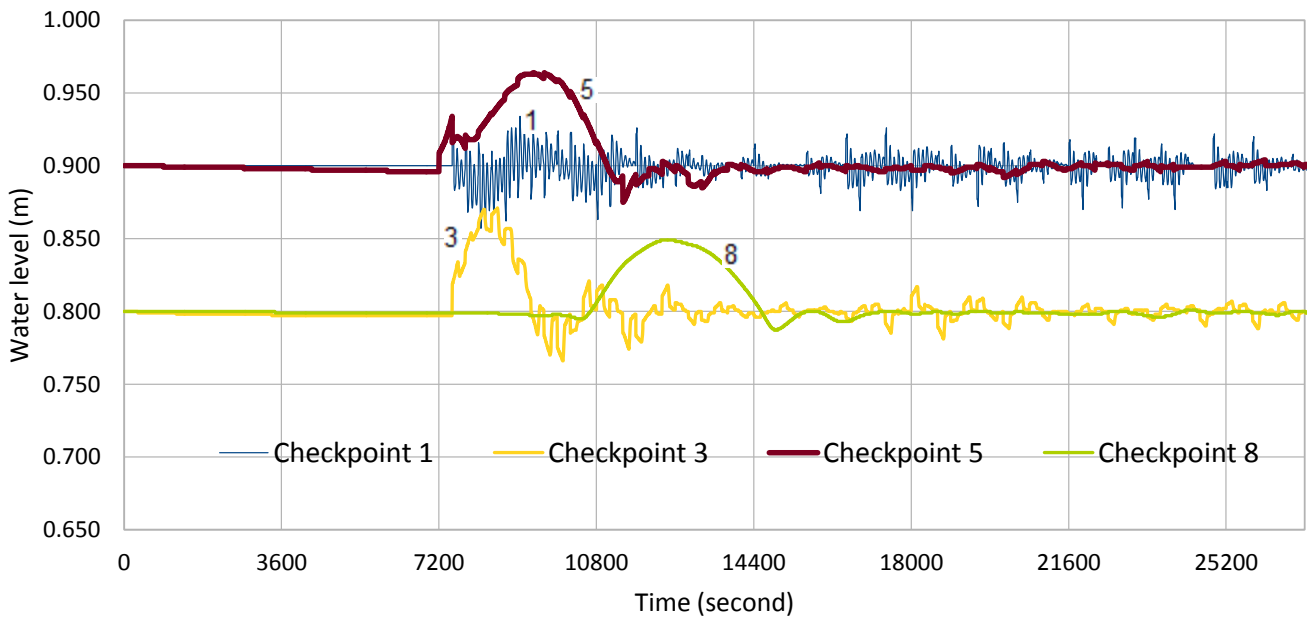


Figure 11.6: Water level at checkpoints 1, 3, 5 and 8 in Test-Case 1-1 (Maricopa Stanfield).

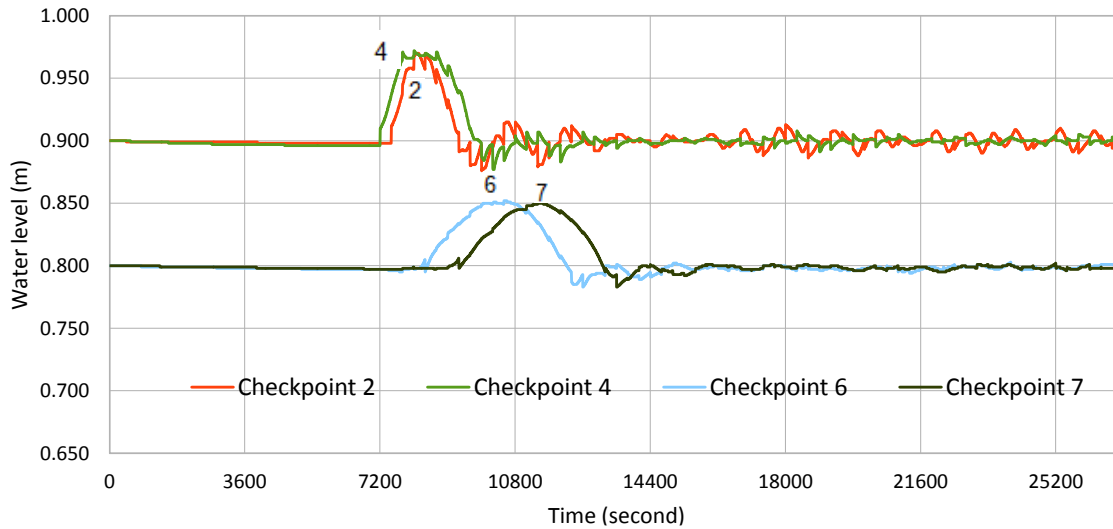


Figure 11.7: Water level at checkpoints 2, 4, 6 and 7 in Test-Case 1-1 (Maricopa Stanfield).

11.5.2 Test Case 1-2 (Maricopa Stanfield)

The scheduled offtakes and the unscheduled water changes which are introduced 2 hours after starting the test are shown in the next table:

Pool number	Scheduled offtake changes (m ³ /s)	Unscheduled offtake changes at 2 hours (m ³ /s)
Heading	-	-
1	0.2	-
2	0.0	0.2
3	0.4	-0.2
4	0.0	0.2
5	0.0	0.2
6	0.3	-0.1
7	0.2	-
8	0.9	-0.3

Table 11.17: Unscheduled offtake changes in test-case 1-2 (Maricopa Stanfield).

This test 1-2 is much more difficult than the test 1-1, because the unscheduled water deliveries are higher in all targets. The canal is in steady state during the first two hours. After the firsts two hours, unscheduled water deliveries are introduced to the system at 7200 s, although the algorithm has not any notice until the next regulated period, once the water

level is measured at the checkpoints. It is important to remark that the unscheduled water deliveries are relevant in all targets but especially at target 8, because just in one regulation period, the water level at checkpoint 8 increases from 0.8 m to 1.08 m (Figure 11.11).

It is introduced a water delivery change of $0.3 \text{ m}^3/\text{s}$ at checkpoint 8, with a total flow rate change at target 8 from $0.6 \text{ m}^3/\text{s}$ to $0.9 \text{ m}^3/\text{s}$, that is, a flow rate change around of 33.3 %. The extracted flow is calculated by CSI which establishes the value a maximum numerical error of 2% between the demanded extracted flow and the value by CSI, as we show in Figure 11.8.

The unscheduled offtake changes in target 1 to 5 are $0.2 \text{ m}^3/\text{s}$, in target 6 is $0.1 \text{ m}^3/\text{s}$ and in target 8 is $0.3 \text{ m}^3/\text{s}$, in a canal with a total flow rate of $2 \text{ m}^3/\text{s}$ that has not storage capacity and a steep slope, this is a hard test for the control algorithm.

Once the GoRoSoBo algorithm knows that the water level is increasing less at checkpoints 2, 4 and 5 at time 7500 s (Figure 11.11 and Figure 11.12), all gates less 7 and 8 have the tendency of closing (Figure 11.9 and Figure 11.10). Although the water level is increasing in several checkpoints, the maximum water level variation is at checkpoint 8 at 8100 s, for that reason, the gates 7 and 8 close whereas the gates 1, 2, 3, 4, 5, 6 open for readjusting the water level in all checkpoints from 8100 s to 9900 s. We can check that the lower gate position at 18000 s is the gate 8 that is logic because the more important flow change is at checkpoint 8.

The sluice gate trajectories have the same shape that the water level at the checkpoints, due to the gate trajectories and water levels fluctuate around the desired solution.

The water level decreases in all targets at 9900 s due to changes in gate trajectories, and the water level returns to the desired water level in all checkpoints at 21600 s.

The water levels were almost equal to the desired levels at the end of the test, with a maximum error between them around 0.25%. The results obtained in this Test Case 1-2 are so good as the results obtained in Test-Case 1-1. GoRoSoBo needs one hour more to drive the system to the target value than in Test Case 1-1, but the unscheduled offtake changes are more important in this Test Case 1-2.

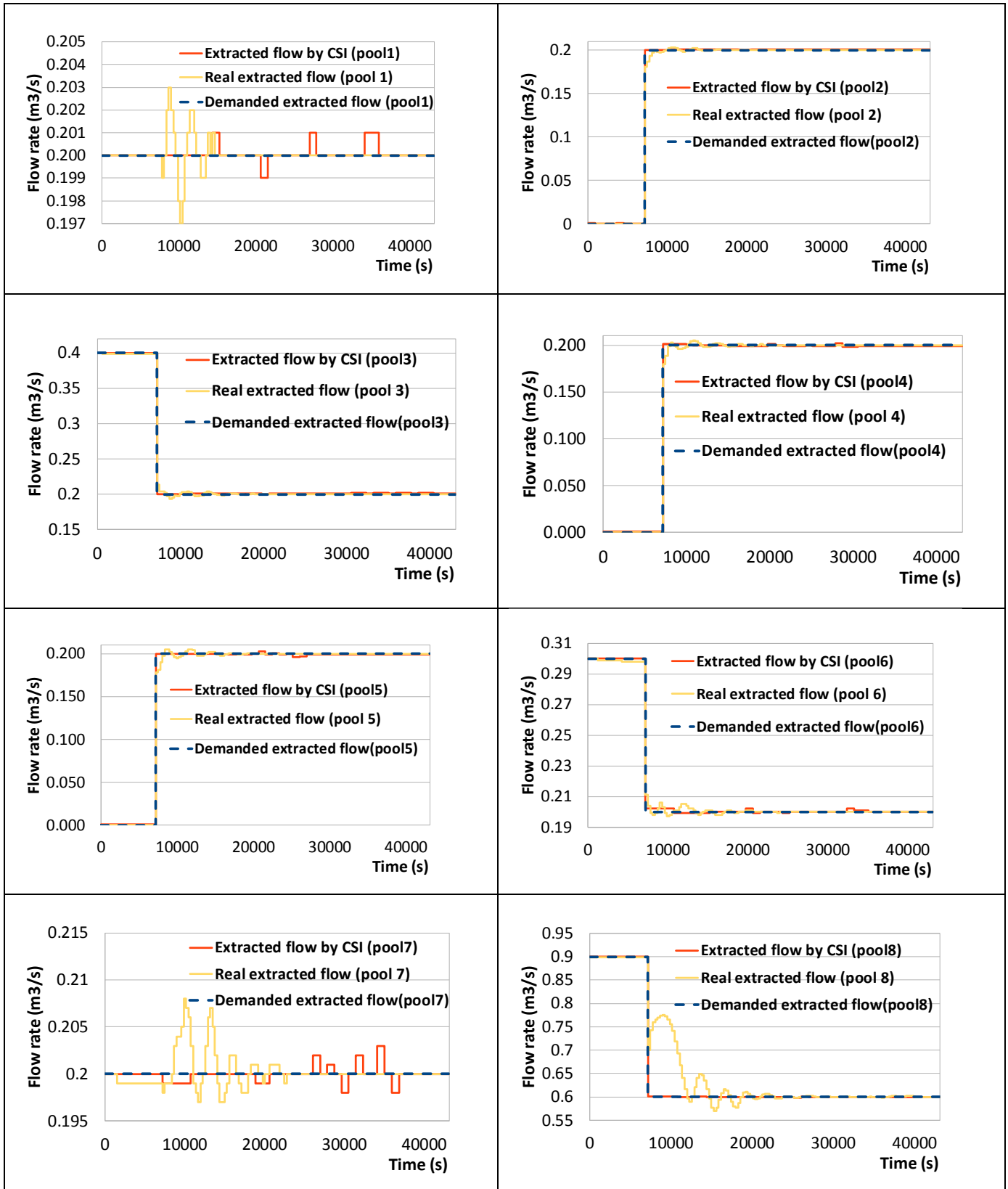


Figure 11.8: Demanded extracted flow, extracted flow by CSI and real extracted flow in Test-Case 1-2 (Maricopa Stanfield).

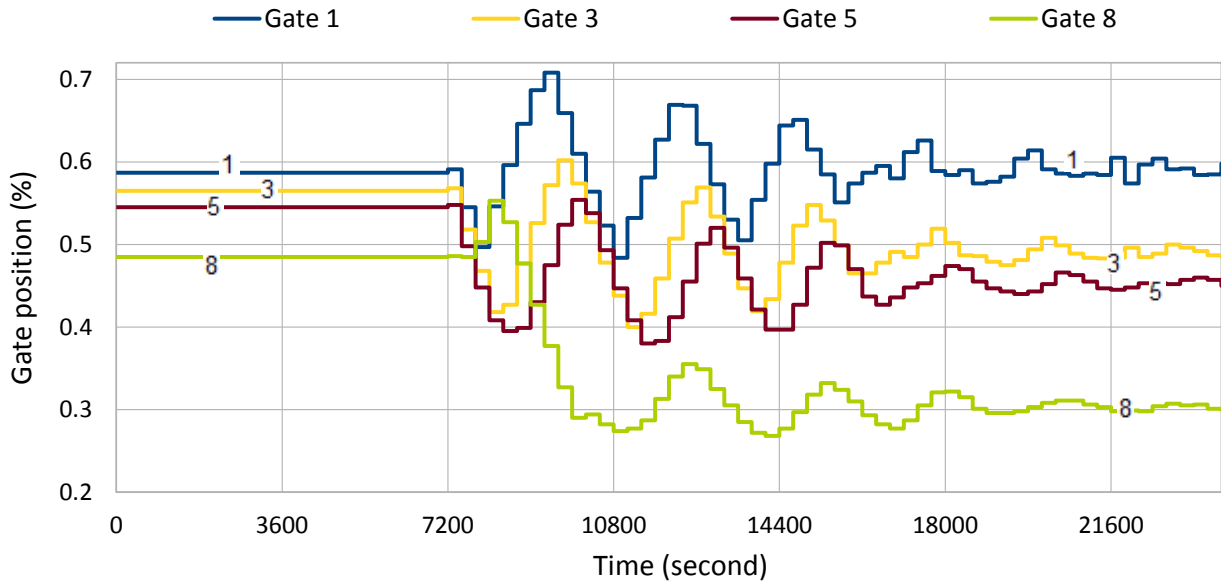


Figure 11.9: Gate trajectories (1, 3, 5 and 8) in Test-Case 1-2 (Maricopa Stanfield).

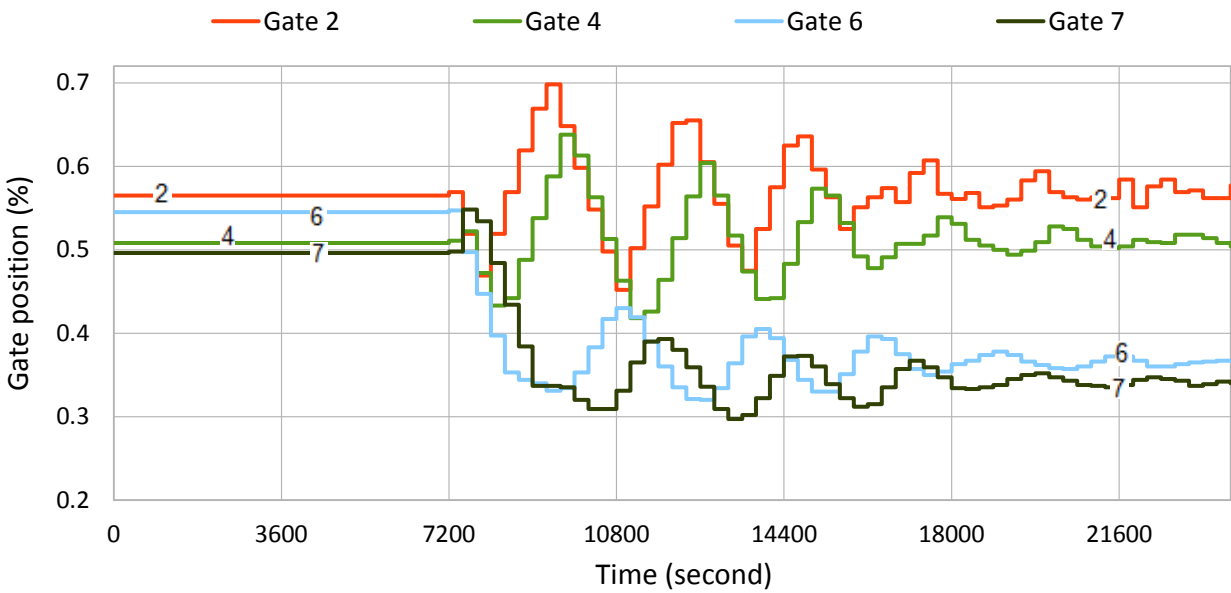


Figure 11.10: Gate trajectories (2, 4, 6 and 7) in Test-Case 1-2 (Maricopa Stanfield).

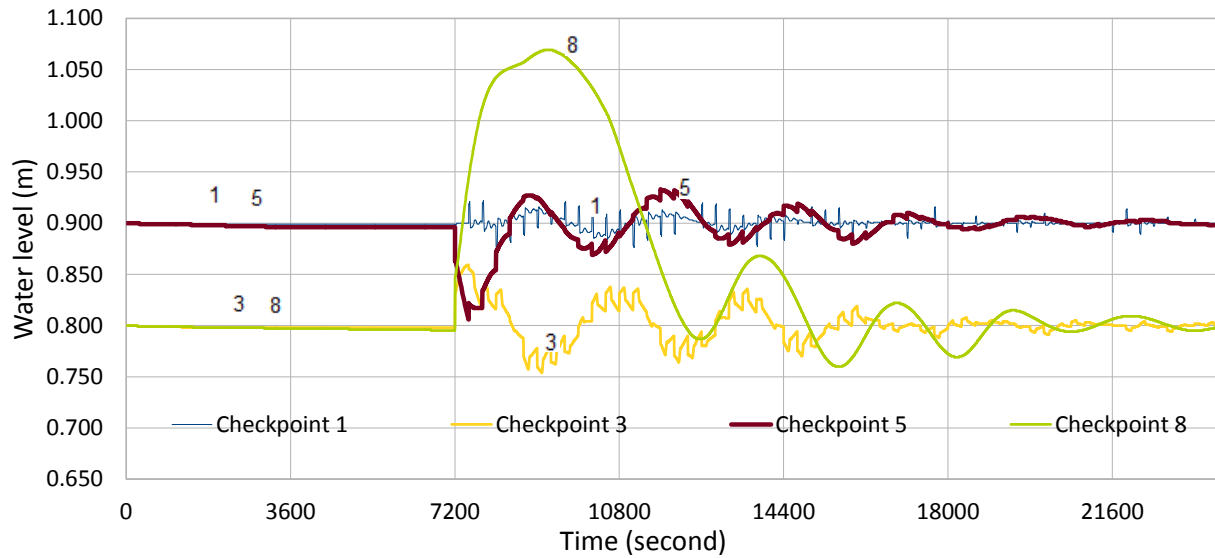


Figure 11.11: Water level at checkpoints 1, 3, 5 and 8 in Test-Case 1-2 (Maricopa Stanfield).

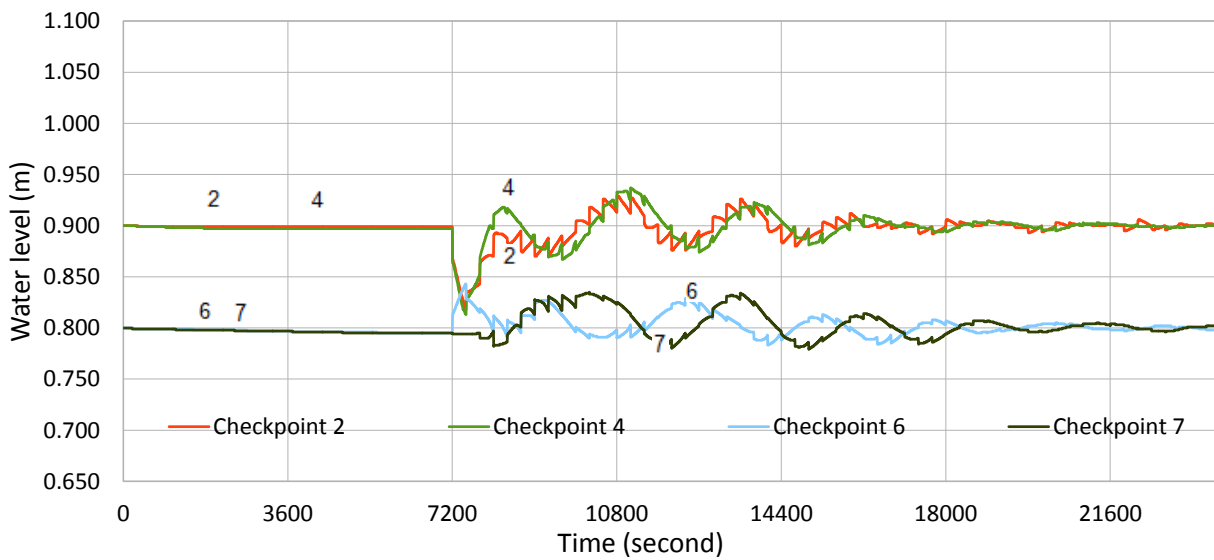


Figure 11.12: Water level at checkpoints 2, 4, 6 and 7 in Test-Case 1-2 (Maricopa Stanfield).

11.5.3 Test-Case 2-1 (Corning canal)

The scheduled offtakes and the unscheduled water changes which are introduced 2 hours after starting the test are shown in the next table:

Pool number	Scheduled offtake changes (m ³ /s)	Unscheduled offtake changes at 2 hours (m ³ /s)
Heading	-	-
1	1.0	-
2	1.0	-
3	1.0	-
4	1.0	-
5	2.5	-
6	2.0	-2
7	1.0	-
8	1.0	-

Table 11.18: Unscheduled offtake changes in test-case 2-1 (Corning canal).

The steady state is the initial condition. The sluice gates remain in their positions during the first two hours. The unscheduled water deliveries are introduced into the system at time 7200 s, although the algorithm has not any notice until the next regulation period when the water level at checkpoint 6 is already increased close to 10 cm. The measurements are sent to CSI which establishes the extracted flow with a maximum numerical error of 0.7% between the demanded extracted flow and the value by CSI as we show in Figure 11.13.

The change in water delivery is quite important at target 6, as it is introduced a water delivery change of 2 m³/s, and the total flow rate at target 6 is 7 m³/s (that is, a flow rate change of 28 % of the total flow rate).

Once GoRoSoBo algorithm realizes that the water level has increased at checkpoint 6 at 8100 s (Figure 11.16), the gates 1 to 6 start to close (Figure 11.14 and Figure 11.15), and the gates 7 and 8 start to open from 8100 s to 9900 s to reduce the increased water level at checkpoint 6 (Figure 11.14), so the water level at checkpoint 7 and 8 start to increase at 9000 s (Figure 11.17) and the gate 7 and 8 start to close at 9900 s. The water level start to decrease in all checkpoints at 12600 s, and the water level returns to the desired level at 19800 s, see Figure 11.16 and Figure 11.17. The gate movements were almost zero in the last six hours of the test because the water levels at checkpoints were equal to the desired water levels, with a maximum error of 0.2%. These little errors are due to residual errors in the computation of both algorithms (CSI and GoRoSoBo).

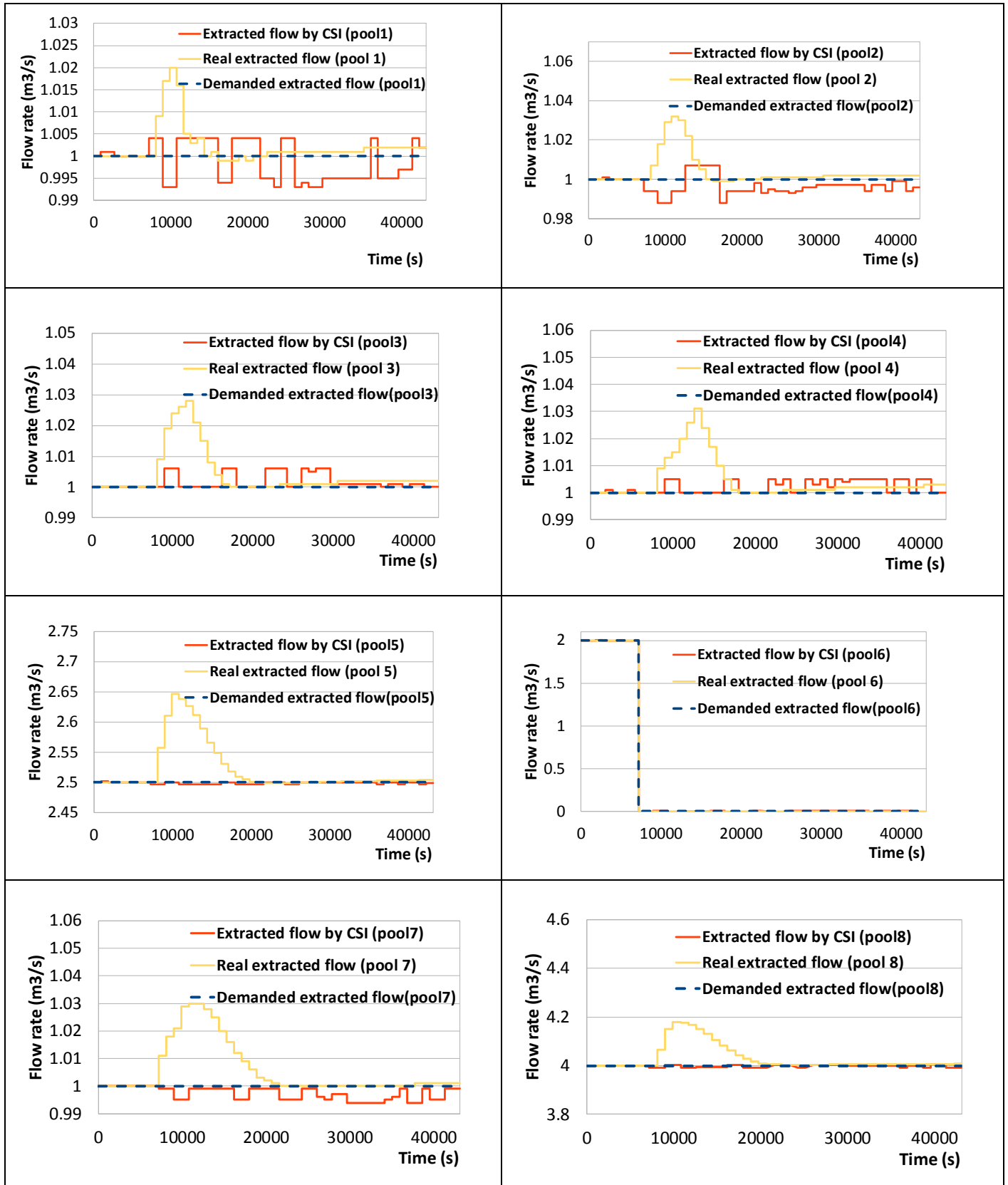


Figure 11.13: Demanded extracted flow, extracted flow by CSI and the real extracted flow in Test-Case 2-1 (Corning canal).

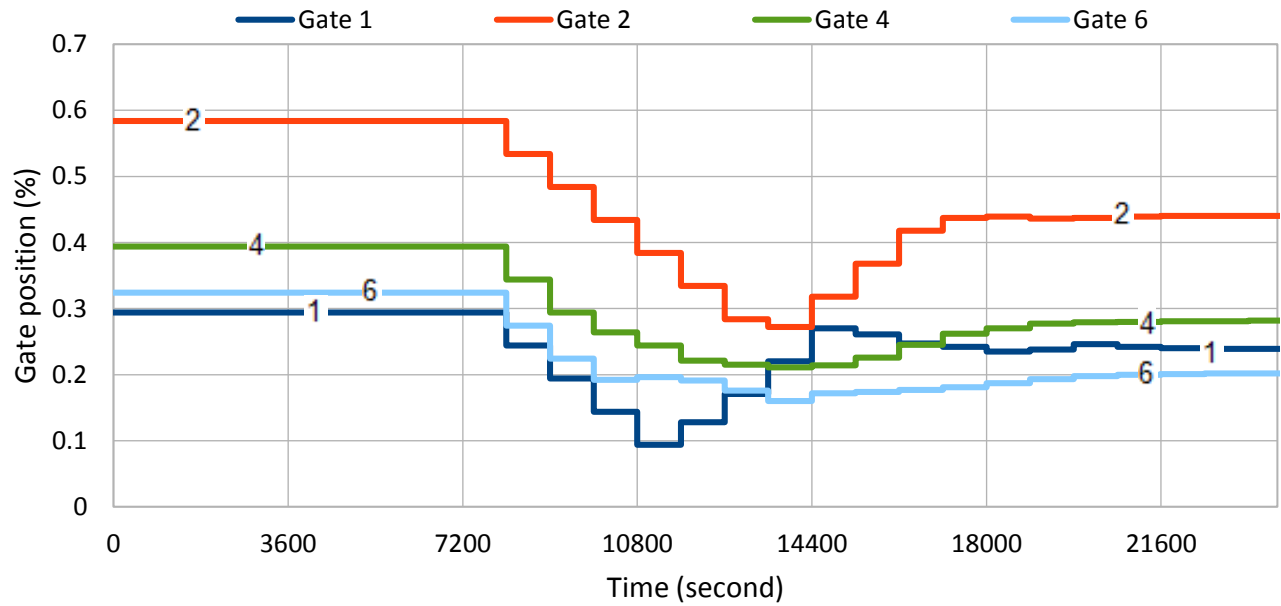


Figure 11.14: Gate trajectories (1, 2, 4 and 6) in Test-Case 2-1 (Corning canal).

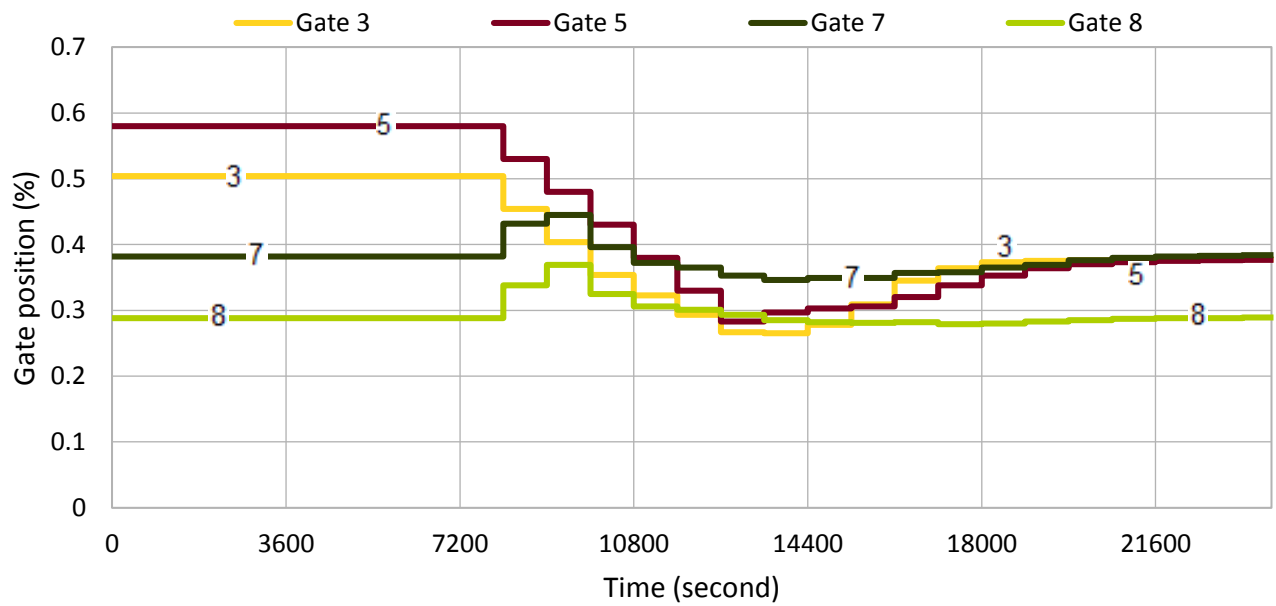


Figure 11.15: Gate trajectories (3, 5, 7 and 8) in Test-Case 2-1 (Corning canal).

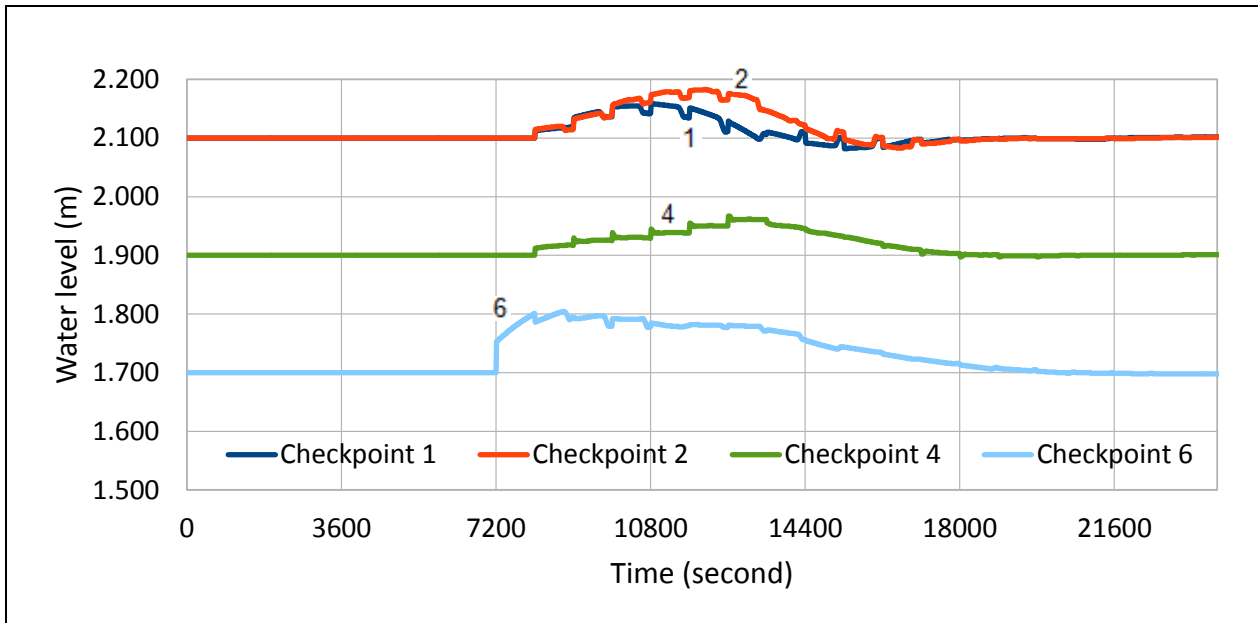


Figure 11.16: Water level at checkpoints 1, 2, 4 and 6 in Test-Case 2-1 (Corning canal).

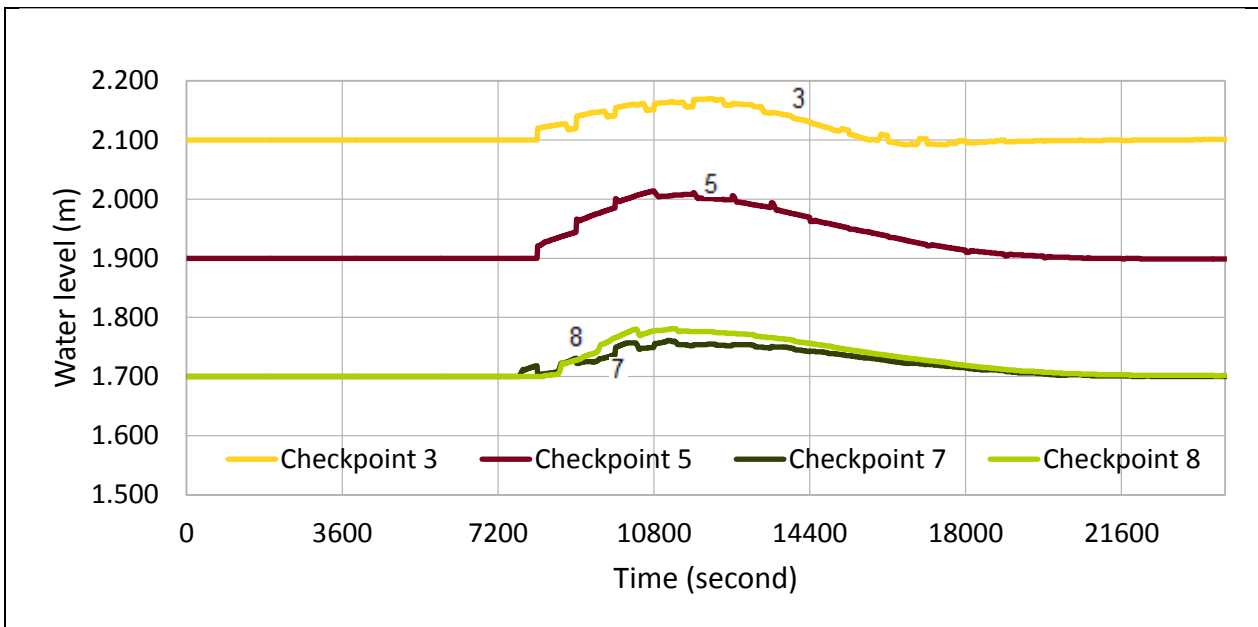


Figure 11.17: Water level at checkpoints 3, 5, 7 and 8 in Test-Case 2-1 (Corning canal).

We can check easily in this test-case 2-1 the good behavior of CIS and GoRoSoBo because there is only a flow change at checkpoint 6 and the algorithm modifies all gates according to this only flow change.

11.5.4 Test Case 2-2 (Corning canal)

In the next table, we show the scheduled offtakes and the unscheduled water changes which are introduced 2 hours after starting the test:

Pool number	Scheduled offtake changes (m ³ /s)	Unscheduled offtake changes at 2 hours (m ³ /s)
Heading	-	-
1	1.7	-1.5
2	1.8	-1.5
3	2.7	-2.5
4	0.3	-
5	0.2	-
6	0.8	-0.5
7	1.2	-1.0
8	0.3 +2.0*	-2.0*

Table 11.19: Unscheduled offtake changes in test-case 2-1 (Corning canal).

*Changes in downstream pump discharge with no change in offtake flow. The outlet flow at checkpoint 8 at initial time step is 2.7 m³/s and the outlet flow at checkpoint 8 at 7200 s is 0.7 m³/s.

The test 2-2 is the most difficult test due to the significant unscheduled water deliveries in all targets. The canal state is steady during the first two hours, as in the previous test-cases. There are no unscheduled water deliveries for the first two hours, so the sluice gates remain in their initial positions. After the first two hours, unscheduled water deliveries are introduced to the system. The water level measurements are sent to CSI which establishes the real extracted flow with a maximum error of 3% between the demanded extracted flow and the value obtained by CSI, as we show in Figure 11.18. The unscheduled water deliveries are so important in all targets that the total unscheduled deliveries were 9 m³/s in a canal with a total flow of 13.7 m³/s. GoRoSoBo obtained good results in Test Case 2-2, as we show at Figure 11.21 and Figure 11.22, although it is important to remark that the unscheduled water deliveries are so important, especially in target 8, that once the algorithm has notice of the rising water level at checkpoint 8 at 8100 s, the water level has already increased 20 cm, that is, it reaches to the top of the cross-section. In that circumstances, GoRoSoBo closes the sluice gates 1 to 3 and open the sluice gates 4 to 8 (Figure 11.19 and Figure 11.20), so in this way the sluice gate trajectories reduce the increased water lever in all checkpoints.

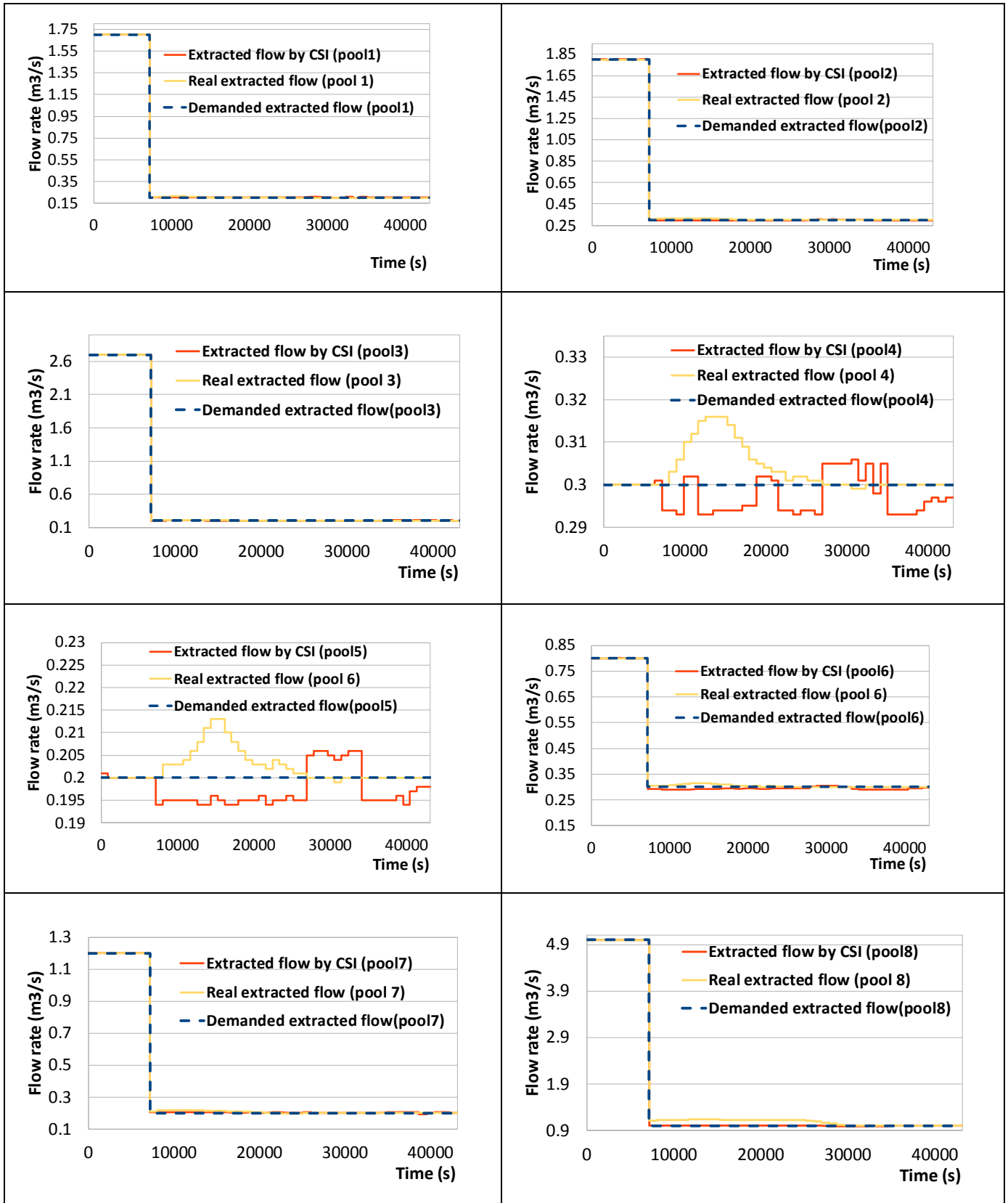


Figure 11.18: Demanded extracted flow, extracted flow by CSI and the real extracted flow in Test-Case 2-2 (Corning canal).

Redirecting the water level at checkpoints to the desired water level in a centralized system is not a duty of only one sluice gate. It is possible that some sluice gates have increased the water level error in some checkpoints at a regulation period (for instance, at checkpoint 4 and 5) when there are not unscheduled offtake changes in these checkpoints, but the main objective is redirect the water level measured to the desired values in all checkpoints as soon as possible.

The water level decreases in all checkpoints at 25200 s and the water level returns to the desired value in all the checkpoints at 30600 s. The unscheduled water delivery is quite important at target 8 introducing a total flow change of $4 \text{ m}^3/\text{s}$ at 7200 s because the total flow rate at pool 8 is $5 \text{ m}^3/\text{s}$ at initial time step and the flow rate at pool 8 is $1 \text{ m}^3/\text{s}$ at 7200 s, so the flow change is close to 80 % of the total flow rate of the pool.

The gate movements are almost zero in the last two hours of the test because the water levels at checkpoints are equal to the desired values at the end of the test, with a maximum error around of 0.23%.

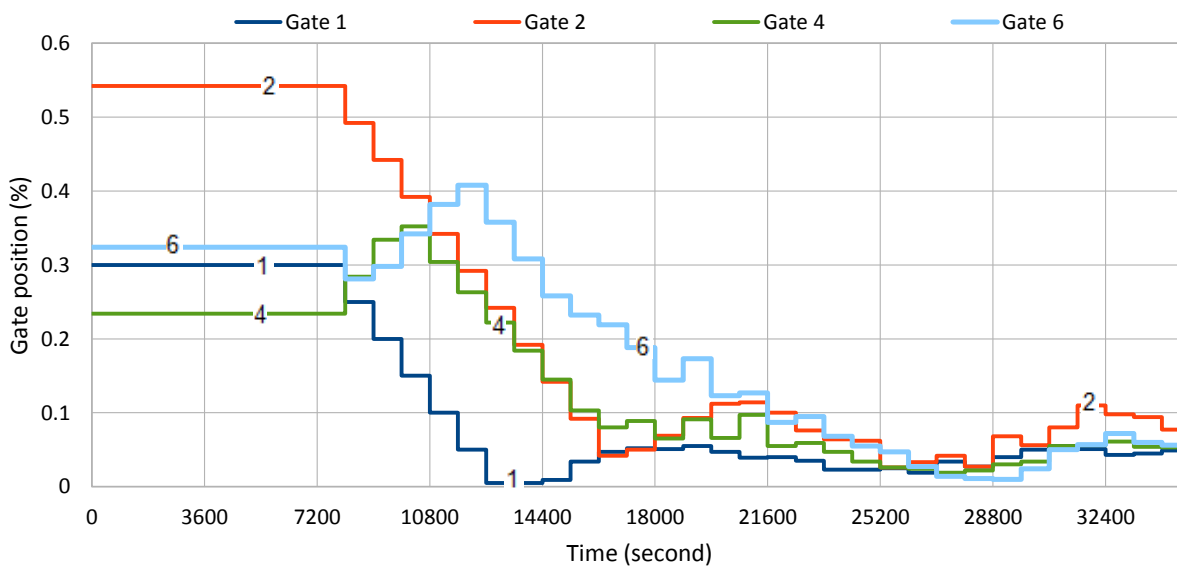


Figure 11.19: Gate trajectories (1, 2, 4 and 6) in Test-Case 2-2 (Corning canal).

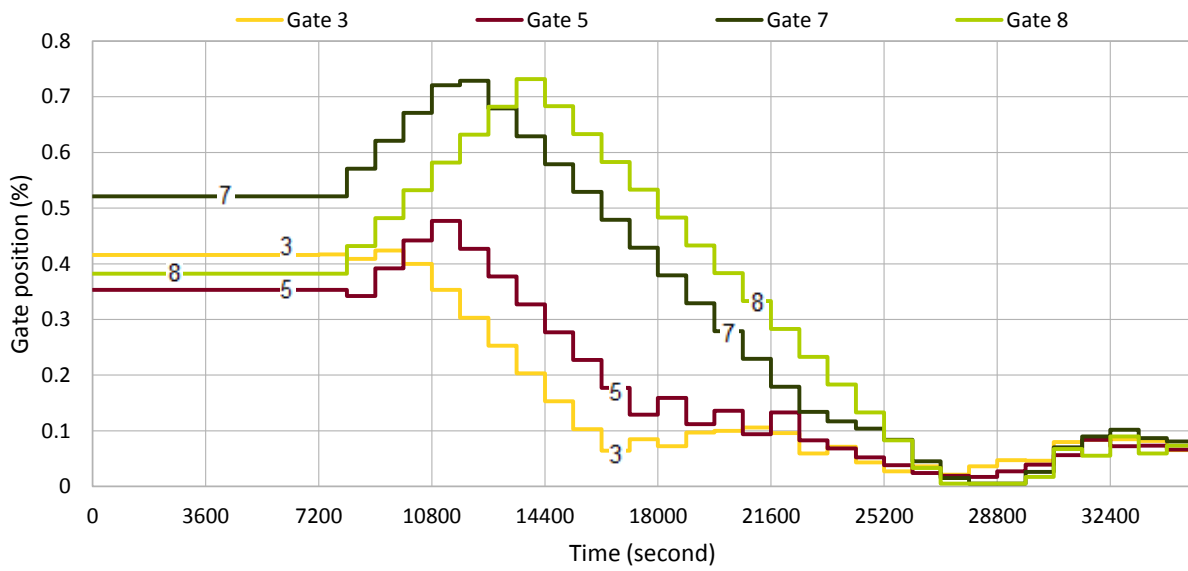


Figure 11.20: Gate trajectories (3, 5, 7 and 8) in Test-Case 2-2 (Corning canal).

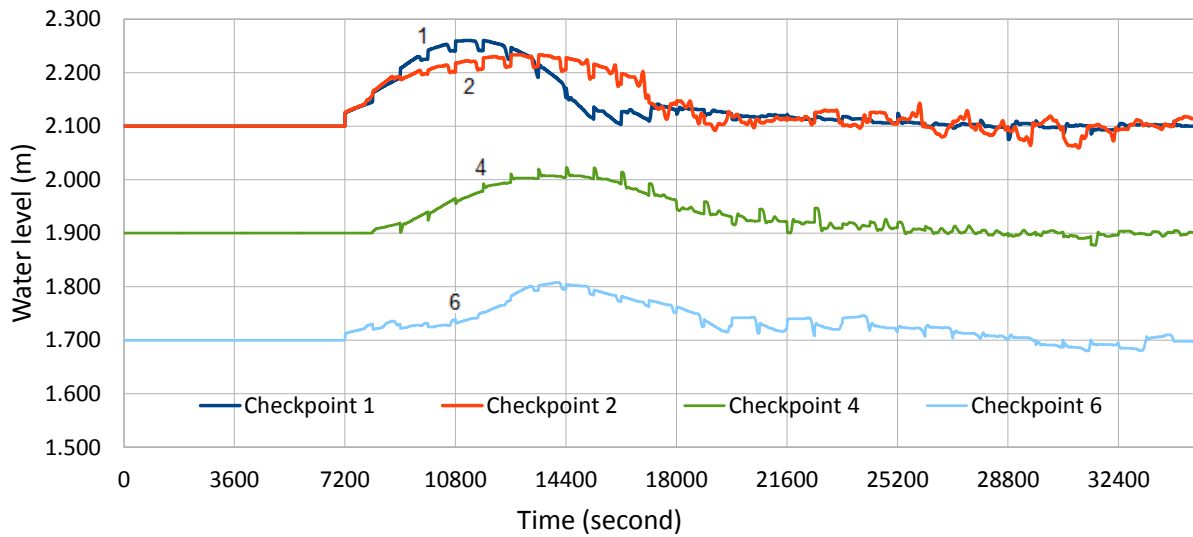


Figure 11.21: Water level at checkpoints 1, 2, 4 and 6 in Test-Case 2-2 (Corning canal).

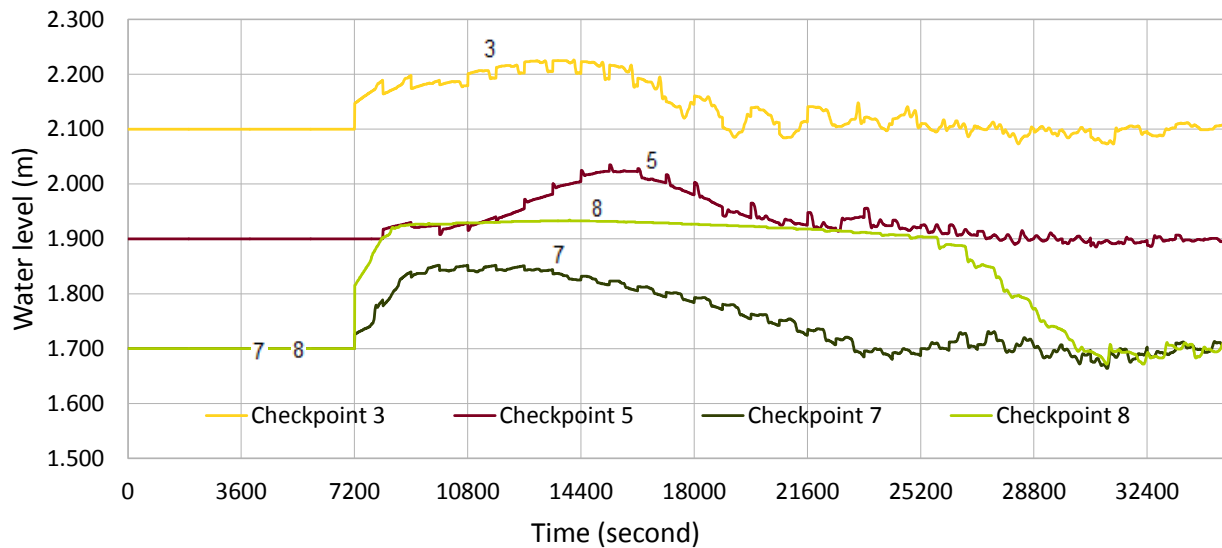


Figure 11.22: Water level at checkpoints 3, 5, 7 and 8 in Test-Case 2-2 (Corning canal).

11.6. Performance indicators

These indicators were introduced to compare different control algorithms, as it is not easy to evaluate them and judge the controller's ability to deliver water. This is the reason why the ASCE Task Committee devised these test-cases.

Although not all these performance indicators evaluate the deviations between the measured and the desired water level at the checkpoints, they also take into account other variables as the changes in flow too.

There are many ways to assess the error between the water level measured and the desired at checkpoints. First, we might calculate the maximum error at the checkpoint. Second, we might calculate how much time the control algorithm needs to remove the error at the checkpoint. Third, we might calculate the water level error at the checkpoint at the end of the test, that is, the water level should not stabilize at a wrong level. The fourth performance indicator controls the variations of flow rate between the initial flow through the gate and the final flow. These four performance indicators are recommended by the ASCE Task Committee, and we show the maximum and the average of these performance indicators, as suggested by the Task Committee.

11.6.1. Maximum Absolute Error (MAE)

$$MAE = \frac{\max(|y_t - y_{target}|)}{y_{target}} \quad (11.3)$$

where y_t is the observed or measured (computed from simulation) water level at time t ; and y_{target} is the target or desired water level.

11.6.2. Integral of Absolute Magnitude of Error (IAE)

$$IAE = \frac{\frac{\Delta t}{T} \sum_{t=0}^T |y_t - y_{target}|}{y_{target}} \quad (11.4)$$

where Δt is the regulation time step; and T is the time period for test (12 hours). The time step must be held constant over the 12 hours interval specified.

11.6.3. Steady-State Error (StE)

The steady-state error is defined as the maximum of the average error over the last two hours of the test. The conclusion is that the conditions should be stable during this period of time.

$$StE = \frac{\max(|y_{22,24} - y_{target}|)}{y_{target}} \quad (11.5)$$

where $y_{22,24}$ is the average water level between 10 and 12 hours. Some control algorithms have not performed well because of constant oscillation around the sluice gate trajectory solution. To detect excessive oscillations in water levels at the end of the test, this indicator is proposed.

11.6.4. Integrated Absolute Discharge Change (IAQ)

To avoid excessive oscillation of sluice gates during the irrigation cycle, this performance indicator related to the changes in discharge due to the sluice gate movements is proposed.

$$IAQ = \sum_{t=t_1}^{t_2} (|Q_t - Q_{t-1}|) - |Q_{t_1} - Q_{t_2}| \quad (11.6)$$

Where Q_t is the check gate discharge at time t , so Q_{t-1} is the gate discharge at the previous time step. The second term is simply the difference between the initial and the final flow rate for the irrigation cycle. In that sense, the performance indicator IAQ could be zero, in case that $Q_{t1} > Q_{t2}$ and $Q_{t-1} > Q_t$ for all time steps during the period of time (t_1-t_2) , or in case that, $Q_{t1} < Q_{t2}$ and $Q_{t-1} < Q_t$ for all time steps during the same period of time. Use of this performance indicator requires that the times for flows introduced in the summation include the times at all reversals in the sign of flow changes (all the peaks and valleys of the flow hydrograph are included to set this performance indicator).

11.7. Comparison between controllers

We compare the performance indicators obtained with GoRoSoBo in Test-cases with the performance indicators obtained with other control algorithms as CLIS (Liu et al. 1998) and Pilote (Malaterre et al. (1995)). The CLIS is based on an inverse solution method of the Saint Venant equations, and it is designed for the automation of demand oriented systems. The Pilote is a LQR closed-loop controller and it is obtained from the steady-state solution of the Riccati equation. A Kalman Filter is used to reconstruct the state variables and the unknown perturbations from a reduced number of observed variables.

11.7.1 Test-Case 1-1 (Maricopa Stanfield)

In this test 1-1, GoRoSoBo shows the best values of maximum and average MAE performance indicator and Pilote shows the second best values of the maximum MAE and CLIS shows the second best values of the average MAE. The maximum and average MAE values for GoRoSoBo are 8.5 % and 6.9 %, respectively, see Table 11.20.

The GoRoSoBo shows the best results of the maximum IAE indicator (1.6%) but it does not show the best result of the average IAE indicator which is very close of the best result obtained by Pilote. The values of the average IAE indicator of both control algorithms are 1.38% and 1.28%, respectively. This is the only test that GoRoSoBo does not show the best result of this performance indicator and the difference with the best control algorithm (Pilote) is only 0.1%.

The GoRoSoBo shows the second best values on the maximum StE indicator for this test-case (2.4%), whereas Pilote shows the best results 0.35%. The values obtained by GoRoSoBo and CLIS in the average StE indicator are similar (1.39% and 1.4%, respectively) whereas the values obtained by Pilote are excellent (0.12%).

GoRoSoBo algorithm gives low values of this indicator in all test-cases. However it is possible to make a modification on the control algorithm to reduce this indicator to zero. Once CSI algorithm calculates in several regulation periods that the unscheduled offtake changes are constant and close to an exact value, CSI can fix this constant value as unscheduled offtake changes. In that way, GoRoSoBo will obtain the same gate trajectories every regulation period, so the water level in all checkpoints would be the desired water level and the canal state would be steady. But the results of StE performance indicator obtained by GoRoSoBo algorithm are enough good in these tests to introduce this modification in the control algorithm.

Instead, the values obtained by GoRoSoBo in the IAQ indicator are not good. These values are far away of the values obtained by CLIS which are close to zero. We can conclude that the optimum gate trajectories obtained by GoRoSoBo show the best values of MAE and IAE indicators, but these gate trajectories are not the best to obtain good values in IAQ indicator.

The IAE performance indicator establishes the total changes on flow rate through the gates introduced by the gate trajectories to reach the desired final flow rate. When the unscheduled offtake changes introduced to the system are important, it is necessary quick changes on gate trajectories to recover the desired value at checkpoints as soon as possible, so the changes on flow rate will be significant too. Limiting more the constraints about gate movements to obtain better values of this indicator is not our principal objective, because the MAE and StE indicators will be worse.

	TEST CASE 1-1 TUNED-UNSCHEDULED							
	MAE (%)		IAE (%)		StE (%)		IAQ (m3/s)	
	12-24h		12-24h		12-24h		12-24h	
	Maximum	Average	Maximum	Average	Maximum	Average	Maximum	Average
CLIS	19,9	7,8	3,8	1,8	2,6	1,4	0,2	0,1
PILOTE	13,4	8,6	1,8	1,3	0,4	0,1	2,5	1,7
GoRoSoBo	8,5	6,9	1,6	1,4	2,4	1,4	8,9	5,3

Table 11.20: The performance indicators obtained in Test-Case 1-1 (Maricopa Stanfield).

11.7.2 Test-Case 1-2 (Maricopa Stanfield)

As it has happened in the last test cases, GoRoSoBo shows the best results of maximum and average MAE and IAE indicators. The maximum MAE value on GoRoSoBo is very similar to CLIS, 33'5 % and 34'5 %, respectively, although GoRoSoBo obtains excellent values of average MAE indicator (10.3 %) comparing with CLIS and Pilote (14.2% and 24.9 %, respectively), see Table 11.21.

Taking into account the maximum IAE indicator, CLIS and GoRoSoBo obtained the best results around of 5%, whereas Pilote obtained the worst results (9.2%). As regards the average IAE indicator, GoRoSoBo obtained the best results (1.2%), the second one was CLIS (2%) and the third one was Pilote (5.2%).

GoRoSoBo was the best in both performance indicators (StE), whereas CLIS obtained better results than Pilote. As regards the IAQ indicator, GoRoSoBo obtained values quite similar to Test Case 1-1, we could arrive at the same conclusion about IAQ than the last one.

	TEST CASE 1-2 TUNED-UNSCHEDULED							
	MAE (%)		IAE (%)		StE (%)		IAQ (m3/s)	
	12-24h		12-24h		12-24h		12-24h	
	Maximum	Average	Maximum	Average	Maximum	Average	Maximum	Average
CLIS	34,5	14,2	5,0	2,0	3,6	1,1	0,2	0,1
PILOTE	43,0	24,9	9,2	5,2	11,2	2,9	2,9	1,4
GoRoSoBo	33,5	10,3	5,0	1,2	2,1	0,7	6,7	3,6

Table 11.21: The performance indicators obtained in Test-Case 1-2 (Maricopa Stanfield).

11.7.3 Test-Case 2-1 (Corning canal)

In this Test Case 2-1, GoRoSoBo also shows the best maximum and average MAE indicator (6.1% and 4.2%, respectively), the second one was CLIS which obtained a values of 6.8% and 4.4%, respectively and third one was Pilote which obtained a values of 7.5% and 4.8%, respectively.

The results about the IAE indicator are similar in all control algorithms, although the values obtained by CLIS in the maximum and the average IAE are difficult to reduce 1.4% and 1%, respectively, GoRoSoBo obtains the best values 0.7% and 0.5%.

The maximum and average StE values are similar between the control algorithms, the better results are obtained by Pilote, and GoRoSoBo shows the second best results, although the values of StE indicator are very low in all control algorithms, below 0.5 %.

The IAQ performance indicator is when worst values obtain GoRoSoBo in this Test Case 2-2. Although, as we introduced previously, the main objective is drive the canal state to the desired state as soon as possible and this forces to the algorithm calculates significant changes in gate trajectories in those tests which is the opposite to move the gates progressively to avoid significant changes in flow through the gates.

	MAE (%)		IAE (%)		StE (%)		IAQ (m3/s)	
	12-24h		12-24h		12-24h		12-24h	
	Maximum	Average	Maximum	Average	Maximum	Average	Maximum	Average
CLIS	6,8	4,4	1,4	1,0	0,5	0,4	5,8	3,0
PILOTE	7,5	4,8	2,0	1,3	0,2	0,1	7,6	4,7
GoRoSoBo	6,1	4,2	0,7	0,5	0,3	0,2	17,1	5,2

Table 11.22: The performance indicators obtained in Test-Case 2-1 (Corning canal).

11.7.4 Test-Case 2-2 (Corning canal)

GoRoSoBo shows a maximum MAE value quite lower (13.7%) than the value obtained by others controllers as CLIS and Pilote (21.1% and 34.2%, respectively), see Table 11.23. GoRoSoBo reduces the maximum MAE indicator in 35% (CLIS) and 60 % (Pilote), respectively. The average MAE indicator shows a similar tendency to the maximum MAE,

whereas GoRoSoBo obtains an average MAE value around of 7.8%, the CLIS and Pilote obtain a value of 14.9 % and 17.1 %, respectively.

Checking this performance indicator, the GoRoSoBo obtains the better values, whereas the maximum and average IAE indicators show similar values in GoRoSoBo (6.3% and 2.1%, respectively) and CLIS (7.6% and 2.8%, respectively). Pilote obtains worst results on this performance indicator (10.6% and 7.1%).

GoRoSoBo is not the best control algorithm in the StE indicator in this Test Case 2-2, but the value of the maximum StE indicator is very close to CLIS, 0.8% and 0.7%, respectively. The average StE indicator for GoRoSoBo and CLIS are very close too, 0.5% and 0.4%, respectively.

The values of IAQ performance indicator were not good in this test. The best values of IAQ indicator were obtained by CLIS.

In a general view, the results of all control algorithms are similar in the performance indicators. We can conclude that GoRoSoBo is a good control algorithm. The performance indicator values obtained by GoRoSoBo are quite good, and most specially, the MAE and IAE indicators. The biggest difference between GoRoSoBo and CLIS and Pilote is, among other things, that GoRoSoBo calculates the optimum gate trajectories using the HIM matrix which is recalculated in every time step, so this matrix is updated in real time and it is not an approximation. In this way, the relation between water level error and gates movements is evaluated in every time step.

We propose several things to improve the value of IAQ in those tests. For instance, we could reduce the functional constraints ($dU_{0,max}$ and dU_{max}), if the gate movements are more restricted so there are less changes in gate trajectories and so less changes in flow through the gates. We could also increase the duration of the operation period, if there are less changes in gate positions during the irrigation cycle there are less changes in flow through the gates. We have to remember that all these changes would improve the IAQ indicator but we would obtain worst values in MAE and IAE indicators.

	TEST CASE 2-2 TUNED-UNSCHEDULED							
	MAE (%)		IAE (%)		StE (%)		IAQ (m3/s)	
	12-24h		12-24h		12-24h		12-24h	
	Maximum	Average	Maximum	Average	Maximum	Average	Maximum	Average
CLIS	21,1	14,9	7,6	2,8	0,7	0,4	9,7	5,5
PILOTE	34,2	17,1	10,6	7,1	8,8	4,3	10,4	6,1
GoRoSoBo	13,6	7,8	6,3	2,1	0,8	0,5	15,2	11,7

Table 11.23: The performance indicators obtained in Test-Case 2-2 (Corning canal).

11.8. Conclusions

Once we have analysed all test-cases and the performance indicators, we can conclude:

1. The test-cases checked in these examples are extremely difficult especially the Test Case 1-2 and 2-2. In test 2-2 is introduced to the system a disturbance at the checkpoint 8 with a value of 80 % on the total flow rate conveyed at checkpoint 8 and the water level reaches the top of the cross-section.
2. The unscheduled deliveries calculated by CSI are quite accurate, with a maximum error of 4%. The extracted flow vector obtained by CSI is very important, because CSI obtain the current canal state and the disturbance. All these data aid to GoRoSoBo to calculate the optimum gate trajectories.
3. The gate constraints drive the algorithm's response when the measured water levels is far away from the desired water level at the checkpoints and in this way the gate trajectory is smoother.
4. Taking into account the performance indicators, we can establish a comparison between the different control algorithms:
 - The MAE is the maximum deviation of the controlled water level at the checkpoint with regard to the desired water level. From the values obtained by GoRoSoBo with this performance indicator, GoRoSoBo shows the best results in all tests.
 - The IAE is the integrated deviation of controlled water level on the target water depth. From the maximum and average IAE values obtained by GoRoSoBo, this algorithm shows the best results in seven of eight cases.
 - The StE is the deviation of controlled water level at steady state over target water depth, so we only consider the last two hours of the irrigation cycle to

calculate this performance indicator. GoRoSoBo obtained the best results in two cases, the second one in five cases, and the third one in one case.

- The IAQ gives us an idea of how many changes on flow rate are introduced by the gates to reach the desired flow rate at the end of the test. GoRoSoBo shows the worst values on this performance indicator. When a control algorithm has, as an only objective, to maintain a desired water level at the checkpoints, the gate trajectories are calculated for that objective and there are performance indicators more benefited (as MAE/IAE) than others (as IAQ). IAQ indicator does not consider an important thing, the flow rate can change a lot in a target, but if the water level does not vary significantly, the flow rate extracted by the orifice offtake will be the same, because the gravity offtake is not very sensitive to little changes in water level, for that reason it is so important maintain the water level at the target. In that sense, when significant flow changes are introduced in the system, it is necessary quick changes on gate trajectories to recover the desired value at checkpoints as soon as possible. For this reason, limiting more the constraints in gate movements to obtain better values of this performance indicator is not our principal objective. We can conclude that the magnitude of flow changes through the gates cannot be very important because the control algorithm maintains the measured water level close to the desired water level and it exists functional constraint to restrict the gate movement.
5. As we introduced in chapter 9, in case of significant unscheduled flow changes, we have to update the HIM matrix at every regulation period, because it establishes the influence of a gate movement on the hydrodynamic canal state which is significantly modified every regulation period during the irrigation cycle in the Test Cases.
 6. The main objective of GoRoSoBo was reached if we look at the results obtained in sections 11.5 and 11.7.
 7. The calculating time using CSI and GoRoSoBo in a feedback strategy is quite large and it is dependent on the duration of the prediction horizon, the number of checkpoints, control structures,... In the Corning canal, the calculating time for a prediction horizon of 12 hours is approximately fifty minutes for every regulation period with a PC with a processor speed of 2GHz, a system RAM of 2 GB and a hard drive of 250 Gb. We could reduce the calculation time decreasing the duration of the prediction horizon, parallelizing the algorithm code with OpenMP, MPI or CUDA. The

algorithm wastes a lot of time doing a high number of iterations in the optimization problem, when the residual value of the objective function is under 1×10^{-4} we could finish the process. Other possibility would be not updating the HIM when there is not a significant disturbance because this step is the most time demanding or using the HIM at the previous time step whereas other algorithm calculates the HIM at the current time step at the same time for being used by GoRoSoBo in the next time step.

Chapter 12

Conclusions and Future Work

12.1 Summary of conclusions

The algorithms proposed (GoRoSoBo and CSI), developed and tested in this thesis, show good results in all tests. GoRoSoBo establishes the gate trajectories that satisfy a target required whereas CSI calculates the history of extracted flow in a canal. These algorithms can operate together in a feedback strategy during an irrigation cycle. The CSI algorithm calculates the extracted flow vector and the current canal state in every regulation time step and GoRoSoBo algorithm recomputes the gate trajectories considering the extracted flow vector obtained by CSI to drive the system to the desired behavior.

12.2 Conclusions of CSI

The CSI algorithm is able to estimate any change in flow from a past time to the current time so any disturbance that modifies the free surface is characterized by this algorithm.

CSI algorithm uses the Levenberg-Marquardt method to solve an unconstrained optimization problem. CSI is a robust algorithm which does not need constraints and it converges very fast. The hydraulic influence matrix obtained analytically is the tool which the CSI uses to establish a relation between changes of water level at checkpoints and the extracted flow at a particular section.

The test results have shown the accuracy of this algorithm (chapter 7 and 8). In all examples, the CSI algorithm finds the disturbances introduced into the canal. An analysis of the CSI as follow:

- CSI is able to estimate big disturbances (20 % of the total flow rate in the canal) and small disturbances (1 % of the total flow rate in the canal) accurately.
- Every extraction point should have a sensor measuring the water level. But in case a disturbance is not localized at a checkpoint, it is possible to set the total water volume extracted and localize the outflow point in the canal.
- The Manning roughness coefficient is extremely important in CSI, because the algorithm is very sensitive to this coefficient. For that reason, we have to establish it accurately.
- The initial flow conditions are not important data to supply CSI, because sometime after the test begun, the initial conditions did not affect the results of CSI. This period depends on the canal and flow features (canal length, flow conditions,...).
- The water level measurements should be accurate because the results obtained by CSI are extremely dependent on the water level measures. Even an error in depth gages of 1% from the total water level is acceptable for CSI.
- The water level error must be lower than the water level variation caused by the disturbance, but CSI could not calculate the disturbance accurately.
- Once CSI calculates the extracted flow vector, the algorithm can establish the hydrodynamic canal state, water level and velocity in all sections of the canal.

12.3 Conclusions of GoRoSoBo

The GoRoSoBo algorithm is able to find the optimum gate trajectory during a predictive horizon from demand deliveries, initial gate trajectories, desired water level vector, disturbances and the current canal state which are obtained by CSI. All these data is introduced to GoRoSoBo and the algorithm recomputes the optimum gate trajectory to keep constant the desired water level at checkpoints.

The GoRoSoBo algorithm uses the Lagrange-Newton method to solve a constrained optimization problem. This method is considered the most efficient when you have compiled

the Jacobian matrix and the Hessian matrix which are used in the computation of the gate trajectories.

The introduction of constraints was absolutely necessary to ensure stability in our optimized problem, due to inherent instability in the unconstrained problem, which is caused by the condition number of the Hessian and the HIM matrixes. Not all elements of the HIM have similar values, there are gates that have a significant influence in certain checkpoints and little influence in others. This disparity of influence between elements of the matrix inevitably leads to a band matrix, of course, badly conditioned. This was a reason to use the Marquardt coefficient which improves the Hessian matrix condition.

The watermaster can be more or less strict about the constraints in the optimization problem, because he must take care about the main priority of the canal. For instance, if the main priority is maintain constant the water level at any price, the constraints on the gates would be lax, so the constraints have not an important role in computing the gate trajectories by GoRoSoBo. If the main priority is to keep constant the water level, with a reduced gate movements, the constraints would be more restrictive, so they will have an important role in the optimization problem.

As we shown in chapter 9. GoRoSoBo needs to consider a forecasted prediction of disturbances to calculate the optimum gate trajectories, because the algorithm has to calculate gate trajectories at present time to drive the system in future and some changes in gate positions at present time are needed to prevent water level variations in future. In case that the disturbances disappear in future, our forecasted prediction of disturbances would be wrong, but the results obtained with this strategy in GoRoSoBo are good. These results will be more or less accurate depending on the duration of regulation period that CSI comes back to calculate the new extracted flow in the canal.

The performance indexes in ASCE Test-Cases have shown the accuracy of the algorithm, especially as regards to the indexes involved in the water level errors (MAE and IAE). GoRoSoBo was the best controller proposed in all test case for these indexes. The gate trajectory obtained with GoRoSoBo is the one that gets a lower water level error at checkpoints and reaches faster the desired water level at checkpoints.

Not all results obtained by GoRoSoBo are excellent, as for instance:

- The IAQ value is much bigger in GoRoSoBo than in other controller proposed. This index is strongly linked to the main priority of the canal. In case that the priority is to keep constant the water level at the checkpoints, the gate movements are significant and the flow rate variations through the sluice gate too, so the IAQ value will be also significant.
- The GoRoSoBo algorithm obtained good results about the STE index, although the algorithm has great scope for improvement. We could restrict the gate movements when the water level error is lower than a certain threshold, (we could introduce a dead band).

12.4 Future Work

Following the investigations described in this thesis, a number of projects could be taken up:

- Testing the CSI algorithm in the Canal PAC-UPC, using other pool configurations and other flow conditions. In this laboratory canal, it is possible to change the working conditions with small effort, providing an excellent opportunity to extend the results obtained in this thesis.
- A further study on the performance index with GoRoSoBo in other kind of canals and in other hydrodynamic conditions used by other feedback algorithms.
- Developing a new block in our overall control diagram of irrigation canal of chapter 5. It would be interesting to develop the Off-line Parameter Identification. The Manning roughness coefficient is fairly difficult to define in a canal and it could change a lot after a large period of time. On the other hand, the Manning roughness coefficient has a great weight in the Saint-Venant equation as in CSI and GoRoSoBo.
- Parallelizing the code of GoRoSoBo algorithm with CUDA or OpenMP and speed-up the calculating process. In cases that the predictive horizon is large, the regulation interval is small, the canal length and the number of checkpoints are significant, the calculation time of the algorithm is too large for operating in real time. Much more if it is necessary to update the hydraulic influence matrix in every regulation time step. A real solution with the time conflict, it would be parallelize the algorithm with CUDA or OpenMP to reduce the CPU time. Other possibility to speed-up the calculating process would be that GoRoSoBo calculated the gate trajectories using the HIM in the previous time step, while at the same time, other algorithm calculated the HIM at the current time step to be used by GoRoSoBo in the next time step.

- We propose several things to improve the value of IAQ in these tests. We could reduce the functional constraints (dU_{0max} and dU_{max}), if the gate movements are more restricted so there are less changes in gate trajectories and so less changes of flow through the gates. We could also increase the operation period, if there are less changes in gate positions during the irrigation cycle there are less changes of flow through the gates. Other possibility would be penalize the changes of flow through the gates from the objective function. The objective function only depends in our optimization problem of the water level, we could reformulate the objective function so as this function was dependent of the error in water level at checkpoints and the flow through the gates.

REFERENCES

Abbott, M.B. (1966). Introduction to the Method of Characteristics. Thames and Hudson, London and American Elsevier. New York, US.

Adams, J.C. (2009). The Fortran 2003 handbook. Ed. Springer.

AEMET (Agencia Estatal de Meteorología). (2009). Brunet, M., Casado, M.J., de Castro, M. Generación de escenarios regionalizados de cambio climático para España. Ministerio de Agricultura, Pesca y Alimentación .España.

Aguilar, J., Langarita, P., Linares, L., Rodellar, J., and Soler, J. (2012). Adaptive predictive expert control of levels in large canals for irrigation water distribution. International Journal of Adaptive Control and Signal Processing, 10.1002/acs.2272, 945-960.

Akouz K., Benhammou A. (1995) Modelling and Predictive Control of Irrigation Channels. M²SABI/IFAC. First International Symposium on Mathematical Modelling and Simulation in Agriculture & Bio-industries. Université Libre de Bruxelles.

Alvarez B. X. (2004). Control predictivo de canales de riego utilizando modelos de predicción de tipo Muskingum (primer orden) y de tipo Hayami (segundo orden). Master's thesis, Technical University of Catalonia, Spain.

Alvarez A., Ridao M.A, Ramirez D.R. and Sanchez L. (2013). Constrained Predictive Control of an Irrigation Canal. Journal of Irrigation and Drainage Engineering, ASCE, Vol. 139, No. 10, pp. 841-854.

Amein, M. (1966). Streamflow Routing on Computer by Characteristics. Water Resources Research, Vol.2, No. 1, and pp.123-130.

Amein, M. and Fang, C.S. (1970). Implicit Flood Routing in Natural Channels. Journal of the Hydraulics Division, ASCE, Vol.96, NoHY12, pp.2481-2500.

Amein, M. and Chu, H.L. (1975). Implicit Numerical Modelling of Unsteady Flows. Journal of the Hydraulics Division, ASCE, Vol.101, No.HY6, pp.93-108.

Ames, W.F. (1977). Numerical Methods for Partial Differential Equations. Academic Press Inc. Orlando, Florida, US.

Balogun O.S. (1985). Design of real-time feedback control for canal systems using linear quadratic regulator theory, PhD thesis, Department of Mechanical Engineering, University of California at Davis, 230 p.

Balogun O.S, Hubbard M. & DeVries J.J. (1988). Automatic control of canal flow using linear quadratic regulator theory. Journal of Hydraulic Engineering, Vol. 114, n° 1, pp. 75-102.

Baltzer, R.A. and Lai, C. (1968). Computer Simulation of Unsteady Flow in Waterways. Journal of the Hydraulics Division. ASCE. Vol.94, No. HY4, pp.1083-117.

Baume J. P., Sally H., Malaterre P.O. & Rey J. (1993). Development and field-installation of a mathematical simulation model in support of irrigation canal management. IIMI and CEMAGREF Research paper, 89 p.

Bautista, E., Clemmens, A.J., Strelkoff T. (1997). Comparison of numerical procedures for gate stroking. Journal of Irrigation and Drainage Engineering. ASCE, 113(2), pp.129-136.

Bautista, A.M., Strelkoff, T.S. & Clemmens, A.J. (2003). General Characteristics of Solutions to the Open-Channel Flow, Feedforward Control Problem. Journal of Irrigation and Drainage Engineering. ASCE, Vol. 129, No. 2, pp. 129-137.

Bautista, E., Clemmens, A. J. and Strand. (2006). Salt River Project Canal Automation Pilot Project: Simulation Tests. Journal of Irrigation and Drainage Engineering. ASCE, pp.143-152.

Bedient, B., and Huber, C. (2002). Hydrology and Floodplain Analysis. Addison-Wesley, New York.

Begovich, O., Ruiz, V. M., Georges, D., and Besancon, G. (2005). Real-time application of a fuzzy gain scheduling control scheme to a multi-pool open irrigation canal prototype. Journal of intelligent fuzzy systems, 16(3), pp.189-199.

Begovich, O., Felipe, J., and Ruiz V. (2007). Real-time implementation of decentralized control for an open irrigation canal prototype. *Asian Journal of Control*. Volume 9, Issue 2, pp.170–179.

Bos, M. G., Burton, M. A., & Molden, D. J. (2005). *Irrigation and drainage performance assessment: Practical guidelines*. Wallingford: CABI Publishing.

Bos, M. G., & Nugteren, J. (1990). *On irrigation efficiencies* (4th ed.). Wageningen: International Institute for Land Reclamation and Improvement.

Brogan, W.L. (1985). *Modern Control Theory*. Prentice-Hall, New Jersey, US.

Bruinsma, J. (2009). *The Resource Outlook to 2050: By How Much do Land, Water and Crop Yields Need to Increase by 2050?*. Prepared for the FAO Expert Meeting on 'How to Feed the World in 2050', 24–26 June 2009, Rome.

Burt, C.M. (1983). *Regulation of sloping canals by automatic downstream control*. PhD. Utah State University. US.

Buyalski, C.P., Serfozo, E.A. (1979). *Electronic filter level offset (ELFLO) plus reset equipment for automatic downstream control of canals*. Technical Report REC-ERC-79-3, U.S., Department of Interior, Bureau of Reclamation, Denver, US.

Buyalski, C. P., Ehler, D. G., Falvey, H. T., and Seffozo, E. A. (1991). *Canal system automation manual, Vol. 1*. Department of the Interior, Bureau of Reclamation, Denver, U.S.

Cardona, J., Gómez, M., and Rodellar, J. (1997). *A decentralized adaptive predictive controller for irrigation canals*. In *Proceedings of the International Workshop on the Regulation of Irrigation Canals: State of the Art of Research and Applications, RIC97*, volume 1, pp. 215-219, Marrakech, Morocco.

Camacho, E. F. and Bordons, C. (2004). *Model Predictive Control*. Springer-Verlag, London, second edition.

Chan S.P. & Yao B. (1990). *Performance study of GPC for processes with time delay*. IEEE, pp. 384-389.

Chavez A.A., Gonzalez E.P., Kosuth P., Daval E. (1994). Informe final del proyecto automatizacion de canales SH 9408. IMTA, CEMAGREF, CNA, 83 p.

Chentouf, B. (2001). Robust regulation of a river reach governed by Hayami model. In Proceedings of the 40th IEEE conference on decision and control, pp. 4962-4967.

Chevereau, G. (1991). Contribution à l'étude de la régulation dans les systèmes hydrauliques à surface libre, Thèse de Doctorat de l'Institut National Polytechnic de Grenoble, FR.

Clemmens A.J. (1987). Delivery system schedules and required capacities. Proceedings ASCE Portland, Zimbelman D.D. (ed.).

Clemmens, A.J., Kacerek, T.F., Grawitz, B., Schuurmans, W. (1998). Test cases for canals control algorithms. Journal of Irrigation and Drainage Engineering. ASCE, 124 (1), pp. 23-30.

Clemmens, A. J. and Wahlin, B. T. (2004). Simple Optimal Downstream Feedback Canal Controllers: ASCE Test Case Results. Journal of Irrigation and Drainage Engineering. ASCE, pp.35-46.

Clemmens, A.J. & Schuurmans, J. (2004). Simple Optimal Downstream Feedback Canal Controllers: Theory. Journal of Irrigation and Drainage Engineering, ASCE, pp.26-34.

Clemmens, A. J., Bautista, E., Wahlin, B. T., and Strand, R. J. (2005). Simulation of Automatic Canal Control Systems. Journal of Irrigation and Drainage Engineering. ASCE, pp.324-335.

Clemmens, A.J., Strand, R.J., Bautista, E. (2005). Field testing of SacMan automated canal control system, Proceedings of USCID Third International Conference on Irrigation and Drainage, San Luis Obispo, California.

Coeuret C. (1977). Stabilité et précision de la régulation dynamique. La Houille Blanche n° 2/3, pp 272-277.

Corriga G., Sanna S. & Usai G. (1982). Sub-optimal level control of open-channels. Proceedings International AMSE conference Modelling & Simulation, Vol. 2, pp. 67-72.

Corriga G., Sanna S. & Usai G. (1983). Sub-optimal constant-volume control for open channel networks. *Appl. Math. Modelling*, Vol. 7, pp. 262-267.

Courant, R., Hilbert, K.O. (1962). *Methods of Mathematical Physics*, V.II, Ed. John Wiley & Sons, US.

Crandall. S.H. (1956). *Engineering Analysis-A Survey of Numerical Procedures*, Ed. Willey, New York, US.

Cunge, J. (1969). On the subject of a flood propagation computation Method (Muskingum Method)", *Journal of Hydraulic Research*, 7(2), pp. 205-230.

Cunge, J.A., Holly, F.M.Jr., Verwey, A. (1980). *Practical Aspects of Computational River Hydraulics*, Pitman, London, UK.

De Leon Mojarro B. (1986). *Contribution à l'amélioration de la gestion des périmètres irrigués*, Thèse de Doctorat, USTL Montpellier, 137 p.

De Leon B., Kosuth P., Argueta J. (1992). Use of the SIC simulation model to improve the management of an irrigation canal in la Begoña, Guanajuato, Mexico, CEMAGREF-IIMI International Workshop, Montpellier, October 1992, pp. 173-187.

Delgoda, D.K., Saleem, S.K, Halgamuge, M.N. (2013). Multiple Model Predictive Flood control in Regulated River System with Uncertain Inflows. *Water Resources Management*. Volume 27, Issue 3, pp. 765-790.

Deltour, J.L. (1992). *Application de l'automatique numérique à la régulation des canaux; Proposition d'une méthodologie d'étude*, Thèse de Doctorat, Institut National Polytechnic de Grenoble, FR.

Dooge, J. C. I., Strupczewski, W. B., and Napiorkowski, J. J. (1982). Hydrodynamic derivation of storage parameters of the Muskingum model", *J. Hydrol.*, 54, pp.371-387.

Durdu, O. F. (2004). Optimal control of irrigation canals using recurrent dynamic neural network (RDNN). In *Critical Transitions in Water and Environmental Resources Management*, pp. 1-12.

Ellis, J. (1970). Unsteady Flow in Channel of Variable Cross Section. *Journal of the Hydraulics Division, ASCE*, Vol.96, No.HY10, pp. 1927-1945.

FAO (Food and Agriculture Organization of the United Nations). (2011a). *The State of the World's Land and Water Resources: Managing Systems at Risk*. London, Earthscan.

FAO (Food and Agriculture Organization of the United Nations). (2011b). AQUASTAT onlinedatabase.Rome,FAO.<http://www.fao.org/nr/water/aquastat/data/query/index.html>.

Fereres, E.; Goldhamer, D.A. (2000). Avances recientes en la programación de los riegos. *Ingeniería del Agua*, vol. 7, núm. 1.

Filipovic V. & Milosevic Z. (1989). Dyn2 method for optimal control of water flow in open channels. *Journal of irrigation and drainage engineering*. Vol. 115, n° 6, pp. 973-981.

Fletcher, R. (1987). *Practical Methods of Optimization*, 2nd edition. John Wiley & Sons, UK.

Fread, D.L. (1985). Channel routing, in *Hydrological Forecasting*, Anderson, M.G. and Burt, T.P., eds., John Wiley and Sons, New York, pp. 437-503.

Garcia A., Hubbard M., De Vries J. J. (1992). Open channel transient flow control by discrete time LQR methods. *Journal Automatica (Journal of IFAC)*, Vol.28 Issue 2, pp.255-267.

Garrison, J.M., Granju, J.P. and Price, J.T. (1969). Unsteady Flow Simulation in Rivers and Reservoirs. *Journal of the Hydraulics Division, ASCE*, Vol.95, No.HY5, pp. 1559-1576.

Gómez, M. (1988). *Contribución al estudio del movimiento variable en lámina libre, en las redes de alcantarillado*. Aplicaciones, Tesi doctoral UPC, Barcelona.

Gómez, M., Rodellar, J., Vea, F., Mantecón, J., and Cardona, J. (1998). Decentralized predictive control of multi-reach canals. 1998 IEEE International Conference on Systems, Man, and Cybernetics, pp. 3885-3890.

Gómez M., Rodellar J., and Mantecon J.A. (2002). Predictive control method for decentralized operation of irrigation canals. *Applied Mathematical Modelling*. Elsevier.

Goussard, J. (1987). NEYRTEC automatic equipment for irrigation canals. Planning, operation, rehabilitation and automation of irrigation water delivery systems, D. C. Zimbelman, ed., ASCE, New York, N.Y.

Goussard, J. (1993). Automation of Canal Irrigation Systems. International Commission on Irrigation and Drainage (ICID), New Delhi.

Goussard, J. (1995). Interaction between water delivery and irrigation scheduling. Proceedings of the ICID/FAO Workshop on Irrigation Scheduling, Roma, Italia.

Gill, P.E., Murray, W., Wright, M.H. (1981). Practical Optimization. Academic Press Inc., Scotland.

Henderson, F.M. (1966). Open Channel Flow. The Macmillan Co., New York, N.Y.

Horváth, K. (2013). Model predictive control of resonance sensitive irrigations canals. PhD. Polytechnic University of Catalonia.

Huising, K.C. (2004). Slimme Stuwen. Master thesis, Delft University of Technology, The Netherlands.

Isapoor, S., Montazar, A., Van Overloop, P., and Van De Giesen, N. (2011). Designing and evaluating control systems of the Dez main canal. Irrigation and Drainage, 10.1002/ird.545, pp. 70-79.

Lemos J. M., Pinto L. F., Rato L. M., Rijo M. (2013). Multivariable and distributed LQG control of water delivery canal. Journal of Irrigation and Drainage Engineering, ASCE, Vol. 139, No. 10, pp. 855-863.

Khaladi A. (1992). Gestion automatique des transferts d'eau en réseaux hydrauliques maillés à surface libre; Application au réseau des Wateringues, Thèse de Doctorat, Ecole Centrale de Lyon, CEMAGREF, 237 p.

Kosuth P., Roux A., Lafaysse J.L. (1992). Use of a simulation model to improve operation of dam-river systems for irrigation purpose: elements for a methodology, CEMAGREF-IIMI International Workshop, Montpellier, pp. 81-90.

Kuhn, H.W., Tucker, A.V. (1951). Nonlinear programming, Proceedings of Second Berkeley Symposium on Mathematical Statistics and Probability. Ed. J. Neyman, University of California Press.

Lee T.H., Lai W.C. & Kwek K.H. (1990). Extended generalised predictive control incorporating feedforward. Proceedings of the American Control Conference, pp. 2703-2708.

Lefebvre, J., (1977). La régulation dynamique, sa mise en oeuvre au canal de provence. La Houille Blanche, pp.265-270.

Liggett, J.A. and Woolhiser, D.A. (1967). Difference Solutions of the Shallow-Water Equation. Journal of the Engineering Mechanics Division, ASCE, Vol395, No. EM2, pp.39-71.

Liggett, J.A., (1975). Basic equations of unsteady flow, Cap. 2 de Unsteady Flow in Open Channels. Water Resources Publications, Colorado, US.

Liggett, J.A., Cunge, J.A. (1975). Numerical methods of solution of the unsteady flow equations", Cap. 4 de Unsteady Flow in Open Channels, V.1, Water Resources Publications, Colorado, US.

Lin, Z., Manz, D.H. (1992). Optimal operation of irrigation canal Systems using nonlinear programming - dynamic simulation model. Cemagref-IIMI International Workshop, Montpellier, October 1992, pp. 297-306.

Linares, P., Ramos, A., Sanchez, P., Sarabia, A., Vitoriano, B. (2001). Modelos Matemáticos de Optimización, Universidad Pontífice Comillas (Madrid).

Linkens D.A., and Mahfouf M. (1992). Generalized predictive control with feedforward (GPCF) for multivariable anaesthesia. Int. J. Control, Vol 56, nº 5, pp. 1039-1057.

Liria, J. (2001). Canales hidráulicos. Proyecto, construcción, gestión y modernización, Col. Ing.de Cam., Can. y Puertos, Madrid.

Litrico, X., Fromion, V., Baume, J.P. and Rijo, M. (2003). Modelling and PI controller design for an irrigation canal, European Control Conference, Cambridge, U.K.

Litrice, X. and Fromion, V. (2006). Tuning of Robust Distant Downstream PI Controllers for an Irrigation Canal Pool. I: Theory. J. of Irrig. and Drain. Eng., ASCE, pp.359-368.

Liu, F., Feyen, J., Berlamont, J. (1992). Computation method for regulating unsteady flow in open channels. J. of Irrig. and Drain. Eng., ASCE, 118 (10), pp. 674-689.

Liu, F., Feyen, J., Berlamont, J. (1994). Downstream control algorithm for irrigation canals. J. of Irrig. and Drain. Eng., ASCE, 120 (3), pp.468-483.

Liu, F., Feyen, J., Berlamont, J. (1995). Downstream control of multireach canal systems. J. Irrig. Drain. Eng., ASCE, 121 (2), pp. 179–190.

Liu F., Feyen, J., Malaterre P.O., Baume J.P., and Kosuth P. (1998). Development and evaluation of canal automation algorithm CLIS. J. Irrig. And Drain. Eng., 124(1), pp. 40-46.

Luenberger, D.G. (1984). Linear and Nonlinear Programming, 2nd ed., Addison-Wesley, Massachusetts, US.

Malaterre, P.O. (1994). Modélisation, analyse et commande optimale LQR d'un canal d'irrigation. Tesi doctoral. CEMAGREF, Montpellier, FR.

Malaterre, P.O. (1995). PILOTE: optimal control of irrigation canals. First International Conference on Water Resources Engineering, Irrigation and Drainage, San Antonio, Texas, USA, 14-18 August 1995.

Malaterre, P.O. and Rodellar, J. (1996). Multivariable predictive control of irrigation canals. Design and evaluation on a 2-pool model", Journadas Hispano-Francesas: Sistemas inteligentes y control avanzado, Barcelona (Spain), UPC

Malaterre, P.O., and Baume, J.P. (1998). Modeling and regulation of irrigation canals: existing applications and ongoing researches. J. of Irrig. and Drain. Eng., ASCE, pp.203-208.

Malaterre, P. O., Rogers, D. C., Schuurmans J. (1998). Classification of Canal Control Algorithms. Journal of Irrigation and Drainage Engineering, Vol. 124, No. 1, pp. 3-10.

Malaterre, P.O., Baume, J.P. (1999). SIC 3.0, a simulation model for canal automation design. Cemagref (Montpellier).

Malaterre, Pierre-Olivier and Chateau, Christophe. (2007). SCADA interface of the SIC software for easy real time application of advanced regulation algorithms. A USCID Water Management Conference - Denver, Colorado

Martín-Sánchez, J.M., Rodellar, J. (1996). Adaptive Predictive Control: from the concepts to plant optimization, Series in Systems and control Engineering, Prentice Hall.

Marzouki T.Z. (1989). Le système hydraulique de la rivière Iarcis (barrages, rivière, périmètres irrigués). Modélisation hydraulique, analyse et amélioration de la gestion, CEMAGREF, ENGREF, CARA, 63 p.

Metcalf, M. and Reid, J. (1996). Fortran 90, 95 explained, 2nd Ed - Oxford University Press.

Ministerio de Medio Ambiente. (1998). Libro Blanco del Agua en España, MMA, Madrid.

Ministerio de Medio Ambiente. (2000). Libro Blanco del Agua en España, MIMAM, Madrid.

Montazar, van Overloop P. J., and Brouwer R. (2005). Centralized controller for the Narmada Main Canal. Irrigation and Drainage, 54(1), pp.79–89.

Moré J.J., Stephen Wright J. (1993). Optimization Software Guide. Philadelphia: SIAM.

Moussa, R. (1996). Analytical Hayami solution for the diffusive wave flood routing problem with lateral inflow. Hydrol. Process., VOL. 10, pp. 1209-1227.

NAG. (1991). Minimizing or Maximizing a Function. E04, The Numerical Algorithms Group Fortran Library Manual, (NAG), Mark 15, Oxford, U.K.

Numerical Recipes in FORTRAN 77. (1992). Cambridge University Press, ISBN 0 521 43064-X, UK.

Perrin G. (1989). Automatisation d'un système d'irrigation à gestion centralisée. Conservatoire National des Arts et Métiers, Centre Régional Associé de Metz, 154 p.

Piquereau A. & Villocel A. (1982). Gestion automatique des eaux d'été ; Cas de la rivière Arrats. ONERA, CERT/DERA Toulouse, CACG, 125 p.

Plusquellec H. (1988). Improving the Operation of Canal Irrigation Systems. An Audiovisual Production, The Economic Development Institute and the Agriculture and Rural Development Department of the World Bank, 154 p.

Plusquellec H., Burt C. & Wolter H.W. (1994). Modern Water Control in Irrigation. World Bank Technical Paper Number 246, Irrigation and Drainage Series, 104 p.

PNACC. (2011). http://www.magrama.gob.es/es/cambio-climatico/temas/impactos-vulnerabilidad-y-adaptacion/2_informe_seguimiento_pnacc_tcm7-197096.pdf. Accessed: 10/10/2013.

Oca J. (2009). El regadiu en Catalunya (UPC). Nota d'Economia, N°. 93-94, pp. 53-65. <https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CCEQFjAA&url=http%3A%2F%2Feconomia.gencat.cat%2Fweb%2Fcontent%2Fdocuments%2Farticles%2Farxius%2Felregadiuacatalunya.pdf&ei=li60VIOGNMmxaeiUgeAD&usq=AFQjCNGmRM1wYv03adiWjQbd7lv- TuJEQ>. Accessed: 19/05/2012.

Qing-Song Qiao and Kai-Lin Yang, (2010). Modeling Unsteady Open-Channel Flow for Controller Design. J. of Irrig. and Drain. Eng., ASCE, pp.383-391.

Reddy J.M., Dia A. & Oussou A. (1992). Design of control algorithm for operation of irrigation canals. Journal of Irrigation and Drainage Engineering, Vol. 118, n° 6, pp. 852- 867.

Reddy J.M. (1996). Design of global control algorithm for irrigation canals. J. Hydrol. Eng., 122 (9) (1996), pp. 503–511.

Rodellar, J., Gómez, M., Bonet, L. (1993). Control method for on demand operation of open-channel flow. J. of Irrig. and Drain. Eng., ASCE, 119 (2), pp.225-241.

Rogers, D.C., Goussard, J. (1998). Canal control algorithms currently in use. J. of Irrig. and Drain. Eng., ASCE, 124 (1), pp. 11-15.

Romanowicz, R. J., Dooge, J. C. I., and Kundzewicz, Z. W. (1988). Moments and cumulants of linearized St. Venant equation. Adv.Wat.Resour, .11: pp. 92-100.

Sabet M.H., Coe J.Q., Ramirez H.M. & Ford D.T. (1985). Optimal operation of California Aqueduct. *Journal of Water Resources Planning and Management*, Vol. 111, n° 2, pp. 222-237.

Sanders, B.F., Katapodes, N.D. (1999). Control of canal flow adjoint sensitivity method. *J. of Irrig. and Drain. Eng.*, ASCE, 125 (5), pp.287- 297.

Sawadogo S., Achaibou A.K. & Aguilar-Martin J. (1991). An application of adaptative predictive control to water distribution systems, IFAC, ITAC 91, Singapour, 6 p.

Sawadogo S., Achaibou A.K. & Aguilar-Martin J. (1992). Long-range predictive control of an hydraulic systems. CEMAGREF-IIMI Workshop, 9 p.

Sawadogo, S., Malaterre, P.O. and Kosuth, P. (1995). Multivariable optimal control for on-demand operation of irrigation canals. *The International Journal of Systems Science*, Vol 26:1, pp. 161-178.

Schaalje, M. and Manz, D.H. (1993). ANN controller, Personal communication.

Schuermans J. (1991). Models and controllers of open channel flow for irrigation purposes; A literature study, Delft University of Technology, 46 p.

Schuermans J. (1992). Controller Design for a Regional Downstream Controlled Canal. Delft University of Technology, Laboratory for Measurement and Control, 100 p.

Sen, D. J. and Garg, N. K. (2002). Efficient Algorithm for Gradually Varied Flows in Channel Networks. *J. of Irrig. and Drain. Eng.*, ASCE, pp.351-357.

Sepúlveda C. (2007). Instrumentation, model identification and control of an experimental irrigation canal, PhD, Polytechnic University of Catalonia.

Shand M.J. (1971). Automatic downstream control systems for irrigation canals. PhD. University of California, Berkeley, 159 p.

Soeterboek R. (1990). Predictive control: a unified approach. PhD. Delft University of Technology.

Soler, J. (1996). Integració de les equacions de Saint-Venant: aplicació al reg superficial per taules. Proj. Final de Carrera, Universitat de Lleida, Lleida.

Soler J. (2003). Contribució a l'estudi del control de canals de regadiu per mitjà de mètodes numèrics de programació no lineal. Aplicació al càlcul de les trajectòries de referencia. PhD. UPC, Polytechnic University of Catalonia.

Soler J., Gómez M., Rodellar J. (2013). GoRoSo: Feedforward Control Algorithm for Irrigation Canals Based on Sequential Quadratic Programming. *Journal of Irrigation and Drainage Engineering*, Vol. 139, No. 1, January 2013, pp. 41-54.

Soler J., Bonet E., Gómez M. (2013). Algoritmo CSI: Canal Survey Information para el seguimiento de los caudales extraídos en canales de regadío. La protección contra los riesgos hídricos, III Jornadas de Ingeniería del Agua, Valencia, Spain.

Soler J., Bonet E., Gómez M. (2014). CSI algorithm: Canal Survey Information to estimate the flow rate extracted from irrigation canals. Congress on Industrial & Agricultural Canals, Lleida, Spain.

Strelkoff, T. (1970). Numerical Solution of Saint-Venant equations. *Journal of the Hydraulics Division, ASCE*, Vol.96, No. HY1, pp. 223-252.

Tardieu H. (1988). Automatic dam management and river regulation for irrigation purposes. *Irrigation and Drainage Systems*, n° 2, pp. 53-61.

Theodore W. M. and James T. Cain. (1999). Genetic Algorithms vs. Simulated Annealing: A Comparison of Approaches for Solving the Circuit Partitioning Problem. University of Pittsburgh, Dept. of Electrical Engineering. Available in the Web: <<http://www.pitt.edu/~twmst5/tr-96-101.html>>. Accessed: 25/01/2011.

Tomicic B. (1989). A general optimization module for real-time control of surface water resources, Master of Science, The International Institute for Hydraulic and Environmental Engineering, Delft University, 174 p.

Toudeft A., Kosuth P. & Gallinari P. (1994). A PID Neural Controller for Unstable Delayed Linear Systems, ICANN'94 Proceedings, University of Salerno, Italy.

UNDESA (United Nations Department of Economic and Social Affairs). (2009). World Population Prospects: The 2008 Revision, Highlights. Working Paper No. ESA/P/WP.210. New York, UN.

Van Overloop, P. J., van Schuurmans J, Brouwer R, Burt CM. (2005). Multiple-model optimization of proportional integral controllers on canals. *Irrigation and Drainage Engineering* 131: pp. 190–196.

Van Overloop, P. J. (2006). Model predictive control on open water systems. Ph.D. Delft Univ. of Technology, Delft (Netherlands).

Van Overloop, P.J., Weijs S., Dijkstra S. (2008). Multiple model predictive control on a drainage canal system. *Control Eng Pract* 16(5): pp. 531–540.

Van Overloop, P., Clemmens, A., Strand, R., Wagemaker, R., and Bautista, E. (2010). Real-Time Implementation of Model Predictive Control on Maricopa-Stanfield Irrigation and Drainage District's WM Canal. *J. Irrig. Drain Eng.*, 136(11), pp. 747–756.

Viala Y., Malaterre P-O., Deltour J-L, Sanfilippo F., Sau J. (2004). A multivariable approach for the command of Canal de Provence Aix Nord water supply subsystem", *Proc. Water rights and related water supply issues USCID conference*, Salt Lake City, Utah, USA.

Viessman, W; Lewis, G (1995). *Introduction to Hydrology*. Harper Collins 4ta ed.

Wahlin, B.T. & Bautista, A.M. (2003). Feedforward Control: Volume Compensation versus Model Predictive Control. *USCID-Proceedings International Conference*, Phoenix.

Wahlin B. T. (2004). Performance of Model Predictive Control on ASCE Test Canal. *J. of Irrig. and Drain. Eng.*, ASCE, pp. 227-238.

Wahlin B. T. and Clemmens Albert J. (2006). Automatic Downstream Water-Level Feedback Control of Branching Canal Networks: Simulation Results. *J. of Irrig. and Drain. Eng.*, ASCE, pp. 208-219.

Walker, W.R., Skogerboe, G.V. (1987). *Surface Irrigation. Theory and Practice*. Prentice-Hall, Inc. Englewood Cliffs, New Jersey, UK.

Weyer, E. (2003). LQ control of irrigation channel *Proceedings of the 42nd IEEE Conference on Decision and Control*, Hawaii, USA pp. 750-755.

William H.Press, Saul A.Teukolsky, William T.Vetterling, Brian P.Flanery. (1992). *Numerical Recipes in Fortran 90 Vol 1&2*. Cambridge University Press.

Wylie E.B. (1969). Control of transient free-surface flow. J. of the Hydr. Div., (ASCE), pp. 347-361.

Yevjevich, V. (1975). Unsteady flow in open channels Vol. I. Water Resource Publications, Fort Collins, Colo.

Zagona, E.A. (1992). Model predictive control of automated canals. Ph.D. University of Colorado, Colorado.

Zimbelman D.D. (1987). Planning, operation, rehabilitation and automation of irrigation water delivery systems. Proceedings of a symposium ASCE, Portland, Oregon, USA, 28-30 July 1987, 377 p.