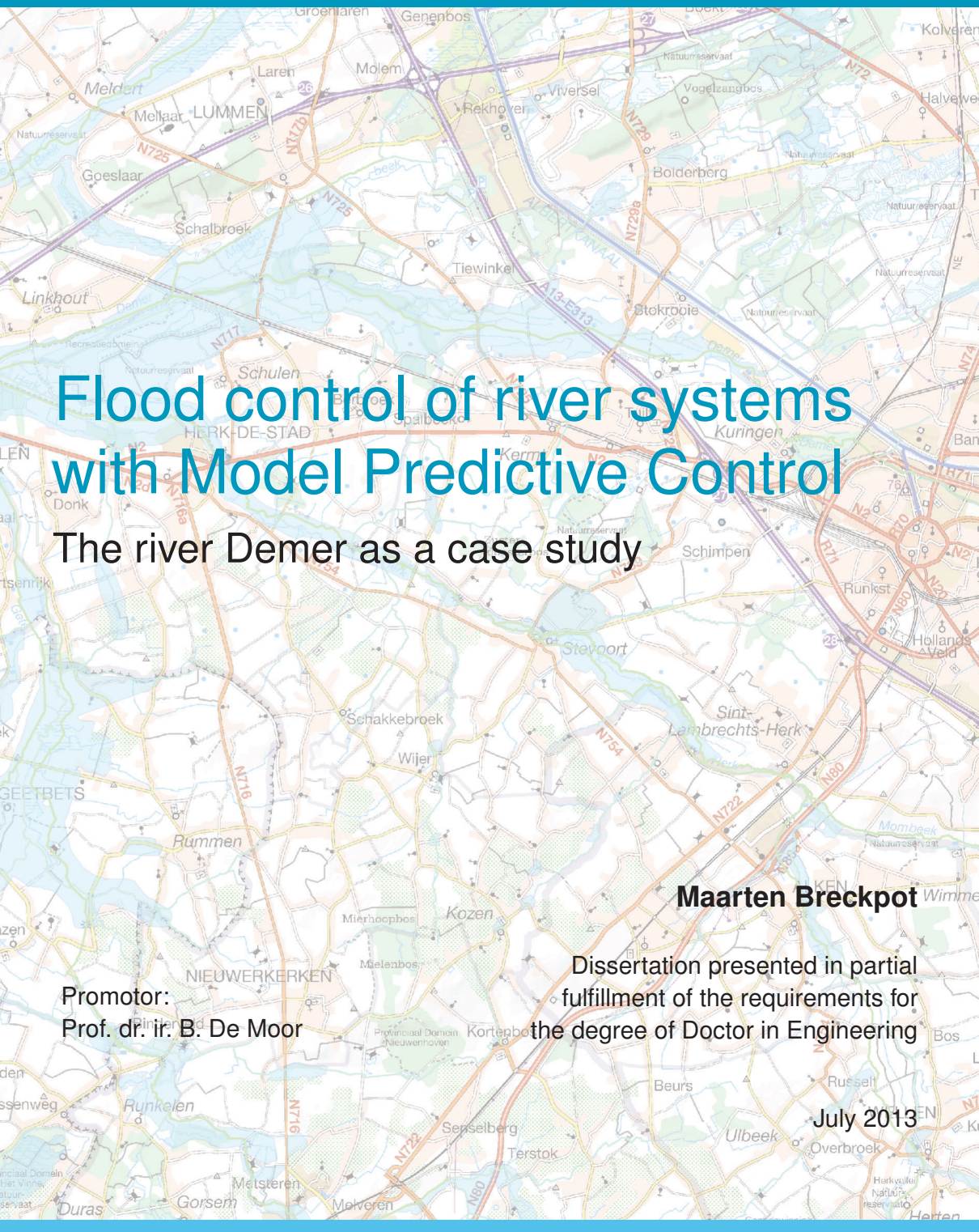
KU LEUVEN

ARENBERG DOCTORAL SCHOOL
FACULTY OF ENGINEERING SCIENCE



Flood control of river systems with Model Predictive Control

The river Demer as a case study

Maarten Breckpot

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

Promotor:
Prof. dr. ir. B. De Moor

July 2013

Cover image:
Part of the flood map of the Demer basin for the flood event of 1998.
Source: Division Operational Water Management of the VMM and the engineering and environmental consulting firm Antea Group.

Flood control of river systems with Model Predictive Control

The river Demer as a case study

Maarten BRECKPOT

Promotor:
Prof. dr. ir. B. De Moor

Members of the
Examination Committee:
Prof. dr. ir. Y. Willems, chair
Prof. dr. ir. P. Willems
Prof. dr. M. Diehl
Prof. dr. ir. B. De Schutter
(Delft University of Technology)
Dr. ir. B. Pluymers
(IPCOS NV)

Dissertation presented in
partial fulfillment of the
requirements for the degree of
Doctor in Engineering

July 2013

© 2013 KU Leuven – Faculty of Engineering Science
Kasteelpark Arenberg 10, B-3001 Heverlee (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

ISBN 978-94-6018-681-3
D/2013/7515/67

Voorwoord

Voor Opa.

De tekst van een proefschrift kan dan wel door velen beschouwd worden als het orgelpunt van vier jaar zwoegen en zweten, in mijn ogen gaat het om veel meer dan enkel dit eindwerk. De afgelegde weg is eens zo belangrijk als het eindpunt van de tocht. Het samenwerken met mensen met verschillende interesses en achtergronden, de vele (en verre) conferenties, de didactische verantwoordelijkheden, de wetenschappelijke successen, maar ook het samenwonen met medelotgenoten, de lange middaglunches met collega's en vrienden, de voetbalmatchen, drie maanden aan de andere kant van de wereld gaan wonen alvorens te gaan samenwonen, trouwen en een eigen huis delen met mijn hartsvriendin, peter worden van een schat van een ventje, iemand moeten laten gaan die je heel dierbaar is. Het waren vier heel leerrijke jaren die samengevat kunnen worden als een rollercoaster van emoties. En zoals u kan lezen was het geen eenzame tocht. Ik heb de voorbije jaren kunnen genieten van de steun van verschillende mensen. En dit is het gepaste moment om deze mensen te bedanken.

In de eerste plaats bedank ik mijn promotor Prof. Bart De Moor. Tijdens het volgen van het vak Computergestuurde Regeltechniek heeft uw enthousiasme en geestdrift me overtuigd om bij u een masterproef te schrijven en vervolgens om een doctoraatsstudie aan te vangen onder uw vleugels. Bart, ik zou je willen bedanken voor je steun de voorbije vier jaar, zowel op inhoudelijk als organisatorisch vlak. Ik kan me geen betere motivator voorstellen, vooral de woorden "Geeft er een lap op!" of "Make it happen." zullen me nog lang bijblijven. Ik waardeer enorm de vrijheid die u mij heeft gegeven in het uitzoeken van waar ik nu juist naar toe wou met dit doctoraat. Of de invalshoek nu theoretisch of eerder praktisch van aard ging zijn, vanaf de eerste dag gaf u me het vertrouwen dat het tot een goed einde ging komen, en met succes zo blijkt. Daarnaast wens ik de verschillende leden van mijn doctoraatsjury te bedanken. Prof. Patrick Willems, ik heb genoten van onze samenwerking de voorbije jaren en ik dank u voor de geleverde data van de Demer. Het

kunnen toepassen van de ontwikkelde technieken op een nauwkeurig model van een bestaande rivier heeft voor een belangrijke toegevoegde waarde gezorgd voor dit doctoraat. Daarnaast wil ik ook Prof. Moritz Diehl bedanken. Uw kennis over predictieve controle en optimalisatie vormden een verrijking voor mijn onderzoek. Ik bewonder uw kunde om verschillende mensen te doen samenwerken en te motiveren. Prof. Bart De schutter en Dr. Bert Pluymers zou ik willen bedanken voor het nalezen van dit proefschrift en voor de waardevolle feedback die ik mocht ontvangen. Jullie opmerken en suggesties hebben tot een verbetering van de tekst geleid. Tenslotte dank ik Prof. Yves Willems om voorzitter te willen zijn van de jury.

Er is ook een woordje van dank op zijn plaats voor de verschillende mensen van de administratie van SISTA (Ilse, Ida, John, ...) die ervoor gezorgd hebben dat de administratieve "last" zeer beperkt is gebleven en ik me volledig op mijn doctoraat heb kunnen concentreren.

Ik bedank de KU Leuven voor het verlenen van een opvangmandaat van een jaar. Daarnaast ben ik ook het FWO erkentelijk voor de financiële ondersteuning tijdens de laatste drie jaar van mijn doctoraat en de extra ondersteuning voor mijn onderzoeksverblijf aan de University of California, Los Angeles.

Daarnaast wil ik mijn collega's bedanken. In de eerste plaats is er een grote dank u wel op zijn plaats voor de steun die ik van Mauricio de voorbije drie jaar heb gekregen. *Su apoyo durante los últimos años fue crucial para alcanzar la meta. Aún cuando usted estaba muy ocupado, siempre encontró tiempo para ayudarme. Mauro, muchas gracias!* Daarnaast bedank ik Kim die samen met mij het doctoraatsparcours heeft aangevat en vaak een luisterend oor was voor mijn geklaag. Ik wil ook Carlos bedanken voor de introductie in de wonderlijke Apple-wereld, de vele niet-Alma lunchen en de aangename sfeer op het eiland samen met Mauricio en Rocco. Ik zou ook Toni willen bedanken die mij het eerste jaar mee op weg gezet heeft. Tenslotte bedank ik al mijn andere collega's (Attila, Siamak, Marco, Philippe, Nico, Maarten, Liesbeth, Fabian, Pieter, Dries, Raghvendra, Villen en vele anderen) voor de gezellige werksfeer.

Hanne en Emerik, onze etentjes tijdens de middagpauze zorgden voor een aangename afwisseling met het bureauwerk en maakten het mogelijk de dagelijkse beslommeringen rond het doctoraat te relativeren.

Ik zou ook mijn Bomma en Bompa, en Oma en Opa willen bedanken. Elk van hen heeft een meer dan positieve invloed gehad op de persoon die ik nu ben geworden. Ieder van hen heeft op zijn of haar manier een steentje bijgedragen aan dit doctoraat. Ik weet hoe graag mijn Opa hier vandaag had willen bijzijn en mijn publieke verdediging wenste bij te wonen, en ik zal nooit zijn oprechte interesse in wetenschap en wiskunde in het algemeen en in mijn onderzoek in het bijzonder vergeten. Daarom draag ik dit doctoraat met veel trots en blijdschap aan hem op.

Ellen, Alexander, Felix, Jeroen en Elien. Jullie hebben alle vijf, de ene al wat langer dan de andere, voor de nodige afleiding gezorgd waardoor ik niet constant aan het doctoraat en dit "boekje" heb liggen denken. Jullie hebben altijd voor mij klaar gestaan om te helpen.

Daarnaast wil ik mijn ouders en schoonouders bedanken. Door jullie steun heb ik mij de voorbije vier jaar onbezorgd kunnen concentreren op dit doctoraat. Jullie aanmoedigingen hebben er mee voor gezorgd dat ik hier vandaag nu sta.

Stephanie, jij hebt meer dan iedereen ervoor gezorgd dat deze tocht alles behalve eenzaam was. Meer dan eens heb je voor een glimlach gezorgd tijdens de mindere momenten en de donkere wolken doen verdwijnen door het zonnetje dat je bent. Je eindeloos geduld en vertrouwen in de goede afloop hebben er eens te meer voor gezorgd dat ik de moed niet heb verloren. Jij was de drijvende kracht waardoor ik de eindmeet heb gehaald. Dank je.

Maarten Breckpot
Leuven, juli 2013

Abstract

This dissertation explores the applicability of Model Predictive Control (MPC) for the purpose of set-point control and flood control of river systems. The first part of this work discusses the modelling aspects of river systems. The dynamics of a single reach can be described with the well known hydrodynamic equations of de Saint-Venant. Combining these hyperbolic Partial Differential Equations (PDEs) for every reach, together with the nonlinear equations modelling the hydraulic structures and the boundary conditions related to junctions, mathematical models can be constructed for a wide range of river systems. However, these models are typically too complex to be used directly in the design of a controller. A new type of approximate model is proposed in this dissertation. A significant reduction in computational complexity with respect to using the full hydrodynamic model while still achieving accurate results can be obtained by approximating the dynamics of every reach with a linear model in combination with the nonlinear gate equations. Model reduction techniques can be used to further decrease the computational complexity.

The main part of this dissertation focuses on the design of the predictive controllers. The key ingredient is to work with the gate discharges as optimization variables instead of the gate openings. A linear approximate model is sufficiently accurate in this configuration and the resulting optimization problem is a Quadratic Programming problem (QP). It is explained how this controller can be used for set-point control and flood control at the same time and how it can recover the used buffer capacity of the reservoirs in an efficient way. Attention is paid to minimize the computation time needed to solve this QP at every time step by decreasing the number of optimization variables and the number of inequality constraints. The use of a Kalman filter as state estimator is also discussed. All closed loop simulations are performed with the full hydrodynamic models.

Besides some academic test examples, a mathematical model of the Demer based on real field data is used to test the performance of the proposed control scheme. It is discussed how the controller can deal with the irregular bed slope and the irregular cross sectional profiles of the river system without having

to rely on nonlinear advanced control techniques. The performance of the predictive controller is tested for the historical rainfall data of the Demer for the flood event of 2002 on the full hydrodynamic model and compared with the control performance of the current controller installed. The proposed predictive control scheme reduces significantly the number and the magnitude of floods, leads to a better set-point tracking and recovers the buffer capacity in a faster way than the current controller.

Beknopte samenvatting

Dit proefschrift onderzoekt of modelgebaseerde voorspellende regeling gebruikt kan worden voor zowel instelpuntcontrole als overstromingsbeheersing van rivieren. Het eerste deel van dit werk behandelt het modelleren van rivieren. De dynamica van een riviertak wordt beschreven door de hydrodynamische vergelijkingen van de Saint-Venant. Dit is een stelsel van twee hyperbolische partiële differentiaalvergelijkingen die het behoud van massa en momentum modelleren. Door deze partiële differentiaalvergelijkingen te gebruiken voor het modelleren van meerdere riviertakken, in combinatie met de niet-lineaire modelvergelijkingen voor hydraulische structuren en de randvoorwaarden gerelateerd aan knooppunten van riviertakken, kunnen mathematische modellen opgesteld worden voor een diverse waaier van riviernetwerken. Zulke modellen zijn typisch te complex om direct gebruikt te worden voor het ontwerpen van een regelaar. In dit proefschrift wordt een nieuw soort van benaderend model geïntroduceerd. Dit model combineert een lineaire benadering van de verschillende partiële differentiaalvergelijkingen met de niet-lineaire modelvergelijkingen voor de hydraulische structuren. Simulatieresultaten tonen aan dat deze benaderende modellen een sterk verminderde rekentijd vergen in vergelijking met de modellen op basis van de Saint-Venant vergelijkingen en toch voldoende nauwkeurig zijn. Er wordt ook uitgelegd dat een extra reductie in de rekentijd verkregen wordt bij het gebruik van modelreductietechnieken toegepast op de gelineariseerde vergelijkingen.

Het belangrijkste deel van dit proefschrift concentreert zich op het ontwerp van de modelgebaseerde voorspellende regelaar. Het kernidee bestaat erin om niet te werken met de standen van de hydraulische structuren als controlevariabelen, maar met het debiet vloeiend over deze structuren. In dit geval kan de dynamica van de te controleren rivier voldoende nauwkeurig benaderd worden met een lineair model. Hierdoor is het optimaliseringsprobleem een kwadratisch programmeerprobleem. Dit programmeerprobleem wordt zodanig geformuleerd dat de regelaar focust op overstromingsbeheersing tijdens kritieke momenten en op instelpuntcontrole op de andere tijdstippen. Hetzelfde optimaliseringsprobleem kan gebruikt worden om de gebruikte buffercapaciteit

van reservoirs tijdens periodes van hevige regenval op een efficiënte manier te recupereren. De regelaar kan gebruikt worden in combinatie met een Kalman filter voor het schatten van de niet-gemeten waterstanden en debieten van de verschillende riviertakken. Er wordt ook uitgelegd hoe de tijd nodig voor het oplossen van het optimaliseringsprobleem gereduceerd kan worden door het aantal optimalisatievariabelen en het aantal ongelijkheidsbeperkingen te verminderen.

Naast enkele academische testvoorbeelden wordt een wiskundig model van de Demer op basis van opgemeten data gebruikt om de regelkracht van de regelaar te testen. Enkele aanpassingen aan de regelaar worden geformuleerd zodat de regelaar overweg kan met de onregelmatigheden in de helling van de rivierbedding en in de profielen van de dwarssecties van de rivier zonder dat het nodig is om met een niet-lineaire regelstrategie te werken. Op basis van de historische neerslagdata tijdens de overstroming van de Demer in 2002 wordt een vergelijking gemaakt tussen de regelkracht van de ontwikkelde voorspellende regelaar en van de huidige regelaar geïnstalleerd langs de Demer. De voorspellende regelaar resulteert in een kleiner aantal overstromingen met een lagere maximale hoogte, een betere instelpuntregeling en een snellere recuperatie van de gebruikte buffercapaciteit van een reservoir.

Acronyms

D-MPC	Distributed Model Predictive Control
H-MPC	Hierarchical Model Predictive Control
HIC	Hydrologic Information Center
L-MPC	Model Predictive Control based on a linear approximate model
LN-model	Linear-Nonlinear model
LN-MPC	Model Predictive Control based on the Linear-Nonlinear model
LQR	Linear Quadratic Regulator
M-MPC	Multiple Model Predictive Control
MHC	Moving Horizon Control
MHE	Moving Horizon Estimation
MPC	Model Predictive Control or Model Predictive Controller depending on the context
NLP	Nonlinear Programming problem
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PI-controller	Proportional Integral controller
QP	Quadratic Programming problem
RHC	Receding Horizon Control
RMSE	Root-Mean-Square Error
SSE	Sum of Squared Error
SVD	Singular Value Decomposition

TAW	reference height for altimetry in Belgium (*Tweede Algemene Waterpassing*)
VMM	Flemish Environment Agency *(Vlaamse Milieu Maatschappij)*

Notation and list of symbols

Notation

$\mathbf{x} \in \mathbb{R}^n$	a bold lower case symbol denotes a vector of variables
$\mathbf{X} \in \mathbb{R}^{n \times m}$	a bold upper case symbol denotes a matrix of variables
$\mathbf{x}^{\mathrm{T}}$	the transpose of the vector $\mathbf{x}$
$\mathbf{X}^{\mathrm{T}}$	the transpose of the matrix $\mathbf{X}$
x_i	the ith component of the vector $\mathbf{x}$
$[\mathbf{x};\mathbf{y}]$, $\mathbf{x},\mathbf{y} \in \mathbb{R}^n$	stacked vectors: $\left[\mathbf{x}^{\mathrm{T}}, \mathbf{y}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{2n}$
$\langle \mathbf{x}, \mathbf{y} \rangle$, $\mathbf{x},\mathbf{y} \in \mathbb{R}^n$	the Euclidean inner product: $\mathbf{y}^{\mathrm{T}}\mathbf{x}$
$\left\|\mathbf{x}\right\|_2$	l_2-norm of the vector $\mathbf{x}$: $\sqrt{\mathbf{x}^{\mathrm{T}}\mathbf{x}}$
$\left\|\mathbf{x}\right\|_{\mathbf{Q}}$	weighted l_2-norm of the vector $\mathbf{x}$: $\sqrt{\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x}}$
$\mathbf{X} \in \mathbb{R}^{n \times n} \succ 0$	a positive definite matrix: $\forall \mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\} : \mathbf{x}^{\mathrm{T}}\mathbf{X}\mathbf{x} > 0$
$\mathbf{X} \in \mathbb{R}^{n \times n} \succeq 0$	a positive semidefinite matrix: $\forall \mathbf{x} \in \mathbb{R}^n : \mathbf{x}^{\mathrm{T}}\mathbf{X}\mathbf{x} \geq 0$
$\mathbf{I}_n$	identity matrix of size $n \times n$
$\mathbf{I}_{n,m}$	a matrix of size $n \times m$ with ones on the main diagonal
$\mathbf{0}_{n,m}$	a matrix of zeros of size $n \times m$
$\mathbf{0}_n$	a vector of zeros of size n
$\mathbf{1}_n$	a vector of ones of size n
$\mathbb{R}^n$	the set of real n-tuples

General variables

g	the gravitational acceleration constant	$\left[\mathrm{m} \cdot \mathrm{s}^{-2}\right]$
t	the time variable	[s]
z	the spatial variable	[m]

Variables related to control theory

$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{F}$	the state space matrices of the discrete time model: $\begin{cases}\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{F}\mathbf{d}(k) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k)\end{cases}$
$\mathbf{d} \in \mathbb{R}^{n_d}$	the n_d disturbance variables of the system
$\mathbf{u} \in \mathbb{R}^{n_u}$	the n_u input variables of the system
$\mathbf{x} \in \mathbb{R}^{n_x}$	the n_x state variables of the system
$\mathbf{y} \in \mathbb{R}^{n_y}$	the n_y output variables of the system
$\mathbf{L}$	the Kalman gain matrix
$\hat{\mathbf{x}}$	the estimate of the state $\mathbf{x}$

Channel specific variables

A	the cross-sectional flow area	$[\mathrm{m}^2]$
B	the bottom width of a trapezoidal channel	$[\mathrm{m}]$
h	the water level	$[\mathrm{m}]$
L	the channel length	$[\mathrm{m}]$
n_{mann}	the Manning coefficient	$[\mathrm{s} \cdot \mathrm{m}^{-1/3}]$
n_{eqmann}	the equivalent Manning coefficient	$[\mathrm{s} \cdot \mathrm{m}^{-1/3}]$
P	the wetted perimeter of a cross section	$[\mathrm{m}]$
Q	the water discharge	$[\mathrm{m}^3 \cdot \mathrm{s}^{-1}]$
R	the hydraulic radius of a cross section	$[\mathrm{m}]$
S_0	the bed slope	$[-]$
S_1, S_2	the side slopes of a trapezoidal channel	$[-]$
S_{f}	the friction slope	$[-]$
$.^{(i)}$	indicates the channel, gate or reservoir the variable or parameter belongs to	
n_h	total number of water levels of the entire river system	
$n_h^{(i)}$	total number of water levels of reach i	
n_Q	total number of discharges of the entire river system	
$n_Q^{(i)}$	total number of discharges of reach i	
n_c	the number of channels or reaches in a river system	

Contents

Voorwoord i

Abstract v

Beknopte samenvatting vii

Acronyms ix

Notation and list of symbols xi

Contents xiii

1 General Introduction 1

1.1 Introduction and motivation 1

1.2 Objectives 5

1.3 Chapter by chapter overview 6

1.4 Contributions 9

1.5 Comparison with previous work related to flood control of the Demer 12

2 Predictive Control and Model Reduction 15

2.1 Introduction 15

2.2 Linear Quadratic Regulator 15

2.3 Model Predictive Control 17
2.3.1 Linear Model Predictive Control 18
2.4 Kalman Filter 21
2.5 Proper Orthogonal Decomposition 22
2.5.1 POD basis vectors generation 23
2.5.2 Model reduction via Galerkin projection 25
2.6 Conclusions 27

3 Modelling of River Systems 29

3.1 Introduction 29
3.2 Modelling a single reach 30
3.2.1 The Saint-Venant equations 30
3.2.2 Modelling the friction slope S_f 31
3.2.3 Limitations of the Saint-Venant equations 34
3.3 Boundary conditions 35
3.3.1 Given discharge or water level 35
3.3.2 Rating curves 36
3.3.3 Hydraulic structures 36
3.3.4 Junctions 42
3.4 Modelling of reservoirs 43
3.5 Numerical solution method 44
3.5.1 Preissmann scheme 45
3.5.2 Preissmann scheme applied to staggered grid structure in combination with upwinding approach 47
3.5.3 Illustrative example: dam break problem 53
3.6 Linear-Nonlinear model & other approximate models 56
3.6.1 Literature review 56
3.6.2 Linear-Nonlinear model 59

3.7 Conclusions 74

4 Model Predictive Control Used for Set-point Control and Flood Control 77

4.1 Introduction 77

4.2 Why using Model Predictive Control? 79

4.2.1 Literature review 79

4.2.2 Advantages of Model Predictive Control 81

4.3 Model Predictive Control applied to a single reach 83

4.3.1 Mathematical model of a single reach 83

4.3.2 Approximate model 84

4.3.3 Formulation of the optimization problem 85

4.3.4 Simulation results 88

4.3.5 Simulation results with state estimator 96

4.4 Model Predictive Control in combination with gate equations . 103

4.4.1 Mathematical model of the test system 104

4.4.2 Approximate model and the selection of control variables 104

4.4.3 Formulation of the optimization problem 106

4.4.4 Simulation results 113

4.5 Model Predictive Control applied to a river system with a reservoir 122

4.5.1 The test example 123

4.5.2 Approximate model 125

4.5.3 Formulation of the optimization problem 125

4.5.4 Simulation results 128

4.6 Implementation aspects & speed-up of the controller 140

4.6.1 Reducing the number of optimization variables 142

4.6.2 Reducing the number of inequality constraints 145

4.6.3 Application of the speeding-up techniques to a river system with a small reservoir 146
4.7 Conclusions 152

5 Model Predictive Control Applied to the Demer 157
5.1 Introduction 157
5.2 The Demer 159
5.2.1 Study area of the Demer and control objectives 159
5.2.2 Mathematical model 161
5.3 Three-position controller 165
5.4 Design of Model Predictive Controller 165
5.4.1 Approximate model 166
5.4.2 Formulation of the optimization problem 169
5.4.3 State estimator 174
5.5 Results 175
5.6 Conclusions 185

6 General Conclusions 189
6.1 Concluding remarks 189
6.2 Future research 191

A Modelling of Hydraulic Structures 195
A.1 Gated weir 195
A.2 Spill 198

B Test examples 201
B.1 Single reach 201
B.1.1 Steady state problem with analytic solution 201
B.1.2 Mass conservation test 205

B.1.3 Linear increase of downstream discharge and constant upstream water level 205

B.1.4 Changing upstream and downstream discharges 207

B.2 Multiple reaches 207

Bibliography **215**

Curriculum Vitae **233**

Publications by the author **235**

1

General Introduction

1.1 Introduction and motivation

From the earliest times floods have caused worldwide serious and vast problems. Facts and figures show that floods are the most common natural disaster. The number of heavy floods is increasing since the seventies of the 20th century on a worldwide scale, in Europe and in Belgium (see Table 1.1). More than 100 large floods occurred between 1998 and 2004 resulting in an insured economic loss of more than 25 billion euro. More than 700 people lost their live and half a million lost their homes. Floods are also a grave problem in Belgium. Recently very large floods occurred in the winters of 1993-1994 and 1994-1995, August 1996, September 1998, December 1999, February 2002, December/January 2003, July 2005, July 2007 and November 2010. One example of a river in Belgium with a history of large floods is the Demer. The Demer is located in Flanders and

Table 1.1: Evolution of the number of heavy floods worldwide, in Europe and in Belgium between 1970 and 2009. Source: [1].

	1970-1979	1980-1989	1990-1999	2000-2009
worldwide	263	526	780	1729
Europe	23	38	94	239
Belgium	1	2	4	6

is part of the catchment area of the Scheldt. The river is 85 km long and originates in Ketsingen. It connects the cities Hasselt, Zichem and Aarschot and it flows at Werchter into the river Dijle. The Demer is fed by water coming from its many tributaries and from groundwater. Therefore, the Demer shows high discharge peaks during heavy rainfall periods. Its average flow varies from 6.4 $m^3 \cdot s^{-1}$ during summer and periods of drought to 33.9 $m^3 \cdot s^{-1}$ during winter or periods of heavy rainfall. Because of these large peak discharges, already in the 17th century people started taking measures to prevent the river from flooding. Dikes were constructed, the river was made wider and deepened, and meanders were removed. This process of eliminating river bends continued until 1980. However, these large efforts could not prevent the river from flooding during periods of heavy rainfall, e.g. in 1905, 1926, 1965, 1966, 1993-1994, 1995, 1998, 2002 and 2010. The flooding of 1998 resulted in a financial damage of more than 16 million euro. Fig. 1.1 visualizes the total flooded area: the floods are not limited to one particular location but occur along the entire river system.

Up until the flood event of 1998, the adjustments made to the Demer were considered to have a positive effect for flood prevention. However, people started to realize that these adjustments actually increased the impact of the floods. Constructing dikes and straightening the river resolve the problems locally, but increase the risk at places further downstream the river. The flooding problem is only relocated to other places along the river. For the Demer this meant that the adjustments had the effect that the total damage increased because more cities and villages are at the downstream side of the river than at the upstream side. This awareness has completely shifted the vision on how to deal with the flooding problem. It is impossible to prevent a river from flooding at all times: floods are a natural element of rivers and will always take place. Therefore the focus shifted from flood prevention to flood control or flood management. There is a need of properly managing the floods to limit the total damage. Floods are less dramatic if they occur in places where they can cause only a limited amount of damage. The flooding of a grassland is not a disaster compared to the flooding of a city. Solutions in line with this new vision on flood control or flood management are the following ones [27, 167]:

- the preservation and if possible the restoration of the natural flood areas of rivers,
- the construction of artificial reservoirs
- and a computer controlled management of river basins.

The start of this approach for a more intelligent flood regulation in Flanders was given by the Flemish Environment Agency (VMM, *Vlaamse Milieu*

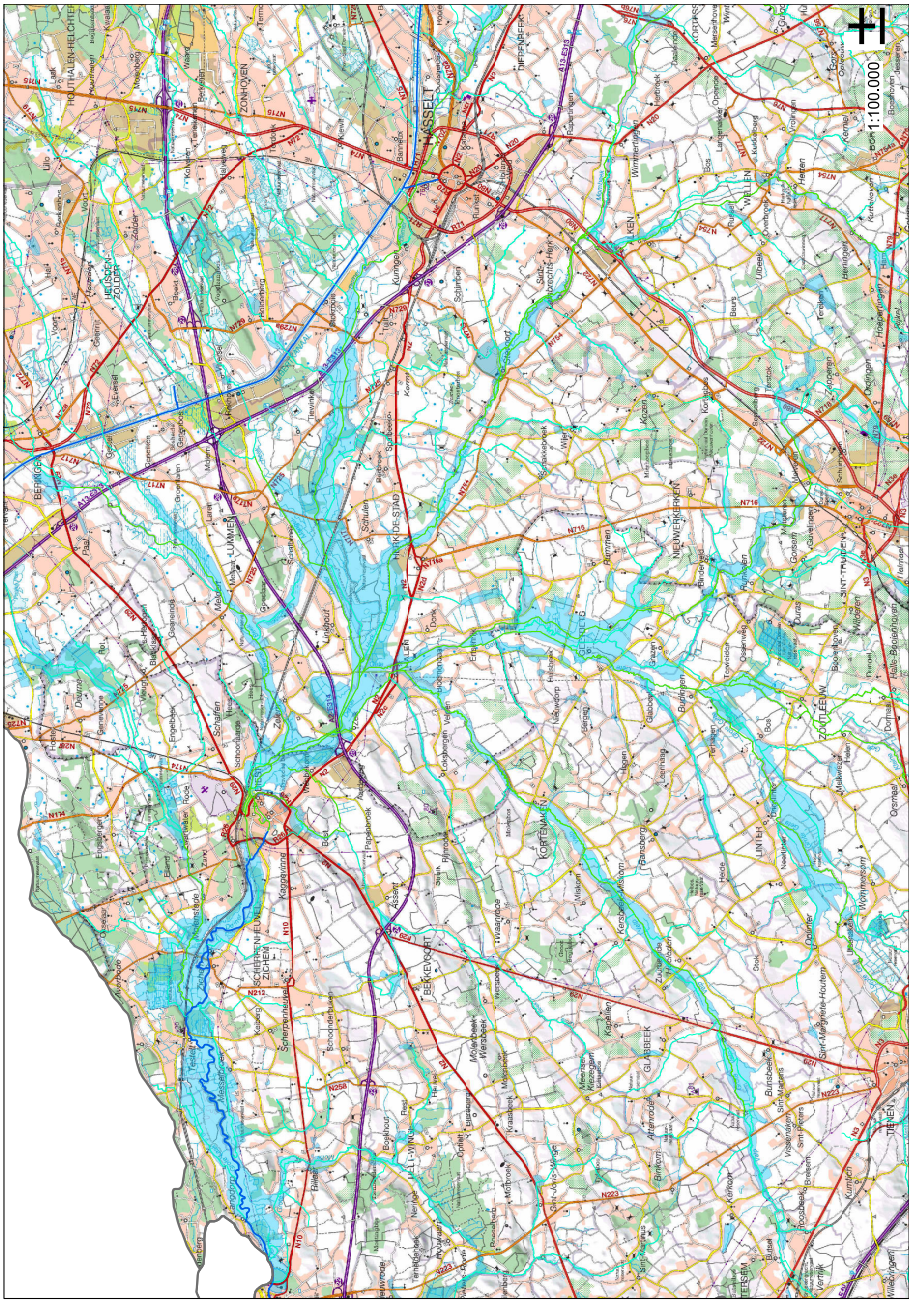


Figure 1.1: Flooded area of the Demer for the flood event of 1998. Source: Division Operational Water Management of the VMM and the engineering and environmental consulting firm Antea Group.

Maatschappij) and the Hydrologic Information Center (HIC). Three water basins were installed along the Demer with a total buffer capacity of more than 18 million m^3 and hydraulic structures were constructed to control the flow at different places along the river. Also a computer model of the Demer was made which is currently used for flood prediction purposes [77]. The local water administration implemented an advanced three-position controller for controlling the hydraulic structures. This controller is in its essence a set of logical rules based on the many years of experience of operators in controlling these hydraulic structures. This controller has already reduced the flood risk, but it has its limitations. New simulations performed by the local water administration on historical rainfall data have demonstrated that the flooding damage during the last decades could have been limited if a more advanced control strategy would have been used.

On behalf of the Division Operational Water Management of the VMM, the research group SISTA-SCD of the Department of Electrical Engineering of the KU Leuven and the Hydraulics Laboratory of the Department of Civil Engineering of the KU Leuven have conducted research to evaluate the applicability of Model Predictive Control (MPC) as control strategy for flood control of the Demer. This research resulted in the doctoral thesis of T. Barjas Blanco [18]. Based on a conceptual model of the Demer [44, 189], it demonstrates that MPC has some specific advantages compared to the three-position controller concerning the application of flood control [18–20, 37, 38]. However there are some limitations related to the type of model used for modelling the Demer and to the formulation of the predictive controller. The conceptual model has already proven its strength, but some remarks can be made:

- A conceptual model describes the dynamics of the water levels and discharges of a river system at only a limited number of locations in order to keep the computational burden low. E.g. the water levels of a single reach are often modelled with one point. However for reaches with irregular flood levels, this is not the optimal choice. It is not certain that if the modelled water level is kept below its flood level, the entire water profile of the reach is below its flood level. A conceptual model is the right choice for testing a new control strategy for the first time, but before it can be applied in reality, it should be tested on a full hydrodynamical model of the river system modelling the water levels and discharges on a very fine grid.

- The construction, calibration and validation of a conceptual model is far from straightforward. It involves a very time-consuming process and expert knowledge is required for determining the optimal model structure

and the values of the many parameters. This process has to be redone every time one wants to use MPC to control another river system.

Also some comments can be made related to the predictive controller presented in these works:

- The control scheme needs to find multiple linear approximate models based on the conceptual model at every time step. Because of the small structure of this conceptual model, the impact of this operation on the total computation time was limited. However if this technique is applied to a much more detailed model, the computational burden would increase significantly and the total time needed by the controller can become too large to be applicable in practice.
- During periods of heavy rainfall, multiple optimization problems need to be solved before new control actions can be found. This means that the controller has the highest computation time during periods when the situation is critical.
- The controller implemented in the thesis was only based and tested on the conceptual model of the Demer. Due to the specific form and equations used to construct this conceptual model, it is very difficult to make conclusions on the applicability of MPC for flood control of river systems in general.

1.2 Objectives

The main objective of this dissertation is to test whether **Model Predictive Control** can be used for **set-point control** and **flood control** of **river systems** in **general**. Taking the remarks at the end of the previous section into account, this objective can be translated into the following tasks:

- Test an alternative approach to the use of conceptual models by setting up a general modelling framework of river systems that can be used for different types of river networks. The resulting model should describe the dynamics of the water levels and discharges along the entire reach on a fine grid.
- Based on this alternative modelling approach, find an approximate river model that can be used inside the predictive controller. This river model should approximate the dynamics of every reach along the entire profile.

- Adapt the general principles of Model Predictive Control in order to comply with the different objectives of controlling river systems:
  - set-point control,
  - in combination with flood control,
  - and fast buffer capacity recovery of reservoirs.
- Keep the total computation time needed by the advanced controller as small as possible at all times without reducing its control performance related to set-point control, flood control and buffer capacity recovery:
  - find an approach such that the same linear model can be used at every time step
  - and prevent the necessity of solving multiple optimization problems before new control actions can be found.

1.3 Chapter by chapter overview

This dissertation consists of six chapters in total. Fig. 1.2 gives an overview of the different chapters and the connections between them. A summary of each of the chapters is given as follows:

- **Chapter 2:** This chapter introduces the fundamental concepts of some well-known techniques that will be used in the subsequent chapters. It first introduces a classical control strategy, namely the Linear Quadratic Regulator theory. Afterwards it discusses the main principles of Model Predictive Control. It also explains the basics of Kalman filtering and it introduces how Proper Orthogonal Decomposition (POD) and Galerkin projection can be used for deriving reduced order models.
- **Chapter 3:** The chapter presents a general framework for constructing mathematical models of river systems. It first introduces the full hydrodynamic equations of de Saint-Venant used to model the dynamics of a single reach. Afterwards it explains how to model the dynamics of river systems consisting of multiple reaches. Different types of connections, like hydraulic structures and junctions, are discussed and it is explained how these connections result in boundary conditions for each of the reaches. Two approaches for modelling a reservoir are introduced. The chapter explains into detail how a numerical solution method can be constructed for simulating these complex river systems. Some examples are given to show the accuracy of the simulator. The last part of this chapter focusses on finding appropriate approximate models of the

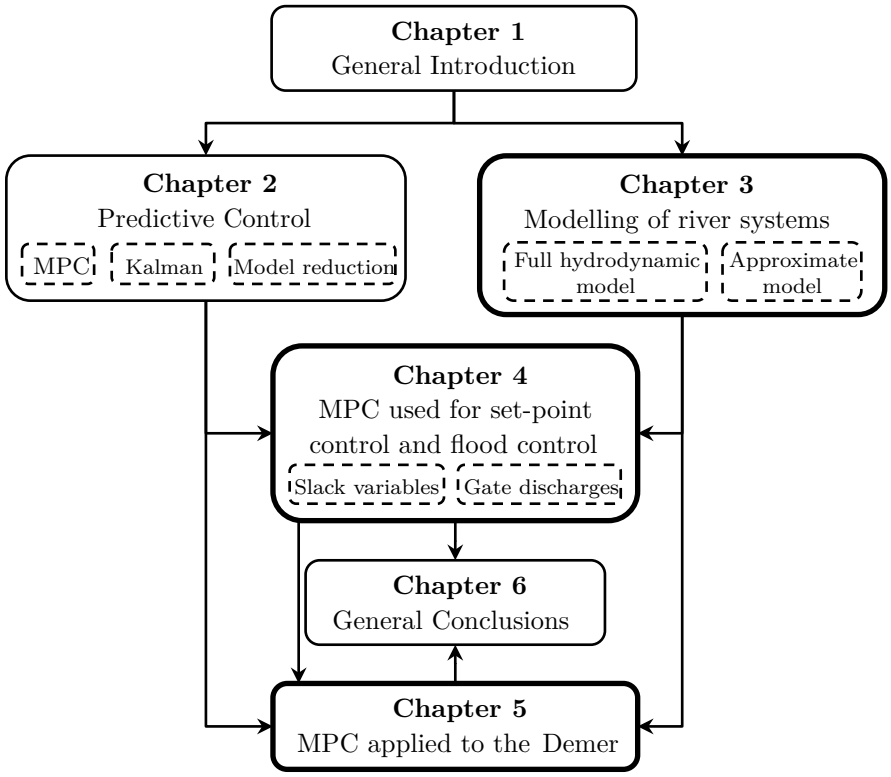


Figure 1.2: Overview and connection between the different chapters in this dissertation.

nonlinear river dynamics. The proposed approximate model combines a linear approximation of the dynamics of every reach with the nonlinear equations modelling the discharges over the hydraulic structures. The performance of this approximate model is compared with the one of the original nonlinear model. It is also explained how a reduction in the computational load can be achieved with this approximate model by applying POD and Galerkin projection [34].

- **Chapter 4:** This chapter addresses the control of some academic test examples with Model Predictive Control. The chapter first explains why Model Predictive Control is suitable for set-point control and flood control of river systems in contrast to other control techniques. In the next three sections, the design of a predictive controller for different test examples is discussed in detail. The test system in every subsequent section is more complex than the example in the previous one and it is explained how the

design of the predictive controller can be adapted to deal with this extra complexity. The first test example is a single reach. After introducing the linear approximate process model used by the controller, the optimization problem solved by the controller at every time step is formulated. It is explained how slack variables ensure the feasibility of the optimization problem at every time step and how the control objectives can be translated into the different parameters of the optimization problem. The performance of the proposed controller with respect to set-point control, disturbance rejection and flood control is assessed, together with the performance of a Linear Quadratic Regulator [32]. Also the influence of a Kalman filter on the control performance is investigated. The next test example contains a hydraulic structure. It is explained that by optimizing over the gate discharge instead of the gate position, a linear approximate model is accurate enough. Again the optimization problem is formulated and discussed in detail. Extra elements are the translation of the physical limits on the gate positions to limits on the gate discharges and a method for preventing possible uncontrollability of the gates. Simulation results show the performance of the predictive controller without and with a Kalman filter as state estimator [33]. The last example is a complex river network consisting of multiple reaches, hydraulic structures, junctions and a reservoir. It is explained how the optimization problem can be formulated to use the buffer capacity of the reservoir in an optimal way and how the used buffer capacity can be recovered after a heavy rainfall period [36]. Again the influence of the Kalman filter on the control performance is checked. The last section discusses the implementation of the optimization problem. It is explained how the time needed for solving the optimization problem can be decreased by reducing the number of optimization variables and the number of inequality constraints. Two methods are discussed for both approaches. A first approach for reducing the number of optimization variables uses POD and Galerkin projection to reduce the number of states of the approximate model. This results also in a smaller number of equality constraints. Another approach is to use a condensed implementation: the water levels and discharges are removed from the optimization problem by writing them as a function of the control actions based on the linear approximate model. The number of inequality constraints can be reduced by using a greedy selection algorithm in combination with POD and Galerkin projection, exploiting the similarity between the coefficients of consecutive water level constraints. Another approach is to impose the water level constraints at only a limited number of locations along every reach.

- **Chapter 5:** The chapter presents the application of the predictive controller introduced in the previous chapter to a full hydrodynamic model of the Demer build up from real field data. After introducing the

study area of the Demer and giving its mathematical model formulation, the disadvantages of the advanced three-position controller currently used for regulating the Demer, are discussed. The next part explains the design of the predictive controller and how it is influenced by the irregularity of the river bed profile and the irregularity of the cross sectional profiles. The irregular cross sectional profiles are approximated with a trapezoidal shape in order to increase the range for which the linear approximate model is accurate enough. A prediction step with a nonlinear model with a low computational complexity is used to estimate the lower and upper limits on the gate discharges. These nonlinear predictions are compared with the predictions made with the linear approximate model. A simple model update of the linear approximate model is performed based on the difference between both predictions. It is explained how the number of inequality constraints can be kept limited by working with only the most critical flood levels. An adapted version of the Kalman filter is used as state estimator. Based on only three water level measurements, the entire state of the system is estimated. The performance of the predictive controller is compared with the one of the three-position controller when applied to a full hydrodynamic model of the Demer with the historical rainfall data of the flood event of 2002. MPC succeeds in reducing the number and the magnitude of the floods, it recovers the buffer capacity almost two days earlier and it delivers a better set-point regulation [35].

- **Chapter 6:** This chapter presents the general conclusions of this thesis and discusses some future research subjects.

The text of this dissertation has been written in such a way that it is possible to reproduce the results obtained in this work. Chapter 3 explains in detail how a numerical scheme for simulating river systems can be implemented by discussing the discretization approach, the resulting discretized equations and how numerical stability and accuracy can be achieved. Chapter 4 explains in detail every aspect of the controller such that, it is possible to implement the same type of controllers. The parameters of the river systems as well as the values of the different tuning parameters of the controllers are always given in order to make it possible to reproduce the results.

1.4 Contributions

The main contributions of this dissertation are the following ones:

- **Linear-Nonlinear model.** The first contribution is the proposed hybrid model in Section 3.6.2.2. We have shown that the nonlinear river

dynamics can be accurately approximated with a linear state space model describing the dynamics of every reach with the gate discharges as input variables, in combination with the nonlinear gate equations. The use of the linear state space model reduces the computation time drastically. This reduction can even be improved by reducing the model order of the linear state space model with POD in combination with Galerkin projection.

[34] Breckpot, M., Agudelo, O. M., and De Moor, B. Modelling of a river system with multiple reaches. In *Proc. of the 16th IFAC Symposium on System Identification* (Brussels, Belgium, July 2012), pp. 1454-1459.

- **Gate discharges as optimization variables.** One of the major contributions is the idea to use the hybrid model as approximate model for the dynamics of a river system in the controller and to leave the gate equations out of the optimization problem (Section 4.4.2). This means that we are optimizing over the gate discharges instead of the gate positions. Once the optimal gate discharges are found, a conversion step is needed to translate the gate discharges to their corresponding gate positions. This conversion step can easily be performed based on the full nonlinear gate equations. This special choice of optimization variables has as implication that we need to derive only one linear approximate model for the river system and we do not need to work with a nonlinear advanced process controller. Therefore the resulting optimization problem will be a Quadratic Programming problem (QP) for which efficient solvers exist.

- **Formulation of the optimization problem.** Along Chapter 4 we propose an optimization problem that effectively can be used for set-point control and flood control of river systems:

  - By imposing the flood levels as upper limits on the water levels, the controller will try to avoid any flooding (**flood control**). By minimizing over the deviation between the water levels and their set-points, the controller will focus on **set-point control**. The upper limits on the water levels are imposed as soft constraints in order to keep the QP feasible at all times: only one QP needs to be solved at every sampling time.
  - In Section 4.4.3.1 we explain how the (known) lower and upper limits on the gate positions and their maximal rate of change can be translated to time-varying lower and upper limits on the gate discharges along the prediction horizon. For river systems with a smooth channel bed profile and trapezoidal cross sectional profiles, this prediction step can be performed with a linear model. Therefore, the total computation time of this step is limited.

  - One problem with flood control is the possibility of uncontrollability of hydraulic structures. A gate can operate in different modes. The discharge over a gate is independent of the gate position for some of these modes. This means that the predictive controller can not use this structure any more. Section 4.4.3.2 explains how this uncontrollability can easily be prevented by keeping the gates always in their controllable region or at least on their boundary.
  - In Section 4.5.3.2 we explain how the used **buffer capacity** of a reservoir can be **recovered** as fast as possible. We only need to set the set-point of the water levels of the channels downstream of the reservoir below the set-point of the reservoir, and increase the weights related to the deviation of the set-point for these channels inside the objective function.

[32] Breckpot, M., Agudelo, O. M., and De Moor, B. Control of a single reach with model predictive control. In *River Flow 2012* (Costa Rica, September 2012), R. E. Murillo Muñoz, Ed., Proc. of the International Conference on Fluvial Hydraulics, CRC Press, Taylor & Francis Group, pp. 1021-1028.

[33] Breckpot, M., Agudelo, O. M., and De Moor, B. Model Predictive Control of a river system with two reaches. In *Proc. of the 51st IEEE Conference on Decision and Control* (Maui, Hawaii, USA, December 2012), pp. 4549-4554.

[36] Breckpot, M., Agudelo, O. M., and De Moor, B. Flood control with Model Predictive Control for river systems with water reservoirs. *Journal of Irrigation and Drainage Engineering 139*, 7 (July 2013), 532-541.

- **State estimator.** For every test example in Chapter 4, we have shown that a Kalman filter can be used for estimating the unknown states from a very limited number of measurements.

- **Speed-up of the controller.** Section 4.6 discusses how the time needed to solve the QP at every time step can be reduced by combining a condensed implementation, reducing the number of optimization variables and eliminating the equality constraints implementing the linear approximate model, with a method for reducing the number of inequality constraints. This last reduction can be achieved by imposing the upper limits on the water levels at only a limited number of locations along the channels, or by exploiting the fact that the coefficients of consecutive constraints are similar when used in combination with POD and Galerkin projection. The condensed implementation in combination with either of the two techniques for reducing the number of inequality constraints, results in a speed-up with a factor of 60 for the used optimization solver.

- **MPC applied to the Demer.** In Chapter 5 we show that only minor modifications of the controllers designed in Chapter 4 are needed such that the controller can handle the irregularity in the field data of the Demer. The only modifications are the use of a simple nonlinear model for finding the lower and upper limits on the gate discharges and for finding the state estimates with the Kalman filter, and a straightforward model updating technique. The simulations performed for the rainfall data of the flood event of the Demer in 2002 indicate that MPC outperforms the advanced three-position controller.

  [35] Breckpot, M., Agudelo, O. M., Meert, P., Willems, P., and De Moor, B. Flood control of the Demer by using Model Predictive Control. Internal Report 13-24, ESAT-SISTA, KU Leuven (Leuven, Belgium). *Submitted for publication in Control Engineering Practice* (2013).

At the end we have demonstrated that MPC can be used for set-point control and flood control of different complex river systems. The main idea is to optimize over the gate discharges such that the linear approximate model has to be derived only once. The resulting optimization problem is a QP that is at all times feasible.

1.5 Comparison with previous work related to flood control of the Demer

As mentioned at the end of Section 1.1 there exists previous work performed at the KU Leuven with the focus on flood control of the Demer [18–20,37,38,189]. The work performed in this dissertation differs from these studies in multiple ways:

- **Process model.** The previous works used a conceptual model with a limited number of nodes for simulating the Demer. In this dissertation a full hydrodynamic model of the Demer is used for testing the control performance of the developed predictive controller. This mathematical model makes use of the full Saint-Venant equations where field data of the river is used regarding the (irregular) cross sectional profiles and the varying slope of the river bedding at a large number of grid points. Also different roughness coefficients related to the different parts of each irregular cross sectional profiles are used to model the frictional bed resistance.

- **Control strategy.** Regarding the predictive controller, the main difference lies in the **choice** of the **control variables** used inside the optimization problem. In the previous works the authors worked with the gate positions as optimization variables resulting in the requirement to use nonlinear model based predictive control schemes in order to achieve a good control performance. At every time step a new sequence of time-varying linear state space models needs to be derived over the prediction window based on the sequence of optimal control actions calculated at the previous time step. Because of the limited accuracy of the linear approximation of the gate equations, a trust region is used inside the optimization problem in order to limit the difference between the new optimal sequence of control actions and the previous one. Furthermore a line search is used as post-processing step to increase the effectiveness of the computed control actions. All these operations are not needed in this work by working with the gate discharges as optimization variables. Eliminating the gate equations out of the optimization problem has as effect that the dynamics of the river systems can be accurately approximated with one linear state space model: a linear predictive control scheme can be used. There is no need to derive new linear state space models, to work with trust regions or to perform a line search as post-processing step.

  Another difference lies in how the upper limits on the water levels are imposed. The previous works impose these inequality constraints as hard constraints, which can lead to infeasibilities during periods of excessive rainfall. Therefore a constraint relaxation strategy is implemented that iteratively drops inequality constraints until the optimization problem becomes feasible. This approach can result in solving multiple optimization problems. This disadvantage is prevented in this dissertation by working with slack variables and imposing the inequality constraints as soft constraints. Following this approach results in an optimization problem that is always feasible.

- **State estimator.** The previous works as well as this work implement a state estimator taking into account that in practice only a limited number of water levels are measured. Because the previous works used the gate positions as control variables, Moving Horizon Estimation (MHE) was used as estimation technique. This technique results in solving an optimization problem at every time step taking the measurements of the last sampling times into account as well. By working with the gate discharges as control variables in this dissertation, MHE is not required to obtain accurate estimates of the unknown states. It is shown that a Kalman filter is accurate enough requiring only a limited number of mathematical operations at every time step.

2

Predictive Control and Model Reduction

2.1 Introduction

This chapter introduces Model Predictive Control (MPC), Linear Quadratic Regulator theory (LQR), Kalman filtering and model reduction based on Proper Orthogonal Decomposition (POD) and Galerkin projection. These techniques will be used throughout this thesis in the next chapters.

The chapter has the following structure. LQR is discussed in Section 2.2. Section 2.3 explains the basic concepts of MPC. Special attention is paid to the case where a linear state space model is used to approximate the system dynamics. This type of controller is the corner stone of this thesis. Section 2.4 introduces the Kalman filter which is used for estimating the states of the river systems in the next chapters. Model reduction based on POD and Galerkin projection is explained in Section 2.5. Section 2.6 concludes this chapter.

2.2 Linear Quadratic Regulator

In optimal control theory one wants to control a given dynamical system at a minimum cost. One of the main results in optimal control theory is the Linear Quadratic Regulator (LQR), which applies to the situation when the system dynamics can be described with a linear differential equation or a linear

difference equation. The derivation of LQR will be given for the linear state space model in discrete time:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \tag{2.1}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k),$$

with $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$, $\mathbf{C} \in \mathbb{R}^{n_y \times n_x}$ and $\mathbf{D} \in \mathbb{R}^{n_y \times n_u}$ the state space matrices, n_x the number of states, n_u the number of inputs and n_y the number of outputs.

LQR minimizes the following cost function

$$\sum_{k=0}^{\infty} \big(\mathbf{x}(k)^{\mathrm{T}}\mathbf{Q}\mathbf{x}(k) + \mathbf{u}(k)^{\mathrm{T}}\mathbf{R}\mathbf{u}(k)\big), \tag{2.2}$$

with $\mathbf{Q} \in \mathbb{R}^{n_x \times n_x} \succeq 0$ and $\mathbf{R} \in \mathbb{R}^{n_u \times n_u} \succ 0$ positive semidefinite and definite weighting matrices. The higher the value of a diagonal element in any of these matrices, the more important the corresponding variable is. E.g. if all the elements in $\mathbf{Q}$ are much larger than the elements in $\mathbf{R}$, then the controller will mainly focus on steering the states as close as possible to the origin. The solution of the optimization problem with Eq. (2.2) as cost function and with the state equation Eq. (2.1) as constraint, is the linear feedback law [5, 65, 93]:

$$\mathbf{u}(k) = -\mathbf{K}\mathbf{x}(k) \tag{2.3}$$

where the feedback matrix $\mathbf{K} \in \mathbb{R}^{n_u \times n_x}$ is given by

$$\mathbf{K} = \big(\mathbf{R} + \mathbf{B}^{\mathrm{T}}\mathbf{S}\mathbf{B}\big)^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{S}\mathbf{A}.$$

$\mathbf{S} \in \mathbb{R}^{n_x \times n_x} \succ 0$ is the positive definite solution of the discrete time algebraic Riccati equation

$$\mathbf{S} = \mathbf{A}^{\mathrm{T}}\Big(\mathbf{S} - \mathbf{S}\mathbf{B}\big(\mathbf{R} + \mathbf{B}^{\mathrm{T}}\mathbf{S}\mathbf{B}\big)^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{S}\Big)\mathbf{A} + \mathbf{Q}.$$

More information on how to solve the discrete algebraic Riccati equation can be found in [9]. If there is a rate of change constraint for the inputs or if the inputs have some operational limits, then they have to be enforced using a saturator after calculating the control actions with the feedback law. The saturation of any of the control actions will destroy the optimality property of these control actions.

Combining the state feedback law with the state equation results in

$$\mathbf{x}(k+1) = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}(k).$$

It can be shown that the state feedback matrix $\mathbf{K}$ places the eigenvalues of the matrix $\mathbf{A} - \mathbf{BK}$ inside the unit circle (if $(\mathbf{A}, \mathbf{B})$ is stabilizable) [93]. The closed loop system is asymptotically stable and the controller will steer the states (and the outputs) to the origin.

The control actions given by Eq. (2.3) will always try to steer the outputs back to the origin. However in practice, it can be important that the outputs follow a certain reference signal $\mathbf{r}(k)$. In this thesis this reference signal will only contain step changes. It can be shown that a zero steady state error to this step input $\mathbf{r}(k)$ is achieved by the following modification of the feedback control law [65]:

$$\mathbf{u}(k) = -\mathbf{K}\mathbf{x}(k) + (\mathbf{N}_u + \mathbf{K}\mathbf{N}_x)\mathbf{r}(k) \tag{2.4}$$

where $\mathbf{N}_x \in \mathbb{R}^{n_x \times n_y}$ and $\mathbf{N}_u \in \mathbb{R}^{n_u \times n_y}$ can be found by solving

$$\begin{bmatrix} \mathbf{A} - \mathbf{I}_{n_x} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{N}_x \\ \mathbf{N}_u \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{n_x, n_y} \\ \mathbf{I}_{n_u, n_y} \end{bmatrix}.$$

Just as the control laws in Eqs. (2.3) or (2.4) do not know the limits on the control actions, the controller is also not aware of any possible constraints on the outputs or the states. Therefore it is difficult to tune the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ such that an LQR controller does not violate the process constraints. This situation does not occur with MPC.

2.3 Model Predictive Control

Model Predictive Control was first reported in the seventies of the 20th century and originates from the process industry [54, 140]. MPC was used to increase the energy efficiency of power plants and petroleum refineries, and to comply with the stricter environmental and safety regulations. Nowadays MPC, also known as Receding Horizon Control (RHC) or Moving Horizon Control (MHC), is one of the most advanced process control techniques successfully used in practice in a wide range of applications, going from the chemical sector and food processing to the automotive industry and aerospace applications [14, 15, 136, 154]. Reviews regarding the theoretical and practical aspects of MPC can be found in [66, 94, 124, 125, 128].

MPC does not refer to one specific control strategy, but to a collection of control techniques that explicitly makes use of a process model to predict the future system response and that optimizes the control actions within a certain prediction window. MPC solves at every time step a finite horizon open loop optimal control problem taking the current state of the process into

account. The result is a sequence of optimal future control actions over the entire prediction window from which only the first one is applied to the process. After taking new measurements and observations to determine the new current state of the system, the entire procedure is reiterated. In this way feedback is introduced in the control loop.

Solving at every time step an on-line optimization problem gives some specific advantages to MPC compared to conventional controllers that use a pre-computed control law [125]. Constraints on the process inputs and outputs can easily be taken into account by the controller improving the quality, effectiveness and safety of the process. MPC can also efficiently control systems with many internal interactions by using a multivariable process model in the formulation of the optimization problem. This process model can also include the effect of disturbances on the output of the system. Therefore, the controller can react on (estimates of) future disturbances before they actually occur. Another advantage is that the controller can easily be tailored to the specific application purpose because of the generality of the optimization problem. The only requirement is that the control problem should be formulated as an optimization problem that can be solved efficiently with respect to the control sampling time.

2.3.1 Linear Model Predictive Control

In theory there is no limitation on the type of the process model used in MPC. Transfer function models, state space models and finite impulse response models are the model types most often used with MPC [143]. For practical reasons regarding the optimization problem, this process model will often be a linear model. The model type used in combination with MPC throughout this thesis is a discrete-time linear time-invariant state space model of the following form:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{F}\mathbf{d}(k),$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k),$$

with $\mathbf{x}(k) \in \mathbb{R}^{n_x}$ the state vector, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ the input vector, $\mathbf{y}(k) \in \mathbb{R}^{n_y}$ the output vector, $\mathbf{d}(k) \in \mathbb{R}^{n_d}$ the disturbance vector and $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$, $\mathbf{C} \in \mathbb{R}^{n_y \times n_x}$, $\mathbf{D} \in \mathbb{R}^{n_y \times n_u}$ and $\mathbf{F} \in \mathbb{R}^{n_x \times n_d}$ the state space matrices modelling the system dynamics.

The receding horizon principle of MPC is visualized in Fig. 2.1. The linear model is used by the controller to predict the future states of the plant $\mathbf{x}(k+1), \ldots, \mathbf{x}(k+N_\mathrm{P})$ over a prediction window with size N_P from the current state of the system $\mathbf{x}(k)$. The sequence of control actions $\mathbf{u}(k), \ldots, \mathbf{u}(k+N_\mathrm{P}-1)$ is determined in such a way that a predefined objective or cost function is

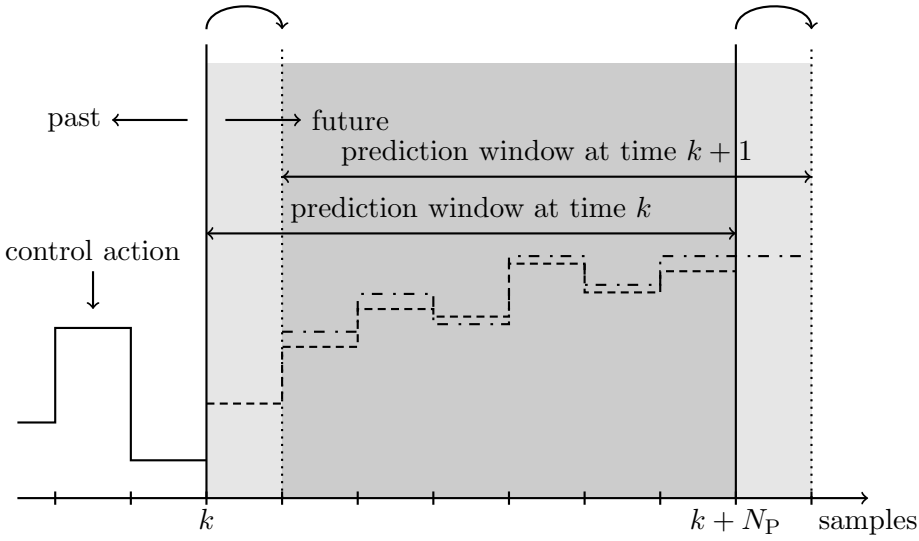


Figure 2.1: The receding horizon principle of MPC. At time step k the optimal control actions are calculated over a prediction horizon of size N_P. Only the first element of this sequence is applied to the process. Afterwards the prediction window is shifted one sampling step forward in time. Source: [133].

minimized. Only the first element of this sequence of optimal control actions is applied to the process. The new state of the process is measured or estimated at the sampling time $k+1$, and the entire procedure is repeated.

In order to track a reference trajectory for the state vector, the optimization problem within the MPC can be formulated as follows (e.g. at time step k equal to 0):

$$\min_{\mathbf{u},\mathbf{x}} \sum_{j=1}^{N_\mathrm{P}} \|\mathbf{x}(j) - \mathbf{r}_x(j)\|^2_{\mathbf{Q}} + \sum_{j=0}^{N_\mathrm{P}-1} \|\mathbf{u}(j) - \mathbf{u}(j-1)\|^2_{\mathbf{R}}$$

$$\begin{aligned}
\text{s.t. } & \mathbf{x}(0) = \hat{\mathbf{x}}, \\
& \mathbf{x}(j+1) = \mathbf{A}\mathbf{x}(j) + \mathbf{B}\mathbf{u}(j) + \mathbf{F}\mathbf{d}(j), && j = 0, \ldots, N_\mathrm{P} - 1 \\
& \mathbf{u}(-1) = \mathbf{u}_\mathrm{prev}, \\
& \mathbf{x}(j) \in \mathcal{X}, && j = 1, \ldots, N_\mathrm{P} \qquad (2.5) \\
& \mathbf{u}(j) \in \mathcal{U}, && j = 0, \ldots, N_\mathrm{P} - 1 \qquad (2.6)
\end{aligned}$$

with $\mathbf{r}_x \in \mathbb{R}^{n_x}$ the reference vector for the states, $\mathbf{Q} \in \mathbb{R}^{n_x \times n_x} \succeq 0$ and $\mathbf{R} \in \mathbb{R}^{n_u \times n_u} \succ 0$ positive semidefinite and definite weighting matrices, $\hat{\mathbf{x}} \in \mathbb{R}^{n_x}$ the current state of the system, $\mathbf{u}_{\text{prev}} \in \mathbb{R}^{n_u}$ the control actions applied in the previous time step, $\|\mathbf{y}\|_{\mathbf{Q}}^2 = \mathbf{y}^{\mathrm{T}}\mathbf{Q}\mathbf{y}$, Eqs. (2.5) and (2.6) the state and input constraints and $\mathcal{X}$ and $\mathcal{U}$ convex sets satisfying the following definition [31]:

Definition 1. *A set $\mathcal{S} \subseteq \mathbb{R}^n$ is called convex iff all convex combinations for any two points $\mathbf{x}$, $\mathbf{y} \in \mathcal{S}$ are also an element of the set $\mathcal{S}$:*

$$\forall \theta \in [0,1],\, \forall \mathbf{x},\, \mathbf{y} \in \mathcal{S} :\, (1-\theta)\mathbf{x} + \theta\mathbf{y} \in \mathcal{S}.$$

The formulated optimization problem is called a convex optimization problem because a convex objective function is minimized over convex sets. The important consequence of this convexity property is that any local minimum is also the global minimum of the optimization problem [31,130]. In this thesis the convex sets $\mathcal{X}$ and $\mathcal{U}$ are linear inequality constraints resulting in the following optimization problem:

$$\begin{aligned}
\min_{\mathbf{u},\mathbf{x}} \; & \sum_{j=1}^{N_{\mathrm{P}}} \|\mathbf{x}(j) - \mathbf{r}_x(j)\|_{\mathbf{Q}}^2 + \sum_{j=0}^{N_{\mathrm{P}}-1} \|\mathbf{u}(j) - \mathbf{u}(j-1)\|_{\mathbf{R}}^2 \\
\text{s.t. } & \mathbf{x}(0) = \hat{\mathbf{x}}, \\
& \mathbf{x}(j+1) = \mathbf{A}\mathbf{x}(j) + \mathbf{B}\mathbf{u}(j) + \mathbf{F}\mathbf{d}(j), && j = 0, \ldots, N_{\mathrm{P}} - 1 \\
& \mathbf{u}(-1) = \mathbf{u}_{\text{prev}}, \\
& \underline{\mathbf{x}} \leq \mathbf{x}(j) \leq \overline{\mathbf{x}}, && j = 1, \ldots, N_{\mathrm{P}} \\
& \underline{\mathbf{u}} \leq \mathbf{u}(j) \leq \overline{\mathbf{u}}, && j = 0, \ldots, N_{\mathrm{P}} - 1 \\
& |\mathbf{u}(j) - \mathbf{u}(j-1)| \leq \boldsymbol{\Delta}_u, && j = 0, \ldots, N_{\mathrm{P}} - 1
\end{aligned}$$

where $\underline{\mathbf{x}}$, $\overline{\mathbf{x}} \in \mathbb{R}^{n_x}$ are the lower and upper limits of the states, $\underline{\mathbf{u}}$, $\overline{\mathbf{u}} \in \mathbb{R}^{n_u}$ are the lower and upper limits of the inputs and $\boldsymbol{\Delta}_u \in \mathbb{R}^{n_u}$ is the maximal rate of change constraint for the inputs. The resulting optimization problem is called a Quadratic Programming problem (QP). For a positive semidefinite $\mathbf{Q}$ and a positive definite $\mathbf{R}$ it can be shown that this QP is strictly convex [31, 130]. Efficient solvers exist for solving convex QPs [7,76,123]. This is one of the main reasons why it is recommended to work with a linear state space model if it is an accurate enough approximation of the (possibly nonlinear) real process. Using a nonlinear state space model inside the optimization problem will in general destroy the convexity of the optimization problem resulting in a Nonlinear Programming problem (NLP) instead of a QP. These NLPs are in general much more difficult to solve and can suffer from multiple local minima.

2.4 Kalman Filter

Both MPC and LQR require the knowledge of the current state of the system to be able to generate proper control actions. In many applications it is practically impossible to measure every element of the state vector $\mathbf{x}(k)$ and in some cases the state vector does not even have a physical meaning. For the case of river systems, it would require an infinite number of sensors to know the water level profile and the discharge profile of every channel. However based on the measured outputs, the applied control actions and a mathematical model of the process, it is still possible to get an (accurate) estimate of the current state of the system by using an observer or soft-sensor. The state estimate will be denoted with $\hat{\mathbf{x}}(k)$. In general the observer of a physical system is derived from the mathematical model of the system, that is the linear state space model

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k),$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k).$$

Extra terms are included to ensure that, based on the measurements of the inputs and outputs over time, the state estimate converges to the process states. The so-called Luenberger observer [110] combines the plant model with the difference between the predicted outputs and the measured outputs:

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{L}(\mathbf{y}(k) - \hat{\mathbf{y}}(k)),$$

$$\hat{\mathbf{y}}(k) = \mathbf{C}\hat{\mathbf{x}}(k) + \mathbf{D}\mathbf{u}(k),$$

with $\mathbf{L} \in \mathbb{R}^{n_x \times n_y}$ the feedback gain matrix or observer matrix. The first term $\mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k)$ can be considered as a prediction step or the open loop estimation, while the second term $\mathbf{L}(\mathbf{y}(k) - \hat{\mathbf{y}}(k))$ is a correction step. It can be shown that the dynamics of the estimation error $\mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$ are given by $\mathbf{e}(k+1) = (\mathbf{A} - \mathbf{L}\mathbf{C})\mathbf{e}(k)$. If the eigenvalues of $\mathbf{A} - \mathbf{L}\mathbf{C}$ are within the unit circle, the observer is asymptotically stable and the estimation error converges to zero for k going to infinity. The velocity of this convergence is influenced by $\mathbf{L}$.

One approach to calculate the observer matrix $\mathbf{L}$ is based on optimal estimation techniques and is called the linear quadratic estimator, better known as the Kalman filter. This filter was introduced by R. E. Kalman in 1960 [84]. The so-called Kalman gain $\mathbf{L}$ minimizes the covariance of the estimation error $\mathbf{x}(k) - \hat{\mathbf{x}}(k)$ given by

$$\varepsilon\left\{\sum_{k=0}^{\infty} \|\mathbf{x}(k) - \hat{\mathbf{x}}(k)\|_2^2\right\}$$

for the following discrete-time process model

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{G}\mathbf{w}(k),$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) + \mathbf{v}(k).$$

where $\varepsilon\{\cdot\}$ denotes expectation, $\|\cdot\|_2$ the l_2-norm or the Euclidean norm, $\mathbf{G} \in \mathbb{R}^{n_x \times n_w}$ is a weighting matrix and $\mathbf{w}(k) \in \mathbb{R}^{n_w}$ and $\mathbf{v}(k) \in \mathbb{R}^{n_v}$ are random variables representing the process noise and the measurement noise respectively. Both random variables are assumed to be Gaussian white noise with zero mean and a covariance matrix $\mathbf{R}_w \in \mathbb{R}^{n_w \times n_w}$, $\mathbf{R}_v \in \mathbb{R}^{n_v \times n_v}$ respectively, and uncorrelated to each other:

$$\varepsilon\{\mathbf{w}(k)\} = \mathbf{0}_{n_w},$$

$$\varepsilon\{\mathbf{v}(k)\} = \mathbf{0}_{n_v},$$

$$\varepsilon\big\{\mathbf{w}(k)\mathbf{w}(k)^{\mathrm{T}}\big\} = \mathbf{R}_w,$$

$$\varepsilon\big\{\mathbf{v}(k)\mathbf{v}(k)^{\mathrm{T}}\big\} = \mathbf{R}_v,$$

$$\varepsilon\big\{\mathbf{v}(k)\mathbf{w}(k)^{\mathrm{T}}\big\} = \mathbf{0}_{n_v,n_w}.$$

The Kalman gain $\mathbf{L}$ is given by

$$\mathbf{L} = \mathbf{A}\mathbf{P}\mathbf{C}^{\mathrm{T}}\big(\mathbf{R}_v + \mathbf{C}\mathbf{P}\mathbf{C}^{\mathrm{T}}\big)^{-1}$$

with $\mathbf{P} \in \mathbb{R}^{n_x \times n_x} \succeq 0$ the positive semidefinite covariance matrix of the estimation error satisfying the discrete algebraic Riccati equation

$$\mathbf{P} = \mathbf{G}\mathbf{R}_w\mathbf{G}^{\mathrm{T}} + \mathbf{A}\mathbf{P}\mathbf{A}^{\mathrm{T}} - \mathbf{A}\mathbf{P}\mathbf{C}^{\mathrm{T}}\big(\mathbf{R}_v + \mathbf{C}\mathbf{P}\mathbf{C}^{\mathrm{T}}\big)^{-1}\mathbf{C}\mathbf{P}\mathbf{A}^{\mathrm{T}}.$$

More information about the derivation can be found in [65, 93].

2.5 Proper Orthogonal Decomposition

Proper Orthogonal Decomposition and Galerkin projection are well-known techniques for deriving reduced order models of high-dimensional systems. This high-dimensionality typically results from discretizing partial differential equations in space. An orthonormal basis for modal decomposition is extracted from a set of data called snapshots. These snapshots can be generated via simulations or experiments [101, 155]. The resulting basis functions are called POD basis vectors [96]. Besides providing the orthonormal basis, the POD technique also gives a measure of the importance of each basis vector. Selecting

the most relevant basis vectors and projecting (Galerkin projection) the high-dimensional model on the space spanned by these vectors results in a reduced order model of the process.

One can find many other names for POD. The most important ones are Karhunen-Loève Decomposition [108], Singular Value Decomposition (SVD) [8,42] and Principal Component Analysis [81]. Many people have contributed to the development of POD [96]. POD dates back to the 1950's with contributions by Kosambi (1943) [90], Loève (1945), Karhunen (1946), Pougachev (1953) and Obukhov (1954) [111]. PCA and SVD have even a longer history with the work of Pearson introducing PCA in 1901 [132], and the contributions of Beltrami (1873) [25], Jordan (1874) [82,83], Sylvester (1889) [165,166], Schmidt (1907) [146] and Weyl (1912) [187] for developing the theory around SVD. POD can be realized by performing the Karhunen-Loève Decomposition, via the Principal Component Analysis or via the Singular Value Decomposition. Liang et al. proved the equivalence of these three methods [96]. In this work the SVD will be used to realize the POD decomposition.

POD has been applied with success in a broad range of applications, for instance in studies of turbulence [26,88,112,155], vibration analysis [53,64], data analysis or compression [89,156], signal analysis, process identification, control in chemical engineering [2,3,11,17,87,92], etc. Also contributions in the field of water management can be found, e.g. for groundwater modelling [153,190], tsunami forecasting [68] and fluid control [137,192].

We will now explain how POD in combination with Galerkin projection can be used to reduce the order of a linear state space model. The derivation in the next two sections is based on [2].

2.5.1 POD basis vectors generation

The first step for finding the POD basis vectors is the generation of the snapshot matrix $\mathbf{X} \in \mathbb{R}^{n_x \times M}$ with n_x the number of states and M the number of samples. The snapshot matrix contains M samples of the state vector $\mathbf{x} \in \mathbb{R}^{n_x}$:

$$\mathbf{X} = [\mathbf{x}(k), \mathbf{x}(k+1), \ldots, \mathbf{x}(k+M-1)].$$

Each of these snapshots can be written as a linear combination of a set of ordered orthonormal basis vectors $\boldsymbol{\varphi}_j \in \mathbb{R}^{n_x}$, the so-called POD basis vectors:

$$\mathbf{x}(k+i) = \sum_{j=1}^{n_x} a_j(k+i)\boldsymbol{\varphi}_j, \text{ for } i = 0, \ldots, M-1, \tag{2.7}$$

with

$$a_j(k+i) = \langle \mathbf{x}(k+i), \boldsymbol{\varphi}_j \rangle = \boldsymbol{\varphi}_j^{\mathrm{T}} \mathbf{x}(k+i), \text{ for } j = 1, \ldots, n_x.$$

$\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product and $a_j(k+i)$ the POD coefficient, which is the coordinate of $\mathbf{x}(k+i)$ with respect to the basis vector $\boldsymbol{\varphi}_j$. The reduction in model order is achieved by working with an nth order approximation of $\mathbf{x}$, which is constructed for the n most relevant basis vectors:

$$\mathbf{x}_n(k+i) = \sum_{j=1}^{n} a_j(k+i)\boldsymbol{\varphi}_j, \text{ for } i = 0, \ldots, M-1,$$

with $n \ll n_x$. It can be shown that the orthonormal basis vectors $\boldsymbol{\varphi}_j$ calculated with POD have the property that the first n most relevant basis vectors reconstruct the snapshots in an optimal way: the Sum of Squared Error (SSE) E_n between the original samples $\mathbf{x}(k+i)$ and their nth order approximation $\mathbf{x}_n(k+i)$

$$E_n = \sum_{i=0}^{M-1} \|\mathbf{x}(k+i) - \mathbf{x}_n(k+i)\|_2^2$$

is minimal. The POD basis vectors are the solution of the following optimization problem:

$$\min_{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_n} \sum_{i=0}^{M-1} \left\| \mathbf{x}(k+i) - \sum_{j=1}^{n} \langle \mathbf{x}(k+i), \boldsymbol{\varphi}_j \rangle \boldsymbol{\varphi}_j \right\|_2^2$$

$$\text{s.t. } \boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{\varphi}_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

As shown in [92, 96], the solution of this optimization problem can be found by calculating the SVD of the snapshot matrix $\mathbf{X}$. The linear combination of a set of ordered orthonormal basis vectors used to rewrite $\mathbf{x}(k+i)$ (Eq. (2.7)), allows us to rewrite the snapshot matrix $\mathbf{X}$ in the following way:

$$\underbrace{[\mathbf{x}(k), \ldots, \mathbf{x}(k+M-1)]}_{\mathbf{X}} = \underbrace{[\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_{n_x}]}_{\boldsymbol{\Phi} \in \mathbb{R}^{n_x \times n_x}} \underbrace{\begin{bmatrix} a_1(k) & \ldots & a_1(k+M-1) \\ \vdots & \ddots & \vdots \\ a_{n_x}(k) & \ldots & a_{n_x}(k+M-1) \end{bmatrix}}_{\boldsymbol{\Gamma} \in \mathbb{R}^{n_x \times M}}$$

with $\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} = \mathbf{I}_{n_x}$. $\mathbf{X} = \boldsymbol{\Phi}\boldsymbol{\Gamma}$ is the proper orthogonal decomposition of $\mathbf{X}$ [8]. The matrix $\boldsymbol{\Phi}$ contains the orthonormal basis vectors while the matrix $\boldsymbol{\Gamma}$ contains the evolution of the POD coefficients. Both matrices can be found from the SVD of the snapshot matrix $\mathbf{X}$:

$$\mathbf{X} = \boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Psi}^{\mathrm{T}}$$

with $\mathbf{\Phi} = \left[\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_{n_x}\right] \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{\Psi} = [\psi_1, \ldots, \psi_M] \in \mathbb{R}^{M \times M}$ unitary matrices, $\mathbf{\Sigma} \in \mathbb{R}^{n_x \times M}$ a matrix containing the singular values σ_i, for $i = 1, \ldots, n_x$ of $\mathbf{X}$ in decreasing order on its main diagonal and $\mathbf{\Gamma} = \mathbf{\Sigma}\mathbf{\Psi}^{\mathrm{T}}$. These singular values are positive real numbers. The orthonormal basis vectors are the left singular vectors of $\mathbf{X}$. The minimum value of the SSE is a function of the singular values related to the unused POD basis vectors:

$$E_n = \sum_{j=n+1}^{n_x} \sigma_j^2.$$

The singular values quantify the importance of each of the POD basis vectors in capturing the information present in the snapshot matrix. This means that the first POD basis vector is the most relevant one and the last one is the least important. It is important to note that the POD basis vectors can only capture the information present in the snapshot matrix. Therefore it is of vital importance that the snapshot matrix is built with representative data of the process around the operating conditions of interest.

A criterion often used for selecting the value of n is the energy criterion [63]. This criterion checks the ratio between the modelled energy and the total energy contained in $\mathbf{X}$:

$$\bar{P}_n = \frac{\sum_{j=1}^{n} \sigma_j^2}{\sum_{j=1}^{n_x} \sigma_j^2}, \; n = 1, \ldots, n_x.$$

The value for n should be chosen such that the fraction of the first singular values is large enough to capture most of the energy in the data. The closer $\bar{P}_n$ is to 1, the better the approximation.

2.5.2 Model reduction via Galerkin projection

Model reduction based on the Galerkin projection will be explained for the linear state space model

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k),$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k),$$

with $\mathbf{x} \in \mathbb{R}^{n_x}$ the state vector, $\mathbf{u} \in \mathbb{R}^{n_u}$ the input vector, $\mathbf{y} \in \mathbb{R}^{n_y}$ the output vector, $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$, $\mathbf{C} \in \mathbb{R}^{n_y \times n_x}$ and $\mathbf{D} \in \mathbb{R}^{n_y \times n_u}$ the state space matrices.

The goal of model reduction is to approximate the linear state space model with a model with a lower complexity. This means finding a model with a

(much) smaller number of state variables. POD can be used to select the n most relevant basis vectors. The most common method for deriving the dynamical model for the POD coefficients a_j, for $j = 1, \ldots, n$ associated with the selected POD basis vectors is the Galerkin projection [2, 10, 16, 28, 79, 137] and is used in this work. The method starts by defining the residual function $R(\mathbf{x}(k+1), \mathbf{x}(k))$ for the state space equation:

$$R(\mathbf{x}(k+1), \mathbf{x}(k)) = \mathbf{x}(k+1) - (\mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)).$$

$R(\mathbf{x}_n(k+1), \mathbf{x}_n(k))$ is the residual for nth order approximation of the state vector $\mathbf{x}$

$$\mathbf{x}_n(k) = \sum_{j=1}^{n} a_j(k)\boldsymbol{\varphi}_j = \boldsymbol{\Phi}_n \mathbf{a}(k), \quad n \ll n_x,$$

with $\boldsymbol{\Phi}_n = [\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_n]$ and $\mathbf{a}(k) = [a_1(k), \ldots, a_n(k)]^{\mathrm{T}}$. The projection of the residual $R(\mathbf{x}_n(k+1), \mathbf{x}_n(k))$ on the space spanned by the basis vectors $\boldsymbol{\Phi}_n$ vanishes (the Galerkin projection):

$$\langle R(\mathbf{x}_n(k+1), \mathbf{x}_n(k)), \boldsymbol{\varphi}_j \rangle = 0, \text{ for } j = 1, \ldots, n. \tag{2.8}$$

The orthogonality of the residual to the span of the basis vectors means that the residual is minimal [10]. The model for the POD coefficients is found by replacing $\mathbf{x}$ with its nth order approximation $\mathbf{x}_n = \boldsymbol{\Phi}_n \mathbf{a}$ in the linear state space model:

$$\boldsymbol{\Phi}_n \mathbf{a}(k+1) = \mathbf{A}\boldsymbol{\Phi}_n \mathbf{a}(k) + \mathbf{B}\mathbf{u}(k),$$

$$\mathbf{y}(k) = \mathbf{C}\boldsymbol{\Phi}_n \mathbf{a} + \mathbf{D}\mathbf{u}(k).$$

Applying the inner production criterion (Eq. (2.8))

$$\langle \boldsymbol{\Phi}_n \mathbf{a}(k+1), \boldsymbol{\varphi}_j \rangle = \langle \mathbf{A}\boldsymbol{\Phi}_n \mathbf{a}(k) + \mathbf{B}\mathbf{u}(k), \boldsymbol{\varphi}_j \rangle, \text{ for } j = 1 \ldots, n,$$

$$\boldsymbol{\Phi}_n^{\mathrm{T}} \boldsymbol{\Phi}_n \mathbf{a}(k+1) = \boldsymbol{\Phi}_n^{\mathrm{T}} \mathbf{A}\boldsymbol{\Phi}_n \mathbf{a}(k) + \boldsymbol{\Phi}_n^{\mathrm{T}} \mathbf{B}\mathbf{u}(k)$$

and using the orthonormality of the basis vectors $\boldsymbol{\Phi}_n^{\mathrm{T}} \boldsymbol{\Phi}_n = \mathbf{I}_n$, the following reduced order state space model in terms of the POD coefficients can be found:

$$\mathbf{a}(k+1) = \mathbf{A}_{\mathrm{r}} \mathbf{a}(k) + \mathbf{B}_{\mathrm{r}} \mathbf{u}(k),$$

$$\mathbf{y}(k) = \mathbf{C}_{\mathrm{r}} \mathbf{a}(k) + \mathbf{D}_{\mathrm{r}} \mathbf{u}(k),$$

with $\mathbf{A}_{\mathrm{r}} = \boldsymbol{\Phi}_n^{\mathrm{T}} \mathbf{A}\boldsymbol{\Phi}_n \in \mathbb{R}^{n \times n}$, $\mathbf{B}_{\mathrm{r}} = \boldsymbol{\Phi}_n^{\mathrm{T}} \mathbf{B} \in \mathbb{R}^{n \times n_u}$, $\mathbf{C}_{\mathrm{r}} = \mathbf{C}\boldsymbol{\Phi}_n \in \mathbb{R}^{n_y \times n}$ and $\mathbf{D}_{\mathrm{r}} = \mathbf{D} \in \mathbb{R}^{n_y \times n_u}$. The resulting reduced order model is of order $n \ll n_x$.

2.6 Conclusions

The Linear Quadratic Regulator theory has been introduced. Its disadvantages related to the incapability of incorporating constraints on the states and inputs have been discussed.

These disadvantages are not present with Model Predictive Control (MPC). We have discussed the fundamentals of MPC, namely, the receding horizon principle, the optimization problem and the process model. We have shown that for a linear state space model with linear inequality constraints, the resulting optimization problem is a Quadratic Programming problem.

Also the basic concepts of Kalman filtering have been explained. The Kalman filter can be used to estimate the state of a system based on a limited number of measurements.

The last section reviewed how model reduction can be achieved with Proper Orthogonal Decomposition (POD) and Galerkin projection. POD is used to approximate the states of a system based on only a limited number of basis vectors. The reduced order model is derived by projecting (Galerkin projection) the original high-dimensional state space model onto the space spanned by the derived POD basis vectors.

3

Modelling of River Systems

3.1 Introduction

This chapter explains the different steps required to find an accurate model of a reach or an entire river system with tributaries, hydraulic structures, junctions and reservoirs. These modelling techniques will be used in the next chapters to assess the performance of the Model Predictive Control (MPC) strategies.

Section 3.2 explains how the dynamics of a single reach can be modelled. Here, the full hydrodynamic equations of de Saint-Venant will be introduced together with possible resistance laws taking the boundary friction of a channel into account. Section 3.3 explains how river systems can be built based on the equations for a single reach. Most attention is paid on how hydraulic structures can be modelled. Section 3.4 discusses two approaches usable for modelling reservoirs. Because the equations of de Saint-Venant do not have an analytical solution, numerical techniques are required to solve the model equations of river systems. Section 3.5 introduces a method based on finite-differences and on a staggered grid structure that can be used for simulating the dynamics of river systems. Attention is paid to the ordering of the variables, and the performance of the method is illustrated for the dam break problem. Due to the complexity of the mathematical models of river systems, these models cannot directly be used by MPC. Section 3.6 first gives a literature review of different types of approximate models. The second part of this section introduces a new type of approximate model, which is based on a linear approximation of the equations of de Saint-Venant in combination with the nonlinear gate

equations. The quality of the approximate model is evaluated based on two examples. This section also shows how an extra computational speed-up can be achieved by using Proper Orthogonal Decomposition in combination with Galerkin projection to reduce the order of the linear part of the approximate model. Section 3.7 concludes this chapter.

3.2 Modelling a single reach

3.2.1 The Saint-Venant equations

The full hydrodynamic equations of de Saint-Venant, or the so-called Saint-Venant equations, are often used in practice to model the evolution of water levels and discharges in river reaches. The equations are valid under the following five assumptions [43, 52, 104, 163]:

- The vertical pressure distribution is assumed to be hydrostatic. This is a valid assumption if the streamlines do not have sharp curvatures.
- The channel bottom slope is small so that the flow depth measured perpendicularly to the channel bottom or measured vertically are approximately the same.
- The bedding of the channel is stable: the bed elevation does not change with time.
- The flow is assumed to be one-dimensional: the water surface across any cross section is horizontal (transverse velocities are negligible) and an average boundary shear stress can be applied to the whole section.
- The frictional bed resistance is the same in unsteady flow as in steady flow meaning that steady state resistance laws can be used to evaluate the average boundary shear stress.

The Saint-Venant equations are the following set of hyperbolic partial differential equations (PDEs) [43, 52, 104, 163]:

$$\frac{\partial A(z,t)}{\partial h(z,t)}\frac{\partial h(z,t)}{\partial t} + \frac{\partial Q(z,t)}{\partial z} = 0, \tag{3.1}$$

$$\frac{\partial Q(z,t)}{\partial t} + \frac{\partial}{\partial z}\frac{Q(z,t)^2}{A(z,t)} + gA(z,t)\left(\frac{\partial h(z,t)}{\partial z} + S_{\mathrm{f}}(z,t) - S_0\right) = 0, \tag{3.2}$$

with

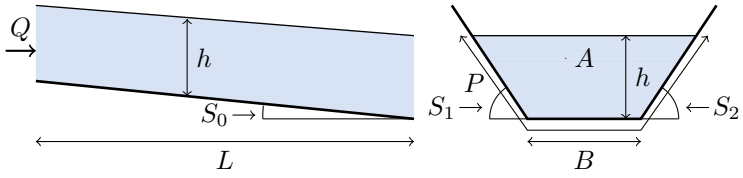


Figure 3.1: Parameters of a trapezoidal channel.

- t the time variable [s],
- z the space variable [m],
- $Q(z,t)$ the water discharge $\left[\mathrm{m}^3 \cdot \mathrm{s}^{-1}\right]$,
- $h(z,t)$ the water depth [m],
- $A(z,t)$ the cross-sectional flow area $\left[\mathrm{m}^2\right]$,
- g the gravity acceleration $\left[\mathrm{m} \cdot \mathrm{s}^{-2}\right]$,
- S_0 the bed slope $[-]$
- and $S_\mathrm{f}(z,t)$ the friction slope $[-]$.

Equation (3.1) describes the conservation of mass and Eq. (3.2) the conservation of momentum. The derivation of these two PDEs can be found in [43, 52]. $S_\mathrm{f}(z,t)$ represents the steady state resistance law that accounts for the frictional bed resistance. Examples of this resistance law are given in the next subsection.

The channels used in Chapters 3 and 4 will have a trapezoidal cross sectional shape and a constant bed slope S_0. An example of such a channel is given in Fig. 3.1, with L the length of the channel [m], B the bottom width [m], S_1 and S_2 the side slopes and P the wetted perimeter of the cross section [m]. When the control strategy is applied to the Demer in Chapter 5, the channels will have an irregular bed slope and irregular cross sectional profiles. The modelling of these channels is based on the data of the VMM of the Demer.

3.2.2 Modelling the friction slope S_f

The friction slope S_f is used to model the effect of the boundary friction of the channels. The two empirical resistance laws which are used the most in the literature are the resistance law of Chézy and the resistance law of Manning and will be shortly discussed here.

3.2.2.1 The resistance law of Chézy

The first resistance law was proposed in 1769 by the French engineer Antoine Chézy and is commonly known as the resistance law of Chézy [45]:

$$S_{\mathrm{f}} = \frac{Q|Q|}{C^2 A^2 R} \tag{3.3}$$

with C the Chézy coefficient $\left[\mathrm{m}^{1/2} \cdot \mathrm{s}^{-1}\right]$ and $R = A/P$ the hydraulic radius [m]. The problem with the Chézy equation is the estimation of the Chézy coefficient C. Because C depends on several parameters apart from the channel roughness, the attempts to develop a rational procedure for estimating the value of C have not been very successful [43].

3.2.2.2 The resistance law of Manning

A resistance law often used in practice is the resistance law of Manning [41, 43, 45, 163]:

$$S_{\mathrm{f}} = \frac{n_{\mathrm{mann}}^2 Q|Q|}{A^2 R^{1/3}} \tag{3.4}$$

with n_{mann} the Manning coefficient $\left[\mathrm{s} \cdot \mathrm{m}^{-1/3}\right]$. Comparing Eqs. (3.3) and (3.4), one can find the relation between the Chézy and the Manning coefficients:

$$C = \frac{R^{1/6}}{n_{\mathrm{mann}}}.$$

The advantage of the Manning formula over the Chézy equation is that the value of the Manning coefficient n_{mann} is found to be a characteristic of only the surface roughness of the channel, the amount of vegetation and channel irregularity, and to a lesser degree by channel alignment, stage, scour and deposition. One can find in [45] extensive tables with values of n_{mann} (maximum, minimum and normal) in function of the channel type and the channel material. Some typical values as a function of the channel material for natural streams are given in Table 3.1.

3.2.2.3 Equivalent Manning coefficient for channels with composite roughness

When one is working with academic or artificial examples of river systems, it is often assumed that the flow surface at every channel cross section has the same roughness along the entire wetted perimeter: the Manning coefficient is constant for every channel. However natural river systems have a

Table 3.1: Typical values of the Manning coefficient n_{mann} for natural streams (Source: [45]).

Channel type	$n_{\text{mann}}\ \left[\text{s} \cdot \text{m}^{-1/3}\right]$		
	minimum	normal	maximum
clean and straight	0.025	0.030	0.033
+ more stones and weeds	0.030	0.035	0.040
clean and winding	0.033	0.040	0.045
+ some stones and weeds	0.035	0.045	0.050
+ more stones and weeds	0.045	0.050	0.060

varying roughness (and hence a varying n_{mann} across their wetted perimeter). E.g. the VMM has approximated the cross sectional profiles with piecewise-linear functions at different locations along the Demer and its tributaries. A specific Manning coefficient is determined for every linear segment for each of these piecewise-linear approximations. The linear segments close to the bedding of the river typically have a lower value than the segments close to the flood levels because of the difference in vegetation. To simplify the computations one needs to determine an equivalent Manning coefficient n_{eqmann} based on these individual coefficients that is valid for the entire section and that can be used in Eq. (3.4).

Different approaches can be found in the literature for finding these equivalent coefficients. Consider an irregular cross section that can be divided in n subareas having a wetted perimeter P_i, a hydraulic radius R_i and a Manning constant $n_{\text{mann},i}$ $(i = 1, \ldots, n)$. A first approach assumes that the mean flow velocity in each of the segments is equal to the mean flow velocity in the entire section [60, 73]. This leads to the following equation:

$$n_{\text{eqmann}} = \left(\frac{\sum_{i=1}^{n} P_i n_{\text{mann},i}^{3/2}}{\sum_{i=1}^{n} P_i} \right)^{2/3} . \tag{3.5}$$

Many other formulas can be derived for the equivalent roughness starting from different assumptions. By assuming that the total force resisting the flow is equal to the sum of the forces resisting the flow in each segment [61, 127], the

following expression can be derived for the equivalent Manning constant:

$$n_{\text{eqmann}} = \left(\frac{\sum_{i=1}^{n} P_i n_{\text{mann},i}^2}{\sum_{i=1}^{n} P_i} \right)^{1/2}.$$

Starting from the assumption that the total discharge is equal to the sum of the discharge in each segment, the following formula is obtained [109]:

$$n_{\text{eqmann}} = \frac{P R^{5/3}}{\sum_{i=1}^{n} \left(\frac{P_i R_i^{5/3}}{n_{\text{mann},i}} \right)}.$$

Assuming a logarithmic velocity distribution, [91] derived the following equation:

$$\ln n_{\text{eqmann}} = \frac{\sum_{i=1}^{n} P_i y_i^{3/2} \ln n_{\text{mann},i}}{\sum_{i=1}^{n} P_i y_i^{3/2}}.$$

with y_i the flow depth. Based on the data for 36 natural channel cross sections obtained by U.S. Geological Survey, the equivalent Manning coefficient calculated with Eq. (3.5) gives the least error of the four mentioned approaches for calculating n_{eqmann} [126]. Therefore this equation will also be used for simulating the dynamics of the Demer in Chapter 5.

3.2.3 Limitations of the Saint-Venant equations

The Saint-Venant equations are derived based on the assumptions mentioned in Section 3.2.1. There are some cases for which these hypotheses do not hold [104].

3.2.3.1 Two-dimensional flow

The Saint-Venant equations assume the flow is one-dimensional. However for flood plains or large rivers, the flow cannot be considered to be one-dimensional and they need to be modelled with the two-dimensional Saint-Venant equations [43]. In this work we restrict ourselves to examples that can be reasonably modelled with the one-dimensional Saint-Venant equations. This is a valid assumption for most of the open channels that are controlled with hydraulic structures, like irrigation or drainage canals, regulated rivers and sewers.

3.2.3.2 Non-hydrostatic pressure distribution

For a non-hydrostatic pressure distribution the pressure term in the Saint-Venant equations needs to be modified. The need for a non-hydrostatic pressure distribution is usually linked to hydraulic phenomena with a small wave length (due to geometric variations or hydraulic variations). A more accurate model is provided by the Boussinesq equations which add a third order derivative term to the Saint-Venant equations [30, 172].

3.2.3.3 Sharp discontinuities

The Saint-Venant equations are not longer applicable when the flow encounters sharp discontinuities. These discontinuities can be caused by hydraulic structures. As it will be explained in the next section, these structures are treated as boundary conditions for the Saint-Venant equations and are modelled with a static algebraic relationship.

3.3 Boundary conditions

The Saint-Venant equations describe the dynamics of a channel. However one can only simulate the evolution of the water level and the discharge along the channel if these equations are combined with two boundary conditions. Possibilities are

- a given discharge or water level,
- a rating curve,
- a hydraulic structure
- or a junction.

These four types of boundary conditions will now be discussed in more detail.

3.3.1 Given discharge or water level

The most simple boundary condition for a reach is where the upstream or downstream discharge or water level is given. Examples when a discharge is given are a pump extracting or adding water to a reach or the upstream inflow

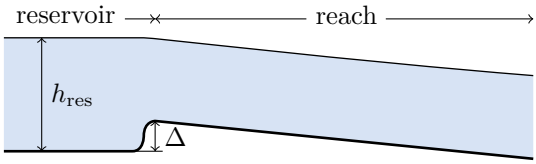


Figure 3.2: Reservoir connected to a reach.

of a reach modelled as a disturbance signal. For this last case, the boundary condition is equal to the following formulation:

$$Q(0,t) = Q_{\text{dist}}(t)$$

with Q_{dist} the (time-varying) disturbance signal. For the case of a given water level, one can think of the example where the channel is connected to a very large lake or reservoir. One can make an abstraction of this reservoir and assume that its water level will be a (known) constant value. For the reach in Fig. 3.2 this means that its upstream water level will be constant in time and equal to the water level of the reservoir (with correction of the difference in the height of their bed levels):

$$h(0,t) = h_{\text{res}} - \Delta.$$

3.3.2 Rating curves

A type of boundary condition often used at the downstream end of a reach are rating curves. A rating curve defines a relationship between the downstream water level and the downstream discharge. The rating curve can be determined by fitting a predefined model equation on a data set of measured water levels and discharges.

3.3.3 Hydraulic structures

3.3.3.1 Controllable hydraulic structures

In general

Hydraulic structures are installed in river systems to manipulate the discharges and the water levels along reaches as well as to influence the flow of water between the reaches and reservoirs. The predictive controllers designed in Chapters 4 and 5 will determine the optimal flow for each of these hydraulic

structures. Many types of hydraulic structures can be found in practice. Examples are sluices, weirs, culverts, pipes and orifices. In general the discharge through a gate can be modelled in the following way:

$$Q_{\text{gate}}(t) = \tilde{f}(c(t), h_{\text{up}}(t), h_{\text{down}}(t))$$

where $Q_{\text{gate}}(t)$ is the discharge controlled by the gate, $c(t)$ is the gate position [m], $h_{\text{up}}(t)$ is the water level on the upstream part of the gate, $h_{\text{down}}(t)$ is the water level on the downstream part and $\tilde{f} : \mathbb{R}^3 \to \mathbb{R}$ is a nonlinear function.

Given this equation, the boundary conditions for two channels i and j connected to each other with a gate m can be formulated as follows:

$$Q^{(i)}(L^{(i)}, t) = Q^{(m)}_{\text{gate}}(t), \tag{3.6}$$

$$Q^{(j)}(0, t) = Q^{(m)}_{\text{gate}}(t), \tag{3.7}$$

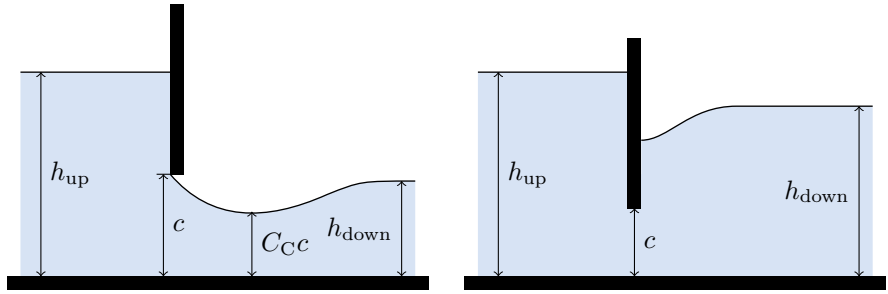
$$Q^{(m)}_{\text{gate}}(t) = \tilde{f}\Big(c^{(m)}(t), h^{(i)}(L^{(i)}, t), h^{(j)}(0, t)\Big) \tag{3.8}$$

where the notation with superscript $\cdot^{(i)}$ is used to indicate the channel, gate or reservoir the variable or parameter belongs to. E.g. $L^{(i)}$ corresponds with the length of channel i, while $c^{(m)}$ is the gate position of the gate m. Equations (3.6)-(3.8) simply mean that the gate discharge is equal to the discharge leaving the first channel and equal to the discharge entering the second channel.

The specific form of the function $\tilde{f}$ depends on the type of the structure. Two types of structures are used in this work and will be discussed in detail: the underflow vertical sluice and the gated weir.

Underflow vertical sluice

Modelling the discharge controlled by an underflow vertical sluice is extensively described in the literature. Most of the models make a distinction between free flow and submerged flow (see Fig. 3.3). In the free flow condition the downstream water level has no influence on the discharge. This can only happen if this water level is sufficiently small. If this is not the case, then the sluice is in the submerged flow condition and the downstream water level influences the discharge together with the gate opening and the upstream water level. A limit is needed to define the condition "small enough". This boundary is called the distinguishing condition and depends on the upstream water level, the gate opening and the contraction coefficient C_C. The contraction coefficient is defined as the ratio of the water level at the vena contracta (the minimum water depth in the downstream channel) to the gate opening. If the downstream



(a) Free flow (b) Submerged flow

Figure 3.3: Free flow (a) and submerged flow (b) condition for a vertical sluice with h_{up} the upstream water level, h_{down} the downstream water level, c the gate opening and C_C the contraction coefficient.

water level exceeds this limit then the sluice is in the submerged flow condition, in the other case it is in the free flow condition.

In the literature one can find different procedures for defining the discharge law $\tilde{f}$ for a vertical sluice [21, 23, 69, 70, 80, 98, 100, 117, 147, 152, 164, 195]. One way is to start from conservation laws to find a theoretical discharge law which is applicable in most cases. A second approach is to start from the theoretical discharge law, but further refine it by incorporating experimental knowledge.

Theoretical discharge law. Based on theoretical results one can derive the following expression for calculating the gate discharge for a vertical sluice with a rectangular cross section [69, 100, 152, 164, 195]:

$$Q_{\text{gate}}(t) = C_{\text{D}}(t) w c(t) \sqrt{2 g h_{\text{up}}(t)} \tag{3.9}$$

with C_{D} the discharge coefficient and w the width of the sluice [m]. The specific equation for the discharge coefficient depends on the flow condition:

$$\text{free flow:} \qquad C_{\text{D}}(t) = \frac{C_{\text{C}}}{\sqrt{1+\alpha(t)}}, \tag{3.10}$$

$$\text{submerged flow:} \quad C_{\text{D}}(t) = C_{\text{C}} \frac{\left[\beta(t) - \sqrt{\beta(t)^2 - \left(\frac{1}{\alpha(t)^2} - 1\right)^2 \left(1 - \frac{1}{\gamma(t)^2}\right)}\right]^{1/2}}{\frac{1}{\alpha(t)} - \alpha(t)} \tag{3.11}$$

with C_{C} the contraction coefficient, $\beta(t) = (1/\alpha(t) - 1)^2 + 2(\gamma(t) - 1)$, $\alpha(t) = C_{\text{C}} c(t)/h_{\text{up}}(t)$ and $\gamma(t) = h_{\text{up}}(t)/h_{\text{down}}(t)$. In practice the contraction

coefficient varies with the gate opening, the shape of the gate, the upstream and downstream water level, etc. Theoretical reasoning indicates that C_{C} can vary between 0.598 and 0.611 for a sharp-edged vertical sluice, while experimental results show a variation between 0.61 and 0.74 [195]. One can take C_{C} equal to 0.61 for engineering applications as a practical value with adequate precision [69, 98].

The following equation can be found for the distinguishing condition: [100, 152, 195]:

$$h_{\mathrm{down,max}}(t) = \frac{C_{\mathrm{C}}c(t)}{2}\left(\sqrt{1+\frac{16}{\alpha(t)(1+\alpha(t))}}\right). \tag{3.12}$$

The sluice is in free flow if $h_{\mathrm{down}}(t)$ is below $h_{\mathrm{down,max}}(t)$. Otherwise the sluice is in the submerged flow condition. Given Eqs. (3.10)-(3.12) everything is known to calculate the discharge for a vertical sluice with a rectangular cross section with Eq. (3.9) given the upstream water level, the downstream water level and the gate opening.

Until now it was assumed that $h_{\mathrm{up}} \geq h_{\mathrm{down}}$. Equation (3.9) can easily be extended to the general case. If $h_{\mathrm{down}}(t)$ becomes larger than $h_{\mathrm{up}}(t)$, the roles of $h_{\mathrm{up}}(t)$ and $h_{\mathrm{down}}(t)$ interchange and the discharge becomes negative.

Empirical discharge laws. In the literature other equations can be found for describing the discharge controlled by an underflow vertical sluice. These equations are often based on the theoretical discharge law (Eq. (3.9)) but some adaptations are performed based on numerical experiments.

A first group of models uses exactly the theoretical discharge law (for free flow and submerged flow) but determines the value for C_{D} using experimental results. E.g. in [70] the discharge coefficient is evaluated experimentally which resulted in a nomogram which can be used to find the value of C_D given the current upstream and downstream water levels and the gate opening. Another example can be found in [164] where a nonlinear regression is performed on experimental data.

A second group of models uses an adapted version of the theoretical discharge law for both the free flow condition and the submerged flow condition. A vertical sluice is modelled in [147] by the equations

$$\text{free flow:} \qquad Q_{\mathrm{gate}}(t) = C_D w C_C c(t)\sqrt{2g(h_{\mathrm{up}}(t) - C_C c(t))},$$

$$\text{submerged flow:} \qquad Q_{\mathrm{gate}}(t) = C_D w C_C c(t)\sqrt{2g(h_{\mathrm{up}}(t) - h_{\mathrm{down}}(t))}.$$

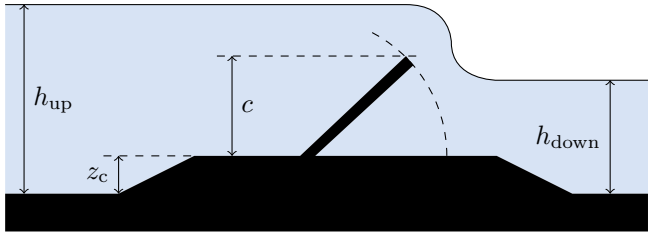


Figure 3.4: Gated weir with h_{up} the upstream water level, h_{down} the downstream water level, c the height of the gate and z_c the sill level.

Other modifications of the theoretical discharge law can be found in [21, 23, 80, 117], but all of them are very similar to each other.

One step further in using empirical discharge laws for modelling a vertical sluice can be found in software packages like SIC, HEC-RAS [39], SOBEK and InfoWorks. These packages use different equations for specific ranges of the influencing variables resulting in accurate results but also many more parameters to be tuned.

Equations used. In this work Eqs. (3.9)-(3.12) will be used for modelling the vertical sluices. Experimental results indicate that this approach showed a high performance consistency for three different real gates [152]. The experimental discharge laws achieve equal or better results but require experimental data for calibrating the different parameters.

Gated weir

One of the structures used to control the Demer is the gated weir and is shown in Fig. 3.4 where z_{c} is the bottom elevation of the weir [m], the so-called sill level. The discharge passing over the gate can be influenced by changing the rotation angle of the weir. There are two different operating modes depending on the gate position and the upstream and downstream water level: the gate control mode and the throat control mode. In the gate control mode the discharge over the gate is influenced by the gate position: decreasing or increasing the gate position will increase, respectively decrease the discharge over the gate. One can reduce the rotation angle until the discharge over the gate is at its maximal value (for a given upstream and downstream water level). Further decreasing the rotation angle will not affect the discharge over the gate: the gate is in throat control mode and the discharge is influenced by z_{c}. Again just as for the underflow vertical sluice both control modes can be further divided in a free flow condition and a submerged flow condition. In the free flow condition

the discharge is independent of the downstream water level, in the submerged flow condition the downstream water level influences the discharge.

Equations describing the relation between the discharge over the weir, the upstream and downstream water level and the gate position can be found in [147]. For example, for the throat control mode the following equations are typically used:

$$\text{free flow:} \qquad Q_{\text{gate}}(t) = 0.5 C_{\text{D}} w \sqrt{g} \left(\frac{2}{3} (h_{\text{up}} - z_{\text{c}}) \right)^{1.5}, \quad (3.13)$$

$$\text{submerged flow:} \qquad Q_{\text{gate}}(t) = 0.5 \gamma C_{\text{D}} w \sqrt{g} \left(\frac{2}{3} (h_{\text{up}} - z_{\text{c}}) \right)^{1.5} \quad (3.14)$$

with γ a coefficient for correcting the effect of submerged flow. Similar equations can be found for the gate control mode.

In this work the gated weirs are used when MPC is tested on the Demer. The equations used to model these gated weirs are taken from the InfoWorks model. These equations are similar to Eqs. (3.13) and (3.14) with some small modifications based on measurement data. More information can be found in Appendix A.1 and in the manual of the software InfoWorks-RS [78].

3.3.3.2 Fixed hydraulic structures

Many different types of structures exist in river systems for which the discharge over the structure cannot be changed by an operator in contrast to e.g. a gated weir. An example of such a structure is a spill. A spill provides the release of flow from an upstream area to a downstream area over a jagged or irregular weir. It can be used to model in-line flows over irregular weirs, as well as lateral flows, such as those over embankments between two open channels or between an open channel and a flooded area. The discharge over a spill can in general be modelled with the following equation:

$$Q_{\text{spill}}(t) = \tilde{g}(h_{\text{up}}(t), h_{\text{down}}(t))$$

where $Q_{\text{spill}}(t)$ is the discharge over the spill, $h_{\text{up}}(t)$ is the water level on the upstream part of the spill, $h_{\text{down}}(t)$ is the water level on the downstream part of the spill and $\tilde{g} : \mathbb{R}^2 \to \mathbb{R}$ is a nonlinear function.

This model structure will only be used in Chapter 5 where a mathematical model is constructed for the Demer. The equations used to model the spill are based on the equations used by the InfoWorks model of the Demer [78]. Internally the function $\tilde{g}$ splits the calculation of the flow over the irregular weir into separate flows over segments. The equation calculating these separate flows

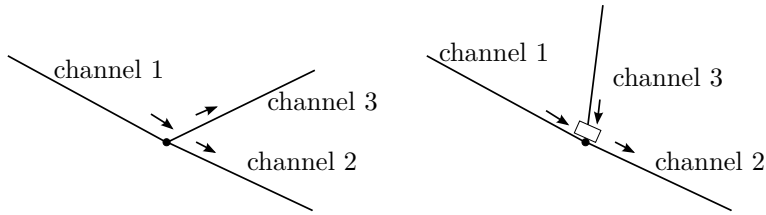


Figure 3.5: Two examples of junctions where the white rectangle represents a gate.

takes into account whether the weir is in free flow or in submerged flow and in forward or reverse mode. These separate flows are summed together to give the discharge of the entire spill. The equations are given in Appendix A.2. More information can be found in the manual of the software InfoWorks-RS [78].

3.3.4 Junctions

Junctions are places where three or more reaches coincide or where a reach (or reservoir) is connected via a hydraulic structure to two other reaches. At these points all the water levels of the channels starting or ending in this junction should be equal and the sum of the discharges entering the junction should be equal to the sum of the discharges leaving the junction. Applying this to the example on the left of Fig. 3.5, results in the following equations:

$$h^{(1)}(L^{(1)},t) = h^{(2)}(0,t),$$

$$h^{(1)}(L^{(1)},t) = h^{(3)}(0,t),$$

$$Q^{(1)}(L^{(1)},t) = Q^{(2)}(0,t) + Q^{(3)}(0,t).$$

The boundary conditions for the three channels of the example on the right are a combination of the boundary conditions of the junction and the boundary conditions of the gate:

$$h^{(1)}(L^{(1)},t) = h^{(2)}(0,t),$$

$$Q^{(1)}(L^{(1)},t) + Q_{\text{gate}}(t) = Q^{(2)}(0,t),$$

$$Q^{(3)}(L^{(3)},t) = Q_{\text{gate}}(t),$$

$$Q_{\text{gate}}(t) = \tilde{f}\Big(c^{(\text{gate})}(t), h^{(3)}(L^{(3)},t), h^{(2)}(0,t)\Big).$$

The discharge through the gate is used as downstream boundary condition of the third reach.

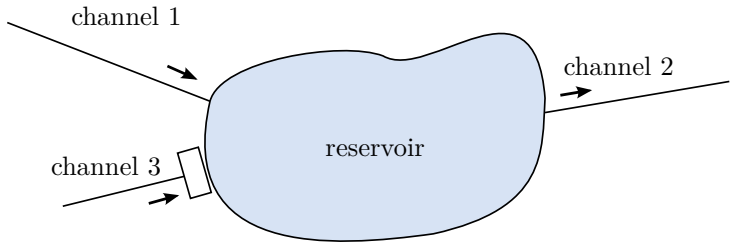


Figure 3.6: Example of a reservoir connected to two reaches and one gate.

3.4 Modelling of reservoirs

Water reservoirs are used to temporally store the excess of water during periods of heavy rainfall. One approach to describe the dynamics of such a water reservoir is to model it as a storage node or as one big tank that maintains water continuity. This means that the change in water volume of the reservoir is related to the amount of water entering and leaving the reservoir at every time instant:

$$dV_{\text{res}}/dt = Q_{\text{in}}(t) - Q_{\text{out}}(t) \tag{3.15}$$

with V_{res} the volume of the reservoir $\left[\text{m}^3\right]$, Q_{in} the sum of inflowing discharges through gates or reaches and Q_{out} the sum of outgoing discharges through gates or reaches. Applying Eq. (3.15) to the system presented in Fig. 3.6, the reservoir dynamics are described by

$$dV_{\text{res}}/dt = Q^{(1)}(L^{(1)}, t) + Q_{\text{gate}}(t) - Q^{(2)}(0, t).$$

The boundary condition of the reaches connected to the reservoir is given by the water level of this reservoir. Also the discharge through gates directly connected to a reservoir depends on the actual water level of the reservoir. Therefore the relationship between the volume of the reservoir V_{res} and its corresponding water level h_{res} needs to be known. This relation can be derived based on the actual shape of the reservoir and is typically nonlinear.

The advantage of modelling a reservoir as a tank is the simplicity of the model: only one variable is needed to model the dynamics of the entire reservoir. However it has as disadvantage that the time needed for water to travel from one point to another is not included in the model. If the total inflowing discharge Q_{in} and the total outgoing discharge Q_{out} increase with the same amount at every time step, the water level of the reservoir will not change. This means that the extra amount of water coming into the reservoir at one point is instantaneously levelled out with the amount of water being removed at

another point around the reservoir without taking into account the time needed for the water to travel from the first point to the second. In reality there will be a difference in the water levels at the point where water is added, the extraction point and the middle of the reservoir. For small reservoirs, this effect could be neglected, but for larger reservoirs these delay effects will increase and it can become necessary to include them.

One approach to overcome this disadvantage is to model the dynamics of a reservoir in the same way as a channel. This means that the Saint-Venant equations (Eqs. (3.1) and (3.2)) will be used to model the dynamics of the water levels and discharges along the reservoir. In this way the effect of adding and extracting water of the reservoir at different places is automatically included but at the cost of having a higher number of variables. Because one of the assumptions of the Saint-Venant equations is that the flow is one-dimensional, this approach is only suitable if the reservoir is not too wide. Wider reservoirs can be modelled with the equations describing unsteady two-dimensional flows or based on the one-dimensional equations in the main flow direction [43]. However this approach is considered to be beyond the scope of this work and will therefore not be used.

3.5 Numerical solution method

Based on the modelling techniques seen in Sections 3.2, 3.3 and 3.4 a mathematical model can be derived for a large class of river systems. However because of the Saint-Venant equations, an analytical solution cannot be found for the resulting system of equations, except for some simplified cases. Therefore these equations need to be integrated numerically. Several approaches can be found in the literature [18, 43]:

- finite-difference methods [43, 134, 135, 163],
- method of characteristics [43, 121, 122, 158, 159, 163, 191],
- finite-element methods [12, 71, 86],
- finite-volume methods [72, 196]
- spectral methods [40]
- and boundary-element methods [97].

Because the first category is used the most in simulating the Saint-Venant equations, we implemented a finite-difference method for simulating river systems. The development of the specific finite-difference method used in this work will be explained in the following subsections and is based on [157].

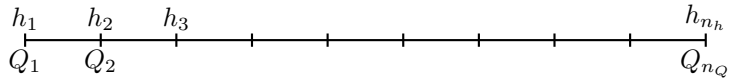


Figure 3.7: Grid structure used by the Preissmann scheme.

3.5.1 Preissmann scheme

3.5.1.1 The method

The Preissmann scheme is a numerical method for solving the Saint-Venant equations which is often used in the literature [52, 99, 135]. The infinite dimensional variables h and Q are approximated on a finite grid in space and time. The spatial discretization is performed on a grid structure where at every discretization point a water level and a discharge is modelled (see Fig. 3.7 where the leftmost grid point coincides with the upstream part of the reach and the rightmost grid point with the downstream part of the reach). The total number of water levels n_h is equal to the number of discharges n_Q. The partial derivatives are approximated with finite differences with respect to space and time (where the following notation $f(z_j, t_k) = f_{j,k}$ is used for clarity and ease of notation):

$$f(z,t) \cong \theta(\phi f_{j+1,k+1} + (1-\phi) f_{j,k+1}) + \\ + (1-\theta)(\phi f_{j+1,k} + (1-\phi) f_{j,k}) \tag{3.16}$$

$$\frac{\partial f}{\partial z} \cong \frac{1}{\Delta z}(\theta(f_{j+1,k+1} - f_{j,k+1}) + (1-\theta)(f_{j+1,k} - f_{j,k})) \tag{3.17}$$

$$\frac{\partial f}{\partial t} \cong \frac{1}{\Delta t}(\phi(f_{j+1,k+1} - f_{j+1,k}) + (1-\phi)(f_{j,k+1} - f_{j,k})) \tag{3.18}$$

with Δt the time step, Δz the spatial discretization step, $\theta \in [0,1]$ the discretization parameter related to the time domain and $\phi \in [0,1]$ the discretization parameter related to the spatial domain. Equations (3.16)-(3.18) are used for approximating the different terms in the two PDEs related to the variables h, Q, A and S_f. The method is schematically visualized in Fig. 3.8. In most situations ϕ is set equal to 0.5 for the ease of calculation. Depending on the discretization parameter θ, one can find different schemes in the literature:

- $\theta = 0$: explicit scheme,
- $\theta = 1/2$: classic Preissmann,
- $\theta = 1$: complete implicit scheme.

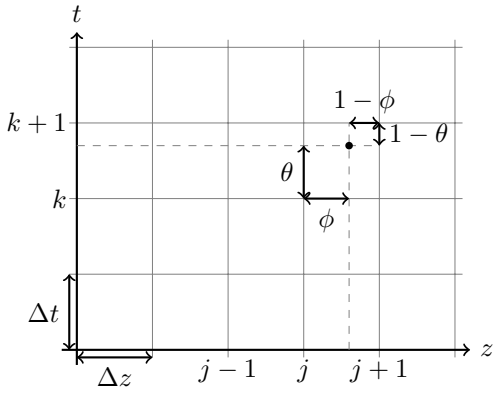


Figure 3.8: Schematic presentation of the working procedure of the Preissmann scheme (Source [52]).

Applying these approximations to the different terms in Eqs. (3.1) and (3.2), the Saint-Venant equations are transformed into a system of nonlinear equations. Combining these equations for every reach together with their boundary conditions, gives rise to one big system of equations which has to be solved for the water levels and discharges at the grid points at time t_{k+1} given their values at time t_k.

3.5.1.2 Stability analysis

One can show that for a linearized form of the Saint-Venant equations the condition for numerical stability of the Preissmann scheme depends on θ [163]. For θ in the range between 0.5 and 1 the Preissmann scheme is unconditionally stable [52]. For $\theta < \frac{1}{2}$ the solution grows with time and is always unstable. The scheme is second order accurate for $\theta = \frac{1}{2}$. In this case the scheme is non-dissipative (the solution is not damped with time nor does it grow with time) and it can cause numerical instability, solution failure or undesirable oscillations in the solution [47, 50]. For these reasons, a value of $\theta > 0.5$ is needed to damp out the (numerically induced) oscillations. For $\theta = 1$ the output will have no oscillations but it may also have numerical diffusion which reduces the accuracy [50]. Different ranges of values for θ are recommended in the literature, e.g. $0.6 \leq \theta \leq 1$ [98], $0.55 \leq \theta \leq 0.6$ [50,160], $0.6 \leq \theta \leq 0.67$ [47], while [49, 80] uses $\theta = 0.6$ and in [4] θ is set equal to 0.65.

Even though the Preissmann scheme is unconditionally stable for $\theta \in [0.5, 1]$, not every Δz and Δt are appropriate for this implicit scheme to achieve a good accuracy and numerical stability [160]. A high value of the Courant number

typically leads to numerical instabilities [47], with the Courant number given as the ratio between the actual wave velocity and the numerical wave velocity:

$$C_{\mathrm{n}} = \frac{|v| \pm \sigma}{\Delta z / \Delta t}$$

with $v = Q/A$ the flow velocity $\left[\mathrm{m} \cdot \mathrm{s}^{-1}\right]$ and $\sigma = \sqrt{gD}$ the wave celerity $\left[\mathrm{m} \cdot \mathrm{s}^{-1}\right]$ with D the hydraulic depth (area over top width) [m]. Since it is difficult to change Δz during a simulation the only option to decrease C_{n} is by decreasing Δt. Numerical stability and accuracy are generally improved by using smaller time steps. However this comes at the cost of greater computation time and expense [47]. When one is simulating a river system, one should adapt Δt during the simulation such that the maximum Courant number for all grid points is close to 1 [52]. If C_{n} is too large, Δt should be decreased, the equations need to be resolved, the Courant number is recalculated for every grid point and checked again. If the Courant number is too small, the simulation solution can be accepted but Δt can be increased to allow larger time steps in the future.

3.5.1.3 Disadvantage of the Preissmann scheme

One disadvantage of the Preissmann scheme is that one needs to make a choice between using forward or backward differences for approximating the partial derivatives. The best choice depends on the river characteristics, and the initial and boundary conditions. In the next part we will introduce an approach that does not suffer from this drawback.

3.5.2 Preissmann scheme applied to staggered grid structure in combination with upwinding approach

3.5.2.1 The method

To overcome the disadvantage of the Preissmann scheme, the numerical scheme that will be used throughout this dissertation, uses a staggered grid structure for the spatial discretization (see Fig. 3.9) following the approach introduced by Stelling and Duinmeijer [157, 193]. Every grid point now models only a water level or a discharge, not both (except at the upstream and downstream boundary). Following the approach of the standard Preissmann scheme, the partial derivatives in the first PDE of the Saint-Venant equations (Eq. (3.1)) for a channel i are approximated in the following way (note $Q^{(i)}(z_j, t_k) = Q^{(i)}_{j,k}$

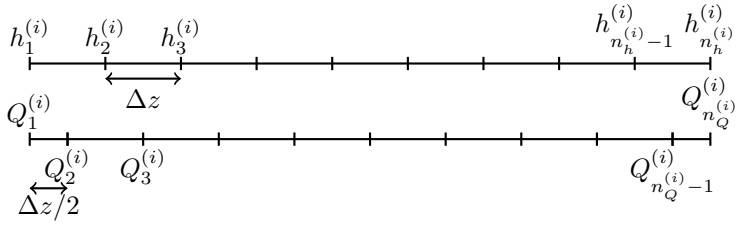


Figure 3.9: Staggered grid structure used by the adapted Preissmann scheme.

and $Q^{(i)}_{j,k+\theta} = \theta Q^{(i)}_{j,k+1} + (1-\theta) Q^{(i)}_{j,k}$):

$$\frac{\partial h^{(i)}_{j,k}}{\partial t} \simeq \frac{h^{(i)}_{j,k+1} - h^{(i)}_{j,k}}{\Delta t},$$

$$\frac{\partial Q^{(i)}_{j,k}}{\partial z} \simeq \frac{Q^{(i)}_{j,k+\theta} - Q^{(i)}_{j-1,k+\theta}}{\Delta z},$$

$$\frac{\partial A^{(i)}_{j,k}}{\partial h} \simeq \left(\frac{\partial A}{\partial h}\right)^{(i)}_{j,k+\theta}.$$

Notice that in the approximation of the partial derivative $\partial Q/\partial z$, it is not needed to make a choice between using forward or backward differences. The reason is that the grid point modelling the water level lies in between two grid points modelling a discharge (in contrast to the standard Preissmann scheme). A similar approach is used for the terms $\partial Q(z,t)/\partial t$, $A(z,t)$, $\partial h(z,t)/\partial z$ and $S_\mathrm{f}(z,t)$ in the second PDE (Eq. (3.2)). The advection term $\partial(Q^2(z,t)/A(z,t))/\partial z$ is approximated with an upwinding approach:

$$\frac{\partial}{\partial z}\left(\frac{{Q^{(i)}}^2}{A^{(i)}}\right)_{j,k} \simeq \begin{cases} \frac{1}{\Delta z}\left(\left(\frac{{Q^{(i)}}^2}{A^{(i)}}\right)_{j+1,k+\theta} - \left(\frac{{Q^{(i)}}^2}{A^{(i)}}\right)_{j,k+\theta}\right) & Q^{(i)}_{j,k} < 0, \\ \frac{1}{\Delta z}\left(\left(\frac{{Q^{(i)}}^2}{A^{(i)}}\right)_{j,k+\theta} - \left(\frac{{Q^{(i)}}^2}{A^{(i)}}\right)_{j-1,k+\theta}\right) & Q^{(i)}_{j,k} \geq 0. \end{cases}$$

Fig. 3.10 visualizes the stencils used to approximate a term related to a water level or a discharge at z_j. In this way the two PDEs describing the dynamics of a single channel i are transformed into a system of nonlinear equations. The actual form of the staggered grid structure at its boundaries depends on the boundary conditions of the reach. Fig. 3.11 shows the two possibilities for the upstream boundary of a reach. A similar structure exists for the downstream part of the reach. The left plot applies to the situation where the upstream boundary condition is a given discharge, a rating curve, a hydraulic structure or a junction and the right plot applies when the upstream water level is given

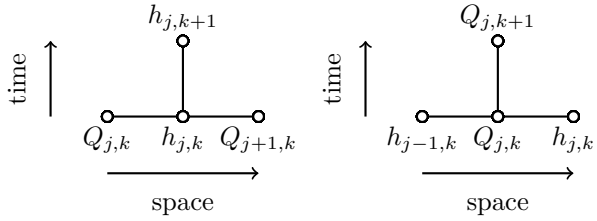


Figure 3.10: Stencils used in the approximation of the Saint-Venant equations for the terms related to water levels (left) and discharges (right).

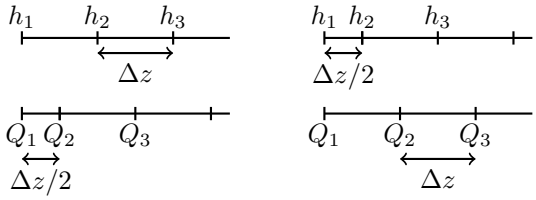


Figure 3.11: Possible boundaries of the staggered grid structure of a reach at its upstream side. The left plot applies when the upstream boundary condition is a given discharge, a rating curve, a hydraulic structure or a junction. The right plot applies for a given upstream water level or when a reservoir is directly connected to the upstream part of the reach.

or when a reservoir is directly connected to the upstream part of the reach. For the situation where the upstream and downstream boundary conditions are given discharges, the first PDE is transformed into the following equation:

$$\frac{\partial A(z,t)}{\partial h(z,t)}\frac{\partial h(z,t)}{\partial t} + \frac{\partial Q(z,t)}{\partial z} = 0$$

$$\Rightarrow \text{ for } j = 1, \ldots, n_h^{(i)} :$$

$$\left(\frac{\partial A}{\partial h}\right)^{(i)}_{j,k+\theta} \frac{h^{(i)}_{j,k+1} - h^{(i)}_{j,k}}{\Delta t} + \frac{Q^{(i)}_{j,k+\theta} - Q^{(i)}_{j-1,k+\theta}}{\Delta z} = 0 \qquad (3.19)$$

where Δz is replaced by $\Delta z/2$ for $j = 1$ and $j = n_h^{(i)}$. The second PDE is approximated with the following equations:

$$\frac{\partial Q(z,t)}{\partial t} + gA(z,t)\left(\frac{\partial h(z,t)}{\partial z} + S_{\mathrm{f}}(z,t) - S_0\right) + \frac{\partial}{\partial z}\frac{Q(z,t)^2}{A(z,t)} = 0$$

$$\Rightarrow \text{ for } j = 2, \ldots, n_Q^{(i)} - 1 :$$

$$\frac{Q_{j,k+1}^{(i)} - Q_{j,k}^{(i)}}{\Delta t} + gA_{j,k+\theta}^{(i)} \left(\frac{h_{j,k+\theta}^{(i)} - h_{j-1,k+\theta}^{(i)}}{\Delta z} + S_{\mathrm{f}_{j,k+\theta}}^{(i)} - S_0 \right) +$$

$$+ \begin{cases} \frac{1}{\Delta z} \left(\left(\frac{Q^{(i)2}}{A^{(i)}} \right)_{j+1,k+\theta} - \left(\frac{Q^{(i)2}}{A^{(i)}} \right)_{j,k+\theta} \right) & Q_{j,k}^{(i)} < 0, \\ \frac{1}{\Delta z} \left(\left(\frac{Q^{(i)2}}{A^{(i)}} \right)_{j,k+\theta} - \left(\frac{Q^{(i)2}}{A^{(i)}} \right)_{j-1,k+\theta} \right) & Q_{j,k}^{(i)} \geq 0 \end{cases} = 0. \quad (3.20)$$

This set of equations can be written more compactly as follows:

$$\mathbf{f}\Big(\mathbf{h}^{(i)}(t_{k+1}), \mathbf{h}^{(i)}(t_k), \mathbf{q}^{(i)}(t_{k+1}), \mathbf{q}^{(i)}(t_k) \Big) = \mathbf{0}_{n_h^{(i)} + n_Q^{(i)} - 2}, \quad (3.21)$$

with $\mathbf{h}^{(i)}(t_k) = \left[h_1^{(i)}(t_k), \ldots, h_{n_h^{(i)}}^{(i)}(t_k) \right]^{\mathrm{T}}$, $\mathbf{q}^{(i)}(t_k) = \left[Q_1^{(i)}(t_k), \ldots, Q_{n_Q^{(i)}}^{(i)}(t_k) \right]^{\mathrm{T}}$ and $\mathbf{f} : \mathbb{R}^{2 \cdot \left(n_h^{(i)} + n_Q^{(i)} \right)} \rightarrow \mathbb{R}^{n_h^{(i)} + n_Q^{(i)} - 2}$. The system of nonlinear equations consisting of Eq. (3.21) for each channel together with all the boundary conditions needs to be solved for $\mathbf{h}^{(i)}(t_{k+1})$ and $\mathbf{q}^{(i)}(t_{k+1})$ of every channel given the values of these variables at time step t_k. This can be done with the Newton-Raphson method. The same limits on θ, Δz and Δt apply as the one mentioned for the standard Preissmann scheme. In the remaining part of this work θ is set equal to 0.6.

This discretization technique is also applied to the equations of Sections 3.3 and 3.4 describing the boundary conditions and the reservoir dynamics respectively. E.g., when applied to Eq. (3.15), the resulting equation is as follows:

$$dV_{\mathrm{res}}/dt = Q_{\mathrm{in}}(t) - Q_{\mathrm{out}}(t)$$

$$\Rightarrow \frac{V_{\mathrm{res}}(t_{k+1}) - V_{\mathrm{res}}(t_k)}{\Delta t} = \theta(Q_{\mathrm{in}}(t_{k+1}) - Q_{\mathrm{out}}(t_{k+1})) + (1 - \theta)(Q_{\mathrm{in}}(t_k) - Q_{\mathrm{out}}(t_k)).$$

3.5.2.2 Structure exploitation

At every time step a nonlinear system of equations needs to be solved. One important aspect to solve this system of equations in an efficient way is to calculate the Jacobian $\mathbf{J}$ of the river equations in an analytical way (and not e.g. by using finite differences):

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}},$$

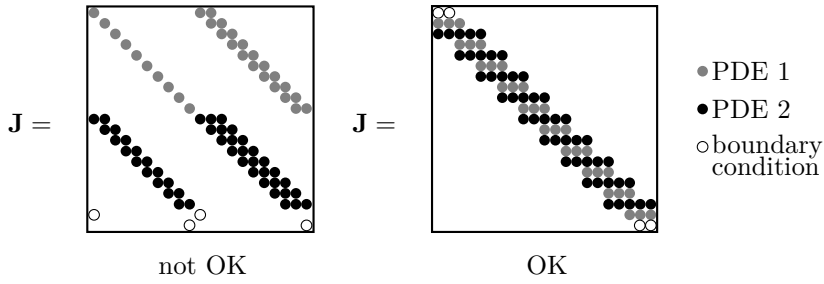


Figure 3.12: Sparsity pattern of the Jacobian for the nonlinear system (Eqs. (3.19)-(3.20)) for different ordering of the water levels and the discharges and of the nonlinear equations.

with $\mathbf{x}$ the unknown water levels and discharges.

Another important aspect is to take care how the variables for every channel are internally organized such that the Jacobian presents a well-defined structure. One possible approach is to store for a reach i first all the water levels followed by the discharges:

$$\mathbf{x}^{(i)} = \left[h_1^{(i)}, h_2^{(i)}, \ldots, h_{n_h^{(i)}}^{(i)}, Q_1^{(i)}, \ldots, Q_{n_Q^{(i)}}^{(i)} \right]^{\mathrm{T}}. \tag{3.22}$$

There is only a direct coupling via the discretized equations between the variables on neighbouring grid points. However by storing the variables as in Eq. (3.22) and sequentially handling the discretized equations of the first PDE, the discretized equations of the second PDE and the boundary conditions, this structure is not preserved in the matrix $\mathbf{J}$. This can be seen in the left plot in Fig. 3.12 where the sparsity pattern for $\mathbf{J}$ is shown. The gray and black dots represent elements different from zero of the first and second PDE respectively and the black circles represent elements different from zero associated with the upstream and downstream boundary conditions. As we can see the structure is totally destroyed. This is prevented by using a smarter ordering for the variables:

$$\mathbf{x}^{(i)} = \left[Q_1^{(i)}, h_1^{(i)}, Q_2^{(i)}, h_2^{(i)}, Q_3^{(i)}, \ldots, h_{n_h^{(i)}}^{(i)}, Q_{n_Q^{(i)}}^{(i)} \right]^{\mathrm{T}}.$$

Using this ordering in combination with an intelligent ordering of the Eqs. (3.19) and (3.20) and the boundary conditions, the structure is preserved and the matrix $\mathbf{J}$ has a banded diagonal structure with bandwidth three (see the right plot in Fig. 3.12).

3.5.2.3 Steady state solution

For most of the simulation results shown in this dissertation the initial condition of the river systems will be the steady state solution for a given set of initial discharges and upstream or downstream water levels for some of the reaches. In steady state, there is no change in the water levels or discharges:

$$\frac{\partial Q(z,t)}{\partial t} = 0,$$
$$\frac{\partial h(z,t)}{\partial t} = 0. \tag{3.23}$$

Using this information in the first PDE of the Saint-Venant equations (Eq. (3.1)) results in:

$$\frac{\partial Q(z,t)}{\partial z} = 0.$$

This means that in steady state the discharge along a reach is constant. The corresponding steady state water levels can be found from substituting Eq. (3.23) into the second PDE of the Saint-Venant equations (Eq. (3.2)):

$$\frac{\partial}{\partial z}\frac{Q_0^2}{A(z,t)} + gA(z,t)\left(\frac{\partial h(z,t)}{\partial z} + S_{\mathrm{f}}(z,t) - S_0\right) = 0 \tag{3.24}$$

This is actually an Ordinary Differential Equation (ODE) for a given constant value for the discharge Q_0 along the reach. This ODE can be solved if the water level at one grid point is known. This ODE can be rewritten as

$$\frac{\partial h(z,t)}{\partial z} = \pm\left(\frac{Q_0^2}{A(z,t)^2}\frac{\partial A}{\partial h} - gA(z,t)\right)^{-1} gA(z,t)(S_{\mathrm{f}}(z,t) - S_0)$$

where the plus sign needs to be used to calculate the water levels downstream of the given water level, and the minus sign for the water levels upstream of the given water level. The water levels can be found by integrating this ODE with e.g. a Runge-Kutta solver [58]. However the calculated steady state values are not necessarily the steady state values of the discretized Saint-Venant equations. Therefore the steady state values of the water levels are calculated by applying the same finite difference approximations used for discretizing the Saint-Venant equations to Eq. (3.24). This results in a system of nonlinear equations which can be solved with a Newton-Raphson method. The advantage of this approach is that the resulting water levels and discharges are steady state values of the numerical approximation of the PDEs.

3.5.3 Illustrative example: dam break problem

In this section the simulation results obtained with the implemented numerical solution method for a dam break problem will be discussed. This example is used to verify the correctness of the implementation, to illustrate the accuracy of the numerical scheme and as didactic example to illustrate the process dynamics of a river system. The simulation results for other test examples are given in Appendix B. There it is shown that the maximal difference between the numerical steady state solution and the analytic steady state solution for two test cases is less than 1.6 mm. A mass conservation test indicates that the fictitious amount of water added or removed during simulation is limited. Also the effect of a time-varying given upstream and/or downstream discharge as boundary conditions for a single reach is visualized. The last example consists of two reaches connected to each other with a gated weir. In this example the concept of "uncontrollability" related to hydraulic structures is explained. This concept will return in Chapter 4.

A problem extensively studied in the literature is the dam break problem. Initially the channel is in steady state with the water levels upstream of the dam considerably larger than the water levels downstream of the dam. At a certain time the dam breaks and the water flows from the upstream side of the dam to the downstream side. For some simple cases an analytical solution can be found [51,67]. Consider the example of a 4 km reach with a uniform rectangular cross section, with a bottom width of 1 m, no friction and a horizontal bed slope. The initial condition is defined as

$$h(0,z) = \begin{cases} h_{\mathrm{l}} & z \leq z_{\mathrm{dam}}, \\ h_{\mathrm{r}} & z > z_{\mathrm{dam}}, \end{cases}$$

with z_{dam} the position of the dam, while the flow is zero everywhere. For the values $h_{\mathrm{l}} = 8$ m, $h_{\mathrm{r}} = 2$ m and $z_{\mathrm{dam}} = 2000$ m, Fig. 3.13 shows the analytic solution together with the calculated solution at four different time instances. For the formulation of the analytic solution, the reader is referred to [67]. In spite of the large discontinuity in the initial condition, the formulated numerical scheme gives a good approximation of the analytic solution. Fig. 3.14 shows the evolution of the water levels after the dam break in space and time for the first 100 seconds.

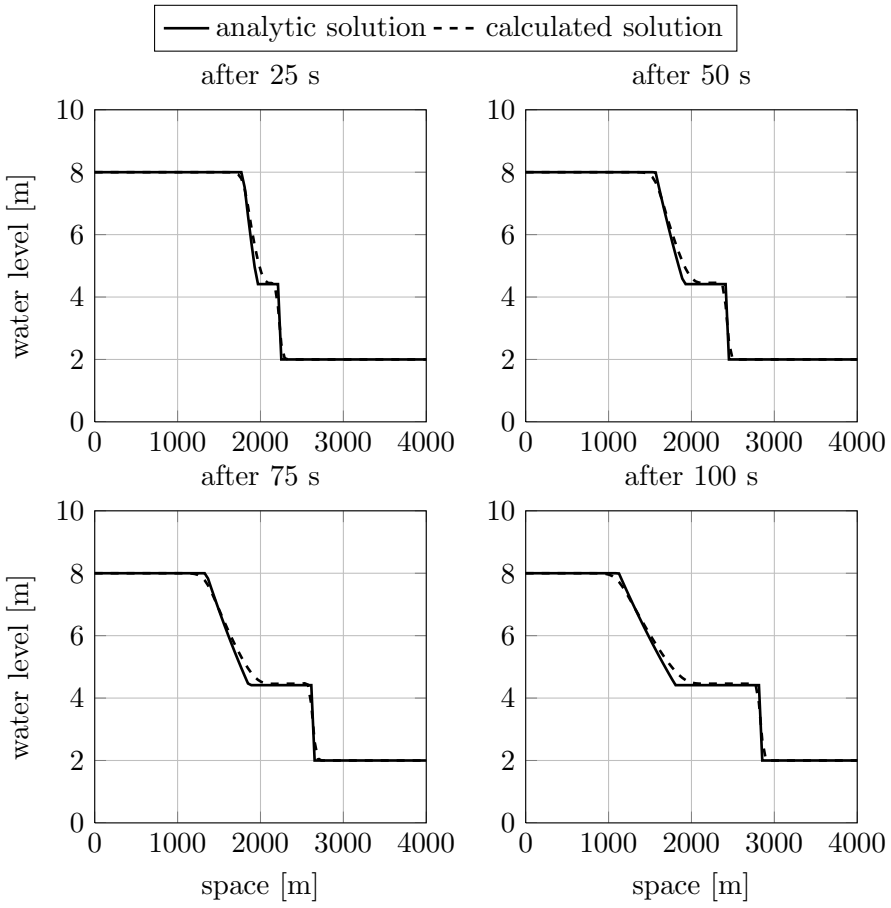


Figure 3.13: Comparison of the water levels of the analytic solution of the dam break example with the water levels calculated with the numerical simulator at four different time instants.

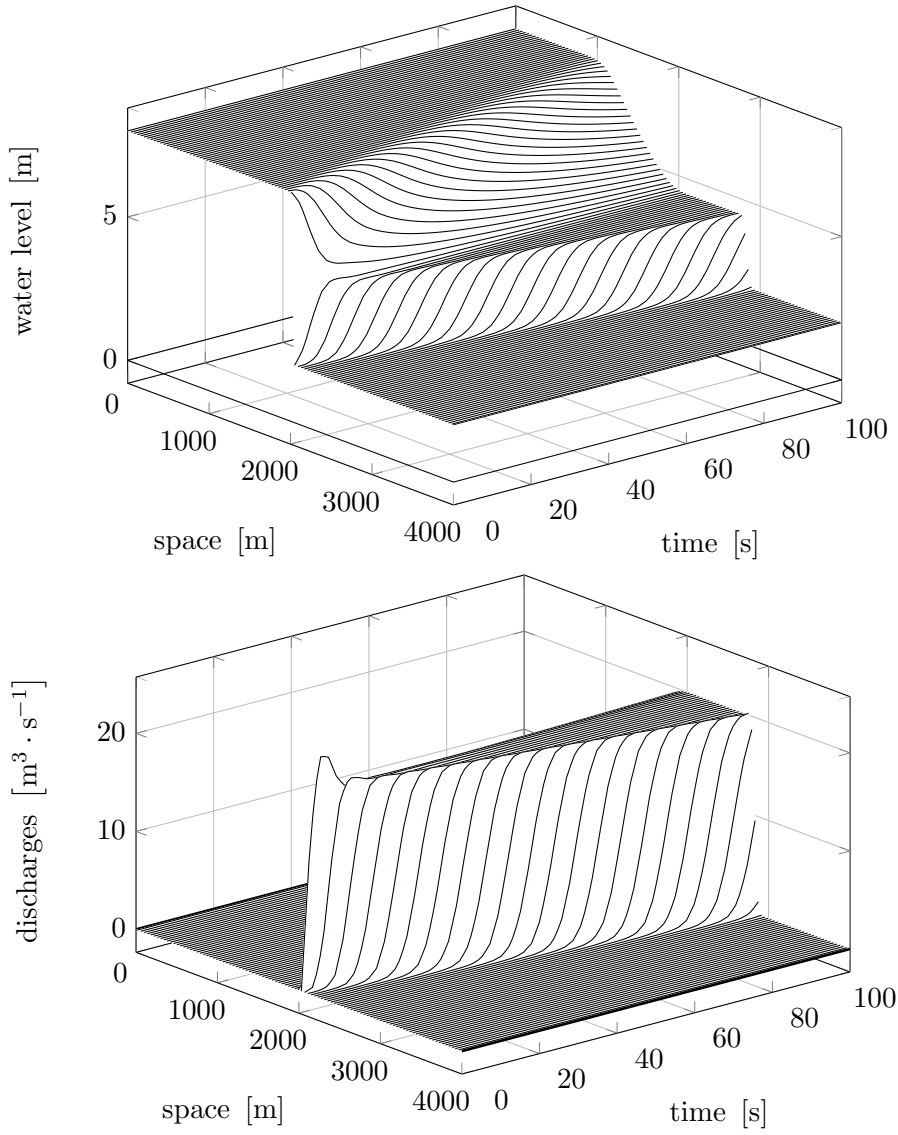


Figure 3.14: Simulation results for the test regarding the dam break problem of the numerical simulator ($n_h = 101$). The top plot shows the evolution of the water levels in space and time for the first 100 seconds. The plane on the bottom represents the bedding of the reach. The bottom plot shows the evolution of the discharges in space and time for the first 100 seconds. The bold lines correspond with the upstream and downstream boundary conditions. The upstream discharge and downstream discharge are kept constant at 0 $\mathrm{m}^3 \cdot \mathrm{s}^{-1}$.

3.6 Linear-Nonlinear model & other approximate models

One important aspect of MPC is the model used by the controller to predict the future behavior of the system. This model should provide a good mathematical characterization of the system dynamics but at the same time it should not be too complex (cfr. Occam's razor principle): the most simple model that approximates the system behavior and is suitable for its purpose, should be favored. After given a short review of the different types of approximate models found in the literature for channels and reaches, a new type of approximate model usable for flood control is proposed. This model is used throughout the next chapters in this dissertation.

3.6.1 Literature review

In general there are three types of models for approximating a complex system:

- A **white** box or physical model is a model based on first principles. E.g. based on the Newton equations one can model many different mechanical systems.
- In many applications it can be too time consuming to derive a white box model or the resulting white box model can be too complex. System identification algorithms can be used in this situation to build a model based on measurement or simulation data. In the domain of system identification one can find two types of models. **Black** box models are derived solely based on data and result in an empirical input-output relationship [107]. **Grey** box modelling is based on both insight into the system and experimental data. Some unknown free parameters can be estimated with system identification techniques.

All three types of models can be found in the literature for finding approximate models of river systems. White box models are based on the Saint-Venant equations to describe the dynamics of a single reach. Based on measured water levels and discharges, a black box model can be derived with system identification techniques. Integrator-delay models and reservoir models are examples of grey box models.

3.6.1.1 Finite-difference method applied to linearized Saint-Venant equations

Different works can be found in the literature where finite-difference approximations have been used for set-point control of irrigation canals or reaches [13, 115, 138]. These methods start by linearizing the Saint-Venant equations around a desired working point. These linearized equations are then discretized in space and time by using a finite difference method in order to generate discrete-time linear (white box) models. For a high number of grid points, these models provide a good approximation of the original nonlinear dynamics for small variations around their working points. However when gates are present, these approximations deteriorate fast for larger variations.

3.6.1.2 System identification techniques

Several applications of system identification techniques for finding approximate models for river systems can be found. E.g. the authors in [141] identified different black box model structures based on experimental data to approximate the dynamics of a canal pool. Linear and nonlinear grey box models were proposed in [62, 131, 186] where the parameters of these models were found with system identification techniques. Neural networks were used in [85, 194] to model rivers.

3.6.1.3 Integrator-delay model

The finite-difference approximations are often too complex if one is only interested in set-point control. To limit the computational burden for MPC, model approximations were derived. The most popular approximate model in literature for the purpose of set-point control is the integrator-delay model [148]. This model starts from the idea that the most common flow profile in controlled open-channels is the backwater curve and divides a channel into two parts: a part where the flow is uniform and a part that is affected by backwater. Fig. 3.15 visualizes both parts: the water depth at the downstream end is above the normal depth and decreases in the upstream direction until it reaches the upstream end of the reach or until the water depth equals the normal depth. Uniform flow is rarely present in open channels: geometrical irregularities of the bed slope and the cross sectional profiles, backwater effects, etc. cause the water depth to change over the distance [147]. However if the depth of flow varies within a few percentages of the normal depth, it may be designated to be uniform for practical purposes [46].

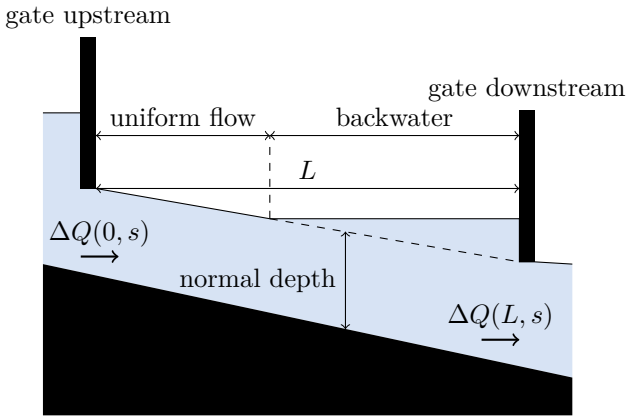


Figure 3.15: Example of a backwater curve and variables for the integrator-delay model (source: [147]).

The resulting integrator-delay model in continuous time is the following [147, 182]:

$$\frac{\mathrm{d}e}{\mathrm{d}t} = \frac{1}{A_\mathrm{s}}(\Delta Q(0, t-\tau) - \Delta Q(L, t))$$

with e the deviation of the downstream water level from its operating point [m], τ the time-delay of the normal depth portion of the channel [s], A_s the surface of the backwater part $\left[\mathrm{m}^2\right]$ and $\Delta Q(0, t)$ and $\Delta Q(0, L)$ the deviation of the upstream, resp. downstream discharge from their operating point. The time-delay τ and the surface area of the backwater part A_s are channel specific and need to be estimated. More information about the determination of the values of these parameters can be found in [147, 182]. Multiple examples can be found in literature where integrator-delay models are successfully used for set-point control of river systems [24, 48, 49, 102, 103, 105, 106, 150, 176, 178, 182, 184]. Although its successful application in many examples, the author in [182] pointed out that MPC based on an integrator-delay model for set-point control can result in a worse performance compared to more traditional control techniques due to the simplifications made for generating the model.

3.6.1.4 Reservoir model

The use of integrator-delay models is limited to set-point regulation since they are only accurate enough in a small region around the point for which they were calibrated. Because periods of heavy rainfall will cause very large deviations from the calibration point, these models are not usable for flood control. Also

they approximate the water level only at the end of every reach, and not along the entire reach. A model type successfully used for flood control by our research group [18–20, 37, 38] is the reservoir model type [174]. These models have the advantage that they are more accurate than integrator-delay models and are computationally less intensive than models based on finite-differences.

The first step in constructing a reservoir model for a river system consists of lumping the system in space. Different parts of the river are lumped into relevant locations. The number of these relevant locations is typically some orders of magnitude smaller than the order of the finite-difference models, which explains the computational advantage. These relevant locations are locations upstream and downstream of hydraulic structures and places where there is a high risk of potential flooding. These locations divide the river in a connected set of reaches. In each reach the water continuity is modelled based on the reservoir model type. The dynamics of such a reservoir are similar to the one described in Section 3.4 which corresponds with the conservation of mass: the change in volume of the reservoir is equal to the difference between the inflowing and outgoing discharges. By connecting all these small reservoirs, one can approximate a river system with a limited number of nodes.

This approach has been applied to simulate sewer and river networks [173,188]. In particular this procedure has been used to develop a conceptual model of the Demer by the Hydraulics Laboratory of the KU Leuven under the supervision of Prof. P. Willems and has successfully been used in combination with MPC in the PhD research of T. Barjas Blanco [18] and by the author [37, 38]. For more information about the calibrated conceptual model and the calibration process, the reader is referred to [18, 20, 44, 189].

3.6.2 Linear-Nonlinear model

Every model type discussed in the previous section has its own advantages and disadvantages. The finite-difference models have in general a high accuracy but their accuracy deteriorates quickly once the state of the river systems starts deviating from the working point. The black box models can be very efficient. However they need to be derived and calibrated for every new river system. It is also far from certain that the same identification algorithm can be used for different examples. Furthermore these models are only a good approximation of a river system for situations similar to their calibration data. That is why the usage of these models is limited to set-point control. The same limitation holds for the integrator-delay model because it models only the downstream water level of a reach. The reservoir models used to construct a conceptual model of a river system have already proven their strength: if well calibrated these

models can give an accurate approximation of the water levels and discharges of a river system. However this approach has some disadvantages:

- The water levels for an entire reach are often represented by only one water level. If the flood level of reaches are very irregular, this does not seem to be the optimal choice. For example, it is hard to guarantee that even if the modelled water level of the reach is below its flood level, the entire water profile of the reach is below the flood limit.
- Constructing, calibrating and validating these conceptual models is a very time-consuming process and requires expert knowledge to make good choices for the many parameters. Furthermore, every time one wants to apply e.g. MPC to another river system, this entire procedure has to be repeated.

Because in this work we want to find out whether MPC can be used for flood control in general, the approximate model should therefore also have some generality in its design. The conceptual model of the Demer is valid only for the Demer and hence does not directly lead to any general conclusion.

Therefore to be able to make more general conclusions, the approximate model used in this work is based directly on the Saint-Venant equations. This model should accurately approximate the water levels and discharges of a river system for a wide range of variables such that it can be used for flood control. The approximate model proposed in this section is based on a linearization of the discretized Saint-Venant equations but combines this with the nonlinear gate equations and is called the "Linear-Nonlinear" model or in short the LN-model. This model will now be discussed in more detail.

3.6.2.1 Single reach

As mentioned earlier the dynamics of a single reach will be approximated with a linearization of the discretized Saint-Venant equations (Eq. (3.21)). Here, the procedure will be explained for the situation where the boundary conditions are given upstream and downstream discharges of the reach. Linearizing these discretized Saint-Venant equations together with the boundary conditions around a nominal operating point ($\mathbf{h}_{\text{ss}} \in \mathbb{R}^{n_h}$ for the water levels, $\mathbf{q}_{\text{ss}} \in \mathbb{R}^{n_Q}$ for the discharges and $\mathbf{u}_{\text{ss}} = [Q_{\text{in,ss}}, Q_{\text{out,ss}}]^{\text{T}} \in \mathbb{R}^2$ for the discharges at the boundaries), the following linear state space model can be found:

$$\begin{bmatrix} \Delta\mathbf{h}(k+1) \\ \Delta\mathbf{q}(k+1) \end{bmatrix} = \mathbf{A} \begin{bmatrix} \Delta\mathbf{h}(k) \\ \Delta\mathbf{q}(k) \end{bmatrix} + \mathbf{B}_1 \Delta\mathbf{u}(k) + \mathbf{B}_2 \Delta\mathbf{u}(k+1) \tag{3.25}$$

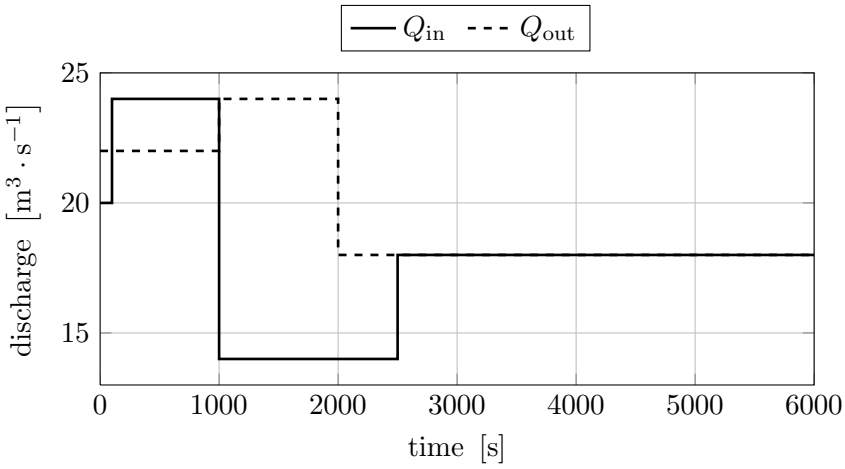


Figure 3.16: Evolution of the inflowing discharge Q_{in} and outgoing discharge Q_{out} used to evaluate the quality of the linear approximation of the Saint-Venant equations in a single reach.

with $\Delta\mathbf{h}(k) = \mathbf{h}(k) - \mathbf{h}_{\text{ss}}$, $\Delta\mathbf{q}(k) = \mathbf{q}(k) - \mathbf{q}_{\text{ss}}$, $\Delta\mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}_{\text{ss}}$, $\mathbf{A} \in \mathbb{R}^{(n_h+n_Q)\times(n_h+n_Q)}$, $\mathbf{B}_1 \in \mathbb{R}^{(n_h+n_Q)\times 2}$ and $\mathbf{B}_2 \in \mathbb{R}^{(n_h+n_Q)\times 2}$. Note that $\Delta\mathbf{u}(k+1)$ is needed to predict $\Delta\mathbf{h}(k+1)$ and $\Delta\mathbf{q}(k+1)$ since we work with a linearized version of an implicit nonlinear scheme. For a single reach this does not cause a problem since the boundary discharges are assumed to be given.

The quality of this type of approximate models is tested on a channel with the following characteristics:

- the length L of the channel is 4000 m,
- the channel has a bed slope S_0 of 0.001,
- it has a Manning roughness coefficient of n_{mann} 0.040,
- the channel cross sectional profile has a trapezoidal shape with a bottom width B of 6.1 m and side slopes S_1 and S_2 of 0.5.

Initially the channel is in steady state with a downstream water level of 7 m and an initial flow of 20 $\text{m}^3 \cdot \text{s}^{-1}$. Fig. 3.16 shows the evolution of the upstream discharge Q_{in} and the downstream discharge Q_{out}. Fig. 3.17 visualizes the water level profile (left) and discharge profile (right) of the reach at different time instants calculated with the Saint-Venant equations and its linear approximation. Both have used the same value $n_h = 201$ for discretizing the spatial domain. The approximate profiles form a good approximation

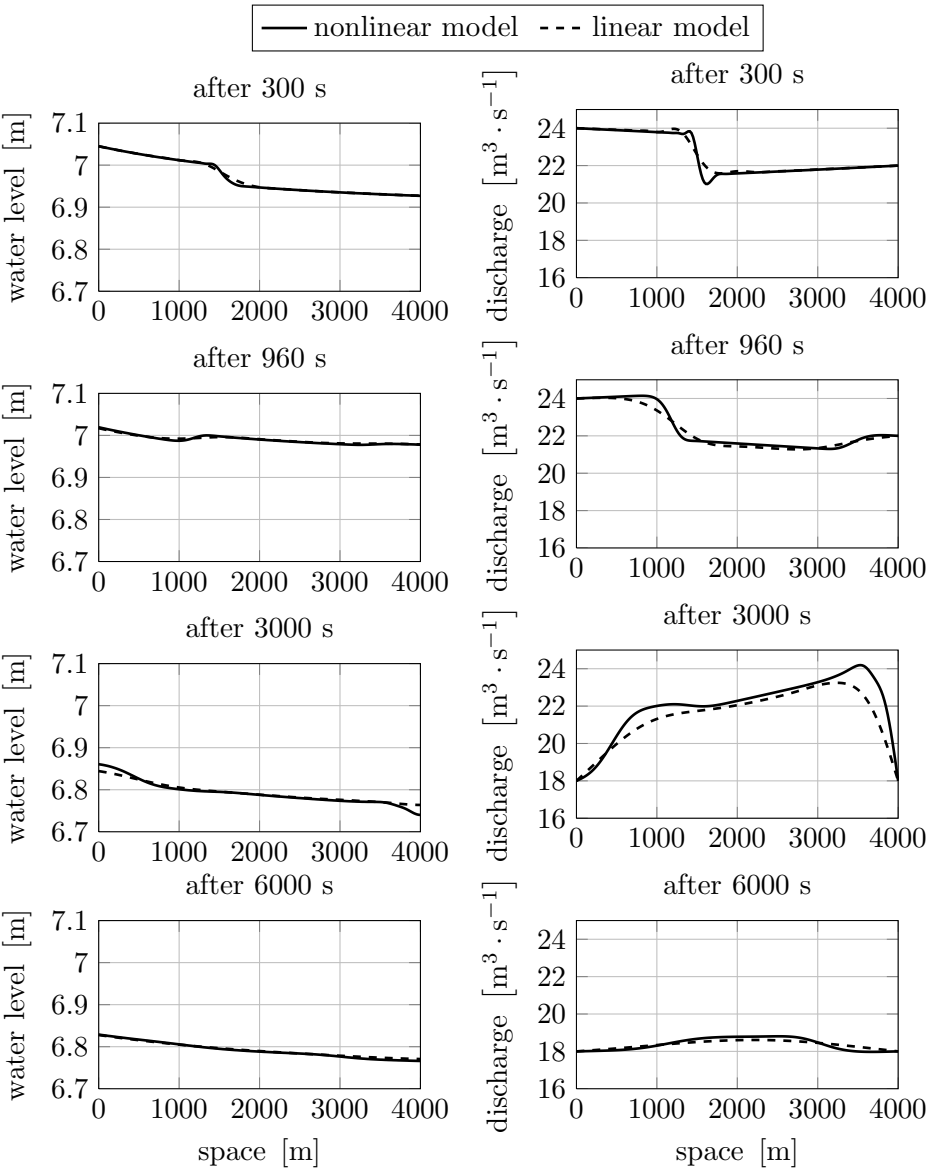


Figure 3.17: The water level profile (left column of plots) and discharge profiles (right column of plots) calculated with the approximate linear model and the Saint-Venant equations after 300 s, 960 s, 3000 s and 6000 s for a single reach.

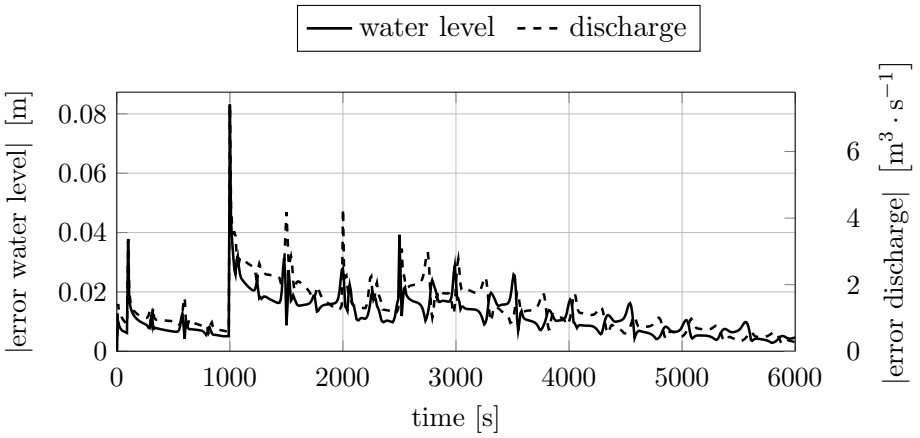


Figure 3.18: Maximum absolute error for the water levels and the discharges between the Saint-Venant equations (nonlinear model) and their linear approximation.

of the real profiles. This can also be concluded from Fig. 3.18. This plot shows the maximum value of the absolute difference between the water levels, resp. the discharges calculated with the nonlinear simulator and with the linear approximate model at every time step. The larger the step change in any of the discharges at the boundary, the larger the error will be. However this error decreases rapidly in time. Even though the value of the discharges at the boundaries at the end of the simulation differ from their linearization point, the steady state error on the water levels and discharges is very small.

3.6.2.2 Multiple reaches

The approach followed in this part is very similar to the approach introduced in the following publication:

[34] Breckpot, M., Agudelo, O. M., and De Moor, B. Modelling of a river system with multiple reaches. In *Proc. of the 16th IFAC Symposium on System Identification* (Brussels, Belgium, July 2012), pp. 1454-1459.

The main difference is the complexity of the example. The publication worked with channels with a rectangular cross sectional profile while here the approximate model is tested on channels with a trapezoidal cross sectional shape.

Linear model

The dynamics of a single reach can be accurately approximated with a linear state space model if the upstream and downstream discharges are given. One could use the same approximation for a river system consisting of a series of multiple reaches with hydraulic structures in between. Linearizing the Saint-Venant equations together with the gate equations and the boundary conditions around a nominal operation point ($\mathbf{h}_{\text{ss}} \in \mathbb{R}^{n_h}$ for the water levels with $n_h = n_h^{(1)} + n_h^{(2)} + \ldots + n_h^{(n_c)}$ the total number of water levels, n_c the number of channels, $\mathbf{q}_{\text{ss}} \in \mathbb{R}^{n_Q}$ for the discharges with $n_Q = n_Q^{(1)} + n_Q^{(2)} + \ldots + n_Q^{(n_c)}$ the total number of discharges and $\mathbf{u}_{\text{ss}} \in \mathbb{R}^{n_u}$ for the inputs with n_u the number of inputs), the following linear state space model can be found:

$$\begin{bmatrix} \Delta\mathbf{h}(k+1) \\ \Delta\mathbf{q}(k+1) \end{bmatrix} = \bar{\mathbf{A}} \begin{bmatrix} \Delta\mathbf{h}(k) \\ \Delta\mathbf{q}(k) \end{bmatrix} + \bar{\mathbf{B}}_1 \Delta\mathbf{u}(k) + \bar{\mathbf{B}}_2 \Delta\mathbf{u}(k+1) \tag{3.26}$$

with $\bar{\mathbf{A}} \in \mathbb{R}^{(n_h+n_Q)\times(n_h+n_Q)}$, $\bar{\mathbf{B}}_1 \in \mathbb{R}^{(n_h+n_Q)\times n_u}$ and $\bar{\mathbf{B}}_2 \in \mathbb{R}^{(n_h+n_Q)\times n_u}$. The input vector $\mathbf{u}$ contains the upstream and downstream river discharges together with the gate positions: $\mathbf{u} = \left[Q_{\text{up}}, c^{(1)}, c^{(2)}, \ldots, c^{(n_u-2)}, Q_{\text{down}}\right]^{\text{T}}$. Again $\Delta\mathbf{u}(k+1)$ is needed to predict $\Delta\mathbf{h}(k+1)$ and $\Delta\mathbf{q}(k+1)$. This is not a problem because in general one knows what the gate position will be at time instant t_{k+1}.

As will be shown, the performance of this linear model degrades if the input values do not stay close to their linearization point. Because a good performance was achieved with this model for a single reach, the addition of hydraulic structures is responsible for this performance degradation if the gates do not stay close to their nominal operation point.

Linear-Nonlinear model

A solution for this problem is to work with a combination of the linearized version of the Saint-Venant equations of each reach together with the nonlinear gate equations. This means that the effect of the gates on the water levels and the discharges is "pulled" out of the linear model. From now on this system will be referred to as the LN-model (Linear-Nonlinear model). The linear part of the LN-model has the following form:

$$\begin{bmatrix} \Delta\mathbf{h}(k+1) \\ \Delta\mathbf{q}(k+1) \end{bmatrix} = \mathbf{A} \begin{bmatrix} \Delta\mathbf{h}(k) \\ \Delta\mathbf{q}(k) \end{bmatrix} + \mathbf{B}_1 \Delta\mathbf{u}(k) + \mathbf{B}_2 \Delta\mathbf{u}(k+1)$$

with $\mathbf{A} \in \mathbb{R}^{(n_h+n_Q)\times(n_h+n_Q)}$, $\mathbf{B}_1 \in \mathbb{R}^{(n_h+n_Q)\times n_u}$ and $\mathbf{B}_2 \in \mathbb{R}^{(n_h+n_Q)\times n_u}$. Every reach has now a given discharge as boundary condition for the upstream and downstream end. The state space matrices can be found by linearizing the

Algorithm 1 Simulation of the LN-model.

for $k = 1, \ldots$ **do**
  $\mathbf{h}(k+1) = \Delta\mathbf{h}(k) + \mathbf{h}_{\text{ss}}$
  for $i = 1, \ldots$ until convergence/max number of iterations **do**
    for each gate j **do**
      select $h_{\text{up}}^{(j)}$ and $h_{\text{down}}^{(j)}$ out of $\mathbf{h}(k+1)$
      use gate equation to calculate $Q_{\text{gate}}^{(j)}(k+1)$ given $c^{(j)}(k+1)$, $h_{\text{up}}^{(j)}$ and $h_{\text{down}}^{(j)}$ and build up $\Delta\mathbf{u}(k+1)$
    end for
    $$\begin{bmatrix} \Delta\mathbf{h}(k+1) \\ \Delta\mathbf{q}(k+1) \end{bmatrix} = \mathbf{A} \begin{bmatrix} \Delta\mathbf{h}(k) \\ \Delta\mathbf{q}(k) \end{bmatrix} + \mathbf{B}_1\Delta\mathbf{u}(k) + \mathbf{B}_2\Delta\mathbf{u}(k+1)$$
  end for
end for

Saint-Venant equations for each reach in combination with these new boundary conditions. The main difference with Eq. (3.26) is that $\mathbf{u}$ now contains the gate discharges instead of the gate openings: $\mathbf{u} = \left[Q_{\text{up}}, Q_{\text{gate}}^{(1)}, \ldots, Q_{\text{gate}}^{(n_u-2)}, Q_{\text{down}}\right]^{\text{T}}$. Hence, the only extra work is to make the transformation from the gate openings $\mathbf{c}(k) = \left[c^{(1)}, \ldots, c^{(n_u-2)}\right]^{\text{T}}$ and $\mathbf{c}(k+1)$ to the discharges over the gates $\mathbf{q}_{\text{gate}}(k) = \left[Q_{\text{gate}}^{(1)}, \ldots, Q_{\text{gate}}^{(n_u-2)}\right]^{\text{T}}$ and $\mathbf{q}_{\text{gate}}(k+1)$. At time step k the current upstream and downstream water levels for each hydraulic structure are known. Given these water levels together with the current gate opening $\mathbf{c}(k)$, $\mathbf{q}_{\text{gate}}(k)$ can be computed from the gate equations as defined in Section 3.3.3. The only difficulty lies in making the transformation from $\mathbf{c}(k+1)$ to $\mathbf{q}_{\text{gate}}(k+1)$ since the water levels $\mathbf{h}(k+1)$ are unknown. A solution is to work in an iterative way (see Algorithm 1). The water levels $\mathbf{h}(k)$ are used in combination with $\mathbf{c}(k+1)$ to estimate $\mathbf{q}_{\text{gate}}(k+1)$. With this estimate of $\mathbf{u}(k+1)$ we can calculate $\mathbf{h}(k+1)$ and $\mathbf{q}(k+1)$. $\mathbf{q}_{\text{gate}}(k+1)$ can now be refined by using $\mathbf{h}(k+1)$ and $\mathbf{c}(k+1)$, etc. The author does not claim that this algorithm has a theoretical proof of convergence. However simulation results for different types of channels indicate that this algorithm can be used in practice.

Reduced order LN-model

Notice that the LN-model has the same (high) number of states as the nonlinear model. An extra gain in computation time can be obtained by reducing the number of states while still achieving accurate results (see e.g. [59]). One method successfully used in many applications is Proper Orthogonal Decomposition (POD). POD is a data-driven method where a suitable set of ordered orthonormal basis vectors are derived from simulation or experimental

data. Reduced order models are typically found by projecting (Galerkin projection) the full-order models on the most relevant basis vectors ([10], [137]).

POD and Galerkin projection were introduced in Section 2.5. Based on the snapshot matrices $\tilde{\mathbf{X}} \in \mathbb{R}^{n_h \times M}$ for the water levels and $\hat{\mathbf{X}} \in \mathbb{R}^{n_Q \times M}$ for the discharges of M samples of simulation data obtained with the linear part of the LN-model, POD is used to find a $\tilde{n}_h$th order approximation of $\Delta\mathbf{h}(k)$ and a $\hat{n}_Q$th order approximation of $\Delta\mathbf{q}(k)$:

$$\Delta\mathbf{h}(k) \approx \Delta\mathbf{h}_{\tilde{n}_h}(k) = \sum_{j=1}^{\tilde{n}_h} a_j(k)\tilde{\boldsymbol{\varphi}}_j = \tilde{\boldsymbol{\Phi}}_{\tilde{n}_h}\mathbf{a}(k),\ \tilde{n}_h \ll n_h,$$

$$\Delta\mathbf{q}(k) \approx \Delta\mathbf{q}_{\hat{n}_Q}(k) = \sum_{j=1}^{\hat{n}_Q} b_j(k)\hat{\boldsymbol{\varphi}}_j = \hat{\boldsymbol{\Phi}}_{\hat{n}_Q}\mathbf{b}(k),\ \hat{n}_Q \ll n_Q,$$

where $\tilde{\boldsymbol{\varphi}}_j \in \mathbb{R}^{n_h}$, for $j = 1, \ldots, \tilde{n}_h$, and $\hat{\boldsymbol{\varphi}}_j \in \mathbb{R}^{n_Q}$, for $j = 1, \ldots, \hat{n}_Q$, are the $\tilde{n}_h$ and $\hat{n}_Q$ most relevant orthonormal basis vectors, $\tilde{\boldsymbol{\Phi}}_{\tilde{n}_h} = \left[\tilde{\boldsymbol{\varphi}}_1, \ldots, \tilde{\boldsymbol{\varphi}}_{\tilde{n}_h}\right] \in \mathbb{R}^{n_h \times \tilde{n}_h}$, $\mathbf{a}(k) = [a_1(k), \ldots, a_{\tilde{n}_h}(k)]^{\mathrm{T}} \in \mathbb{R}^{\tilde{n}_h}$, $\hat{\boldsymbol{\Phi}}_{\hat{n}_Q} = \left[\hat{\boldsymbol{\varphi}}_1, \ldots, \hat{\boldsymbol{\varphi}}_{\hat{n}_Q}\right] \in \mathbb{R}^{n_Q \times \hat{n}_Q}$ and $\mathbf{b}(k) = \left[b_1(k), \ldots, b_{\hat{n}_Q}(k)\right]^{\mathrm{T}} \in \mathbb{R}^{\hat{n}_Q}$. Combining this result with the Galerkin projection for the linear part of the LN-model results in the following reduced order model:

$$\begin{bmatrix} \mathbf{a}(k+1) \\ \mathbf{b}(k+1) \end{bmatrix} = \mathbf{A}_{\mathrm{r}} \begin{bmatrix} \mathbf{a}(k) \\ \mathbf{b}(k) \end{bmatrix} + \mathbf{B}_{1,\mathrm{r}}\Delta\mathbf{u}(k) + \mathbf{B}_{2,\mathrm{r}}\Delta\mathbf{u}(k+1)$$

with $\mathbf{A}_{\mathrm{r}} = \boldsymbol{\Phi}_n^{\mathrm{T}}\mathbf{A}\boldsymbol{\Phi}_n \in \mathbb{R}^{(\tilde{n}_h+\hat{n}_Q)\times(\tilde{n}_h+\hat{n}_Q)}$, $\mathbf{B}_{1,\mathrm{r}} = \boldsymbol{\Phi}_n^{\mathrm{T}}\mathbf{B}_1 \in \mathbb{R}^{(\tilde{n}_h+\hat{n}_Q)\times n_u}$ and $\mathbf{B}_{2,\mathrm{r}} = \boldsymbol{\Phi}_n^{\mathrm{T}}\mathbf{B}_2 \in \mathbb{R}^{(\tilde{n}_h+\hat{n}_Q)\times n_u}$. The matrix $\boldsymbol{\Phi}_n$ is constructed from the $\tilde{n}_h$ and the $\hat{n}_Q$ most relevant orthonormal basis vectors $\tilde{\boldsymbol{\Phi}}_{\tilde{n}_h}$ and $\hat{\boldsymbol{\Phi}}_{\hat{n}_Q}$ respectively:

$$\boldsymbol{\Phi}_n = \begin{bmatrix} \tilde{\boldsymbol{\Phi}}_{\tilde{n}_h} & \\ & \hat{\boldsymbol{\Phi}}_{\hat{n}_Q} \end{bmatrix}.$$

More information on the derivation of the POD basis vectors and a criterion to determine a value for $\tilde{n}_h$ and $\hat{n}_Q$ are given in Section 2.5.

The steps for simulating the POD model are described by Algorithm 2.

Test example

The linear model, the LN-model as well as the reduced order LN-model will be tested to approximate a river system consisting of two reaches connected to each other through an underflow vertical sluice. Both reaches are identical and have the following parameters:

Algorithm 2 Simulation of the POD based LN-model

build the snapshot matrices $\tilde{\mathbf{X}}$ and $\hat{\mathbf{X}}$
use SVD to find $\tilde{\mathbf{\Phi}}_{n_\mathrm{h}}$ and $\hat{\mathbf{\Phi}}_{n_\mathrm{Q}}$ and construct $\mathbf{\Phi}_n$

$$\begin{bmatrix} \mathbf{a}(0) \\ \mathbf{b}(0) \end{bmatrix} = \mathbf{\Phi}_n \cdot \left(\begin{bmatrix} \mathbf{h}(0) \\ \mathbf{q}(0) \end{bmatrix} - \begin{bmatrix} \mathbf{h}_\mathrm{ss} \\ \mathbf{q}_\mathrm{ss} \end{bmatrix} \right)$$

for $k = 1, \ldots$ **do**
  $\mathbf{h}_{\tilde{n}_h}(k+1) = \tilde{\mathbf{\Phi}}^\mathrm{T}_{\tilde{n}_h} \mathbf{a}(k) + \mathbf{h}_\mathrm{ss}$
  for $i = 1, \ldots$ until convergence/max number of iterations **do**
    for each gate j **do**
      select $h^{(j)}_\mathrm{up}$ and $h^{(j)}_\mathrm{down}$ out of $\mathbf{h}_{\tilde{n}_h}(k+1)$
      use gate equation to calculate $Q^{(j)}_\mathrm{gate}(k+1)$ given $c^{(j)}(k+1)$, $h^{(j)}_\mathrm{up}$ and $h^{(j)}_\mathrm{down}$ and build up $\Delta\mathbf{u}(k+1)$
    end for

$$\begin{bmatrix} \mathbf{a}(k+1) \\ \mathbf{b}(k+1) \end{bmatrix} = \mathbf{A}_\mathrm{r} \begin{bmatrix} \mathbf{a}(k) \\ \mathbf{b}(k) \end{bmatrix} + \mathbf{B}_{1,\mathrm{r}} \Delta\mathbf{u}(k) + \mathbf{B}_{2,\mathrm{r}} \Delta\mathbf{u}(k+1)$$

  $\mathbf{h}_{\tilde{n}_h}(k+1) = \hat{\mathbf{\Phi}}^\mathrm{T}_{\tilde{n}_h} \mathbf{a}(k+1) + \mathbf{h}_\mathrm{ss}$
  end for
end for

- a length L of 4 km,
- the cross-sectional profiles have a bottom width B of 6.1 m and side slopes S_1 and S_2 of 0.5,
- a bed slope S_0 of 0.001
- and a Manning roughness coefficient n_mann of 0.040.

The underflow vertical sluice is modelled with the equations given in Section 3.3.3.1 and has the same width as the bottom width of the channels. The number of grid points per reach was set to $n_h = 201$. The POD basis vectors of the reduced order LN-model are derived from the system response of the linearized reach dynamics when applying step changes to the boundary conditions of each reach (1000 samples were gathered). Based on the energy criterion introduced in Section 2.5, a decision can be made to select the $\tilde{n}_h$ and $\hat{n}_Q$ most relevant POD basis vectors. Fig. 3.19 shows the plot of $1 - \bar{P}_{\tilde{n}_h}$ and $1 - \bar{P}_{\hat{n}_Q}$ for the first 200 basis vectors. Fig. 3.20 and Fig. 3.21 show the basis vectors $\tilde{\boldsymbol{\varphi}}_j$, resp. $\hat{\boldsymbol{\varphi}}_j$ associated with the 10 largest singular values. The first half of each basis vector is associated with the normalized deviations of the water levels/discharges of the first reach, while the second half belongs to the deviations for the second reach. Based on a truncation degree of $1 - \bar{P}_{\tilde{n}_h} = 2.83 \cdot 10^{-4}$ ($\bar{P}_{\tilde{n}_h} = 0.9997$) and $1 - \bar{P}_{\hat{n}_Q} = 1.89 \cdot 10^{-4}$ ($\bar{P}_{\hat{n}_Q} = 0.9998$),

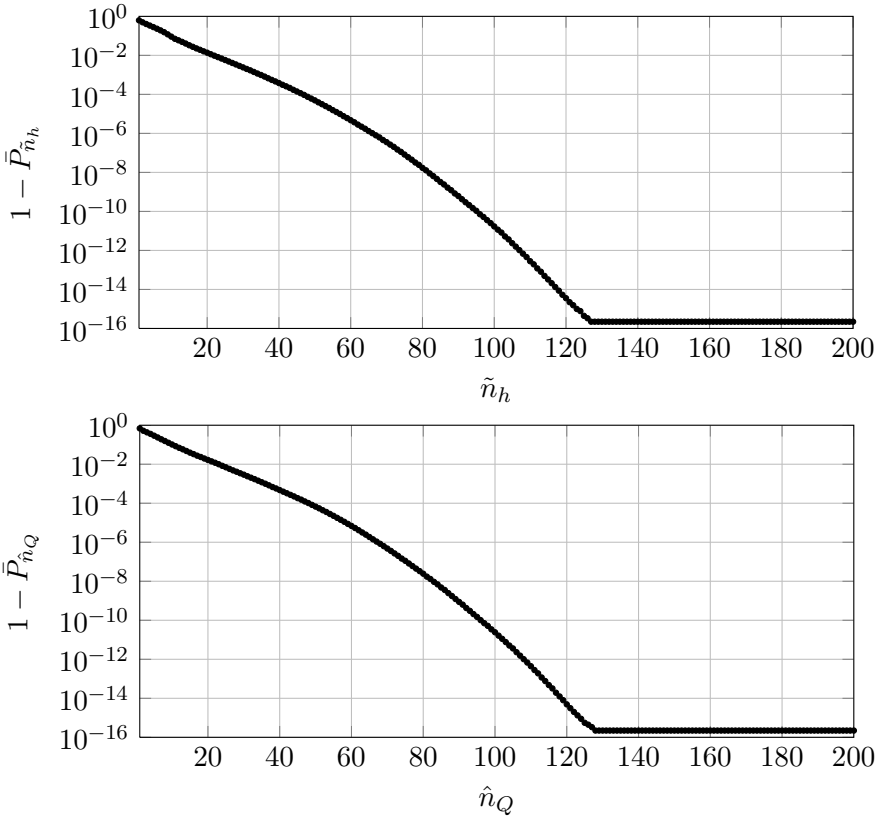


Figure 3.19: Logarithmic plot of $1 - \bar{P}_{\tilde{n}_h}$ (top) and $1 - \bar{P}_{\hat{n}_Q}$ (bottom) for determining the truncation degree of the POD basis vectors.

the first $\tilde{n}_h = 41$ POD basis vectors were selected for the water levels and the first $\hat{n}_Q = 43$ POD basis vectors were selected for the discharges. This means a reduction from 806 states to 83 states.

Fig. 3.22 shows the evolution of the upstream and downstream discharges together with the gate position. Fig. 3.23 visualizes the water level profile (left) and discharge profile (right) of both reaches at different time instants calculated with the full Saint-Venant equations and the three approximate models. Fig. 3.24 shows the maximal absolute error on the water levels and the discharges at every time step for the three approximate models. The performance of the linear model is worse than the performance of the LN-model and the POD based LN-model. The result after 10 000 seconds show that there is a significant steady state error in the water levels for the linear model. This is not the case for the LN-models. The figure also shows that

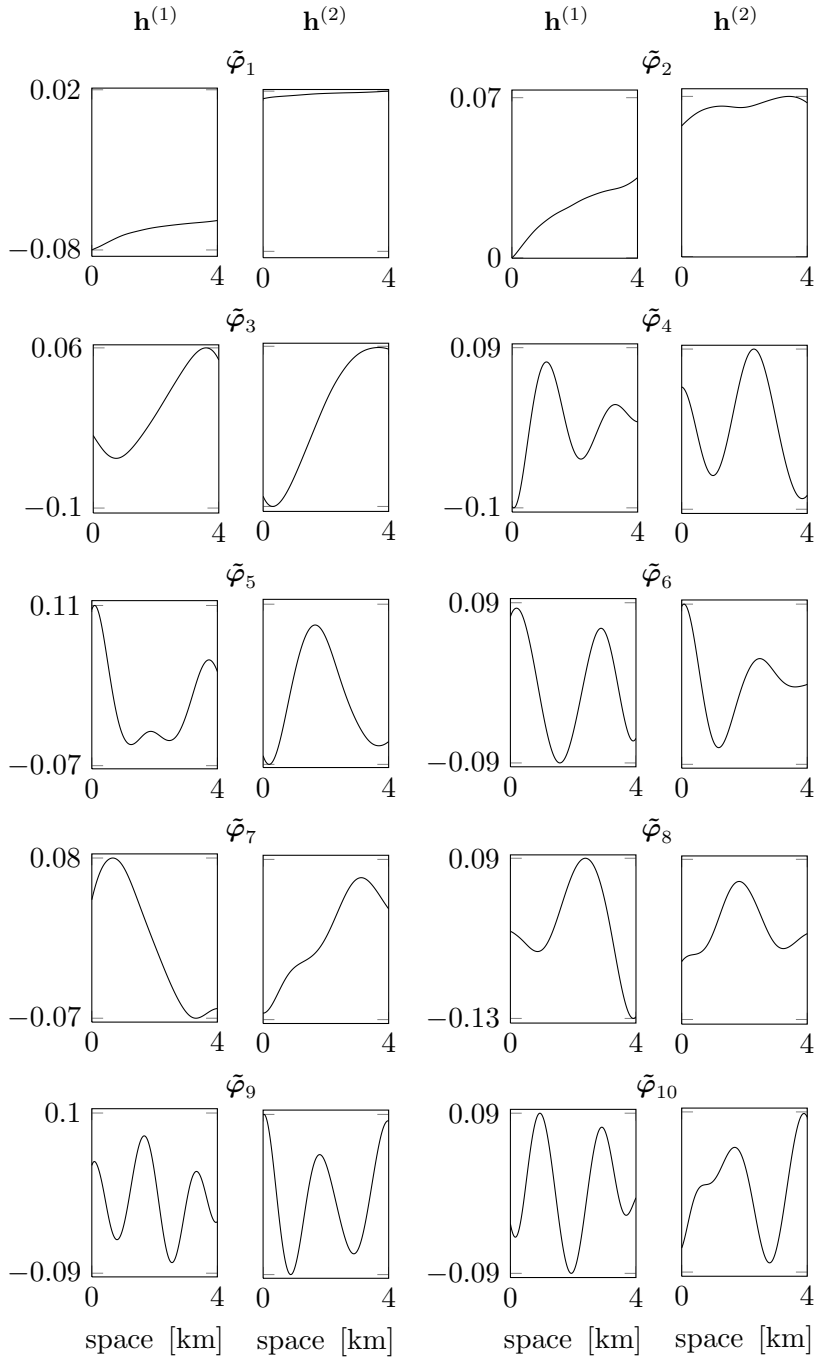


Figure 3.20: First 10 POD basis vectors for the water levels: $\tilde{\boldsymbol{\varphi}}_1$, $\tilde{\boldsymbol{\varphi}}_2$, ... $\tilde{\boldsymbol{\varphi}}_{10}$

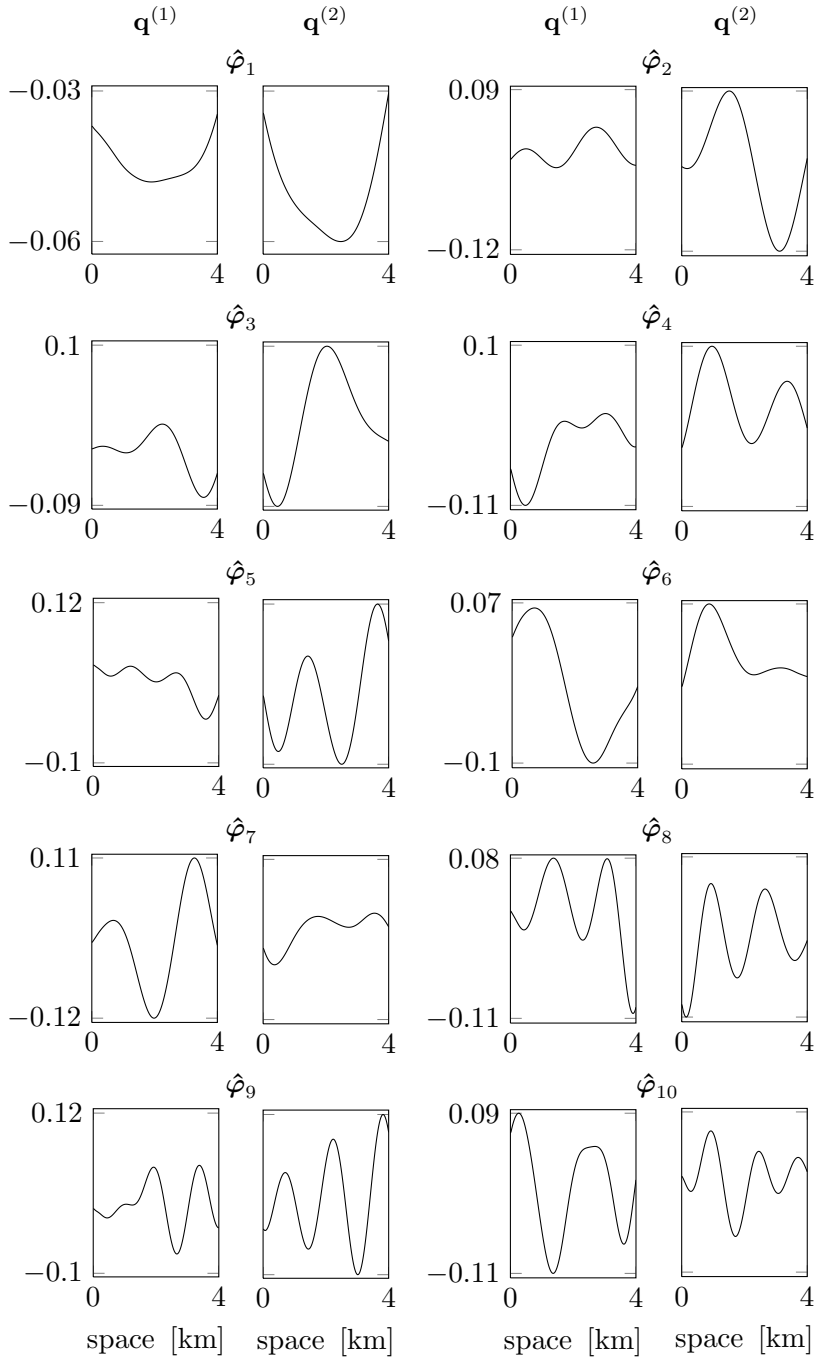


Figure 3.21: First 10 POD basis vectors for the discharges: $\hat{\boldsymbol{\varphi}}_1$, $\hat{\boldsymbol{\varphi}}_2$, $\ldots$ $\hat{\boldsymbol{\varphi}}_{10}$

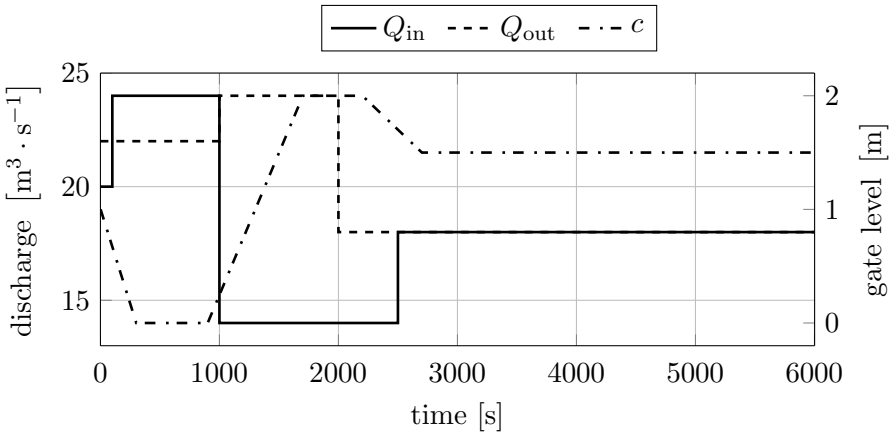


Figure 3.22: Evolution of the inflowing discharge Q_{in}, the outgoing discharge Q_{out} and the gate position c used to evaluate the performance of the approximate models of the Saint-Venant equations.

Table 3.2: The RMSE for the three approximate models with respect to the nonlinear model for the water levels and the discharges.

	RMSE_h [m]	RMSE_Q $[\text{m}^3 \cdot \text{s}^{-1}]$
linear model	0.0842	1.9743
LN-model	0.0095	0.5009
reduced order LN-model	0.0103	0.5511

the model reduction with POD has almost no influence on the simulated water levels or discharges: there is no visual distinction between the profile of the water levels and the discharges of the LN-model and the POD based LN-model for the four time instants. If necessary, a higher accuracy for the water levels or the discharges can always be achieved by taking more basis vectors into account for modelling these variables. Table 3.2 gives the Root-Mean-Square Error (RMSE) for the simulation results of the linear model, the LN-model and the reduced order LN-model with respect to the nonlinear model for the water levels and the discharges. The RMSE for the water levels is defined as

$$\text{RMSE}_h = \sqrt{\frac{\sum_{j=1}^{n_h} \sum_{k=1}^{T_{\text{N}}} \left(\hat{h}(z_j, t_k) - h(z_j, t_k) \right)^2}{T_{\text{N}} \cdot n_h}}$$

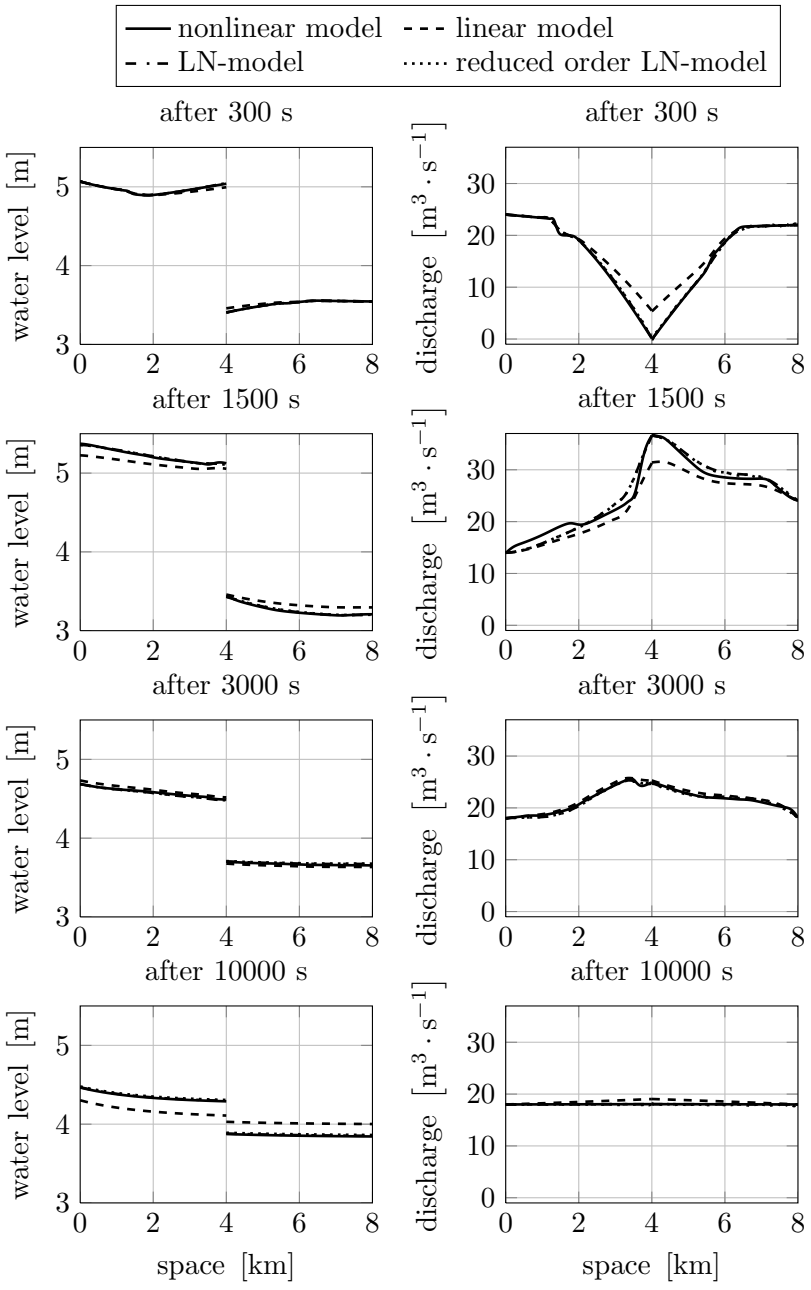


Figure 3.23: The water level (left column of plots) and discharge (right column of plots) profiles calculated with the linear model, the LN-model, the reduced order LN-model and the Saint-Venant equations after 300 s, 960 s, 3000 s and 6000 s for two reaches connected to each other through an underflow vertical sluice.

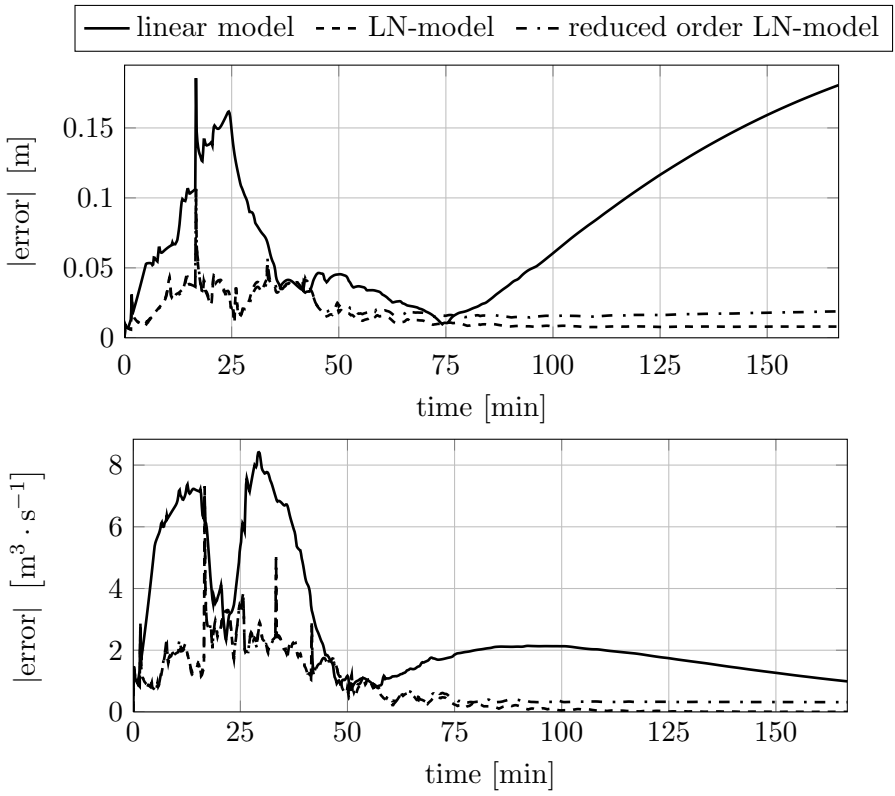


Figure 3.24: Maximum absolute error for the water levels (top) and the discharges (bottom) between the approximate models and the Saint-Venant equations.

with T_N the total number of time steps, h the results obtained with the nonlinear model and $\hat{h}$ the results obtained with the linear model, the LN-model or the reduced order LN-model. A similar formula is used for the discharges. The RMSE for the water levels and for the discharges lead to the same conclusions: the RMSE is much larger for the results obtained with the linear model as for the results obtained with the LN-model, while the effect of the model reduction with POD on the results is limited.

Table 3.3 gives the average total time for 20 runs needed to perform the simulation for the nonlinear model, the linear model, the LN-model and the reduced order LN-model for a simulation time of 10 000 seconds. The computation time for the three approximate models are significantly lower than the one of the nonlinear model. The effect of including the nonlinear

Table 3.3: The average simulation time for 20 runs for the four different models with a PC with a 2.8 GHz Intel Core i7 and 8 GB of RAM.

	simulation time [s]
nonlinear model	59.03
linear model	2.95
LN-model	3.84
reduced order LN-model	0.71

gate equations into the LN-model has almost no effect on the simulation time compared with the linear model. The time needed to calculate the gate discharge for a gate position is ignorable. An extra reduction in computation time is achieved by the model reduction technique. However the speed-up is not as big as the reduction of states because of the loss of structure in the system matrices. Although the matrices $\mathbf{A}$, $\mathbf{B}_1$ and $\mathbf{B}_2$ are very sparse, their sparsity is destroyed by the multiplications with $\mathbf{\Phi}_n^{\mathrm{T}}$ and $\mathbf{\Phi}_n$.

3.7 Conclusions

In general, we have shown how to construct accurate mathematical models of river systems. The dynamics of the water levels and discharges along a single reach are modelled with the Saint-Venant equations. These are two partial differential equations modelling the conservation of mass and momentum. One particular element in these equations is the resistance law used to model the boundary friction of the channel. We have explained how these resistance laws can be used when channels have different roughness coefficients along its cross-sectional profile. This will be important when a model is constructed for the Demer in Chapter 5.

Different types of boundary conditions were used with the Saint-Venant equations such that we can construct mathematical models of complex river systems. The most important examples are the boundary conditions determined by the presence of hydraulic structures and junctions. It is also explained how reservoirs can be modelled. Based on these concepts, mathematical models of many real life examples of river systems can be constructed. A discretization scheme is introduced to solve these mathematical models. A staggered grid structure is chosen in combination with an upwinding approach. Multiple examples have been used to assess the performance of this numerical scheme.

The last part of this chapter dealt with finding computationally inexpensive but accurate approximate models. Because we are interested in the application of flood control, these models need to give accurate approximations for a wide range of water levels and discharges. Furthermore it is highly recommended for flood control that the approximate models give an approximation of the water level profile along the entire length of every reach. However this is not the case for many approximate models found in the literature (based on system identification techniques, integrator-delay models, reservoir type models, ...). A new type of approximate model is presented in this chapter which is called the **Linear-Nonlinear model** (LN-model). It uses a linear version of the Saint-Venant equations to approximate the water levels and discharges along every reach, while using the nonlinear gate equations for calculating the discharges over the gates. The simulations have shown that this hybrid model provides accurate results and it requires less computational power than the one needed by the original nonlinear model. For one test example the RMSE for the water levels is only 0.0095 m while the RMSE for the discharges is limited to 0.5009 $m^3 \cdot s^{-1}$. The total simulation is decreased from 59.03 s to 3.84 s. This computational gain can be further increased by using model order reduction techniques. Proper Orthogonal Decomposition in combination with Galerkin projection has been used to reduce the order of the linear part of the LN-model. Simulation results showed an extra speed-up with only a limited effect on the accuracy of the model: the total computation time is further reduced to 0.71 s, while the RMSE for the water levels and the discharges increase to only 0.0103 m, respectively 0.5511 $m^3 \cdot s^{-1}$.

4

Model Predictive Control Used for Set-point Control and Flood Control

4.1 Introduction

The previous chapter explains how a dynamic model can be built for a river system consisting of multiple reaches, hydraulic structures, junctions and reservoirs. This chapter describes how Model Predictive Control (MPC) can be used for set-point control and flood control at the same time for such river systems. During no or little rainfall the controller should keep the most important water levels close to their set-points without using the buffer capacity of the reservoirs. When there is a risk of flooding, the controller should automatically start using the water reservoirs to prevent the river from flooding. Afterwards the controller should empty the water reservoirs as fast as possible.

In this chapter we will explain in detail how all these different objectives can be fulfilled with MPC. There are two key concepts. The first one is the approximate model used by the controller. By working with the LN-model introduced in the previous chapter and considering the gate discharges as control variables instead of the gate positions, we avoid working with an approximation of the highly nonlinear gate equations. The second important concept is the formulation of the optimization problem such that all different control objectives can be achieved. Along the chapter, this optimization problem will gradually be built up starting with a simple example of controlling

a single reach, and ending with the control of a complex river system consisting of multiple reaches, gates, junctions and a reservoir. Also the influence of a Kalman filter as state estimator on the control performances will be assessed. In practice only a limited number of water levels is measured. The Kalman filter is used to get an estimate of the full state based on this small amount of measurements. All closed loop simulations are performed with the full hydrodynamic nonlinear model of the different test systems.

This chapter is organized as follows. In Section 4.2 we explain why we propose to use MPC as control strategy for controlling river systems. A comparison is made with existing controllers found in the literature and the advantages of Model Predictive Control with respect to these existing techniques are underlined. MPC is applied in Section 4.3 to a single reach. The optimization problem is formulated such that the controller can track a time-varying reference trajectory as well as prevent the reach from flooding. By working with slack variables it is ensured that the optimization problem is feasible at all times. Its performance will be compared with a classical controller, a Linear Quadratic Regulator. Section 4.4 expands the proposed controller for a single reach to a system consisting of two reaches and a gate. This section explains the advantage of working with gate discharges as control variables and how this affects the optimization problem. Also the problem of possible uncontrollability of a gate is addressed and a solution is given. Section 4.5 shows how MPC can be used to control a more complex river system with six channels, three gates, two junctions and one reservoir. The optimization problem is further expanded such that the buffer capacity of the reservoir is used and recovered in an optimal way. Attention is paid in Section 4.6 to the implementation aspects of the controller. It is explained how the time needed to solve the optimization problem at every iteration can be decreased by reducing the number of optimization variables and the number of equality and inequality constraints. Section 4.7 summarizes the main conclusions of this chapter.

The main sections of this chapter are based on the following publications:

[32] Breckpot, M., Agudelo, O. M., and De Moor, B. Control of a single reach with model predictive control. In *River Flow 2012* (Costa Rica, September 2012), R. E. Murillo Muñoz, Ed., Proc. of the International Conference on Fluvial Hydraulics, CRC Press, Taylor & Francis Group, pp. 1021-1028.

[33] Breckpot, M., Agudelo, O. M., and De Moor, B. Model Predictive Control of a river system with two reaches. In *Proc. of the 51st IEEE Conference on Decision and Control* (Maui, Hawaii, USA, December 2012), pp. 4549-4554.

[36] Breckpot, M., Agudelo, O. M., and De Moor, B. Flood control with Model Predictive Control for river systems with water reservoirs. *Journal of Irrigation and Drainage Engineering 139*, 7 (July 2013), 532-541.

4.2 Why using Model Predictive Control?

4.2.1 Literature review

Many different types of control strategies can be found in the literature that address the control of reaches, irrigation canals or river systems. A classification of the control strategies can be performed based on general control theory and leads to the following classes [18, 176]:

- feedback and feedforward control,
- optimal control,
- heuristic control,
- three-position control
- and Model Predictive Control.

4.2.1.1 Feedback and feedforward control

The objective of a feedback or a feedforward controller is to keep a water level as close as possible to a given target level. For the case of feedback control, this water level is measured at every sampling time and compared to its set-point. Based on this deviation the feedback controller determines its control action. This feedback controller is in practice often a Proportional Integral controller (PI-controller) [119, 142]. If the PI-controller is well tuned, a good performance for set-point control can be achieved. Examples where feedback control is successfully applied can be found in [24, 106, 147, 149, 150, 180, 183].

In feedforward control measurements or predictions of a disturbance are used in combination with an inverse model of the effect of this disturbance on the water level to calculate the required control actions. In theory this would result in a zero water deviation from the set-point [22]. However, due to model inaccuracies and uncertainties on the model parameters as well as on the measurements, this deviation will never be zero in practice. To compensate for these model imperfections and uncertainties, a feedforward controller is often used in combination with a feedback controller [175].

4.2.1.2 Optimal control

Most of the optimal controllers used in water systems are based on the Linear Quadratic Regulator (LQR) theory [93], which is presented in Section 2.2. These controllers are called optimal controllers because the control actions are based on the minimization of an objective function. This objective function is a relative weighting between the sum of the square of the deviations between the water levels and their set-points and the sum of the square of the change in the structure settings. The higher the penalty on the water level deviation, the faster the control will be. Increasing the weighting factor on the control actions, the smoother the operation of the structures will be. Successful applications can be found in [47, 48, 116, 118, 120, 138, 139].

4.2.1.3 Heuristic control

Unlike the first two deterministic control methods, heuristic controllers are not based on physical laws but on a more heuristic approach. This group can further be divided into control based on rules-of-thumb, neural networks control [169,170], fuzzy logic control [161,162,181] and genetic algorithm control [185]. One drawback is that these controllers need to be tuned on a complete simulation model or on the real system because there exists no mathematical formalism for this. Another drawback is that the dynamic behavior of water systems is seen as a black box. These heuristic controllers are not frequently applied in practice.

4.2.1.4 Three-position controller

A three-position controller is a combination of a feedback controller and a heuristic controller [142]. It is used in practice for set-point control and it consists of a set of simple if-then-else rules for increasing or decreasing the gate position based on the deviation of the water level from its set-point. These controllers have the advantage that they can easily be implemented in practice and they require only a limited number of measurements. This type of controller is installed at the Demer by the VMM (Flemish Environment Agency). As we will show in Chapter 5, this controller will however not result in a good flood control performance, even if these simple rules are replaced with more advanced ones based on the experience of operators. More information about this controller can be found in Section 5.3.

4.2.1.5 Model Predictive Control

The general philosophy behind Model Predictive Control (MPC) was explained in Chapter 2. MPC is a combination of feedback control, feedforward control and optimal control. It is an optimal controller because it solves an optimization problem in order to find the optimal control actions. Feedback is introduced by solving this optimization problem at every time step and considering measurements and estimates of the current water levels and discharges. Feedforward control is achieved by taking predictions of future rainfall into account inside the optimization problem. Successful applications of MPC on water systems can be found in [144, 176, 177, 179, 182, 193].

4.2.2 Advantages of Model Predictive Control

Any of the first four examples of controllers applied to river systems discussed in the previous part have their own disadvantages regarding the problem of flood control. Feedback and feedforward controllers determine the control actions only based on measurements of water levels. A feedback controller does not take the effect of future rainfall into account and both feedback and feedforward controllers are not aware of flood levels. Optimal controllers based on the LQR theory suffer from the drawback that constraints on the inputs and/or states are not taken into account. Heuristic controllers have the drawback to be too site specific and are hardly used in practice even for set-point control. The if-then-else rules of a standard three-position controller are too simple for flood control since they only focus on set-point control of one specific water level. Also the advanced three-position controller used by the VMM for flood prevention has its limitations because it does not take rain predictions into account and the control actions are only based on local information.

MPC does not suffer from these drawbacks. In comparison to these different type of controllers, MPC is suitable for flood control because of the following reasons:

- MPC can take all types of constraints that are present in river systems into account. Examples are the physical lower and upper limits on gate positions as well as the maximal rate of change constraints on the gate movement, which can be incorporated inside the optimization problem as inequality constraints. The flood levels can be included in the optimization problem as upper limits on the water levels.
- River systems are typically highly interactive multiple-input-multiple-output systems. For such a system it is very difficult for traditional controllers to find good control actions based on only local information.

MPC however makes use of a mathematical model of the river system in the optimization problem to predict the effect of the control actions on all future water levels over a given prediction window. This allows MPC to find control actions that are not just locally optimal but optimal for the global river system.

- The use of a process model also allows MPC to take rain predictions into account. The effect of future rainfall on the water levels and discharges can be incorporated in this process model. Using this information in combination with the flood levels allows the controller to take effective actions in favor of flood prevention.
- MPC can be used at the same time for flood control as well as for set-point control. By minimizing the deviation of the water levels with respect to their references, MPC will try to keep these water levels as close as possible to their set-point when there is no risk of violating the flood levels.

One disadvantage in general of MPC is that it has to solve an on-line optimization problem at every time step. Since solving an optimization problem can be computationally intensive, MPC can only be used if the time needed to solve the optimization problem is smaller than the sampling time of the controller. Given that river systems have relatively slow dynamics, this sampling time is typically big enough.

As it was already mentioned, there exists many studies where MPC is applied to river systems. However most of these works focus on set-point control and not on flood control. In [168] MPC is used for flood prevention. However this work neglects the nonlinear behavior of hydraulic structures which is a big oversimplification of the problem.

In previous work done by our research group a conceptual model was used to model the Demer [18–20, 37, 38, 44, 189]. As mentioned in Section 3.6.2, this approach has some limitations:

- The water levels and discharges are modelled only at a very limited number of points. It is never certain that no floods are present along the entire river just because the water level modelled at such a point is not above its flood level.
- Building such a conceptual model is very time-consuming and requires expert knowledge during the derivation. The resulting model is only valid for that particular river system. It does not really allow you to make general conclusions about the proposed control scheme for other river systems.

To overcome these drawbacks, the MPC proposed in this chapter will work with the Linear-Nonlinear approximate model proposed in Section 3.6.2. In the remaining of this chapter, MPC schemes usable for flood control and set-point control of river systems will be presented.

4.3 Model Predictive Control applied to a single reach

This section explains how MPC can be used for set-point control and flood control of a single reach. The performance of the proposed control scheme will be compared with the performance of a more classical controller, the Linear Quadratic Regulator (LQR) which was introduced in Section 2.2. Also the influence of adding a Kalman filter for estimating the states based on a very limited number of measurements is assessed.

4.3.1 Mathematical model of a single reach

Based on the modelling concepts defined and explained in Section 3.2, a mathematical model of a single reach can be defined. It is assumed that the most upstream and most downstream discharge of the reach are given. They will be either determined by the controller or they are a disturbance signal. The resulting mathematical model describing the dynamics of a single reach is the following one:

$$Q(0,t) = Q_{\text{in}}(t), \tag{4.1}$$

$$\frac{\partial A(z,t)}{\partial h(z,t)}\frac{\partial h(z,t)}{\partial t} + \frac{\partial Q(z,t)}{\partial z} = 0, \tag{4.2}$$

$$\frac{\partial Q(z,t)}{\partial t} + \frac{\partial}{\partial z}\frac{Q(z,t)^2}{A(z,t)} + gA(z,t)\left(\frac{\partial h(z,t)}{\partial z} + S_{\text{f}}(z,t) - S_0\right) = 0, \tag{4.3}$$

$$Q(L,t) = Q_{\text{out}}(t), \tag{4.4}$$

where Q_{in} and Q_{out} are the given upstream, resp. downstream discharge. The friction slope S_{f} is modelled with the resistance law of Manning:

$$S_{\text{f}} = \frac{n_{\text{mann}}^2 Q|Q|}{A^2 R^{1/3}}.$$

4.3.2 Approximate model

When using MPC a mathematical model of the process is required inside the optimization problem. To keep this optimization problem as simple as possible, the controller will not work with Eqs. (4.1)-(4.4). A linear state space model will be used to approximate the dynamics of a single reach. Because we are interested in flood control, this model should give an approximation of the water levels and discharges along the entire reach. This linear state space model will be based on a linear version of the Saint-Venant equations with a fine spatial discretization. The quality of this model was assessed in Section 3.6.2. Linearizing the discretized version of the Eqs. (4.1)-(4.4) (by following the approach mentioned in Section 3.5.2) around a steady state point ($\mathbf{h}_{\text{ss}} \in \mathbb{R}^{n_h}$ for the n_h water levels, $\mathbf{q}_{\text{ss}} \in \mathbb{R}^{n_Q}$ for the n_Q discharges, $\mathbf{u}_{\text{ss}} \in \mathbb{R}^{n_u}$ for the n_u control inputs and $\mathbf{d}_{\text{ss}} \in \mathbb{R}^{n_d}$ for the n_d disturbances), results in the following model:

$$\Delta\mathbf{x}(k+1) = \mathbf{A}\Delta\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k) + \mathbf{F}\Delta\mathbf{d}(k), \tag{4.5}$$

with $\Delta\mathbf{x}(k) = \mathbf{x}(k) - \mathbf{x}_{\text{ss}}$, $\mathbf{x}(k) = [\mathbf{h}(k); \mathbf{q}(k)]$, $\mathbf{x}_{\text{ss}} = [\mathbf{h}_{\text{ss}}; \mathbf{q}_{\text{ss}}]$, $\Delta\mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}_{\text{ss}}$, $\Delta\mathbf{d}(k) = \mathbf{d}(k) - \mathbf{d}_{\text{ss}}$, $\mathbf{A} \in \mathbb{R}^{(n_h+n_Q)\times(n_h+n_Q)}$, $\mathbf{B} \in \mathbb{R}^{(n_h+n_Q)\times n_u}$ and $\mathbf{F} \in \mathbb{R}^{(n_h+n_Q)\times n_d}$. The input vector $\mathbf{u}(k)$ contains the controllable upstream and/or downstream river discharges at time instant k while the disturbance vector $\mathbf{d}(k)$ contains the uncontrollable ones. E.g. if only the upstream discharge can be controlled, then we have that $\mathbf{u}(k) = Q_{\text{in}}(t_k)$ and $\mathbf{d}(k) = Q_{\text{out}}(t_k)$.

Notice that the linear state space model in Eq. (4.5) differs slightly from the linear approximate model in Eq. (3.25). The matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ are combined into the matrix $\mathbf{B}$ and the term with $\Delta\mathbf{u}(k+1)$ is dropped. This approximation can be performed because the dynamics of river systems are much smaller than the sampling time of the controller. Also the matrix $\mathbf{F}$ is added to the equation to take the effect of the disturbance signals on the water levels and discharges into account.

Equation (4.5) can be rewritten as a function of the nominal values for $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{d}$:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{F}\mathbf{d}(k) + \boldsymbol{\beta}$$

with

$$\boldsymbol{\beta} = \mathbf{x}_{\text{ss}} - \mathbf{A}\mathbf{x}_{\text{ss}} - \mathbf{B}\mathbf{u}_{\text{ss}} - \mathbf{F}\mathbf{d}_{\text{ss}}.$$

4.3.3 Formulation of the optimization problem

At every time step, the controller solves following Quadratic Programming problem (QP):

$$\min_{\mathbf{u},\mathbf{x},\boldsymbol{\zeta}} \sum_{j=1}^{N_\mathrm{P}} \|\mathbf{x}(j) - \mathbf{r}_x(j)\|^2_{\mathbf{W}} + \sum_{j=0}^{N_\mathrm{P}-1} \|\mathbf{u}(j) - \mathbf{u}(j-1)\|^2_{\mathbf{R}} + \|\boldsymbol{\zeta}\|^2_{\mathbf{V}} + \mathbf{v}^\mathrm{T}\boldsymbol{\zeta}$$

$$\begin{aligned}
\text{s.t. } & \mathbf{x}(0) = \hat{\mathbf{x}}, \\
& \mathbf{x}(j+1) = \mathbf{A}\mathbf{x}(j) + \mathbf{B}\mathbf{u}(j) + \mathbf{F}\mathbf{d}(j) + \boldsymbol{\beta}, && j = 0, \ldots, N_\mathrm{P} - 1 \\
& \underline{\mathbf{u}} \leq \mathbf{u}(j) \leq \overline{\mathbf{u}}, && j = 0, \ldots, N_\mathrm{P} - 1 \\
& |\mathbf{u}(j) - \mathbf{u}(j-1)| \leq \boldsymbol{\Delta}_u, && j = 0, \ldots, N_\mathrm{P} - 1 \\
& \mathbf{u}(-1) = \mathbf{u}_\mathrm{prev}, \\
& \tilde{\mathbf{H}}\mathbf{x}(j) \leq \mathbf{h}_\mathrm{max} + \eta(j)\boldsymbol{\zeta}, && j = 1, \ldots, N_\mathrm{P} \qquad (4.6) \\
& \boldsymbol{\zeta} \geq 0, \qquad (4.7)
\end{aligned}$$

with N_P the prediction horizon, $\mathbf{W} \in \mathbb{R}^{(n_h+n_Q)\times(n_h+n_Q)} \succeq 0$, $\mathbf{R} \in \mathbb{R}^{n_u \times n_u} \succ 0$ and $\mathbf{V} \in \mathbb{R}^{n_h \times n_h} \succ 0$ three diagonal weighting matrices, $\mathbf{v} \in \mathbb{R}^{n_h}$ a weighting vector, $\mathbf{r}_x \in \mathbb{R}^{n_h+n_Q}$ a time-varying vector containing the set-points for all the states, $\hat{\mathbf{x}} \in \mathbb{R}^{n_x}$ the current state of the process, $\underline{\mathbf{u}} \in \mathbb{R}^{n_u}$ and $\overline{\mathbf{u}} \in \mathbb{R}^{n_u}$ the lower and upper limits on the inputs, $\boldsymbol{\Delta}_u \in \mathbb{R}^{n_u}$ a vector with the maximal allowed rate of change for the inputs, $\mathbf{u}_\mathrm{prev} \in \mathbb{R}^{n_u}$ the control action applied in the previous time step, $\tilde{\mathbf{H}} \in \mathbb{R}^{n_h \times (n_h+n_Q)}$ a matrix selecting the water levels from the state vector $\mathbf{x}$, $\mathbf{h}_\mathrm{max} \in \mathbb{R}^{n_h}$ the flood levels, $\eta(j) = 1/r_\mathrm{c}^{j-1}$ a time-dependent inverse weight (with $r_\mathrm{c} > 1$) and $\boldsymbol{\zeta} \in \mathbb{R}^{n_h}$ a vector of slack variables (one slack variable per water level). It can be shown that for positive semi-definite weighting matrices $\mathbf{W}$ and $\mathbf{V}$, and a positive definite weighting matrix $\mathbf{R}$, the QP has only one (global) solution [130]. In this chapter we use `quadprog` of the Optimization Toolbox of Matlab to solve the QP [123]. The reason to work with slack variables and with extra terms in the objective function as well as with weighting matrices and vectors will be explained in the next two sections.

4.3.3.1 The use of slack variables

During periods of heavy rainfall, the flood limits can become too restrictive and make the QP infeasible if they are imposed as hard constraints. This is avoided

by working with a slack variable approach with l_∞-norm and time-dependent inverse weights [74, 75]. The flood limits are implemented as soft constraints (Eq. (4.6)) by using the vector of slack variables $\boldsymbol{\zeta}$ and imposing a positivity constraint on the elements of this vector (Eq. (4.7)). To keep the violations of these limits as small as possible, the slack variables are penalized in the objective function by the terms $\|\boldsymbol{\zeta}\|_{\mathbf{V}}^2 + \mathbf{v}^{\mathrm{T}}\boldsymbol{\zeta}$. A sufficiently large $\mathbf{v}$ ensures that the constraints will only be violated when no feasible solution exists for the hard constrained optimization problem [74]. This means that the upper limits on the water levels are enforced as exact soft constraints. If the constraints cannot be prevented from being violated, the controller will minimize these violations and hence reduce the flood risk. The quadratic term $\|\boldsymbol{\zeta}\|_{\mathbf{V}}^2$ is included to have a well-posed QP and it is used as an additional tuning parameter [151]. A time-dependent inverse weight $\eta(j)$ is used to penalize future predicted constraint violations increasingly to avoid long-lasting constraint violations [74].

It is also possible to impose the flood limits as hard constraints. If the QP becomes infeasible for large disturbance signals, a solution can be to start dropping some of these constraints in such a way that the QP becomes feasible and new control actions can be found. In this situation one has to develop a strategy to remove some of these hard constraints in combination with a change of the values of the weighting matrices (in order to reduce possible violations of these constraints). This strategy was applied in [18]. However during periods when everything becomes critical, multiple QP's have to be formulated and solved. This is not needed when slack variables are used because they ensure the feasibility of the QP at all times.

Notice that we use only one vector of slack variables $\boldsymbol{\zeta}$ for the entire prediction horizon. Another possibility could be to use a time-varying vector $\boldsymbol{\zeta}(j)$ which can have different values at every time step in the prediction horizon. However this would increase the number of optimization variables with $n_h \cdot (N_{\mathrm{P}} - 1)$ extra variables. This increase can be avoided by working with the time-dependent inverse weight $\eta(j)$.

Remark. It should be noted that the use of the time-dependent inverse weight $\eta(j)$ can have a negative impact on the control performance. If it is not possible to prevent the river from flooding at the end of the prediction horizon, then the value of the corresponding entry of $\boldsymbol{\zeta}$ will be very large due to the small value of $\eta(N_{\mathrm{P}})$. This large value for the slack variable can give the controller the freedom to allow large flooding at the beginning of the prediction horizon that could have been avoided. This phenomenon did never occur for any of the tests performed in this work. However, if this would occur, the solution is to work with a time-varying vector of slack variables at the cost of an increase in the number of optimization variables.

4.3.3.2 Translating the control objectives to the weighting matrices and vector

The weighting matrices $\mathbf{W}$, $\mathbf{R}$ and $\mathbf{V}$ define the relative importance of the difference between the states and their set-points, the changes of the control actions and the vectors of slack variables $\boldsymbol{\zeta}$. To impose the flood levels as exact soft constraints, the diagonal elements of $\mathbf{V}$ and the entries of $\mathbf{v}$ should be sufficiently large compared to the diagonal elements of the other weighting matrices $\mathbf{W}$ and $\mathbf{R}$. This forces the controller to keep the violations of the flood limits as small as possible or if possible equal to zero when a large disturbance takes place (**flood control**). If there is no risk of flooding, the controller sets $\boldsymbol{\zeta}$ equal to zero and the third and the fourth term in the objective function are eliminated. The controller needs to focus on **set-point control** in this situation and start steering the different states towards their reference signal. This is achieved by setting the elements in the vector $\mathbf{r}_x$ associated with the water levels equal to the set-point of these water levels. The most important water levels get a higher weight in the matrix $\mathbf{W}$. The water levels that are less important get a smaller weight such that their deviation is allowed to increase if this could result in a lower deviation of the more important water levels. As we are only interested in set-point control of the water levels, the discharges get a lower weight. $\mathbf{R}$ influences the control effort of the different input variables. The higher the values in $\mathbf{R}$ the smaller the changes in the control actions will be. This means that the controller will react slower and less aggressive than a controller with smaller elements in $\mathbf{R}$. However a less aggressive controller will typically lead to less wear on the actuators and is more robust with respect to model errors. A trade-off between the reaction speed of the controller and the lifetime of the equipment has to be made.

4.3.3.3 Advantages of Model Predictive Control

The particular formulation of the optimization problem has some specific advantages compared to other control techniques:

- The flood levels are incorporated as upper limits on the water levels. Therefore the controller will try to avoid the violation of these upper limits and hence reduce the risk of flooding.
- Future rainfall is incorporated in the model equations by means of the matrix $\mathbf{F}$. If accurate estimates exist of the future rainfall $\mathbf{d}(k)$, this will increase the performance of the controller with respect to flood control.
- During periods of no or little rainfall, the controller will focus on set-point control because it will try to minimize the deviation of the states from

their set-points. It can even react on future set-point changes if they are known in advance.

4.3.4 Simulation results

A single reach will be controlled with LQR and MPC and their performance will be compared for three different test cases: set-point control, disturbance rejection and flood control. The reach has a trapezoidal cross section with side slopes S_1 and S_2 of 0.5 and a bottom width B equal to 4 m, it has a length L of 4000 m, the channel slope S_0 is equal to 0.0001 and the Manning coefficient n_{mann} is taken equal to $0.014\ \text{s}\cdot\text{m}^{-1/3}$. In all three test cases the reach is initially in steady state with a downstream water level equal to 3 m and a discharge of 1 $\text{m}^3 \cdot \text{s}^{-1}$ at every point along the reach. The linear state space model is derived based on this steady state condition. The reach is approximated by a grid structure with $n_h = 91$ water levels. The controllable discharges should always stay between -7 $\text{m}^3 \cdot \text{s}^{-1}$ ($\underline{\mathbf{u}}$) and 7 $\text{m}^3 \cdot \text{s}^{-1}$ ($\overline{\mathbf{u}}$) and have a maximal rate of change constraint of $\mathbf{\Delta}_u = 2\ \text{m}^3 \cdot \text{s}^{-1}$. The controllers work with a sampling time of 5 min, and the size of the prediction horizon N_{P} is 15. In all the three test cases we have assumed that we know at every time step the current state $\hat{\mathbf{x}}$ of the process and the disturbances along the prediction window. The control actions provided by both controllers will be applied to the full hydrodynamical model defined in Section 4.3.1.

The same values for the weighting matrices are used by the LQR and the MPC in the three test cases (LQR makes only use of the matrices $\mathbf{W}$ and $\mathbf{R}$). Table 4.1 gives the numerical values of the weight matrices $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{R}$, as well as the entries of the weight vector $\mathbf{v}$. Notice that the size of the matrix $\mathbf{R}$ depends on the number of inputs that can be controlled. For the test case of set-point control, it will be of size 2×2, while for the other two test cases it will be a scalar because Q_{in} is a disturbance signal. The weights are chosen in such a way that flooding will be prevented ($\mathbf{V}$, $\mathbf{v} \gg \mathbf{W}$, $\mathbf{R}$) and that the controllers will mainly focus on steering the most downstream water level towards its reference value, and if possible also the other water levels.

The control law used by LQR does not take the constraints on the control actions into account. In order to not violate these constraints, the control actions will be saturated after their calculation.

Following the approach discussed in [18], performance indicators will be used in order to quantify and compare the control performance of the different control strategies used in this dissertation. A first performance indicator is used to evaluate the cost with respect to the deviation of water levels from their set-

Table 4.1: Diagonal elements of the weight matrices $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{R}$, and elements of the weight vector $\mathbf{v}$ for set-point control, disturbance rejection and flood control of a single reach.

	reach
$\mathbf{W} \in \mathbb{R}^{163\times163}$	
water levels	$\begin{bmatrix} \mathbf{1}_{80} \\ 100 \end{bmatrix}$
discharges	$10^{-2} \cdot \mathbf{1}_{82}$
$\mathbf{V} \in \mathbb{R}^{81\times81}$	
flood levels	$10^{4} \cdot \mathbf{1}_{81}$
$\mathbf{v} \in \mathbb{R}^{81}$	
flood levels	$10^{4} \cdot \mathbf{1}_{81}$

	Q_{in}	Q_{out}
$\mathbf{R} \in \mathbb{R}^{2\times2}$ or $\mathbb{R}^{1\times1}$		
control actions	1	1

points:

$$J_h = \sum_{k=1}^{T_{\text{N}}} \|\mathbf{h}(k) - \mathbf{r}_h(k)\|_2,$$

with T_{N} the length of the simulation and $\mathbf{r}_h \in \mathbb{R}^{n_h}$ the time-varying set-points for the water levels. This performance indicator will be called the **set-point deviation cost**. The set-point deviation cost can also be calculated for one specific water level based on its deviation from its set-point. A second performance indicator is used to indicate how well the controller can prevent the river from flooding and is called the **flood cost**:

$$J_{h_{\max}} = \sum_{k=1}^{T_{\text{N}}} \|\max(\mathbf{h}(k) - \mathbf{h}_{\max}, \mathbf{0}_{n_h})\|_2.$$

Only the water levels larger than their corresponding flood levels result in an increase for the cost $J_{h_{\max}}$.

4.3.4.1 Set-point control

The ability to track a time-varying reference trajectory for the water levels is tested for both controllers. There is no risk of flooding ($\mathbf{h}_{\max}$ is set to ∞) and the controllers can control both the upstream and the downstream discharges. There are no disturbances.

Table 4.2: The set-point deviation cost J_h obtained with LQR and MPC when tracking a time-varying set-point for a single reach for the set-point control test case.

Simulation cost	MPC	LQR
J_h [m]	920	2117

Fig. 4.1 shows the evolution of the water levels in space and time relative to the bottom of the reach for MPC and LQR. The reference trajectory for all the water levels is shown by the thick lines. The references have a step change after 3 000 s and 13 000 s. The applied control actions together with their upper and lower limits for both controllers can be seen in Fig. 4.2.

Both controllers succeed in tracking the reference trajectory. However because MPC sees the step changes in the reference trajectory earlier due to its prediction capabilities, it reaches the new set-point much earlier. LQR only reacts when the step change effectively takes place. Both controllers satisfy the limits on the inputs, LQR by using a saturator and MPC because it incorporates these limits in its optimization problem. Notice that not only the downstream water level is following its reference point, but actually all the water levels are doing this. This is possible because the controllers can control both the upstream and the downstream discharge. A quantitative comparison between both controllers can be made by calculating the set-point deviation cost J_h. Table 4.2 contains the cost J_h based on the closed loop simulation results with MPC, respectively LQR. The cost J_h for LQR largely exceeds the cost for MPC because of the predictive power of MPC: MPC can react on future set-point changes and keeps therefore the deviation of the water levels from their set-points overall smaller than the deviation obtained with LQR.

4.3.4.2 Disturbance rejection

This test case checks how the controllers react when a disturbance takes place. The goal is to keep the most downstream water level as close as possible to its steady state value. The controllers can now only change the downstream discharge Q_{out}, the upstream discharge Q_{in} is the disturbance signal. After 6 000 s the upstream discharge jumps from 1 $\text{m}^3 \cdot \text{s}^{-1}$ to 3 $\text{m}^3 \cdot \text{s}^{-1}$ (e.g. a gate upstream is opened). Because we only want to compare the performance of both controllers for disturbance rejection, the flood levels $\mathbf{h}_{\max}$ are set to ∞ such that they will not influence the control actions of the MPC.

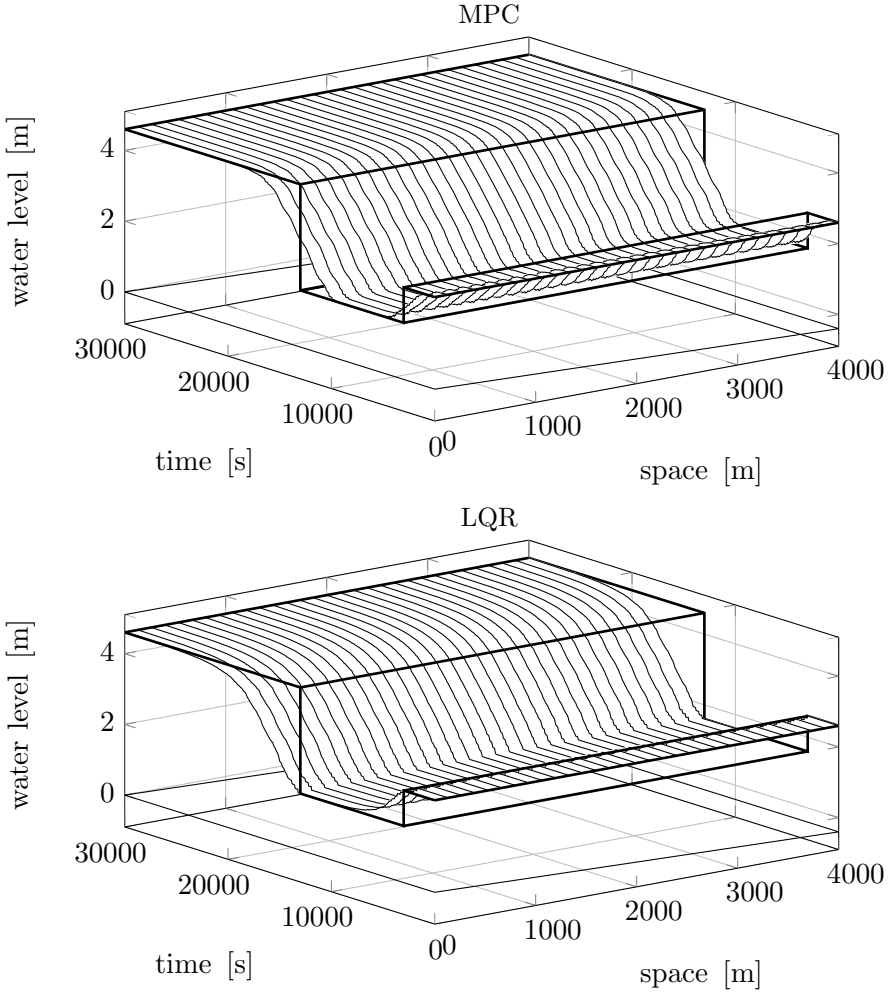


Figure 4.1: Evolution of the water levels in space and time together with the time-varying reference trajectory for a single reach controlled by means of MPC (top) and LQR (bottom). Both controllers succeed in tracking the changing reference trajectory. MPC reacts in advance to a change in the reference trajectory because of its predictive capabilities while LQR reacts only when the change effectively takes place.

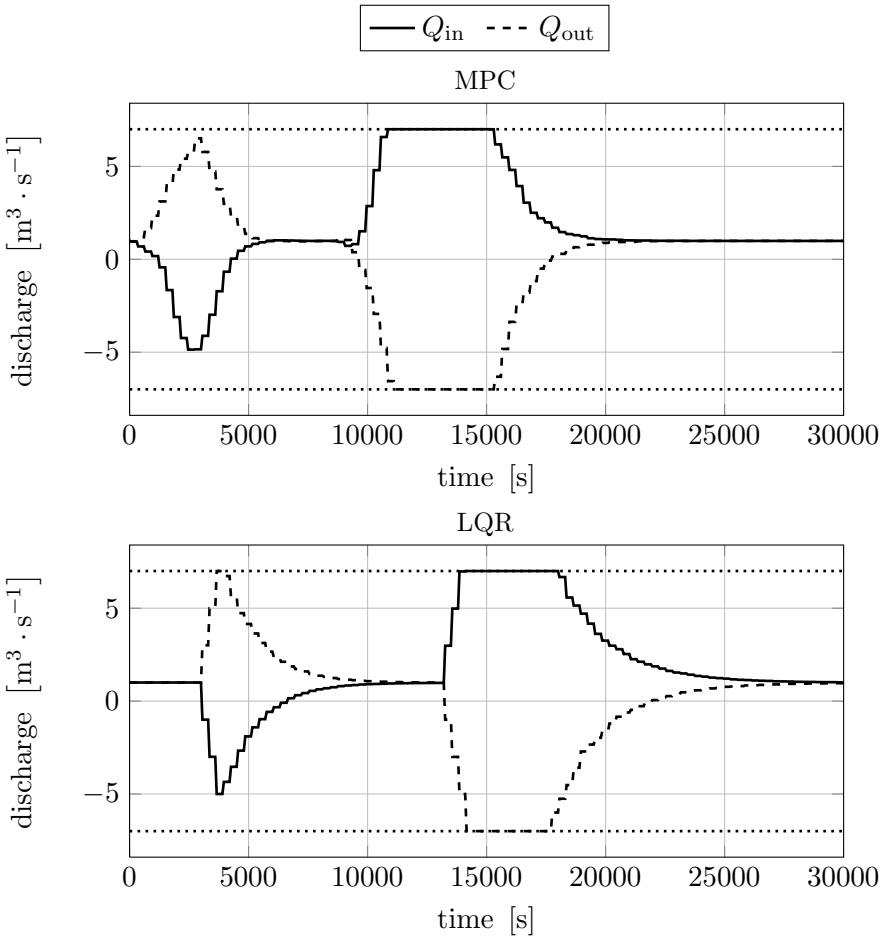


Figure 4.2: Control actions Q_{in} and Q_{out} applied by MPC (top) and LQR (bottom) to a single reach for tracking the changing set-points. The control actions remain between their upper and lower limits (dotted lines): MPC because of the incorporation of these limits inside the optimization problem, LQR because of the use of a saturator. MPC changes the control actions earlier than LQR because it sees a change in the reference trajectory in advance due to the use of a prediction window.

Table 4.3: The set-point deviation cost J_h calculated for the most downstream water level controlled with LQR and MPC for the disturbance rejection test case.

Simulation cost	MPC	LQR
J_h [m]	7	64

The results are visualized in Fig. 4.3. On top we have the reference signal for the most downstream water level, the water level controlled by LQR and the water level controlled by MPC. The bottom plot shows the disturbance signal and the control action for both controllers. It is evident that MPC has better disturbance rejection capabilities than LQR. At the end we have a deviation of 34 cm for the most downstream water level from the set-point for LQR, and a deviation of only 0.8 cm for MPC. Table 4.3 contains the set-point deviation cost J_h only for the most downstream water level of the reach. The cost J_h confirms the conclusion that the use of MPC results in a better disturbance rejection than when LQR is used.

4.3.4.3 Flood control

The last test case is very similar as the previous one, but now the disturbance signal becomes so large that there is a risk of flooding. Again the upstream discharge cannot be controlled, only the downstream discharge can be adjusted by the controllers. The upstream discharge is given by the following function:

$$Q_{\text{in}}(k) = \begin{cases} 1 & \text{for } k \leq 5\,000 \text{ or } k \geq 10\,000, \\ 1 + 15 \cdot \sin\left(\pi \frac{k - 5\,000}{5\,000}\right) & \text{elsewhere.} \end{cases}$$

There is a risk of flooding because Q_{in} exceeds the upper limit of Q_{out} for a long period. The flood levels are taken equal to $\mathbf{h}_{\max} = \mathbf{h}_{\text{ss}} + 0.5$.

Fig. 4.4 shows the results for the two controllers. The top plot visualizes the evolution of the maximal flooding. The maximal flooding is defined as

$$m(k) = \max(\mathbf{h}(k) - \mathbf{h}_{\max}).$$

A negative $m(k)$ means that none of the water levels violates the flood limit at time k. The value of m corresponds in this case with the minimal margin before a flood level is violated. If the reach is flooding, then $m(k)$ is positive and it indicates the maximal violation of a flood level. The middle plot shows

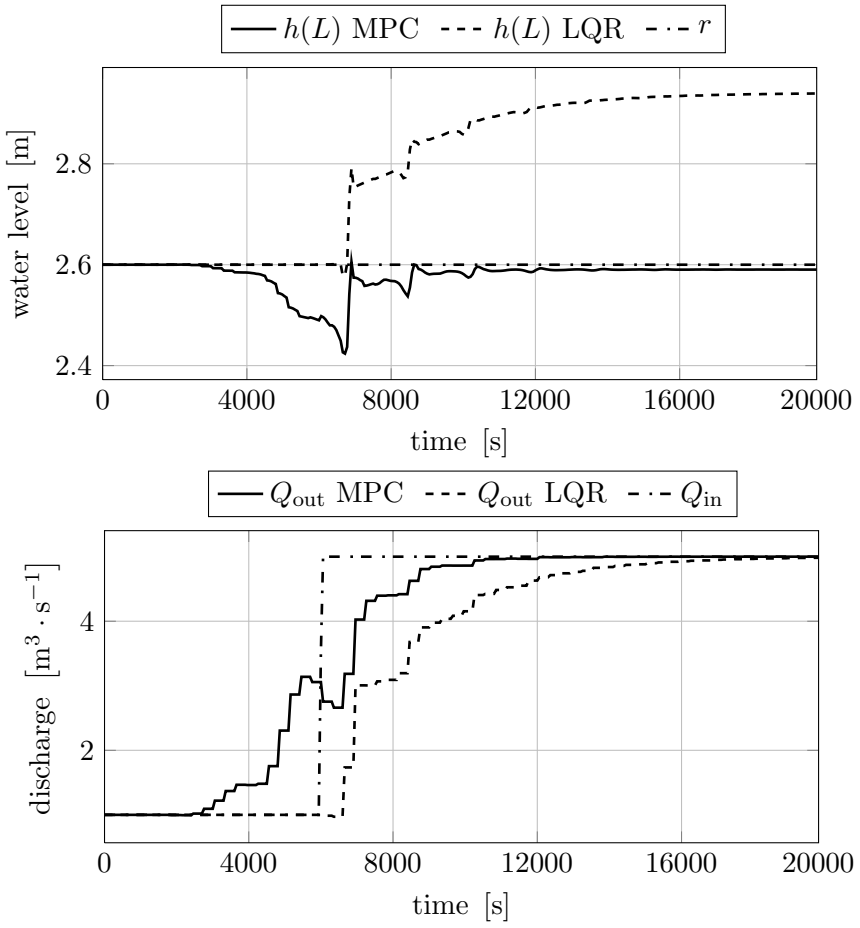


Figure 4.3: The top plot shows the effect of a disturbance on the evolution of the most downstream water level controlled by MPC and LQR together with its set-point. The bottom plot shows the control actions (changes in Q_{out}) for both controllers together with the disturbance signal Q_{in}. The steady state error for the downstream water level with MPC is much smaller than the steady state error with LQR: 8 mm vs. 34 cm.

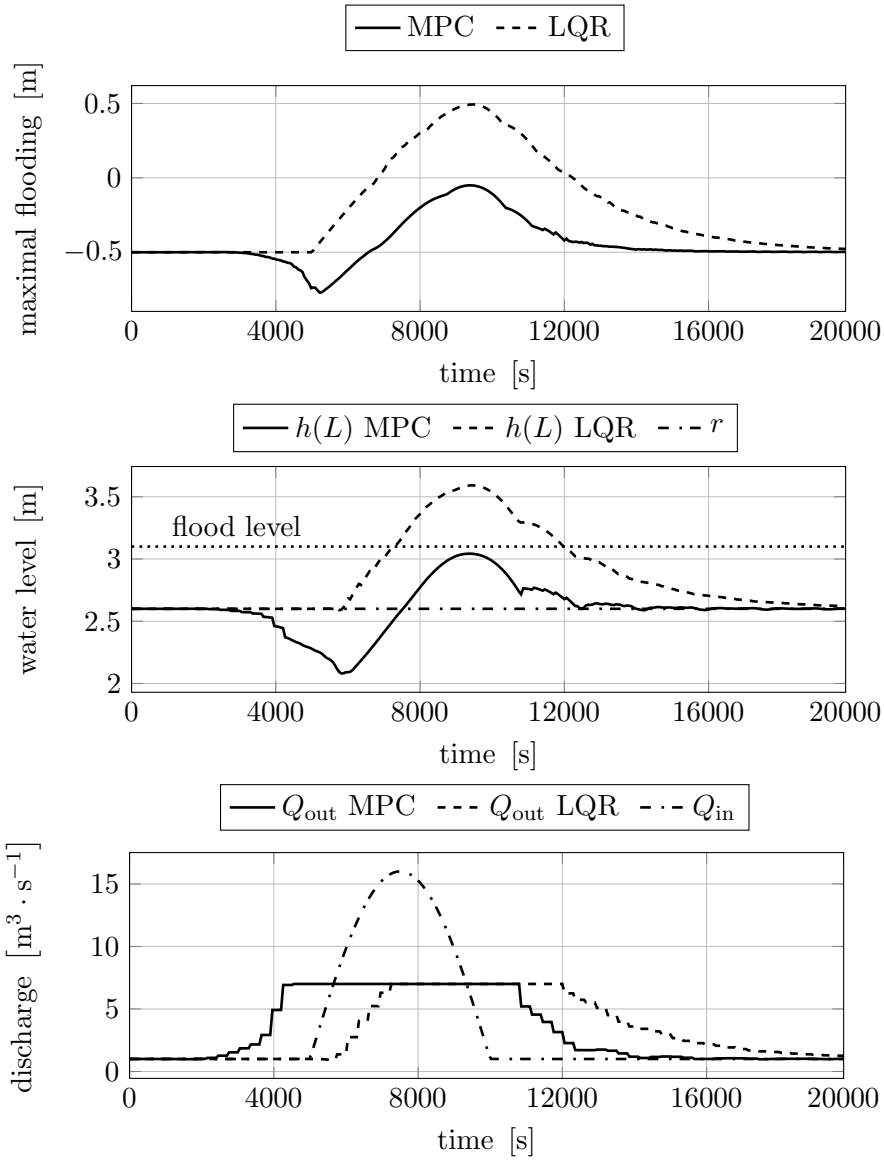


Figure 4.4: The top plot shows the maximal flooding along the reach for the MPC and LQR. The most downstream water level is shown in the middle plot while the bottom plot visualizes the control actions (changes in Q_{out}) of the two controllers together with the disturbance signal Q_{in}. MPC outperforms LQR because it takes rainfall predictions and flood levels into account: MPC reacts before the disturbance takes place.

Table 4.4: The set-point deviation cost J_h and the flood cost $J_{h_{\max}}$ obtained with LQR and MPC applied to a single reach for the flood control test case.

Simulation cost	MPC	LQR
J_h [m]	341	879
$J_{h_{\max}}$ [m]	0	224

the evolution of the most downstream water level for both controllers together with its reference value and flood level. The bottom plot shows the disturbance signal and the control actions for both controllers. Fig. 4.5 visualizes the evolution of all the water levels controlled by means of the MPC and LQR.

LQR cannot prevent the reach from flooding: the maximal flooding m becomes highly positive. The maximal violation of the flood levels is 0.5 m. This is not the case for the MPC where m remains always negative. The reason for this difference can be seen in the middle plot. Long before the increase in the upstream discharge takes place, the MPC steers the downstream water level below its reference value. MPC executes this preventive control action because the prediction horizon allows it to see the big increase in the upstream discharge before it actually happens, in combination with the flood levels imposed as inequality constraints inside the optimization problem. The bottom plot clearly shows how MPC reacts before the disturbance takes place, much earlier than LQR. These control actions have as effect that all the water levels decrease and make m more negative: the controller creates extra buffer capacity. After the disturbance has taken place both controllers succeed in steering the downstream water level back to its reference value.

Table 4.4 contains the set-point deviation cost J_h and the flood cost $J_{h_{\max}}$ for both controllers. Because MPC can keep the water levels closer to their set-points, the cost J_h with MPC is more than 2.5 times smaller than with LQR. The cost $J_{h_{\max}}$ is equal to zero with MPC because flooding is prevented. This is not the case with LQR.

4.3.5 Simulation results with state estimator

In the previous example it was assumed that all the states are known at every time step. However this is practically impossible because this would require a large amount of sensors for measuring the water levels and the discharges along the entire reach. In practice only a limited number of measurements

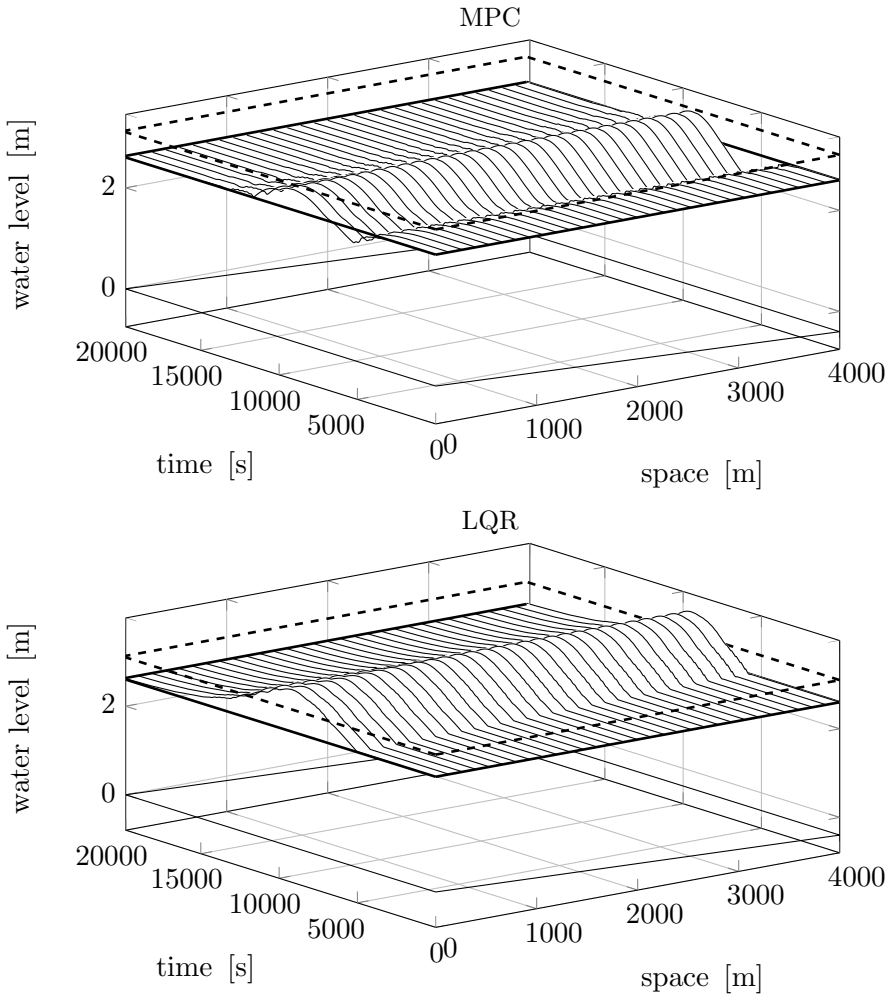


Figure 4.5: Evolution of the water levels in space and time together with their reference trajectory (bold line) for a single reach controlled by means of MPC (top) and LQR (bottom). The dashed lines represent the flood level of the reach. MPC succeeds in keeping all the water levels below their flood level while this is not the case with LQR. This is achieved by creating temporarily extra buffer capacity by lowering the water levels of the reach before the disturbance takes place.

Table 4.5: The set-point deviation cost J_h obtained with MPC with full state information and MPC in combination with a Kalman filter applied to a single reach for the set-point control test case.

Simulation cost	MPC	MPC + Kalman
J_h [m]	920	877

are taken. Based on this little amount of information in combination with the model equations it is still possible to get an accurate estimate of the entire state of the system by using a state estimator. In this section a Kalman filter is used to estimate the states of the reach when only the most downstream water level is measured and the applied control actions and disturbances are known [84]. The Kalman filter was introduced in Section 2.4.

The results of using MPC together with a Kalman filter will be discussed in the next subsections. We will only consider the set-point and flood control test cases. The Kalman filter will be based on the same linear state space model as the one used by the controller (Eq. (4.5)). The following weighting matrices were used to find the Kalman filter gain $\mathbf{L}$:

$$\mathbf{Q}_{\text{est}} = 10^{-3} \cdot \mathbf{I}_{n_h+n_Q},$$

$$\mathbf{R}_{\text{est}} = 10^{-6} \cdot \mathbf{I}_{n_u}.$$

4.3.5.1 Set-point control

Fig. 4.6 compares the control actions for MPC with and without the Kalman filter. The top plot shows the applied upstream discharges Q_{in} of both controllers, while the middle plot shows the applied downstream discharges Q_{out}. The difference between the applied control actions for both controllers can be found in the bottom plot. It is clear that this difference is small. This means that with measuring only one water level, MPC in combination with the Kalman filter can effectively track the time-varying reference trajectory for the water levels. This can also be seen in Fig. 4.7 where the evolution of the water levels in space and time is shown. The figure shows also the evolution of the most downstream water level when the reach is controlled with MPC and MPC in combination with the Kalman Filter. The differences between both controllers are very small. This is also confirmed by the set-point deviation cost J_h given in Table 4.5. The costs for both controllers are almost the same.

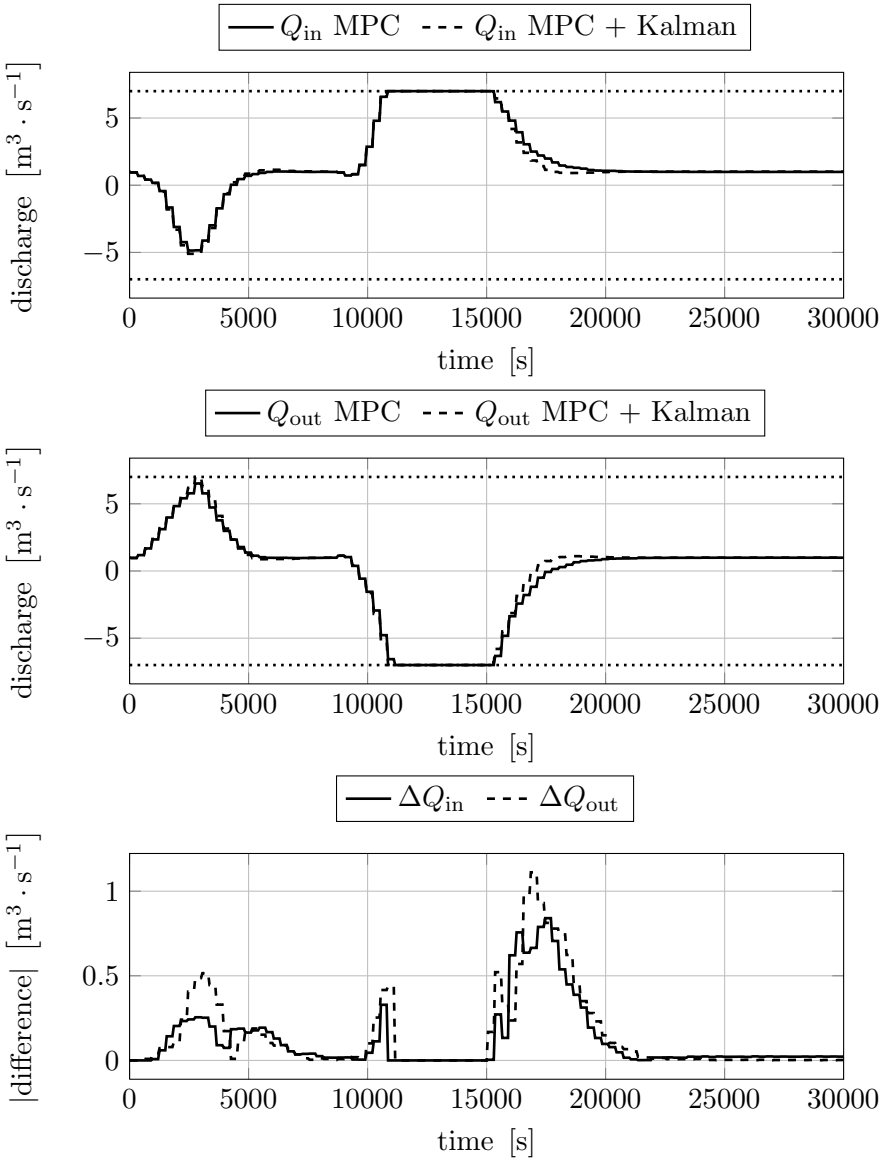


Figure 4.6: Comparison of the control actions Q_{in} (top) and Q_{out} (middle) for MPC with and without a Kalman filter applied to a single reach for set-point control. The bottom plot shows the difference for both control actions for both controllers. The effect of estimating the states based on only one water level measurement on the control actions is limited.

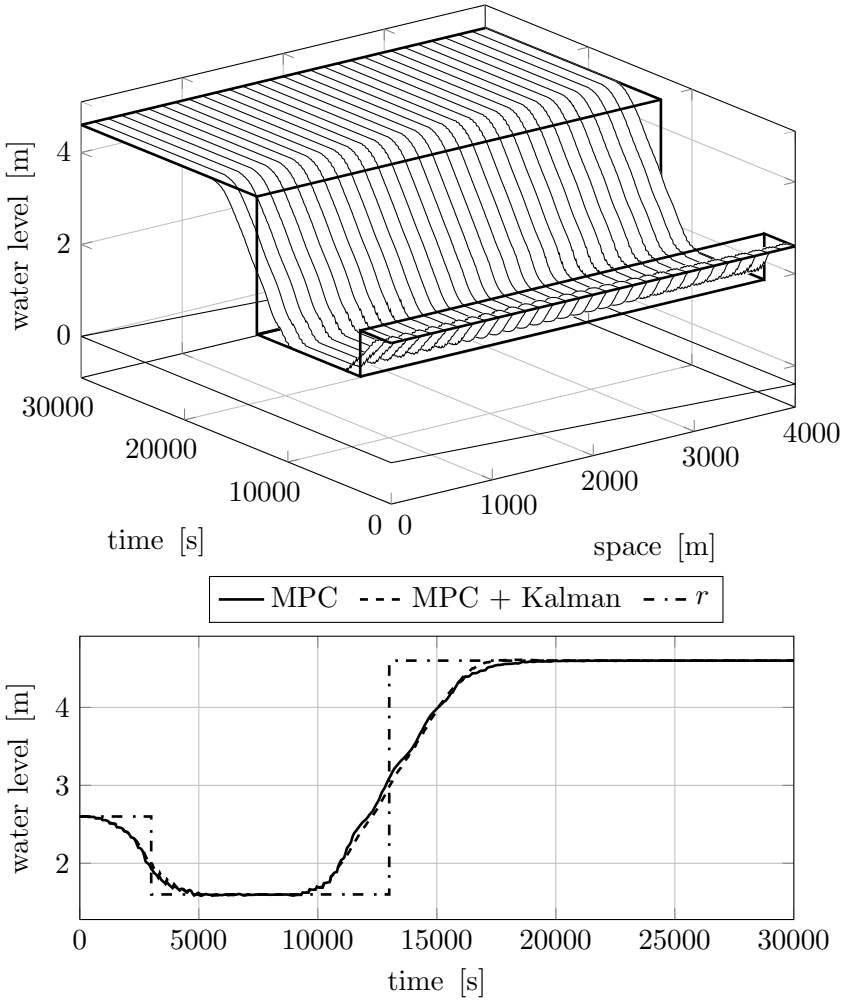


Figure 4.7: The top figure shows the evolution of the water levels of the single reach together with the time-varying reference trajectory when MPC together with a Kalman filter is used. The bottom figure shows the downstream water level for the MPC and Kalman-based MPC cases, together with its set-point. The Kalman filter based MPC effectively tracks the reference trajectory. The difference in tracking performance with the situation where all the states are measured is very limited.

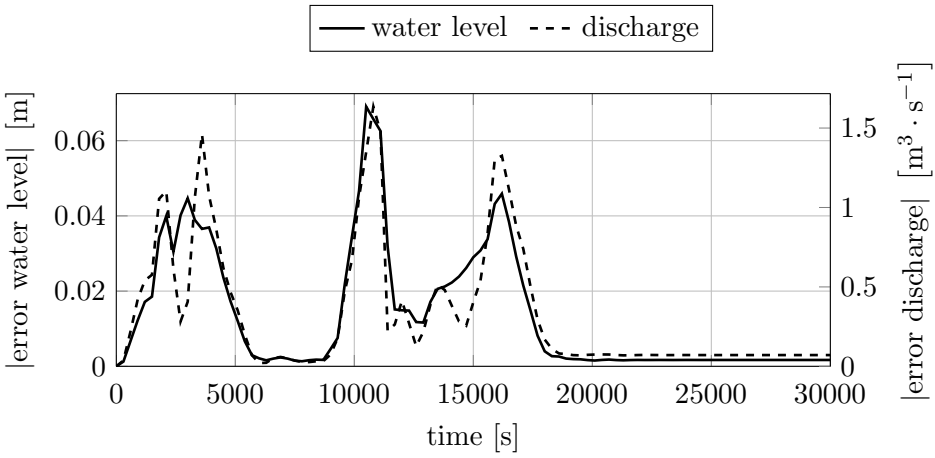


Figure 4.8: Maximum estimation errors for the water levels and the discharges when a Kalman filter is applied to a single reach for the set-point control test case.

The estimation errors for the Kalman filter can be seen in Fig. 4.8. This plot shows the maximal absolute estimation error for the water levels and the discharges at every time step. Overall these estimation errors are small. They increase when there is a change in the reference trajectory. However afterwards the errors converge to a value close to zero.

4.3.5.2 Flood control

The same disturbance signal of Section 4.3.4.3 has been used to see the influence of the state estimator on the flood control performance of MPC. Fig. 4.9 shows the maximal flooding m and the control actions (Q_{out}) of the MPC controller with and without the Kalman filter. Both controllers succeed in avoiding any flooding and keeping the maximal flooding below zero. The maximal value for m is for both controllers very similar. The only difference is that MPC starts decreasing the water levels earlier than the Kalman-based MPC. This also can be seen from the control actions where Q_{out} is increased faster to its maximal value when no state estimator is needed. Table 4.6 gives the set-point deviation cost J_h and the flood cost $J_{h_{\max}}$ for both controllers. The set-point deviation cost for MPC in combination with the Kalman filter is slightly larger than the set-point deviation cost for MPC with full state information. Both controllers have a flood cost equal to zero because both controller succeed in preventing the reach from flooding. The influence of the Kalman filter on the control performance is very limited.

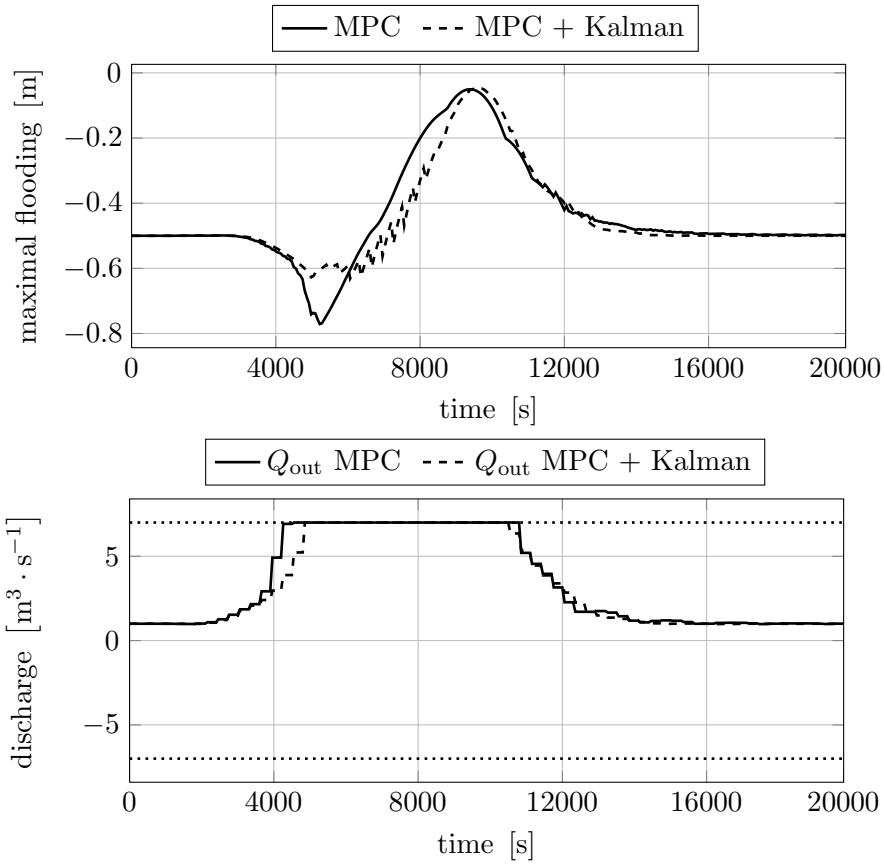


Figure 4.9: The maximal flooding (top) and the control actions (bottom) for MPC with and without the Kalman filter applied to a single reach. MPC without a Kalman filter reacts a little more aggressive than when a Kalman filter is needed. Both controllers succeed in avoiding any flooding of the reach.

The state estimation errors are given in Fig. 4.10. These errors increase during the period of heavy rainfall. Afterwards they converge to a small value. The estimation errors are larger than those of the previous test case. This is because the variables deviate much more from their linearization point. However the effect of these state estimation errors on the control performance is limited.

Table 4.6: The set-point deviation cost J_h and the flood cost $J_{h_{\max}}$ obtained with MPC with full state information and MPC in combination with a Kalman filter applied to a single reach for the flood control test case.

Simulation cost	MPC	MPC + Kalman
J_h [m]	341	296
$J_{h_{\max}}$ [m]	0	0

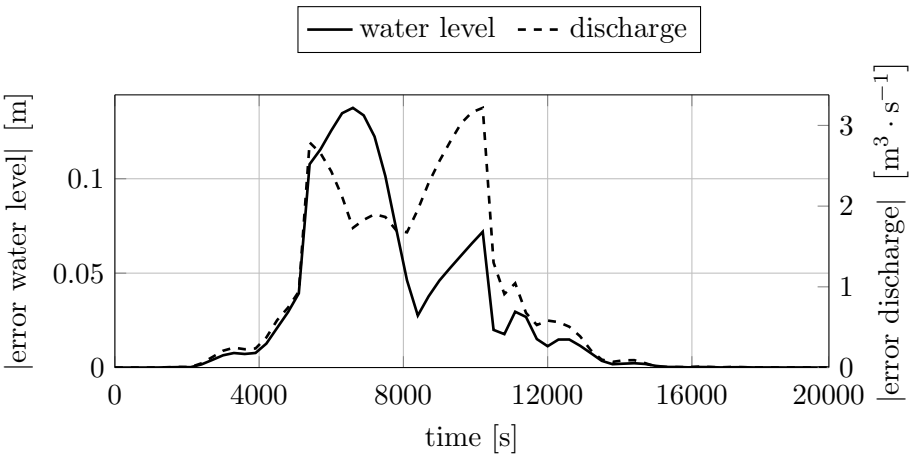


Figure 4.10: Maximum estimation errors for the water levels and the discharges for a Kalman filter applied to a single reach for the flood control test case.

4.4 Model Predictive Control in combination with gate equations

Until now the test system was very simple: the controller only needed to control a single reach. In this section we will expand the MPC formulation given in the previous section in such a way that it can effectively be used to control a river system consisting of two reaches connected to each other by a hydraulic structure. We will explain how the nonlinear gate equations can be taken into account by the controller while the resulting optimization problem still remains a QP.

4.4.1 Mathematical model of the test system

The river system that will be used to test the control performance consists of two consecutive trapezoidal reaches with a vertical sluice in between (see Fig. 4.11). The most downstream discharge Q_{out} can be directly controlled (e.g. with a pump), the most upstream discharge Q_{in} is either a disturbance signal or also an input variable. A gate controls the discharge going from the first reach to the second reach. The system dynamics are described by the following mathematical model (for details check Chapter 3):

$$Q^{(1)}(0,t) = Q_{\text{in}}(t), \tag{4.8}$$

$$\begin{cases} \dfrac{\partial A^{(1)}}{\partial h}\dfrac{\partial h^{(1)}}{\partial t} + \dfrac{\partial Q^{(1)}}{\partial z} = 0, \\ \dfrac{\partial Q^{(1)}}{\partial t} + \dfrac{\partial}{\partial z}\dfrac{{Q^{(1)}}^2}{A^{(1)}} + gA^{(1)}\left(\dfrac{\partial h^{(1)}}{\partial z} + S_{\text{f}}^{(1)} - S_0^{(1)}\right) = 0, \end{cases} \tag{4.9}$$

$$\begin{cases} Q^{(2)}(0,t) = \tilde{f}(c(t), h^{(1)}(L^{(1)},t), h^{(2)}(0,t)), \\ Q^{(1)}(L^{(1)},t) = Q^{(2)}(0,t), \end{cases} \tag{4.10}$$

$$\begin{cases} \dfrac{\partial A^{(2)}}{\partial h}\dfrac{\partial h^{(2)}}{\partial t} + \dfrac{\partial Q^{(2)}}{\partial z} = 0, \\ \dfrac{\partial Q^{(2)}}{\partial t} + \dfrac{\partial}{\partial z}\dfrac{{Q^{(2)}}^2}{A^{(2)}} + gA^{(2)}\left(\dfrac{\partial h^{(2)}}{\partial z} + S_{\text{f}}^{(2)} - S_0^{(2)}\right) = 0, \end{cases} \tag{4.11}$$

$$Q^{(2)}(L^{(2)},t) = Q_{\text{out}}(t), \tag{4.12}$$

where Eqs. (4.8) and (4.12) correspond to the upstream and downstream boundary condition of the river system, Eq. (4.10) with the internal boundary condition due to the presence of the gate and Eqs. (4.9) and (4.11) with the Saint-Venant equations modelling the dynamics of the water levels and discharges of both reaches. The function $\tilde{f}$ is given by Eq. (3.9), which corresponds to the gate equation modelling the discharge over a vertical sluice.

4.4.2 Approximate model and the selection of control variables

The most logical choice for the control variables would be the gate position c together with the upstream discharge Q_{in} and the downstream discharge Q_{out}. At the same time we want to work with a linear state space model as approximate model because then the optimization problem will be a QP. This means that not only the Saint-Venant equations will be linearized, but also

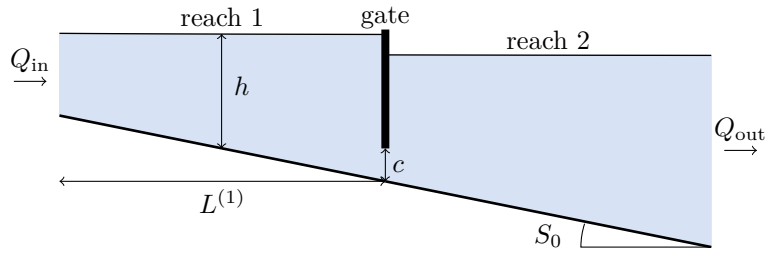


Figure 4.11: Schematic structure of the river system with Q_{in} and Q_{out} the discharges at the boundaries, c the gate position, h the water levels, $L^{(i)}$ the length of the ith reach and S_0 the channel slope.

the nonlinear gate equations. However as we have shown in Section 3.6.2, a linear state space model based on a linearized version of Eqs. (4.8)-(4.12) does not accurately approximate the dynamics of the full nonlinear model. The problem is that the gate equations are too complex to be approximated with one linear model for a wide range of upstream and downstream water levels and gate positions. The solution presented in that section is to use the Linear-Nonlinear model (LN-model). This hybrid model combines a linear state space model for approximating the dynamics of the reaches with the nonlinear gate equations. The effect of the gates on the water levels and discharges is pulled out of the linear model, which increases the accuracy of the resulting model. The approximate model is the following one:

$$\underbrace{\begin{bmatrix} \mathbf{x}^{(1)}(k+1) \\ \mathbf{x}^{(2)}(k+1) \end{bmatrix}}_{\mathbf{x}(k+1)} = \underbrace{\begin{bmatrix} \mathbf{A}^{(1)} & \\ & \mathbf{A}^{(2)} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \mathbf{x}^{(1)}(k) \\ \mathbf{x}^{(2)}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{B}^{(1)} \\ \mathbf{B}^{(2)} \end{bmatrix}}_{\mathbf{B}} \mathbf{u}(k) + \\ + \underbrace{\begin{bmatrix} \mathbf{F}^{(1)} \\ \mathbf{F}^{(2)} \end{bmatrix}}_{\mathbf{F}} \mathbf{d}(k) + \underbrace{\begin{bmatrix} \boldsymbol{\beta}^{(1)} \\ \boldsymbol{\beta}^{(2)} \end{bmatrix}}_{\boldsymbol{\beta}}, \tag{4.13}$$

$$Q_{\text{gate}}(k) = \tilde{f}\left(c(k), h^{(1)}_{n_h^{(1)}}(k), h^{(2)}_1(k)\right), \tag{4.14}$$

where $\mathbf{x}^{(i)}(k) = \left[\mathbf{h}^{(i)}(k); \mathbf{q}^{(i)}(k)\right]$. The matrices $\mathbf{A}^{(i)} \in \mathbb{R}^{(n_h^{(i)}+n_Q^{(i)})\times(n_h^{(i)}+n_Q^{(i)})}$, $\mathbf{B}^{(i)} \in \mathbb{R}^{(n_h^{(i)}+n_Q^{(i)})\times n_u}$ and $\mathbf{F}^{(i)} \in \mathbb{R}^{(n_h^{(i)}+n_Q^{(i)})\times n_d}$ and the vector $\boldsymbol{\beta}^{(i)} \in \mathbb{R}^{n_h^{(i)}+n_Q^{(i)}}$ can be found in a similar way as for the approximate model for the single reach. The input vector $\mathbf{u}(k)$ contains the controllable upstream and/or downstream river discharges at time instant k together with the gate discharge $Q_{\text{gate}}(k)$.

The disturbance vector $\mathbf{d}(k)$ contains the uncontrollable upstream and/or downstream river discharges. E.g. for an uncontrollable upstream discharge, we have that $\mathbf{u}(k) = [Q_{\text{out}}(k), Q_{\text{gate}}(k)]^{\text{T}}$ and $\mathbf{d}(k) = Q_{\text{in}}(k)$.

Using the linear state space model (Eq. (4.13)) together with the gate equation (Eq. (4.14)) inside the optimization problem would not result in solving a QP, but a Nonlinear Programming problem (NLP). As will be shown in the next section, this NLP can be approximated with a QP if we do not work with the gate positions c as optimization variables, but with the gate discharges Q_{gate}. The gate equations are excluded from the optimization problem. Instead of finding a sequence of optimal gate positions, the optimizer will now search for a **sequence of optimal gate discharges**. Therefore, the constraints on the gate positions need to be translated to constraints on the gate discharges and a conversion is needed from the optimal gate discharges to the corresponding optimal gate positions once the QP is solved.

4.4.3 Formulation of the optimization problem

Using the LN-model directly inside the optimization problem results in solving the following NLP at every time step:

$$\min_{\mathbf{u},\mathbf{x},\boldsymbol{\zeta},c} \sum_{j=1}^{N_{\text{P}}} \|\mathbf{x}(j) - \mathbf{r}_x(j)\|^2_{\mathbf{W}} + \sum_{j=0}^{N_{\text{P}}-1} \|\mathbf{u}(j) - \mathbf{u}(j-1)\|^2_{\mathbf{R}} + \|\boldsymbol{\zeta}\|^2_{\mathbf{V}} + \mathbf{v}^{\text{T}}\boldsymbol{\zeta}$$

$$\begin{aligned}
\text{s.t. } & \mathbf{x}(0) = \hat{\mathbf{x}}, \\
& \mathbf{x}(j+1) = \mathbf{A}\mathbf{x}(j) + \mathbf{B}\mathbf{u}(j) + \mathbf{F}\mathbf{d}(j) + \boldsymbol{\beta}, && j = 0, \ldots, N_{\text{P}} - 1 \\
& u^{(\text{gate})}(j) = \bar{f}(c(j), \mathbf{x}(j)), && j = 0, \ldots, N_{\text{P}} - 1 \\
& \underline{\mathbf{u}} \leq \mathbf{u}(j) \leq \overline{\mathbf{u}} && j = 0, \ldots, N_{\text{P}} - 1 \\
& |\mathbf{u}(j) - \mathbf{u}(j-1)| \leq \boldsymbol{\Delta}_u, && j = 0, \ldots, N_{\text{P}} - 1 \\
& \underline{c} \leq c(j) \leq \overline{c}, && j = 0, \ldots, N_{\text{P}} - 1 \\
& |c(j) - c(j-1)| \leq \Delta_c, && j = 0, \ldots, N_{\text{P}} - 1 \\
& \mathbf{u}(-1) = \mathbf{u}_{\text{prev}}, \\
& c(-1) = c_{\text{prev}}, \\
& \tilde{\mathbf{H}}\mathbf{x}(j) \leq \mathbf{h}_{\max} + \eta(j)\boldsymbol{\zeta}, && j = 1, \ldots, N_{\text{P}}
\end{aligned}$$

$$\boldsymbol{\zeta} \geq 0,$$

with N_P the prediction horizon, $\mathbf{W} \in \mathbb{R}^{n_x \times n_x} \succeq 0$, $\mathbf{R} \in \mathbb{R}^{n_u \times n_u} \succ 0$ and $\mathbf{V} \in \mathbb{R}^{n_h \times n_h} \succ 0$ three diagonal weighting matrices, $\mathbf{v} \in \mathbb{R}^{n_h}$ a weighting vector, $n_x = n_h + n_Q$ the total number of states, $n_h = n_h^{(1)} + n_h^{(2)}$ the total number of water levels, $n_Q = n_Q^{(1)} + n_Q^{(2)}$ the total number of discharges, $\mathbf{r}_x \in \mathbb{R}^{n_x}$ a (possible time-varying) vector containing the set-points for all the states, $\hat{\mathbf{x}} \in \mathbb{R}^{n_x}$ the current state of the process (measured or estimated), $\underline{\mathbf{u}}$ and $\overline{\mathbf{u}}$ the lower and upper limits on the controllable discharges, $\underline{c}$ and $\overline{c}$ the upper and lower limit on the gate position, $\boldsymbol{\Delta}_u$ and Δ_c the maximal allowed rate of change for the controllable discharges, respectively the gate position, $\mathbf{u}_\mathrm{prev}$ and c_prev the discharges and gate position applied in the previous time step, $\tilde{\mathbf{H}} \in \mathbb{R}^{n_h \times (n_h + n_Q)}$ a matrix selecting all the water levels from the state vector $\mathbf{x}$, $\mathbf{h}_\mathrm{max} \in \mathbb{R}^{n_h}$ the flood levels, $\eta(j) = 1/r_\mathrm{c}^{j-1}$ a time-dependent inverse weight (with $r_\mathrm{c} > 1$) and $\boldsymbol{\zeta} \in \mathbb{R}^{n_h}$ a vector of slack variables (one slack variable for each water level). For ease of notation the gate equation $\tilde{f}\left(c(j), h^{(1)}_{n_h^{(1)}}(j), h^{(2)}_1(j)\right)$ is rewritten in a more compact form as $\bar{f}(c(j), \mathbf{x}(j))$.

This NLP can be rewritten without the gate position c as optimization variable by exploiting the fact that there is (most of the time) a one-to-one relation ship between the gate position and the gate discharge for given surrounding water levels. The gate position c can be expressed as a function of the gate discharge and the surrounding water levels:

$$c(j) = \hat{f}\left(u^{(\mathrm{gate})}(j), \mathbf{x}(j)\right).$$

Performing this substitution results in the following NLP:

$$\min_{\mathbf{u},\mathbf{x},\boldsymbol{\zeta}} \sum_{j=1}^{N_\mathrm{P}} \|\mathbf{x}(j) - \mathbf{r}_x(j)\|^2_{\mathbf{W}} + \sum_{j=0}^{N_\mathrm{P}-1} \|\mathbf{u}(j) - \mathbf{u}(j-1)\|^2_{\mathbf{R}} + \|\boldsymbol{\zeta}\|^2_{\mathbf{V}} + \mathbf{v}^\mathrm{T}\boldsymbol{\zeta}$$

$$\begin{aligned}
\text{s.t. } & \mathbf{x}(0) = \hat{\mathbf{x}}, \\
& \mathbf{x}(j+1) = \mathbf{A}\mathbf{x}(j) + \mathbf{B}\mathbf{u}(j) + \mathbf{F}\mathbf{d}(j) + \boldsymbol{\beta}, && j = 0, \ldots, N_\mathrm{P} - 1 \\
& \underline{\mathbf{u}} \leq \mathbf{u}(j) \leq \overline{\mathbf{u}}, && j = 0, \ldots, N_\mathrm{P} - 1 \\
& |\mathbf{u}(j) - \mathbf{u}(j-1)| \leq \boldsymbol{\Delta}_u, && j = 0, \ldots, N_\mathrm{P} - 1 \\
& \underline{c} \leq \hat{f}\left(u^{(\mathrm{gate})}(j), \mathbf{x}(j)\right) \leq \overline{c}, && j = 0, \ldots, N_\mathrm{P} - 1 \\
& \left|\hat{f}\left(u^{(\mathrm{gate})}(j), \mathbf{x}(j)\right) + \right. && j = 0, \ldots, N_\mathrm{P} - 1
\end{aligned}$$

$$-\hat{f}\big(u^{(\mathrm{gate})}(j-1),\mathbf{x}(j-1)\big)\Big| \leq \Delta_c,$$

$$\mathbf{u}(-1) = \mathbf{u}_{\mathrm{prev}},$$

$$\tilde{\mathbf{H}}\mathbf{x}(j) \leq \mathbf{h}_{\max} + \eta(j)\boldsymbol{\zeta}, \qquad j = 1,\ldots,N_{\mathrm{P}}$$

$$\boldsymbol{\zeta} \geq 0.$$

This NLP does not contain c any more as an optimization variable. Following an heuristic approach, which will be discussed in Section 4.4.3.1, the nonlinear inequality constraints related to the gate position and to the change in the gate position are translated into time-varying lower and upper limits on the gate discharges. In this way the optimization problem that needs to be solved at every time step becomes a QP with the following objective function and constraints:

$$\min_{\mathbf{u},\mathbf{x},\boldsymbol{\zeta}} \sum_{j=1}^{N_{\mathrm{P}}} \|\mathbf{x}(j) - \mathbf{r}_x(j)\|^2_{\mathbf{W}} + \sum_{j=0}^{N_{\mathrm{P}}-1} \|\mathbf{u}(j) - \mathbf{u}(j-1)\|^2_{\mathbf{R}} + \|\boldsymbol{\zeta}\|^2_{\mathbf{V}} + \mathbf{v}^{\mathrm{T}}\boldsymbol{\zeta}$$

$$\text{s.t. } \mathbf{x}(0) = \hat{\mathbf{x}},$$

$$\mathbf{x}(j+1) = \mathbf{A}\mathbf{x}(j) + \mathbf{B}\mathbf{u}(j) + \mathbf{F}\mathbf{d}(j) + \boldsymbol{\beta}, \qquad j = 0,\ldots,N_{\mathrm{P}}-1$$

$$\underline{\mathbf{u}}(j) \leq \mathbf{u}(j) \leq \overline{\mathbf{u}}(j), \qquad j = 0,\ldots,N_{\mathrm{P}}-1$$

$$|\mathbf{u}(j) - \mathbf{u}(j-1)| \leq \boldsymbol{\Delta}_u, \qquad j = 0,\ldots,N_{\mathrm{P}}-1$$

$$\mathbf{u}(-1) = \mathbf{u}_{\mathrm{prev}},$$

$$\tilde{\mathbf{H}}\mathbf{x}(j) \leq \mathbf{h}_{\max} + \eta(j)\boldsymbol{\zeta}, \qquad j = 1,\ldots,N_{\mathrm{P}}$$

$$\boldsymbol{\zeta} \geq 0.$$

The resulting QP is very similar to the QP formulated for controlling a single reach in the previous section. Again the flood levels are implemented as soft constraints to avoid possible infeasibilities such that only one QP has to be solved at every time step (see Section 4.3.3.1). The different weights of the matrices $\mathbf{W}$, $\mathbf{R}$ and $\mathbf{V}$, and the vector $\mathbf{v}$ are chosen in a similar fashion as the weights used for controlling the single reach. More information is given in Section 4.3.3.2. In the next subsections it is explained why the upper limits and lower limits are time-varying and how they can be found. Also the issue of possible uncontrollability of the gates and a solution is discussed.

4.4.3.1 Time-varying lower and upper limits for gate discharges

The lower and upper limits for the controllable discharges at the upstream and downstream end of the river are constant and assumed to be given. Because the QP works with the gate discharges as optimization variables, the lower and upper limit $\underline{c}$ and $\overline{c}$, and the maximal rate of change Δ_c for a gate need to be converted to a lower and upper limit of the corresponding gate discharge at every time step. This conversion will result in time-varying lower and upper limits for the gate discharges because they depend on the gate position and on the upstream and downstream water levels of the gate at every time step.

These lower and upper limits can be calculated with Algorithm 3. Assume that a sequence of optimal control actions $\mathbf{u}_{\text{opt}}(k-1+j)$ for $j = 0, \ldots, N_{\text{P}} - 1$ was found by the controller in the previous time step $k-1$. Given the applied control action for the gate $c(k-1)$ in the previous time step and (an estimate of) the current state of the system, the maximal and minimal possible gate discharges that can be achieved by an underflow vertical sluice at time step k can be found with:

$$\underline{u}^{(\text{gate})}(k) = \tilde{f}\left(c(k-1) - \Delta_c, h^{(1)}_{n^{(1)}_h}(k), h^{(2)}_1(k)\right), \tag{4.15}$$

$$\overline{u}^{(\text{gate})}(k) = \tilde{f}\left(c(k-1) + \Delta_c, h^{(1)}_{n^{(1)}_h}(k), h^{(2)}_1(k)\right). \tag{4.16}$$

The maximal rate of change Δ_c is incorporated in this way. For calculating the lower and upper limit at time step $k+1$, an estimation is needed of the position of the gate at time step k and of the water levels at time step $k+1$ in order to be able to reuse the same equations. Given the current water levels at time step k and the optimal gate discharge of $\mathbf{u}_{\text{opt}}$ calculated at time k, the estimate of the corresponding gate position at time step k can be found. This is possible because (most of the time) there is a one-to-one relationship between the gate discharge and the gate position for a given upstream and downstream water level. In the algorithm this conversion is performed with the function l. More information about this function will be given in the next subsection. The only thing missing for reusing Eqs. (4.15) and (4.16) are the water levels at time $k+1$. These can be found by performing a prediction step with the linear state space model (Eq. (4.13)) of the LN-model:

$$\tilde{\mathbf{x}}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}_{\text{opt}}(k) + \mathbf{F}\mathbf{d}(k) + \boldsymbol{\beta},$$

where $\tilde{\mathbf{x}}$ represents the predicted states. The wanted water levels $h^{(1)}_{n^{(1)}_h}(k+1)$ and $h^{(2)}_1(k+1)$ can be extracted from these predicted states (this is represented with the function s in the algorithm), and Eqs. (4.15) and (4.16) can be used in combination with $c(k)$ to find an estimate of $\underline{u}^{(\text{gate})}(k+1)$ and $\overline{u}^{(\text{gate})}(k+1)$.

Algorithm 3 Algorithm to find the time-varying limits $\underline{u}^{(\text{gate})}(j)$ and $\overline{u}^{(\text{gate})}(j)$ for a vertical sluice. The function s selects the upstream and downstream water levels of the gate while the function l calculates the gate position corresponding to the desired discharge for the gate and a given upstream and downstream water level. $\tilde{\mathbf{x}}$ represents the predicted states based on the linear state space model and $\mathbf{u}_{\text{opt}}(k+j-1)$, $j = 1, \ldots, N_{\text{P}}$ is the sequence of optimal gate discharges found in the previous time step at time $k-1$.

```
% at time step t_k
x̃ = x(k)
c_prev = c(k − 1)
for j = 1, ..., N_P do
    [h_up, h_down] = s(x̃)
    if h_up ≥ h_down then
        u̲^(gate)(j) = f̃(max(c_prev − Δ_c, c̲), h_up, h_down)
        ū^(gate)(j) = f̃(min(c_prev + Δ_c, c̄), h_up, h_down)
    else
        u̲^(gate)(j) = −f̃(min(c_prev + Δ_c, c̄), h_down, h_up)
        ū^(gate)(j) = −f̃(max(c_prev − Δ_c, c̲), h_down, h_up)
    end if
    c_prev = l(u_opt^(gate)(j), h_up, h_down)
    x̃(k + j) = Ax̃(k + j − 1) + Bu_opt(k + j − 1) + Fd(k + j − 1) + β
end for
```

The same procedure can be used to estimate the time-varying lower and upper limits of the gate discharges for the entire prediction window. The algorithm also takes the lower limit $\underline{c}$ and upper limit $\overline{c}$ of the gate position into account when calculating the limits of the gate discharge.

The nonlinear model could also be used to perform the prediction step instead of using the linear model. The predictions with the nonlinear model will be more accurate but every prediction step will then also take much more time. If the river data is not too irregular (cfr. the bed slope and cross sectional profiles), the linear prediction step is accurate enough (in Chapter 5 where MPC is applied to the Demer, it will be explained how a prediction step based on the nonlinear model can be performed with only a little increase in the computation time).

Because the upper and lower limits of the gate discharges depend on the water levels and the gate positions, we should actually iterate over Algorithm 3 and solving the optimization problem. The new sequence of optimal gate discharges will typically be different from the solution sequence found in the previous time step $\mathbf{u}_{\text{opt}}$. Therefore they will result in other future states and hence also other upper and lower limits on the gate discharges. However simulation results show

that this iteration is not needed because from one time step to the next the optimal control actions do not change drastically. This strategy is similar to real-time iteration in combination with feasibility improvement [29, 55].

Notice that the formulated optimization problem puts a rate of change constraint on all the control variables. For the controllable upstream and downstream discharges these rates are given. For the gate discharge there is no rate of change constraint because the rate of change constraint Δ_c on the gate position results in the time-varying upper and lower limit on the gate discharge. Therefore $\Delta_u^{(\text{gate})}$ is set equal to ∞ for the gate discharge.

Equations (4.15) and (4.16) are valid for an underflow vertical sluice. The same procedure can also be used for a gated weir if the signs in these equations for Δ_c are changed. A discharge over a gated weir decreases when the gate position increases and vice versa.

Remark. It should be noted that the solution found with the QP is not necessarily a (local) minimum of the corresponding NLP, even if we would iterate until convergence. The reason for this is that the inequality constraints related to the gate positions are approximated with linear inequality constraints without taking the dependency of these nonlinear inequality constraints on the states into account. Another approach would be to follow the work done by Q. Tran Dinh in [171] and approximate the nonlinear inequalities by linearizing the function $\hat{f}$ with respect to the gate discharge and the water levels. However based on the simulation results shown in the remaining part of this dissertation, we can conclude that this extra linearization step is not required for achieving a good control performance related to flood control of river systems.

4.4.3.2 Controllability of the gates

The algorithm used to find the time-varying lower and upper limits on the gate discharges over the prediction window, makes use of a function l to find the gate position corresponding with a desired gate discharge for a given upstream and downstream water level. Only when the gate is in its controllable region there is a one-to-one relationship between the gate position and the gate discharge, and the corresponding gate position can be easily found. A gate is considered to be controllable if a change in its gate position results in a change in the gate discharge, otherwise it is called uncontrollable. An underflow vertical sluice is uncontrollable when its gate opening is larger than the upstream and downstream water level: the sluice is lifted out of the water. A gated weir is uncontrollable when the gate is completely closed or when it is in throat control model (see Section 3.3.3.1 for more information).

This uncontrollability of the gates can degrade the control performance if this

is not taken into account. Consider the situation where the gate opening of a vertical sluice should be as large as the upstream water level to achieve a desired positive gate discharge. The upstream and downstream water level for the sluice can be decreased at the next time step such that increasing or decreasing the gate opening over a distance Δ_c will not affect the gate discharge: the sluice is completely lifted out of the water. Hence Eqs. (4.15) and (4.16) will return the same value. This can happen for every time step within the prediction window and at the end the lower limit is equal to the upper limit at all times. This means that the value for the gate discharge is fixed and the controller will stop using this gate, losing in this way one degree of freedom. A similar situation is possible with a gated weir.

This problem can easily be avoided if we make sure that the gate position found by the function l is always within its controllable region or at least on the borderline between both regions. In this way changing the gate opening over a distance Δ_c will always result in different values for the gate discharge. For a vertical sluice this means that when a maximal possible gate discharge is requested, the gate position returned by the function l will be just above the upstream and downstream water level. Similarly if a zero flow is requested for a gated weir, its returned gate position will be just above the upstream and downstream water level. For a maximal discharge for a gated weir the function l returns the gate position such that the gate is at the borderline between the throat control mode and the gate control mode. In all other situations there is a one-to-one relationship between the desired gate discharge and gate position, and this gate position can be found with a simple bisection search method.

This function l is also used to make the conversion from the first discharge of the sequence of optimal gate discharges to its corresponding gate position after solving the QP. This gate position is then applied to the hydraulic structure. This solution can be compared with the approach presented in [18] where the author worked with the gate positions as optimization variables. First of all the author of [18] had to work with nonlinear MPC to achieve a good control performance. Based on the (time-varying) state space matrix $\mathbf{B}$ it is possible to see whether the gate is controllable or not. If the gate is uncontrollable, one has to use a reference value for the gate position inside the optimization problem in combination with a large weight such that the controller will steer this gate back to its controllable region. Checking at every time step over the prediction window whether the gate is controllable or not, updating the reference trajectory for uncontrollable gates and their corresponding weights in the objective function is not needed in our approach. The problem of possible uncontrollability of a gate is automatically solved by optimizing over the gate discharges and the use of the function l.

Table 4.7: Parameters for the two trapezoidal channels used to test the performance of MPC.

	L	S_0	n_{mann}	B	S_1, S_2	n_h
reach 1	1000	0.0004	0.045	6	0.5	10
reach 2	2000	0.0002	0.030	6	0.5	20

4.4.4 Simulation results

The proposed MPC will be tested on a river system for which the dynamics are given in Section 4.4.1. The parameters for both trapezoidal reaches are given in Table 4.7. The vertical sluice in between both channels is as wide as the bottom of both channels. The performance of MPC with the gate discharges as optimization variables proposed in the previous subsections will be compared with an MPC with the gate positions as optimization variables. The capabilities for set-point tracking and flood control will be tested for both MPC formulations. The controller based on the LN-model will be denoted with LN-MPC while L-MPC will be used for the controller based on the full linear model. The closed loop simulations are performed with the full hydrodynamic model of the river system.

Initially the flow is constant in both reaches and equal to 4 $\text{m}^3 \cdot \text{s}^{-1}$. The gate opening is 0.8 m and the most downstream water level is 4 m. The flood levels of the first reach are 1 m above its initial water levels while for the second reach they are only 40 cm above its initial water levels. In order to prevent the second reach from flooding, the controllers will have to use the extra buffer capacity of the first reach. The gate opening c of the vertical sluice has to be between 0 m and 2 m and can only move 20 cm every 15 min. The controllable upstream discharge Q_{in} and downstream discharge Q_{out} have to remain between -7 and 7 $\text{m}^3 \cdot \text{s}^{-1}$ and can only change with 5 $\text{m}^3 \cdot \text{s}^{-1}$ every 15 min. Both controllers have a prediction window size N_{P} of 20. Table 4.8 shows the diagonal elements of the weight matrices $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{R}$, and the entries of the weight vector $\mathbf{v}$. The values for $\mathbf{V}$ and $\mathbf{v}$ are larger than the other weights such that the controllers will focus on flood control when there is a risk of flooding. The highest weight of the matrix $\mathbf{W}$ is used for the downstream water level of both reaches. During set-point control the controllers will try to steer these water levels as close as possible to their reference. The values for the matrix $\mathbf{R}$ are one order smaller to give the controllers enough freedom to use the input variables.

Table 4.8: Diagonal elements for the weight matrices $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{R}$, and the elements for the weight vector $\mathbf{v}$ for set-point control and flood control of two reaches.

	reach 1	reach 2
$\mathbf{W} \in \mathbb{R}^{62\times62}$		
water levels	$\begin{bmatrix} 10^{-3}\cdot\mathbf{1}_9 \\ 10 \end{bmatrix}$	$\begin{bmatrix} 10^{-3}\cdot\mathbf{1}_{19} \\ 10 \end{bmatrix}$
discharges	$10^{-3}\cdot\mathbf{1}_{11}$	$10^{-3}\cdot\mathbf{1}_{21}$
$\mathbf{V} \in \mathbb{R}^{30\times30}$		
flood levels	$10^{4}\cdot\mathbf{1}_{10}$	$10^{4}\cdot\mathbf{1}_{20}$
$\mathbf{v} \in \mathbb{R}^{30}$		
flood levels	$10^{4}\cdot\mathbf{1}_{10}$	$10^{4}\cdot\mathbf{1}_{20}$

	Q_{in}	Q_{out}	Q_{gate} or c
$\mathbf{R} \in \mathbb{R}^{3\times3}$ or $\mathbb{R}^{2\times2}$			
control actions	1	1	1

4.4.4.1 Set-point control

The first test case checks the tracking performance of LN-MPC with full state information and L-MPC with full state information. In this test case both controllers can use the upstream discharge Q_{in}, the downstream discharge Q_{out} and the gate position c, and there is no risk of flooding ($\mathbf{h}_{\max} = \infty$). Fig. 4.12 shows the evolution of the water levels for both controllers together with the changing reference trajectory and the gate opening of the vertical sluice. Fig. 4.13 shows the control actions applied to the upstream discharge Q_{in}, the downstream discharge Q_{out} and the gate opening c. There is almost no difference in the evolution of the water levels and the control actions between both controller for the first three hours. Both controllers succeed in following the first step change in the reference signal. The results for the second step change in the reference signal are completely different. LN-MPC succeeds in keeping the water levels of the first reach almost constant while it decreases the water levels of the second reach towards their new set-point. This is not the case with L-MPC: the water levels of the first reach increase while the water levels of the second reach decrease below their set-point. The reason is

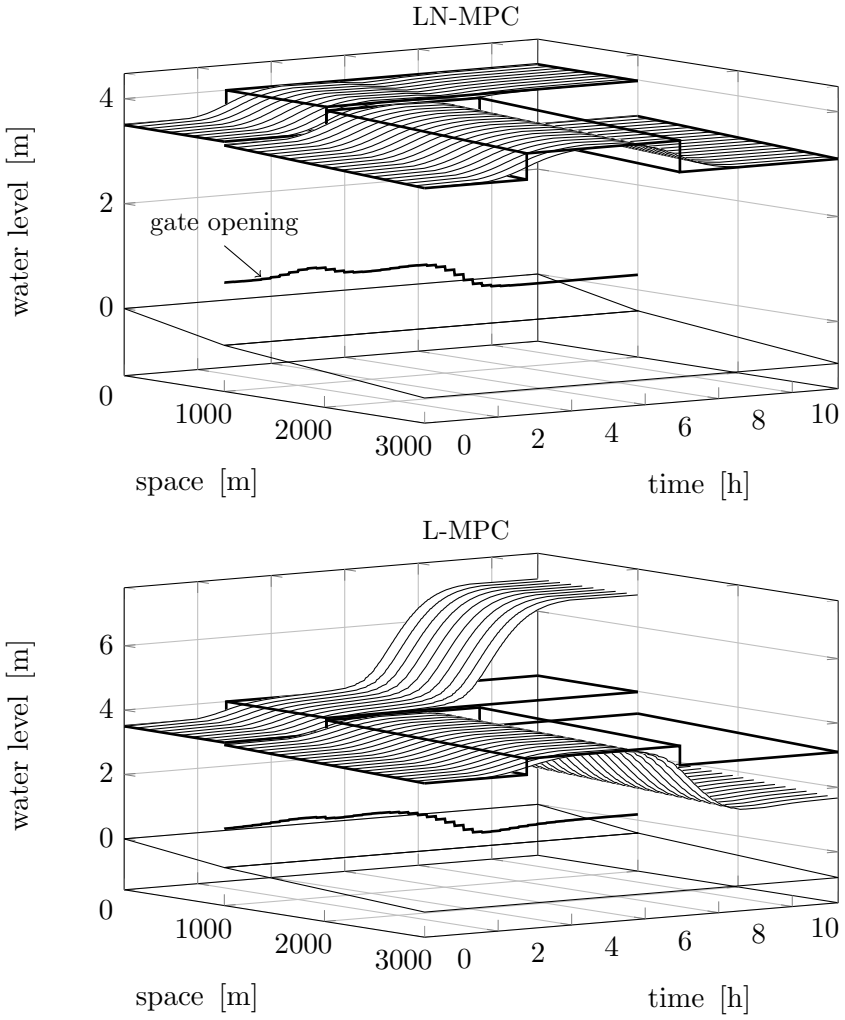


Figure 4.12: Evolution of the water levels in space and time for both reaches and the gate opening of the sluice together with the time-varying reference trajectory for LN-MPC (top) and L-MPC (bottom). By optimizing the gate discharges LN-MPC tracks the reference trajectory very well. L-MPC does not follow the second change in the reference trajectory. The linear approximation of the gate equation is only accurate if the difference between the upstream and downstream water level of the gate does not change too much.

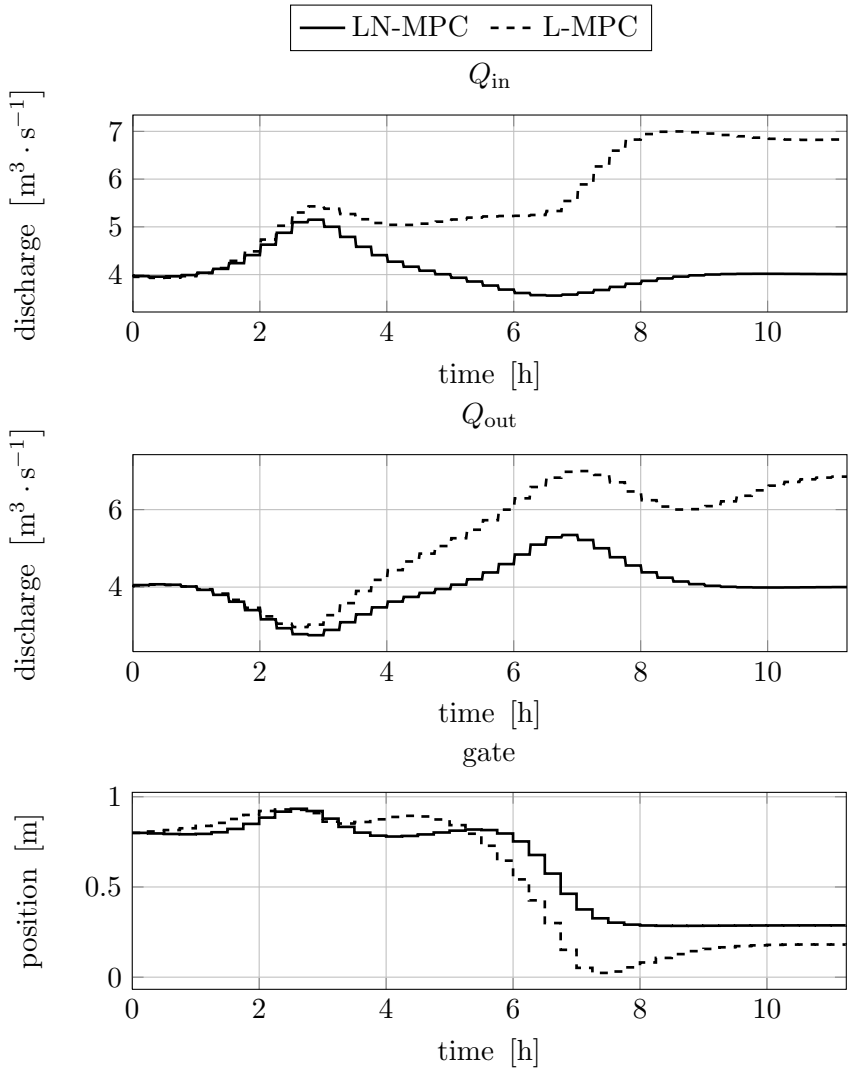


Figure 4.13: The computed upstream discharge Q_{in} (top), the downstream discharge Q_{out} (middle) and the gate position c (bottom) by the LN-MPC and L-MPC controllers when applied to two reaches for set-point tracking. The control actions for the first 3 hours are very similar. As long as the difference between the upstream and downstream water level of the gate does not bring the system far away from the linearization point, the linear approximation of the gate equation is accurate enough. After 3 hours, this deviation increases and the control actions of both controllers are not similar any more.

Table 4.9: The set-point deviation cost J_h obtained with LN-MPC and L-MPC when tracking a time-varying reference trajectory for the water levels off two reaches connected to each other with a vertical sluice.

Simulation cost	LN-MPC	L-MPC
J_h [m]	194	2814

that the linear approximation of the nonlinear gate equations used by L-MPC is really poor. This linear approximation is accurate enough if the difference between the water level upstream and downstream of the gate is comparable to the difference that is present around the linearization point. This is why L-MPC succeeds in tracking the first change in the reference trajectory: the water levels of both reaches have to increase with the same amount, hence the difference between the water level upstream and downstream of the gate will not deviate much from its initial value. Because the second change in the reference trajectory applies only to the second reach, this difference increases and the accuracy of the model decreases resulting in a bad control performance. LN-MPC does not suffer from this problem because the gate equations are pulled out of the optimization problem. The optimal gate positions are found by converting the optimal gate discharges into the gate positions with the use of the nonlinear gate equations. Because L-MPC does not succeed in tracking the second change in the reference trajectory, its set-point deviation cost J_h given in Table 4.9 is much larger than the cost for LN-MPC.

4.4.4.2 Flood control

The upstream discharge Q_{in} is a disturbance signal and is given in Fig. 4.14. Because it exceeds the upper limit of 7 $\text{m}^3 \cdot \text{s}^{-1}$ for the downstream discharge Q_{out} for a long time, there is a risk of flooding. Fig. 4.15 shows the evolution of the water levels of both reaches for LN-MPC with full state information and L-MPC with full state information, while Fig. 4.16 shows the maximal flooding and the control actions for both controllers. Again LN-MPC outperforms L-MPC. By first optimizing over the gate discharges and then performing the gate conversion based on the nonlinear gate equations, LN-MPC succeeds in preventing both reaches from flooding. It successfully uses the extra amount of buffer capacity of the first reach to prevent the second reach from flooding. This is not the case with L-MPC. Because of the inaccuracy of the linear model once the difference between the upstream and downstream water level of the gate starts changing, the L-MPC makes the gate opening too small resulting in a

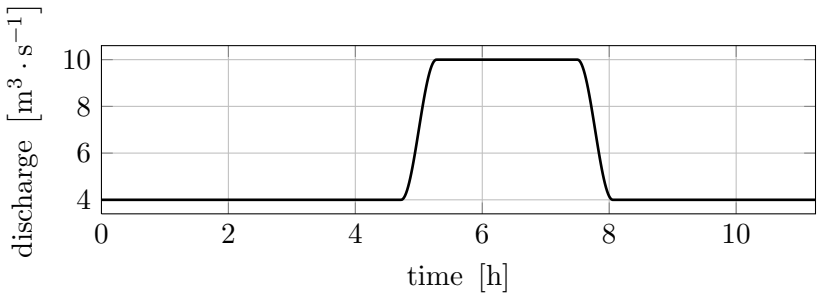


Figure 4.14: Disturbance signal for the upstream discharge Q_{in} used to test the flood control performance of LN-MPC and L-MPC.

Table 4.10: The set-point deviation cost J_h and the flood cost $J_{h_{\max}}$ obtained with LN-MPC and L-MPC controlling a river system consisting of two reaches connected to each other with a vertical sluice.

Simulation cost	LN-MPC	L-MPC
J_h [m]	810	1972
$J_{h_{\max}}$ [m]	0	478

poor control performance and large floods for the first reach. These conclusions are also confirmed by looking at the set-point deviation cost J_h and flood cost $J_{h_{\max}}$ given in Table 4.10. Because LN-MPC keeps the water levels of both reaches much closer to their set-points and below the flood levels at all times, both costs are much smaller than the cost obtained with L-MPC.

Until now it was assumed that all the water levels and discharges of both reaches are known at every time step. In this part this assumption is dropped and only the downstream water level of every reach is measured. A Kalman filter is used to estimate the unknown states. The Kalman gain matrix $\mathbf{L}$ is computed from the linear state space model of the LN-Model (Eq. (4.13)) in conjunction with the following weighting matrices:

$$\mathbf{Q}_{\text{est}} = 10^{-3} \cdot \mathbf{I}_{n_x},$$

$$\mathbf{R}_{\text{est}} = 10^{-6} \cdot \mathbf{I}_{n_u}.$$

Fig. 4.17 shows the maximal flooding for both reaches together with the control actions of LN-MPC with and without the Kalman filter. There is almost no difference between the maximal flooding for both controllers. Both controllers

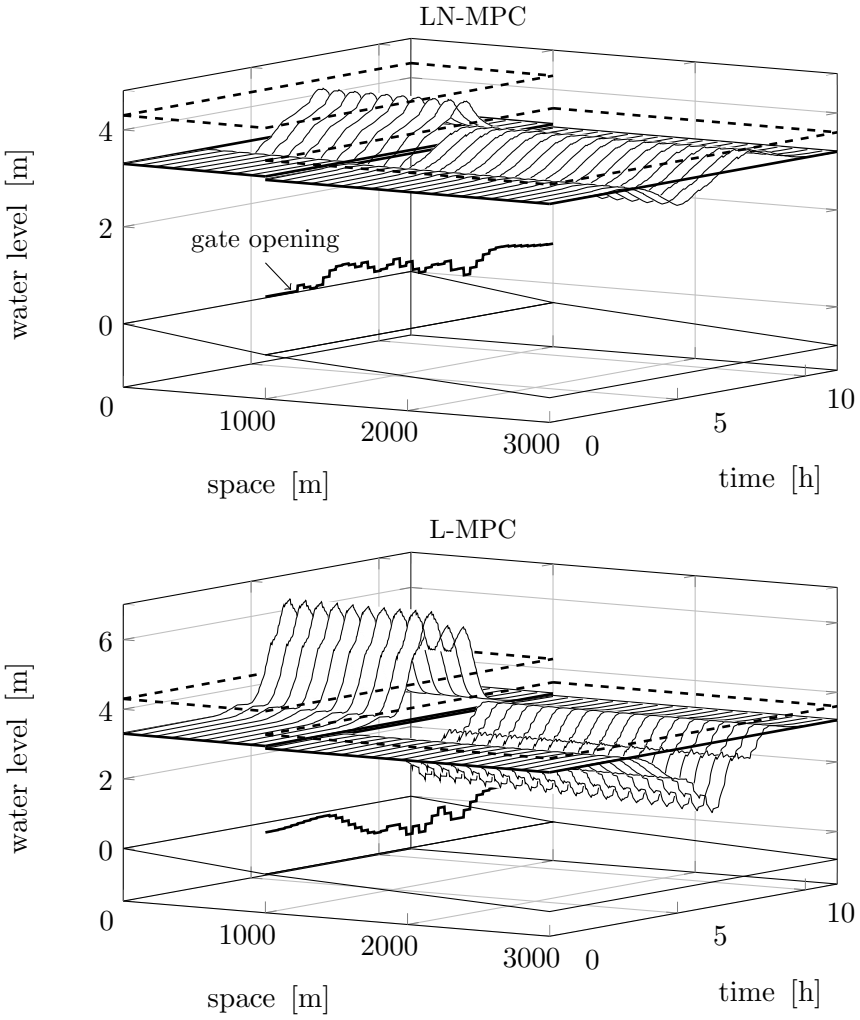


Figure 4.15: Evolution of the water levels in space and time for both reaches and the gate opening of the sluice together with the reference trajectory and the flood levels for LN-MPC (top) and L-MPC (bottom). By optimizing over the gate discharges, the LN-MPC succeeds in keeping all the water levels below their flood levels. The L-MPC cannot prevent the first reach from flooding. The linear approximation of the gate equation is not accurate if the difference between the upstream and downstream water level of the gate increases. The mismatch between the actual effect of changing the gate position and the modelled effect is too large resulting in a bad control performance.

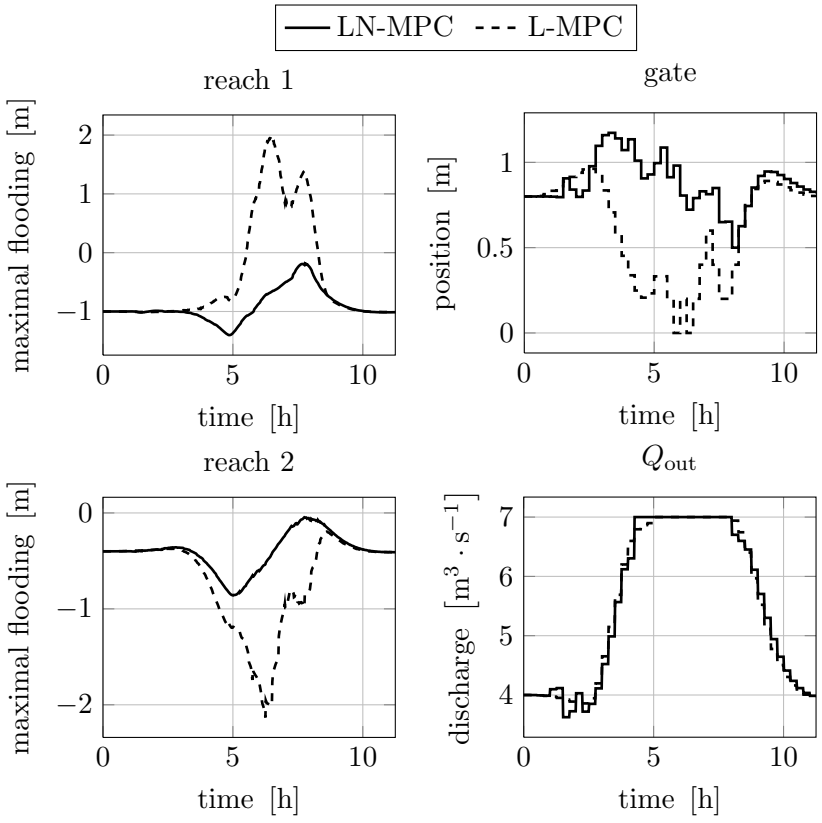


Figure 4.16: The maximal flooding of the first reach (top left) and the second reach (bottom left) together with the control actions for the gate position c (top right) and the downstream discharge Q_{out} (bottom right) applied with LN-MPC and L-MPC used for flood control of the two reaches. The LN-MPC succeeds in keeping the maximal flooding of both reaches below zero: there is no flooding. The control actions applied by the L-MPC controller results in very large floods for the first reach. The inaccurate approximation of the gate equation results in making the gate opening too small.

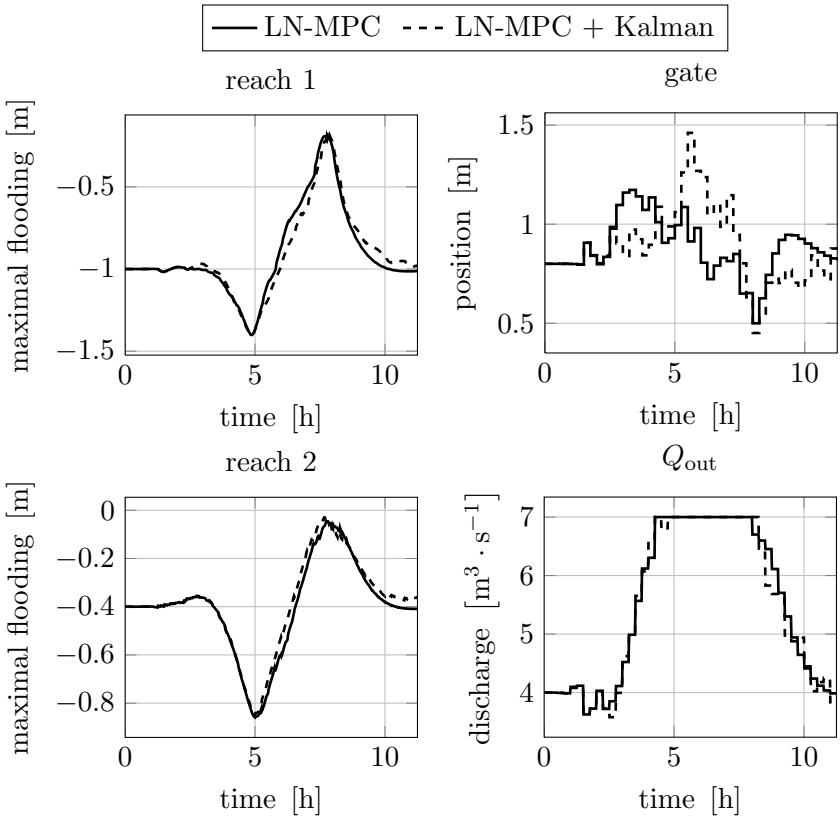


Figure 4.17: The maximal flooding of the first reach (top left) and the second reach (bottom left) together with the control actions for the gate position c (top right) and the downstream discharge Q_{out} (bottom right) applied with LN-MPC with full state information and LN-MPC in combination with a Kalman filter used for flood control of the two reaches. The difference between both controllers is very limited. Despite the limited number of measurements, the LN-MPC in combination with the Kalman filter succeeds in preventing both reaches from flooding. The remaining margin before a flood level is violated is slightly smaller for both reaches when a Kalman filter is needed.

Table 4.11: The set-point deviation cost J_h and the flood cost $J_{h_{\max}}$ obtained with LN-MPC with full state information and LN-MPC used in combination a Kalman filter controlling a river system consisting of two reaches connected to each other with a vertical sluice.

Simulation cost	LN-MPC	LN-MPC + Kalman
J_h [m]	810	758
$J_{h_{\max}}$ [m]	0	0

can prevent the reaches from flooding. The margin before flooding is slightly smaller when the Kalman filter is used. Also the difference for the downstream discharge Q_{out} is very small for both controllers. The biggest influence of the limited number of measurements can be seen in the control actions for the gate opening. These differences are caused by the estimation errors on the downstream water level of the gate (only the upstream water level of the gate is measured). However the effect on the control performance is small. The same conclusions can be made based on the set-point deviation cost J_h and flood cost $J_{h_{\max}}$ given in Table 4.11. Using the Kalman filter even lowers J_h with a small amount.

4.5 Model Predictive Control applied to a river system with a reservoir

Until now the examples used to explain and test MPC based on the LN-model were very simple. In this section an artificial test example will be constructed which resembles the upstream part of the Demer that will be controlled in the next chapter. This river system contains a reservoir, which the controller is only allowed to use for flood prevention. After introducing the river system and the approximate model used by the controller, the optimization problem will be formulated. The addition of this reservoir has its consequences for the optimization problem: extra terms need to be added to the cost function and a strategy is needed such that the buffer capacity is recovered in a fast way. Again a Kalman filter will be used for estimation the unknown states.

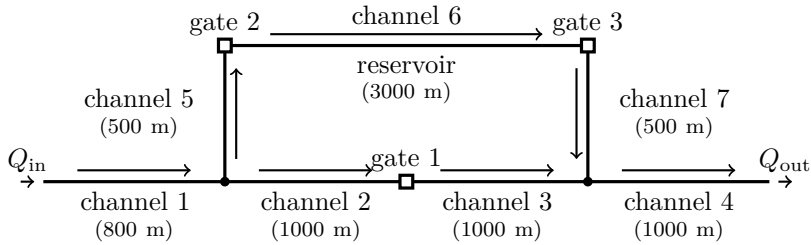


Figure 4.18: The river system to be prevented from flooding. The arrows indicate the general flow direction from upstream to downstream. Channels 1 to 4 form the main part of the river. Channels 5 and 7 connect the river with the reservoir (channel 6). The squares represent vertical sluices. The upstream inflow is given by Q_{in} while the outflow is determined by Q_{out}. The controller can control the position of the three sluices as well as the downstream discharge Q_{out}. The upstream discharge Q_{in} is a disturbance signal.

4.5.1 The test example

4.5.1.1 The river system and control objectives

Fig. 4.18 gives a schematic overview of the river system used to test the control performance. Channels 1 until 4 form the main part of the river while channels 5 and 7 connect the water reservoir (channel 6) with the river. The squares represent vertical sluices which can be used to control the discharges locally. $Q_{\text{in}}(t)$ is the discharge entering the river while $Q_{\text{out}}(t)$ corresponds with the discharge leaving the river. The discharge $Q_{\text{in}}(t)$ entering the river system is a disturbance signal. The controller can control the water levels with $Q_{\text{out}}(t)$ and the three gates. Every control variable has an associated upper and lower limit and the gates have a maximal rate of change constraint.

When there is no flooding risk, the controller should keep the water levels of the first and the fourth channel as close as possible to their set-points. During periods of heavy rainfall the controller needs to use the available buffer capacity of the reservoir in an optimal way. The controller should also empty the reservoir as fast as possible before focussing on set-point control again. Each channel has a safety limit and a flood limit. Only when the water levels risk violating their safety limits, the controller is allowed to use the buffer capacity of the water reservoir up to its own safety limit. Once this limit is reached, the channels are allowed to violate their safety levels. If the water levels risk violating also the flood limits, then all the available buffer capacity of the water reservoir can be used. The controller should also take into account that only a very limited number of water levels are measured in practice: only

the upstream and downstream water levels of every reach. It is assumed that the future disturbances are known at every time step.

4.5.1.2 Mathematical model

Based on the equations and boundary conditions defined in the previous chapter the mathematical model corresponding to this system consists of the following set of equations:

- the Saint-Venant equations modelling the dynamics of every channel individually (with $i = 1, \ldots, 7$):

$$\begin{cases} \dfrac{\partial A^{(i)}}{\partial h}\dfrac{\partial h^{(i)}}{\partial t} + \dfrac{\partial Q^{(i)}}{\partial z} = 0, \\ \dfrac{\partial Q^{(i)}}{\partial t} + \dfrac{\partial}{\partial z}\dfrac{{Q^{(i)}}^2}{A^{(i)}} + gA^{(i)}\dfrac{\partial h^{(i)}}{\partial z} + gA^{(i)}(S_{\mathrm{f}}^{(i)} - S_0^{(i)}) = 0, \end{cases}$$

- the gate equations as boundary conditions between channels 2 and 3, channels 5 and 6 and channels 6 and 7:

$$\begin{cases} Q^{(2)}(L^{(2)}, t) = \tilde{f}\big(c^{(1)}(t), h^{(2)}(L^{(2)}, t), h^{(3)}(0, t)\big), \\ Q^{(3)}(0, t) = \tilde{f}\big(c^{(1)}(t), h^{(2)}(L^{(2)}, t), h^{(3)}(0, t)\big), \end{cases}$$

$$\begin{cases} Q^{(5)}(L^{(5)}, t) = \tilde{f}\big(c^{(2)}(t), h^{(5)}(L^{(5)}, t), h^{(6)}(0, t)\big), \\ Q^{(6)}(0, t) = \tilde{f}\big(c^{(2)}(t), h^{(5)}(L^{(5)}, t), h^{(6)}(0, t)\big), \end{cases}$$

$$\begin{cases} Q^{(6)}(L^{(6)}, t) = \tilde{f}\big(c^{(3)}(t), h^{(6)}(L^{(6)}, t), h^{(7)}(0, t)\big), \\ Q^{(7)}(0, t) = \tilde{f}\big(c^{(3)}(t), h^{(6)}(L^{(6)}, t), h^{(7)}(0, t)\big), \end{cases}$$

  where the function $\tilde{f}$ is modelled with Eq. (3.9),

- the boundary conditions for all channels ending in or starting from a junction:

$$\begin{cases} h^{(1)}(L^{(1)}, t) = h^{(2)}(0, t), \\ h^{(1)}(L^{(1)}, t) = h^{(5)}(0, t), \\ Q^{(1)}(L^{(1)}, t) = Q^{(2)}(0, t) + Q^{(5)}(0, t), \end{cases}$$

$$\begin{cases} h^{(3)}(L^{(3)}, t) = h^{(4)}(0, t), \\ h^{(7)}(L^{(7)}, t) = h^{(4)}(0, t), \\ Q^{(3)}(L^{(3)}, t) + Q^{(7)}(L^{(7)}, t) = Q^{(4)}(0, t), \end{cases}$$

- and the following two last boundary conditions:

$$\begin{cases} Q^{(1)}(0,t) = Q_{\text{in}}(t), \\ Q^{(4)}(L^{(4)},t) = Q_{\text{out}}(t). \end{cases}$$

4.5.2 Approximate model

The approximate model that will be used inside the optimization problem is very similar as the one used in Section 4.4.2 and is derived for a given steady state of the river system:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{F}\mathbf{d}(k) + \boldsymbol{\beta},$$

with $\mathbf{x} \in \mathbb{R}^{n_x}$ the state vector containing all the water levels and discharges of the reaches, $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$, $\mathbf{F} \in \mathbb{R}^{n_x \times n_d}$, $\boldsymbol{\beta} \in \mathbb{R}^{n_x}$ the vector containing the information related to the linearization point, $\mathbf{u}(k) \in \mathbb{R}^4$ the input vector containing the downstream river discharge Q_{out} and the gate discharges $Q^{(1)}_{\text{gate}}$, $Q^{(2)}_{\text{gate}}$ and $Q^{(3)}_{\text{gate}}$, and $\mathbf{d} \in \mathbb{R}$ the disturbance signal Q_{in}.

4.5.3 Formulation of the optimization problem

The optimization problem that needs to be solved at every time step is based on the QP formulated in Section 4.4.3 and is defined as follows:

$$\min_{\mathbf{u},\mathbf{x},\boldsymbol{\xi},\boldsymbol{\zeta}} \sum_{j=1}^{N_{\text{P}}} \|\mathbf{x}(j) - \mathbf{r}_x(j)\|^2_{\mathbf{W}} + \sum_{j=0}^{N_{\text{P}}-1} \|\mathbf{u}(j) - \mathbf{u}(j-1)\|^2_{\mathbf{R}} + \\ + \|\boldsymbol{\xi}\|^2_{\mathbf{S}} + \mathbf{s}^{\text{T}}\boldsymbol{\xi} + \|\boldsymbol{\zeta}\|^2_{\mathbf{V}} + \mathbf{v}^{\text{T}}\boldsymbol{\zeta} \tag{4.17}$$

$$\text{s.t. } \mathbf{x}(0) = \hat{\mathbf{x}},$$

$$\mathbf{x}(j+1) = \mathbf{A}\mathbf{x}(j) + \mathbf{B}\mathbf{u}(j) + \mathbf{F}\mathbf{d}(j) + \boldsymbol{\beta}, \qquad j = 0, \ldots, N_{\text{P}} - 1$$

$$\underline{\mathbf{u}}(j) \leq \mathbf{u}(j) \leq \overline{\mathbf{u}}(j), \qquad j = 0, \ldots, N_{\text{P}} - 1 \tag{4.18}$$

$$|\mathbf{u}(j) - \mathbf{u}(j-1)| \leq \boldsymbol{\Delta}_u, \qquad j = 0, \ldots, N_{\text{P}} - 1$$

$$\mathbf{u}(-1) = \mathbf{u}_{\text{prev}},$$

$$\text{for } i = 1, \ldots, n_{\text{c}}:$$

$$\tilde{\mathbf{H}}^{(i)}\mathbf{x}(j) \leq \mathbf{h}^{(i)}_{\text{max},1} + \mathbf{1}_{n_h^{(i)}} \cdot \eta(j)\xi_i, \qquad j = 1, \ldots, N_{\text{P}} \tag{4.19}$$

$$\tilde{\mathbf{H}}^{(i)}\mathbf{x}(j) \leq \mathbf{h}^{(i)}_{\max,2} + \mathbf{1}_{n_h^{(i)}} \cdot \eta(j)\zeta_i, \qquad j = 1, \ldots, N_\mathrm{P} \tag{4.20}$$

$$\boldsymbol{\xi} \geq 0, \tag{4.21}$$

$$\boldsymbol{\zeta} \geq 0, \tag{4.22}$$

with N_P the prediction horizon, $\mathbf{W} \in \mathbb{R}^{n_x \times n_x} \succeq 0$, $\mathbf{R} \in \mathbb{R}^{n_u \times n_u} \succ 0$, $\mathbf{S} \in \mathbb{R}^{n_\mathrm{c} \times n_\mathrm{c}} \succ 0$ and $\mathbf{V} \in \mathbb{R}^{n_\mathrm{c} \times n_\mathrm{c}} \succ 0$ four diagonal weighting matrices, $\mathbf{s} \in \mathbb{R}^{n_\mathrm{c}}$ and $\mathbf{v} \in \mathbb{R}^{n_\mathrm{c}}$ two weighting vectors, n_c the total number of channels, $\mathbf{r}_x \in \mathbb{R}^{n_x}$ a (possible time-varying) vector containing the set-points for all the states, $\hat{\mathbf{x}} \in \mathbb{R}^{n_x}$ the current state of the process (measured or estimated), $\underline{\mathbf{u}} \in \mathbb{R}^{n_u}$ and $\overline{\mathbf{u}} \in \mathbb{R}^{n_u}$ the lower and upper limits on the inputs, $\boldsymbol{\Delta}_u \in \mathbb{R}^{n_u}$ a vector with the maximal allowed rate of change for the inputs, $\mathbf{u}_\mathrm{prev} \in \mathbb{R}^{n_u}$ the control action applied in the previous time step, $\tilde{\mathbf{H}}^{(i)} \in \mathbb{R}^{n_h^{(i)} \times (n_h + n_Q)}$ a matrix selecting the water levels of reach i from the state vector $\mathbf{x}$, $\mathbf{h}_{\max,1} \in \mathbb{R}^{n_h}$ the safety levels, $\mathbf{h}_{\max,2} \in \mathbb{R}^{n_h}$ the flood levels, $\eta(j) = 1/r_\mathrm{c}^{j-1}$ a time-dependent inverse weight (with $r_\mathrm{c} > 1$) and $\boldsymbol{\xi}$, $\boldsymbol{\zeta} \in \mathbb{R}^{n_\mathrm{c}}$ two vectors of slack variables related to the safety levels and flood levels respectively.

Most of the elements of this optimization problem have been explained in previous sections:

- The safety levels and flood levels are imposed as soft constraints with the slack variables $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ (Eqs. (4.19) and (4.20)) together with the addition of these slack variables in the objective function (Eq. (4.17)) and positivity constraints on these variables (Eqs. (4.21) and (4.22)). More information is given in Section 4.3.3.1. Notice that only one slack variable is used for every reach in contrast to the previous sections where one slack variable was used for every water level of every reach. This reduces the number of optimization variables and the complexity of the QP.
- Section 4.4.3.1 explains how the limits $\underline{\mathbf{c}}$, $\overline{\mathbf{c}}$ and $\boldsymbol{\Delta}_c$ can be translated to the time-varying limits $\underline{\mathbf{u}}$ and $\overline{\mathbf{u}}$ on the gate discharges (Eq. (4.18)). Algorithm 3 can easily be extended from the case of a single gate to the situation where multiple gates are present.
- Uncontrollability of the gates is avoided with the same approach discussed in Section 4.4.3.2.

4.5.3.1 The weight matrix S and the weight vector s

Until now there were only flood levels for the water levels. Section 4.3.3.2 explains how the weighting matrices $\mathbf{W}$, $\mathbf{R}$, $\mathbf{V}$ and the weight vector $\mathbf{v}$ have to be chosen such that the controller can be used for set-point control and flood

control. This reasoning can simply be extended to the situation where there are also safety levels present for the water levels. The elements in the matrix $\mathbf{V}$ and $\mathbf{v}$ should be sufficiently large compared to the matrices $\mathbf{W}$ and $\mathbf{R}$ in order to impose the flood levels as exact soft constraints. The same reasoning holds for the matrix $\mathbf{S}$ and the vector $\mathbf{s}$. Because the buffer capacity of the reservoir above the safety limit may not be used for keeping the water levels of all the other channels below their safety limits, the diagonal element of $\mathbf{S}$ and the element of $\mathbf{s}$ corresponding to the reservoir are set higher than the elements corresponding with the other channels. The diagonal elements of $\mathbf{V}$ and the elements of $\mathbf{v}$ are set higher than all the elements of $\mathbf{S}$ and $\mathbf{s}$: a violation of the flood levels is penalized more than the violation of the safety levels. Therefore the controller will use the remaining buffer capacity of the reservoir above the safety level for flood prevention of all channels. The reasoning for the matrices $\mathbf{W}$ and $\mathbf{R}$ is the same as before.

Remark. Another solution often used in industrial advanced process control software packages to deal with the difference in importance between the different safety and flood levels is to work with priorities. E.g. the flood levels will have a higher priority than the safety levels. In a first step a static optimization problem is solved looking for the optimal steady state targets. Starting with the most important constraints (i.e. the constraints with the highest priority), a series of QPs is solved adding every time the next priority class of constraints to the optimization problem until it becomes infeasible. Although multiple optimization problems have to be solved, the total computation time is limited: the QPs are much smaller than the original problem because we are only interested in the steady state targets. These optimal steady state targets are then used as new set-points in the dynamic optimization problem that has to be solved only once.

4.5.3.2 Buffer capacity recovery

After a period of heavy rainfall the controller should recover the buffer capacity to handle future rainfall without flooding. This can be achieved by working with two different weighting matrices $\mathbf{W}$ and reference signals $\mathbf{r}_x$. During normal operation (set-point control), the diagonal elements of the matrix $\mathbf{W}$ corresponding to the most important water levels will have the highest values in comparison to the diagonal elements corresponding to the other water levels and $\mathbf{r}_x$ is set equal to the desired set-points. However when some of the buffer capacity is used for flood prevention, the set-points of the water levels of the most downstream channel (channel 4) are set lower than the set-point of the reservoir and the diagonal elements of $\mathbf{W}$ corresponding to these water levels are increased. By decreasing these water levels, the third gate can be used to remove the excess of water in the reservoir. This change in $\mathbf{W}$ and $\mathbf{r}_x$ is

performed when one of the water levels in the reservoir is 20 cm above its set-point. This is checked every time a new state estimate of the process is computed. When the buffer capacity is recovered, $\mathbf{W}$ and $\mathbf{r}_x$ are set equal to their initial values (set-point control). This is done when all the water levels of the reservoir are at the most 10 cm above their set-point for at least 2.5 hours.

4.5.4 Simulation results

In this section we show the simulations results when LN-MPC with full state information and LN-MPC with a Kalman filter are applied to full hydrodynamic model of the river system visualized in Fig. 4.18. The sampling time of every controller is 15 min. A spatial discretization Δz of 50 m is used for all seven reaches. Initially the discharges in the main river part are all equal to 3 $\mathrm{m}^3 \cdot \mathrm{s}^{-1}$, while there is no flow in the reservoir and the two side channels. This means that the second and the third gate are closed. The most downstream water level of the second channel and of the fourth channel are equal to 5 m, while the most downstream water level of the reservoir is equal to 4 m. The set-points for all the water levels are equal to the initial water levels. The downstream discharge Q_{out} should remain between 0 $\mathrm{m}^3 \cdot \mathrm{s}^{-1}$ and 5 $\mathrm{m}^3 \cdot \mathrm{s}^{-1}$. All the gate openings should be between 0 m and 5 m and they can only be changed over a maximum distance of 20 cm every 15 min. The differences between the initial water levels and the safety levels for channels 1, 2 and 5 are equal to 0.6 m, for channels 3, 4 and 7 are equal to 0.6 m and for the reservoir are equal to 1.2 m. The differences with respect to the flood levels are 1 m, 0.8 m and 2 m respectively. Two tests will be performed. One for a relative small disturbance signal in combination with a small reservoir, and another one for a large disturbance signal together with a larger reservoir. Table 4.12 shows the parameters for the different reaches. The first value for the bottom width of the reservoir is used in the first test, the second value for the second test. The width of every gate is equal to 6 m.

Table 4.13 shows the elements of the weight matrices $\mathbf{W}$, $\mathbf{S}$, $\mathbf{V}$ and $\mathbf{R}$, and the entries of the weight vectors $\mathbf{s}$ and $\mathbf{v}$. As explained in the section describing the problem formulation, during normal operation the water levels of the channels 1 and 4 and of the reservoir are the most important ones. Therefore the weights of $\mathbf{W}$ corresponding to these water levels are set equal to 10, 1 and 1000 respectively. All the other weights for the other water levels are fixed to 0.001. The weights for the discharges for all the channels are set equal to 0.001. If the buffer capacity is used to prevent the water levels from violating their safety limits, the set-points for the most downstream water levels (channel 4) are set below the set-point of the reservoir and the corresponding weights in $\mathbf{W}$ are increased to 800 such that the buffer capacity can be recovered. All the diagonal elements of $\mathbf{R}$ are set equal to 175 for the gate discharges and 10

Table 4.12: Channel and gate parameters for the river system of Fig. 4.18. The first value for the bottom width of the reservoir is used when the performance of the controller is tested for a small reservoir, the second value for testing the performance on a large reservoir.

	channels			
	channel 1	channels 2-4	channels 5 & 7	reservoir
L [m]	800	1000	500	3000
S_0	0.0002	0.0002	0.0001	0.0001
n_{mann} $\left[\text{s} \cdot \text{m}^{-1/3}\right]$	0.02	0.02	0.02	0.02
B [m]	6	6	6	6/50
S_1, S_2	1	1	1	1

for Q_{out}, the weights in $\mathbf{S}$ and $\mathbf{s}$ for the slack variable $\boldsymbol{\xi}$ associated with the reservoir are set to 10^6, the weights associated with the other channels are set equal to 10^5, the weights in $\mathbf{V}$ and $\mathbf{v}$ for the slack variable $\boldsymbol{\zeta}$ are taken equal to 10^8. r_{c} is set to 1.2 and the size of the prediction window N_{P} is equal to 15. The following weight matrices were used to compute the Kalman gain matrix $\mathbf{L}$:

$$\mathbf{Q}_{\text{est}} = 10^{-3} \cdot \mathbf{I}_{n_x},$$

$$\mathbf{R}_{\text{est}} = 10^{-6} \cdot \mathbf{I}_{n_u}.$$

4.5.4.1 Small reservoir

The disturbance signal is depicted in Fig. 4.19. Initially the discharge entering the river system is equal to 3 $\text{m}^3 \cdot \text{s}^{-1}$, then after 4 hours it is increased to 12 $\text{m}^3 \cdot \text{s}^{-1}$ and finally after 6 hours it is decreased back to 3 $\text{m}^3 \cdot \text{s}^{-1}$.

Fig. 4.20 shows the evolution of the maximal difference between the water levels and their set-points for LN-MPC with and without the Kalman filter. In order to limit the number of plots, the results for the upstream part of the river (channels 1, 2 and 5) are plotted together (the top plot) as well as the results for the downstream part of the river (channels 3, 4 and 7, the middle plot). The bottom plot shows the results for the reservoir (channel 6). Fig. 4.21 shows the control actions applied to the three gates and $Q_{\text{out}}(t)$. The control actions as well as the water levels have a similar general trend for both controllers. The addition of the estimator in the control loop has only a limited effect on the control performance. There is only a noticeable difference in the

Table 4.13: Diagonal elements of the weight matrices $\mathbf{W}$, $\mathbf{S}$, $\mathbf{V}$ and $\mathbf{R}$, and the entries of the weight vectors $\mathbf{s}$ and $\mathbf{v}$ for controlling a river system with a reservoir.

	channel 1	channel 2	channel 3	channel 4	channel 5	reservoir	channel 7
$\mathbf{W} \in \mathbb{R}^{333\times333}$							
water levels	$10{\cdot}\mathbf{1}_{17}$	$0.001{\cdot}\mathbf{1}_{21}$	$0.001{\cdot}\mathbf{1}_{21}$	$1{\cdot}\mathbf{1}_{21}$ $800{\cdot}\mathbf{1}_{21}$ [a]	$0.001{\cdot}\mathbf{1}_{11}$	$1000{\cdot}\mathbf{1}_{61}$	$0.001{\cdot}\mathbf{1}_{11}$
discharges	$0.001{\cdot}\mathbf{1}_{18}$	$0.001{\cdot}\mathbf{1}_{22}$	$0.001{\cdot}\mathbf{1}_{22}$	$0.001{\cdot}\mathbf{1}_{22}$	$0.001{\cdot}\mathbf{1}_{12}$	$0.001{\cdot}\mathbf{1}_{62}$	$0.001{\cdot}\mathbf{1}_{12}$
$\mathbf{S} \in \mathbb{R}^{7\times7}$							
safety levels	10^5	10^5	10^5	10^5	10^5	10^6	10^5
$\mathbf{s} \in \mathbb{R}^{7\times1}$							
safety levels	10^5	10^5	10^5	10^5	10^5	10^6	10^5
$\mathbf{V} \in \mathbb{R}^{7\times7}$							
flood levels	10^8	10^8	10^8	10^8	10^8	10^8	10^8
$\mathbf{v} \in \mathbb{R}^{7\times1}$							
flood levels	10^8	10^8	10^8	10^8	10^8	10^8	10^8

	$Q_{\text{gate}}^{(1)}$	$Q_{\text{gate}}^{(2)}$	$Q_{\text{gate}}^{(3)}$	Q_{out}
$\mathbf{R} \in \mathbb{R}^{4\times4}$				
control actions	175	175	175	10

[a] Value used to recover the buffer capacity of the reservoir.

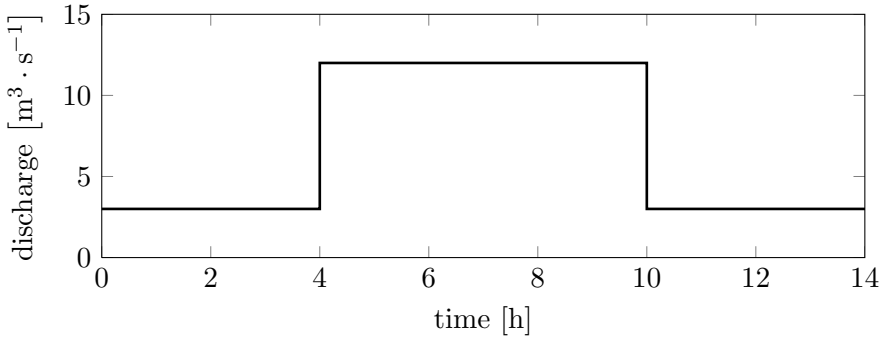


Figure 4.19: Disturbance signal Q_{in} used to test the performance of LN-MPC when applied to a river system with a small reservoir.

control actions for the two controllers during the period of heavy rainfall and right after this period. However since the trend of the control actions for the two controllers is similar, the water levels show the same behavior. Initially the controllers try to avoid to use the water reservoir and react on the future rainfall by decreasing the upstream water levels. This is done by increasing the gate opening of the first gate and increasing the discharge at the end of the river towards its maximal value. Once the water levels risk violating the safety levels, the controllers start using the water reservoir by opening the second and the third gate. Because of the magnitude of the upstream disturbance, the controllers cannot avoid the violation of the safety levels for the upstream part of the river, however they can prevent the violation of the flood levels or in the worst case minimize their violation. Table 4.14 shows the maximal difference between the water levels of each channel and their flood levels for both controllers. A positive value corresponds to the maximal violation of the flood levels, a negative value corresponds to the minimal margin before a flood level is violated. There is almost no difference between both controllers. The highest flooding is less than 9 mm without and with a Kalman filter. The different floods last for less than 2 minutes. The water levels of the downstream part can be kept below their safety levels at all times. Once the disturbance is decreased again, the controllers first steer the upstream water levels below their safety limits. Afterwards they keep these water levels close to the safety limits (the opening of the first gate is decreased and the second gate is completely closed) while the downstream water levels are further decreased below their initial set-point because of the change in the matrix $\mathbf{W}$ and the set-point $\mathbf{r}_x$. To decrease these water levels as fast as possible the controllers keep the downstream discharge at its maximal value. Once the buffer capacity is recovered, the matrix $\mathbf{W}$ and the set-point $\mathbf{r}_x$ are set to their initial value. The controllers close the second and the third gate to prevent the water levels

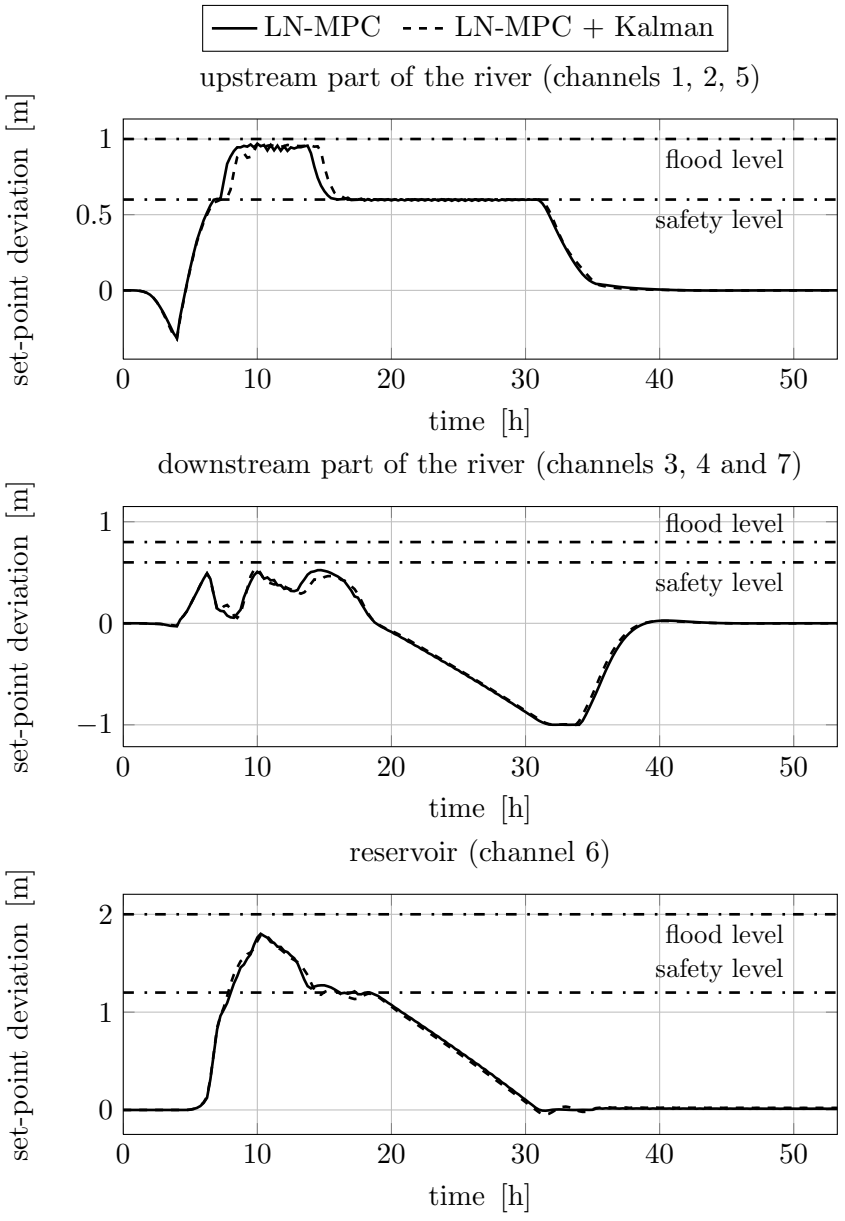


Figure 4.20: Evolution of the maximal difference between water levels and their set-points for LN-MPC and LN-MPC in combination with the Kalman filter. The top plot shows the maximal deviation of the water levels of channels 1, 2 and 5 at every time step, the middle plot the maximal deviation of the water levels of channels 3, 4 and 7 and the bottom plot the maximal deviation of the water levels of channel 6. The flood levels correspond with the highest horizontal lines, the safety levels with the lowest ones.

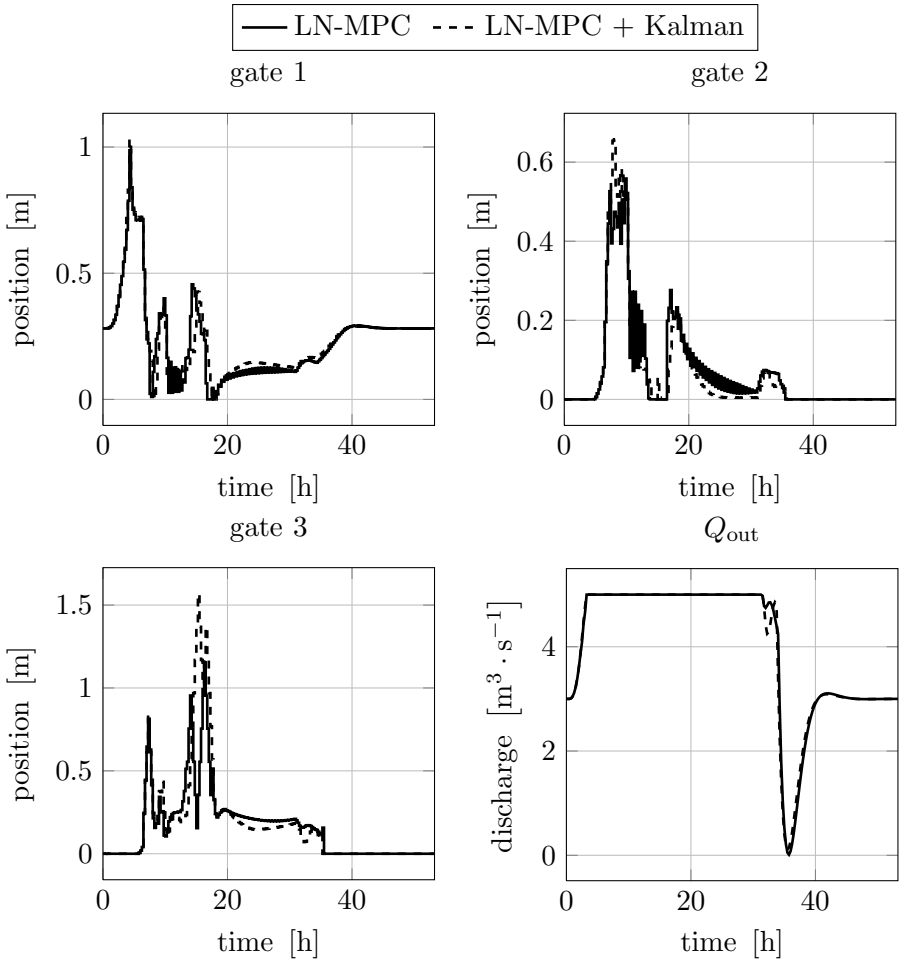


Figure 4.21: Control actions of the LN-MPC with and without the Kalman filter. The top plots and the bottom left plot show the evolution of the three gates and the bottom right plot shows the discharge $Q_{\text{out}}(t)$ at the end of the river system. There is almost no influence on the resulting control actions when a state estimator is included in the control loop.

Table 4.14: Maximal flooding of each channel for the two controllers in meters. A negative value means that there is no flooding and some buffer capacity is left for that channel.

	LN-MPC	LN-MPC + Kalman
channel 1 [m]	0.0081	0.0090
channel 2 [m]	0.0018	0.0030
channel 3 [m]	-0.2735	-0.2591
channel 4 [m]	-0.2738	-0.2583
channel 5 [m]	-0.0080	-0.0126
channel 6 [m] (reservoir)	-0.1992	-0.2057
channel 7 [m]	-0.2742	-0.2582

in the reservoir to increase. Immediately the controllers bring the upstream water levels as well as the downstream water levels to their (original) set-point by opening the first gate and changing $Q_{\text{out}}(t)$. Table 4.15 shows the average deviation of the water levels of every channel from their set-points at the end of the simulation. The controllers succeed in steering the water levels back to their set-points. The largest deviations are for the reservoir with 1 cm and 2 cm for LN-MPC used without and with the Kalman filter respectively. However these differences are negligible. The same conclusions can be made by comparing the set-point deviation cost J_h and the flood cost $J_{h_{\max}}$ for LN-MPC with full state information and LN-MPC in combination with the Kalman filter given in Table 4.16. Both costs are practically the same for both controllers: the addition of the Kalman filter has almost no influence on the control performance of the predictive controller.

Table 4.17 contains the mean, the minimal and the maximal computation time needed for computing the control actions for each step (prediction, optimization, conversion) separately and the total time. Also the timings for the Kalman filter are given. The time needed by the LN-MPC at every time step is dominated by the time needed to solve the optimization problem. Solving the QP during the period of the large disturbance signal takes the most time because more constraints become active (the safety levels, flood levels and upper limit on Q_{out}). The time needed by the estimator at every time step is negligible because only a limited number of matrix-vector multiplications are required.

Table 4.15: Average deviation of the water levels from their set-points at the end of the simulation.

	LN-MPC	LN-MPC + Kalman
channel 1 [m]	$1.51{\cdot}10^{-5}$	$1.10{\cdot}10^{-5}$
channel 2 [m]	$1.54{\cdot}10^{-5}$	$1.10{\cdot}10^{-5}$
channel 3 [m]	$2.31{\cdot}10^{-5}$	$7.09{\cdot}10^{-6}$
channel 4 [m]	$2.27{\cdot}10^{-5}$	$6.88{\cdot}10^{-6}$
channel 5 [m]	$1.56{\cdot}10^{-5}$	$1.14{\cdot}10^{-5}$
channel 6 [m] (reservoir)	0.01	0.02
channel 7 [m]	$2.24{\cdot}10^{-5}$	$6.48{\cdot}10^{-6}$

Table 4.16: The set-point deviation cost J_h and the flood cost $J_{h_{\max}}$ obtained with LN-MPC without and with a Kalman filter.

Simulation cost	LN-MPC	LN-MPC + Kalman
J_h [m]	17590	17591
$J_{h_{\max}}$ [m]	0.010	0.012

Table 4.17: Average, minimal and maximal computation time (in seconds) needed for each step separately of the controllers, for all steps together performed by the controllers, and for the estimation step performed by the Kalman filter. The simulation was performed on a PC with a 2.8 GHz Intel Core i7 CPU and 8 GB of RAM. The QP is solved with `quadprog` of Matlab [123].

	LN-MPC			LN-MPC + Kalman		
	mean	min	max	mean	min	max
prediction [s]	0.089	0.047	0.140	0.088	0.050	0.102
optimization [s]	4.747	3.270	8.282	4.633	3.306	7.791
conversion [s]	0.009	0.006	0.012	0.008	0.006	0.011
total [s]	4.844	3.363	8.367	4.730	3.399	7.878
estimation [s]	–	–	–	< 0.001	< 0.001	0.006

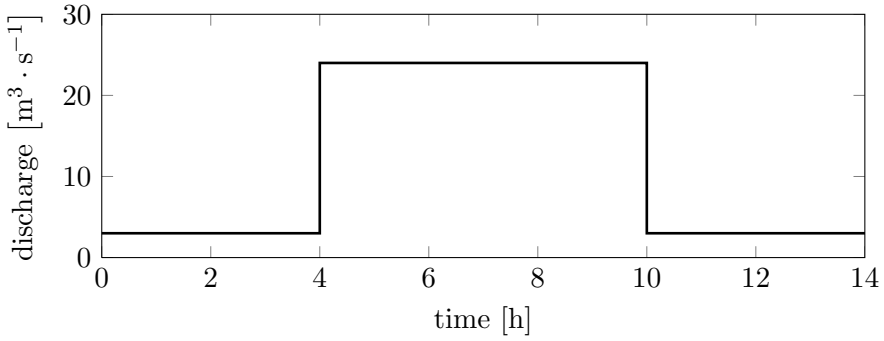


Figure 4.22: Disturbance signal Q_{in} used to test the performance of LN-MPC applied to a river system with a large reservoir.

4.5.4.2 Large reservoir

A similar test has been conducted for a larger reservoir in combination with a larger disturbance signal (Fig. 4.22). This time Q_{in} increases after 4 hours to 24 $\text{m}^3 \cdot \text{s}^{-1}$ and it decreases back to 3 $\text{m}^3 \cdot \text{s}^{-1}$ 6 hours later.

The results are very similar to the ones with the small reservoir. Fig. 4.23 shows the evolution of the maximal difference between the water levels and their set-points for LN-MPC with and without the Kalman filter for the upstream part of the river, the downstream part and the reservoir. The control actions are visualized in Fig. 4.24. There is again almost no difference between the results without and with a Kalman filter. The controllers first create extra buffer capacity by decreasing the water levels of the upstream part of the reach. This is done by opening the first gate and increasing the downstream discharge Q_{out} towards its maximal value. Once there is a risk of violating the safety levels and flood levels, the controllers open the second and the third gate to start using the buffer capacity of the reservoir. Table 4.18 shows the maximal flooding for the different channels for both controllers. LN-MPC without the Kalman filter succeeds in keeping all the water levels at all times below their flood levels. There is a flooding of the first, second and fifth channel when the Kalman filter is included in the control loop. However these maximal floods are less than 3.8 cm and last for less than 10 minutes. Afterwards both controllers bring the water levels below their safety levels and they start emptying the water reservoir. Once the buffer capacity of the reservoir is recovered, the water levels of the other channels are steered back to their set-points. Table 4.19 shows the average deviation between the water levels and their set-points for every channel at the end of the simulation. These deviations are small enough for both controllers. Table 4.20 contains the set-point deviation cost J_h and the flood cost $J_{h_{\max}}$ for both controllers. The flood cost is equal to zero for the case

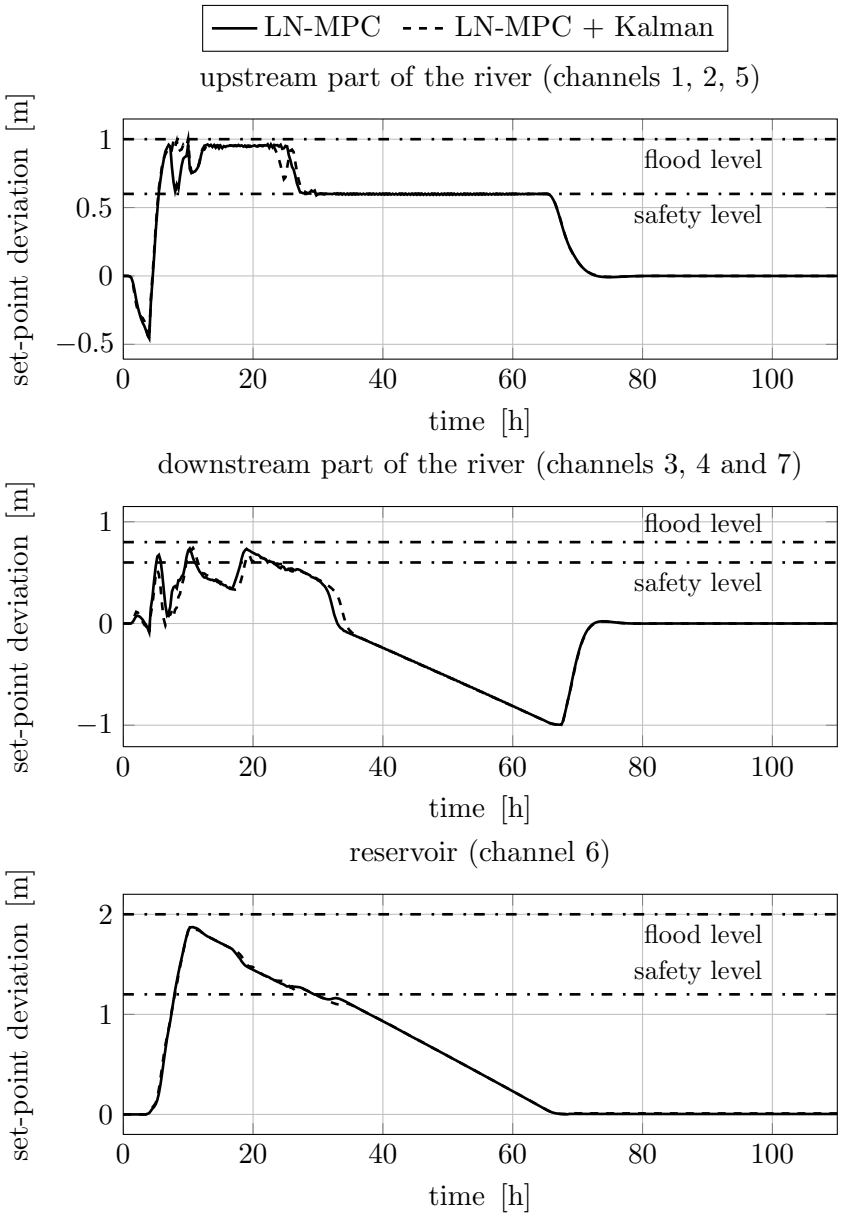


Figure 4.23: Evolution of the maximal difference between the water levels and their set-points for LN-MPC and LN-MPC in combination with the Kalman filter. The top plot shows the maximal deviation of the water levels of channels 1, 2 and 5 at every time step, the middle plot the maximal deviation of the water levels of channels 3, 4 and 7 and the bottom plot the maximal deviation of the water levels of channel 6. The flood levels correspond with the highest horizontal lines, the safety levels with the lowest ones.

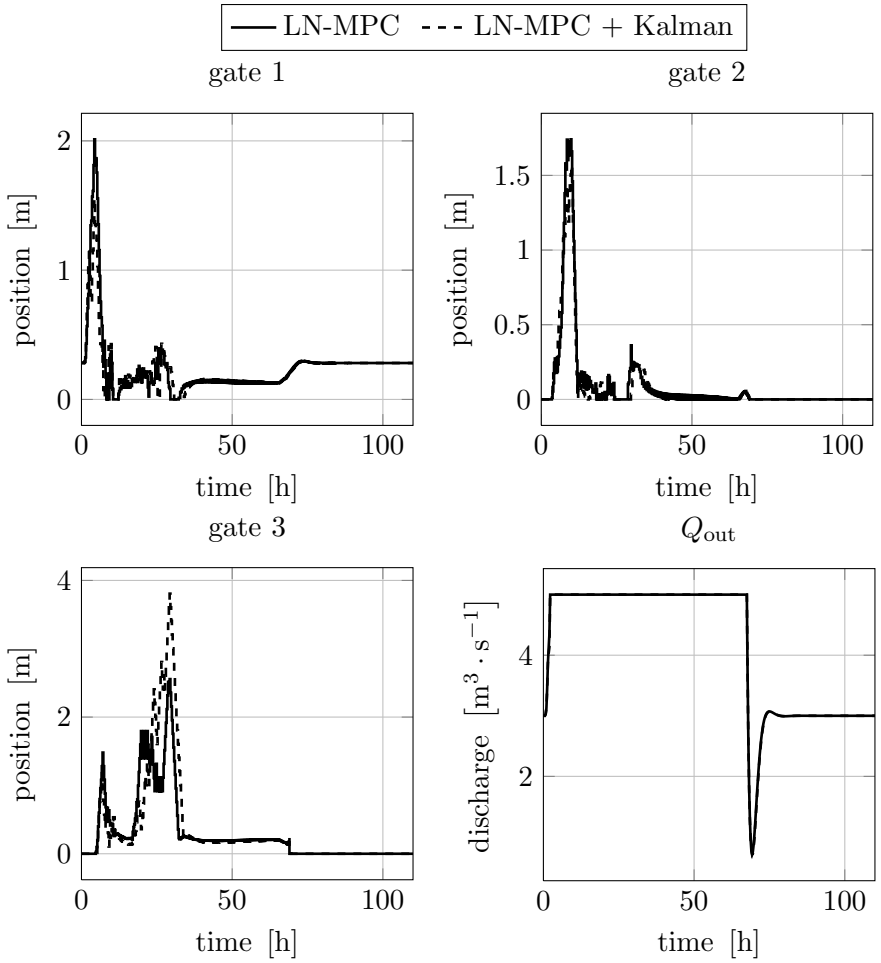


Figure 4.24: Control actions of the LN-MPC with and without the Kalman filter. The top plots and the bottom left plot show the evolution of the three gates and the bottom right plot shows the discharge $Q_{\text{out}}(t)$ at the end of the river system.

Table 4.18: Maximal flooding of each channel for the two controllers. A negative value means that there is no flooding and some buffer capacity is left.

	LN-MPC	LN-MPC + Kalman
channel 1 [m]	-0.0251	0.0174
channel 2 [m]	-0.0083	0.0371
channel 3 [m]	-0.0546	-0.0596
channel 4 [m]	-0.0502	-0.0565
channel 5 [m]	-0.0258	0.0214
channel 6 [m] (reservoir)	-0.1248	-0.1318
channel 7 [m]	-0.0720	-0.0637

Table 4.19: Average deviation of the water levels from their set-points at the end of the simulation.

	LN-MPC	LN-MPC + Kalman
channel 1 [m]	$-6.24 \cdot 10^{-5}$	$-7.54 \cdot 10^{-5}$
channel 2 [m]	$-6.24 \cdot 10^{-5}$	$-7.54 \cdot 10^{-5}$
channel 3 [m]	$6.40 \cdot 10^{-6}$	$-1.07 \cdot 10^{-5}$
channel 4 [m]	$6.42 \cdot 10^{-6}$	$-1.07 \cdot 10^{-5}$
channel 5 [m]	$-6.24 \cdot 10^{-5}$	$-7.54 \cdot 10^{-5}$
channel 6 [m] (reservoir)	$4.34 \cdot 10^{-3}$	$9.67 \cdot 10^{-3}$
channel 7 [m]	$6.42 \cdot 10^{-6}$	$-1.07 \cdot 10^{-5}$

of full state information because all the reaches can be prevented from flooding. This is not the case with the Kalman filter resulting in a flood cost different from zero. However this value is still very small. The set-point deviation cost is the same for both controllers. Both simulation costs indicate that the influence of the Kalman filter on the control performance is very limited.

Table 4.21 shows the timing results for both controllers. The same conclusions can be made as for LN-MPC applied to the small reservoir. Most time is spent for solving the optimization problem. The time required for the other steps can be neglected.

Table 4.20: The set-point deviation cost J_h and the flood cost $J_{h_{\max}}$ obtained with LN-MPC with full state information and LN-MPC in combination with a Kalman filter.

Simulation cost	LN-MPC	LN-MPC + Kalman
J_h [m]	39400	39400
$J_{h_{\max}}$ [m]	0	0.221

Table 4.21: Average, minimal and maximal computation time (in seconds) needed for each step separately and all of them together by the controllers and the state estimator at every time step. The simulation was performed on a PC with a 2.8 GHz Intel Core i7 CPU and 8 GB of RAM. The QP is solved with `quadprog` of Matlab [123].

	LN-MPC			LN-MPC + Kalman		
	mean	min	max	mean	min	max
prediction [s]	0.088	0.047	0.123	0.087	0.051	0.137
optimization [s]	4.987	3.369	9.552	4.964	3.518	10.437
conversion [s]	0.006	0.006	0.009	0.006	0.006	0.015
total [s]	5.083	3.460	9.637	5.060	3.610	10.527
estimation [s]	–	–	–	< 0.001	< 0.001	0.009

4.6 Implementation aspects & speed-up of the controller

It can be shown that the optimization problem given by Eqs. (4.17)-(4.22) can be rewritten into the following general formulation of a QP [18]:

$$\min_{\mathbf{z}} 0.5\,\mathbf{z}^{\mathrm{T}}\mathbf{H}\mathbf{z} + \mathbf{f}^{\mathrm{T}}\mathbf{z}$$

$$\text{s.t. } \mathbf{A}_{\mathrm{in}}\mathbf{z} \leq \mathbf{b}_{\mathrm{in}},$$

$$\mathbf{A}_{\mathrm{eq}}\mathbf{z} = \mathbf{b}_{\mathrm{eq}},$$

with $\mathbf{z} \in \mathbb{R}^{n_z}$ the vector containing the n_z optimization variables, the Hessian $\mathbf{H} \in \mathbb{R}^{n_z \times n_z}$, $\mathbf{f} \in \mathbb{R}^{n_z}$, $\mathbf{A}_{\text{in}} \in \mathbb{R}^{n_{\text{in}} \times n_z}$ and $\mathbf{b}_{\text{in}} \in \mathbb{R}^{n_{\text{in}}}$ the matrix and the vector defining the n_{in} linear inequality constraints, $\mathbf{A}_{\text{eq}} \in \mathbb{R}^{n_{\text{eq}} \times n_z}$ and $\mathbf{b}_{\text{eq}} \in \mathbb{R}^{n_{\text{eq}}}$ the matrix and vector defining the n_{eq} linear equality constraints.

In the expanded or uncondensed implementation of the QP, the vector $\mathbf{z}$ is formed by stacking the states and inputs at every time instant of the prediction horizon, together with the slack variables:

$$\mathbf{z} = [\mathbf{x}(1); \mathbf{x}(2); \ldots; \mathbf{x}(N_{\text{P}}); \mathbf{u}(0); \ldots; \mathbf{u}(N_{\text{P}} - 1); \boldsymbol{\xi}; \boldsymbol{\zeta}].$$

The model equations are translated into the matrix $\mathbf{A}_{\text{eq}}$ and the vector $\mathbf{b}_{\text{eq}}$, while the flood levels, the safety levels, the time-varying lower and upper limits and the rate of change constraint of the inputs and the positivity constraints on the slack variables are contained in the matrix $\mathbf{A}_{\text{in}}$ and the vector $\mathbf{b}_{\text{in}}$. The Hessian $\mathbf{H}$ is a block diagonal matrix with one block for the states, the inputs and the two slack variables respectively. The diagonal blocks related to the states and the slack variables have a diagonal structure while the block related to the inputs contains elements on the main diagonal and on the n_uth and $-n_u$th diagonal (because the difference between consecutive inputs $\mathbf{u}(k) - \mathbf{u}(k-1)$ is weighted). It can be shown that the matrix $\mathbf{A}_{\text{eq}}$ only depends on the state space matrices $\mathbf{A}$ and $\mathbf{B}$ and is independent of $\hat{\mathbf{x}}$, $\boldsymbol{\beta}$, $\mathbf{F}$ and the disturbances $\mathbf{d}(k)$. This means that every time the predictive controller has to solve the QP, $\mathbf{A}_{\text{eq}}$ is the same in contrast to $\mathbf{b}_{\text{eq}}$. Also the matrices $\mathbf{A}_{\text{in}}$ and $\mathbf{H}$ remain unchanged, while the vectors $\mathbf{b}_{\text{in}}$ and $\mathbf{f}$ change at every time step. This property has been exploited in the implementation of the controller by constructing the matrices $\mathbf{A}_{\text{eq}}$, $\mathbf{A}_{\text{in}}$ and $\mathbf{H}$ only once [29]. This results in much smaller computation times than if they would be rebuilt for every iteration.

The computation time needed to solve the QP for the river system is still small compared to the sampling time of the controller (15 min). However this computation time can quickly increase if we want to apply our controller to more complex river systems with more variables and/or if we need to use a longer prediction horizon. As seen in the previous section, on average 98 % of the total computation time needed by the controller at every time step is spent by solving the optimization problem. Some strategies will now be discussed that can result in a decrease in the computation time. The first strategies focus on decreasing the number of optimization variables, the other strategies on reducing the number of inequality constraints.

Remark. The author does not claim that al the mentioned strategies will always result in a reduction in computation time. Whether a reduction is achieved as well as the amount of reduction depends on the specific solver used for solving the QP. E.g. the approaches to decrease the number of optimization variables will reduce the sparsity structure of the QP. State-of-the-art QP solvers, like `cplexqp` of IBM [76], that exploit this sparsity

can have larger computation times after reducing the number of optimization variables. However, in combination with an approach to also reduce the number of inequality constraints, it can still be possible to decrease the computation time.

4.6.1 Reducing the number of optimization variables

The total number of optimization variables is very large due to the large number of water levels and discharges. A first solution is to use POD and Galerkin projection to reduce the order of the model. This will reduce the optimization variables and at the same time decrease the number of equality constraints. Another (and better) solution is to eliminate the equality constraints and the water levels and discharges completely from the optimization problem by using a condensed implementation.

4.6.1.1 Model reduction with POD

Reducing the model order of the linear state space model will decrease the total number of optimization variables and decrease the number of equality constraints. In a similar fashion as it was done in Section 3.6.2.2, POD and Galerkin projection will be used for reducing the order of the linear state space model.

POD is used to find the $\tilde{n}_n$th order approximation of $\Delta\mathbf{h}(k) = \mathbf{h}(k) - \mathbf{h}_{\text{ss}}$ and the $\hat{n}_Q$th order approximation of $\Delta\mathbf{q}(k) = \mathbf{q}(k) - \mathbf{q}_{\text{ss}}$:

$$\Delta\mathbf{h}(k) \approx \Delta\mathbf{h}_{\tilde{n}_h}(k) = \sum_{j=1}^{\tilde{n}_h} a_j(k)\tilde{\boldsymbol{\varphi}}_j = \tilde{\boldsymbol{\Phi}}_{\tilde{n}_h}\mathbf{a}(k),\ \tilde{n}_h \ll n_h,$$

$$\Delta\mathbf{q}(k) \approx \Delta\mathbf{q}_{\hat{n}_Q}(k) = \sum_{j=1}^{\hat{n}_Q} b_j(k)\hat{\boldsymbol{\varphi}}_j = \hat{\boldsymbol{\Phi}}_{\hat{n}_Q}\mathbf{b}(k),\ \hat{n}_Q \ll n_Q,$$

where $\tilde{\boldsymbol{\varphi}}_j \in \mathbb{R}^{n_h}$, for $j = 1, \ldots, \tilde{n}_h$ and $\hat{\boldsymbol{\varphi}}_j \in \mathbb{R}^{n_Q}$, for $j = 1, \ldots, \hat{n}_Q$ are the $\tilde{n}_h$ and $\hat{n}_Q$ most relevant orthonormal basis vectors, $\tilde{\boldsymbol{\Phi}}_{\tilde{n}_h} = \left[\tilde{\boldsymbol{\varphi}}_1, \ldots, \tilde{\boldsymbol{\varphi}}_{\tilde{n}_h}\right] \in \mathbb{R}^{n_h \times \tilde{n}_h}$, $\mathbf{a}(k) = \left[a_1(k), \ldots, a_{\tilde{n}_h}(k)\right]^{\mathrm{T}} \in \mathbb{R}^{\tilde{n}_h}$, $\hat{\boldsymbol{\Phi}}_{\hat{n}_Q} = \left[\hat{\boldsymbol{\varphi}}_1, \ldots, \hat{\boldsymbol{\varphi}}_{\hat{n}_Q}\right] \in \mathbb{R}^{n_Q \times \hat{n}_Q}$ and $\mathbf{b}(k) = \left[b_1(k), \ldots, b_{\hat{n}_Q}(k)\right]^{\mathrm{T}} \in \mathbb{R}^{\hat{n}_Q}$. The Galerkin projection performed on the linear state space model results in the following reduced order model describing the dynamics of the POD coefficients $\mathbf{a}$ and $\mathbf{b}$:

$$\begin{bmatrix} \mathbf{a}(k+1) \\ \mathbf{b}(k+1) \end{bmatrix} = \mathbf{A}_{\text{r}} \begin{bmatrix} \mathbf{a}(k) \\ \mathbf{b}(k) \end{bmatrix} + \mathbf{B}_{\text{r}}\Delta\mathbf{u}(k) + \mathbf{F}_{\text{r}}\Delta\mathbf{d}(k)$$

with $\mathbf{A}_\mathrm{r} = \mathbf{\Phi}_n^\mathrm{T}\mathbf{A}\mathbf{\Phi}_n \in \mathbb{R}^{(\tilde{n}_h+\hat{n}_Q)\times(\tilde{n}_h+\hat{n}_Q)}$, $\mathbf{B}_\mathrm{r} = \mathbf{\Phi}_n^\mathrm{T}\mathbf{B} \in \mathbb{R}^{(\tilde{n}_h+\hat{n}_Q)\times n_u}$, $\mathbf{F}_\mathrm{r} = \mathbf{\Phi}_n^\mathrm{T}\mathbf{F} \in \mathbb{R}^{(\tilde{n}_h+\hat{n}_Q)\times n_d}$, $\Delta\mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}_\mathrm{ss}$, $\Delta\mathbf{d}(k) = \mathbf{d}(k) - \mathbf{d}_\mathrm{ss}$ and

$$\mathbf{\Phi}_n = \begin{bmatrix} \tilde{\mathbf{\Phi}}_{\tilde{n}_h} & \\ & \hat{\mathbf{\Phi}}_{\hat{n}_Q} \end{bmatrix}.$$

More information on the derivation of the POD basis vectors and a criterion to determine a value for $\tilde{n}_h$ and $\hat{n}_Q$ are given in Section 2.5.

This reduced order model can be used inside the optimization problem resulting in the following QP:

$$\min_{\substack{\mathbf{u},\mathbf{a},\mathbf{b},\\ \boldsymbol{\xi},\boldsymbol{\zeta}}} \sum_{j=1}^{N_\mathrm{P}} \left\| \begin{bmatrix} \mathbf{a}(j) \\ \mathbf{b}(j) \end{bmatrix} - \mathbf{\Phi}_n^\mathrm{T}(\mathbf{r}_x(j) - \mathbf{x}_\mathrm{ss}) \right\|^2_{\mathbf{\Phi}_n^\mathrm{T}\mathbf{W}\mathbf{\Phi}_n} +$$

$$+ \sum_{j=0}^{N_\mathrm{P}-1} \|\mathbf{u}(j) - \mathbf{u}(j-1)\|^2_\mathbf{R} + \|\boldsymbol{\xi}\|^2_\mathbf{S} + \mathbf{s}^\mathrm{T}\boldsymbol{\xi} + \|\boldsymbol{\zeta}\|^2_\mathbf{V} + \mathbf{v}^\mathrm{T}\boldsymbol{\zeta}$$

$$\text{s.t.} \quad \begin{bmatrix} \mathbf{a}(0) \\ \mathbf{b}(0) \end{bmatrix} = \mathbf{\Phi}_n^\mathrm{T}(\hat{\mathbf{x}} - \mathbf{x}_\mathrm{ss}),$$

$$\begin{bmatrix} \mathbf{a}(j+1) \\ \mathbf{b}(j+1) \end{bmatrix} = \mathbf{A}_\mathrm{r} \begin{bmatrix} \mathbf{a}(j) \\ \mathbf{b}(j) \end{bmatrix} + \mathbf{B}_\mathrm{r}\Delta\mathbf{u}(j) + \mathbf{F}_\mathrm{r}\Delta\mathbf{d}(j), \quad j = 0,\ldots,N_\mathrm{P}-1$$

$$\underline{\mathbf{u}}(j) \le \mathbf{u}(j) \le \overline{\mathbf{u}}(j), \qquad j = 0,\ldots,N_\mathrm{P}-1$$

$$|\mathbf{u}(j) - \mathbf{u}(j-1)| \le \boldsymbol{\Delta}_u, \qquad j = 0,\ldots,N_\mathrm{P}-1$$

$$\mathbf{u}(-1) = \mathbf{u}_\mathrm{prev},$$

$$\text{for } \; i = 1,\ldots,n_\mathrm{c}:$$

$$\tilde{\mathbf{H}}^{(i)}\mathbf{\Phi}_{\tilde{n}_h}^\mathrm{T}\mathbf{a}(j) + \mathbf{h}_\mathrm{ss}^{(i)} \le \mathbf{h}_{\mathrm{max},1}^{(i)} + \mathbf{1}_{n_h^{(i)}} \cdot \eta(j)\xi_i, \qquad j = 1,\ldots,N_\mathrm{P}$$

$$\tilde{\mathbf{H}}^{(i)}\mathbf{\Phi}_{\tilde{n}_h}^\mathrm{T}\mathbf{a}(j) + \mathbf{h}_\mathrm{ss}^{(i)} \le \mathbf{h}_{\mathrm{max},2}^{(i)} + \mathbf{1}_{n_h^{(i)}} \cdot \eta(j)\zeta_i, \qquad j = 1,\ldots,N_\mathrm{P}$$

$$\boldsymbol{\xi} \ge 0,$$

$$\boldsymbol{\zeta} \ge 0.$$

The number of optimization variables is decreased from $N_\mathrm{P}\cdot(n_h + n_Q + n_u) + 2\cdot n_c$ to $N_\mathrm{P}\cdot(\tilde{n}_h + \hat{n}_Q + n_u) + 2\cdot n_c$ and the number of equality constraints

decreases from $N_\mathrm{P} \cdot (n_h + n_Q)$ to $N_\mathrm{P} \cdot (\tilde{n}_h + \hat{n}_Q)$. The disadvantage is that the diagonal structure of the matrix $\mathbf{W}$ is destroyed by the multiplication with the dense matrices $\mathbf{\Phi}_n$ and $\mathbf{\Phi}_n^\mathrm{T}$. Also the structure of the state space matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{F}$ is destroyed for the same reason when they are replaced with the reduced order state space matrices $\mathbf{A}_\mathrm{r}$, $\mathbf{B}_\mathrm{r}$ and $\mathbf{F}_\mathrm{r}$.

4.6.1.2 Condensed implementation

The water levels and discharges, and the equality constraints coming from the linear state space model can be completely eliminated from the optimization problem by writing the states at every time step as a function of the current state estimate $\hat{\mathbf{x}}$, the control actions $\mathbf{u}(0), \ldots, \mathbf{u}(N_\mathrm{P} - 1)$, the disturbances $\mathbf{d}(0), \ldots, \mathbf{d}(N_\mathrm{P} - 1)$ and the vector $\boldsymbol{\beta}$ [29]. This results in the following equations:

$$\begin{bmatrix} \mathbf{x}(1) \\ \mathbf{x}(2) \\ \vdots \\ \mathbf{x}(N_\mathrm{P}) \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \vdots \\ \mathbf{A}^{N_\mathrm{P}} \end{bmatrix} \hat{\mathbf{x}} + \begin{bmatrix} \mathbf{I}_{n_x} \\ \mathbf{A} + \mathbf{I}_{n_x} \\ \vdots \\ \sum_{i=0}^{N_\mathrm{P}-1} \mathbf{A}^i \end{bmatrix} \boldsymbol{\beta} +$$

$$+ \begin{bmatrix} \mathbf{B} & & & \\ \mathbf{AB} & \mathbf{B} & & \\ \vdots & \vdots & \ddots & \\ \mathbf{A}^{N_\mathrm{P}-1}\mathbf{B} & \mathbf{A}^{N_\mathrm{P}-2}\mathbf{B} & \cdots & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{u}(0) \\ \mathbf{u}(1) \\ \vdots \\ \mathbf{u}(N_\mathrm{P} - 1) \end{bmatrix} +$$

$$+ \begin{bmatrix} \mathbf{F} & & & \\ \mathbf{AF} & \mathbf{F} & & \\ \vdots & \vdots & \ddots & \\ \mathbf{A}^{N_\mathrm{P}-1}\mathbf{F} & \mathbf{A}^{N_\mathrm{P}-2}\mathbf{F} & \cdots & \mathbf{F} \end{bmatrix} \begin{bmatrix} \mathbf{d}(0) \\ \mathbf{d}(1) \\ \vdots \\ \mathbf{d}(N_\mathrm{P} - 1) \end{bmatrix}.$$

By performing this substitution, the states are eliminated from the optimization problem and the size of the vector of optimization variables is reduced from $(n_u + n_h + n_Q) \cdot N_\mathrm{P} + 2 \cdot n_c$ to $n_u \cdot N_\mathrm{P} + 2 \cdot n_c$. All the $N_\mathrm{P} \cdot (n_h + n_Q)$ equality constraints have been removed from the optimization problem. This substitution needs also to be performed on the Hessian $\mathbf{H}$, which will destroy the block diagonal structure of the Hessian. Also the sparsity present in the matrix $\mathbf{A}_\mathrm{in}$ is destroyed. This is the cost we have to pay to decrease the number of optimization variables and eliminate the equality constraints.

4.6.2 Reducing the number of inequality constraints

Notice that the number of inequalities remains the same after reducing the number of optimization variables with one of the two approaches discussed in the previous subsection. The total number of inequality constraints is very large because of the safety and the flood levels for all the water levels. We will now discuss two approaches for decreasing the number of inequality constraints.

4.6.2.1 Direct constraint selection approach

For the moment the safety and flood levels are imposed on every water level for every channel. However if the channel bed slope, the safety levels and the flood levels are very smooth, imposing these constraints on all the water levels can be considered as unnecessary since the water level profile will not show big jumps from one spatial grid point to the next one. Therefore it is sufficient to impose these constraints on the water levels once every p grid points. The larger p, the smaller the number of constraints and less time is needed for solving the QP. However p cannot be taken too large because otherwise the guarantee of avoiding any flooding vanishes. An appropriate p can be found by trail and error.

This constraint selection approach can be used in a straightforward way in combination with the condensed implementation discussed in the previous part. The only difference is that in the optimization problem the matrix $\mathbf{M}_{\text{red}}^{(i)} \in \mathbb{R}^{n_{\text{con}}^{(i)} \times n_h^{(i)}}$ is needed for selecting the $n_{\text{con}}^{(i)}$ chosen constraints for reach i:

$$\mathbf{M}_{\text{red}}^{(i)} \tilde{\mathbf{H}}^{(i)} \mathbf{x}(j) \leq \mathbf{M}_{\text{red}}^{(i)} \mathbf{h}_{\text{max},1}^{(i)} + \mathbf{1}_{n_{\text{con}}^{(i)}} \cdot \eta(j) \xi_i,$$

$$\mathbf{M}_{\text{red}}^{(i)} \tilde{\mathbf{H}}^{(i)} \mathbf{x}(j) \leq \mathbf{M}_{\text{red}}^{(i)} \mathbf{h}_{\text{max},2}^{(i)} + \mathbf{1}_{n_{\text{con}}^{(i)}} \cdot \eta(j) \zeta_i.$$

4.6.2.2 Model reduction in combination with a greedy selection algorithm

Another method is to follow the approach presented in [2]. In this work a greedy selection algorithm is proposed to reduce the number of inequality constraints for a POD-based MPC controller applied to a tubular reactor. The reactor is divided into a high number of grid points and the temperature at every grid point should not exceed a given limit. It was observed that the coefficients of consecutive temperature constraints are quite similar due to the fact that the part of the POD vectors associated with the temperature profile are smooth. An algorithm is proposed for selecting a reduced set of constraints from the full set by comparing the coefficients of these constraints. The main idea of the algorithm is the following. The mean absolute error between the coefficients

of two constraints is used as a measure to quantify the distance between two constraints. The algorithm consists of two operations: dropping a constraint and reducing the feasible region after dropping a constraint. If a constraint c_2 is within a predefined distance of constraint c_1, then constraint c_2 can be dropped. The new feasible region delimited by the reduced set of constraints can now contain a small area that was not part of the feasible region of the original problem. In order to reduce the size of this extra area, a shrinking parameter γ is introduced to tighten nonconsecutive constraints leading to a reduction of the feasible region. This will reduce the extra area of the new feasible region but has as drawback that a part of the original feasible region is also removed. The author remarks that the algorithm does not guarantee that the selected set of constraints is the optimal one, in the sense that it minimizes the difference between the feasible regions delimited by the full and the reduced set of constraints. However simulation results have shown that the number of inequality constraints could significantly be reduced and the behavior of the predictive controller was practically identical to the behavior of the controller based on the complete set of constraints. More details can be found in [2].

Because the POD vectors associated with the water levels are smooth (see e.g. the derived POD vectors in Fig. 3.20 on page 69), this approach can also be used in our setting to reduce the number of inequality constraints related to the water levels. This approach can be applied for the optimization problem given in terms of the POD coefficients $\mathbf{a}$ and $\mathbf{b}$ in Section 4.6.1.1. The only differences are the inequality constraints regarding the flood levels and safety levels. They need to be replaced with

$$\mathbf{M}_{\mathrm{POD}}^{(i)}\left(\tilde{\mathbf{H}}^{(i)}\boldsymbol{\Phi}_{\tilde{n}_h}^{\mathrm{T}}\mathbf{a}(j)+\mathbf{h}_{\mathrm{ss}}^{(i)}\right)\leq\mathbf{M}_{\mathrm{POD}}^{(i)}\mathbf{h}_{\mathrm{max},1}^{(i)}+\mathbf{1}_{n_{\mathrm{con}}^{(i)}}\cdot\eta(j)\xi_i,$$

$$\mathbf{M}_{\mathrm{POD}}^{(i)}\left(\tilde{\mathbf{H}}^{(i)}\boldsymbol{\Phi}_{\tilde{n}_h}^{\mathrm{T}}\mathbf{a}(j)+\mathbf{h}_{\mathrm{ss}}^{(i)}\right)\leq\mathbf{M}_{\mathrm{POD}}^{(i)}\mathbf{h}_{\mathrm{max},2}^{(i)}+\mathbf{1}_{n_{\mathrm{con}}^{(i)}}\cdot\eta(j)\zeta_i,$$

where $\mathbf{M}_{\mathrm{POD}}^{(i)}\in\mathbb{R}^{n_{\mathrm{con}}^{(i)}\times n_h^{(i)}}$ is found by the greedy selection algorithm and it selects $n_{\mathrm{con}}^{(i)}$ constraints for reach i.

4.6.3 Application of the speeding-up techniques to a river system with a small reservoir

The following combinations of the speeding-up techniques will be tested on the full hydrodynamic model of the river system with the small reservoir described in Section 4.5 for $\Delta z = 25$ and $N_{\mathrm{P}} = 15$:

- a POD-based uncondensed implementation with all inequality constraints,

- a condensed implementation with all inequality constraints,
- a POD-based condensed implementation together with the greedy selection algorithm
- and a condensed implementation together with the direct constraint selection approach.

Both constraint selection methods are used in combination with a condensed implementation resulting in the minimal number of optimization variables, no equality constraints and a limited number of inequality constraints. The only difference between both methods besides the construction of the selection matrices $\mathbf{M}_{\mathrm{r}}^{(i)}$ and $\mathbf{M}_{\mathrm{POD}}^{(i)}$ is that the optimization problem based on the direct constraint selection is implemented with the original state space model ($\mathbf{A}$, $\mathbf{B}$ and $\mathbf{F}$), while the optimization problem based on the greedy selection algorithm is implemented with the reduced order state space model ($\mathbf{A}_{\mathrm{r}}$, $\mathbf{B}_{\mathrm{r}}$ and $\mathbf{F}_{\mathrm{r}}$). The performance of the four different implementations will be compared with the original uncondensed implementation. `Quadprog` is used to solve every QP [123]. The interior-point algorithm of `quadprog` is used in the case of the two uncondensed implementations, while the active-set algorithm is used for the three condensed implementations. The first algorithm is a large-scale algorithm, which exploits the sparsity structure present in the matrices $\mathbf{H}$, $\mathbf{A}_{\mathrm{in}}$ and $\mathbf{A}_{\mathrm{eq}}$ for the uncondensed implementations. The second algorithm is more suitable for medium-scale problems with a dense structure and is therefore used for the condensed implementations.

The POD basis vectors are found from 3000 samples. The energy criterion is used to determine $\tilde{n}_h$ and $\hat{n}_Q$. The plot of $1 - \bar{P}_{\tilde{n}_h}$ and $1 - \bar{P}_{\hat{n}_Q}$ for the first 120 basis vectors are shown in Fig. 4.25. Based on the truncation degree $1 - \bar{P}_{\tilde{n}_h} = 6.22 \cdot 10^{-5}$ and $1 - \bar{P}_{\hat{n}_Q} = 1.59 \cdot 10^{-7}$, the first $\tilde{n}_h = 11$ and $\hat{n}_Q = 27$ POD basis vectors respectively are chosen. The shrinking parameter γ is set to 0.01, while the maximum distance before dropping a constraint is set to 0.02 for the greedy selection algorithm. The direct selection approach retains the constraints on the water levels for every 250 meters ($p = 10$).

Table 4.22 shows the average, minimum and maximum time needed to solve the QP together with the number of optimization variables and the number of (in)equality constraints for the uncondensed implementation of the full order model, the uncondensed implementation of the reduced order model, the condensed implementation, the condensed implementation in combination with the POD based greedy selection algorithm and the condensed implementation combined with the direct constraint selection approach. There is a big gain by using one of both techniques for reducing the number of optimization variables. The biggest gain is achieved by using the condensed implementation. The advantage of eliminating the water levels and discharges as optimization

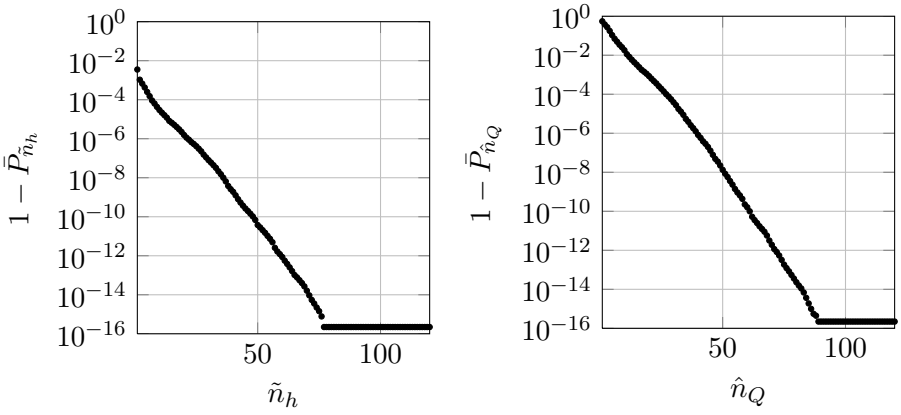


Figure 4.25: Logarithmic plot of $1 - \bar{P}_{\tilde{n}_h}$ (left) and $1 - \bar{P}_{\hat{n}_Q}$ (right) for determining the truncation degree of the POD basis vectors.

variables and removing all equality constraints dominates over the disadvantage of the loss in structure in the Hessian $\mathbf{H}$ and the matrix $\mathbf{A}_{\text{in}}$. An extra reduction is achieved in the average computation time by decreasing the number of inequality constraints. The resulting optimization problems have only 74 optimization variables, no equality constraints and 1304 or 1004 inequality constraints (compared to 9749 optimization variables, 9675 equality constraints and 9734 inequality constraints for the original problem). The average computation time is reduced from 12.7 seconds with a factor more than 60 to 0.18 seconds and 0.2 seconds. This reduction in computation time does not affect the control performance of the predictive controller. Fig. 4.26 shows the evolution of the maximal difference between the water levels and their set-points for the upstream part of the river, the downstream part of the river and the reservoir for LN-MPC implemented with the uncondensed approach taking all constraints into account, the condensed approach in combination with the POD based greedy selection algorithm and the condensed approach combined with the direct constraint selection procedure. Fig. 4.27 shows the applied control actions for the three different implementations. There is almost no difference in the resulting evolution of the water levels and the control actions between the three implementations. The reduction in computation time is achieved without degrading the performance of the controller.

Remark. There is one disadvantage with the greedy selection algorithm. This algorithm is based on the POD basis vectors of the reduced order model. It can occur that the selected POD basis vectors are not accurate enough and the performance of the controller is degraded. The solution is to construct new snapshot matrices, select new POD basis vectors and repeat the algorithm.

Table 4.22: Comparison between the average, minimum and maximum time needed to solve the QP for the uncondensed implementation of the full order model, the uncondensed implementation of the reduced order model, the condensed implementation with all the inequality constraints, the condensed implementation in combination with the POD based greedy selection algorithm and the condensed implementation combined with the direct constraint selection approach together with the number of optimization variables and the number of equality and inequality constraints for $\Delta z = 25$ and $N_\mathrm{P} = 15$. The simulation was performed on a PC with a 2.8 GHz Intel Core i7 CPU and 8 GB of RAM. The interior-point algorithm of `quadprog` is used for the first two uncondensed implementations, while the active-set algorithm is used for the three condensed implementations.

	mean [s]	min [s]	max [s]	no. optimization variables	no. equality constraints	no. inequality constraints
uncondensed	12.698	9.033	21.989	9749	9675	9734
POD-based uncondensed	2.704	1.722	4.893	599	525	9734
condensed	1.156	0.271	10.085	74	–	9734
condensed + POD-based greedy selection	**0.177**	0.051	0.672	**74**	–	**1004**
condensed + direct selection	**0.200**	0.060	0.821	**74**	–	**1304**

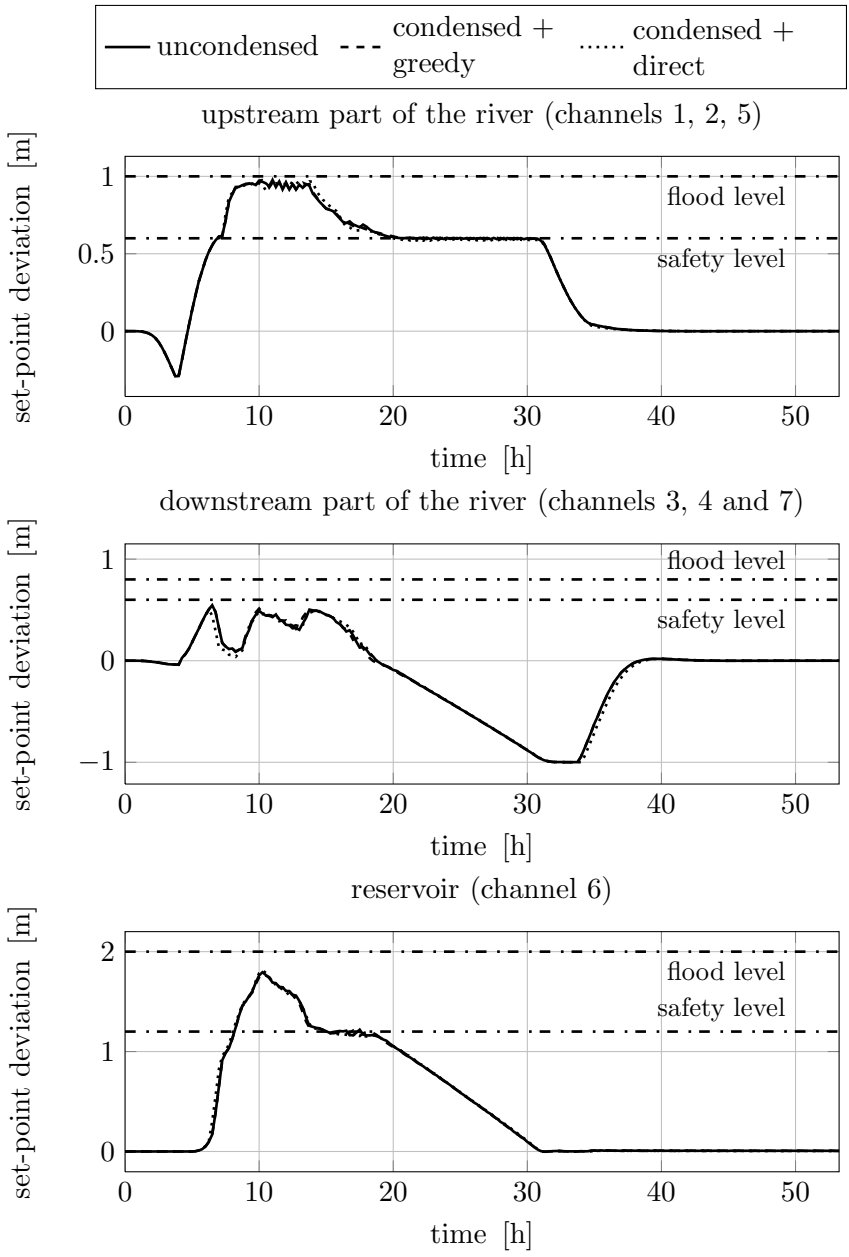


Figure 4.26: Evolution of the maximal difference between water levels and their set-points obtained with three different implementations. The top plot shows the maximal deviation of the water levels of channels 1, 2 and 5 at every time step, the middle plot the maximal deviation of the water levels of channels 3, 4 and 7 and the bottom plot the maximal deviation of the water levels of channel 6. The flood levels correspond to the highest horizontal lines, the safety levels with the lowest ones.

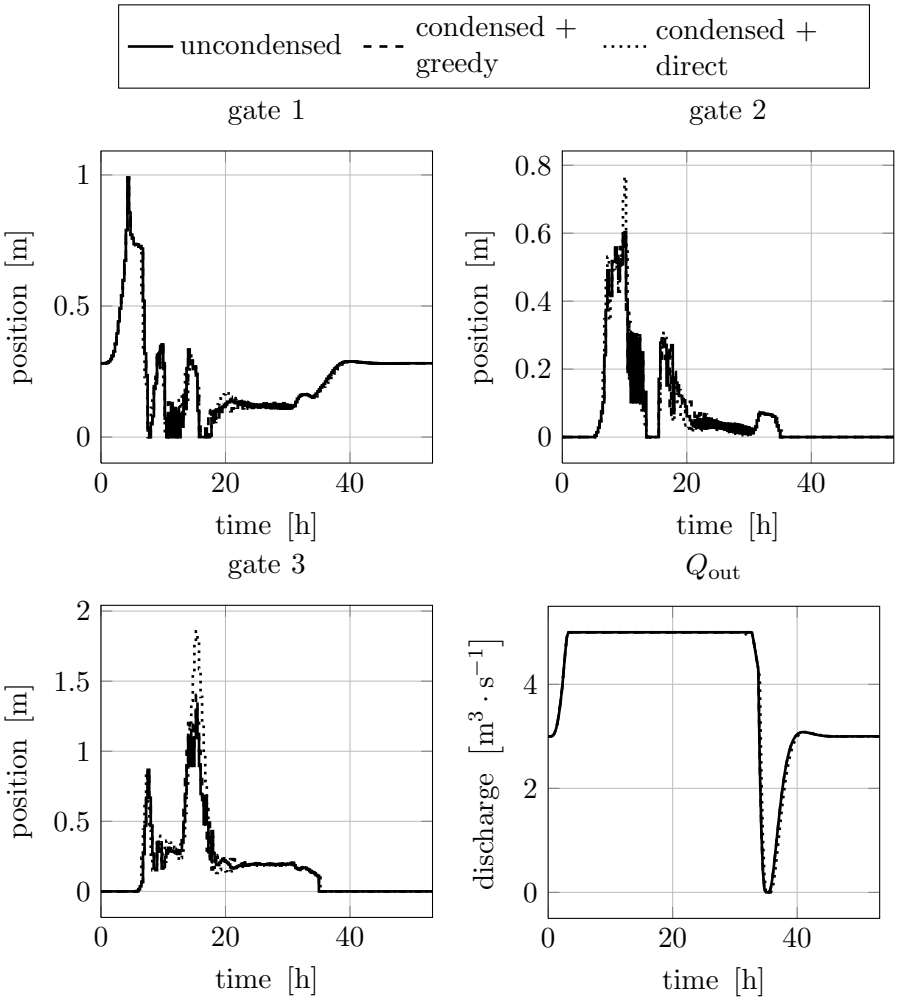


Figure 4.27: Control actions for LN-MPC obtained with the uncondensed implementation, the condensed implementation with the POD-based greedy selection algorithm and the condensed implementation with the direct selection approach. The top plots and the bottom left plot show the evolution of the three gates and the bottom right plot shows the discharge Q_{out} at the end of the river system.

Remark. A similar test has been performed with the QP-solver `cplexqp` of IBM [76]. For this solver the results are a little bit different. If one of the two strategies for reducing the number of optimization variables is used without a reduction in the number of inequality constraints, the computation time will increase instead of decrease. The loss in sparsity has a higher impact than the advantage of reducing the number of optimization variables. However the condensed implementation with any of the two methods for reducing the number of inequality constraints leads to a reduction in the computation time (with up to a factor of 16). And this reduction is larger then when an uncondensed implementation is used with any of the two approaches for reducing the number of inequality constraints.

4.7 Conclusions

In this chapter we have shown that MPC is suitable for flood control and set-point control of river systems at the same time. MPC has some specific advantages for controlling river systems compared to more classical control schemes:

- it can take flood levels into account as inequality constraints,
- limitations on hydraulic structures can be incorporated,
- set-point control can be achieved by minimizing the deviation between the water levels and their set-points,
- working with a process model results in finding optimal control actions for the entire river system
- and working with a prediction window in combination with a process model allows us to incorporate rain predictions.

MPC is first applied to the most simple example: the control of a single reach. The dynamics of the reach are approximated with a linear state space model such that the resulting optimization problem is a Quadratic Programming problem (QP). The optimization problem makes use of slack variables to impose the flood levels as soft constraints. This has the advantage that the QP remains feasible at all times. These flood levels are imposed as exact soft constraints by taking their corresponding weights in the objective function large enough. Imposing the flood levels as inequality constraints and minimizing over the deviation between the water levels and their set-points ensure that MPC can be used for flood control and set-point control of the reach. Simulation results have shown that MPC outperforms LQR, especially for the task of flood control.

MPC reacts preventively on future rainfall because of its prediction window in combination with the flood levels imposed as inequality constraints. These preventive control actions lower the water levels of the reach resulting in extra buffer capacity. This temporal extra buffer capacity prevents the reach from flooding during the period of heavy rainfall. LQR reacts too late to prevent the reach from flooding: the maximal violation of the flood levels is 0.5 m.

We have shown that the optimization problem formulated for set-point control and flood control of a single reach can easily be extended to the case of two reaches connected with each other through a hydraulic structure. However one has to be careful how to deal with the nonlinear gate equations. Using a linear approximate model for the Saint-Venant equations as well as for the nonlinear gate equations results in a poor control performance. The solution to overcome this problem is to work with the LN-model introduced in the previous chapter as approximate model and keeping the gate equations out of the optimization problem: the gate discharges are used as optimization variables instead of the gate positions. The linear model of a river system based on these gate discharges is accurate enough to be used such that the resulting optimization problem is still a QP. The constraints on the gate positions can be easily translated to (time-varying) constraints on the gate discharges by performing a prediction step based on the linear model before solving the QP. During this prediction step care has to be taken that the gate position always has an influence on the gate discharge. Otherwise the lower and upper limits of the gate discharge can become equal and the gate becomes uncontrollable: the controller loses one control freedom. This uncontrollability is avoided by keeping the gates always in their controllable region or at least on the boundary. The simulation results show that the MPC scheme that optimizes over the gate discharges and then performs a conversion step to find the corresponding gate positions (LN-MPC) outperforms the MPC scheme that directly optimizes over the gate positions (L-MPC). LN-MPC succeeds in tracking a time-varying reference trajectory for two reaches which is not the case with L-MPC. The simulation cost related to the deviation of the water levels with respect to their set-points over the entire simulation is almost 15 times smaller with LN-MPC. For the case of flood control LN-MPC succeeds in keeping all the water levels below their flood levels at all times, while L-MPC results in floods up to 2 m.

MPC can also be used when a water reservoir is present in the river system. The specific restrictions that the reservoir is only allowed to be used when there is a flooding risk and that its buffer capacity should be recovered as fast as possible, can easily be incorporated inside the optimization problem. Using a large penalty factor on the deviation of the water levels of the reservoir from their set-point ensures that the controller will not use the reservoir for a better set-point control of the water levels of the other reaches during no or little rainfall. Penalizing the violation of safety limits and flood limits has as effect

that the controller will automatically use the buffer capacity of the reservoir to minimize these violations. The buffer capacity is recovered by decreasing the set-point for the water levels downstream of the reservoir such that the excess of water in the reservoir can be released in a fast way. The simulation results show that the proposed control scheme can be used for river systems with small and large reservoirs. When there is no risk of flooding the water levels are kept close to their set-points, when there is a risk of flooding the controller uses the buffer capacity of the reservoir resulting in a maximal flooding of only 8 mm, and afterwards it recovers the used buffer capacity almost completely (only 1 cm of excess of water is remaining in the reservoir) and steers the water levels back to their set-points with deviations smaller than $2.5 \cdot 10^{-2}$ mm.

For all the different test cases the influence of a Kalman filter as state estimator on the control performance of MPC is found to be small. Based on measuring only one or two water levels per reach, the Kalman filter gives an estimate of the current state of the river system which is accurately enough to be used by the controller. Comparison of the results of MPC with full state information and MPC in combination with a Kalman filter reveals only small differences in the control performance. Similar set-point deviations in the range of only 10^{-2} mm are achieved at the end of the simulations. The simulation cost related to the set-point deviations based on the entire simulation results is never increased with more than 1 %, for some tests it is even reduced with more than 15 %. For the test cases of controlling a single reach and a river system consisting of two reaches, both MPC with full state information and MPC in combination with a Kalman filter prevent any flooding resulting in a flood cost of 0. For the first test case of MPC applied to a river system with a reservoir, MPC could not prevent three reaches from flooding and the use the Kalman filter increases the maximal flooding with (only) 0.9 mm to 9 mm. However these floods last for less than 2 minutes. The set-point deviation cost and the flood cost are almost identical compared to the test with full state information. The biggest effect of the use of the Kalman filter on the flood control performance is seen in the last test case. None of the reaches are flooding for the case of full state information, while the use of the Kalman filter results in the flooding of 3 reaches. However the total impact of these floods is limited because they last less than 10 minutes and the maximal flooding is only 3.7 cm. The use of the Kalman filter has no influence on the set-point deviation cost compared to the results obtained with full state information.

Solving the QP at every time step takes the largest computation time when we are working with the expanded formulation of the optimization problem. This computation time can be reduced by decreasing the number of optimization variables. One approach is to use POD in combination with Galerkin projection to reduce the model order of the linear state space model. However a better approach is to work with a condensed formulation of the optimization

problem by writing the water levels and discharges as a function of the current state estimate and the (future) inputs and disturbances. Performing this substitution eliminates all the equality constraints and the (future) states from the optimization problem. An extra reduction can be achieved by reducing the number of inequality constraints. This reduction can be done by performing a greedy constraint selection based on a reduced order model with POD. By exploiting the similarity between the different coefficients related to the constraints on the water levels, a large number of inequality constraints can be removed. Another approach is to impose the safety levels and flood levels at only a limited number of grid points. Both techniques in combination with a condensed implementation reduced the average computation time needed to solve the QP at every iteration with a factor more than 60 for the given case study when `quadprog` of Matlab is used and with a factor more than 16 when `cplexqp` of IBM is used, without losing any performance regarding flood control and set-point control [76, 123].

5

Model Predictive Control Applied to the Demer

5.1 Introduction

The river systems used to evaluate the performance of the controllers in the previous chapter can be considered as artificial academic examples. In this chapter the proposed control scheme in Section 4.5 is applied to a mathematical model of the Demer built from field data.

The current controller used for preventing the Demer from flooding is an advanced three-position controller. Although the rules of this controller are based on expert knowledge, the controller could not prevent the Demer from flooding in 1998 and 2002. This is because the control actions are only based on local information and therefore the control actions are not necessarily optimal for the entire river system. Furthermore rainfall predictions are not taken into account. Model Predictive Control does not suffer from these disadvantages. By working with a process model and solving one large optimization problem for all the hydraulic structures at the same time, the controller finds optimal control actions for the entire river system. By imposing the flood levels as upper limits on the water levels and combining the process model together with future rainfall inside the optimization problem, the controller will react preventively on future rainfall to avoid flooding or it will try to reduce the amount of flooding.

The formulation of the optimization problem is very similar as the one used

by the LN-MPC controller in the previous chapter. We optimize over the gate discharges instead of the gate openings and a linear state space model is used inside the optimization problem such that the resulting optimization problem is a Quadratic Programming problem (QP). The use of slack variables for the upper limits on the water levels ensures the feasibility of the QP at all times. The weights related to the objective function are chosen in a similar fashion as before and the used buffer capacity is recovered in the same manner. However, the irregularities of the river bedding profile and the cross section profiles of the Demer have their effect on the method for finding the limits on the gate discharges. While previously a linear model was accurate enough, this is not the case for the Demer during periods of heavy rainfall. A nonlinear model is needed for the prediction step to get accurate enough estimates of the limits on the gate discharges. However the computation time needed for performing this prediction step can be kept limited. These predictions based on the nonlinear model are also used to perform a simple model update of the linear state space model to improve its accuracy over a wide range of operation conditions. The irregularities also affect the Kalman filter used for estimating the unknown states. The linear open loop prediction is replaced with a prediction based on a nonlinear model.

In our research group, MPC has already been used for flood control of the Demer [18–20, 37, 38, 189]. There are three major distinctions with these works:

- **Process model.** In this work, the Demer is represented with a full hydrodynamic model based on field data of the river, while previously a much simpler conceptual model with a limited number of nodes was used for modelling the Demer [18–20, 37, 38, 189].
- **Control strategy.** Because the gate positions are used as optimization variables, a nonlinear predictive controller is used in those works [18–20, 37, 38, 189]. At every iteration, the river dynamics are approximated by a sequence of time-varying linear state space models. A trust region is required inside the optimization problem in combination with a line search method as post-processing step for finding good gate positions. The upper limits on the water levels are imposed as hard constraints. Therefore a constraint relaxation strategy is needed to make the optimization problem feasible [18–20, 189]. During periods of heavy rainfall this can result in the requirement of solving multiple optimization problems at every time step. In this work, there is no need for working with a nonlinear predictive controller because we optimize over the gate discharges: no linearization step is required. By implementing the upper limits on the water levels as soft constraints, the optimization problem does not have infeasibility issues.

- **State estimator.** The unknown water levels and discharges are found by solving at every time step an optimization problem based on the recent measurements in a window looking back in time [18–20,37,38,189]. This state estimation technique is called Moving Horizon Estimation (MHE). In this work, there was no need of using MHE since the Kalman filter proved to be accurate enough.

This chapter has the following structure. Section 5.2 introduces the study area of the Demer that will be used to test the controllers. In addition, a mathematical model is derived for the upstream part of the Demer. The working principles of the three-position controller is explained in Section 5.3. Section 5.4 discusses how Model Predictive Control can be used for controlling the Demer. This section presents how the irregularity of the field data of the Demer affects the approximate model, the optimization problem and the state estimator. Section 5.5 compares the performance of the proposed MPC control strategy with the performance of the three-position controller for the historical rainfall data of the Demer of the flood event of 2002. Section 5.6 summarizes the main conclusions of this chapter.

This chapter is based on the following publication:

[35] Breckpot, M., Agudelo, O. M., Meert, P., Willems, P., and De Moor, B. Flood control of the Demer by using Model Predictive Control. Internal Report 13-24, ESAT-SISTA, KU Leuven (Leuven, Belgium). *Submitted for publication in Control Engineering Practice* (2013).

5.2 The Demer

5.2.1 Study area of the Demer and control objectives

Fig. 5.1 gives a schematic overview of a large part of the Demer together with its tributaries, water reservoirs and hydraulic structures. Because this is the first time the LN-MPC controller is tested on real river data, this dissertation focuses only on the upstream part of the Demer in the rectangle area of Fig. 5.1 with the assumption that the reservoir Schulensmeer is directly connected to the Demer through gate D (similarly as the models used in [18,20,37,38]). Fig. 5.2 shows the part of the Demer that will be modelled and controlled in this work. This upstream part contains the Demer and its tributary Mangelbeek, the reservoir Schulensmeer and the gated weirs A, K7 and D. Gates A and D can be used to control the water flowing between the Demer and the reservoir. Gate

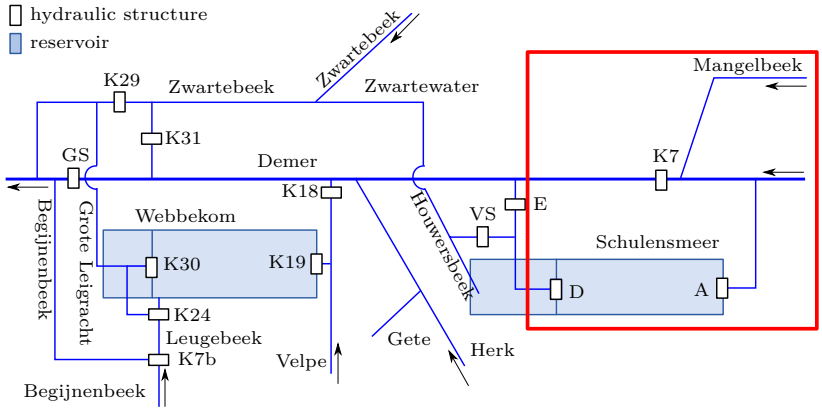


Figure 5.1: Schematic overview of the Demer together with its tributaries, the hydraulic structures and the reservoirs Webbekom and Schulensmeer. The area of interest corresponds to the upstream part of the Demer inside the rectangle.

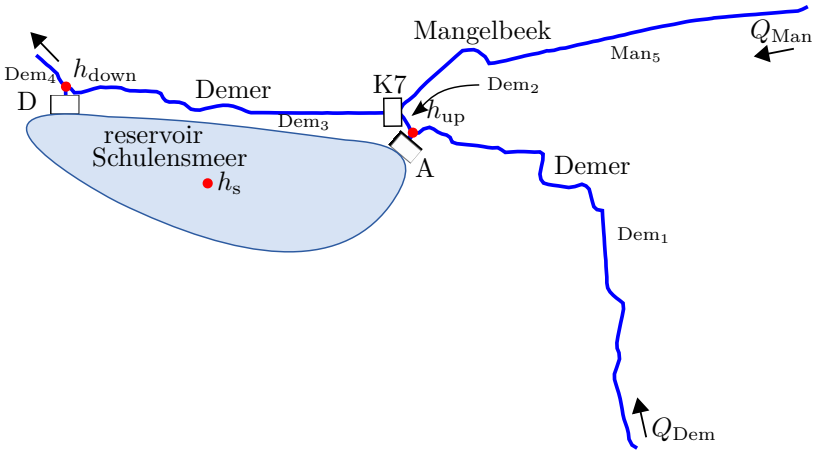


Figure 5.2: Upstream part of the Demer that is modelled and controlled in this study. Gated weirs A and D can be used to control the flow between the Demer and the reservoir Schulensmeer, while gated weir K7 is used to control the water levels in the Demer and its tributary Mangelbeek.

K7 can be used to control the water levels in the Demer and the Mangelbeek. Historical data will be used to model the incoming discharges upstream of the Demer and the Mangelbeek, respectively Q_{Dem} and Q_{Man}. The Demer itself consists of four reaches (Dem_1 until Dem_4) while the Mangelbeek contains one reach (Man_5).

During normal operation the controller should focus on **set-point control** by keeping h_{up} as close as possible to 21.5 m TAW (where TAW is the reference height for altimetry in Belgium) without increasing the water level h_{s} of the reservoir. During a period of heavy rainfall, the buffer capacity of the reservoir cannot be used as long as any of the water levels will stay below their given **safety limits**. Once the model predicts a future violation of these safety limits, the buffer capacity of the reservoir can be used until its own safety limit is reached. In this situation the water levels of the reaches are allowed to further increase until the moment that they violate their **flood levels**. At that moment the last remaining buffer capacity can be used. After a period of heavy rainfall the water level of the reservoir should be decreased as fast as possible to its original height of 20.4 m TAW in order to have sufficient buffer capacity for possible future rainfall. After **recovering** the **buffer capacity** the controller should focus again on set-point control. Every hydraulic structure has an upper and lower limit on the gate position together with a rate of change constraint.

5.2.2 Mathematical model

5.2.2.1 The Saint-Venant equations

The full hydrodynamic equations of de Saint-Venant introduced in Section 3.2.1 will be used to model the dynamics of reaches Dem_1, Dem_2, Dem_3, Dem_4 and Man_5:

$$\frac{\partial A^{(i)}(z,t)}{\partial h}\frac{\partial h^{(i)}(z,t)}{\partial t}+\frac{\partial Q^{(i)}(z,t)}{\partial z}=0,$$

$$\frac{\partial Q^{(i)}(z,t)}{\partial t}+\frac{\partial}{\partial z}\frac{Q^{(i)}(z,t)^2}{A^{(i)}(z,t)}+gA^{(i)}(z,t)\left(\frac{\partial h^{(i)}(z,t)}{\partial z}+S_{\text{f}}^{(i)}(z,t)-S_0^{(i)}(z)\right)=0,$$

with $i=1,\ldots,5$ and where the friction slope $S_{\text{f}}^{(i)}(z,t)$ is modelled with the resistance law of Manning [45]:

$$S_{\text{f}}^{(i)}(z,t)=\frac{n_{\text{eqmann}}^{(i)}(z)^2Q^{(i)}(z,t)|Q^{(i)}(z,t)|}{A^{(i)}(z,t)^2R^{(i)}(z,t)^{1/3}}. \tag{5.1}$$

The reaches of the Demer have an irregular bed slope and an irregular cross section as shown in Fig. 5.3 for the Mangelbeek. As will be explained in

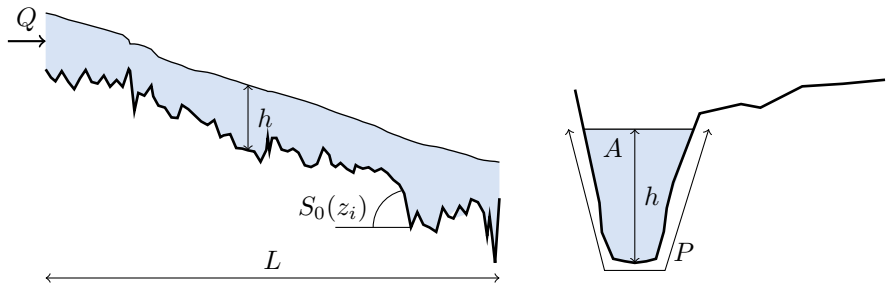


Figure 5.3: The bed level of the Mangelbeek (left) and an example of a cross sectional profile (right).

Section 5.4.1, these cross sections will be approximated with a trapezoidal shape with bottom width B and side slopes S_1 and S_2 for deriving the linear state space models used by MPC inside the optimization problem.

Every irregular cross section consists of a piecewise-linear profile where one specific Manning coefficient is defined for every linear segment. The Manning coefficients for the segments in the lower part of the river bed are typically smaller than the coefficients for the segments in the higher part or close to the flood levels because of their difference in roughness. One needs to calculate an equivalent Manning coefficient based on the current water level for every section every time Eq. (5.1) is evaluated. Section 3.2.2.3 gives different methods for calculating the equivalent Manning roughness coefficient n_{eqmann}. By assuming that the mean flow velocity over every subdivision i is equal to the mean flow velocity over the entire section [73], the equivalent Manning coefficient can be calculated with

$$n_{\text{eqmann}}(z) = \left(\frac{\sum_{i=1}^{m} P_i n_{\text{mann},i}(z)^{3/2}}{\sum_{i=1}^{m} P_i} \right)^{2/3}$$

where $n_{\text{mann},i}(z)$ is the Manning coefficient at position z for the ith linear segment with a wetted perimeter P_i and m the number of segments (partially) under water.

5.2.2.2 Gated weirs

The hydraulic structures A, D and $K7$ are gated weirs, which were introduced in Section 3.3.3.1. The discharge over the gate depends on the up- and downstream water levels of the gate and the gate position c. The hydraulic

structure equations used in this work are based on the equations used in an existing full hydrodynamic model implemented in the InfoWorks-RS software [78] built by order of the Flemish Environment Agency [189]. The gate discharges are given with the following equation:

$$Q_{\text{gate}}(t) = \tilde{f}(c(t), h_{\text{up}}(t), h_{\text{down}}(t)),$$

where $\tilde{f}$ is a nonlinear scalar function, $h_{\text{up}}(t)$ and $h_{\text{down}}(t)$ represent the upstream and downstream water level of the gate respectively. The specific form of this equation depends on the flow condition of the gate. If the gate position c is too small in comparison with h_{up} and h_{down}, the gate position itself has no influence on the discharge over the gate, and the gate is said to be in throat control mode. If the gate position has an influence on the discharge over the gate, then the gate is said to be in gate control mode. The specific form of this equation can be found in Appendix A.1. This gate equation can be used to model the connection between Dem_2 and Dem_3 through gate K7:

$$Q_{\text{gate}}^{(K7)}(t) = \tilde{f}\Big(c^{(\text{K7})}(t), h^{(2)}(L^{(2)}t), h^{(3)}(0,t)\Big),$$

$$Q^{(2)}(L^{(2)}, t) = Q_{\text{gate}}^{(K7)}(t),$$

$$Q^{(3)}(0,t) = Q_{\text{gate}}^{(K7)}(t).$$

This simply means that the discharge leaving Dem_2 is equal to the discharge entering Dem_3 and they are equal to the gate discharge. Similar equations holds for gates A and D.

5.2.2.3 (Artificial) spill

The Mangelbeek Man_5 flows into the Demer at the end of Dem_2. At this connection point the bedding of the Mangelbeek is more than 1 m higher than the bedding of the Demer. In order to take this significant height difference into account, the connection of the Mangelbeek with the Demer is modelled with a spill. The shape of the top of the spill is taken equal to the cross sectional profile of the downstream end of the Mangelbeek. This means that the spill is not really blocking any amount of water and is actually an artificial structure used for modelling purposes only.

The flow over the spill depends in a nonlinear way on the downstream water level of the Mangelbeek and the downstream water level of Dem_2:

$$Q_{\text{spill}}(t) = \tilde{g}(h^{(5)}(L^{(5)}, t), h^{(2)}(L^{(2)}, t)).$$

More details about the specific form of the nonlinear scalar function $\tilde{g}$ can be found in Appendix A.2.

5.2.2.4 Junctions

Junctions are places where three or more reaches coincide or where a reach or reservoir is connected via a hydraulic structure to two other reaches. Examples for the Demer are gate A connecting the reservoir with Dem_1 and Dem_2 and gate D connecting the reservoir with Dem_3 and Dem_4. The water levels of the reaches at these junctions should be equal and the sum of the discharges entering the junction should be equal to the sum of the discharges leaving the junction. Applying this to the first junction results in the following equations:

$$h^{(1)}(L^{(1)}, t) = h^{(2)}(0, t),$$

$$Q^{(1)}(L^{(1)}, t) = Q^{(2)}(0, t) + Q^{(\text{A})}_{\text{gate}}(t).$$

Similar equations can be derived for the other junction.

5.2.2.5 Reservoir Schulensmeer

The reservoir Schulensmeer is modelled as one large tank. The change in volume is related to the water entering and leaving through gates A and D:

$$dV_{\text{s}}/dt = Q^{(\text{A})}_{\text{gate}}(t) - Q^{(\text{D})}_{\text{gate}}(t)$$

with V_{s} the volume of the reservoir $\left[\text{m}^3\right]$. This volume depends in a nonlinear way on the water level h_{s} of the reservoir. The nonlinear relation used in this work is based on the conceptual model developed for this reservoir, based on detailed topographical data, by Willems et al. [189] and Chiang and Willems [44]. It has the following form:

$$V_{\text{s}} = \begin{cases} (h_{\text{s}} - 20.38)/0.000771 & h_{\text{s}} < 20.38, \\ ((h_{\text{s}} - 20.38)/0.000771)^{1/0.838549} & h_{\text{s}} \geq 20.38. \end{cases}$$

More information on how these equations are derived, can be found in [44].

5.2.2.6 Downstream boundary condition

The downstream boundary condition for Dem_4 will be modelled with the following rating curve

$$h^{(4)}(L^{(4)}, t) = 0.9722 \cdot Q(L^{(4)}, t)^2.$$

This rating curve is derived based on the conceptual model of the Demer [189].

5.3 Three-position controller

A standard three-position controller is often used in practice for set-point control and its control actions are based on some very simple rules [142]:

- if the water level is between an upper and lower limit of the set-point, the gate remains unchanged,
- if the water level exceeds the upper limit, then the gate is lowered with a fixed step to lower the water level,
- and if the water level is below the lower limit, then the gate position is increased to increase the water level.

These standard rules are used by the advanced three-position controller installed by the Flemish Environment Agency for set-point control during normal operation. During periods of heavy rainfall the working mode of the controllers shifts towards flood control. These standard rules are replaced by new if-then-else rules and were formulated by the local water administration. Based on their many years of experience in controlling the Demer these rules can be considered as expert knowledge. An advantage of this controller is that the gate movement is limited, which restricts the wear of the gates. Another advantage is that this type of controller is easily implementable in practice and requires only a very limited number of measurements at every time step. However, the performance of this controller is limited because rain predictions are not taken into account. Furthermore the control actions are only based on local information which has as a consequence that the control actions are not globally optimal. Therefore the operators have to intervene often and set the set-point for the hydraulic structures manually. This however introduces subjectivity in the control loop and it means that the control actions depend on the experience of the operator.

5.4 Design of Model Predictive Controller

MPC does not suffer from the drawbacks of the three-position controller. This section explains how a Model Predictive Controller can be designed for flood control and set-point control of the Demer. After introducing the approximate model, attention is paid to all the concepts of the optimization problem solved by the controller at every time step. Also the design of the Kalman filter as state estimator is discussed.

5.4.1 Approximate model

The model of the river system derived in Section 5.2.2 is too complex to be directly used inside the optimization problem. Similar as the approach followed in Sections 4.4 and 4.5, a linear state space model will be used to approximate the river dynamics with the gate discharges $Q_{\text{gate}}^{(\text{A})}$, $Q_{\text{gate}}^{(\text{K7})}$ and $Q_{\text{gate}}^{(\text{D})}$ as input variables instead of the gate positions $c^{(\text{A})}$, $c^{(\text{K7})}$ and $c^{(\text{D})}$:

$$\mathbf{x}(k+1) - \mathbf{x}_{\text{ss}} = \mathbf{A}(\mathbf{x}(k) - \mathbf{x}_{\text{ss}}) + \mathbf{B}(\mathbf{u}(k) - \mathbf{u}_{\text{ss}}) + \mathbf{F}(\mathbf{d}(k) - \mathbf{d}_{\text{ss}})$$

with $\mathbf{x}(k) = \left[\mathbf{x}^{(1)}(t_k); \ldots; \mathbf{x}^{(5)}(t_k); h_{\text{s}}(t_k)\right]$, $\mathbf{d}(k) = [Q_{\text{Dem}}(t_k); Q_{\text{Man}}(t_k)]$, $\mathbf{u}(k) = \left[Q_{\text{gate}}^{(\text{A})}(t_k); Q_{\text{gate}}^{(\text{K7})}(t_k); Q_{\text{gate}}^{(\text{D})}(t_k)\right]$, $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$ and $\mathbf{F} \in \mathbb{R}^{n_x \times n_d}$ with n_x the number of states, n_u the number of inputs and n_d the number of disturbances. The state space matrices are derived for the linearization points $\mathbf{x}_{\text{ss}}$, $\mathbf{u}_{\text{ss}}$ and $\mathbf{d}_{\text{ss}}$ which correspond to the desired steady state values of the water levels and discharges, the steady state gate discharges and the nominal incoming discharges. The state space model can be rewritten as

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{F}\mathbf{d}(k) + \boldsymbol{\beta}$$

with

$$\boldsymbol{\beta} = \mathbf{x}_{\text{ss}} - \mathbf{A}\mathbf{x}_{\text{ss}} - \mathbf{B}\mathbf{u}_{\text{ss}} - \mathbf{F}\mathbf{d}_{\text{ss}}.$$

The irregular cross sectional data will not be used in the linearization procedure. The resulting linear model would only be a good approximation of the nonlinear dynamics locally around the linearization point. To have a linear model that provides a good approximation for a wide range of water levels and discharges, every irregular profile is first approximated with a trapezoidal profile. The parameters B, S_1 and S_2 of every trapezoidal cross section can found in multiple ways.

One approach is based on an example given in [31]. Boyd and Vandenberghe show that the least-squares fit of a convex function to given data (x_i, y_i), $i = 1, \ldots, m$ formulated as the following infinite-dimensional problem

$$\min_f \sum_{i=1}^{m} (y_i - f(x_i))^2$$

$$\text{s.t. } f : \mathbb{R} \to \mathbb{R} \text{ is convex, } \text{dom} f = \mathbb{R}$$

is a convex piecewise-linear function, which can be formed from the solution of the following QP:

$$\min_{\hat{y}_i, g_i} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

$$\text{s.t. } \hat{y}_j \geq \hat{y}_i + g_i(x_j - x_i), \quad i,\, j = 1, \ldots, m.$$

The piecewise-linear function f is given by

$$f(x) = \max_{i=1,\ldots,m} (y_i + g_i(x - x_i)).$$

The variable g_i corresponds to the slope of the function f at point x_i. For more information we refer the reader to [31]. This optimization problem can be adapted in order to favor a convex function similar to a trapezoidal shape as optimal solution in the following way:

$$\min_{\hat{y}_i, g_i} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 + \lambda \sum_{i=2}^{m} (g_i - g_{i-1})^2$$

$$\text{s.t. } \hat{y}_j \geq \hat{y}_i + g_i(x_j - x_i), \qquad i,\, j = 1, \ldots, m,$$

$$g_k = 0, \tag{5.2}$$

$$g_i \leq 0, \qquad i = 1, \ldots, k-1, \tag{5.3}$$

$$g_i \geq 0, \qquad i = k+1, \ldots, m. \tag{5.4}$$

with $\lambda > 0$ a regularization parameter and k the index corresponding with the bed point of the cross sectional profile. A high value of λ will make the difference between consecutive values of g small, which leads to a piecewise-linear function f with a small number of line segments with different slopes. Equations (5.2)-(5.4) ensure that the function f consists of a monotone decreasing convex piecewise-linear function, followed by a horizontal segment and ending with a monotone increasing convex piecewise-linear function. This reflects the general trend of a trapezoidal shape. An extra post-processing step is required to reduce the monotone decreasing convex piecewise-linear function to a decreasing linear function followed by a horizontal segment. A similar reasoning applies to the monotone increasing convex part of f. Furthermore these horizontal segments should have the same height. The implementation of this post-processing step is far from straightforward.

A much more simpler approach is to determine the values of the parameters B, S_1 and S_2 of every trapezoidal cross section by solving the following constrained least-squares problem

$$\min_{B,S} \sum_{i=1}^{m} \left(A_i - \left(h_i B + h_i^2 S\right)\right)^2$$

$$\text{s.t. } B \geq 0,\, S \geq 0,$$

with A_i the cross sectional area corresponding with the water level h_i, for $i = 1, \ldots, m$ and $S_1 = S_2 = S$. If one is interested in finding different values for

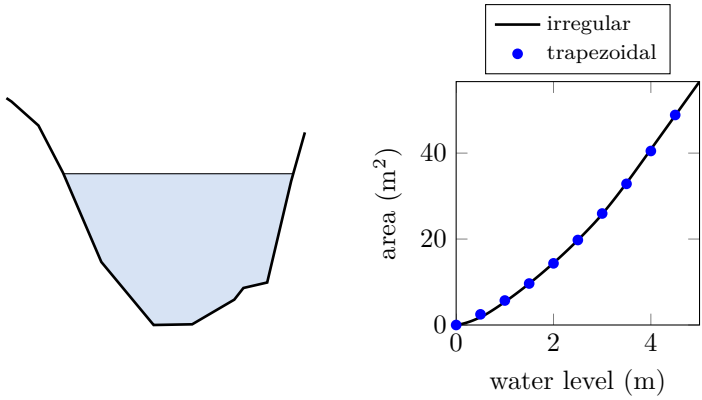


Figure 5.4: An irregular cross sectional profile (left) together with its cross sectional area as a function of the water level and its optimal approximation in the least-squares sense via a trapezoidal shape (right).

the side slopes, a possible approach is to include the wetted perimeter inside the optimization problem:

$$\min_{B,S_1,S_2} \sum_{i=1}^{m} \big(A_i - \big(h_i B + h_i^2 (S_1 + S_2)/2\big)\big)^2 + \\ + \lambda \sum_{i=1}^{m} \left(P_i - \left(B + h_i^2 \left(\sqrt{1+S_1^2} + \sqrt{1+S_2^2}\right)\right)\right)^2$$

$$\text{s.t. } B \geq 0,\, S_1 \geq 0, S_2 \geq 0,$$

with P_i the wetted perimeter corresponding to the water level h_i, for $i = 1, \ldots, m$ and $\lambda > 0$ a regularization parameter. However this turns the least-squares problem into an NLP. It has been noticed that solving this more complex optimization problem has only a minor influence on the resulting value of the parameters and it suffers from local minima. On the right of Fig. 5.4, the fitting result for an irregular cross section of Dem_3, which is visualized on the left of Fig. 5.4, can be observed. It is clear that the area of the irregular profile can be approximated accurately with a trapezoidal shape.

As mentioned in Section 5.2.2, every irregular cross section can have different Manning coefficients for every linear segment. The Manning coefficient for the trapezoidal approximation is taken equal to the equivalent Manning coefficient of the irregular cross section for a given steady state water level.

5.4.2 Formulation of the optimization problem

The optimization problem that needs to be solved at every time step is the following QP:

$$\min_{\mathbf{u},\mathbf{x},\boldsymbol{\xi},\boldsymbol{\zeta}} \sum_{j=1}^{N_\mathrm{P}} \|\mathbf{x}(j) - \mathbf{r}_x\|_{\mathbf{W}}^2 + \sum_{j=0}^{N_\mathrm{P}-1} \|\mathbf{u}(j) - \mathbf{u}(j-1)\|_{\mathbf{R}}^2 + \\ + \sum_{j=0}^{N_\mathrm{P}-1} \|\mathbf{u}(j) - \mathbf{r}_u\|_{\mathbf{U}}^2 + \|\boldsymbol{\xi}\|_{\mathbf{S}}^2 + \mathbf{s}^\mathrm{T}\boldsymbol{\xi} + \|\boldsymbol{\zeta}\|_{\mathbf{V}}^2 + \mathbf{v}^\mathrm{T}\boldsymbol{\zeta}$$

s.t. $\mathbf{x}(0) = \hat{\mathbf{x}}$,

$$\mathbf{x}(j+1) = \mathbf{A}\mathbf{x}(j) + \mathbf{B}\mathbf{u}(j) + \mathbf{F}\mathbf{d}(j) + \tilde{\boldsymbol{\beta}}(j), \qquad j = 0, \ldots, N_\mathrm{P} - 1 \quad (5.5)$$

$$\underline{\mathbf{u}}(j) \leq \mathbf{u}(j) \leq \overline{\mathbf{u}}(j), \qquad j = 0, \ldots, N_\mathrm{P} - 1$$

$$\mathbf{u}(-1) = \mathbf{u}_\mathrm{prev},$$

for $i = 1, \ldots, 5$:

$$\mathbf{M}^{(i)}\tilde{\mathbf{H}}^{(i)}\mathbf{x}(j) \leq \mathbf{M}^{(i)}\mathbf{h}_{\mathrm{max},1}^{(i)} + \mathbf{1}_{n_\mathrm{con}^{(i)}} \cdot \eta(j)\xi_i, \quad j = 1, \ldots, N_\mathrm{P}$$

$$\mathbf{M}^{(i)}\tilde{\mathbf{H}}^{(i)}\mathbf{x}(j) \leq \mathbf{M}^{(i)}\mathbf{h}_{\mathrm{max},2}^{(i)} + \mathbf{1}_{n_\mathrm{con}^{(i)}} \cdot \eta(j)\zeta_i, \quad j = 1, \ldots, N_\mathrm{P}$$

$$\tilde{\mathbf{H}}^{(\mathrm{schulen})}\mathbf{x}(j) \leq h_{\mathrm{max},1}^{(\mathrm{schulen})} + \eta(j)\xi_6, \qquad j = 1, \ldots, N_\mathrm{P}$$

$$\tilde{\mathbf{H}}^{(\mathrm{schulen})}\mathbf{x}(j) \leq h_{\mathrm{max},2}^{(\mathrm{schulen})} + \eta(j)\zeta_6, \qquad j = 1, \ldots, N_\mathrm{P}$$

$$\boldsymbol{\xi} \geq 0,$$

$$\boldsymbol{\zeta} \geq 0,$$

with N_P the prediction horizon, $\mathbf{W} \in \mathbb{R}^{n_x \times n_x} \succeq 0$, $\mathbf{R} \in \mathbb{R}^{n_u \times n_u} \succ 0$, $\mathbf{U} \in \mathbb{R}^{n_u \times n_u} \succeq 0$, $\mathbf{S} \in \mathbb{R}^{6\times 6} \succ 0$ and $\mathbf{V} \in \mathbb{R}^{6\times 6} \succ 0$ five diagonal weighting matrices, $\mathbf{s} \in \mathbb{R}^6$ and $\mathbf{v} \in \mathbb{R}^6$ two weighting vectors, $\mathbf{r}_x \in \mathbb{R}^{n_x}$ a vector containing the set-points for all the states, $\mathbf{r}_u \in \mathbb{R}^{n_u}$ a vector containing the set-points for the inputs, $\hat{\mathbf{x}} \in \mathbb{R}^{n_x}$ the current state of the process (measured or estimated), $\tilde{\boldsymbol{\beta}} \in \mathbb{R}^{n_x}$ (which will be defined in Section 5.4.2.3), $\underline{\mathbf{u}} \in \mathbb{R}^{n_u}$ and $\overline{\mathbf{u}} \in \mathbb{R}^{n_u}$ the lower and upper limits on the inputs, $\mathbf{u}_\mathrm{prev} \in \mathbb{R}^{n_u}$ the control action applied in the previous time step, $\tilde{\mathbf{H}}^{(i)} \in \mathbb{R}^{n_h^{(i)} \times (n_h + n_Q)}$ a matrix for selecting the water levels for the ith reach or the reservoir from the state vector $\mathbf{x}$, $\mathbf{h}_{\mathrm{max},1}^{(i)} \in \mathbb{R}^{n_h^{(i)}}$ the safety levels for the ith reach or the reservoir, $\mathbf{h}_{\mathrm{max},2}^{(i)} \in \mathbb{R}^{n_h^{(i)}}$ the flood

levels for the ith reach or the reservoir, $\mathbf{M}^{(i)} \in \mathbb{R}^{n_{\text{con}}^{(i)} \times n_h^{(i)}}$ a matrix for selecting $n_{\text{con}}^{(i)}$ safety and flood levels for the ith reach, $\eta(j) = 1/r_{\text{c}}^{j-1}$ a time-dependent inverse weight (with $r_{\text{c}} > 1$) and $\boldsymbol{\xi}$, $\boldsymbol{\zeta} \in \mathbb{R}^6$ two vectors of slack variables related to the safety and flood levels respectively. This QP is solved with `cplexqp` of IBM [76], which is a state-of-the-art QP solver. The different elements of the QP will now be explained in more detail.

The safety limits and flood limits are imposed as soft constraints in combination with the slack variables $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ to ensure that the QP is feasible at all times. More information on the use of slack variables is given in Section 4.3.3.1.

5.4.2.1 Choice of the weighting matrices and vectors

The weighting matrices $\mathbf{W}$, $\mathbf{R}$, $\mathbf{U}$, $\mathbf{S}$ and $\mathbf{V}$ and the weighting vectors $\mathbf{s}$ and $\mathbf{v}$ define the relative importance of the difference between the states and their set-points, the changes of the control actions, the difference between the inputs and their set-points and the two vectors of slack variables $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ respectively. The buffer capacity of the reservoir above its safety limit cannot be used for keeping the water levels of the channels below their safety limits. Therefore the diagonal element of $\mathbf{S}$ and the element of $\mathbf{s}$ corresponding to the reservoir will be set higher than the elements associated with the channels. To allow the controller to use the buffer capacity above the safety limit of the reservoir, hence to avoid the river from being flooded, the diagonal elements of $\mathbf{V}$ and the elements of $\mathbf{v}$ are set higher than all the elements of $\mathbf{S}$ and $\mathbf{s}$. If there is no flood risk, the controller needs to focus on set-point control. In this situation the controller should avoid using the reservoir. This is achieved by setting the set-point for the reservoir in $\mathbf{r}_x$ equal to 20.4 m TAW in combination with a large weight in the matrix $\mathbf{W}$ (but sufficiently smaller than the weights of the slack variables). At the same time the set-point $\mathbf{r}_u$ for the discharges controlled by gates A and D are set equal to zero together with a large weight on the corresponding diagonal elements of $\mathbf{U}$ to avoid any flow of water to the reservoir. To keep the downstream water level h_{up} of Dem_1 close to its set-point, the corresponding element in $\mathbf{r}_x$ is set equal to 21.5 m TAW in combination with a high weight in $\mathbf{W}$. All the other entries in $\mathbf{r}_x$ are set equal to their steady state values but in combination with a small weighting value. This allows the controller to let these water levels and discharges deviate from their set-points to react on (future) disturbances and keep the effects on h_{up} as small as possible. For the same reason the reference for gate K7 is set equal to its steady state value in combination with a small weighting element in $\mathbf{U}$. $\mathbf{R}$ influences the control effort of the different input variables.

5.4.2.2 Nonlinear prediction step

Because the QP works with the gate discharges as optimization variables, the lower and upper limit of the gates and the rate of change constraint need to be converted to a lower and upper limit on the gate discharges. This can be done in a similar way as in Section 4.4.3.1. Given the gate position for e.g. gate A at time step $k-1$ and its current upstream and downstream water level at time step k, the lower and upper limit on the gate discharge for time step k can easily be calculated with

$$\underline{u}^{(A)}(k) = \tilde{f}\Big(c^{(\mathrm{A})}(t_{k-1}) + \Delta_c^{(A)}, h_{\mathrm{up}}(t_k), h_{\mathrm{s}}(t_k)\Big) \tag{5.6}$$

$$\overline{u}^{(A)}(k) = \tilde{f}\Big(c^{(\mathrm{A})}(t_{k-1}) - \Delta_c^{(A)}, h_{\mathrm{up}}(t_k), h_{\mathrm{s}}(t_k)\Big), \tag{5.7}$$

with $\Delta_c^{(A)}$ the maximal rate of change for gate A. The same equations hold for gates K7 and D. For calculating the lower and upper limit for the time step $k+1$, an estimation is needed for the states at this time step. In the previous chapter, the linear approximate model was used to perform this prediction. However simulations have shown that the prediction with the linear model are not accurate enough when one is working with highly irregular river data for both the bed slopes and the cross sectional profiles. Therefore in this work a nonlinear prediction step is performed with the model of the Demer defined in Section 5.2.2. To keep the computation time limited, this prediction step performs only one Newton iteration and the internal time step Δt of the simulator is taken equal to the sampling time of the controller. This means that only one linear system is solved in order to get an estimate of the states at the next sampling time. Before performing this prediction step, the gate positions corresponding with the optimal gate discharges for time step k found by the optimizer in the previous iteration need to be calculated. Since there is in most situations a one-to-one relationship between the gate position and the gate discharge for a given upstream and downstream water level, the corresponding gate position can be easily found by means of a bisection search method. These gate positions are then used in combination with the current state of the system to estimate the water levels and discharges at $k+1$ (denoted with $\mathbf{x}_{\mathrm{nonlin}}(k+1)$). Based on $\mathbf{x}_{\mathrm{nonlin}}(k+1)$ and $c^{(A)}(k)$, $c^{(D)}(k)$ and $c^{(K7)}(k)$, Eqs. (5.6) and (5.7) can now be used to determine $\underline{\mathbf{u}}(k+1)$ and $\overline{\mathbf{u}}(k+1)$. The same procedure can be used to estimate the time-varying lower and upper limits of the gate discharges for the entire prediction window.

When the conversion is performed from the gate discharge to the corresponding gate position, it is ensured that the resulting gate position is always in its controllable region or at least at the boundary. For a gated weir this means that the gate will always be either in gate control mode or on the boundary. The gates will never be in throat control mode or the gate positions will never be

$\Delta_c^{(\text{gate})}$ higher than their surrounding water levels. This ensures that Eqs. (5.6) and (5.7) will always result in different values and the controller will not lose any control freedom. More information on uncontrollability of hydraulic structures is given in Section 4.4.3.2.

5.4.2.3 Updating of linear model

Simulation results show that the accuracy of the linear state space model is good enough for the controller to handle disturbance signals 3.5 times larger than their nominal value. However during periods of heavy rainfall, these disturbance signals can become 8 times larger and the linear state space model is not accurate enough. This problem can be overcome by performing an update of the linear state space model along the prediction horizon based on the nonlinear predictions $\mathbf{x}_{\text{nonlin}}(k+1)$, which are calculated for finding the limits on the gate discharges. At every time step j along the prediction horizon, the estimate of the next state based on the nonlinear model $\mathbf{x}_{\text{nonlin}}(k+1+j)$ is compared with the next state $\mathbf{x}_{\text{lin}}(k+1+j)$ predicted with the linear model:

$$\mathbf{x}_{\text{lin}}(k+1+j) = \mathbf{A}\mathbf{x}_{\text{nonlin}}(k+j) + \mathbf{B}\mathbf{u}_{\text{opt}}(k+j) + \mathbf{F}\mathbf{d}(k+j) + \boldsymbol{\beta},$$

with $\mathbf{u}_{\text{opt}}(k+j)$, $j = 0\ldots, N_{\text{P}} - 1$ the sequence of optimal gate discharges found by the controller in the previous iteration. The simplest way to match the prediction of the linear model with the prediction of the nonlinear model, is to replace $\boldsymbol{\beta}$ with

$$\tilde{\boldsymbol{\beta}}(k+j) = \boldsymbol{\beta} + (\mathbf{x}_{\text{nonlin}}(k+1+j) - \mathbf{x}_{\text{lin}}(k+1+j)),$$

which will be used inside the optimization problem (Eq. (5.5)). Recall that $\boldsymbol{\beta}$ corresponds with the local information contained in the linearization point. By performing this update on $\boldsymbol{\beta}$, this local information is corrected to match the nonlinear state estimates along the prediction horizon.

This model update has only a minor influence on the total computation time needed by the controller. For a condensed implementation of the QP, the model update only affects the linear part of the objective function $\mathbf{f}$ and the right hand side vector $\mathbf{b}_{\text{in}}$ of the inequality constraints. Because we only need to update two vectors, the total time needed for this operation is limited.

Another approach would be to use nonlinear MPC and work with time-varying state space matrices $\mathbf{A}(k)$, $\mathbf{B}(k)$ and $\mathbf{F}(k)$ derived from linearizing the nonlinear model along the nonlinear predicted state estimates. However deriving these linear models at every iteration along the prediction horizon takes a considerable amount of time. This also means that for a condensed implementation of the QP, the Hessian matrix $\mathbf{H}$ of the objective function and the matrix $\mathbf{A}_{\text{in}}$ containing the inequality constraint coefficients need to be

reconstructed for every iteration. Furthermore the results shown in Section 5.5 indicate that MPC in combination with the proposed model updating yields already a good performance.

5.4.2.4 Buffer capacity recovery

After a period of heavy rainfall the controller should recover the buffer capacity as fast as possible to be able to deal with possible future rainfall. This recovery is achieved by changing some of the values in the weighting matrices $\mathbf{W}$ and $\mathbf{U}$ and the reference signal $\mathbf{r}_x$. Once a part of the buffer capacity of the reservoir is used, the water levels of Dem_3 and Dem_4 get a set-point lower than the set-point of the reservoir and their weights in the matrix $\mathbf{W}$ are increased. At the same time the weight for h_{up} is decreased. Also the weights of $\mathbf{U}$ corresponding with the discharges over gates A and D are decreased. This means that the controller is not forced to keep these discharges close to zero and the controller has the freedom to use the buffer capacity if there is still a risk of flooding, or to get rid of the excess of water in the reservoir via these gates. This change in $\mathbf{W}$, $\mathbf{U}$ and $\mathbf{r}_x$ is performed when the water level in the reservoir is 20 cm above its set-point. When the water level of the reservoir is at the most 20 cm above its set-point for eight consecutive hours, $\mathbf{W}$, $\mathbf{U}$ and $\mathbf{r}_x$ are set back to their original values.

5.4.2.5 Constraint selection

Not all the flood and safety levels are included inside the optimization problem as an upper constraint on the water levels. A matrix $\mathbf{M}^{(i)} \in \mathbb{R}^{n_{\text{con}}^{(i)} \times n_h^{(i)}}$ is used to select $n_{\text{con}}^{(i)}$ safety levels and flood levels for the ith reach. The safety and flood levels are very irregular along every reach. This can be seen in Fig. 5.5 for the flood levels of reach Dem_3 together with its bedding and steady state water levels. The flood levels contain jumps along the reach in contrast to the smooth profile of the steady state water levels. Because of the irregularity of the safety and flood levels, it is not required to impose all the safety and flood levels as inequality constraints. Therefore, only the most critical ones are used inside the optimization problem. This has the advantage that the total number of inequality constraints, and hence the complexity of the QP, is drastically reduced and the computation time needed for solving the QP is decreased. The smaller the distance between the flood level and the water level in steady state at a grid point, the more critical this flood level is. This approach is very similar to the direct selection method proposed in Section 4.6.2.1.

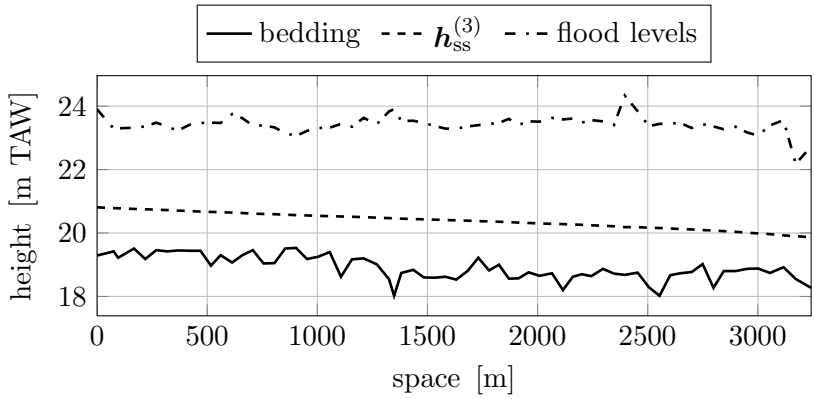


Figure 5.5: The bedding and the flood levels of the reach Dem_3 of the Demer together with the steady state water levels.

5.4.3 State estimator

The current state of the river system needs to be known before the optimization problem defined in the previous subsection can be solved. Because only the water levels h_s, h_{up} and h_{down} are measured in practice and none of the discharges, a state estimator is needed. The estimator used in this study will be a combination of a nonlinear prediction step together with a correction step of the classical Kalman filter introduced in Section 2.4.

The classical Kalman filter corrects the state estimation based on the linear state space model with the error between the estimated and the measured water levels via a feedback gain matrix $\mathbf{L}$:

$$\Delta\hat{\mathbf{x}}(k+1) = \mathbf{L}(\Delta\mathbf{y}(k) - \Delta\hat{\mathbf{y}}(k)) + \mathbf{A}\Delta\hat{\mathbf{x}}(k) + \mathbf{B}\Delta\mathbf{u}(k) + \mathbf{D}\Delta\mathbf{d}(k), \quad (5.8)$$

$$\Delta\hat{\mathbf{y}}(k) = \mathbf{C}\Delta\hat{\mathbf{x}}(k)$$

with $\Delta\mathbf{x}(k) = \mathbf{x}(k) - \mathbf{x}_{ss}$, $\Delta\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}(k) - \mathbf{x}_{ss}$, $\Delta\mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}_{ss}$, $\Delta\mathbf{y}(k) = \mathbf{y}(k) - \mathbf{y}_{ss}$, $\Delta\hat{\mathbf{y}}(k) = \hat{\mathbf{y}}(k) - \mathbf{y}_{ss}$, $\mathbf{y}(k)$ the measured water levels, $\hat{\mathbf{y}}(k)$ the estimated water levels and $\mathbf{C}$ a matrix which selects the measured water levels from the states.

The accuracy of this state estimator is good enough for disturbance signals 3.5 times larger than their nominal value. However during periods of heavy rainfall, the accuracy of the Kalman filter decreases. This problem can be overcome by replacing the linear prediction step in Eq. (5.8) with a state estimate $\mathbf{x}_{nonlin}(k+1)$ based on the nonlinear model:

$$\hat{\mathbf{x}}(k+1) = \mathbf{L}(\Delta\mathbf{y}(k) - \Delta\hat{\mathbf{y}}(k)) + \mathbf{x}_{nonlin}(k+1). \quad (5.9)$$

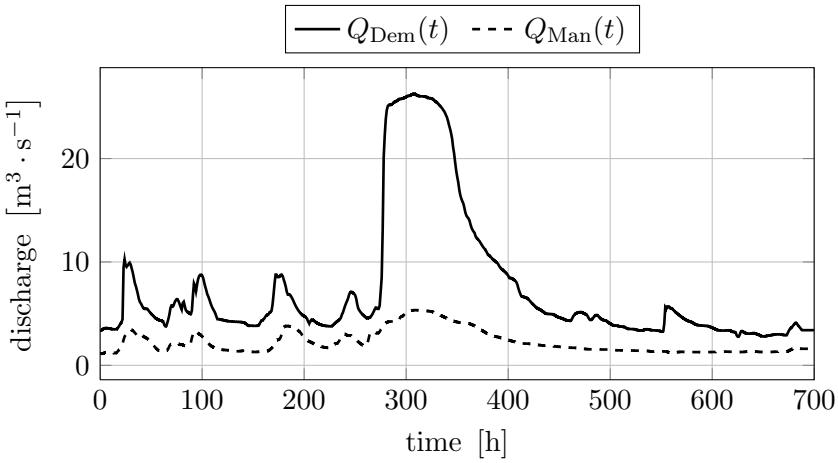


Figure 5.6: The inflowing discharges of the Demer and the Mangelbeek based on the historical rainfall data of the flood event of the Demer of 2002.

$\mathbf{x}_{\mathrm{nonlin}}(k+1)$ is found in the same way as the approach mentioned in Section 5.4.2.2.

5.5 Results

In this section the control performance of the proposed MPC is compared with the performance of the currently used three-position controller. The sampling time of both controllers is equal to 15 min. The disturbance signals are known 24 hours ahead in time. This means that the size of the prediction window N_{P} is taken equal to 96. The disturbance signals $Q_{\mathrm{Dem}}(t)$ and $Q_{\mathrm{Man}}(t)$ are based on the historical rainfall data of the flood event of 2002 and are visualized in Fig. 5.6. It is assumed that there is no uncertainty on these disturbance signals. Table 5.1 presents the limits on the three gates. The only variables that are measured at every time step are h_{up}, h_{s} and h_{down}. For the MPC the other state variables are estimated with the modified Kalman filter (Eq. (5.9)). Unlike the MPC, the three-position controller requires only the measured three water levels for determining the control actions. For Dem_1, Dem_3, Dem_4 and Man_5 only the ten most critical flood levels and safety levels are selected, while for Dem_2 all the six flood and safety levels are taken into account. The full hydrodynamic model of the Demer is used for the closed loop simulations.

Table 5.2 contains the diagonal elements for the weight matrices $\mathbf{W}$, $\mathbf{R}$, $\mathbf{U}$, $\mathbf{S}$ and $\mathbf{V}$, and the entries of the weight vectors $\mathbf{s}$ and $\mathbf{v}$. The chosen values

Table 5.1: Upper and lower limits and maximum rate of change for the gates A, K7 and D.

	gate A	gate K7	gate D
lower limit [m TAW]	20	20.03	18.9
upper limit [m TAW]	22.5	23	22.9
Δ_c [m] each 15 min	0.1	0.1	0.1

are in line with the reasoning given in Sections 5.4.2.1 and 5.4.2.4. During normal operation the controller focuses on keeping the most downstream water level of Dem_1 close to its set-point without using the buffer capacity of the reservoir. This is achieved with the matrices $\mathbf{W}$ and $\mathbf{U}$. When there is a risk of violating the safety or flood levels, the controller will automatically try to minimize these violations because of the high values in $\mathbf{S}$, $\mathbf{s}$, $\mathbf{V}$ and $\mathbf{v}$. The fast recovery of the buffer capacity after a period of heavy rainfall is achieved by replacing the elements in the matrices $\mathbf{W}$ and $\mathbf{U}$ with the marked values as indicated in Table 5.2. The low values in $\mathbf{R}$ ensure that the controller has sufficient freedom to achieve its objectives.

Fig. 5.7 shows the results for MPC in combination with the Kalman filter and the results for the three-position controller for the water levels h_{up}, h_{s} and h_{down}. The control actions for the three gates applied by both controllers are visualized in Fig. 5.8. MPC succeeds in keeping h_{up} closer to its set-point of 21.5 m TAW by using gate $K7$ than the three-position controller. One reason is because MPC can react on future disturbances. Another reason is that MPC keeps gate $K7$ always inside or at the boundary of its controllable region. MPC has the advantage that it can change the discharge over gate $K7$ at any time step, which is not the case with the three-position controller. This controller keeps e.g. decreasing the gate position of this gate when the water level is increasing but without any effect on the resulting gate discharge. Overall MPC attenuates the effect of the bumps in the disturbance signals on h_{up} much better than the three-position controller. Some of these bumps are completely absorbed with MPC. At the beginning of the simulation MPC brings gates A and D, which are initially uncontrollable, immediately at the boundary of their controllable region, without letting water enter the reservoir. Gate A remains 8 cm above h_{up}, while gate D remains 8 cm above its surrounding water levels (h_{s} and h_{down}). The three-position controller however keeps gate A constant during the first part of the simulation while it increases the position of gate D without any effect.

Before the period of heavy rainfall starts, MPC lowers the position of the three gates in order to decrease the water levels upstream of the Demer as much as

Table 5.2: Diagonal elements for the weight matrices $\mathbf{W}$, $\mathbf{S}$, $\mathbf{V}$, $\mathbf{R}$ and $\mathbf{U}$, and the elements for the weight vectors $\mathbf{s}$ and $\mathbf{v}$ for controlling the upstream part of the Demer.

	Dem_1	Dem_2	Dem_3	Dem_4	Man_5	Schulensmeer
$\mathbf{W} \in \mathbb{R}^{534\times534}$						
water levels	$\left[10^{-3}\cdot\mathbf{1}_{94};10\right]$ $\left[10^{-3}\cdot\mathbf{1}_{94};0.01\right]$ [a]	$10^{-3}\cdot\mathbf{1}_6$	$10^{-3}\cdot\mathbf{1}_{66}$ $0.01\cdot\mathbf{1}_{66}$ [a]	$10^{-3}\cdot\mathbf{1}_{11}$ $0.01\cdot\mathbf{1}_{11}$ [a]	$10^{-3}\cdot\mathbf{1}_{86}$	50
discharges	$10^{-3}\cdot\mathbf{1}_{96}$	$10^{-3}\cdot\mathbf{1}_7$	$10^{-3}\cdot\mathbf{1}_{67}$	$10^{-3}\cdot\mathbf{1}_{12}$	$10^{-3}\cdot\mathbf{1}_{87}$	
$\mathbf{S} \in \mathbb{R}^{6\times6}$						
safety levels	10^3	10^3	10^3	10^3	10^3	10^4
$\mathbf{s} \in \mathbb{R}^{6\times1}$						
safety levels	10^3	10^3	10^3	10^3	10^3	10^4
$\mathbf{V} \in \mathbb{R}^{6\times6}$						
flood levels	10^5	10^5	10^5	10^5	10^5	10^5
$\mathbf{v} \in \mathbb{R}^{6\times1}$						
flood levels	10^5	10^5	10^5	10^5	10^5	10^5

	$Q^{(\text{A})}$	$Q^{(\text{K7})}$	$Q^{(\text{D})}$
$\mathbf{R} \in \mathbb{R}^{3\times3}$			
control actions	0.01	0.01	0.01
$\mathbf{U} \in \mathbb{R}^{3\times3}$			
control actions	1000 0.001 [a]	0.001	1000 0.001 [a]

[a] Value used to recover the buffer capacity of the reservoir.

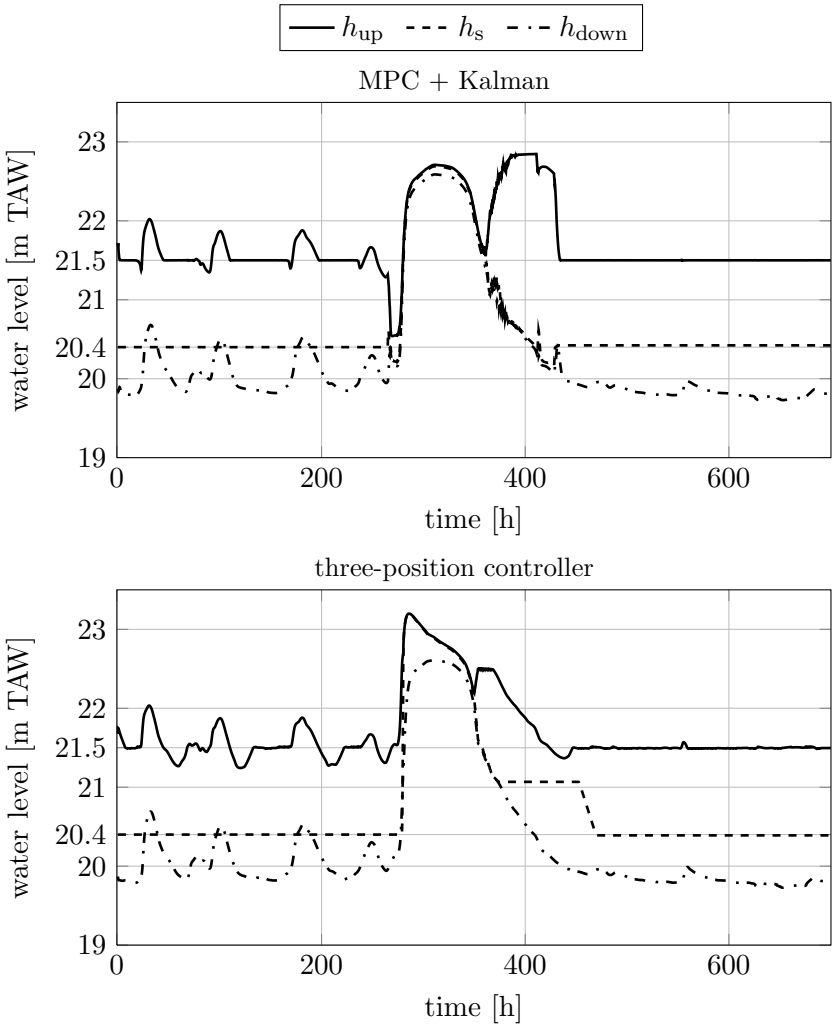


Figure 5.7: The evolution of the water levels h_{up}, h_{s} and h_{down} for MPC in combination with the Kalman filter (top) and the three-position controller (bottom). The set-points for h_{up} and h_{s} are 21.5 m TAW and 20.4 m TAW respectively. MPC keeps h_{up} closer to its set-point, it results in a lower maximal height of the water levels and it recovers the buffer capacity faster than the three-position controller.

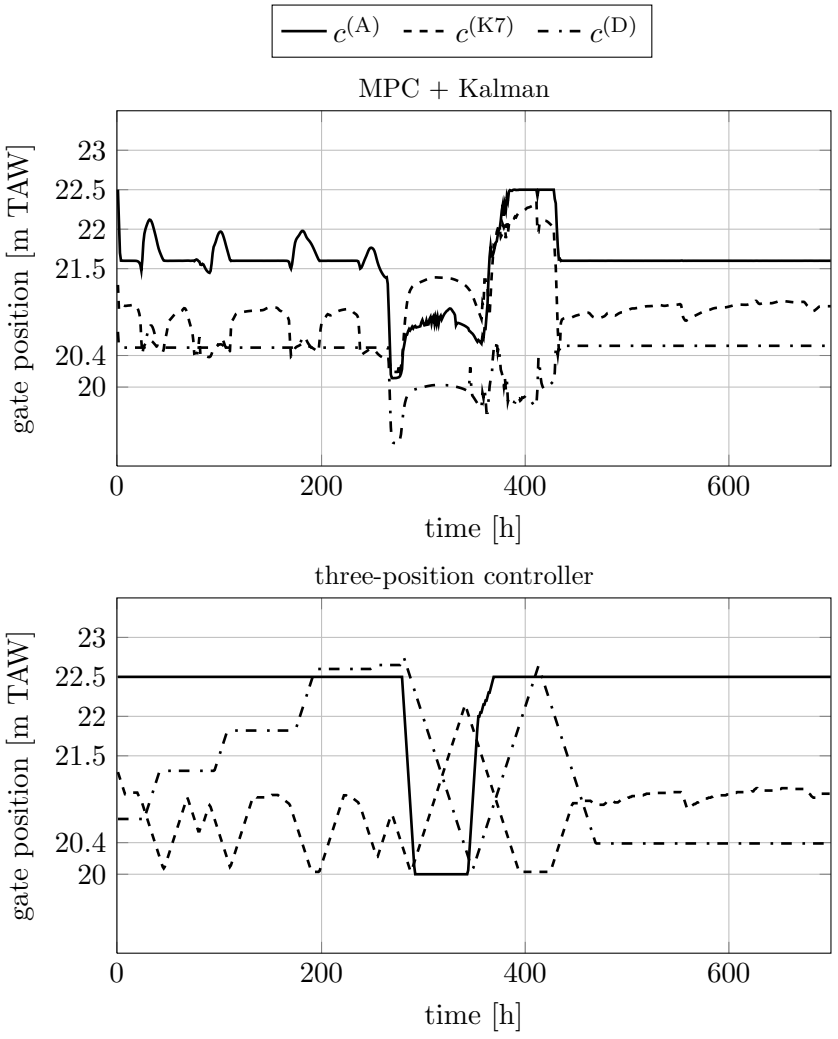


Figure 5.8: The evolution of the gate positions $c^{(A)}$, $c^{(K7)}$ and $c^{(D)}$ for MPC in combination with the Kalman filter (top) and the three-position controller (bottom). MPC prevents the gated weirs from becoming uncontrollable. This is not the case for the three-position controller.

possible. During the period of heavy rainfall it uses the three gates to minimize the floods along the five reaches. The three-position controller lowers gates A and D to start using the buffer capacity. However because the positions of these gates are too high at the beginning of the heavy rainfall period, the controller reacts too late, which results in much higher water levels. Fig. 5.9 and 5.10 show the maximal water level for both controllers for the five reaches together with their flood levels. The area between the flood levels and water levels has a gray color when flooding takes place with MPC and the area is hatched when flooding takes place with the three-position controller. The maximal water levels obtained after using MPC are always lower than the maximal water levels after the use of the three-position controller. Both controllers prevent Dem_2 and Dem_4 from flooding. MPC prevents Dem_1 from flooding at almost every grid point while the three-position controller results in more and larger floods along this reach. Dem_3 shows flooding for both controllers at only one point, while for Man_5 the flood level overtoppings with MPC are much smaller and at less locations than with the three-position controller. Table 5.3 shows the maximal flooding, the total flooding (i.e. the sum of the difference between the water level and the flood level, if positive, at each grid point for every reach at every minute) and the flood duration for both controllers for every reach. A negative value for the maximal flooding corresponds with the minimal margin before a flood level is violated. MPC clearly outperforms the three-position controller. The water levels for reaches Dem_2 and Dem_4 and the reservoir are decreased as well as the maximal violation of the flood levels for the other reaches. Also the total flooding and the flood duration improves when MPC is used to control the hydraulic structures.

After the period of heavy rainfall both controllers start emptying the reservoir. In order to recover the buffer capacity of the reservoir as fast as possible, MPC allows the water levels upstream of the Demer to approach their safety limit reducing the amount of water flowing to the reservoir and the downstream part of the Demer. Once the buffer capacity is recovered, MPC starts focussing again on steering h_{up} back to its set-point. Further bumps in the disturbance signals are completely absorbed by the controller: almost no variations can be seen on h_{up}. The three-position controller recovers the buffer capacity much slower. It also allows h_{up} to increase, however this increase is much smaller than with MPC (keep in mind that the three-position controller does not make use of the flood and safety levels). The decrease of the water level of the reservoir is stalled for 80 hours before the last part of the buffer capacity is recovered: the controller needs to wait before gate D becomes lower than the water level of the reservoir. Afterwards the controller steers h_{up} back to its set-point of 21.5 m TAW. However some variation in the water level is visible due to variation in the disturbance signals. MPC succeeds in recovering the buffer capacity 38 hours earlier than the three-position controller and it steers h_{up} back to its set-point 13 hours earlier than the three-position controller.

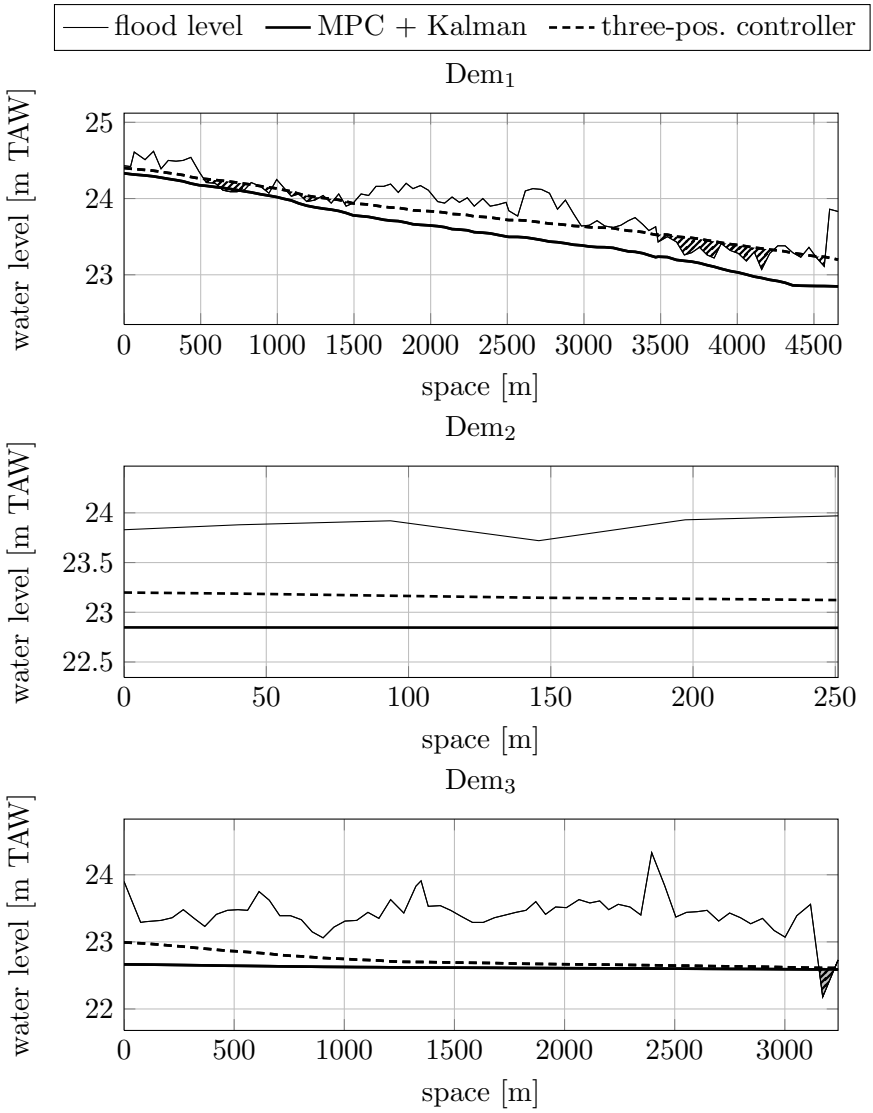


Figure 5.9: The maximal water levels for reaches Dem_1, Dem_2 and Dem_3 for MPC and the three-position controller together with their flood levels. The area between the flood levels and water levels has a gray color when flooding takes place with MPC and the area is hatched when flooding takes place with the three-position controller.

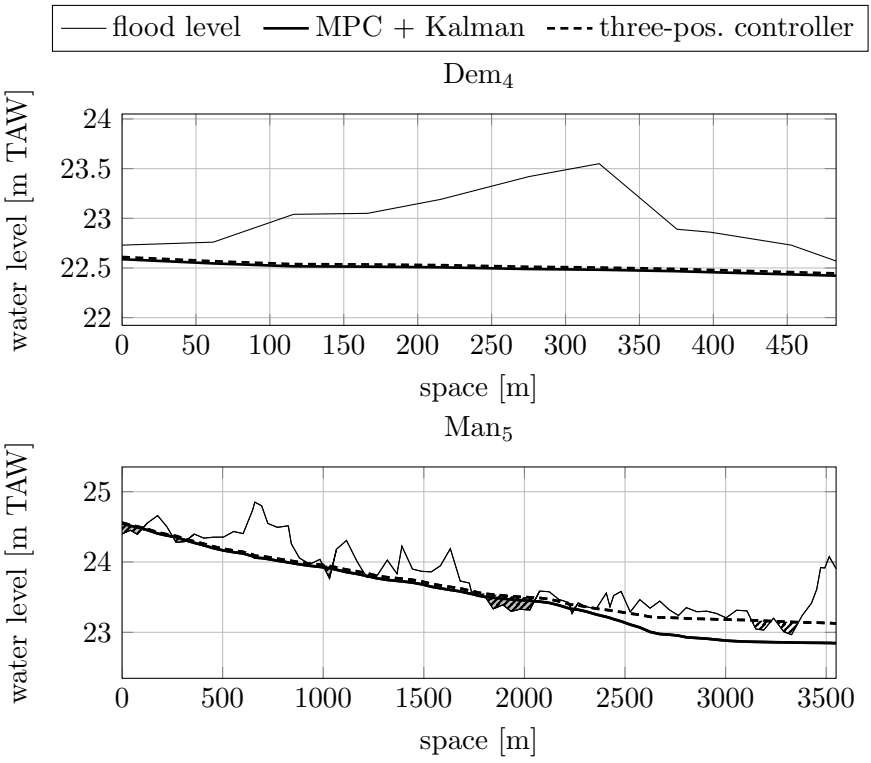


Figure 5.10: The maximal water levels for reaches Dem_4 and Man_5 for MPC and the three-position controller together with their flood levels. The area between the flood levels and water levels has a gray color when flooding takes place with MPC and the area is hatched when flooding takes place with the three-position controller.

The better set-point control and flood control capabilities of MPC in comparison with the three-position controller can also be seen in the set-point deviation cost J_h, calculated for the water level h_{up} for the simulation results before the period of heavy rainfall, and the flood cost $J_{h_{\max}}$ given in Table 5.4. MPC reacts in a more optimal way on the variations present in the disturbance signals for the first 250 hours resulting in a reduction of 15 % in the set-point deviation cost for h_{up}. The use of MPC results also in a reduction of the flood cost with 20 % in comparison with the three-position controller.

Fig. 5.11 shows the time needed by the controller to perform the prediction step and the time needed to solve the optimization problem at every sampling time with `cplexqp`. The time needed to make the conversion step after solving the optimization problem and the time needed for the Kalman filter are negligible

Table 5.3: Maximal flooding, total flooding and flood duration for the five reaches and the reservoir for both controllers. The maximal flooding is defined as the maximal difference between the flood level and the water level and the total flooding as the sum of the water level - flood level difference (if positive) at every grid point every minute. A negative value for the maximal flooding corresponds with the minimal margin before a flood level is violated.

	Dem_1	Dem_2	Dem_3	Dem_4	Man_5	Schulensmeer
maximal flooding [m]						
three-pos. controller	0.275	-0.573	0.432	-0.119	0.216	-0.006
MPC+Kalman	0.036	-0.875	0.409	-0.142	0.168	-0.509
total flooding [m]						
three-pos. controller	2877	0	1243	0	4096	0
MPC+Kalman	69	0	1138	0	2032	0
flood duration [h]						
three-pos. controller	49.1	0	63.1	0	60.5	0
MPC+Kalman	27.3	0	62.9	0	46.6	0

Table 5.4: The set-point deviation cost J_h for the water level h_{up} before the period of heavy rainfall and the flood cost $J_{h_{\max}}$ obtained with MPC and the three-position controller.

Simulation cost	MPC	three-pos. controller
J_h for h_{up} [m]	2138	2544
$J_{h_{\max}}$ [m]	1192	1495

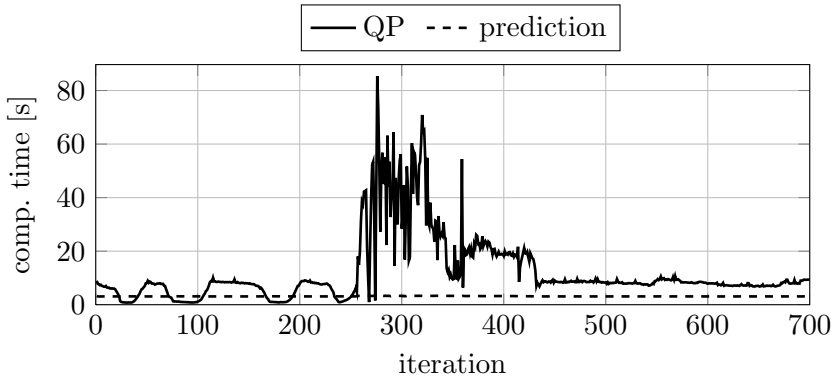


Figure 5.11: Computation time needed by the MPC controller to solve the optimization problem with `cplexqp` and the nonlinear prediction step at every sampling time (performed on a PC with a 3.1 GHz Intel Core i5 CPU and a RAM of 4 GB).

(less than 0.015 s, resp. 0.46 s). One can see that the time needed to solve the QP during the period of heavy rainfall stays well below the sampling time. However during this simulation test the state of the Demer is somehow "frozen": while the optimization problem is being solved, the states of the system are not changing in time while in reality they do. The larger the river system, the bigger this effect will be because of the longer computation time needed to solve the QP. The time for performing the prediction step is more or less the same at every iteration because the nonlinear model is solved with a fixed time step and with only one Newton iteration.

Table 5.5 shows the quantification of the total amount of control actions for the three controllers. For each gate m the total gate movement is calculated

Table 5.5: Total gate movement for the three gates for both controllers.

	$c^{(\mathrm{A})}$	$c^{(\mathrm{K7})}$	$c^{(\mathrm{D})}$
total gate movement [m]			
three-pos. controller	5	14.75	9.63
MPC+Kalman	17.16	21.46	16.87

with the formula

$$\gamma^{(\mathrm{m})} = \sum_{k=0}^{N_{\mathrm{T}}-1} \mid c^{(\mathrm{m})}(k+1) - c^{(\mathrm{m})}(k) \mid$$

where N_{T} is the total number of time instants during the simulation. The higher these numbers, the more demanding the controller is for the hydraulic structures. One can see that the three-position controller moves the three gates significantly less than MPC. This is the price that MPC has to pay to keep the gates at all times close to their controllable region. One could reduce these values for MPC by increasing the value of the diagonal elements in the weight matrix $\mathbf{R}$. However this will only effect the control actions when the gates are in their controllable regions. If the gate is near to its controllable region (e.g. gates A and D before and after the heavy rainfall period) then this gate will not be kept constant. The gate will follow the evolution of the surrounding water levels independently of the value used in $\mathbf{R}$. One approach to solve these "unnecessary" gate movements would be to first look at the predicted future disturbance signals. If no heavy rainfall would be predicted for the future period, then one could keep the gates connecting the river with the reservoir constant. Once a heavy rainfall event is predicted, the controller can also use these gates for flood prevention. This has the advantage that the amount of gate movement will drastically decrease for these gates.

5.6 Conclusions

In this chapter we have shown that Model Predictive Control can be used to control river systems based on real river field data. The Demer is used as a test example. The control scheme proposed in the previous chapter is adapted such that the controller can handle the irregularities of the Demer related to its bedding profile and the cross sectional profiles. The general working principle of the controller remains the same. The optimization problem is formulated as a function of the gate discharges instead of the gate openings in order to avoid the need of approximating the nonlinear equations modelling the gated weirs

installed at the Demer. The resulting optimization problem is a Quadratic Programming problem (QP). The use of slack variables for the safety and flood limits on the water levels ensures the feasibility of the QP at all times. The different weights related to the objective function can be chosen in such a way that the controller will focus on set-point control if possible, it will try to avoid any flooding and it will recover any used buffer capacity.

While in the previous chapter the lower and upper limits on the gate discharges along the prediction horizon could be found by performing a prediction step with the linear approximate model, this is not possible in this chapter due to the irregular river bedding and the irregular channel cross sections. This linear approximate model is accurate enough during periods of little rainfall, however its predictive power decreases during a heavy rainfall period. This problem is overcome by working with the nonlinear mathematical model of the Demer at the moment of performing this prediction. To keep the computation time limited, this nonlinear model works with an internal time step equal to the sampling time of the controller and performs only one Newton iteration. This means that for every time instant in the prediction window the nonlinear system of equations is linearized and solved only once (in contrast to the numerical solver used to simulate the river system). The simulation results indicate that these predictions are accurate enough to be used by the controller. These nonlinear predictions are also used to update the linear state space model along the prediction horizon used inside the optimization problem. The nonlinear prediction is compared to the prediction based on the linear model, and the vector containing the information related to the linearization point is updated to make the linear prediction match with the nonlinear one.

Because the flood and safety levels are not smooth along the river profile, not all the flood and safety levels are imposed as upper limits on the water levels. Only the most critical limits are included inside the optimization problem. The smaller the distance between the steady state water level and flood level, the more critical this flood level is considered to be. This constraint selection approach has the advantage that the number of inequality constraints can be reduced to a very small amount.

A Kalman filter is used to estimate the water levels and discharges of the entire river systems based on only three measured water levels. Just as we needed to perform the prediction step with a nonlinear model instead of a linear model, the Kalman filter needs to be adapted in a similar fashion. The classical Kalman filter combines a prediction step performed with the linear model with a correction step based on the difference between the measured outputs and the predicted outputs. The prediction step performed with the linear model is replaced with a prediction step based on the same nonlinear model used to find the lower and upper limits on the gate discharges.

The control performance was tested on a full hydrodynamical model of the Demer based on field data for the roughness coefficients, the river bedding, the cross sectional profiles and the gated weirs. The disturbances were modelled with the historical rainfall data of the floods of the Demer in 2002. The simulation results show that Model Predictive Control outperforms the three-position controller currently used for controlling the Demer:

- the water levels are kept closer to their set-points resulting in an improved set-point deviation cost with 15 %,
- the number of floods and their magnitude are reduced resulting in a reduction of the total flooding with 60 % and a reduction in the flood cost with 20 %
- and the buffer capacity is recovered in a fast way 38 hours earlier.

Another advantage of MPC is that the gated weirs are always in their gate control mode. This means that if a human operator wants to intervene manually, the change in the gate position will immediately have an effect on the gate discharge. This is not the case with the three-position controller where the gated weirs are often in an uncontrollable mode.

6

General Conclusions

6.1 Concluding remarks

The main goal of this dissertation was to test whether Model Predictive Control (MPC) can be used for set-point control and flood control of river systems in general. Regarding this research objective, the following concluding remarks can be made:

- In order to find a general approach for applying MPC to river systems, a **general modelling framework** was given for deriving a hydrodynamic model for river networks. Based on this framework, a new type of approximate model was introduced, the so-called **Linear-Nonlinear model** (LN-model). By approximating the dynamics of every reach with a linear state space model in combination with the nonlinear gate dynamics, the computational complexity of this LN-model is very low in contrast to the full hydrodynamic model, but it still achieves a high accuracy in approximating the original dynamics. In addition, Proper Orthogonal Decomposition (POD) and Galerkin projection can be used to reduce the computational complexity even further.
- Throughout Chapter 4 we have shown that MPC can be used for set-point control and flood control of different examples of river systems. Set-point control is achieved by minimizing the deviation of the most important water levels from their set-points, while flood control is ensured by imposing flood levels as upper limits on the water levels and incorporating

the effect of future rainfall predictions on the water levels over a prediction window. One key element of the proposed controllers is to work with the **gate discharges** as **optimization variables** instead of the gate positions. In this case a linear approximate model is accurate enough resulting in a linear predictive controller. The limits on the gate positions can be translated to limits on the gate discharges via a simple prediction procedure based on the linear state space model used by the controller. The optimization problem can be formulated in such a way that it remains feasible at all times (by using **slack variables**), the **buffer capacity** of reservoirs is only used when there is a risk of **flooding** and the used buffer capacity can be **recovered** in a fast way. At the end the controller needs to solve only one Quadratic Program at every time step.

- In practice only a limited number of water levels are measured and none of the discharges. We have shown that a classical **Kalman filter** can be used in most cases for finding an estimate of the current state of the systems based on these measurements.

- Different approaches have been introduced to limit the time needed to solve the optimization problem at every time step. The **number** of **optimization variables** can be reduced by working with a condensed implementation: all the states are eliminated as well as the model equations. The **number** of **inequality constraints** related to the upper limits on the water levels can be reduced in different ways. One approach simply imposes these upper limits on a coarser grid in contrast to the grid used for deriving the discretized equations. Another approach is to work with only the p most critical flood levels. A last approach works with a greedy selection algorithm exploiting the similarities between the coefficients of consecutive inequality constraints when we are working with a POD-based linear state space model. These approaches resulted in a significant reduction in the number of optimization variables and inequality constraints, leading to a significant saving of computation time while still achieving a good control performance. For one test example the number of optimization variables decreased from 9749 to 74 and the number of inequality constraints decreased from 9734 to 1004, resulting in a reduction of the total computation time with a factor more than 60.

- The effect of irregularities in the bed slope of reaches and in their cross sectional profiles on the control performance was tested by applying the proposed predictive control scheme to a hydrodynamic model of the **Demer** using real field data. A modified version of the predictive linear controller has been proposed in Chapter 5 in order to handle these irregularities: a **nonlinear prediction step** for finding the **limits** on the **gate discharges**, a simple **updating technique** of the linear models and a **nonlinear prediction step** used inside the **Kalman filter**

for estimating the unknown states. Simulation results for the historical rainfall data of the flood event of the Demer in 2002 clearly show that MPC outperforms the current three-position controller installed at the Demer:

- a better set-point control is achieved: the set-point deviation cost is improved with 15 %,
- the number and height of the floods is reduced resulting in a reduction of the total flooding with 60 % and a reduction of the flood cost with 20 %
- and the used buffer capacity is recovered 38 hours earlier.

6.2 Future research

Some important research objectives have already been achieved for the applicability of Model Predictive Control for flood control of river systems. However before this technique can be applied in practice, some remaining research challenges need to be tackled first:

- In this work the predictive controller is only applied to the upstream part of the Demer. A logical next step is to apply this technique to a much larger part of the Demer containing multiple reservoirs, more hydraulic structures, different types of hydraulic structures and more disturbance signals. The river system in Fig. 6.1 contains approximately seven times more state variables and four times more control variables compared to the river model used in Chapter 5. Due to the large increase in the number of optimization variables and the number of inequality constraints, the total computation time needed by the controller at every time step will become too large compared to the sampling time of 15 min. One approach to reduce this computation time, besides working with a faster QP solver, is to work with **Distributed MPC** (D-MPC) instead of using one centralized controller [6, 56, 57, 95, 129, 145]. In D-MPC, there are multiple agents/controllers solving at every time step a local MPC control problem. Interconnecting constraints are used to link the neighbouring agents to each other. An iteration scheme is needed to agree on the values of the interconnecting variables between the different controllers such that control actions that are optimal for the entire system can be found. Each of these local control problems are much smaller in size than the original control problem and they are solved in parallel, which reduces the total computation time at every time step.

  A river network can easily be divided into different subsystems. The Demer can be split into multiple subsystems where each subsystem

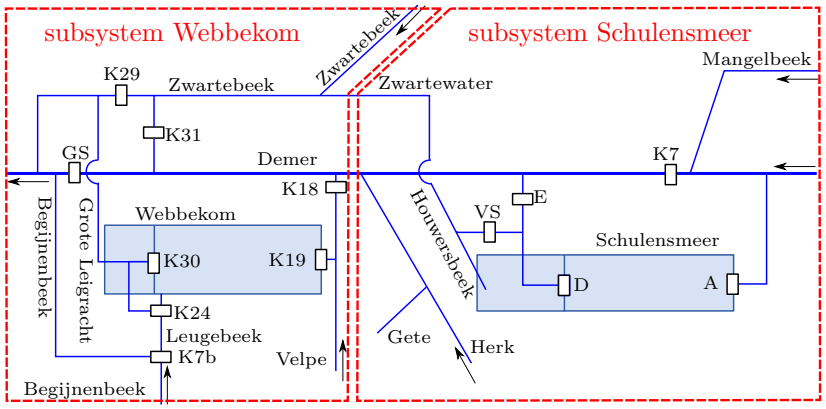


Figure 6.1: D-MPC strategy applied to the Demer. One local predictive controller controls the downstream part of the Demer containing the reservoir Webbekom, while another controller handles the upstream part of the Demer with the reservoir Schulensmeer.

contains one water reservoir. For example the river system in Fig. 6.1 can be divided into two systems where the number of interconnections between both systems is kept minimal. The hydraulic structures in each subsystem are controlled by a local MPC.

- The water flowing through the Demer ends up in the Dijle and eventually in the Scheldt. Just as we can develop a (possibly distributed) predictive controller for the Demer, we can develop a similar controller for the Dijle and the Scheldt. However, it is important that the control actions applied to the Demer does not lead to flooding problems along the Dijle or the Scheldt. Just as communication is needed between the local agents of a D-MPC, communication is needed between the predictive controllers for the different water basins. A solution is to work with **Hierarchical MPC** (H-MPC) [56, 57, 197]. H-MPC consists of different layers of controllers. The top layer consists of one high-level supervisor. This supervisor coordinates the entire network of river basins and sets the set-points and references for the controllers on the lower layer based on global information. This high-level supervisor controls the slowest dynamics of the entire river network. Each of the controllers in the lower layer can be designated to control one particular water basin, for example one controller for the Demer and another one for the Dijle. Such a controller can supervise on its turn the predictive controllers on the lowest layer, each one controlling one part of the water basin. These decentralized controllers use only local information and control the faster dynamics of a particular subsystem of the entire river network. The decentralized

controllers can work with the LN-models, while the controllers on the higher levels can work for example with conceptual models.

- In Chapter 5 we used the historical data of the flood event of 2002 to model the disturbance signals entering the Demer. It was assumed that the rainfall predictions over the prediction horizon are exactly known at every time step. However in reality these predictions are never 100 % accurate. Furthermore these predictions can be found in multiple ways resulting in different values. The further the predictions are ahead in time, the larger the uncertainty on these predictions. Before the LN-MPC can be applied in practice, it is necessary to asses the **robustness** of the controller against **uncertainty** on the **rainfall predictions**. A first approach would be to add noise on the rainfall predictions used by the controller. This noise should be correlated from one time step to the next in the prediction window: a time series predicting the rainfall too high at time k will typically also predict a too high value at time $k+1$. If simulation results would indicate a bad performance, one solution could be to work with Multiple Model Predictive Control (M-MPC) [179]. M-MPC solves one optimization problem taking a nominal, maximal and minimal prediction of the rainfall into account.

- Besides the robustness against uncertainty on the rainfall predictions, it is also important to asses the **robustness** of the controller with respect to **plant-model-mismatches**. The obtained simulation results in this work already indicate that the controller is robust against changes in the Manning roughness coefficient: one equivalent Manning coefficient was used for every grid point for deriving the linear state space model, while the nonlinear model used for simulating the Demer changes the used equivalent Manning coefficient depending on the current water level. However a simplified version of this nonlinear model is used for performing model updates, finding the upper limits on the gate discharges and inside the state estimator. To check the robustness of the controller, one could use different roughness coefficients for the nonlinear model used by the controller and the nonlinear model used for simulating the river system. Another step to test the robustness of the controller would be to connect the controller with the highly accurate InfoWorks model of the Demer.

A

Modelling of Hydraulic Structures

This chapter shows the equations used to model the discharge controlled with a gated weir or the discharge flowing over a spill. These equations are used in Chapter 5 where a mathematical model is derived for the Demer. Section A.1 gives the equations and the parameters for a gated weir, while Section A.2 shows the equations and the parameters for a spill. The equations are based on the manual of the InfoWorks-RS software [78].

A.1 Gated weir

The working principles of a gated weir are introduced in Section 3.3.3.1. The gate discharge can be modelled with the following equation:

$$Q_{\text{gate}}(t) = \tilde{f}(c(t), h_{\text{up}}(t), h_{\text{down}}(t)),$$

with c the gate position, h_{up} the upstream water level, h_{down} the downstream water level and $\tilde{f}$ a scalar nonlinear function. Every gated weir has the following parameters:

- w: the width of the sluice [m],
- z_{c}: the sill level [m TAW],
- h_{g}: the gate depth [m],
- m: the modular limit [−],

- c_t: throat discharge coefficient [−],
- c_g: gate discharge coefficient [−],
- b_up: the height of the bedding upstream of the gate [m TAW],
- b_down: the height of the bedding downstream of the gate [m TAW].

The algorithm for implementing the function $\tilde{f}$ first transforms the water levels and the gate opening to the reference height for altimetry in Belgium:

$$y_\mathrm{up} = h_\mathrm{up} + b_\mathrm{up},$$

$$y_\mathrm{down} = h_\mathrm{down} + b_\mathrm{down},$$

$$y_\mathrm{c} = c + z_\mathrm{c}.$$

The exact formula used for calculating the gate discharge depends on the flow condition and the control mode (where we assume that $y_\mathrm{up} > y_\mathrm{down}$):

- no flow (if $y_\mathrm{up} < y_\mathrm{c}$):

$$Q_\mathrm{gate} = 0.$$

- free weir flow - throat control mode:

$$f_\mathrm{t} = \begin{cases} \left(1 - \frac{y_\mathrm{down} - z_\mathrm{c}}{y_\mathrm{up} - z_\mathrm{c}}\right)/(0.3 \cdot (1-m)), & \text{if } \frac{y_\mathrm{down} - z_\mathrm{c}}{y_\mathrm{up} - z_\mathrm{c}} > m \\ 1, & \text{otherwise} \end{cases}$$

$$Q_\mathrm{gate} = 0.5 \cdot c_\mathrm{t} \cdot w \cdot \sqrt{g} \cdot (y_\mathrm{up} - z_\mathrm{c})^{1.5} \cdot f_\mathrm{t}.$$

- drowned weir flow - throat control mode:

$$f_\mathrm{t} = \sqrt{\left(1 - \frac{y_\mathrm{down} - z_\mathrm{c}}{y_\mathrm{up} - z_\mathrm{c}}\right)/(1-m)},$$

$$Q_\mathrm{gate} = 0.5 \cdot c_\mathrm{t} \cdot w \cdot \sqrt{g} \cdot (y_\mathrm{up} - z_\mathrm{c})^{1.5} \cdot f_\mathrm{t}.$$

- free weir flow - gate control mode:

$$f_\mathrm{g} = \begin{cases} \left((1 - \frac{y_\mathrm{down} - y_\mathrm{c}}{y_\mathrm{up} - y_\mathrm{c}}\right)/(0.3 \cdot (1-m)), & \text{if } \frac{y_\mathrm{down} - y_\mathrm{c}}{y_\mathrm{up} - y_\mathrm{c}} > m \\ 1, & \text{otherwise} \end{cases}$$

$$h_\mathrm{p} = \frac{y_\mathrm{up} - y_\mathrm{c}}{y_\mathrm{c} - z_\mathrm{c}},$$

$$\theta = 57.3 \cdot \operatorname{asin}\left(\frac{y_c - z_c}{h_g}\right),$$

$$\phi = \frac{\theta - 30}{60},$$

$$\psi = \begin{cases} 0.711 \cdot (1 - \phi) + 0.58 \cdot \phi \cdot (1 + 0.13 \cdot h_p), & \text{if } \theta \geq 30 \\ 0.711, & \text{otherwise} \end{cases}$$

$$Q_{gate} = c_g \cdot w \cdot \sqrt{g} \cdot \psi \cdot (y_{up} - y_c)^{1.5} \cdot f_g.$$

- drowned weir flow - gate control mode:

$$f_g = \sqrt{\left(1 - \frac{y_{down} - y_c}{y_{up} - y_c}\right)/(1 - m)},$$

$$h_p = \frac{y_{up} - y_c}{y_c - z_c},$$

$$\theta = 57.3 \cdot \operatorname{asin}\left(\frac{y_c - z_c}{h_g}\right),$$

$$\phi = \frac{\theta - 30}{60},$$

$$\psi = \begin{cases} 0.711 \cdot (1 - \phi) + 0.58 \cdot \phi \cdot (1 + 0.13 \cdot h_p), & \text{if } \theta \geq 30 \\ 0.711, & \text{otherwise} \end{cases}$$

$$Q_{gate} = c_g \cdot w \cdot \sqrt{g} \cdot \psi \cdot (y_{up} - y_c)^{1.5} \cdot f_g.$$

The gated weir in throat control mode is in free flow condition if $(y_{down} - z_c)/(y_{up} - z_c) < m$, otherwise it is in submerged flow condition. If the structure is in gate control mode, then it is in free flow condition if $(y_{down} - y_c)/(y_{up} - y_c) < m$, otherwise it is in submerged flow condition. The gate is in throat control mode if the discharge calculated with the formula for the throat control mode (for the correct flow condition) is smaller than the discharge calculated for the gate control mode (for the correct flow condition). Otherwise it is in gate control mode. Similar functions are used when the flow is in the opposite direction ($y_{up} < y_{down}$). More information can be found in [78].

A.2 Spill

Section 3.3.3.2 introduces a spill as an example of an hydraulic structure for which the discharge cannot be influenced by an operator. The discharge over a spill can be modelled with the scalar nonlinear function $\tilde{g}$:

$$Q_{\text{spill}}(t) = \tilde{g}(h_{\text{up}}(t), h_{\text{down}}(t))$$

with h_{up} the upstream water level and h_{down} the downstream water level. Depending on the upstream and downstream water level and the shape of the crest of the spill, the width of the spill is divided into multiple segments. The flows over each of these segments are calculated separately and are added together to get the total flow Q_{spill} over the spill.

Each of these segments has the following parameters:

- w: the width of the segment [m],
- m: the modular limit [−],
- c_{w}: the weir coefficient [−],
- k_1: the height of the lowest part of the top of the segment [m TAW],
- k_2: the height of the highest part of the top of the segment [m TAW].

Given the water levels $y_{\text{up}} = h_{\text{up}} + b_{\text{up}}$ and $y_{\text{down}} = h_{\text{down}} + b_{\text{down}}$ with b_{up} and b_{down} the heights of the bedding at the upstream and downstream side of the spill respectively, the following variables are calculated:

$$y_{\text{up},1} = y_{\text{up}} - k_1,$$

$$y_{\text{up},2} = y_{\text{up}} - k_2,$$

$$y_{\text{down},1} = y_{\text{down}} - k_1,$$

$$y_{\text{down},2} = y_{\text{down}} - k_2,$$

and are used to determine the flow over the segment. The flow over such a segment depends on the flow condition:

- no flow (if $y_{\text{up},1} < 0$, $y_{\text{down},1} < 0$):

$$Q_{\text{segment}} = 0.$$

- free flow:

$$Q_{\text{segment}} = 2 \cdot c_{\text{w}} \cdot w \cdot \frac{y_{\text{up},2}^{5/2} - y_{\text{up},1}^{5/2}}{5 \cdot (y_{\text{up},2} - y_{\text{up},1})}.$$

- submerged flow:

$$y_{\text{k}} = y_{\text{up},2} - y_{\text{up},1} - y_{\text{down},2} + y_{\text{down},1},$$

$$A = -\frac{c_{\text{w}}/y_k^2}{\sqrt{1-m}},$$

$$y_{\text{i}} = y_{\text{up},2} - y_{\text{up},1},$$

$$d_{\text{down},1} = (y_{\text{up},2} - y_{\text{down},2})^{1.5},$$

$$d_{\text{down},2} = (y_{\text{up},2} - y_{\text{down},2})^{2.5},$$

$$d_{\text{up},1} = (y_{\text{up},1} - y_{\text{down},1})^{1.5},$$

$$d_{\text{up},2} = (y_{\text{up},1} - y_{\text{down},1})^{2.5},$$

$$D = y_{\text{up},2} \cdot d_{\text{down},1} - y_{\text{up},1} \cdot d_{\text{up},1},$$

$$Q_{\text{segment}} = A \cdot w \cdot \left(\frac{2}{3} \cdot y_{\text{k}} \cdot D - \frac{4}{15} \cdot y_{\text{i}} \cdot (d_{\text{down},2} + d_{\text{up},2}) \right).$$

The segment is considered to be in free flow if $\frac{y_{\text{down},1}+y_{\text{down},2}}{y_{\text{up},1}+y_{\text{up},2}} \leq m$, otherwise it is in submerged flow. More information can be found in [78].

B

Test examples

This chapter shows the simulation results for the numerical scheme defined in Section 3.5.2. Section B.1 focusses on test systems consisting of only one reach. First the steady state solution calculated with the numerical scheme is compared with the analytic solution for two examples. Afterwards a mass conservation test is performed. The last two examples visualize the effect of a time-varying given upstream and/or downstream discharge as boundary conditions on the dynamics of a reach. Section B.2 gives one result for two reaches connected to each other with a hydraulic structure. In this example the concept of "uncontrollability" related to hydraulic structures is explained.

B.1 Single reach

B.1.1 Steady state problem with analytic solution

In the literature one can find problem descriptions where an analytic solution exists for the steady state case. E.g. different examples are given in [113, 114]. In those reports a method is presented for constructing test problems with known analytic solutions to the full steady Saint-Venant equation (Eq. (3.24)). The presented method is actually an "inverse" method. Based on a chosen hypothetical water level profile, a bed slope can be found which makes this profile a solution of the steady Saint-Venant equation. This method will be used here to test how far the steady state water level profile calculated with the numerical simulator differs from the analytic steady state values.

Consider a 1 km long and 10 m wide rectangular channel, with a Manning roughness coefficient of 0.03. For the water level profile

$$h(z) = \left(\frac{4}{g}\right)^{1/3}\left(1 + \frac{1}{2}\exp\left(-16\left(\frac{z}{1000} - \frac{1}{2}\right)^2\right)\right) \tag{B.1}$$

and its spatial derivative

$$\frac{\mathrm{d}h(z)}{\mathrm{d}z} = -\left(\frac{4}{g}\right)^{1/3}\frac{2}{125}\left(\frac{z}{1000} - \frac{1}{2}\right)\exp\left(-16\left(\frac{z}{1000} - \frac{1}{2}\right)^2\right)$$

its corresponding bed slope that makes this water level profile the steady state solution of the channel with a steady state discharge of 20 $\mathrm{m}^3 \cdot \mathrm{s}^{-1}$ is given by

$$S_0(z) = \left(1 - \frac{4}{gh(z)^3}\right)\frac{\mathrm{d}h(z)}{\mathrm{d}z} + 0.36\frac{(2h(z)+10)^{4/3}}{(10h(z))^{10/3}}.$$

Fig. B.1 compares the analytic solution for the water level profile (Eq. (B.1)) with the one calculated by means of the procedure explained in Section 3.5.2.3, where the downstream water level was used as boundary condition and n_h was set to 101. The top plot visualizes the analytic solution and the calculated solution, while the middle plot shows the absolute error between both profiles. The maximal error is smaller than 1.6 mm. The bottom plot shows the calculated steady state water profile in combination with the bed slope of the channel.

The second example is derived for a 5 km long trapezoidal channel with a bottom width B of 10 m and side slopes S_1 and S_2 equal to 2. Its Manning roughness coefficient is equal to 0.03. The channel has a bed slope given by

$$S_0(z) = \left(1 - \frac{400(10+4h(z))}{g(10+2h(z))^3h(z)^3}\right)\frac{\mathrm{d}h(z)}{\mathrm{d}z} + 0.36\frac{\left(10+2h(z)\sqrt{5}\right)^{4/3}}{(10+2h(z))^{10/3}h(z)^{10/3}}$$

with

$$h(z) = \frac{9}{8} + \frac{1}{4}\sin\left(\frac{\pi z}{500}\right),$$

$$\frac{\mathrm{d}h(z)}{\mathrm{d}z} = \frac{\pi}{2000}\cos\left(\frac{\pi z}{500}\right).$$

A comparison between the analytic solution and the calculated solution (for $n_h = 501$) is given in Fig. B.2. The top plot and the middle plot show that the calculated steady state water levels are good approximations of the analytic steady state water levels. The maximal error is less than 1.5 mm. The bottom

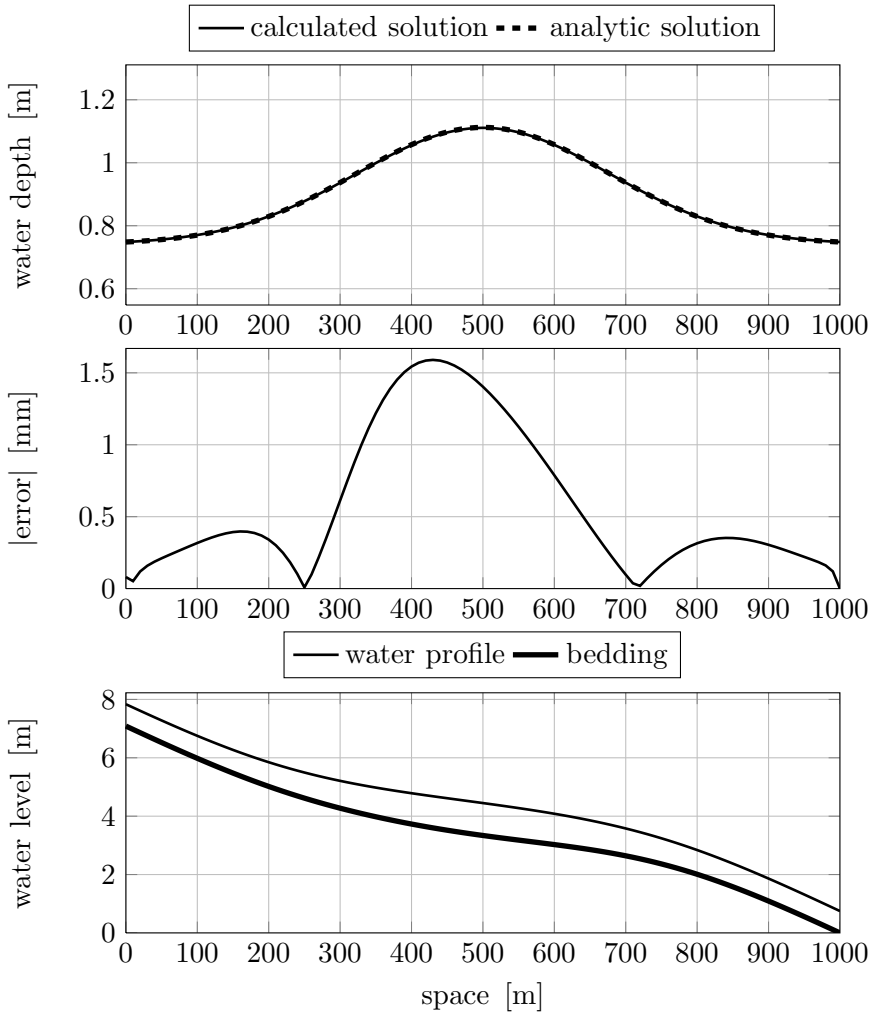


Figure B.1: Comparison between the analytic steady state water level profile and the profile calculated by the numerical simulator ($n_h = 101$) for the bed slope $S_0(z) = \left(1 - \frac{4}{gh(z)^3}\right)\frac{\mathrm{d}h(z)}{\mathrm{d}z} + 0.36\frac{(2h(z)+10)^{4/3}}{(10h(z))^{10/3}}$. The top plot shows both profiles while the middle plot shows the absolute difference between both profiles. The bottom plot visualizes the water profile together with the channel bedding.

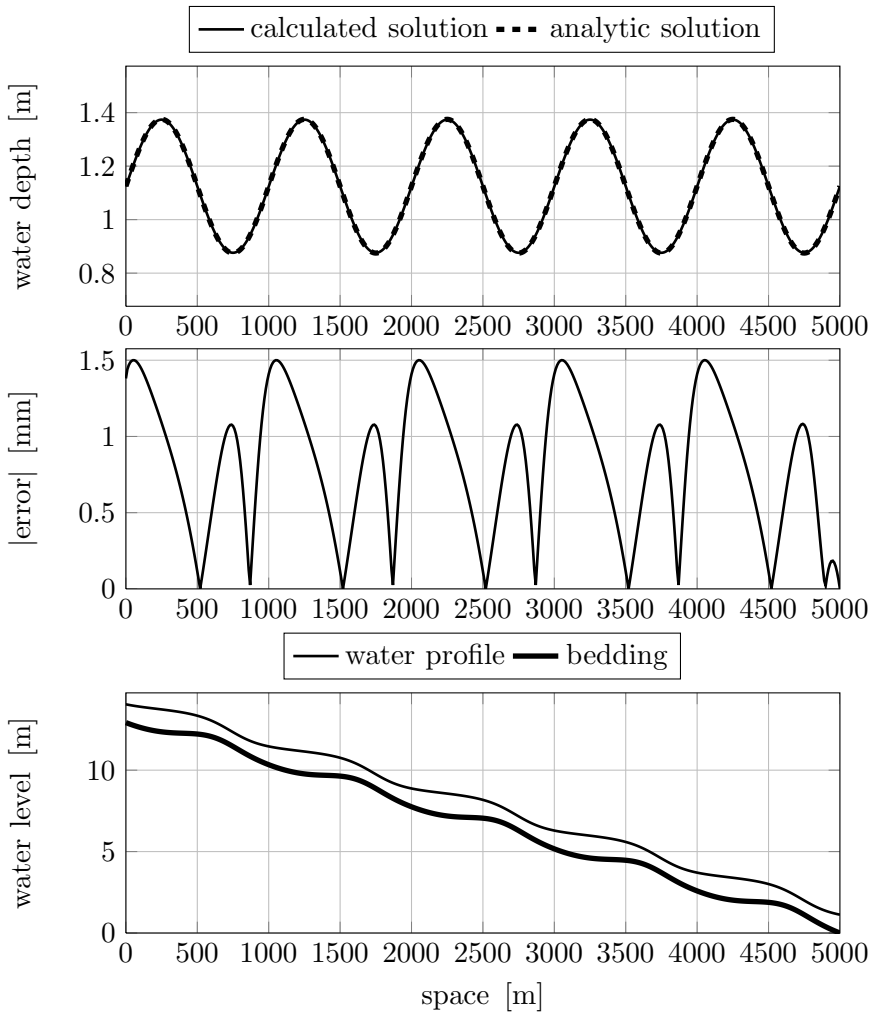


Figure B.2: Comparison between the analytic steady state water level profile and the profile calculated by the numerical simulator ($n_h = 501$) for the bed slope $S_0(z) = \left(1 - \frac{400(10+4h(z))}{g(10+2h(z))^3 h(z)^3}\right)\frac{\mathrm{d}h(z)}{\mathrm{d}z} + 0.36\frac{\left(10+2h(z)\sqrt{5}\right)^{4/3}}{(10+2h(z))^{10/3}h(z)^{10/3}}$. The top plot shows both profiles while the middle plot shows the absolute difference between both profiles. The bottom plot visualizes the water profile together with the channel bedding.

plot shows the calculated steady state water profile together with the channel bed slope.

Both examples indicate that the steady state solution of the discretized Saint-Venant equations is a good approximation of the steady state solution of the original full set of equations.

B.1.2 Mass conservation test

This test checks whether the scheme conserves mass and hence volume [50]. An horizontal trapezoidal channel is used with as initial condition no flow and a constant water depth h_0. At the upstream end the discharge varies sinusoidally for one period T while at the downstream end the discharge remains zero at all times. Since the inflow results in a zero net volume flux, the total volume should not be changed after a period T. When the channel is back in steady state, the water depth should again be equal to h_0. The following parameters are used [50]:

- Canal geometry: a horizontal canal of 10 km (L) with a trapezoidal section with a bottom width B of 10 m and side slopes S_1, S_2 equal to 2. The Manning coefficient n_{mann} is taken equal to 0.04 $\text{s} \cdot \text{m}^{-1/3}$.
- Initial condition: there is no flow along the entire channel and its water depth h_0 is equal to 7 m.
- Boundary conditions: the downstream discharge is equal to 0 $\text{m}^3 \cdot \text{s}^{-1}$ while the upstream discharge varies for the first 6 hours sinusoidally in time $Q = 200 \sin(2\pi t/21600)$, afterwards it is kept equal to 0 $\text{m}^3 \cdot \text{s}^{-1}$ for the rest of the simulation.

The simulation results for $n_h = 51$ for the upwind scheme for the first 6 hours are visualized in Fig. B.3. The maximal relative difference between the water levels after 200 hours and their analytical steady state values is $3.08 \cdot 10^{-5}$, which is satisfactory small.

B.1.3 Linear increase of downstream discharge and constant upstream water level

The last two tests are used to visualize the behavior of the water levels and discharges of a single reach for some simple boundary conditions. The first test is applied to a trapezoidal channel with bottom width B equal to 2 m and side slopes S_1 and S_2 of 1.5, the channel is 1520 m long and has a Manning

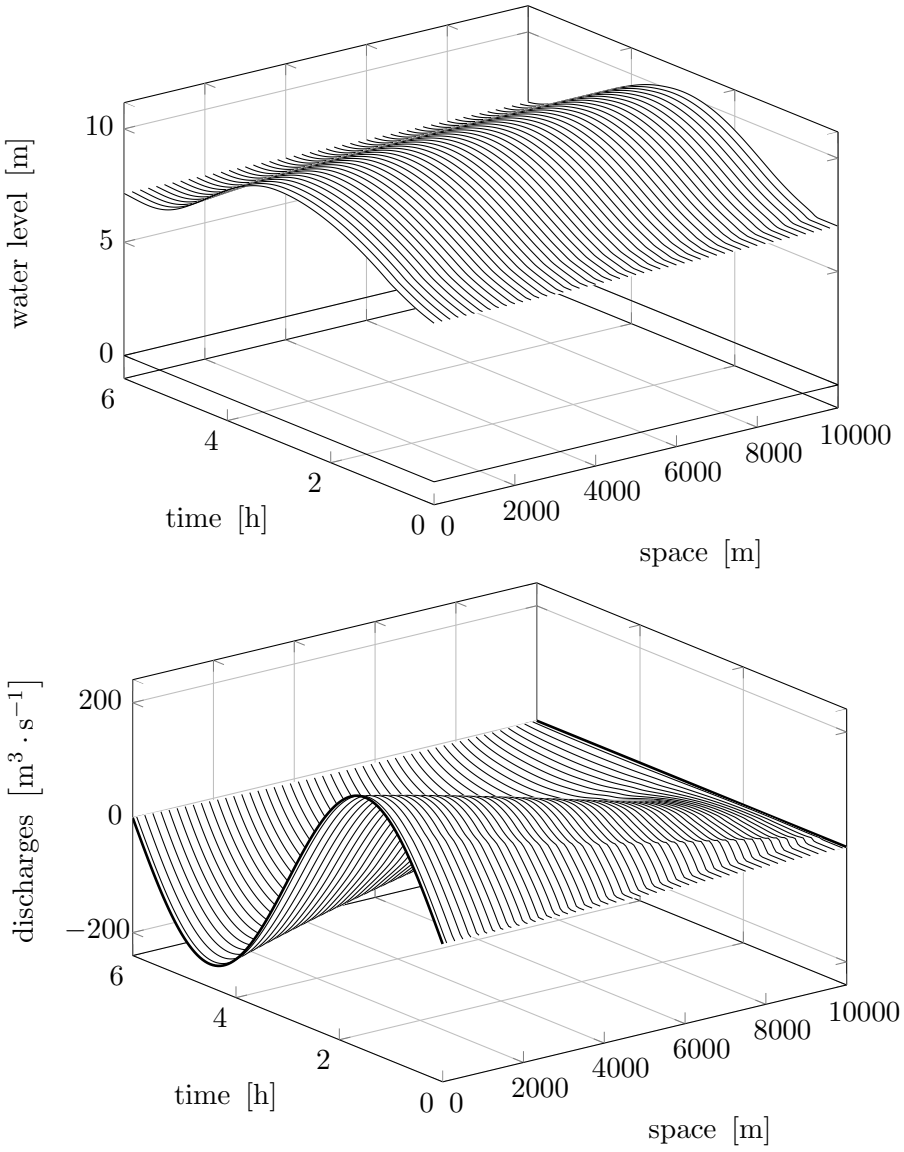


Figure B.3: Simulation results for the test regarding the conservation of mass of the numerical simulator for $n_h = 51$. The top plot shows the evolution of the water levels in space and time for the first 6 hours. The plane on the bottom represents the bedding of the reach. The bottom plot shows the evolution of the discharges in space and time for the first 6 hours. The bold lines corresponds with the upstream and downstream boundary conditions. The upstream discharge varies sinusoidally in time $Q = 200 \sin(2\pi t/21600)$ while the downstream discharge is kept constant at 0 $\mathrm{m}^3 \cdot \mathrm{s}^{-1}$.

coefficient of 0.015 $\mathrm{s} \cdot \mathrm{m}^{-1/3}$. The bed slope of the channel is equal to 0.0002. Initially there is no flow in the channel and its upstream water level is equal to 3.5 m. The channel is connected to a reservoir at its upstream side, resulting in a constant water level of 3.5 m during the simulation. The downstream side is connected to a pump which is initially off. At time $t = 0$ the pump starts working and its flow rate increases linearly to its maximal value of 28.3 $\mathrm{m}^3 \cdot \mathrm{s}^{-1}$ after 60 s. Afterwards it remains constant at this value:

$$Q_{\mathrm{out}}(t) = \begin{cases} 0 & t < 0, \\ 28.3\, t/60 & 0 \leq t < 60, \\ 28.3 & 60 \leq t. \end{cases}$$

Fig. B.4 shows the evolution of the water levels and discharges along the channel in space and time for the first 4000 s. These water levels and discharges are calculated for $n_h = 51$. The bold lines represent the boundary conditions. The increase in the discharge at the downstream end of the channel results in a wave (with negative amplitude) travelling towards the upstream side of the channel. This wave is reflected at the upstream side. To keep the upstream water level constant, the upstream discharge is increased which cause a second wave towards the downstream end of the channel.

B.1.4 Changing upstream and downstream discharges

The last test performed on a single reach is where the upstream and downstream discharges are changing in time. The rectangular channel is 4 km long and 2 m wide. It has a Manning roughness coefficient of 0.05 $\mathrm{s} \cdot \mathrm{m}^{-1/3}$ and a bed slope of 0.001. Fig. B.5 shows the time-varying boundary discharges. The effect of these boundary conditions on the channel water levels and channel discharges for $n_h = 51$ are visualized in Fig. B.6.

B.2 Multiple reaches

Until now all the given examples were for a single reach. In this part one result will be shown for a simple river system. More advanced examples are given in Chapters 4 and 5.

The example consists of two reaches connected with each other through a gated weir. The characteristics of both channels are given in Table B.1. The gated weir has the same width as the bottom width of both channels and its sill level is 0.5 m above the bedding at the downstream end of the first channel and 0.8 m above the bedding at the upstream side of the second channel. This means

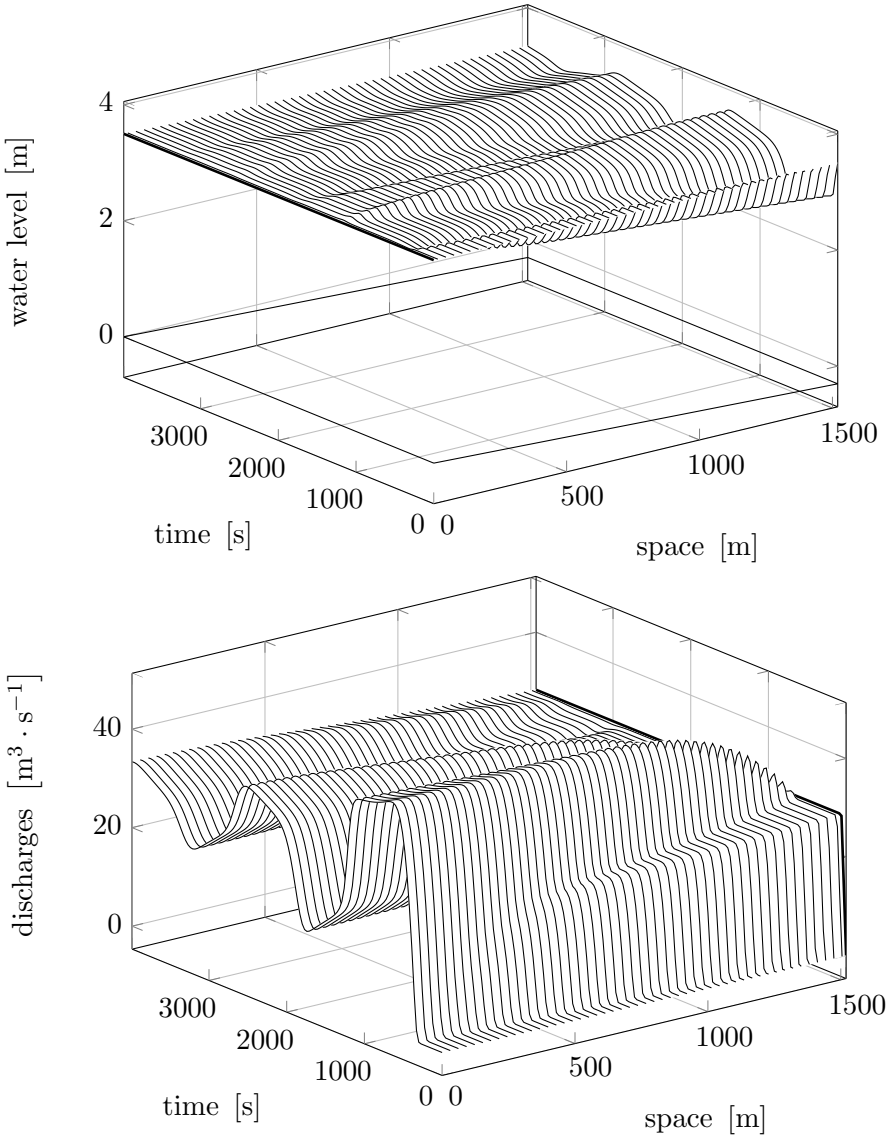


Figure B.4: Evolution of the water levels (top) and discharges (bottom) calculated with the numerical simulator for the situation where the upstream water level is kept constant (bold line) and the downstream discharges increases linearly from 0 to 28.3 $m^3 \cdot s^{-1}$ in 60 seconds and remains constant afterwards (bold line).

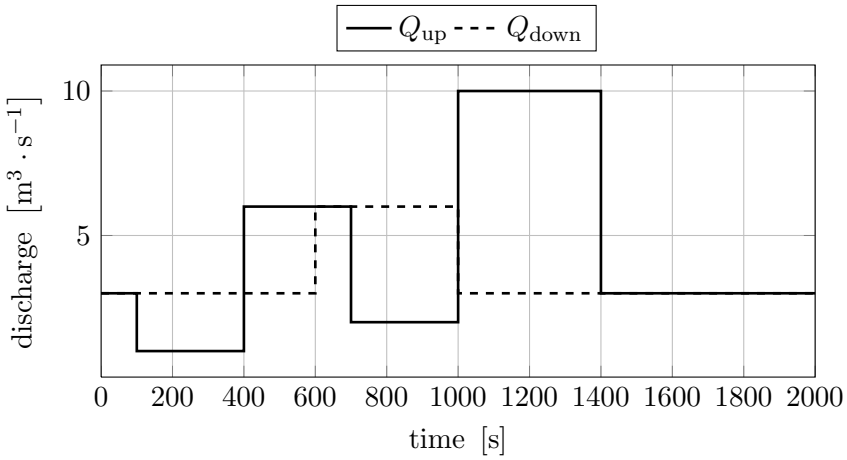


Figure B.5: Time-varying upstream $Q_{\text{up}}(t)$ and downstream $Q_{\text{out}}(t)$ discharges used as boundary conditions.

Table B.1: Parameters of the channels of a simple river system.

	channel 1	channel 2
L	200	100
B	1	1
S_1, S_2	1.5	1.5
n_{mann}	0.012	0.014
S_0	0.0005	0.002

that there is a drop of 0.3 m in the bedding of the second reach just after the gate.

Initially the gate position is equal to 0.5 m, the discharge over the gate Q_{gate} is 10 $\text{m}^3 \cdot \text{s}^{-1}$ and the water level downstream of the gate is 2.3 m. This is enough information to calculate the steady state of both channels. The boundary conditions of the river system are a given time-varying inflowing discharge Q_{in} for channel 1 and a given constant outgoing discharge Q_{out} for channel 2. Also the gate position is changing in time. The top plot of Fig. B.7 shows the time-varying gate position together with the evolution of the water level upstream and downstream of the gate. The bottom plot shows the inflowing discharge Q_{in} and the outgoing discharge of Q_{out} together with the discharge over the gate Q_{gate}. Fig. B.8 visualizes the evolution of all the water levels of both reaches together with the gate position (top plot) and the evolution

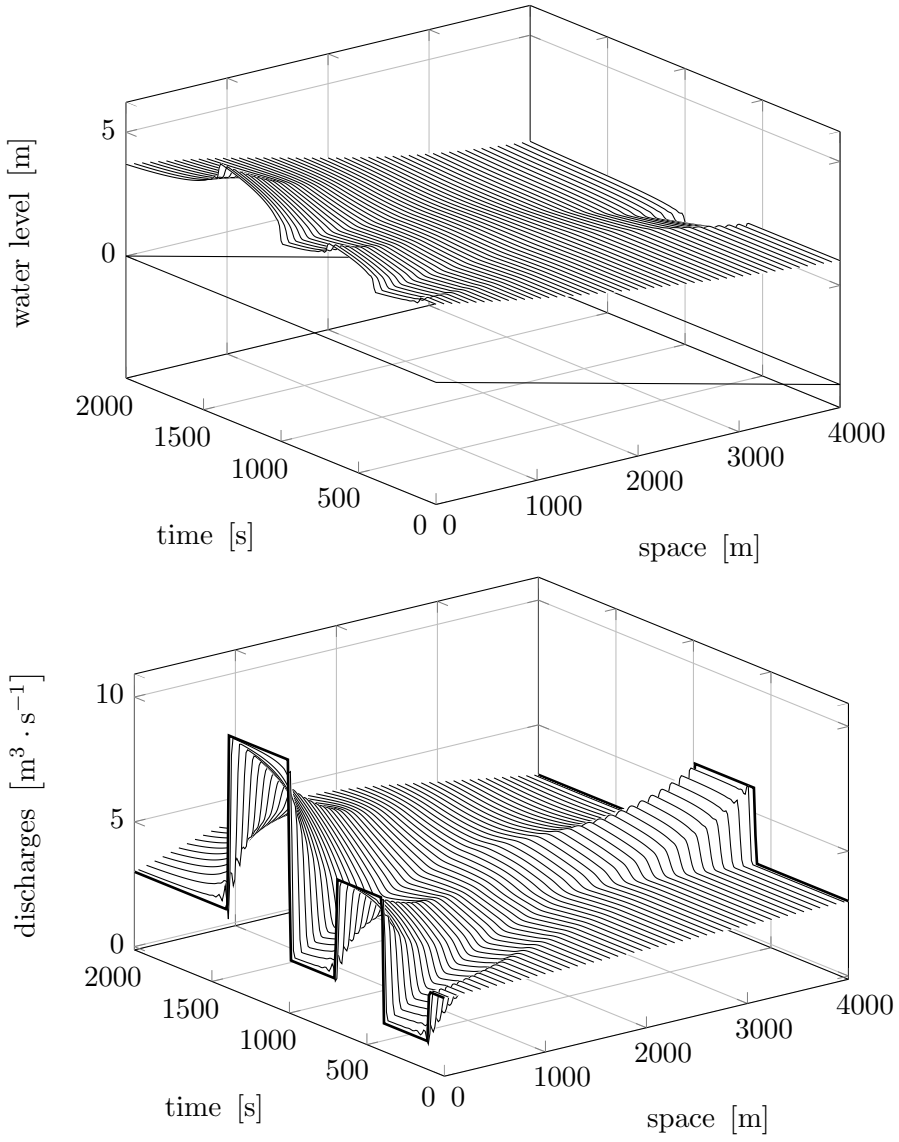


Figure B.6: Evolution of the water levels (top) and discharges (bottom) calculated with the numerical simulator for the situation where the upstream and downstream discharges are step functions in time (bold lines).

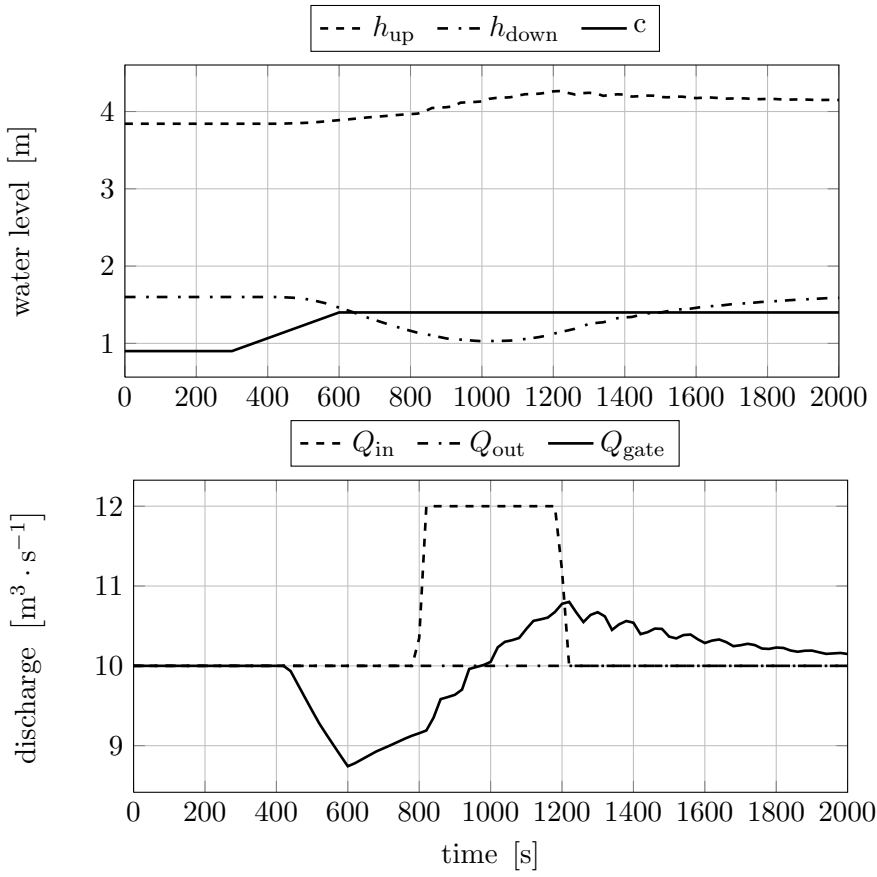


Figure B.7: Evolution of the water levels upstream h_{up} and downstream h_{down} of the gate together with its gate position c (top plot), and the evolution of the time-varying upstream Q_{up} and downstream Q_{out} discharges used as boundary conditions together with the gate discharge Q_{gate} (bottom plot). Because the gate is initially in throat control mode, there is only a change in the gate discharge after 400 seconds while the gate position is already increasing after 300 seconds. The gate discharge decreases, which has as effect that the water levels upstream of the gate will increase (less water is getting out) and the water levels downstream of the gate will decrease (less water is coming in). Once the gate position is kept constant the increased upstream water levels cause the gate discharge to increase. This effect is enforced by increasing the inflowing discharge Q_{in}. Once the inflowing discharge is set to its original value, the discharge over the gate starts converging to its initial value. The ripples in Q_{gate} are caused by waves reflected by the upstream and downstream boundary conditions.

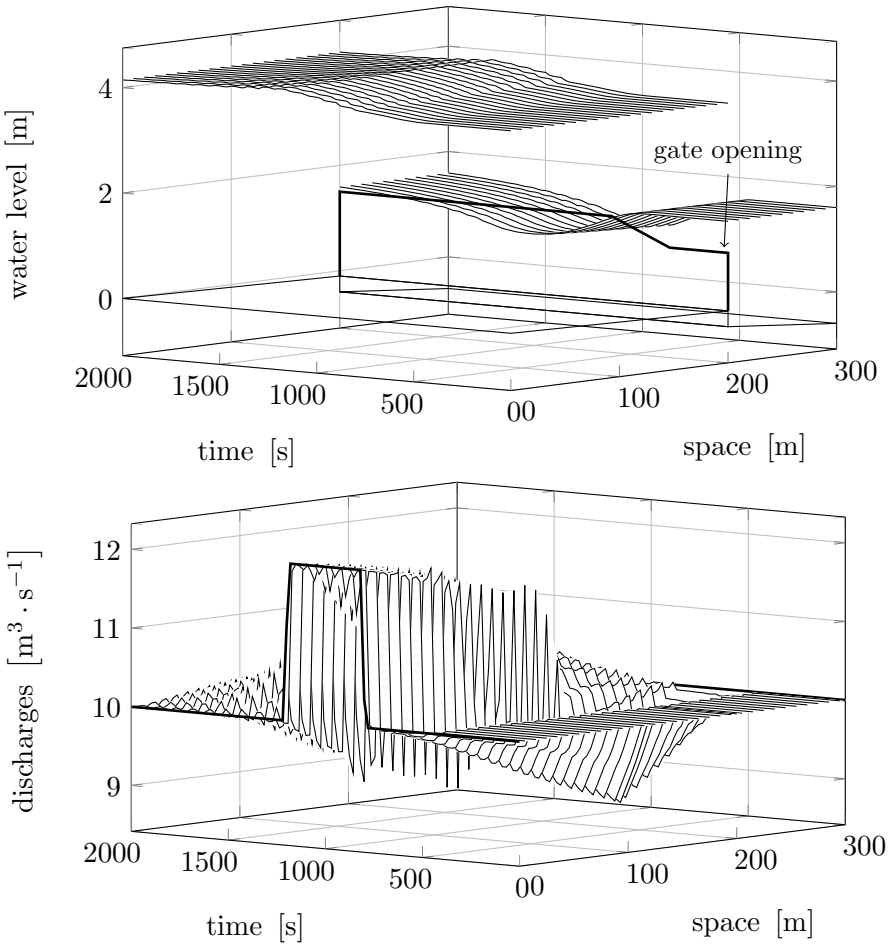


Figure B.8: Evolution of the water levels of both channels and the gate position (top plot) and the discharges of both channels (bottom plot) in space and time. Because the gate is initially in throat control mode, there is only a change in the gate discharge after 400 seconds while the gate position is already increasing after 300 seconds. Increasing the gate position results in an increase of the water levels upstream of the gate and a decrease of the water levels downstream of the gate. Once the gate position is kept constant the increased upstream water levels cause the gate discharge to increase. This effect is enforced by increasing the inflowing discharge Q_{in}. When the inflowing discharge is set to its original value, the system converges to a new steady state.

of the discharges (bottom plot). Although the gate position changes after 300 seconds, there is only a change in the gate discharge (and hence the water levels of both channels) after more than 400 seconds. This means that the gate was initially in throat control mode. This mode will be called uncontrollable in this dissertation: although the gate position is changed, there is no effect on the gate discharge. Only when the gate position is increased enough, the gated weir comes in gate control mode and the discharge over the gate can be reduced. This has as effect that the water levels upstream of the gate increase (less water is getting out) and the water levels downstream of the gate decrease (less water is coming in). This increase of the upstream water levels lead eventually to an increment in the gate discharge once the gate position is not changing any more (after 600 seconds). This effect is amplified by increasing the inflowing discharge Q_{in} after 800 seconds. After 1200 seconds the inflowing discharge is set to its original value and the discharge over the gate starts converging to its initial value. As can be seen in Fig. B.7 and B.8 there is a ripple present in the gate discharge as well as in the discharges and the water levels of the two reaches after 900 s. These ripples are not caused by the numerical solver. The boundary conditions keep the upstream and downstream discharge constant for long time intervals resulting in reflections at the upstream and downstream boundaries causing the ripples in the water levels and discharges of the two reaches. Because these water levels determine also the discharge over the gate, this ripple is also present in the gate discharge.

Bibliography

[1] België: Overstromingen: hoe efficiënt is ons waterbeleid?, July 2011.

[2] Agudelo, O. M. *The application of proper orthogonal decomposition to the control of tubular reactors*. PhD thesis, KU Leuven, Belgium, November 2009.

[3] Agudelo, O. M., Baes, M., Espinosa, J. J., Diehl, M., and De Moor, B. Positive polynomial constraints for pod-based model predictive controllers. *IEEE Transactions on Automatic Control 54*, 5 (May 2009), 988–999.

[4] Albuquerque, F. G., and Labadie, J. W. Optimal nonlinear predictive control for canal operations. *Journal of Irrigation and Drainage Engineering 123*, 1 (January/February 1997), 45–54.

[5] Althans, M. The role and use of the stochastic linear-quadratic-gaussian problem in control system design. *IEEE Transactions on Automatic Control AC-16*, 6 (December 1971), 529–552.

[6] Alvarado, I., Limon, D., Muñoz de la Peña, D., Maestre, J. M., Ridao, M. A., Scheu, H., Marquardt, W., Negenborn, R. R., De Schutter, B., Valencia, F., and Espinosa, J. A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark. *Journal of Process Control 21*, 5 (June 2011), 800–815. Special Issue on Hierarchical and Distributed Model Predictive Control.

[7] Andersen, E. D., Roos, C., and Terlaky, T. On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming 95* (2003), 249–277.

[8] Antoulas, A. C. *Approximation of Large-Scale Dynamical Systems - Advances in Design and Control*. SIAM, Philadelphia, PA, USA, 2005.

[9] Arnold, W.F., I., and Laub, A. Generalized eigenproblem algorithms and software for algebraic riccati equations. *Proceedings of the IEEE 72*, 12 (December 1984), 1746–1754.

[10] Astrid, P. *Reduction of process simulation models: a proper orthogonal decomposition approach*. PhD thesis, Technische Universiteit Eindhoven (Netherlands), 2004.

[11] Astrid, P., Weiland, S., Willcox, K., and Backs, T. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Transactions on Automatic Control 53*, 10 (November 2008), 2237–2251.

[12] Baker, J. A. *Finite-element Computational Fluid Dynamics*. McGraw-Hill, New York, NY, 1983.

[13] Balogun, O. S., Hubbard, M., and DeVries, J. J. Automatic control of canal flow using linear quadratic regulator theory. *Journal of Hydraulic Engineering 114*, 1 (January 1988), 75–102.

[14] Bang, H., and Choong-Seok, O. Predictive control for the attitude maneuver of a flexible spacecraft. *Aerospace Science and Technology 8*, 5 (2004), 443–452.

[15] Banga, J. R., Balsa-Canto, E., Moles, C. G., and Alonso, A. A. Improving food processing using modern optimization methods. *Trends in Food Science & Technology 14*, 4 (2003), 131–144.

[16] Bangia, A. K., Batcho, P. F., Kevrekidis, I. G., and Karniadakis, G. E. Unsteady two-dimensional flows in complex geometries: Comparative bifurcation studies with global eigenfunction expansions. *SIAM Journal on Scientific Computing 18*, 3 (1997), 775–805.

[17] Banks, H. T., del Rosario, R. C. H., and Smith, R. C. Reduced-order model feedback control design: Numerical implementation in a thin shell model. *IEEE Transactions on Automatic Control 45*, 7 (July 2000), 1312–1324.

[18] Barjas Blanco, T. *The river Demer controlled by MPC*. PhD thesis, Faculty of Engineering, KU Leuven, Leuven, Belgium, September 2010.

[19] Barjas Blanco, T., Willems, P., Chiang, P. P.-K., Haverbeke, N., Berlamont, J., and De Moor, B. Flood regulation using nonlinear model predictive control. *Control Engineering Practice 18*, 10 (2010), 1147–1157.

[20] Barjas Blanco, T., Willems, P., Po-Kuan Chiang, P., Cauwenberghs, K., and De Moor, B. *Intelligent Infrastructures*. Springer, 2009, ch. Flood regulation by means of Model Predictive Control, pp. 407–436.

[21] Baume, J.-P., Sau, J., and Malaterre, P.-O. Modelling of irrigation channel dynamics for controller design. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics* (1998), vol. 4, pp. 3856–3861.

[22] Bautista, A. M., Strelkoff, T. S., and Clemmens, A. J. General characteristics of solutions to the open-channel flow, feedforward control problem. *Journal of Irrigation and Drainage Engineering 129*, 2 (2003), 129–137.

[23] Bautista, E., and Clemmens, A. J. Response of ASCE task committee test cases to open-loop control measures. *Journal of Irrigation and Drainage Engineering 125*, 4 (July/August 1999), 179–188.

[24] Bautista, E., Clemmens, A. J., and Strand, R. J. Salt river project canal automation pilot project: Simulation tests. *Journal of Irrigation and Drainage Engineering 132*, 2 (2006), 143–152.

[25] Beltrami, E. Sulle funzioni bilineari. *Giornale di Matematich ad Uso degli Studenti Delle Universita 11* (1873), 98–106.

[26] Berkooz, G. *Studies in Turbulence*. Springer-Verlag, New York, 1991, ch. Observations on the proper orthogonal decomposition, pp. 229–247.

[27] Berlamont, J. Overstromingen: waarom, waar, wanneer, hoe? In *Symposium ruimte voor water, de beste verzekering tegen wateroverlast* (Brussels, May 2001), AMINAL, pp. 1–4.

[28] Beyer, P., Benkadda, S., and Garbet, X. Proper orthogonal decomposition and Galerkin projection for a three-dimensional plasma dynamical systems. *Physical Review E 61*, 1 (January 2000), 813–823.

[29] Bock, H., Diehl, M., Kostina, E., and Schlöder, J. Constrained Optimal Feedback Control of Systems Governed by Large Differential Algebraic Equations. In *Real-Time and Online PDE-Constrained Optimization*, L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, Eds. SIAM, 2007, pp. 3–22.

[30] Boussinesq, V. J. Essai sur la théorie des eaux courantes. *Mémoires présentés par divers savants à l'Académie des Sciences, Paris 23* (1877).

[31] Boyd, S., and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

[32] Breckpot, M., Agudelo, O. M., and De Moor, B. Control of a single reach with model predictive control. In *River Flow 2012* (Costa Rica, September 2012), R. E. Murillo Muñoz, Ed., Proc. of the International Conference on Fluvial Hydraulics, CRC Press, Taylor & Francis Group, pp. 1021–1028.

[33] Breckpot, M., Agudelo, O. M., and De Moor, B. Model predictive control of a river system with two reaches. In *51st IEEE Conference on Decision and Control* (Maui, Hawai, USA, December 2012), pp. 4549–4554.

[34] Breckpot, M., Agudelo, O. M., and De Moor, B. Modelling of a river system with multiple reaches. In *16th IFAC Symposium on System Identification* (Brussels, Belgium, July 2012), pp. 1454–1459.

[35] Breckpot, M., Agudelo, O. M., and De Moor, B. Flood control of the Demer by using Model Predictive Control. Internal Report 13-24, KU Leuven, March 2013.

[36] Breckpot, M., Agudelo, O. M., and De Moor, B. Flood control with model predictive control for river systems with water reservoirs. *Journal of Irrigation and Drainage Engineering 139*, 7 (July 2013), 532–541.

[37] Breckpot, M., Barjas Blanco, T., and De Moor, B. Flood control of rivers with model predictive control. In *2010 American Control Conference* (Baltimore, USA, 2010), pp. 2983–2988.

[38] Breckpot, M., Barjas Blanco, T., and De Moor, B. Flood control of rivers with nonlinear model predictive control and moving horizon estimation. In *49th IEEE Conference on Decision and Control* (Atlanta, USA, 2010), pp. 6107–6112.

[39] Brunner, G. W. *HEC-RAS, River Analysis System Hydraulic Reference Manual*. US Army Corps of Engineers, Hydrologic Engineering Center, November 2002.

[40] Canuto, C., and Hussaini, M. Y. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, New York, NY, 1988.

[41] Chanson, H. *The hydraulics of Open Channel Flow*, second edition ed. Elsevier Butterworth-Heinemann, 2004.

[42] Chatterjee, A. An introduction to the proper orthogonal decomposition. *Current Science 78*, 7 (April 2000), 808–817.

[43] Chaudry, M. H. *Open-Channel Flow*, 2nd ed. Springer, New York, 2008.

[44] Chiang, P.-K., and Willems, P. A conceptual river model to support real-time flood control (demer river, belgium). In *River Flow 2010* (Bundesanstalt für Wasserbau, Karlsruhe, Germany, 2010), A. Dittrich, K. Koll, J. Aberle, and P. Geisenhainer, Eds., vol. 2, pp. 1407–1414.

[45] Chow, V. T. *Open-Channel Hydraulics*. McGraw-Hill Book Co., Inc., New York, N.Y., 1959.

[46] Chow, V. T., Maidment, D. R., and Mays, L. W. *Applied Hydrology*. McGraw-Hill series in water resources and environmental engineering. McGraw-Hill, 1988.

[47] Clemmens, A. J., Bautista, E., Wahlin, B. T., and Strand, R. J. Simulation of automatic canal control systems. *Journal of Irrigation and Drainage Engineering 131*, 4 (August 2005), 324–335.

[48] Clemmens, A. J., and Schuurmans, J. Simple optimal downstream feedback canal controllers: Theory. *Journal of Irrigation and Drainage Engineering 130*, 1 (February 2004), 26–34.

[49] Clemmens, A. J., and Wahlin, B. T. Simple optimal downstream feedback canal controllers: ASCE test case results. *Journal of Irrigation and Drainage Engineering 130*, 1 (February 2004), 35–46.

[50] Contractor, D. N., and Schuurmans, W. Informed use and potential pitfalls of canal models. *Journal of Irrigation and Drainage Engineering 119*, 4 (July/August 1993), 663–671.

[51] Crossley, A. J. *Accurate and efficient numerical solutions for the Saint Venant equations of open channel flow*. PhD thesis, University of Nottingham, October 1999.

[52] Cunge, J. A., Holly, F. M., and Verwey, A. *Practical Aspects of Computational River Hydraulics*. Pitman, London, 1980.

[53] Cusumano, J. P., Sharkady, M. T., and Kimble, B. W. Experimental measurements of dimensionality and spatial coherence in the dynamics of a flexible-beam impact oscillator. *Philosophical Transactions: Physical Sciences and Engineering 347* (May 1994), 421–438.

[54] Cutler, C. R., and Ramaker, B. L. Dynamic matrix control - a computer control algorithm. *AIChE National Meeting* (1979).

[55] Diehl, M., Bock, H., Schlöder, J., Findeisen, R., Nagy, Z., and Allgöwer, F. Real-time optimization and Nonlinear Model Predictive Control of Processes governed by differential-algebraic equations. *Journal of Process Control 12*, 4 (2002), 577–585.

[56] Doan, M. D., Keviczky, T., and De Schutter, B. A distributed optimization-based approach for hierarchical MPC of large-scale systems with coupled dynamics and constraints. In *Proceedings of the 2011 50th IEEE Conference on Decision and Control and European Control*

Conference (CDC-ECC) (Orlando, Florida, December 2011), pp. 5236–5241.

[57] Doan, M. D., Keviczky, T., and De Schutter, B. Application of distributed and hierarchical model predictive control in hydro power valleys. In *Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference* (Noordwijkerhout, The Netherlands, August 2012), pp. 375–383.

[58] Dormand, J. R., and Prince, P. J. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics 6* (1980), 19–26.

[59] Dulhoste, J.-F., Georges, D., and Besancon, G. Nonlinear control of open-channel water flow based on collocation control model. *Journal of Hydraulic Engineering 130*, 3 (March 2004), 254–266.

[60] Einstein, H. A. Der hydraulische oder profil-radius (the hydraulic or cross section radius). *Schweizerische Bauzeitung 103*, 8 (February 1934), 89–91.

[61] Einstein, H. A., and Banks, R. B. Fluid resistance of composite roughness. *Transactions, American Geophysical Union 34*, 4 (August 1950), 603–610.

[62] Eurén, K., and Weyer, E. System identification of open water channels with undershot and overshot gates. *Control Engineering Practice 15* (2007), 813–824.

[63] Fahl, M., and Sachs, E. Reduced order modelling approaches to pde-constrained optimization based on proper orthogonal decomposition. In *Large-Scale PDE-Constrained Optimization*, L. T. Biegler, M. Heinkenschloss, O. Ghattas, and B. Bloemen Waanders, Eds., vol. 30 of *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg, 2003, pp. 268–280.

[64] Feeney, B. F., and Kappagantu, R. On the physical interpretation of proper orthogonal modes in vibrations. *Journal of Sound and Vibration 211*, 4 (1998), 607–616.

[65] Franklin, G. F., Powell, J. D., and Workman, M. L. *Digital Control of Dynamic Systems*, 3rd ed. Addison-Wesley world student series. Addison-Wesley, Boston, MA, 1997.

[66] Garcíía, C. E., Prett, D. M., and Morari, M. Model predictive control: Theory and practice - a survey. *Automatica 25*, 3 (1989), 335–348.

[67] Glaister, P. Difference schemes for the shallow water equations. Numerical Analysis Report 9/97, University of Reading, 1987.

[68] Ha, D. M., Tkalich, P., and Chan, E. S. Tsunami forecasting using proper orthogonal decomposition method. *Journal of Geophysical Research-Oceans 113*, C6 (2008), C06019.

[69] Henderson, F. M. *Open Channel Flow*. Macmillan, 1966.

[70] Henry, H. R. Discussion of "Diffusion of Submerged Jets," by M.L. Albertson, Y.B. Dai, R.A. Jensen, and H. Rouse. *Transactions of the American Society of Civil Engineers 115* (1950), 687–994.

[71] Hicks, F. E., and Steffler, P. M. Comparison of finite element methods for the St. Venant equations. *International Journal for Numerical Methods in Fluids 20*, 2 (January 1995), 99–113.

[72] Hirsch, C. *Numerical Computation of Internal and External Flows, Vol. 1*. Wiley, 1988.

[73] Horton, R. E. Separate roughness coefficients for channel bottom and sides. *Engineering News Record 111*, 22 (1933), 652–653.

[74] Hovd, M., and Braatz, R. D. Handling state and output constraints in MPC using time-dependent weights. In *Proceedings of the American Control Conference (ACC)* (Arlington, USA, 2001), vol. 3, pp. 2418–2423.

[75] Hovd, M., and Braatz, R. D. On the use of soft constraints in MPC controllers for plants with inverse response. In *Proceedings of the 6th IFAC symposium on Dynamics and control of process systems* (Jejudo Island, Korea, 2001), pp. 295–300.

[76] IBM. *IBM ILOG CPLEX V12.1 User's manual for CPLEX*, 1998.

[77] Informatiecentrum, H. De digitale Demer. Een nieuw en krachtig instrument voor waterpeilbeheer, December 2003. Dutch.

[78] Innovyze. *InfoWorks-RS Help Documentation [Version 12.0]*. Innovyze, Oxfordshire, UK, 2011.

[79] Iollo, A., Lanteri, S., and Désidéri. Stability properties of POD-Galerkin approximations for the compressible Navier-Stokes equations. *Theoretical and Computational Fluid Dynamics 13*, 6 (March 2000), 377–396.

[80] Islam, A., Raghuwanshi, N. S., and Singh, R. Development and application of hydraulic simulation model for irrigation canal network. *Journal of Irrigation and Drainage Engineering 134*, 1 (February 2008), 49–59.

[81] Jollife, I. T. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[82] Jordan, C. Mémoire sur les formes bilinéaires. *Journal de Mathématiques Pures et Appliquées 19* (1874), 35–54.

[83] Jordan, C. Sur la réduction des formes bilinéaires. *Comptes Rendus de l'Académie Sciences, Paris 78* (1874), 614–617.

[84] Kalman, R. E. A new approach to linear filtering and prediction problems. *Transactions ASME, Series D, Journal of basic engineering 82*, 1 (March 1960), 34–45.

[85] Karunanithi, N., Grenney, W., Whitley, D., and Bovee, K. Neural networks for river flow prediction. *Journal of Computing in Civil Engineering 8*, 2 (1994), 201–220.

[86] Katopodes, N. A Dissipative Galerkin Scheme for Open-Channel Flow. *Journal of Hydraulic Engineering 110* (April 1984), 450–466.

[87] Kepler, G. M., Tran, H. T., and Banks, H. T. Reduced order model compensator control of species transport in a cvd reactor. *Optimal Control Application & Methods 21*, 4 (2000), 143–160.

[88] Kirby, M., Boris, J. P., and Sirovich, L. A proper orthogonal decomposition of a simulated supersonic shear layer. *International Journal for Numerical Methods in Fluids 10* (1990), 411–428.

[89] Kirby, M., and Sirovich, L. Application of the Karhunen-Loéve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 21*, 1 (1990), 103–108.

[90] Kosambi, D. D. Statistics in function space. *Journal of the Indian Mathematical Society 7* (1943), 76–88.

[91] Krishnamurthy, M., and Christensen, B. A. Equivalent roughness for shallow channels. *Journal of the Hydraulics Division 98*, 12 (1972), 2257–2263.

[92] Kunisch, K., and Volkwein, S. Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition. *Journal of Optimization Theory and Applications 102*, 2 (August 1999), 345–371.

[93] Kwakernaak, H., and Sivan, R. *Linear Optimal Control Systems*. Wiley-Interscience, New York, N.Y., 1972.

[94] Lee, J. H., and Cooley, B. Recent advances in model predictive control and other related areas. In *Proceedings of the Fifth international conference on chemical process control aICHE and CACHE* (1997), J. C. Kantor, C. E. Garcia, and B. Carnahan, Eds., pp. 201–216.

[95] Li, Y., and De Schutter, B. Stability and performance analysis of an irrigation channel with distributed control. *Control Engineering Practice 19*, 10 (October 2011), 1147–1156.

[96] Liang, Y. C., Lee, H. P., Lim, S. P., Lin, W. Z., Lee, K. H., and Wu, C. G. Proper orthogonal decomposition and its applications - part I: Theory. *Journal of Sound and Vibration 252*, 3 (May 2002), 527–544.

[97] Liggett, J. A. The boundary element method - some fluid applications. In *Multi-Dimensional Fluid Transients* (New York, NY, December 1984), M. H. Chaudry and C. S. Martin, Eds., pp. 1–8.

[98] Liggett, J. A., and Cunge, J. A. *Numerical methods of solution of the unsteady flow equations, Unsteady Flow in Open Channels*. Water Resources Publications, Fort Collins, 1975, ch. 4.

[99] Liggett, J. A., and Woolhiser, D. A. Difference solutions of shallow-water equations. *Journal of the Engineering Mechanics Division 93*, EM2 (1975), 39–71.

[100] Lin, C. H., Yen, J. F., and Tsai, C. T. Influence of sluice gate contraction coefficient on distinguishing condition. *Journal of Irrigation and Drainage Engineering 128*, 4 (August 2002), 249–252.

[101] Lin, W. Z., Zhang, Y. J., and Li, E. P. Proper orthognal decomposition in the generation of reduced order models for interconnects. *IEEE Transactions on Advanced Packing 31*, 3 (2008), 627–636.

[102] Litrico, X., and Fromion, V. Simplified modeling of irrigation canals for controller design. *Journal of Irrigation and Drainage Engineering 130*, 5 (2004), 373–383.

[103] Litrico, X., and Fromion, V. Tuning of robust distant downstream PI controllers for an irrigation canal pool, i: Theory. *Journal of Irrigation and Drainage Engineering 132*, 4 (2006), 359–368.

[104] Litrico, X., and Fromion, V. *Modeling and Control of Hydrosystems*. Springer-Verlag, 2009.

[105] Litrico, X., Fromion, V., and Baume, J. P. Tuning of robust distant downstream PI controllers for an irrigation canal pool - II. implementation issues. *Journal of Irrigation and Drainage Engineering 132*, 4 (2006), 3639–379.

[106] Litrico, X., Malaterre, P. O., Baume, J. P., Vion, P.-Y., and Ribot-Bruno, J. Automatic tuning of PI controllers for an irrigation canal pool. *Journal of Irrigation and Drainage Engineering 133*, 1 (2007), 27–37.

[107] Ljung, L. *System Identification: Theory for the User*, 2nd edition ed. Prentice Hall PTR, 1998.

[108] Loève, M. *Probability Theory*, second edition ed. D. Van Nostrand Company, Inc., New Jersy, 1960.

[109] Lotter, G. K. Considerations on hydraulic design of channels with different roughness of walls. *Transactions, All-Union Scientific Research Institute of Hydraulic Engineering 9* (1933), 238–241.

[110] Luenberger, D. G. An introduction to observers. *IEEE Transactions on Automatic Control 16*, 6 (December 1971), 596–602.

[111] Lumley, J. L. *Stochastic Tools in Turbulence*. Academic Press, New York, 1970.

[112] Lumley, J. L., and Blossey, P. Control of turbulence. *Annual Review of Fluid Mechanics 30* (1998), 311–327.

[113] MacDonald, I. *Analysis and Computation of Steady Open Channel Flow*. PhD thesis, Department of Mathematics, University of Reading, UK, September 1996.

[114] MacDonald, I., Baines, M. J., Nichols, N. K., and Samuels, P. G. Steady open channel test problems with analytic solutions. Tech. Rep. Numerical Analysis Report 3/95, Department of Mathematics, University of Reading, 1995.

[115] Malaterre, P. O. *Modelisation, Analysis and LQR Optimal Control of an Irrigation Canal*. PhD thesis, LAAS-CNRS-ENGREF-Cemagref, 1994.

[116] Malaterre, P. O. Pilote: Linear quadratic optimal controller for irrigation canals. *Journal of Irrigation and Drainage Engineering 124*, 4 (July/August 1998), 187–194.

[117] Malaterre, P. O., and Baume, J. P. Modeling and regulation of irrigation canals: existing applications and ongoing researches. In *Proc. IEEE International Conference on Systems, Man and Cybernetics* (1998), vol. 4, pp. 3850–3855.

[118] Malaterre, P. O., and Rodellar, J. Design and comparison of multivariable optimal and predictive control on a 2-pool model. In *International workshop on regulation of irrigation canals.* (1997).

[119] Malaterre, P. O., Rogers, D. C., and Schuurmans, J. Classification of canal control algorithms. *Journal of Irrigation and Drainage Engineering 124*, 4 (January/February 1998), 3–10.

[120] Marinaki, M., and Papageorgiou, M. *Optimal Real-Time Control of Sewer Networks*. Springer, 2005.

[121] Massau, J. Appendice au mémoire sur l'intégration graphique. *Annales de l'Association des Ingénieurs sortis des Ecoles Spéciales de Grand 12* (1889), 135–444.

[122] Massau, J. Mémoire sur l'Intégration Graphique des Equations aux Dérivées Partielles. *Annales de l'Association des Ingénieurs sortis des Ecoles Spéciales de Grand 23* (1900), 95–214.

[123] The MathWorks, Inc., USA. *Optimization Toolbox 3: user's guide*, 2006.

[124] Mayne, D. Q. Nonlinear model predictive control: An assessment. In *Proceedings of the Fifth International Conference on Chemical Process Control AIChE and CACHE* (1997), J. C. Kantor, C. E. Garcia, and B. Carnahan, Eds., pp. 217–231.

[125] Mayne, Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M. Constrained model predictive control: Stability and optimality. *Automatica 36*, 6 (June 2000), 789–814.

[126] Motayed, A. K., and Krishnamurthy, M. Composite roughness of natural channels. *Journal of the Hydraulics Division 106*, 6 (1980), 1111–1116.

[127] Muhlhofer, L. Rauhigkeitsuntersuchungen in einem Stollen mit betonierter Sohle und unverkleideten Wanden. *Wasserkraft und Wasserwirtschaft 28*, 8 (1933), 85–88.

[128] Muske, K. R., and Rawlings, J. B. Model predictive control with linear models. *AIChE Journal 39*, 2 (1993), 262–287.

[129] Negenborn, R. R., van Overloop, P.-J., Keviczky, T., and De Schutter, B. Distributed model predictive control of irrigation canals. *Networks and Heterogeneous Media, 4*, 2 (June 2009), 359–380.

[130] Nocedal, J., and Wright, S. J. *Numerical Optimization*. Springer, New York, USA, Aug. 2000.

[131] Ooi, S. K., and Weyer, E. Control design for an irrigation channel from physical data. *Control Engineering Practice 16* (2008), 1132–1150.

[132] Pearson, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine 2* (1901), 559–572.

[133] Pluymers, B. *Robust Model Based Predictive Control - An Invariant Set Approach -*. PhD thesis, Faculty of Engineering, KU Leuven, Leuven, Belgium, May 2006.

[134] Preissmann, A. Propagation des intumescences dans les canaux et rivières. In *1er Congrès de l'Association Française de Calcul* (Grenoble, France, 1961), pp. 433–442.

[135] Preissmann, A., and Cunge, J. Calcul du mascaret sur machines électroniques. *La Houille Blanche 5* (1961), 588–596.

[136] Qin, S., and Badgwell, T. A survey of industrial model predictive control technology. *Control Engineering Practice 11*, 7 (2003), 733–764.

[137] Ravindran, S. S. A reduced–order approach for optimal control of fluids using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids 34*, 5 (November 2000), 425–448.

[138] Reddy, J. M. Local optimal control of irrigation canals. *Journal of Irrigation and Drainage Engineering 116*, 5 (September/October 1990), 616–631.

[139] Reddy, J. M., Dia, A., and Oussou, A. Design of control algorithm for operation of irrigation canals. *Journal of Irrigation and Drainage Engineering 118*, 6 (November/December 1992), 852–867.

[140] Richalet, J., Rault, A., Testud, J., and Papon, A. Model predictive heuristic control: Applications to industrial processes. *Automatica 14* (1978), 413–428.

[141] Rivas Perez, R. an Feliu Batlle, V., and Sanchez Rodriguez, L. Robust system identification of an irrigation main canal. *Advances in Water Resources 30* (2007), 1785–1796.

[142] Rogers, D. C., and Goussard, J. Canal control algorithms currently in use. *Journal of Irrigation and Drainage Engineering 124*, 1 (1998), 11–15.

[143] Rossiter, J. *Model-Based Predictive Control*. CRC Press, Boca Raton, Fl, 2003.

[144] Ruiz, V., and Ramirez, J. Predictive control in irrigation canal operation. In *IEEE SMC 98* (1998), vol. 4, pp. 3987–3901.

[145] Savorgnan, C., Romani, C., Kozma, A., and Diehl, M. Multiple shooting for distributed systems with applications in hydro electricity production. *Journal of Process Control 21*, 5 (June 2011), 738–745. Special Issue on Hierarchical and Distributed Model Predictive Control.

[146] Schmidt, E. Zur Theorie der linearen un nichtlinearen Integralgleichungen. I. Teil. Entwicklung willkürlicher Funktionen nach System vorgeschriebener. *Mathematische Annalen 63* (1907), 433–476.

[147] Schuurmans, J. *Control of water levels in open channels*. PhD thesis, Delft University of Technology, The Netherlands, 1997.

[148] Schuurmans, J., Clemmens, A. J., Dijkstra, S., Hof, A., and Brouwer, R. Modelling of irrigation and drainage canals for controller design. *Journal of Irrigation and Drainage Engineering 125*, 3 (1999), 338–344.

[149] Schuurmans, J., Hof, A., Dijkstra, S. J., Bosgra, O. H., and Brouwer, R. Simple water level controller for irrigation and drainage canals. *Journal of Irrigation and Drainage Engineering 125* (1999), 189–195.

[150] Schuurmans, J., Schuurmans, W., Berger, H., Meulenberg, M., and Brouwer, R. Control of water levels in the Meuse rivers. *Journal of Irrigation and Drainage Engineering 123*, 3 (1997), 180–184.

[151] Scokaert, P. O. M., and Rawlings, J. B. Feasibility issues in linear model predictive control. *AIChE Journal 45*, 8 (1999), 1649–1659.

[152] Sepúlveda, C., Gómez, M., and Rodellar, J. Benchmark of discharge calibration methods for submerged sluice gates. *Journal of Irrigation and Drainage Engineering 135*, 5 (October 2009), 676–682.

[153] Siade, A. J., Putti, M., and Yeh, Z. W. G. Snapshot selection for groundwater model reduction using proper orthogonal decomposition. *Water Resources Research 46*, 8 (August 2010), W08539.

[154] Singh, A., Forbes, J. F., Vermeer, P. J., and Woo, S. S. Model-based real-time optimization of automotive gasoline blending operations. *Journal of Process Control 10*, 1 (2000), 43–58.

[155] Sirovich, L. Turbulence and the dynamics of coherent structurs. Part I: coherent structures. Part II: symmetries and transformations. Part III: dynamics and scaling. *Quarterly of Applied Mathematics 45* (1987), 561–590.

[156] Sirovich, L., and Kirby, M. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America 4*, 3 (1987), 519–524.

[157] Stelling, G. S., and Duinmeijer, S. P. A. A staggered conservative scheme for every froude number in rapidly varied shallow water flows. *International Journal of Numerical Methods in Fluids 43*, 12 (2003), 1329–1354.

[158] Stoker, J. J. The formation of breakers and bores. *Communications on Pure and Applied Mathematics, New York University 1* (January 1948), 1.

[159] Stoker, J. J. *Water Waves*. New York: John Wiley & Sons, Inc., 1957.

[160] Strelkoff, T. S., and Falvey, H. T. Numerical methods used to model unsteady canal flow. *Journal of Irrigation and Drainage Engineering 119*, 4 (July/August 1993), 637–655.

[161] Stringam, B. L., and Merkley, G. P. Field application of a fuzzy controller for an irrigation canal in Roosevelt Utah. In *International workshop on the regulation of irrigation canals: state of the art research and applications, RIC97* (1997), pp. 349–354.

[162] Stringam, B. L., and Merkley, G. P. Fuzzy controller simulation for local downstream water level control in canals. In *International workshop on the regulation of irrigation canals: state of the art research and applications, RIC97* (1997), pp. 342–348.

[163] Sturm, T. W. *Open Channel Hydraulics*. McGraw-Hill, 2001.

[164] Swamee, P. K. Sluice-gate discharge equations. *Journal of Irrigation and Drainage Engineering 118*, 1 (January/February 1992), 56–60.

[165] Sylvester, J. J. A new proof that a general quadric may be reduced to its canonical form (that is, a linear function of squares) by means of a real orthogonal substitution. *Messenger of Mathematics 19* (1889), 42–46.

[166] Sylvester, J. J. On the reduction of a bilinear quantic of the nth order to the form of a sum of n products by a double orthogonal substitution. *Messenger of Mathematics 19* (1889), 1–5.

[167] Terrens, I. In welke mate zijn overstromingen af te bakenen en te beheersen? In *Symposium ruimte voor water, de beste verzekering tegen wateroverlast* (Brussels, May 2001), AMINAL, pp. 1–13.

[168] Thai, T. *Numerical Methods for Parameter Estimation and Optimal Control for the Red River Network*. PhD thesis, Universität Heidelberg, Germany, 2005.

[169] Toudeft, A., and Gallinari, P. Neural and adaptive controllers for a non-minimum phase varying time-delay system. *Artificial Intelligence in Engineering 11* (1997), 431–439.

[170] Toudeft, A., and Kosuth, P. an Gallinari, P. A PID neural controller for unstable delayed linear systems. In *ICANN'94 Proc.* (University of Salerno, Italy, 1994).

[171] Tran Dinh, Q. *Sequential convex programming and decomposition approaches for nonlinear optimization.* PhD thesis, Faculty of Engineering, KU Leuven, Leuven, Belgium, November 2012.

[172] Treske, A. Undular bores (favre waves) in open channels - experimental studies. *Journal of Hydraulic Research 32*, 3 (1994), 355–370.

[173] Vaes, G. *The influence of rainfall and model simplification on combined sewer system design.* PhD thesis, KU Leuven, 1999.

[174] Vaes, G., Willems, P., and Berlamont, J. Het gebruik van bakmodellen voor de voorspelling van de invoer in waterloopmodellen ter plaatse van riooloverstorten. *Water 4* (2002). Dutch.

[175] van Overloop, P. J. Drainage control in water management of polders in The Netherlands. *Journal of Irrigation and Drainage Systems 20*, 1 (February 2006), 99–109.

[176] van Overloop, P. J. *Model Predictive Control of Open Water Systems.* PhD thesis, Technische Universiteit Delft, The Netherlands, 2006.

[177] van Overloop, P. J., Clemmens, A. J., Strand, R. J., Wagemaker, R. M. J., and Bautista, E. Real-time implementation of model predictive control on Maricopa-Stanfield Irrigation and Drainage District's WM Canal. *Journal of Irrigation and Drainage Engineering 136*, 11 (November 2010), 747–756.

[178] van Overloop, P.-J., Schuurmans, J., Brouwer, R., and Burt, C. M. Multiple-model optimization of proportional integral controllers on canals. *Journal of Irrigation and Drainage Engineering 131*, 2 (April 2005), 190–196.

[179] van Overloop, P.-J., Weijs, S., and Dijkstra, S. Multiple model predictive control on a drainage canal system. *Control Engineering Practice 16*, 5 (May 2008), 531–540.

[180] Vandevegte, J. *Feedback Control Systems.* Prentice Hall, New Jersey, 1990.

[181] Voron, B., and Bouillot, A.-P. Application of the fuzzy set theory to the control of a large canal. In *International workshop on the regulation of irrigation canals: state of the art research and applications, RIC97* (1997), pp. 317–331.

[182] Wahlin, B. T. Performance of model predictive control on ASCE test canal 1. *Journal of Irrigation and Drainage Engineering 130*, 3 (June 2004), 227–238.

[183] Wahlin, B. T., and Clemmens, A. J. Performance of historic downstream canal control algorithms on ASCE test canal 1. *Journal of Irrigation and Drainage Engineering 128*, 6 (2002), 365–375.

[184] Wahlin, B. T., and Clemmens, A. J. Automatic downstream water-level feedback control of branching canal networks: theory. *Journal of Irrigation and Drainage Engineering 132*, 3 (2006), 208–219.

[185] Wardlaw, R., and Bhaktikul, K. Application of a genetic algorithm for water allocation in an irrigation system. *Irrigation and Drainage 50*, 2 (2001), 159–170.

[186] Weyer, E. System identification of an open water channel. *Control Engineering Practice 9* (2001), 1289–1299.

[187] Weyl, H. Das aymptotische Verteilungsgesetz der eigenwert linearer partieller Differentialgleichungen (mit einer Anwendung auf der Theorie der Hohlraumstrahlung). *Mathematische Annalen 74* (1912), 441–479.

[188] Willems, P. *Probabilistic emission modelling of receiving surface waters*. PhD thesis, KU Leuven, 2000.

[189] Willems, P., Barjas Blanco, T., Chiang, P.-K., Cauwenberghs, K., Berlamont, J., and De Moor, B. Real-time control of urban flooding. In *Water & Urban Development Paradigms: Towards an integration of engineering, design and management approaches* (2008), J. Feyen, K. Shannon, and M. Neville, Eds., CRC Press, Taylor & Francis Group, pp. 265–270.

[190] Winton, C., Pettway, J., Kelley, C., Howington, S., and Eslinger, O. J. Application of proper orthogonal decomposition (pod) to inverse problems in saturated groundwater flow. *Advances in Water Resources 34*, 12 (2011), 1519–1526.

[191] Wylie, E. B., Streeter, V. L., and Suo, L. *Fluid Transients in Systems*. Prentice Hall, 1993.

[192] Xu, M. *Real-time Control of Combined Water Quantity & Quality in Open Channels*. PhD thesis, T.U. Delft, the Netherlands, January 2013.

[193] Xu, M., van Overloop, P.-J., and van de Giesen, N. C. On the study of control effectiveness and computational efficiency of reduced Saint-Venant model in model predictive control of open channel flow. *Advances in Water Resources 34* (2011), 282–290.

[194] Yang, H.-C., and Chang, F.-J. Modelling combined open channel flow by artificial neural networks. *Hydrological Processes 19* (2005), 3747–3762.

[195] Yen, J. F., Lin, C. H., and Tsai, C. T. Hydraulic characteristics and discharge control of sluice gates. *Journal of the Chinese Institute of Engineers 24*, 3 (2001), 301–310.

[196] Ying, X., Khan, A., and Wang, S. Upwind conservative schemes for the Saint Venant equations. *Journal of Hydraulic Engineering 130*, 10 (October 2004), 977–987.

[197] Zafra-Cabeza, A., Maestre, J. M., Ridao, M. A., Camacho, E. F., and Sánchez, L. A hierarchical distributed model predictive control approach to irrigation canals: A risk mitigation perspective. *Journal of Process Control 21*, 5 (June 2011), 787–799. Special Issue on Hierarchical and Distributed Model Predictive Control.

Curriculum Vitae

Maarten Breckpot was born on August 17, 1986 in Antwerp, Belgium. He received the degree of Bachelor of Engineering: Computer Science - Electrical Engineering from the KU Leuven with the grade summa cum laude in July 2007. In July 2009, he received the degree of Master of Engineering: Mathematical Engineering from the KU Leuven with the grade summa cum laude. His master thesis was entitled "Flood control of the Demer with Model Predictive Control". He tested whether the software of the spin-off company IPCOS can be used for the application of flood control. His master thesis was awarded with the Besix award 2009, the EOS award 2009 and the Agoria award 2009, and it was nominated for the Vlaamse Scriptieprijs 2009. In October 2009, he started the PhD program in the SCD-SISTA research division of the Electrical Engineering Department of the KU Leuven under the supervision of Prof. dr. ir. Bart De Moor. Since October 2010 M. Breckpot is a PhD fellow of the Research Foundation - Flanders (FWO). At the end of 2010 he stayed for more than three months for a research stay at the Institute for Pure and Applied Mathematics of the University of California Los Angeles, USA.

Publications by the author

International Journal Papers

- Breckpot, M., Agudelo, O. M., and De Moor, B. Flood control with Model Predictive Control for river systems with water reservoirs. *Journal of Irrigation and Drainage Engineering 139*, 7 (July 2013), 532-541.

- Breckpot, M., Agudelo, O. M., Meert, P., Willems, P., and De Moor, B. Flood control of the Demer by using Model Predictive Control. Internal Report 13-24, ESAT-SISTA, KU Leuven (Leuven, Belgium). *Submitted for publication in Control Engineering Practice* (2013).

International Conference Papers

- Breckpot, M., Barjas Blanco, T., and De Moor B. Flood control of rivers with Model Predictive Control – proof of concept based on the river Demer in Belgium. In *Proc. of the 2010 American Control Conference*, (Baltimore, USA, June 2010), pp. 2983-2988.

- Breckpot, M., Barjas Blanco, T., and De Moor B. Flood control of rivers with model predictive control and moving horizon estimation. In *Proc. of the 49th IEEE Conference on Decision and Control* (Atlanta, Georgia, December 2010), pp. 6107 - 6112.

- Breckpot, M., Agudelo, O. M., and De Moor, B. Modelling of a river system with multiple reaches. In *Proc. of the 16th IFAC Symposium on System Identification* (Brussels, Belgium, July 2012), pp. 1454-1459.

- Breckpot, M., Agudelo, O. M., and De Moor, B. Control of a single reach with model predictive control. In *River Flow 2012* (Costa Rica, September 2012), R. E. Murillo Muñoz, Ed., Proc. of the

International Conference on Fluvial Hydraulics, CRC Press, Taylor & Francis Group, pp. 1021-1028.

- Breckpot, M., Agudelo, O. M., and De Moor, B. Model Predictive Control of a river system with two reaches. In *Proc. of the 51st IEEE Conference on Decision and Control* (Maui, Hawaii, USA, December 2012), pp. 4549-4554.

National Journal Papers

- Breckpot, M., and De Moor, B. Overstromingen te slim af. *EOS*, 2 (February 2010), pp. 34-35.

Published abstracts

- Breckpot, M., and De Moor, B. Flood control of the Demer with model predictive control. *Benelux Meeting on Systems and Control* (Heeze, the Netherlands, March/April 2010).
- Breckpot, M., Agudelo, O. M., and De Moor, B. Comparison of Model Predictive Control and Linear Quadratic Regulator when applied to a single reach. *Benelux Meeting on Systems and Control* (Heijen, The Netherlands, March 2012).

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
SISTA-SCD
Kasteelpark Arenberg 10
B-3001 Heverlee
maarten.breckpot@esat.kuleuven.be
http://www.esat.kuleuven.be/scd

