KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
AFDELING ESAT-SCD: SISTA
Kasteelpark Arenberg 10, B-3001 Leuven

THE RIVER DEMER CONTROLLED BY MPC

Jury:
Prof. dr. ir. A. Haegemans, voorzitter
Prof. dr. ir. B. De Moor, promotor
Prof. dr. ir. J. Berlamont, copromotor
Prof. dr. ir. J. Suykens
Prof. dr. ir. P. Willems
Prof. dr. ir. B. De Schutter (T.U. Delft)
dr. ir. R. Negenborn (T.U. Delft)

Proefschrift voorgedragen
tot het behalen van de
graad van Doctor in de
Ingenieurswetenschappen

door

Toni BARJAS BLANCO

September 2010

©2010 Katholieke Universiteit Leuven, Groep Wetenschap & Technologie, Arenberg Doctoraatsschool, W. de Croylaan 6, B-3001 Heverlee, België

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 978-94-6018-249-5
D/2010/7515/88

Voorwoord

Na mijn studies als industrieel ingenieur elektromechanica besloot ik om de studie van burgerlijk ingenieur in de computerwetenschappen aan te vatten. Tijdens deze studie kwam ik in aanraking met de regeltechniek via het vak *Computer Controlled Systems*, gegeven door Prof. Bart De Moor. Door dit vak raakte ik zo geboeid in de regeltechniek dat ik uiteindelijk besloot om een doctoraat te starten in dit topic. Mijn promotor, Prof. Bart De Moor, bood mij deze kans en ik heb onmiddellijk toegehapt. De jaren daarop heb ik me volledig toegelegd op het afwerken van dit doctoraat. Het waren geweldige jaren waarin ik enorm veel heb bijgeleerd, veel interessante mensen heb leren kennen en het hele noordelijk halfrond heb afgereisd. Maar aan alle mooie liedjes komt een eind en ook hier is dit gezegde van toepassing. Wat mij nog rest is om mijn doctoraat af te sluiten in schoonheid en ik zou dit willen doen door al die mensen te bedanken die mij door deze jaren heen geloodst hebben.

Vooreerst zou ik mijn promotor Prof. Bart De Moor willen bedanken. Tijdens mijn doctoraat heeft hij altijd gezorgd voor de nodige ondersteuning en begeleiding. Daarnaast wist hij altijd te vertellen hoe ik het best moest omgaan met moeilijke situaties en in dat opzicht heb ik enorm veel van hem geleerd. Maar ik zou hem vooral willen bedanken omdat hij de persoon is die mij de kans heeft aangeboden om mijn doctoraat te starten. Bart, bedankt hiervoor. Verder zou ik ook mijn dank willen betuigen aan de mensen van het hydraulisch laboratorium aan de K.U. Leuven, met name Prof. Jean Berlamont, Prof. Patrick Willems en Paul Chiang. Zonder de hydraulische modellen die ze me geleverd hebben, zou dit doctoraat nooit tot stand zijn gekomen. Daarnaast waren ze ook altijd bereid om mij de nodige feedback te geven en hebben ze mij ook meermaals begeleid naar de vele vergaderingen bij het VMM. Ik dank jullie hiervoor.

Ik zou ook de andere leden van mijn begeleidings- en examencommissie, Prof. Ann Haegemans, Prof. Johan Suykens, Prof. Bart De Schutter, Dr. Rudy Negenborn willen bedanken voor het nalezen van het proefschrift en de waardevolle suggesties en aanvullingen. Extra dank zou ik willen betuigen aan Prof. Bart De Schutter en Dr. Rudy Negenborn omdat zij het ervoor over hadden om helemaal vanuit Delft naar Leuven over te komen om aanwezig te zijn op mijn verdediging.

Ik zou ook graag al mijn collega's willen bedanken. Zij hebben steeds voor een aangename werksfeer gezorgd en als ik hulp nodig had dan waren ze altijd bereid om me te helpen. Het was een waar genoegen om met jullie te hebben samengewerkt. Speciale dank zou ik ook willen betuigen aan Ernesto, Niels en Paschalis voor de vele koffiepauzes die we samen hebben gehouden en de vele boeiende gesprekken die daaruit voortvloeiden. Verder zou ik ook nog al mijn vrienden buiten ESAT willen bedanken. Zij hebben er elk op hun manier voor gezorgd dat ik mijn gedachten op tijd en stond kon verzetten en me steeds gesteund als dat nodig was. Ik ben jullie hier oprecht dankbaar voor. Ook ben ik het VMM erkentelijk voor de financiële ondersteuning gedurende mijn doctoraat.

Om af te sluiten, zou ik mijn familie willen bedanken. Zonder mijn familie zou ik niet staan waar ik vandaag sta. Mijn familie stond altijd klaar voor mij en heeft mij altijd gesteund wanneer nodig. Onrechtstreeks zijn ze allemaal verantwoordelijk dat ik vandaag mijn doctoraat beëindig. Ik zou daarom mijn grootouders, nonkels en tantes, neven en nichten, en mijn hele familie in Spanje willen bedanken voor al hetgeen ze voor mij gedaan hebben in de loop van mijn leven. Speciale dank zou ik willen betuigen aan mijn ouders, Jose en Juana, mijn broertjes, Claudio en David en mijn zusje Sylvia. Zij stonden altijd klaar om mij te helpen. Ze schepten er zelfs genoegen uit als ze iets konden doen voor hun zoon/grote broer. In moeilijke momenten waren zij degene die mij vooruit hielpen. Ze hebben alles gegeven wat ze konden om ervoor te zorgen dat ik vandaag ingenieur ben en mijn doctoraat beëindig. Mijn allergrootste dank hiervoor.

Toni Barjas Blanco
September 2010

Abstract

Flooding of rivers is a worldwide problem with severe consequences. This is also the case in Belgium where the Demer river causes floods in its basin during periods of heavy rainfall. In order to reduce these floods, the local water administration increased the storage capacity of the river with flood reservoirs to store the excessive amount of water during periods of heavy rainfall. Also hydraulic structures were added to control the discharges and water levels in the river, but also to control the flow of water from and into the reservoirs. Nowadays, these structures are controlled by an advanced three-position controller that determines the control actions based on logical rules that were derived in a heuristic way. Though these measures have significantly reduced the amount of flood damage in the Demer basin, simulations of historical rainfall events on a full hydrodynamic model (InfoWorks) have shown that the flood damage could have been reduced even more if the hydraulic structures would have been controlled differently. Therefore, the goal of this thesis is to determine a more advanced control strategy that performs better than the currently used three-position controller.

In this thesis it is investigated whether a model predictive controller performs better with respect to flood regulation than the three-position controller. In order to do so, a simplified model is derived that is fast and accurate enough to be used in a model predictive control framework. The simplified model is of the reservoir type and is calibrated and validated based on simulation data obtained from simulating historical rainfall events with the InfoWorks model of the Demer river. Next, a nonlinear model predictive controller is described that uses the simplified model in order to determine the future control actions. In contrast to standard model predictive control schemes for setpoint regulation we have chosen to use the gate levels as control inputs instead of the discharges over the gates because the nonlinear gate dynamics cannot be neglected during flood events. The proposed scheme also tackles problems like local uncontrollability of the gates and constraint infeasibility. A nonlinear moving horizon estimator is also added to the model predictive control scheme. Based on historical measurements this estimator determines the most probable value of the actual state of the system. This estimate is then passed to the model predictive controller that uses this state information for its predictions. The control scheme resulting from the combination of the model predictive controller with the moving horizon estimator is then compared with the three-position controller by comparing their respective performance in simulations based on historical rainfall data. Robustness of the new scheme is also tested by adding a realistic amount of uncertainty to the rainfall

predictions. The thesis ends with a theoretical contribution in the stability of model predictive control. More specific, a new algorithm is described to determine low-complexity polytopic invariant sets. The proposed algorithm is then used for improved setpoint regulation of the upstream part of the Demer river.

Korte Inhoud

Overstromingen van rivieren is een wereldwijd probleem dat vaak gepaard gaat met ernstige overstromingsschade. Dit is ook het geval in België waar de Demer in het verleden regelmatig overstromingen heeft veroorzaakt tijdens periodes van zware neerslag. Met het oog de overstromingen in het Demerbekken te verminderen, heeft de lokale waterbeheerder de opslagcapaciteit van de rivieren verhoogd door middel van wachtbekkens. Tijdens periodes van zware neerslag kan het overtollige water opgeborgen worden in de wachtbekkens. Er werden ook hydraulische structuren geïnstalleerd in het Demerbekken. Deze structuren maken het mogelijk om de debieten en waterniveau's van de rivieren te regelen. Daarnaast laten zij ook toe om de wachtbekkens te ledigen of te vullen. Momenteel worden deze structuren aangestuurd door een geavanceerde drie-standen regelaar die haar regelacties bepaalt op basis van logische regels die op een heuristische wijze tot stand zijn gekomen. Al deze maatregelen hebben er inderdaad voor gezorgd dat het aantal overstromingen in het Demerbekken beduidend verminderd zijn. Simulaties van historische neerslagsgebeurtenissen op een hydrodynamisch model (InfoWorks) hebben echter uitgewezen dat de overstromingsschade kleiner had kunnen zijn indien de hydraulische structuren op een andere manier zouden zijn aangestuurd geweest. Daarom is het doel van deze thesis om een meer geavanceerde regelstrategie te ontwerpen die tot betere resultaten leidt dan de huidige drie-standen regelaar.

In deze thesis wordt onderzocht of een model predictieve regelaar beter presteert naar overstromingsbeheersing toe dan een drie-standen regelaar. Hiervoor wordt in eerste instantie een vereenvoudigd model afgeleid die snel en nauwkeurig genoeg is om gebruikt te kunnen worden in een model predictief kader. Het vereenvoudigd model is van het reservoir type en wordt gecalibreerd op basis van simulatiedata gegenereerd door historische neerslagsgebeurtenissen te simuleren met het InfoWorks model van de Demer. Vervolgens wordt een niet-lineaire model predictieve regelaar ontwikkeld die het vereenvoudigde model gebruikt om de regelacties te bepalen. In tegenstelling tot regelingen naar een referentieniveau spelen de niet-lineairiteiten een zeer belangrijke rol bij overstromingsbeheersing. De niet-lineaire dynamica van de hydraulische structuren kan niet meer verwaarloosd worden en daarom hebben wij ervoor gekozen om de klepstanden van de hydraulische structuren als regelingangen te nemen in plaats van de debieten over de kleppen. Het voorgestelde regelschema pakt ook problemen aan zoals lokale oncontroleerbaarheid van de hydraulische structuren en schending van de beperkingen. Er wordt ook een niet-linear schuivend

tijdsvenster toestandsschatter toegevoegd in het regelschema. Op basis van historische meetgegevens bepaalt de toestandsschatter de meest waarschijnlijke waarde voor de huidige toestand. Deze schatting wordt dan doorgestuurd naar de model predictieve regelaar die deze toestandsinformatie gebruikt voor het maken van zijn voorspellingen. Het uiteindelijke regelschema wordt dan vergeleken met de drie-standen regelaar door hun performantie na te gaan door middel van simulaties gebaseerd op historische neerslagsgegevens. De thesis besluit met een theoretische contributie betreffende de stabiliteit van een model predictieve regelaar. Een nieuw algoritme wordt uiteengezet dat toelaat om polytopische invariante sets te bepalen van lage complexiteit. Dit algoritme wordt vervolgens gebruikt om de regeling in het opwaarste deel van de Demer te verbeteren.

Notation

Variables and Symbols

$\alpha, \beta, \gamma \in \mathbb{R}$	Greek symbols denote scalar variables
$a, b, c \in \mathbb{R}^n$	Lower case roman symbols denote scalar or vector variables
$A, B, C \in \mathbb{R}^{m \times n}$	Upper case roman symbols denote matrix variables
$A(i,j), A \in \mathbb{R}^{m \times n}$	Element at the ith row and j^{th} column of A
$A(i,:), A \in \mathbb{R}^{m \times n}$	i^{th} row of a matrix A
$A(:,j), A \in \mathbb{R}^{m \times n}$	j^{th} column of a matrix A
$A \succ 0$	Positive-definit matrix A
A^T	Transpose of a matrix
x^T	Transpose of a vector
$\|x\|_1, \|x\|_2, \|x\|_\infty, x \in \mathbb{R}^n$	$L_1,\ L_2$ and L_∞-norm of a vector x
$\|a\|$	Absolute value of scalar a
$\sum_{i=1}^{n} a_i, a_i \in \mathbb{R}$	Summation of scalars $a_1, \ldots, a_n$
$\mathbb{R}$	Set of real numbers
$\mathbb{N}$	Set of positive integers
I	Identity matrix
$\mathbf{1}$	Vector where all components are equal to 1

Acronyms

ANN	Artificial Neural Networks
ARMAX	Autoregressive Moving Average Exogenous
ARX	Autoregressive Exogenous
EnKF	Ensemble Kalman Filter
EQP	Equality Quadratic Programming
HIC	Hydrologic Information Center
IW	InfoWorks
IQP	Inequality Quadratic Programming
KMI	Koninklijk Meteorologisch Instituut
LP	Linear Program(ming)
LMI	Linear Matrix Inequality
LTI	Linear Time-Invariant
LTV	Linear Time-Varying
LQR	Linear Quadratic Regulator
MAS	Maximal Admissible Set
MHE	Moving Horizon Estimation
MIMO	Multiple-Input / Multiple-Output
MPC	Model based Predictive Control
NLP	Non-Linear Program(ming)
NMHE	Non-linear Moving Horizon Estimation
NMPC	Non-linear Model Predictive Control
OE	Output Error
PID	Proportional / Integral / Differential
QP	Quadratic Program(ming)
SDP	Semi Definite Program(ming)
SQP	Sequential Quadratic Programming
TAW	Tweede Algemene Waterpassing
UKF	Unscented Kalman Filter
VMM	Vlaamse Milieu Maatschappij

Contents

Voorwoord **i**

Abstract **iii**

Korte Inhoud **v**

Notation **vii**

1 Introduction **1**
1.1 Problem description and thesis goal 1
1.1.1 The river Demer 1
1.1.2 Model of the Demer basin 3
1.1.3 Goal of the thesis 5
1.2 Optimization theory 7
1.2.1 Definitions 7
1.2.2 Convex programming 8
1.2.3 Nonlinear nonconvex programming 10
1.3 Model based predictive control 12
1.3.1 Components of MPC 13
1.3.2 Basic algorithm formulation 13
1.4 Moving horizon estimation 16
1.5 Stability framework 18
1.5.1 Stability conditions 19
1.5.2 MPC stability 20
1.5.3 Set invariance 22
1.6 Chapter by chapter overview 24
1.7 Specific contributions of this thesis 25

2 Hydrodynamic Water System Modeling **29**
2.1 Introduction 29
2.2 Hydraulic concepts 30
2.2.1 Definitions 30
2.2.2 Conservation laws 32
2.3 Modeling techniques for water systems 36
2.3.1 Models based on finite-difference methods 36

2.3.2 System identification . . . 42
2.3.3 Integrator-delay model . . . 42
2.3.4 Reservoir model . . . 44
2.4 Conceptual model . . . 47
2.4.1 Conceptual model structure . . . 47
2.4.2 Conceptual model building and calibration process . . . 47
2.4.3 Modeling of hydraulic structures . . . 55
2.4.4 Conceptual model validation . . . 58
2.5 Conclusions . . . 61

3 Flooding Regulation with Model Based Predictive Control 63
3.1 State of the art . . . 63
3.1.1 Controllers . . . 63
3.1.2 MPC in practice . . . 66
3.2 Nonlinear model predictive control . . . 67
3.2.1 Mathematical formulation . . . 68
3.2.2 Constraint and cost function strategy . . . 76
3.2.3 Uncontrollable modes . . . 78
3.3 Experimental results . . . 80
3.3.1 Performance indicators . . . 81
3.3.2 Experimental results on the ‘reduced area’ around Schulensmeer 81
3.3.3 Experimental results on the large area . . . 90
3.4 Uncertainty . . . 98
3.5 Conclusions . . . 103

4 State Estimation 105
4.1 Introduction . . . 105
4.2 State estimators . . . 106
4.2.1 State estimators for processes with Gaussian noise . . . 107
4.2.2 Estimator choice . . . 110
4.3 State of the art . . . 111
4.4 Nonlinear moving horizon estimation . . . 112
4.5 Experimental results . . . 115
4.5.1 Simulations without uncertainty on the rainfall-runoff predictions 115
4.5.2 Simulations with Gaussian uncertainty on the rainfall-runoff predictions . . . 122
4.6 Conclusions . . . 127

5 Low-Complexity Controlled Invariant Polytopes 133
5.1 Set invariance . . . 133
5.1.1 Definitions . . . 134
5.1.2 Special families of invariant sets . . . 135
5.2 State of the art . . . 138
5.2.1 Low-complexity polytopes for the LTI case . . . 138
5.2.2 Low-complexity polytopes for the LTV case . . . 140
5.3 Efficient computation of low-complexity polytopes . . . 143

5.3.1 Invariance conditions 143
5.3.2 Initialization 144
5.3.3 Volume maximization 146
5.4 Extensions 149
5.4.1 Linear systems with bounded disturbances 149
5.4.2 Low-complexity polytopes based on the L_1-norm 153
5.4.3 Rate constraints 156
5.5 Examples 157
5.6 Robust feedback control for a river system 163
5.7 Conclusions 164

6 Conclusions and Future Research **165**
6.1 Conclusions 165
6.1.1 General 165
6.1.2 Flooding regulation of the river Demer 166
6.1.3 Low-complexity controlled invariant sets 168
6.2 Future research 169
6.2.1 Flooding regulation 169
6.2.2 Low-complexity controlled invariant sets 170

Bibliography **173**

Curriculum Vitae **183**

Publications by the author **185**

Chapter 1

Introduction

The river Demer is a river that during periods of heavy rainfall causes floodings in the cities located in the Demer basin. Hydraulic structures in the river try to minimize the floodings by properly managing the flows in the river. These structures are currently controlled by a three-position control strategy. However, simulations indicate that this strategy is far from optimal w.r.t. flooding regulation and therefore it will be tested whether a model predictive controller can obtain better performance. In this chapter the flooding problem in the Demer is outlined in Section 1.1. Next, the necessary optimization concepts that form the core of the control scheme discussed in this thesis are explained in Section 1.2. In Sections 1.3 and 1.4 a basic explanation is given of the controller and state estimator used in this thesis. Section 1.5 discusses the stability framework and provides the necessary tools for ensuring stability of the model predictive controller. The chapter ends with an overview of the chapters in this thesis and how they are related to each other.

1.1 Problem description and thesis goal

1.1.1 The river Demer

The river Demer is a river located in Flanders, the Dutch speaking part of Belgium. The river Demer originates in Ketsingen at Tongeren and flows into the river Dijle at Werchter. The river Demer has a length of 95 km and has as most important tributaries the Herk, the Gete, the Velpe and the Zwarte Beek. In Figure 1.1 an overview is given of the river Demer and its tributaries. The river Demer is fed by water coming from rainfall and groundwater. Such a river typically has high discharge peaks during periods of heavy rainfall and very low discharges in the summer and periods of drought. Already from the prehistory the forests and swamps in the Demer basin were the reason for people to move to the Demer basin. Small settlements were developed and farming activity increased. Till the 11th century the landscape in the Demer basin

hardly changed. The landscape consisted mainly of forests and swamps with locally a small town or village. It was not until the middle ages that there was a big deforestation in order to make way for large farming activities and the construction of large cities. This turned the river Demer into the enemy of the people living in the cultivated areas because of the floodings that were caused during periods of heavy rainfall.

From the 17th century until today measures have been constantly taken in order to avoid floodings. Many meanders, 87 in total, in the river Demer have been cut off, the river has been widened and deepened and dikes have been constructed in order to protect cities. Besides tackling flooding, these measures also have as goal to improve navigation and increase the drainage in the basin for agricultural purposes. Despite these measures even in the past century many floodings have hit the cities in the Demer basin. Floodings hit the Demer basin in 1905, 1926, 1965, 1966, $1993-1994$, 1995, 1998 and 2002. In table 1.1 an overview is given of the damage caused in the Demer basin by the latest severe floodings. From the table one can see that especially in 1998 there was a very severe flooding causing financial damage of over 16 million euros. This flooding has risen many questions about the benefit of the measures that have been taken in the Demer basin. The adjustments made in the Demer basin have always been considered as favorable measures but after the flooding in 1998 one started to realize that these adjustments could lie at the base of the severe flooding damage itself. Without the adjustments more floodings would have occurred at the upstream part but with probably less damage as there are more cities and villages at the downstream side of the river Demer. Therefore, a process has been started where the adjustments are reconsidered and all planned future modifications of the river have been stopped.

Period	flooded area (km^2)	damage in €
$12/1993$-$01/1994$	23.5	$47\,000$
$01/1995-02/1995$	22.9	$11\,000$
$09/1998$	32.6	$16\,169\,000$
$02/2002$	15.7	unknown
$12/2002-01/2003$	18.0	unknown

Table 1.1: The damage caused in the Demer basin by the most recent severe floodings. (table taken from [9])

The new adopted philosophy is one of damage limitation. The awareness has grown that it is impossible to avoid all floodings and that one has to properly manage the floodings in order to limit the caused damage. A flooding is not a disaster as long as it does not occur on places where it can cause a lot of damage. In order to achieve this, and given the limited water storage capacity available along the river, a more intelligent approach for flood regulation would be needed. The start of this approach was given by the Flemish Environment Agency (VMM) and by the HIC (hydrologic information center). These agency and center constantly measure water levels, discharges and the positions of the gates. Based on those measurements, they developed computer models

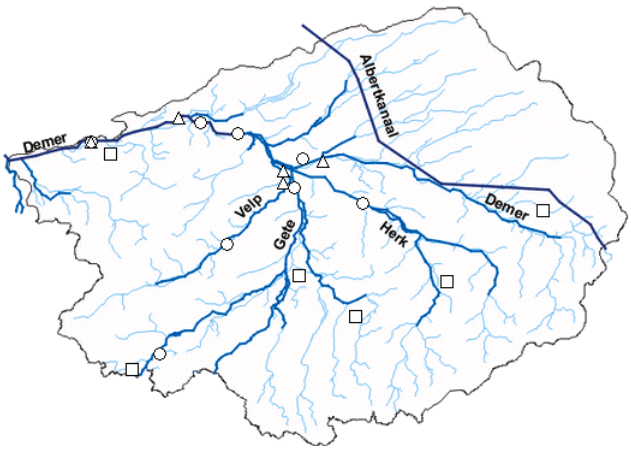


Figure 1.1: The Demer river and its tributaries. (image taken from [1])

of several navigable rivers. One of the modeled rivers is the river Demer. Currently, this model is completely finished and fully operational. For more information we refer the reader to [9].

1.1.2 Model of the Demer basin

The model developed for the river Demer has several useful applications. The first application is as decision support system for the reintroduction of meanders. The local water administration (VMM) plans to reintroduce some of the meanders that were cut off. The reason for this is to delay the flow of the water to the downstream part of the river in order to avoid floodings but also to preserve the current fauna in the basin. In the computer model of the Demer it is fairly easy to introduce several meanders and verify its effect w.r.t. floodings by rerunning historical simulations. Based on these results it will be decided in the future which meanders to reintroduce into the river system. A second application is to decide which hydraulic structures to introduce into the river system in order to achieve a specific goal with minimal cost. For example, along the Dender river there is a flood plane called the Denderbellebroek. During periods of heavy rainfall this flood plain is filled in order to avoid flooding of the river Dender. When the rainfall event is passed this flood plain is emptied again via the river Dender. However, if the water level in the river Dender stays high during the period in between 2 successive rain peaks, then the pumps used in order to empty the flood plain are not sufficient enough to empty the flood plain. In order to come up with a solution several alternatives were introduced and tested in the Demer model. Based on these simulations it turned out that adding additional regulation structures was not only the best but also the most economical alternative [10]. The third and probably most important application of the Demer model is its use as a flood warning system. Based on rain predictions the model is capable of predicting future water levels and discharges 48 hours ahead. The results of these simulations are available on-line [3].

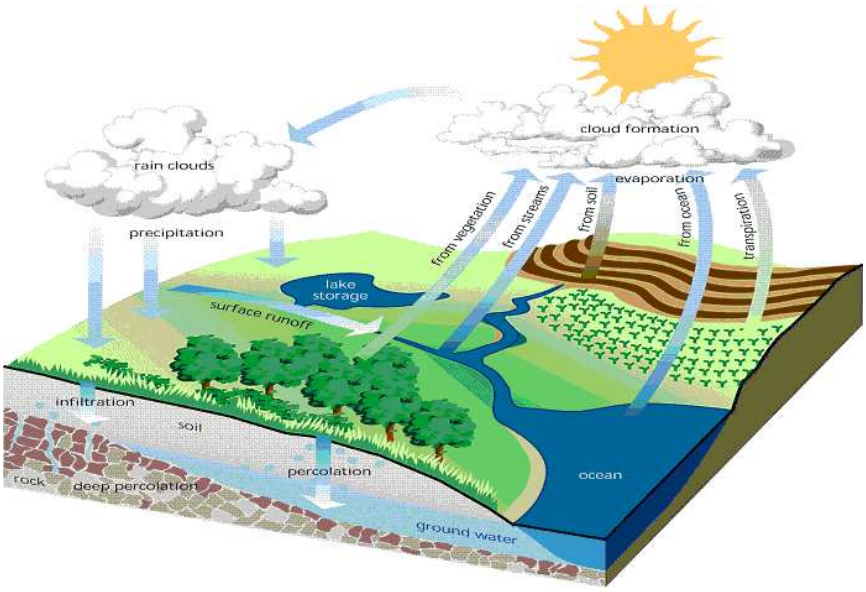


Figure 1.2: Overview of the hydrologic cycle (image taken from [2]).

Based on these simulations warning levels are displayed on-line for each of the areas in the Demer basin on basis of which decisions are made w.r.t. a possible evacuation of the areas under high flooding risk.

The Demer model consists of a hydraulic model and a hydrologic model:

- **hydraulic model:** The hydraulic computer model of the river Demer simulates the behavior of the water in the river Demer. This model is based on the conservation laws of momentum and mass (see Chapter 2). There are three different methods for creating a hydraulic model: the one-dimensional, the quasi two-dimensional and the two-dimensional model. In the one-dimensional model the modeled quantities are assumed to be variable in the flow direction of the river only. The value of the modeled quantities represent mean values of the different sections. Flood plains are added in a very simplistic way. In the quasi two-dimensional model the modeling of the river is also achieved via the one-dimensional method but the flood plains are modeled in more detail by a network of branches and spills [127]. In the two-dimensional model both the river and the flood plains are modeled by means of a two-dimensional model. In practice, a two-dimensional model is usually too time-consuming and memory inefficient. A one-dimensional model is usually not accurate enough. Therefore, the quasi two-dimensional model is usually preferred as it is more accurate than the one-dimensional approach and also less time-consuming than the two-dimensional approach. In order to develop this model for the Demer the VMM performed many measurements: river cross sections (depth, width and shape) at different

locations along the river and also the position and geometry of the different constructions in the river like weirs, culverts, bridges, etc. The resulting model gives an accurate description of the dynamics of the river Demer.

- **hydrologic model:** The hydraulic model simulates how the river reacts on a certain amount of rainfall-runoff. The amount of rainfall-runoff flowing into the river is calculated by the hydrologic model. The hydrologic model has the precipitation and evapotranspiration (water reaching the atmosphere originating from evaporation from open bodies of water and plant transpiration) as model inputs and the rainfall-runoff discharges flowing into the river system as model output. This model is physically based on the same conservation laws as the hydraulic model. The rainfall-runoff of rainwater into the river can occur in three distinct ways: via the surface, via the unsaturated soil and via the groundwater. Rainwater falling on the soil will infiltrate into the underground. If the underground is saturated or if the infiltration capacity is exceeded, rainwater will flow into the river via the surface. This water is referred at as surface runoff. Water that does not runoff via the surface can runoff via the unsaturated zone or via the groundwater, called groundwater runoff. A complete picture of this hydrologic cycle is depicted in Figure 1.2.

In order for the Demer model to be able to predict future water levels the hydrologic model needs future rain predictions as input. These rain predictions are obtained via several sources. The first source is the precipitation data obtained from the radar of the KMI [1] which has a resolution of 1 by 1 kilometer. Based on this data own predictions are made 2 hours ahead. In order to predict 2 to 6 hours ahead the Nimrod predictions from the British MetOffice are used. For predictions going from 6 hours to 2 days ahead the results from the numerical Aladin model of the KMI are used. The combination of these rain predictions with the Demer model have proven to give useful predictions of the water levels in the basin on basis of which appropriate measures have been taken in the past in order to reduce the flood damage [9]. However, every model is still an approximation of the reality and can therefore never be $100\ \%$ accurate. There is always uncertainty on the results it generates. Despite these uncertainties, the model is considered a significant improvement compared to the scientific methods used in the past. For more information about the Demer model we refer the reader to [4].

1.1.3 Goal of the thesis

In accordance to the newly adopted philosophy of damage limitation hydraulic structures were added at several locations along the river Demer in order to manage the flow of water in the river basin. The goal of these gates is to manage the water flow in such a way that floodings are limited as much as possible. These hydraulic structures

[1] KMI is the abbreviation for *Koninklijk Meteorologisch Instituut*. The KMI is a Belgian institute performing scientific research concerning the weather and providing weather-related information to several organizations.

are controlled by an advanced version of a three-position controller (see Chapter 3). This three-position controller bases its control on several logical rules that are based on expert knowledge obtained from several years of in-field experience in controlling the hydraulic structures. Though the combination of the hydraulic structures with the three-position controller has proven to be very useful in order to tackle the flooding problem, new simulations based on past rainfall data have strengthened the feeling at the local water administration that flooding damage could have been even more limited if a different control strategy would have been adopted for the control of the hydraulic structures. Therefore,

the goal of this thesis is to determine a new control strategy for the control of the gates and to verify whether this strategy performs better than the current three-position controller.

In order to achieve this goal several subgoals have to be considered and tackled:

subgoals

(i) A conceptual nonlinear model has to be derived that accurately approximates the Demer dynamics and that simulates fast enough to be used for real-time control purposes. Such a model is necessary for the design of an advanced controller.

(ii) An advanced nonlinear controller has to be derived and implemented that uses rainfall-runoff predictions and the derived conceptual model as underlying model.

(iii) A nonlinear state estimator has to be derived and implemented with the derived conceptual model as underlying model. The state estimated by the state estimator is used by the controller to determine the control action.

(iv) The performance of the nonlinear control scheme combined with the nonlinear state estimator needs to be compared with that of the three-position controller. This comparison is based on historical rainfall-runoff predictions. Besides performance also robustness w.r.t. uncertainty on the rainfall-runoff prediction needs to be tested. Note that all results and comparisons in this thesis are strictly simulation-based.

In the remainder of this chapter all the tools that have been used in this thesis and that are necessary to achieve these subgoals are discussed in more detail.

Remark 1.1. *In this thesis extra theoretical work is done concerning the stability of MPC with low-complexity invariant target sets. This work is not directly related to the main goal of this thesis, namely flood regulation. However, since a significant amount of time has been spent on this topic the results are discussed in detail in the last chapter.*

The obtained theoretical results are applied on the upstream part of the Demer in order to obtain robust setpoint regulation.

1.2 Optimization theory

Optimization is an important tool in decision science and in the analysis of physical systems. The results in this thesis are all based on optimization theory and therefore the concept of optimization will be outlined in more detail. Mathematically speaking, optimization is the minimization (or maximization) of a function subject to constraints on its variables. The unknown variables are typically represented as a vector x. The objective function f to be minimized is a scalar function of the vector x. The functions h and g are constraint functions, which are a vector functions of the vector x and define respectively inequality and equality constraints on the unknown vector x. In its general form an optimization problem is written as follows

$$\min_{x} \quad f(x), \tag{1.1}$$

$$\text{s.t.} \quad h(x) \geq 0 \tag{1.2}$$

$$g(x) = 0 \tag{1.3}$$

with (1.2) and (1.3) representing inequality and equality constraints, respectively. Depending on the specific structure of the cost function f and the constraint functions h, g optimization programs are typically classified in following two important classes:

- convex programming problems
- nonlinear nonconvex programming problems

1.2.1 Definitions

Before going into detail in which optimization programs are of importance in this thesis and how they are solved in practice some important definitions are stated.

Definition 1.1 (**Feasible set**, [39])**.** *The feasible set Ω of optimization problem (1.1) is defined as the set of points x that satisfy constraints (1.2) and (1.3), that is,*

$$\Omega = \{x | h(x) \geqslant 0, g(x) = 0\} \tag{1.4}$$

Definition 1.2 (**Global minimizer**, [39])**.** *A point x^* is a global minimizer if $f(x^*) \leq f(x)$ for all $x \in \Omega$.*

Definition 1.3 (**Local minimizer**, [39])**.** *A point x^* is a local minimizer if there is a neighborhood $\mathcal{N}$ of x^* such that $f(x^*) \leq f(x)$ for all $x \in \mathcal{N}$.*

Definition 1.4 (**Affine function**, [16])**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is affine if it can be written in the following form:*

$$f(x) = a^T x + b, \quad a \in \mathbb{R}^n, b \in \mathbb{R} \tag{1.5}$$

A linear function can be defined as a special case of an affine function where $b = 0$.

Definition 1.5 (**Convex set**, [39])**.** *A set* C *is convex if the line segment between any two points in* C *lies in* C*, i.e. if for any* $x_1, x_2 \in C$ *and any* θ *with* $0 \leq \theta \leq 1$*, we have*

$$\theta x_1 + (1 - \theta) x_2 \in C. \tag{1.6}$$

Definition 1.6 (**Convex function**, [39])**.** *A function* $f : \mathbb{R}^n \to \mathbb{R}$ *is convex if* ***dom*** f *is a convex set and if for all* $x, y \in$ ***dom*** f*, and* θ *with* $0 \leq \theta \leq 1$*, we have*

$$f(\theta x + (1 - \theta) y) \leq \theta f(x) + (1 - \theta) f(y). \tag{1.7}$$

Definition 1.7 (**Descent direction**, [91])**.** *A vector* $p_k \in \mathbb{R}^n$ *is a descent direction for* $f(x_k)$ *if*

$$\nabla^T f(x_k) p_k < 0. \tag{1.8}$$

Definition 1.8 (**Lagrangian function**, [91])**.** *The Lagrangian for the optimization program defined by cost function (1.1) and the constraints (1.2), (1.3) is defined as*

$$L(x, \lambda, \mu) = f(x) - \lambda^T g(x) - \mu^T h(x) \tag{1.9}$$

with the Lagrange multipliers $\lambda \in \mathbb{R}^p$ *and* $\mu \in \mathbb{R}^q$ *and* $\mu \geq 0$.

1.2.2 Convex programming

A convex programming problem is a special case of the general programming problem (1.1) in which the following three requirements are satisfied [39]:

- the objective function is convex (in case the optimization problem concerns a minimization),
- the inequality constraints are convex functions,
- the equality constraints are affine.

The latter two conditions imply that the feasible set Ω is a convex set. An important property of convex optimization problems is that any local optimum is also a global optimum. This means that the global optimum can be determined by local search methods. Several algorithms exist (e.g. interior point algorithms [90], active set method [91]) that are capable of solving convex programming problems efficiently (polynomial time) which is one of the reasons of the popularity of this type of optimization problems. In the following several important classes of convex programming problems are discussed.

1.2.2.1 Linear programming

A Linear Program (LP) can be written in the following standard form [16]

$$\begin{aligned} \min_{x} \quad & c^T x, \\ \text{s.t.} \quad & Ax - b = 0, \\ & Cx - d \geqslant 0, \end{aligned} \tag{1.10}$$

with $c \in \mathbb{R}^n, A \in \mathbb{R}^{p \times n}, b \in \mathbb{R}^p, C \in \mathbb{R}^{q \times n}, d \in \mathbb{R}^q$. LP's can be solved very efficiently with the *simplex method* or *interior point methods*. If the LP's are sparse or have some other exploitable structure it is even possible to solve them for problems containing tens or hundreds of thousands variables and constraints [39]. We refer to [39, 55, 80, 112] for a detailed overview in linear programming.

1.2.2.2 Quadratic programming

A Quadratic Program (QP) can be written in the following standard form [16]

$$\begin{aligned} \min_{x} \quad & \tfrac{1}{2} x^T H x + c^T x, \\ \text{s.t.} \quad & Ax - b = 0, \\ & Cx - d \geqslant 0, \end{aligned} \tag{1.11}$$

with $H \in \mathbb{R}^{n \times n}, c \in \mathbb{R}^n, A \in \mathbb{R}^{p \times n}, b \in \mathbb{R}^p, C \in \mathbb{R}^{q \times n}, d \in \mathbb{R}^q$. If the *Hessian matrix* H is positive semi-definite (i.e. $x^T H x \geqslant 0$, for all non-zero vectors with real entries, i.e. $x \in \mathbb{R}^n \backslash \{\mathbf{0}\}$) the QP is convex. If the Hessian matrix H is strictly positive-definite (i.e. $\forall x \in \mathbb{R}^n \setminus \{\mathbf{0}\} : x^T H x > 0$) the QP is strictly convex. Strictly convex QP's can be solved very efficiently [39, 91]. If the QP is sparse it is even possible to solve it for problems containing many thousands of variables and constraints [39]. In this thesis the computationally efficient MATLAB toolbox Mosek [5] is used for solving QP's.

1.2.2.3 Semi-definite programming

A Semi-Definite Program (SDP) can be written in the following standard form [36]

$$\min_{x} \quad c^T x, \tag{1.12}$$

$$\begin{aligned} \text{s.t.} \quad & F_0 + x_1 F_1 + \ldots + x_n F_n \succeq 0, \\ & Ax - b = 0, \end{aligned} \tag{1.13}$$

with $c \in \mathbb{R}^n, F_0, F_1, \ldots, F_n \in \mathcal{S}^k$ and $A \in \mathbb{R}^{p \times n}, b \in \mathbb{R}^p$. Inequality (1.13) is a linear matrix inequality. $\mathcal{S}^k$ represents the vector space of symmetric matrices of dimension $\mathbb{R}^{k \times k}$. An SDP can be solved efficiently using *interior-point methods*. SDP's play a very important role in control theory [36–38]. In this thesis the MATLAB toolbox CVX [64] is used for solving SDP's.

Remark 1.2. *QP's are used throughout the whole thesis as they form the basis of the components of the control scheme. LP's and SDP's are mainly used in Chapter* 5 *for the determination of low-complexity invariant sets on basis of which a robust linear feedback controller is determined for setpoint regulation of the upstream part of the river Demer.*

1.2.3 Nonlinear nonconvex programming

Nonlinear nonconvex programming [91] is the term used to describe an optimization problem in which the objective function and/or at least one constraint function are nonconvex. There are no effective methods for solving nonlinear nonconvex programs (NLP's). Solving this type of optimization problem can be very challenging and even intractable for problems with a few hundreds of variables. Methods for solving this type of optimization problem can be divided in those optimizing locally (local optimization) and those optimizing globally (global optimization).

In *global optimization* the global minimum of optimization problem (1.1) is determined. However, in practice finding the global minimum turns out to be intractable and is therefore only applied to problems with few unknown variables [91]. In *local optimization* one seeks only a local minimum of the nonlinear programming problem. Local optimization methods can be fast and handle large-scale problems. Besides not necessarily finding the true global minimum of the nonlinear programming problem another disadvantage of local optimization methods are that they require an initial guess for the optimization variable. This initial guess is critical and has an effect on the optimality of the final obtained solution. Despite these disadvantages local optimization methods are heavily used in practical applications, especially in applications with MPC because the solution obtained at the previous time step can be used to construct a good initial guess for the current time step. Therefore, in this thesis the focus lies on local optimization.

1.2.3.1 Globalisation strategies

All algorithms for solving nonlinear optimization problems start from a starting point x_0. Beginning at the starting point x_0 a sequence of iterates $\{x_k\}_{k=0}^{\infty}$ is generated that terminates when no more progress can be made or when it seems that a solution has been approximated with sufficient accuracy. There are two fundamental strategies for moving from the current iterate x_k to the next iterate x_{k+1}. In the following a description of both strategies is outlined for unconstrained NLP's. The first strategy is the *trust region* method [54,91]. In this method the nonlinear cost function f is usually approximated by a quadratic function of the form

$$m_k(x_k + p) = f_k + p^T \nabla f_k + \frac{1}{2} p^T B_k p, \tag{1.14}$$

with $x_k \in \mathbb{R}^n, f_k = f(x_k) \in \mathbb{R}^n, B_k \in \mathbb{R}^{n \times n}$ and step $p \in \mathbb{R}^n$. The matrix B_k is either the Hessian $\nabla^2 f_k$ or some approximation to it. After solving the following

subproblem:

$$\min_p m_k(x_k + p) \tag{1.15}$$

a new iterate $x_{k+1} = x_k + p$ is obtained. The step p is bounded by a trust region which is typically a ball defined by $\|p\|_2 \leq \Delta$. The trust region can be seen as a region around the iterate x_k for which the quadratic function m_k is an accurate enough description of the the nonlinear function f. If the new iterate x_{k+1} does not produce a sufficient decrease in f, the trust region is too large and the subproblem is resolved with a smaller trust-region radius Δ.

The second strategy for solving a nonlinear programming problem is called the *line-search* method [91, 95]. At each iteration of a line-search method first a descent direction p_k is computed by e.g. using the Newton method. The new point is then given by

$$x_{k+1} = x_k + t_k p_k \tag{1.16}$$

with $t_k \in (0, 1]$ a scalar called the step length. In the ideal case the step length t_k is found as the solution of following optimization program

$$\min_{t_k} f\left(x_k + t_k p_k\right) \quad s.t. \quad t_k \in (0,\, 1] \tag{1.17}$$

However, exactly solving this optimization problem is too time-consuming. Therefore, the line-search problem is solved approximately. Note that it is important that the approximate solution is such that there is a sufficient decrease in the objective function f in order to get stuck in a point that is not locally optimal. This is typically achieved by imposing *the Armijo condition* [91]:

$$f(x_k + t_k p_k) \leq f(x_k) + c_1 t_k \nabla f_k^T p_k, \tag{1.18}$$

for some constant $c_1 \in (0, 1)$.

1.2.3.2 Sequential quadratic programming (SQP)

SQP is a method used in order to solve NLP's. The SQP algorithm achieves this by solving a sequence of QP's until the sequence converges to a local optimum of the NLP. We focus on SQP methods that are based on active sets. An active constraint is defined as follows:

Definition 1.9 (Active constraint, [91])**.** *An inequality constraint $h_i(x) \geq 0$ is called "active" at $x^* \in \Omega$ iff $h_i(x^*) = 0$ and otherwise inactive,*

with Ω the feasible set as defined in (1.4). The active set is then defined as

Definition 1.10 (Active set, [91])**.** *The index set $A(x^*) \subset \{1, \ldots, q\}$ of active constraints is called the "active set".*

There are two types of active set SQP methods, namely IQP (Inequality QP) and EQP (Equality QP) [91]. In the EQP approach first an estimate is made of the optimal active set. Then an equality-constrained QP is solved in order to find the step p_k leading to the next iterate x_{k+1}. In the IQP approach a general inequality-constrained QP is solved at each iteration that leads to the optimal step p_k and a estimate of the optimal active set. In the IQP approach the nonlinear constraints g and h are linearized around the current iterate x_k and the cost function f is approximated by a quadratic cost function leading to the following constrained QP:

$$\min_{p} \quad f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \left(\nabla^2_{xx} L_k\right) p \tag{1.19}$$

$$\text{s.t.} \quad \nabla h_k^T p + h_k \geqslant 0, \tag{1.20}$$

$$\nabla g_k^T p + g_k = 0. \tag{1.21}$$

The solution of this step leads to an optimal step p_k. The new iterate x_{k+1} can then be calculated as $x_{k+1} = x_k + p_k$. If the iterate x_k lies close to the local solution x^* of the NLP, this method will typically lead to a sequence of iterates that converges to x^*. In order to ensure convergence when the initial iterate x_k is located remote from the local solution x^* one of the globalisation strategies (line-search or trust region method) must be applied. Both globalisation strategies use a merit function in order to determine an appropriate step p_k. A merit function that is typically used is the l_1 merit function defined as

$$T_1(x) = f(x) + \sigma \left\| g(x) \right\|_1 + \sigma \sum_{i=1}^{q} \left| \min\left(0, h_i\left(x\right)\right) \right| \tag{1.22}$$

with penalty parameter σ chosen sufficiently large. The merit function combines the value for the objective function with the amount of constraint violation. This merit function is specifically useful for algorithms that allow iterates to violate the constraints. In feasible methods for constrained optimization in which the starting point and all subsequent iterates satisfy all the constraints, the measures for the amount of constraint violation can be omitted from the merit function. In a line-search approach, given current iterate x_k and optimal step p_k obtained from the solution of the constrained QP in (1.19)-(1.21), a step $t_k p_k$ is accepted if there is sufficient decrease in the merit function (Armijo condition). In this thesis the IQP approach combined with line-search is used for solving the nonlinear programming problems related to the designed optimal controller and optimal state estimator.

1.3 Model based predictive control

Model based predictive control (MPC) is a control strategy originally developed to meet the specialized control needs of power plants [15] and petroleum refineries [48]. Currently, MPC can be found in a wide range of application areas such as chemicals [13], food processing [19], automotive [106], aerospace applications [18], ... MPC is a control strategy that reflects human behavior whereby control actions are selected

which will lead to the best predicted outcome within some limited time window. In the following MPC is discussed in more detail.

1.3.1 Components of MPC

In control applications the main goal is to steer the outputs of a dynamical system as close as possible to a desired reference value, often referred to at as ***setpoint*** and at the same time with a control action as small as possible. It is straightforward to see that both goals are conflicting goals and therefore MPC needs to be ***tuned*** in order to satisfy the demands. Besides that, constraints on the inputs and outputs must always be satisfied. Constraints can be physical limitations of the actuators or upper bounds on the value of the specific output of the system. MPC achieves these goals by making a prediction of the future system outputs within a certain time window, which is typically referred at as the ***output horizon***. In order to be able to make this prediction a ***process model*** is necessary that describes the dynamical behavior of the underlying process that needs to be controlled. Based on this prediction MPC determines an input trajectory that minimizes a desired cost function taking the constraints on the inputs and outputs into account. This is achieved by performing ***on-line*** a ***constrained optimization*** leading to an ***(sub)optimal input trajectory*** which defines the current and future optimal inputs. The ***current input*** of the optimal input sequence is then applied to the real system. At the next time step ***measurements*** are performed on some or all of the system outputs. A state ***estimator*** uses those measurements in order to estimate the current state of the system. MPC uses this estimated state in order to make new predictions and the whole process is repeated (receding horizon principle). By re-estimating the state of the system at each time step and redoing the optimization procedure robustness is obtained against disturbances, model uncertainties and modeling errors.

1.3.2 Basic algorithm formulation

The process model used in MPC can be of any form. The most common used models are transfer function models, state space models, and finite impulse response models [102]. In this thesis the focus lies on state space models because they have some specific advantages compared to the other model types. A linear state space model is defined as

$$x_{k+1} = Ax_k + Bu_k \tag{1.23}$$
$$y_k = Cx_k \tag{1.24}$$

with x_k the vector of states of the system at discrete time k, u_k the vector of inputs of the system at discrete time k and y_k the vector of outputs of the system at discrete time k. In its basic formulation the goal of the controller is to steer the output y_k as fast as possible to the desired setpoint r_{ss}. The setpoint r_{ss} typically coincides with a steady state x_{ss} and a corresponding steady state input u_{ss} and is determined by

$$x_{ss} = Ax_{ss} + Bu_{ss} \tag{1.25}$$

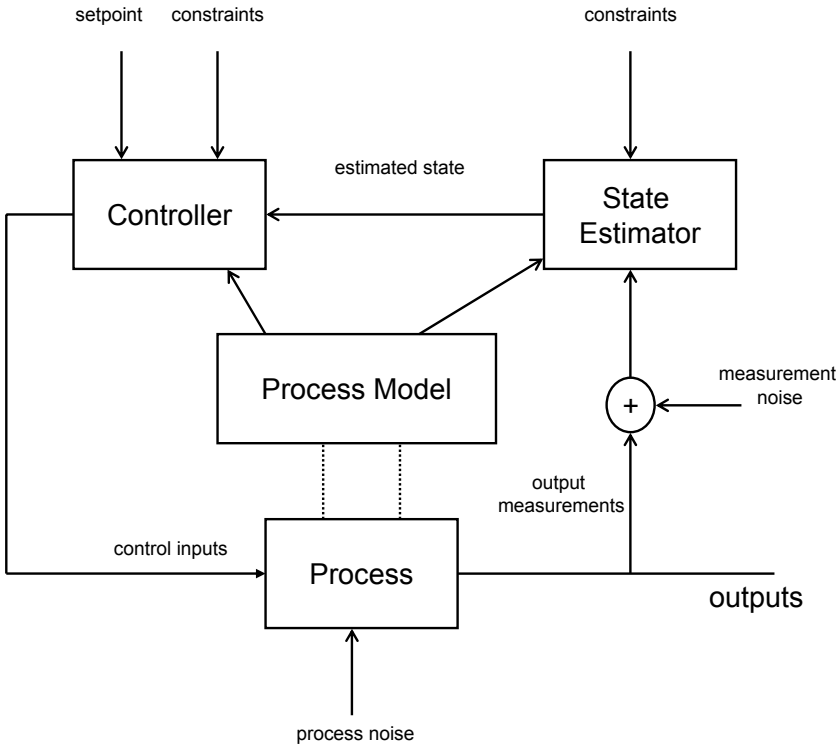


Figure 1.3: Overview of a practical control scheme

$$r_{ss} \quad = \quad Cx_{ss}. \tag{1.26}$$

Once the desired reference output r_{ss} is chosen and the corresponding steady state x_{ss} is determined, the controller tries to steer the output to the reference value r_{ss}. MPC achieves this by solving the following on-line optimization problem

Algorithm 1.1.

$$\begin{aligned} \min_{\substack{u_k,\ldots,u_{k+N_c-1} \\ x_k,\ldots,x_{k+N_p} \\ y_k,\ldots,y_{k+N_p}}} & \sum_{i=0}^{N_p} \left(y_{k+i} - r_{ss}\right)^T R \left(y_{k+i} - r_{ss}\right) + \\ & \sum_{i=0}^{N_c-1} \left(u_{k+i} - u_{ss}\right)^T Q \left(u_{k+i} - u_{ss}\right) \end{aligned} \tag{1.27}$$

subject to the following constraints for $i = 1, \ldots, N_p$*:*

$$x_{k+i+1} \quad = \quad Ax_{k+i} + Bu_{k+i}, \tag{1.28}$$

$$
\begin{aligned}
y_{k+i} &= Cx_{k+i}, && (1.29)\\
u_{k+i} &\in \mathbb{U}, && (1.30)\\
x_{k+i} &\in \mathbb{X}, && (1.31)\\
u_{k+i} &= u_{k+N_c-1}, i \geqslant N_c && (1.32)\\
x_k &= \hat{x}_k, && (1.33)
\end{aligned}
$$

with $\hat{x}_k$ the estimated/measured current state, N_c the control horizon and N_p the prediction horizon. The control horizon N_c is used to reduce the size of the optimization problem by reducing the number of unknown future inputs. The matrices R and Q are positive-definite diagonal weighting matrices, with R weighing the setpoint deviations of the outputs and Q weighing the deviation of the different inputs w.r.t. their steady state value. In practice both weighting matrices need to be tuned in order to obtain a desired performance. Well-tuned weighting matrices Q and R lead to a balance between performance, input activity and speed of response. The solution of this optimization problem needs to be obtained within the sampling time. Note that this optimization problem is a QP due to the quadratic objective funtion, the convex constraints and the system dynamics which are assumed to be linear. However, in this thesis the system dynamics are nonlinear and therefore nonlinear model predictive control (NMPC) is needed. In this thesis the developed NMPC scheme solves an on-line nonlinear optimization problem of the following form

Algorithm 1.2.

$$
\begin{aligned}
\min_{\substack{x_k,\ldots,x_{k+N_p}\\ y_k,\ldots,y_{k+N_p}\\ u_k,\ldots,u_{k+Nc-1}}} \quad & \sum_{i=0}^{N_p} \left(y_{k+i}-y_r\right)^T Q \left(y_{k+i}-y_r\right) + \\
& \sum_{i=0}^{N_c-1} \left(u_{k+i}-u_r\right)^T R \left(u_{k+i}-u_r\right)
\end{aligned} \tag{1.34}
$$

subject to the following constraints for $i = 1,\ldots,N_p, j = 0,\ldots,N_c-1$*:*

$$
\begin{aligned}
x_k &= \hat{x}_k && (1.35)\\
x_{k+i+1} &= f(x_{k+i}, u_{k+i}, d_{k+i}) && (1.36)\\
y_{k+i} &= Cx_{k+i} && (1.37)\\
u_{k+i} &= u_{k+N_c-1}, \textit{for } i \geq N_c && (1.38)\\
\underline{y} &\leq y_{k+i} \leq \bar{y} && (1.39)\\
\underline{u} &\leq u_{k+j} \leq \bar{u} && (1.40)\\
\left\|u_{k+j+1}-u_{k+j}\right\|_\infty &\leq \Delta_{max} && (1.41)
\end{aligned}
$$

In Chapter 3 it will be discussed in more detail how the nonlinear model predictive controller can be used and solved for flood regulation purposes. Note that more general

NMPC formulations exist in which the objective function (1.34) and/or the constraints on the states and inputs are nonlinear.

Remark 1.3. *In order to have the maximum possible degrees of freedom in the developed NMPC scheme of this thesis we assume* $N_p = N_c = N$.

1.4 Moving horizon estimation

In order to solve the optimal control problems defined in algorithms 1.1 and 1.2 the current state x_k of the system needs to be known. One way to obtain this information is to measure all the states of the system. However, in most practical applications not all the state variables are measured. The cost of the required sensors may be prohibitive, or it may be physically impossible to measure all the state variables, as in, for example, a nuclear power plant. Therefore, the state of the system must be reconstructed on the basis of past measurements. This state estimation is performed by the state estimator, also referred to as an observer. In this thesis a moving horizon

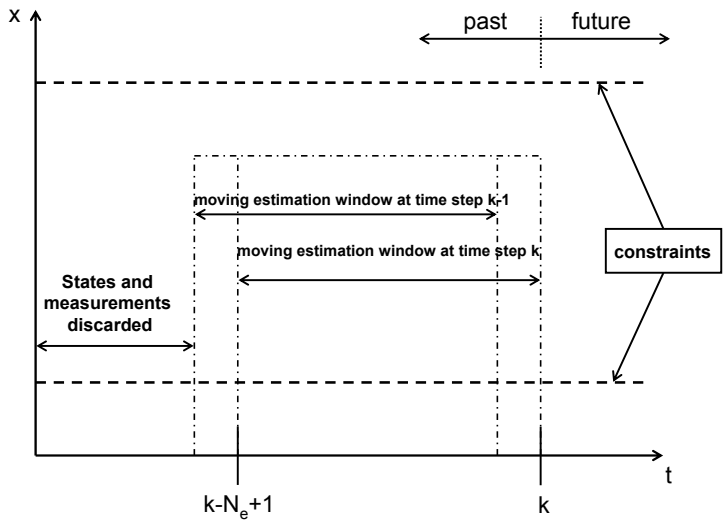


Figure 1.4: The principle of Moving Horizon Estimation. At time step $k-1$ a constrained optimization problem is solved with a window of size N_e leading to a sequence of estimated states. The last state of this sequence is passed to the model predictive controller to determine the optimal control sequence and the estimation window shifts over 1 sampling time.

estimation (MHE) [56, 65, 100] is performed in order to reconstruct the state of the system. In order to do this the moving horizon estimator uses a model of the process. Typically the model used for estimating the states of a linear model is defined as:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{1.42}$$
$$y_k = Cx_k + v_k \tag{1.43}$$

with w_k representing process noise and v_k measurement noise. Then by solving an *on-line* optimization problem the moving horizon estimator calculates the corresponding sequence of states $x_1, x_2, \ldots, x_k$ that is the most probable given the linear model, the sequence of past measurements $y_1, y_2, \ldots, y_k \ldots$ and the covariances of the sequences $\{w_k\}, \{v_k\}$. However, casting this as an optimization problem leads to a programming problem that increases with increasing value of the time k. At some time instant k will be very large and the optimization problem will become intractable. In order to control the size of the optimization problem, a moving estimation window of size N_e is introduced. This is in the same spirit as the moving control horizon used by MPC. Instead of trying to perform a least squares estimate of all the past states, MHE only performs a least squares estimate of the states lying inside the estimation window and therefore limits the size of the optimization problem. Measurements outside the estimation window are discarded and not explicitly taken into account in the optimization problem. The last state of the optimal state sequence $\{x_k\}$ calculated by the moving horizon estimator is passed to the model predictive controller. In the next time step the estimator window shifts over 1 sampling time and the steps are repeated. This moving horizon principle is illustrated in Figure 1.4.

Assume for notational convenience that the sequence $x_1, \ldots, x_{N_e-1}, x_{N_e}$ coincides with the state sequence $x_{k-N_e+1}, \ldots, x_{k-1}, x_k$, then for the linear model described in (1.42)-(1.43) the on-line optimization problem to be solved by the MHE scheme is the following QP:

Algorithm 1.3.

$$\begin{aligned} \min_{\substack{x_{k-N_e+1},\ldots,x_k \\ w_{k-N_e+1},\ldots,w_{k-1} \\ v_{k-N_e+1},\ldots,v_k}} \quad & \left(x^e_{k-N_e+1}\right)^T P^{-1}_{k-N_e+1|k-N_e} \left(x^e_{k-N_e+1}\right) + \\ & \sum_{i=0}^{N_e-1} v^T_{k-i} R^{-1} v_{k-i} + \sum_{i=1}^{N_e-1} w^T_{k-i} Q^{-1} w_{k-i} \end{aligned} \tag{1.44}$$

s.t.
$$v_{k+i} = y_{k+i} - Cx_{k+i}, \qquad i = -N_e+1, \ldots, 0 \tag{1.45}$$
$$x_{k+i+1} = Ax_{k+i} + Bu_{k+i} + w_{k+i}, \quad i = -N_e, \ldots, -1 \tag{1.46}$$
$$x_{k+i} \in \mathbb{X}, \qquad i = -N_e+1, \ldots, 0 \tag{1.47}$$

with $x^e_{k-N_e+1} = x_{k-N_e+1} - x_{k-N_e+1|k-N_e}$ and $x_{k-N_e+1|k-N_e}$ the estimate of the state x_{k-N_e+1} at time $k - N_e$. The positive definite weighting matrices

R^{-1}, Q^{-1} and $P^{-1}_{k-N_e+1|k-N_e}$ are quantitative measures of our confidence in the output model, the dynamic system model and the initial estimate $x^e_{k-N_e+1}$, respectively. At time k the measurements $y_{k-N_e+1,\ldots,y_k}$ appear explicitly in the MHE scheme. The statistics $x_{k-N_e+1|k-N_e}$ and $P_{k-N_e+1|k-N_e}$ represent the information about the states contained in the (previously discarded) measurements $y_1, \ldots, y_{k-N_e}$. Also note that based on knowledge of the process it may be beneficial to constrain the range of values that the unknown variables can assume by means of constraint (1.47). This gives the least squares estimates that lie within the specified bounds, something which cannot be enforced in Kalman filter-based methods.

In this thesis the underlying model is nonlinear and therefore the QP defined in algorithm 1.3 cannot be used to estimate the states. The linear system constraints defined by equation (1.46) have to be replaced by nonlinear system equations and the resulting estimation is referred at as nonlinear moving horizon estimation (NMHE). The nonlinear optimization problem that has to be solved *on-line* in a NMHE scheme is as follows:

Algorithm 1.4.

$$\begin{aligned} \min_{\substack{x_{k-N_e+1},\ldots,x_k \\ w_{k-N_e+1},\ldots,w_{k-1} \\ v_{k-N_e+1},\ldots,v_k}} \quad & \left(x^e_{k-N_e+1}\right)^T P^{-1}_{k-N_e+1|k-N_e}\left(x^e_{k-N_e+1}\right) + \\ & \sum_{i=0}^{N_e-1} v_{k-i}^T R^{-1} v_{k-i} + \sum_{i=1}^{N_e-1} w_{k-i}^T Q^{-1} w_{k-i} \end{aligned} \tag{1.48}$$

$$\text{s.t.} \qquad v_{k+i} = y_{k+i} - Cx_{k+i}, \qquad i = -N_e+1,\ldots,0 \tag{1.49}$$

$$x_{k+i+1} = f(x_{k+i}, Bu_{k+i}, w_{k+i}), \quad i = -N_e,\ldots,-1 \tag{1.50}$$

$$x_{k+i} \in \mathbb{X}, \qquad i = -N_e+1,\ldots,0. \tag{1.51}$$

The nonlinear optimization problem is referred at as nonlinear moving horizon estimation (NMHE). In Chapter 4 it will be shown how this nonlinear MHE scheme can be solved by means of SQP.

Remark 1.4. *Note that the on-line optimization problem has to be solved within the sampling time. If more time is needed MHE cannot be used for state estimation.*

1.5 Stability framework

Classical linear control theory deals with the behavior of linear dynamical systems. Stability is one of these behaviors and linear control theory offers tools to analyse this behavior for linear systems [58]. The considered models are typically transfer function models. Stability can be ensured if the poles of the closed-loop system lie inside the unit circle (in the discrete-time case). A drawback of this theory is that the transfer function models can only be used to model SISO (single input, single output) systems.

A better way to model MIMO (multiple input, multiple output) systems is by means of a state space model. In its general form a state space model is defined by following set of equations:

$$x_{k+1} = Ax_k + Bu_k \tag{1.52}$$
$$y_k = Cx_k. \tag{1.53}$$

A common method for introducing feedback is by state feedback. In state feedback the input is defined as $u_k = Kx_k$. This leads to the following set of equations for the closed-loop system:

$$x_{k+1} = \Phi x_k \tag{1.54}$$
$$y_k = Cx_k \tag{1.55}$$

with closed-loop matrix $\Phi = A + BK$. The poles of this closed-loop system coincide with the eigenvalues of the closed-loop matrix Φ. Again, the closed-loop system is stable if the eigenvalues of the matrix Φ lie inside the unit circle. Though these tools have been proven very useful for the analysis of linear systems, they cannot be used for stability analysis of MPC because the constraints present in the MPC framework turn the controller into a nonlinear controller. Therefore, Lyapunov stability theory is needed [69, 81].

1.5.1 Stability conditions

Let $x \in X_0$ with the set $X_0 \in \mathbb{R}^n$ containing the origin. Let $f : \mathbb{R}^n \to \mathbb{R}^n$ and suppose that the autonomous nonlinear dynamical system

$$x_{k+1} = f(x_k) \tag{1.56}$$

with $x_k \in \mathbb{R}^n$, has an equilibrium point at the origin $x = 0$, that is, $f(0) = 0$. Let $x_0 \in X_0$ and let $\{x_k\} \subset X_0, k \geq 0$ be the resulting sequence satisfying (1.56). Then, following types of stability can be distinguished.

Definition 1.11 (**Lyapunov Stability**, [62])**.** *The system (1.56) is Lyapunov stable in X_0 if for any $\epsilon > 0$, there exists a $\delta > 0$ such that*

$$x_0 \in X_0 \; and \; \|x_0\| < \delta \Rightarrow \|x_k\| < \epsilon \; for \; all \; k \geq 0. \tag{1.57}$$

Definition 1.12 (**Asymptotic Stability**, [62])**.** *The system (1.56) is asymptotically stable if it is Lyapunov stable in X_0 and if there exists a $\delta < 0$ such that*

$$x_0 \in X_0 \; and \; \|x_0\| < \delta \Rightarrow \lim_{k \to \infty} x_k = 0. \tag{1.58}$$

Definition 1.13 (**Exponential Stability**, [62])**.** *The system (1.56) is exponentially stable in X_0 if it is asymptotically stable in X_0 and there exist constants $\alpha > 0$ and $\beta \in (0, 1)$ such that*

$$x \in X_0 \Rightarrow \|x_k\| \leq \alpha \|x_k\| \beta^i \; for \; all \; i \geq 0. \tag{1.59}$$

Since MPC corresponds to a nonlinear feedback scheme due to the presence of constraints, stability analysis of MPC is based on the Lyapunov theory. In order to ensure stability a Lyapunov function needs to be found that satisfies several conditions. These conditions are expressed in the following lemma.

Lemma 1.1 (**Lyapunov Stability**, [16])**.** *The autonomous system (1.56) is Lyapunov stable for all $x_0 \in X_0$ if there exists a convex function $V : X_0 \rightarrow \mathbb{R}$ (Lyapunov function) that satisfies the following conditions:*

- $V(f(x)) \leq V(x), \forall x \in X_0 \setminus \{0\}$,
- $V(0) = 0$.

Remark 1.5. *Note that asymptotic stability can be obtained if $V(f(x)) < V(x), \forall x \in X_0 \setminus \{0\}$.*

The conditions for guaranteeing Lyapunov stability are sufficient for nonlinear controllers in the absence of constraints. However, when constraints are present an extra condition is necessary for guaranteeing stability, namely recursive feasibility.

Definition 1.14 (**Recursive Feasibility**, [16])**.** *A receding horizon scheme is recursively feasible if feasibility of the optimal control problem at time step k ensures the existence of a feasible solution of the optimal control problem at time step $k+1$.*

Note that by means of induction it can be shown that if the optimal control problem at time step $k = 0$ is feasible, recursive feasibility then guarantees feasibility for all time steps $k > 0$.

1.5.2 MPC stability

A typical method to guarantee stability for MPC is offered by the dual-mode MPC paradigm. Within this paradigm one makes the following assumptions:

- The terminal state x_N lies inside a terminal set X_f with $0 \in X_f$.
- The inputs outside the horizon N reverts to a linear closed-loop with known and fixed feedback $\kappa_f(x)$.

Now, let $\mathcal{P}_k(x)$ be the optimal control problem defined at time step k for the model predictive controller with fixed horizon N and let the nonlinear system dynamics be described by $x_{k+1} = f(x_k, u_k)$. In order for the dual-mode MPC to be recursive feasible, the following conditions must be satisfied [85]:

$$X_f \subset \mathbb{X} \tag{1.60}$$
$$\kappa_f(x) \in \mathbb{U}, \forall x \in X_f \tag{1.61}$$

$$X_f \text{ positively invariant under } \kappa_f(.)(f(x,\kappa_f(x)) \in X_f, \forall x \in X_f), \quad (1.62)$$

with $\mathbb{X}$ the feasible constraint set and $\mathbb{U}$ the set of feasible inputs. In order to see that conditions (1.60)-(1.62) ensure recursive feasibility, assume $\mathbf{u}_0^*(x) = \{u_0^*(0;x), u_0^*(1;x), \ldots, u_0^*(N-1;x)\}$ to be the optimal control sequence obtained from the solution of $\mathcal{P}_0(x)$. The input sequence $\mathbf{u}_0^*(x)$ gives rise to the optimal state trajectory $\mathbf{x}_0^*(x) = \{x_0^*(1;x), x_0^*(2;x), \ldots, x_0^*(N;x)\}$. Because the input sequence $\mathbf{u}_0^*(x)$ and state sequence $\mathbf{x}_0^*(x)$ are the optimal solution of $\mathcal{P}_0(x)$ they satisfy the input and state constraints. Moreover, the terminal state $x_0^*(N;x)$ lies inside the terminal set X_f. The input $u_0^*(0;x)$ is applied to the system by which the state of the system at the next time step goes to $x_0^*(1;x)$. At this time step the optimal control problem is solved with $x_0^*(1;x)$ as initial state. This control problem is referred at as $\mathcal{P}_1(x)$. Now assume the input sequence $\mathbf{u}^+ = \{u_0^*(1;x), \ldots, u_0^*(N-1;x), \kappa_f(x_0^*(N;x))\}$. Because the input sequence $\{u_0^*(1;x), \ldots, u_0^*(N-1;x)\}$ is part of the optimal solution of $\mathcal{P}_0(x)$, this sequence satisfies the input constraints and since $x_0^*(N;x) \in X_f$ it follows from (1.61) that the input $\kappa_f(x_0^*(N;x))$ also satisfies the input constraints. It is straightforward to see that this input sequence would lead to the following state sequence $\mathbf{x}^+ = \{x_0^*(2;x), \ldots, x_0^*(N;x), f(x_0^*(N;x), \kappa(x_0^*(N;x)))\}$. Because the state sequence $\{x_0^*(2;x), \ldots, x_0^*(N;x)\}$ is part of the optimal solution of $\mathcal{P}_0(x)$ this state sequence satisfies the state constraints. Since $x_0^*(N;x) \in X_f$ it follows from (1.62) that the state $f(x_0^*(N;x), \kappa(x_0^*(N;x))) \in X_f$ and combined with (1.60) this ensures feasibility of the state $f(x_0^*(N;x), \kappa(x_0^*(N;x)))$. So from feasibility of the solution at time step 0, feasibility at the next time step is shown. Therefore, conditions (1.60)-(1.62) ensure recursive feasibility.

In order to completely ensure stability for MPC a suitable Lyapunov function $V(x)$ has to be found. Such a Lyapunov function can usually be derived from the objective function. Assume, without loss of generality, that the reference lies in the steady state corresponding with $x = 0$ and $u = 0$, then the objective function in a MPC framework is typically quadratic and has the following form at time step k

$$J(\mathbf{x}, \mathbf{u}, k) = \sum_{i=1}^{N} \|x_{k+i}\|_Q^2 + \sum_{i=0}^{N-1} \|u_{k+i}\|_R^2. \quad (1.63)$$

In order to be able to guarantee stability the cost function is typically modified into

$$V(x, \mathbf{u}, k) = \sum_{i=1}^{N-1} \|x_{k+i}\|_Q^2 + \sum_{i=0}^{N-1} \|u_{k+i}\|_R^2 + F(x_{k+N}) \quad (1.64)$$

with x the initial state at time step k, $\mathbf{u} = \begin{bmatrix} u_k^T & u_{k+1}^T & \ldots & u_{k+N-1}^T \end{bmatrix}^T$ and terminal cost $F(x_{k+N})$. Note that the terminal cost function $F(.)$ is also a Lyapunov function. Combined with the proven recursive feasibility property, this cost function is a stabilizing Lyapunov function if the following condition is satisfied

$$F(f(x_0^*(N;x), \kappa(x_0^*(N;x)))) - F(x_0^*(N;x)) + \|x_0^*(N;x)\|_Q^2 + \|\kappa(x_0^*(N;x))\|_R^2 \leq 0. \quad (1.65)$$

In order to see this let $V(x, \mathbf{u}_0^*(x), 0)$ be the value of the cost function at time step 0, then the value of the cost function in the next time step can be written as

$$\begin{aligned} V(x^+, \mathbf{u}^+, 1) = V(x, \mathbf{u}_0^*(x), 0) - \|x_0^*(1;x)\|_Q^2 \\ - \|u_0^*(0;x)\|_R^2 - F(x_0^*(N;x)) \\ + \|x_0^*(N;x)\|_Q^2 + \|\kappa(x_0^*(N;x))\|_R^2 \\ + F(f(x_0^*(N;x), \kappa(x_0^*(N;x)))). \end{aligned} \tag{1.66}$$

with $x^+ = x_0^*(1;x)$. This cost satisfies the stability condition

$$V(x^+, \mathbf{u}^+, 1) \leq V(x, \mathbf{u}_0^*(x), 0) - \|x_0^*(1;x)\|_Q^2 - \|u_0^*(0;x)\|_R^2 \tag{1.67}$$

if condition (1.65) is satisfied. Note that conditions (1.60)-(1.62) combined with condition (1.65) ensure closed-loop asymptotic stability of the MPC scheme. With $\Delta F(x_0^*(N;x), \kappa(x_0^*(N;x))) = F(f(x_0^*(N;x), \kappa(x_0^*(N;x)))) - F(x_0^*(N;x))$ the following box summarizes the conditions for stability.

Stability conditions for MPC

- $X_f \subset \mathbb{X}$
- $\kappa_f(x) \in \mathbb{U}, \forall x \in X_f$
- X_f positively invariant under $\kappa_f(.)(f(x, \kappa_f(x)) \in X_f, \forall x \in X_f)$
- $\Delta F(x_0^*(N;x), \kappa(x_0^*(N;x))) + \|x_0^*(N;x)\|_Q^2 + \|\kappa(x_0^*(N;x))\|_R^2 \leq 0.$

1.5.3 Set invariance

In the previous subsection the importance of recursive feasibility in the stability framework was explained. In order to guarantee recursive feasibility a terminal constraint set was added to the optimal control problem. One of the properties of this terminal set was that it was invariant for a certain feedback input $u_k = \kappa_f(x_k)$. The following definition defines the concept of set invariance:

Definition 1.15. *A set X_f is positively invariant for the system $x_{k+1} = f(x_k, u_k)$ under $u_k = \kappa_f(x_k)$ if and only if*

$$x_k \in X_f \Rightarrow f(x_k, \kappa_f(x_k)) \in X_f. \tag{1.68}$$

Basically, positive invariance states that once the state of the system lies inside the set X_f the system stays inside the set X_f under the feedback $\kappa_f(.)$. But positive invariance is not enough in order to ensure recursive feasibility and therefore an extra condition is necessary, namely the positively invariant set X_f needs also to be feasible:

Definition 1.16. *The set X_f is feasible w.r.t. the state and input constraints defined by the sets $\mathbb{X}$ and $\mathbb{U}$ if and only if $X_f \subseteq \mathbb{X}$ and $\forall x \in X_f : \kappa_f(x) \in \mathbb{U}$.*

Note that there are no restrictions on the shape of the set X_f. However, the set X_f is usually chosen convex because such sets typically lead to convex programming problems which can be solved efficiently. Therefore, the set X_f is typically ellipsoidal or polyhedral. The feedback input $u_k = \kappa_f(x_k)$ can also have any general form but in order to derive efficiently solvable algorithms a linear feedback $u_k = Kx_k$ is typically assumed. In this thesis we will mainly focus on polyhedral sets. More specifically, in Chapter 5 the focus lies on polyhedral low-complexity invariant sets. Such sets have a special symmetric structure that makes them scalable with the state dimension. A method will be discussed how to efficiently calculate polyhedral low-complexity invariant sets for uncertain linear systems. At the same time the feedback gain K corresponding to this optimal set is also determined. It will be shown that the newly proposed algorithm performs better than the existing ones. The new algorithm will then be used to design a robust feedback controller for setpoint regulation of the upstream part of the river Demer.

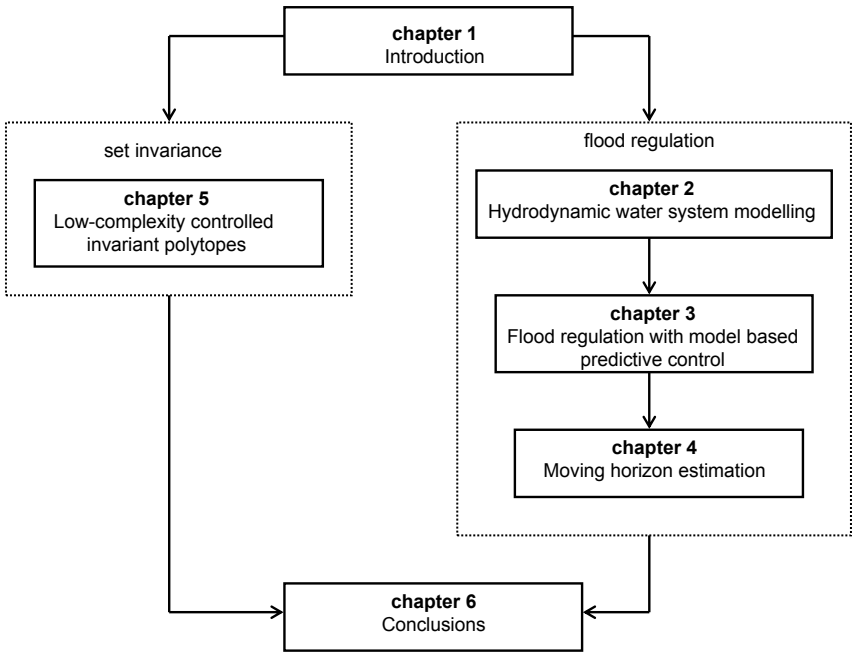


Figure 1.5: Overview and connection between the different chapters in this thesis.

1.6 Chapter by chapter overview

The main goal of this thesis is to design a control scheme based on MPC and MHE and to assess whether this new control scheme is better for flood regulation purposes than the currently used control scheme based on a three-position controller. In order to achieve such a system and test its performance several components had to be worked out and tested in this thesis. Each chapter in this thesis discusses the implementation and the testing of an individual component. In Figure 1.5 an overview of the different chapters in this thesis is given as well as how they relate. In the following a summary will be given of each chapter in this thesis:

Chapter 2: This chapter concerns the derivation of a simplified model of the river Demer. Models that are typically used for river control applications are not suitable for flooding regulation as these models are linear and can therefore not capture the nonlinear dynamics excited during flood events. Their use is limited to setpoint regulation. In order to be able to capture the nonlinear dynamics excited during flood events a new type of model was derived in this chapter, namely a reservoir type model. Models of the reservoir type have never been used before for flooding regulation purposes. The reservoir model derived in this chapter is fast and accurate enough to be used in a MPC framework. The model parameters are calibrated based on historical simulation data from 1995 and 1998 and validated based on historical simulation data from 1999 and 2002.

Chapter 3: In this chapter a nonlinear MPC scheme is outlined in detail. The proposed nonlinear MPC scheme minimizes a quadratic cost function subject to constraints. All constraints are linear with exception of the equality constraints representing the nonlinear system dynamics. Therefore, the resulting *on-line* optimization problem is nonlinear. This nonlinear optimization scheme is solved by means of SQP. A special way of constraint handling is also discussed as upper constraints on some water levels might become infeasible during the flood event. Besides that extra restrictions concerning the usage of the water reservoirs are imposed by the local water administration and have to be dealt with in the constraint handling. The proposed MPC is tested on the historical flood event of 1998 and its performance compared with the current three-position controller. Full state feedback is assumed in the simulations. In order to assess the robustness of the receding horizon strategy of MPC extra simulations are performed in which Gaussian uncertainty is added to the rainfall data. Again, the results of the model predictive controller are compared with those of the three-position controller.

Chapter 4: In this chapter the state estimator used in this thesis is discussed in detail. MPC needs information about the current state of the system before it can make predictions. The current state is derived from current and past measurements by means of a state estimator. The state estimation discussed in this chapter is a nonlinear moving horizon estimator. It is outlined how a moving horizon estimator can be implemented for the nonlinear river model. Then historical simulations of 1998 are performed in which MPC and MHE are combined. It is assumed that only the water

levels are measured. In order to assess the robustness of the moving horizon estimator, Gaussian noise is added to these measurements. Similar to the simulations with full state feedback, the simulations performed in this chapter are with and without Gaussian uncertainty added to the rainfall data. The performance of MPC is again compared with that of the three-position controller and the results show that MPC outperforms the three-position controller w.r.t. flood regulation.

Chapter 5: This chapter mainly deals with a theoretical contribution related to MPC stability. The chapter focuses on the class of linear time-varying (LTV) systems. Stabilization of these systems with MPC can be guaranteed by enforcing the final state to lie inside a feasible invariant terminal set. The terminal set is here considered to be a low-complexity polytopic set. The proposed method allows to calculate such sets with maximal volume by means of a sequence of convex programs. In contrast to existing methods to calculate such sets, the proposed method scales well with the state dimension of the problem. The efficiency of the method is shown by its application on several practical control problems. The chapter ends by using the proposed method to determine a robust linear state feedback gain for improved setpoint regulation of the upstream part of the river Demer.

1.7 Specific contributions of this thesis

The following contributions have been made in this thesis:

- In order to be able to use MPC, a model is needed accurate enough to capture the relevant dynamics of the river system but that at the same time is also fast enough to be used for real-time control purposes. In this thesis a model satisfying these conditions is developed. The resulting model is a nonlinear reservoir model which is calibrated and validated on simulation results with the existing more detailed hydrodynamic model of the Demer river system. The model is developed in cooperation with the Hydraulic Laboratory of the K.U.Leuven. The results of this process are discussed in Chapter 2.

  publications related to this topic: [24–27, 41, 125–127]

- The most important contribution of this thesis is the development of a nonlinear MPC scheme for flood regulation purposes which is fully discussed in Chapter 3. MPC has been used in several works for the control of river systems but these works have always been limited to setpoint regulation of water levels (e.g. [44, 93, 94]). MPC for setpoint regulation is significantly easier than MPC for flooding regulation as simple linear MPC is already enough to obtain satisfactory results. MPC for flooding regulation requires a more sophisticated approach. The first difficulty that arises in MPC for flood regulation is the uncontrollability of the gates. Each gate can operate in different modes. In some modes the discharge over the gate is independent of the gate position. If the gate enters

such a mode the model predictive controller cannot control it anymore. In this thesis this problem is outlined and a solution for this problem is given. Another difficulty that arises in MPC for flooding regulation is the constraint handling of the upper limits of the water levels. During periods of heavy rainfall it is impossible to satisfy all these constraints and a proper constraint handling is needed. Besides that the constraint handling also needs to deal with extra restrictions imposed by the local water administration regarding the usage of the water reservoirs. In this thesis a proper constraint handling scheme is proposed that tackles the aforementioned difficulties. The resulting nonlinear MPC scheme is then tested by simulating the historical rainfall event of 1998 and this one is compared with the simulations of the three-position controller for the same rainfall event. The conclusion of this simulation is that the model predictive controller outperforms the current three-position controller. Also simulations with uncertainty on the rainfall-runoff predictions are performed in order to test the robustness of the model predictive controller. Despite the uncertainty, these simulations show that MPC still outperforms the three-position controller.

publications related to this topic: [24–27, 41, 125–127]

- Another contribution is the implementation of a nonlinear moving horizon estimator which is fully discussed in Chapter 4. In the practical river control problems discussed in literature a simple linear Kalman filter is typically used. Since these applications concern setpoint regulation the underlying models are linear and a linear Kalman filter is sufficient to obtain accurate state estimates. However, in this thesis a linear Kalman filter cannot be used because the underlying model is strongly nonlinear and during the simulations the nonlinear dynamics are all excited. Therefore, a more complex state estimator scheme is needed. In this thesis we opt for a moving horizon estimator because of its specific advantages compared to other estimators (good performance for nonlinear processes, possibility to take constraints into account, ...). MHE has never been used before for estimating the states in river control applications and therefore the results discussed in this thesis are the first results of this kind. In Chapter 4 the moving horizon estimator is implemented and tested in combination with MPC and in order to make realistic simulations noise was added on the water level measurements. These simulations show that MHE returns accurate state estimates and that the combination of the model predictive controller with MHE still outperforms the three-position controller despite the uncertainties added to the rainfall-runoff predictions and the water level measurements.

- A final contribution of this thesis is a methodology that makes it possible to calculate low-complexity controlled invariant polytopic sets for linear time-varying systems. The proposed methodology consists of two steps. In the first step a method is developed that allows to calculate initial invariant low-complexity polytopes by solving a convex programming problem. This is the first method published in literature in which such a set can be found as the

result of a convex programming problem. However, this convex formulation is achieved by adding conservatism into the problem and therefore the resulting set is not optimal w.r.t. volume size. Therefore, in the second step a scheme is developed that increases the volume of this set. The proposed scheme consists of a sequence of convex programming problems. Each sequence leads to a new invariant set with an increased volume until convergence is obtained. The initial invariant set obtained in the first step is needed in order to initiate this sequence. This sequence of convex programming problems is an improvement of the results described in [47]. In [47] a sequence of linear programs is used in order to increase the volume of an initial invariant set. However, this sequence is vertex-based and therefore intractable for high dimensions as the number of vertices scales exponentially with the state dimension. The method already poses problems for systems with 6 state dimensions. Moreover, the method turns out to be very conservative. The new sequence of convex programs proposed in this thesis scales linear with the state dimension and is significantly less conservative than the method described in [47]. The effectiveness of the new approach is shown in several practical examples. The chapter ends with an example in which the proposed methodology is used to determine a robust linear feedback for improved setpoint regulation of the upstream part of the river Demer.

publications related to this topic: [20–23]

Chapter 2

Hydrodynamic Water System Modeling

In this chapter the main focus is the determination of a conceptual hydraulic model of the river Demer. The chapter starts by giving an overview of some basic hydraulic concepts and the three conservation laws used in river modeling in Section 2.2. In Section 2.3 an overview is given of different modeling techniques that are used in practice for river control applications and for each of the techniques their advantages and drawbacks are discussed. In Section 2.4 it is outlined how a nonlinear reservoir model of the river Demer is created and validated by means of historical simulation data. The chapter ends with some conclusions in Section 2.5.

2.1 Introduction

As outlined in chapter 1 the purpose of this thesis is to design a practical control scheme based on MPC in combination with a moving horizon estimator. Both the model predictive controller and the moving horizon estimator need an underlying model of the Demer basin in order to determine the optimal control actions and optimal state estimates, respectively. As already outlined in the first chapter, the water division of the VMM has a full hydrodynamic model of the Demer basin based on the De Saint-Venant equations. This model is very accurate and is used in order to predict future floods. In theory, MPC and MHE could use this model as underlying model. However, in practice MPC and MHE cannot make use of this model because of its computational complexity. As will be seen in chapter 3 in order to solve both optimization problems many simulations of the model are needed and with this model both optimization problems would not be solvable within the sampling time. Therefore, a new model is needed that is complex enough to model the important dynamics of the river basin, but at the same time also fast enough in order for the optimization problems to be solved within the sampling time. In this chapter such a model is derived.

This chapter first starts with some basic hydraulic concepts and definitions and an overview of the 3 basic conservation laws that are usually used in physics based hydraulic models. Also an overview will be given of the different model types that are used in practice for river modeling purposes. Then a conceptual model is derived based on the reservoir model type. This model type has never been used in river control applications. The model is calibrated and validated on historical data. Also an overview will be given of different hydraulic structures that one can find in river control applications, as well as the equations that one uses in order to model the discharges over/through these structures.

2.2 Hydraulic concepts

2.2.1 Definitions

The flow in water systems with the top open is called *open-channel flow*. The flow in water systems with the top closed is called *closed-conduit flow*. In the case where the conduit is flowing full, the flow is called *pressurized flow*. In this thesis only open-channel flow is considered as there are no important closed conduits in the Demer basin. It can therefore be assumed that the mass density ρ of the water in the river system is constant (incompressible flow) [49]. Based on some criteria, free-surface flow may be classified into various types of flow:

- **Steady and Unsteady flow [49]**: If the flow velocity at a given point in the flow does not change with respect to time then the flow is called *steady flow*, else the flow is called *unsteady flow*.

- **Uniform and Nonuniform flow [49]**: A flow whose velocity does not vary along the channel's length is called *uniform flow*. In a uniform flow the partial derivatives with respect to the x,y and z direction are all zero. In the other case the flow is called *nonuniform flow*.

- **Laminar and Turbulent flow [49]**: In *laminar flow*, the fluid glides along in layers, which do not mix so that the flow takes place in smooth continuous lines called streamlines. In *turbulent flow* the fluid moves in irregular paths. Whether the flow is laminar or turbulent depends on the viscous and inertial forces. In laminar flow the viscous forces dominate, in turbulent flow the inertial forces dominate. The ratio of viscous and inertial forces is defined as the *Reynolds number*:

$$\mathbf{R}_e = \frac{VL}{\nu} \tag{2.1}$$

with $\mathbf{R}_e$ = Reynolds number; V = mean flow velocity; L = a characteristic length; and ν = kinematic viscosity of the liquid. The transition from laminar to turbulent flow occurs at a $\mathbf{R}_e$ of 600.

- **Subcritical, Supercritical, and Critical flow [49]**: Critical flow occurs when the flow velocity in the channel equals the velocity of a gravity wave with small amplitude. If the flow velocity is less than the critical velocity, the flow is called *subcritical*. If the flow velocity is higher than the critical velocity, the flow is called *supercritical*. Whether a flow is subcritical, supercritical or critical depends on the Froude number $\mathbf{F}_r$. The Froude number is a dimensionless number that is equal to the ratio of the inertial and gravitational forces. Critical flow occurs for $\mathbf{F}_r = 1$, subcritical flow for $\mathbf{F}_r < 1$ and supercritical for $\mathbf{F}_r > 1$. The Froude number is defined as:

$$F_r = \frac{V}{gy} \tag{2.2}$$

  in which $y =$ flow depth.

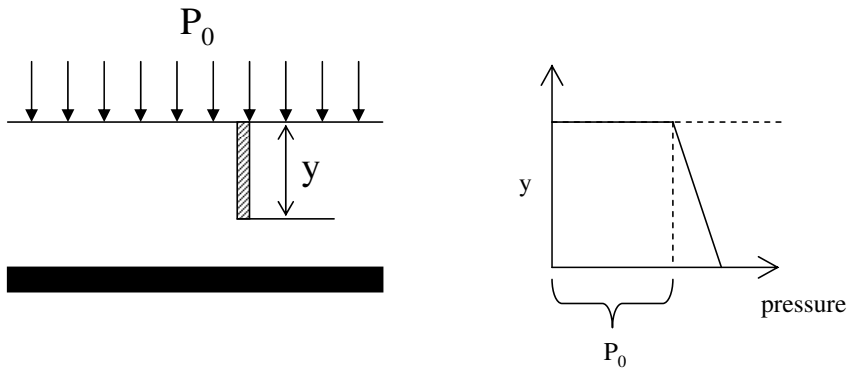


Figure 2.1: Pressure distribution in static flow. (image taken from [49])

In an open channel a differentiation can be made based on the shape of the channel. A channel having the same cross section and bottom slope throughout its length is referred to as a *prismatic channel*. A channel having a cross section and/or bottom slope that varies throughout its length is called a *non-prismatic channel*. The depth of flow, y, at a section is the vertical distance of the lowest point of the channel section from the free surface. The depth of flow, d, is the depth of flow normal to the direction of flow. The *top width*, B, is the width of the channel section at the free surface. The *flow area* A is the surface of the cross section normal to the direction of flow. The *wetted perimeter* P is the perimeter of the cross-sectional area normal to the direction of flow that is wetted by the flow. The *hydraulic radius* R and the *hydraulic depth* D are defined as

$$R = \frac{A}{P} \tag{2.3}$$

$$D = \frac{A}{B} \tag{2.4}$$

Due to the free surface and friction with the channel wall the flow velocity in a channel section varies from one point to another. The flow velocity typically has components in the x, y and z direction. If the coordinate axis is chosen such that the x direction lies according to the main flow direction then the components of the flow velocity according the y and z direction may be neglected. In order to facilitate calculations the mean velocity V_m is used for hydraulic calculations. In order to compensate for the error made by using V_m in the calculations correction coefficients are typically added. Streamlines are a family of curves that are instantaneously tangent to the velocity vector of the flow. These show the direction a fluid element will travel in at any point in time. The pressure distribution in a channel section depends on the flow conditions. In case the flow is *static* (zero flow velocity) it can be shown that the pressure acting on a point in the flow at depth y from the free surface (see Figure 2.1) is equal to $p = P_0 + \rho g y$ with P_0 the atmospheric pressure. If the channel is horizontal, the flow velocity parallel to the channel bottom, no acceleration is present in the direction of flow and the flow is uniform then the pressure distribution in a cross section still satisfies the pressure distribution of that of static flow. Therefore the distribution is called the *hydrostatic pressure distribution*. If the channel bottom has a small slope and the streamlines are parallel to the bottom the pressure distribution may still be assumed to be hydrostatic.

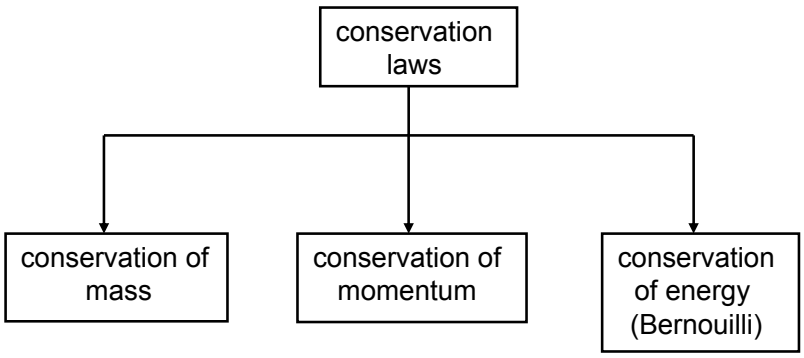


Figure 2.2: conservation laws

2.2.2 Conservation laws

Most models used in practice for modeling river dynamics are based on the 3 conservation laws depicted in Figure 2.2. Because of their importance in the following these conservation laws are discussed in more detail. [1]

[1] The theory outlined in this subsection comes from [50].

2.2.2.1 Conservation of mass

Civil engineers deal primarily with the flow of incompressible liquids in open channels, i.e., the mass density of the liquid is constant. Therefore, the law of conservation of mass between different channel cross sections implies that the volumetric flow rates at these sections are equal. Let us consider the flow of an incompressible liquid in a channel, as shown in Figure 2.3, with no inflow or outflow across the channel boundaries. Let the flow be steady. Let us denote the instantaneous flow velocity at a point by v, the flow depth by y, the mass density by ρ, the flow area by A, top watersurface width by B, and use subscripts 1 and 2 to designate quantities for sections 1 and 2 respectively. Then we may write

$$\text{Rate of mass inflow through area } dA_1 \text{ at section } 1 = \rho_1 v_1 dA_1. \tag{2.5}$$

$$\text{Rate of mass inflow through area } dA_2 \text{ at section } 2 = \rho_2 v_2 dA_2. \tag{2.6}$$

According to the law of conservation of mass, the rate of mass inflow at section 1 must equal the rate of mass outflow at section 2, since the volume of liquid stored in the channel between sections 1 and 2 remains unchanged, i.e.,

$$\int \rho_1 v_1 dA_1 = \int \rho_2 v_2 dA_2. \tag{2.7}$$

Since the liquid is assumed incompressible $\rho_1 = \rho_2$. Therefore,

$$\int v_1 dA_1 = \int v_2 dA_2. \tag{2.8}$$

If the flow velocity is assumed uniform at each section, then equation (2.8) may be written as

$$V_1 \int dA_1 = V_2 \int dA_2. \tag{2.9}$$

or

$$V_1 A_1 = V_2 A_2. \tag{2.10}$$

Note that equation (2.10) is valid for nonuniform velocity distribution provided V_1 and V_2 are the mean flow velocities at sections 1 and 2 respectively. In terms of volumetric flow rate, Q, this equation becomes

$$Q_1 = Q_2. \tag{2.11}$$

In hydraulic engineering, this equation is usually referred to as the *continuity equation for steady flow*. Note that for unsteady flow the mass continuity equations become more complex [123].

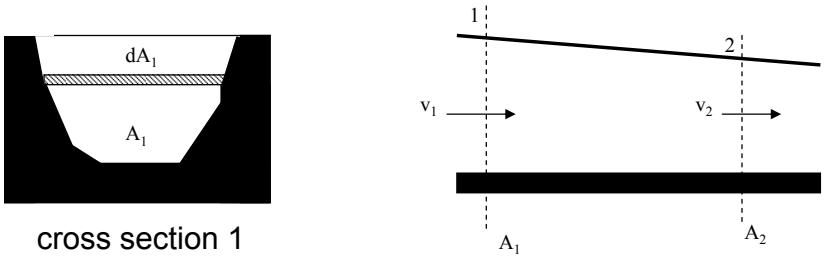


Figure 2.3: channel flow (image taken from [49])

2.2.2.2 Conservation of momentum

To derive an equation describing conservation of momentum, let us consider the steady state flow of an incompressible liquid in a channel, as shown in Figure 2.4 The channel is prismatic and there is no lateral inflow or outflow. Referring to this figure and using subscripts 1 and 2 to designate quantities for section 1 and 2

$$\text{Time rate of mass inflow at section } 1 = \frac{\gamma}{g} Q \tag{2.12}$$

in which γ = specific weight of liquid. If V_1 is the mean flow velocity at section 1, then

$$\text{Time rate of momentum inflow at section } 1 = \frac{\gamma}{g} \beta_1 Q V_1 \tag{2.13}$$

in which β_1 = momentum coefficient introduced to account for the nonuniform velocity distribution. Similarly, we can write for section 2 that

$$\text{Time rate of momentum outflow } = \frac{\gamma}{g} \beta_2 Q V_2. \tag{2.14}$$

Hence, it follows from equations (2.13) and (2.14) that for the liquid volume between sections 1 and 2

$$\text{Time rate of increase of momentum } = \frac{\gamma}{g} Q \left(\beta_2 V_2 - \beta_1 V_1\right). \tag{2.15}$$

The following forces are acting on the volume of liquid between sections 1 and 2.

$$\text{Pressure force at section 1, } P_1 = \gamma \bar{z_1} A_1 \tag{2.16}$$

$$\text{Pressure force at section 2, } P_2 = \gamma \bar{z_2} A_2 \tag{2.17}$$

in which $\bar{z}$ = depth of the centroid of flow area A. Component of the weight of liquid between sections 1 and 2 = $W \sin\theta$ in which W = weight of the volume of liquid

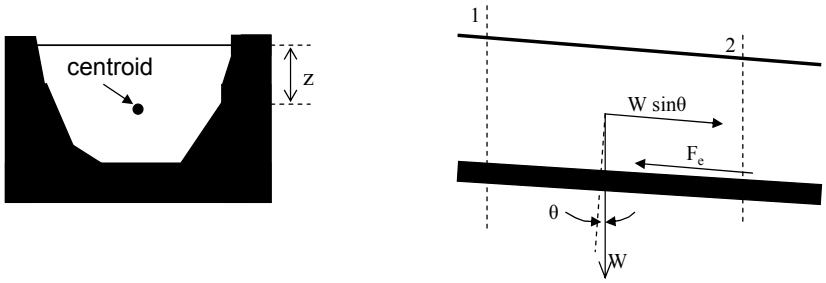


Figure 2.4: conservation of momentum

between sections 1 and 2, and θ = slope of the channel bottom. Note that this weight component is acting in the downstream direction. Let us neglect the shear stress at the free surface between air and liquid and let us designate the external force due to shearing force between the liquid and the channel sides by F_e. Then the resultant force, F_r, acting on the volume of liquid in the downstream direction is

$$F_r = \gamma A_1 \bar{z_1} - \gamma A_2 \bar{z_2} + W \sin\theta - F_e. \tag{2.18}$$

According to Newton's second law of motion, the time rate of change of momentum of the liquid volume is equal to the resultant of the external forces acting on the liquid volume. Hence, noting that $\gamma = \rho g$ we obtain from equation (2.15) and (2.18)

$$\rho\beta_2 Q_2 V_2 - \rho\beta_1 Q_1 V_1 = \rho g A_1 \bar{z_1} - \rho g A_2 \bar{z_2} + W \sin\theta - F_e. \tag{2.19}$$

Note that F_e is the external shearing force acting on the volume of liquid and does not depend upon the energy losses occurring inside the liquid segment. Equation (2.19) is a general application of the momentum principle. For special cases this equation can be simplified.

2.2.2.3 The Bernoulli equation

The Bernoulli equation is usually used to describe the flow through a structure. In order for the Bernoulli equation to be valid following assumptions must be satisfied:

- The mass density ρ of the fluid should be constant.
- Internal energy losses are negligible.
- The fluid should be in steady flow.

According to the Bernoulli equation the total energy H is given by:

$$z + \frac{p}{\gamma} + \frac{V_s^2}{2g} = H = constant \tag{2.20}$$

with z the height above an arbitrary datum, p the hydrostatic pressure and V_s the velocity of the fluid. The Bernoulli equation can be derived from the Euler's equation of motion [50].

2.3 Modeling techniques for water systems

In order to be able to design a model predictive controller the first thing that is needed is a model describing the dynamic behavior of the river system under consideration. Mathematical models are typically classified into black box, grey box or white box models. A white box model, also called physical model, is a model that is based on first principles (see Figure 2.5). In case of river control these first principles are conservation of mass, conservation of momentum and conservation of energy. In case it is too time consuming to derive a white box model a black box model can be derived based on process data. A black box model describes an empirical relationship between the process inputs and outputs. Most system identification algorithms are developed for this type of models [79]. A grey box model is a model that combines first principles with experimental data in order to describe the model behavior. The unknown free parameters can be estimated using system identification techniques from [79]. A grey box model is also called a semi-physical model. In the sequel these different model types will be discussed in detail.

2.3.1 Models based on finite-difference methods

Open-channel flows are described by three conservation laws, namely conservation of mass, conservation of momentum and conservation of energy. Application of these conservation laws to one-dimensional open-channel flow leads to the De Saint Venant equations. These equations form the basis for models based on finite-differences. In the following the De Saint Venant equations are described in more details and how they can be used to obtain very accurate finite-difference based models.

2.3.1.1 De Saint Venant equations

The De Saint Venant equations describe the continuity and momentum equations for one-dimensional open-channel flow [50]. In the derivation of the De Saint Venant equations following assumptions are made.

(i) The pressure distribution is hydrostatic. This is a valid assumption if the streamlines do not have sharp curvatures.

(ii) The channel bottom slope is small so that the flow depth measured normal to the channel bottom or measured vertically are approximately the same.

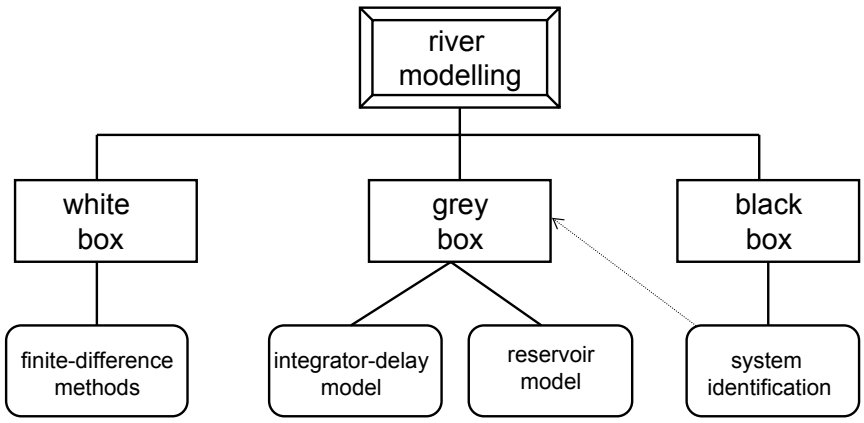


Figure 2.5: Overview of the different modeling techniques for open water systems

(iii) The flow velocity over the entire channel cross section is uniform.

(iv) The channel is prismatic, i.e., the channel cross section and the channel bottom slope do not change with distance. The variations in the cross section or bottom slope may be taken into consideration by approximating the channel into several prismatic reaches.

(v) The head losses (external shear stress acting on the boundaries of the control volume) in unsteady flow may be simulated by using the steady-state resistance laws, such as the Manning or Chezy equation, i.e., head losses for a given flow velocity during unsteady flow are the same as that during steady flow.

Taking these assumptions into account it can be derived that the continuity and momentum equations for one-dimensional open channel flow become respectively

$$\frac{\partial y}{\partial t} + D\frac{\partial V}{\partial x} + V\frac{\partial y}{\partial x} - \frac{q_l}{B} = 0 \tag{2.21}$$

and

$$\underbrace{\underbrace{\underbrace{S_f = S_0}_{Steady,uniform} \quad -\frac{\partial}{\partial x}\left(\frac{V^2}{2g} + y\right)}_{Steady,nonuniform} - \frac{1}{g}\frac{\partial V}{\partial t}}_{Unsteady,nonuniform} \tag{2.22}$$

with the hydraulic depth $D = A/B$, the channel bottom slope S_0 and the friction slope S_f. Note that for any exponential formula for friction losses, the friction slope may be written as

$$S_f = \frac{CV\,|V|^{m-1}}{R^p} \tag{2.23}$$

in which C and p are coefficients that depend upon the formula employed, R = hydraulic radius and m depends upon the flow type. For example, $m = 1$ for laminar flow; $m = 1.75$ for smooth turbulent flow; and $m = 2$ for fully rough turbulent flow. Equations (2.21) and (2.22) are *hyperbolic partial differential equations* and are called the De Saint Venant equations.

Remark 2.1. *If the streamlines in a flow have sharp curvatures, then the pressure distribution is not hydrostatic because of acceleration. In this case the De Saint Venant equations are not valid anymore and need to be replaced by the Boussinesq equations [50].*

2.3.1.2 Finite-difference methods

The unsteady flow in open channels is described by a set of hyperbolic partial differential equations, namely the De Saint Venant equations. These equations describe the conservation of mass and momentum in terms of the partial derivatives of dependent variables: flow velocity, V, and flow depth, y. However, for practical applications, we need to know the value of these variables instead of the values of their derivatives. Therefore, the governing equations need to be integrated. Because of the presence of nonlinear terms, a closed-form solution of these equations is not available, except for very simplified cases. Therefore, they are integrated numerically for which several numerical methods have been proposed [50]:

(i) Finite-difference methods;

(ii) Method of characteristics;

(iii) Finite-element method;

(iv) Finite-volume method;

(v) Spectral method;

(vi) Boundary-element method.

Of these methods, the finite-difference methods have been utilized very extensively. So we will focus on these methods below.

Finite-difference approximations

Basically, finite-difference methods approximate the (partial) derivatives of the De Saint Venant equations by finite differences. Consider a function $f(x)$ of one independent variable x. Assume the value of this function, $f(x_0)$, at x_0 is known. Then, by using a Taylor series expansion, the function $f(x_0 + \Delta x)$ may be written as

$$f(x_0 + \Delta x) = f(x_0) + \Delta x f'(x_0) + \frac{(\Delta x)^2}{2!} f''(x_0) + O\left((\Delta x)^3\right) \tag{2.24}$$

in which a prime $'$ refers to derivative with respect to x, e.g., $f'(x_0) = \frac{df}{dx}$ evaluated at $x = x_0$, and $O(\Delta x)^3$ indicates terms of third- or higher-order of Δx. Similarly $f(x_0 - \Delta x)$ may be expanded as

$$f(x_0 - \Delta x) = f(x_0) - \Delta x f'(x_0) + \frac{(\Delta x)^2}{2!} f''(x_0) + O\left((\Delta x)^3\right). \tag{2.25}$$

Equation (2.24) may be written as

$$f(x_0 + \Delta x) = f(x_0) + \Delta x f'(x_0) + O\left((\Delta x)^2\right). \tag{2.26}$$

It follows from equation (2.26) that

$$\left.\frac{df}{dx}\right|_{x=x_0} = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} + O(\Delta x). \tag{2.27}$$

Similarly, it follows from equation (2.25) that

$$\left.\frac{df}{dx}\right|_{x=x_0} = \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x} + O(\Delta x). \tag{2.28}$$

Neglecting the $O(\Delta x)$ terms in equations (2.27) and (2.28) leads to

$$\left.\frac{df}{dx}\right|_{x=x_0} \approx \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \tag{2.29}$$

and

$$\left.\frac{df}{dx}\right|_{x=x_0} \approx \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x}. \tag{2.30}$$

The finite-difference approximation of equation (2.29) is referred to as the forward finite-difference and that of equation (2.30) as the backward finite-difference. Note that in both of these cases, the discarded terms are of the first order of Δx. Therefore, both forward and backward finite-difference approximations are referred to as first-order

accurate. Subtracting equation (2.25) from equation (2.24), rearranging the terms and dividing by Δx, leads to

$$\left.\frac{df}{dx}\right|_{x=x_0} = \frac{f(x_0+\Delta x) - f(x_0-\Delta x)}{2\Delta x} + O(\Delta x)^2. \tag{2.31}$$

Neglecting the term $O(\Delta x)^2$ leads to

$$\left.\frac{df}{dx}\right|_{x=x_0} \approx \frac{f(x_0+\Delta x) - f(x_0-\Delta x)}{2\Delta x}. \tag{2.32}$$

This approximation is referred to as the central finite-difference approximation. Note that the neglected term in this case is of the order $(\Delta x)^2$; therefore it is referred to as second-order accurate. Note that the central finite-difference approximation is more accurate than the forward or backward finite-difference approximations.

In order to determine finite-difference approximations of partial derivatives, consider function $f(x,t)$ with the two independent variables: x and t. As shown in Figure 2.6 the $x-t$ plane is divided into a grid. The grid interval along the x-axis is Δx and the grid interval along the t-axis is Δt. Note that the grid size does not have to be uniform. For brevity, the $i\Delta x$ grid point will be referred to as i and the $(i+1)\Delta x$ grid point as $i+1$. Similarly, for the time axis the $k\Delta t$ grid point will be referred to as k and the $(k+1)\Delta t$ grid point as $k+1$. To refer to different variables at these grid points, the number of the spatial grid will be used a subscript and that of the time grid as a superscript. For example, the flow depth, y, at the ith spatial grid point and kth time grid point will be denoted by y_i^k. The known time level will be denote by superscript k and the unknown time level by $k+1$. The known time level is the time level for which the values of the different dependent variables are known. The unknown time level is the time level for which the values of the different dependent variables needs to be computed. The known conditions may be the specified values as the initial conditions or they may have been computed during the previous time steps. If the computations progress from one step to the next, then the procedure is referred to as a marching procedure. Most of the phenomenon described by hyperbolic partial differential equations are solved by using marching procedures. The conditions specified at time $t=0$ are referred to as the initial conditions. The conditions specified at the channel ends are called the end or boundary conditions.

There are several possibilities for approximating the partial derivatives. The spatial partial derivatives replaced in terms of the variables at the known time level are referred to as the explicit finite-differences whereas those in terms of the variables at the unknown time level are called implicit finite-differences. So, if k is the known time level and $k+1$ the unknown time level, then some typical finite-difference approximations for the spatial partial derivative, $= \frac{\partial f}{\partial x}$, at the grid point (i,k) are as follows:

Explicit finite-differences

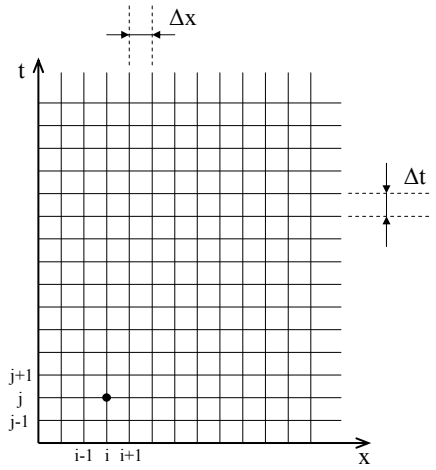


Figure 2.6: 2-dimensional grid (image taken from [49])

$$\text{Backward: } \frac{\partial f}{\partial x} = \frac{f_i^k - f_{i-1}^k}{\Delta x} \tag{2.33}$$

$$\text{Forward: } \frac{\partial f}{\partial x} = \frac{f_{i+1}^k - f_i^k}{\Delta x} \tag{2.34}$$

$$\text{Central: } \frac{\partial f}{\partial x} = \frac{f_{i+1}^k - f_{i-1}^k}{2\Delta x} \tag{2.35}$$

Implicit finite-differences

$$\text{Backward: } \frac{\partial f}{\partial x} = \frac{f_i^{k+1} - f_{i-1}^{k+1}}{\Delta x} \tag{2.36}$$

$$\text{Forward: } \frac{\partial f}{\partial x} = \frac{f_{i+1}^{k+1} - f_i^{k+1}}{\Delta x} \tag{2.37}$$

$$\text{Central: } \frac{\partial f}{\partial x} = \frac{f_{i+1}^{k+1} - f_{i-1}^{k+1}}{2\Delta x} \tag{2.38}$$

Many other finite-difference approximations are possible if the values at three or more grid points, instead of just two grid points, are used. The most extensively used finite-difference scheme is the Preissmann scheme [50]. This scheme is an implicit finite difference scheme using the value of four grid points in order to obtain the finite-difference approximation of the De Saint Venant equations (2.21) and (2.22). For a detailed description of this finite difference scheme we refer the reader to [50].

2.3.1.3 Finite-difference based models in control applications

Finite-difference based models have been used in practice in order to obtain setpoint regulation [17, 82, 98]. This is achieved by first linearizing the De Saint Venant equations around the desired setpoint. Then, the canal is divided in nodes and branches.

For each branch the differentials are approximated by finite differences. This leads to a discrete-time linear model that can be used for control purposes. If the number of nodes is chosen sufficiently high the approximated model can be very accurate for small variations of its inputs around the steady-state value. In most cases this model will be too complicated for controller design. It will usually be necessary to reduce the model order using model reduction techniques [14].

2.3.2 System identification

System identification deals with the problem of building mathematical models of dynamical systems based on observed input-output data of the system [79]. In a system identification approach one can always distinguish three important entities:

- **data set:** The observed input-output data can in some cases be recorded from a specifically designed identification experiment where the user has the flexibility to carefully choose the input signals in order for the data to be maximally informative. In other cases, the user may not have the flexibility to affect the experiment and needs to use less informative data.

- **model structure:** A set of candidate models is obtained by specifying from which collection of models we are going to look for a suitable one. The models can be grey box models based on physical laws but with some unknown parameters, or the models can be black box models in which no physical laws are taken into consideration.

- **determination of the best model:** From the set of all candidate models, the best model needs to be chosen. This assessment of model quality is typically based on how the models perform when they attempt to reproduce the data. Typically, one determines the optimal parameters for the model based on so called test data. In order to avoid overfitting of the model to the observed data set an independent validation data set is used in order to verify the model.

System identification techniques have been used in several river control applications. In [99] different model structures like OE, ARX, ARMAX, and BOX-JENKINS were tested based on experimental data in order to determine the one that best represents the dynamics of the controlled canal pool. In [57], [92] and [122] linear and nonlinear grey box models were proposed. The parameters of these models were identified with system identification techniques and then used for control design. In [11, 113] ANN (artifical neural networks) were used for modeling and controlling irrigation canals. Note that all these works are restricted to setpoint regulation.

2.3.3 Integrator-delay model

Typically a finite-difference model will be too complicated for controller design as its model order is usually very high. Especially in control applications based on

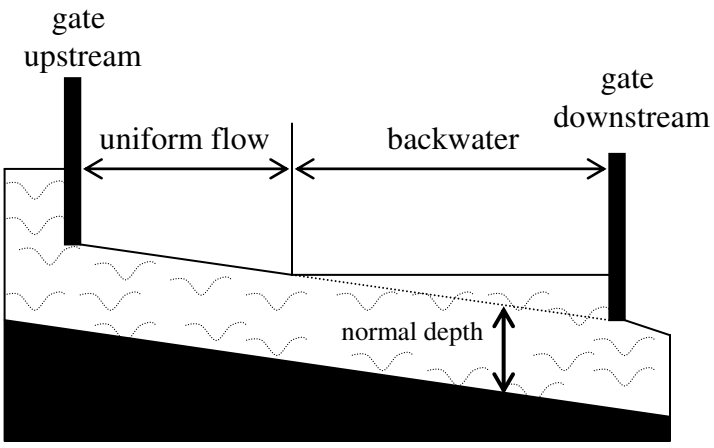


Figure 2.7: Example of backwater curve (image taken from [103])

model predictive control such models cannot be used because of the long computation times needed to simulate them. Therefore, in river regulation applications approximate models are used for controller design and simulation purposes. The most popular approximate model used for river regulation purposes is the integrator-delay model [104]. This type of model is used for regulation purposes where the main goal of the model predictive controller is to steer the water levels to a desired steady state (setpoint regulation). The idea behind the integrator-delay model is that the most common flow profile in controlled open-channels is the *backwater curve* [51]. In Figure 2.7 such a profile is depicted. The depth at the downstream end is above normal depth; the depth increases in the upstream direction until it reaches the upstream end (a control structure) or until the depth of flow equals the normal depth. As depicted in Figure 2.7 two parts can be observed: a part where the flow is uniform and a part that is effected by backwater. As pointed out in [103] many open channels rarely show uniform flow because the depth of flow usually changes over the distance due to geometrical irregularities of the channel wall, backwater effects, etc. However, for practical purposes flow may be designated uniform if the depth of flow varies within a few percentages of the normal depth [51].

In [103] approximate models were derived for the uniform part as well as for the backwater part and both connected in the following way:

$$h(x,s) = \begin{cases} -\frac{\beta_0}{\gamma_0} e^{-T(x)s} q(0,s) & \forall x \in (0, L_u) \\ \frac{1}{As}\left(e^{-T(L_u)s} q(0,s) - q(L,s)\right) & \forall x \in (L_u, L) \end{cases} \tag{2.39}$$

with L_u the length of the part with uniform flow , $T(x)$ and $T(L_u)$ denoting time-delays in the system, A the surface of the backwater part and β_0 and γ_0 two constant parameters. For the specific characterization of the time-delays $T(x)$, $T(L_u)$ and the parameters β_0 and γ_0 we refer to the work in [103]. Note that $q(x,s)$ denotes the Laplace transform of the discharge variable $q(x,t)$ with $q(x,t)$ the discharge at location x and time instant t. Therefore, $q(0,s)$ and $q(L,s)$ denote the Laplace transforms of the discharges $q(0,t)$ and $q(L,t)$ and can be considered as boundary conditions of the considered reach. In Figure 2.8 the most important variables for the integrator-delay model are depicted. Integrator-delay models have been used extensively for setpoint regulation of rivers [29, 43, 53, 74–76, 78, 93, 105, 117, 119].

Remark 2.2. *Though the integrator-delay model has been used successfully in many river control applications in [44] it was pointed out that combining MPC with an integrator-delay model for achieving setpoint regulation in some occasions can result in worse performance than the use of more traditional control techniques due to the simplified model structure.*

2.3.4 Reservoir model

Though integrator-delay models have been used successfully for water level control applications, their use is limited to setpoint regulation. The reason for this is that integrator-delay models are only accurate enough in a small region around the steady-state for which they were calibrated. During periods of heavy rainfall it is impossible to keep the river system in this region. Therefore, integrator-delay models are not suitable for flood regulation purposes. At the other hand, models based on finite-difference methods are accurate enough to capture the river dynamics during heavy rainfall but they are not suited for real-time control due to their excessive computational load. A better option for flood regulation is the use of reservoir model types. These models are more accurate than the integrator-delay models and less time consuming than the models based on finite-differences.

The first step consists of lumping the processes in space. Lumping of the processes in space only happens at relevant locations, which is one of the reasons why the computational load of these models is less than that of models based on finite-differences. These relevant locations are the locations up- and downstream of the hydraulic regulation structures as well as the locations along the river system where potential flooding is induced. Depending on these locations, the river is subdivided in reaches, in which water continuity is modeled (in a spatially lumped way per reach) based on reservoir type of models. A reservoir simply assumes conservation of mass (increase in volume per time step equals inflow minus outflow). The inflow in each

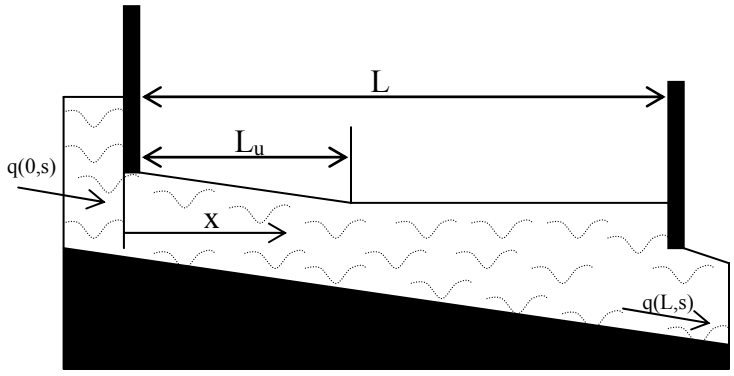


Figure 2.8: Variables for integrator-delay model (image taken from [103])

reservoir (submodel representing a river reach) is the discharge from the more upstream reach (result of the more upstream submodel). The output depends on the water storage in the reach or is assumed equal to the sum of the upstream discharge and the other inflows along the reach (e.g. from tributary rivers). In each reservoir it is assumed there is a one-to-one relationship between the volume and the water level in the reservoir. This relationship depends on the geometry of the channel reach and is nonlinear. This nonlinear relationship is determined based on simulation data.

Reservoir type models have already been used in order to simulate sewer and river networks with reduced calculation time [124], [115]. However, this model type has never been used for control purposes. In this thesis a reservoir model is calibrated and used by the model predictive controller for determining the (sub)optimal gate positions. In the remainder of this chapter a more detailed description is given of the calibrated reservoir model as well as the calibration process. The validity of this model is then justified by a validation process based on historical simulation data. This calibration and validation work has been done as part of an ongoing PhD research by Po-Kuan Chang at the Hydraulics Laboratory of the K.U.Leuven, supervised by Prof. P. Willems & Prof. J. Berlamont. In the following sections, a summary of this work is described.

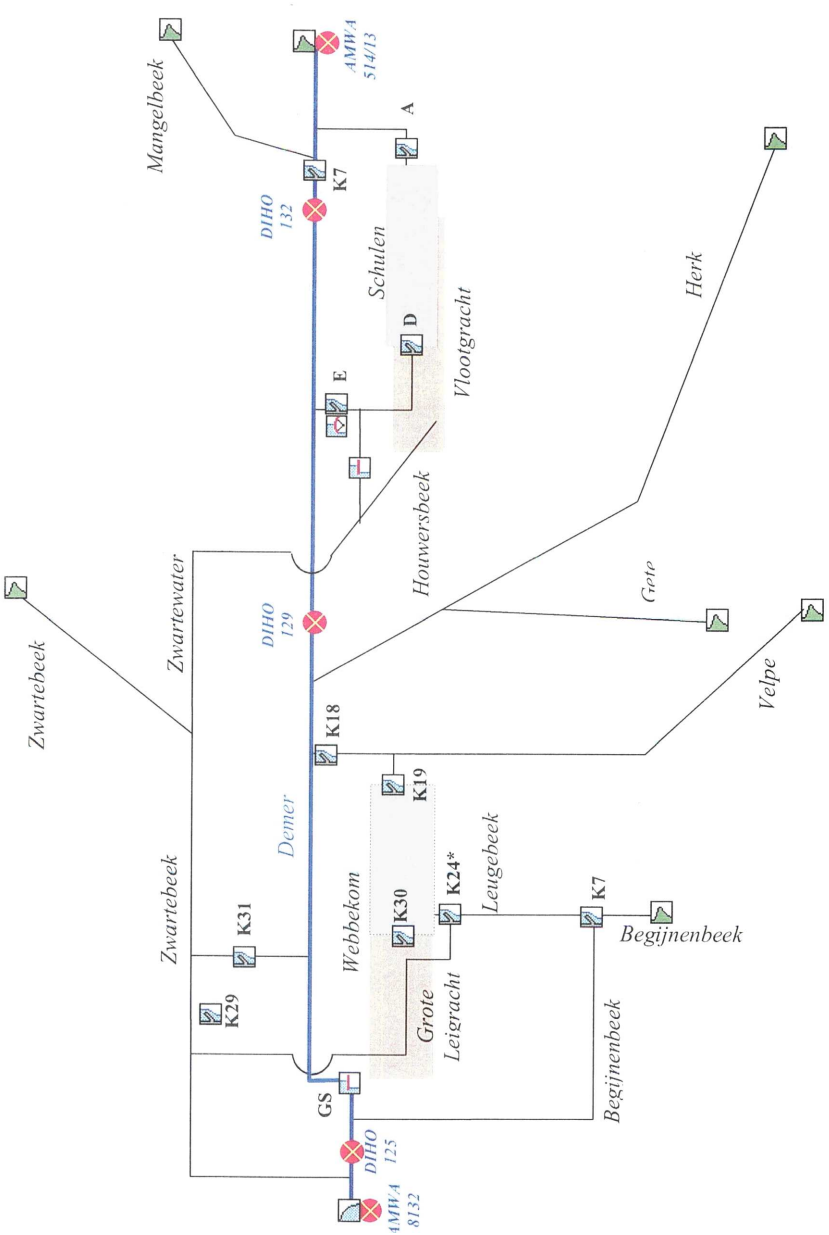


Figure 2.9: Scheme of the river network in the study area, the flood control reservoirs “Webbekom” and “Schulensmeer” at the village Schulen and the hydraulic regulating structures.

2.4 Conceptual model

In this section the conceptual model used in this thesis is outlined in more detail. Subsections 2.4.1 and 2.4.2 are a slightly modified copy of the modeling part outlined in [25]. Note that the work outlined in both subsections is not the work of the author of this thesis but part of an ongoing PhD research by Po-Kuan Chang at the Hydraulics Laboratory of K.U.Leuven.

2.4.1 Conceptual model structure

Figure 2.10 shows the scheme of the model components for the study area around the two flood control reservoirs 'Schulensmeer' and 'Webbekom' in the river Demer basin. This area receives rainfall-runoff inflow via the tributary rivers Mangelbeek, Herk, Gete, Velpe, Zwartebeek, Zwartewater, and Begijnenbeek. By means of the hydraulic regulating structures A and K7, the local water engineers can anticipate on future flood risks. Through closing gate K7 and opening gate A, the Schulensmeer reservoir is being filled, the downstream Demer flow reduced, and consequently the flood risk of the cities Diest and Aarschot downstream of the study area of the reservoirs reduced. After the flood period, the Schulensmeer reservoir (which consists of different reservoir compartments) can be emptied through the hydraulic regulating structures D and E. The second reservoir 'Webbekom' is regulated in a similar way by means of the hydraulic structures K18, K19, K7 at the Leugebeek river, K24* and K30. Figure 2.9 gives an overview of the structure of the conceptual model developed by the Hydraulics Laboratory of K.U.Leuven for the study area. In this scheme the river reaches are represented by means of lines with positive flow in the direction of the arrows, the hydraulic regulating structures by means of the full rectangles, the fixed spills or overflows by open rectangles, and the model units where water storage (in the reservoir compartments or along river reaches) and water levels are simulated by nodes. The symbol 'q' denotes discharges, 'h' water levels, 'v' storage volumes, and 'k' controllable gate crest levels. The water levels and volumes are the model variables describing the state of the water system in the model predictive controller. The gate crest levels are in the inputs in the model predictive controller, the upstream (rainfall-runoff) discharges the disturbances of the model predictive controller.

In order to do some initial tests with the MPC controller, in the first phase of the thesis a separate model has been developed for a 'reduced area' around the Schulensmeer reservoir. The scheme of this model is depicted in Figure 2.11. Figure 2.12 shows photos of the region upstream of the Schulensmeer reservoir and shows some upstream model components and model variables.

2.4.2 Conceptual model building and calibration process

The conceptual model-structure has been identified and its parameters calibrated based on simulation results with the full hydrodynamic InfoWorks model. The reason to calibrate based on simulations results rather than on real measurement data is

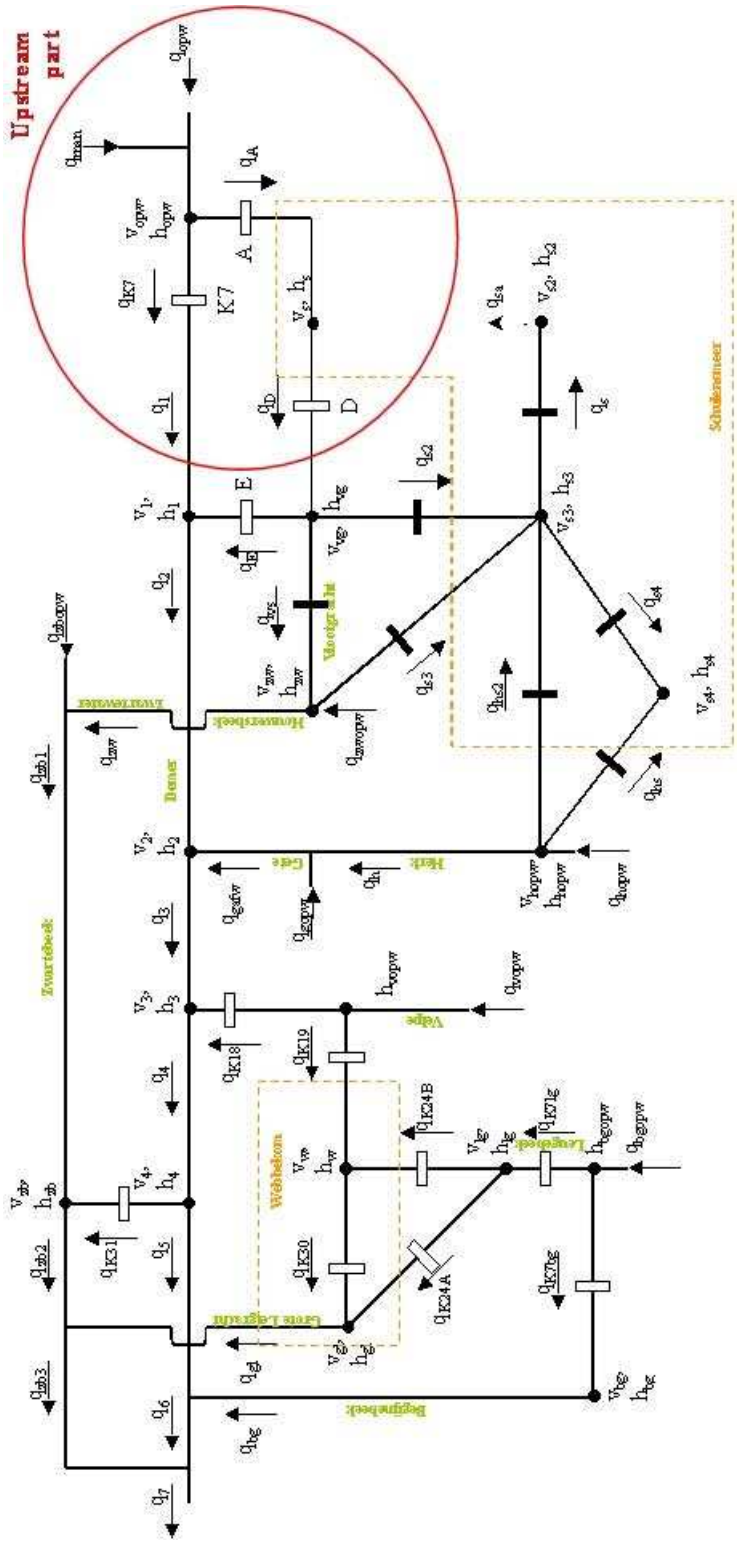


Figure 2.10: Scheme of the conceptual model for the study area (dots for the calculation nodes, i.e., river or reservoir storage elements; lines for the river reaches; open rectangles for the hydraulic regulating structures; closed rectangles for the fixed spills or weirs).

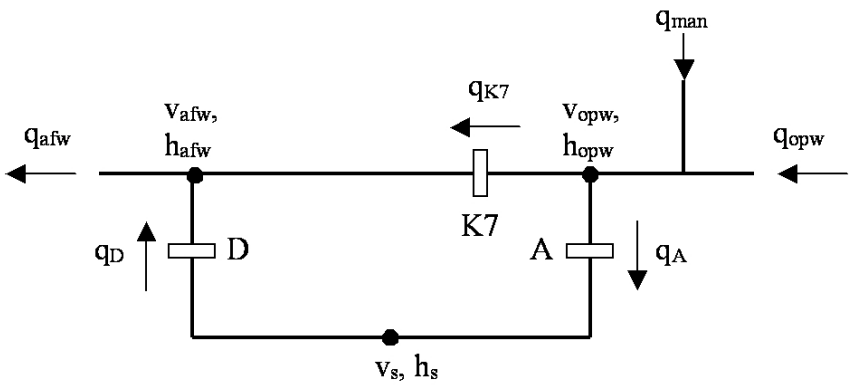


Figure 2.11: Schematic representation of the 'Schulensmeer' reduced area, together with the conceptual model variables.



Figure 2.12: Photo of the river Demer and the 'Schulensmeer' flood control reservoir in the background, together with the locations of the main water level and discharge variables.

because in practice only a small amount of states are measured which complicates the calibration process. Moreover, the measured data is sometimes incomplete. By using simulation results, data is available for all states in the system and therefore a better calibration process can be performed. The fact that the conceptual model is based on simulation results of the InfoWorks model will not influence the final conclusions of this thesis regarding whether MPC performs better than the three-position controller w. r. t. flood regulation. However, if the local water administration plans to use the conceptual model for real-time control of the Demer river, the calibration process has to be redone based on measurement data. The conceptual model calibrated in this thesis can be used as a starting point for this new calibration process.

In order calibrate the conceptual model on simulation results, different modeling concepts have been applied. The flood control reservoirs have been modeled as storage nodes (vs, vs2, vs3, vs4, and vw) that maintain water continuity. Also the large water storage areas, which in the conceptual model are modeled in a combined way with their drainage canals, are schematized as storage nodes. This is the case for the nodes 'vvg' (draining the 'Schulensmeer' reservoir) and 'vgl' (draining the 'Webbekom' reservoir). For each of these storage nodes, a nonlinear relationship has been determined between the storage volume and the mean water level (which is assumed static at each time step). These relationships have been calibrated based on the Digital Elevation Model underlying the InfoWorks model. In Figure 2.13 such a calibration result is shown for the Webbekom flood control reservoir. The water levels are shown as levels above the 'TAW' standard reference level for Belgium.

The static level assumption is only valid for the flood control reservoirs (thus for the water levels hs, hs2, hs3, hs4 and hw), but not for the water levels 'hvg' and 'hgl' of the drainage areas. These water levels indeed vary a lot along the course of the drainage canals. The water level considered in the conceptual model then is taken at one specific location along the drainage canal. Water levels at other locations along the drainage canal are only included in the conceptual model when they are really needed (e.g., because they control hydraulic structures). They are assumed related to the primary levels considered ('hvg' and 'hgl' in this case). In case no unique relationship exists with these primary levels, the drainage canal (together with the flood zones on its banks) is modeled as a long regular river reach, as described next.

Long regular river reaches can be modeled in two different ways: (a) by a serial connection of reservoirs, and (b) by the water surface profile concept (considering the water level differences from down- to upstream along the river). In approach (a), water levels are modeled from up- to downstream, while in approach (b) one starts from the most downstream locations along the reach and considers water level changes to upstream. Approach (a) guarantees water continuity, but with this approach it is possible that more downstream reservoirs in the serial connection do not necessarily have lower water levels. Outflow corrections are then needed to avoid this type of anomalies. With approach (b) this problem is circumvented as water level differences are always taken positive from down- to upstream.

In approach (b) the water level differences are depending on the discharge in the reach and the downstream water level. The following relation was assumed:

$$h = h_{afw} + \ a\ (q^2 / \ (h_{afw} \text{ - } h_{afw,0})^2)^b \tag{2.40}$$

where $h - h_{afw}$ is the water level difference to be modeled depending on the discharge q and the downstream water level h_{afw},where a and b are two parameters to be identified and where $h_{afw,0}$ is the bed level at the downstream location. This equation is based on the assumption of the equation of Manning [51]:

$$q = \frac{1.49}{n}\ A\ R^{\frac{2}{3}}\ S^{\frac{1}{2}} \tag{2.41}$$

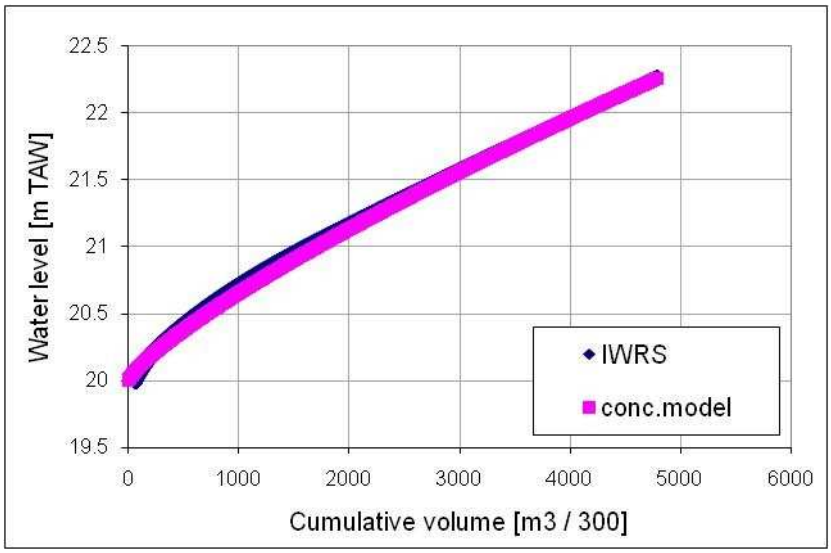


Figure 2.13: Calibration result for the storage volume - mean water level relation of the 'Webbekom' flood control reservoir.

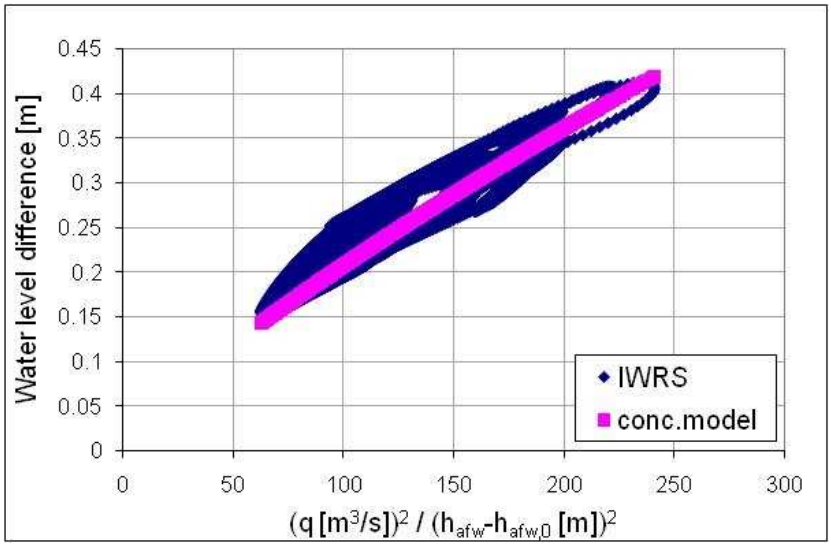


Figure 2.14: Calibration result applying approach (b) for the Demer river reach between the conceptual storage nodes h2 and h3.

where the discharge q depends on cross-section area A, the hydraulic radius R, the friction slope S and the Manning coefficient n. Under the uniform flow approximation, the friction slope S equals the river bed slope, which under the normal depth assumption furthermore equals the water surface slope $h - h_{afw}$. For wide river sections, the hydraulic radius approximately equals the river bed width and thus becomes independent on the water level. For rectangular sections, the cross-section area becomes linearly proportional to the water depth. Under these assumptions, the water level difference is proportional to the ratio of the squared discharge and the squared water depth:

$$q^2 / \left(h_{afw} - h_{afw,0}\right)^2 \tag{2.42}$$

which leads to equation (2.40). Figure 2.14 gives an example of the calibration of equation (2.40) to model the water level differences between h2 en h3 along the Demer. Figure 2.15 shows another example of the same approach but for the Grote Leigracht river (water level differences between hgl en hzb).

Approach (a), on the other hand, makes use of reservoir models, where the outflow discharge (flow to downstream) depends on the reservoir storage volume and on the upstream inflow discharges. The relationships between outflow discharge and storage volume and inflow discharge are calibrated based on simulation results with the full hydrodynamic model. These relationships can be linear, but they are more often of nonlinear nature. However, in this study the nonlinear relation was approximated by a set of piece-wise affine relations. The reservoir-model relations were identified in the following step-wise approach. First the relationship between water storage in the river reach and the outflow discharge is studied based on the IW results. When this

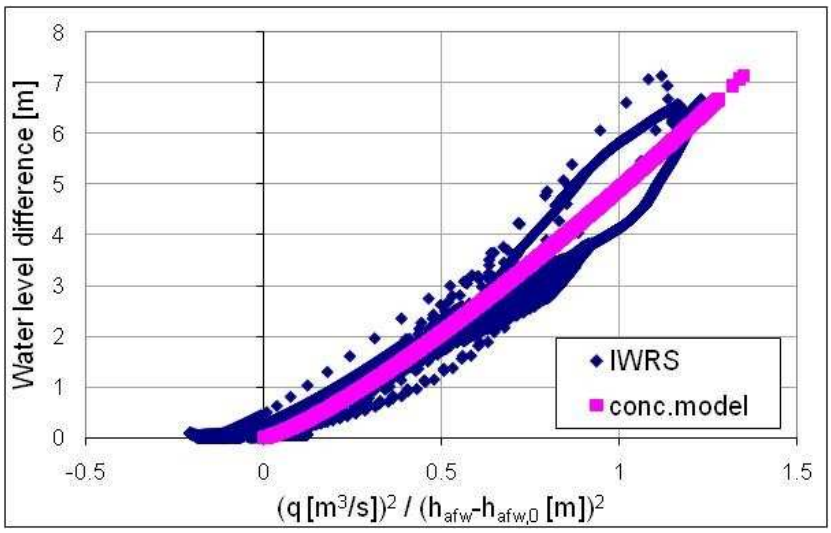


Figure 2.15: Calibration result applying approach (b) for the Grote Leigracht river between the conceptual storage nodes hgl and hzb.

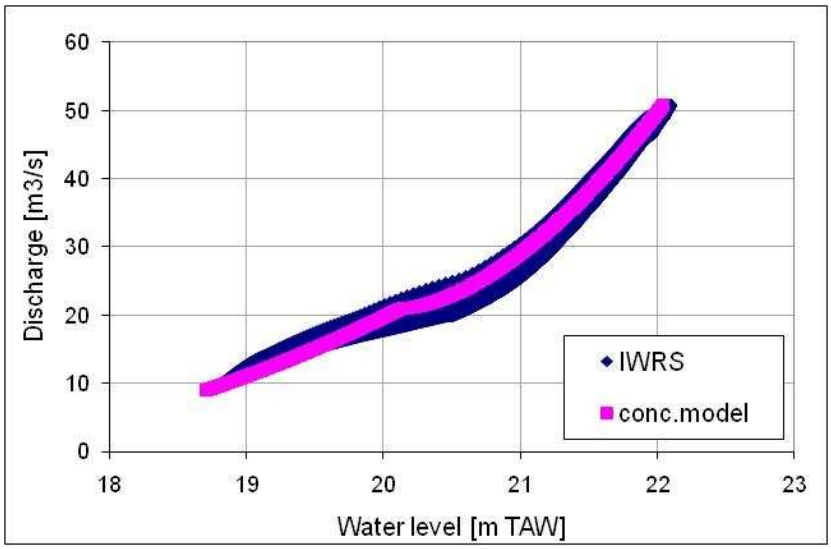


Figure 2.16: Calibration result applying approach (a) for the Demer storage node describing h3. A unique relation is shown between the water level h3, which depends on storage volume v3, and the outflow discharge (q3) of this storage.

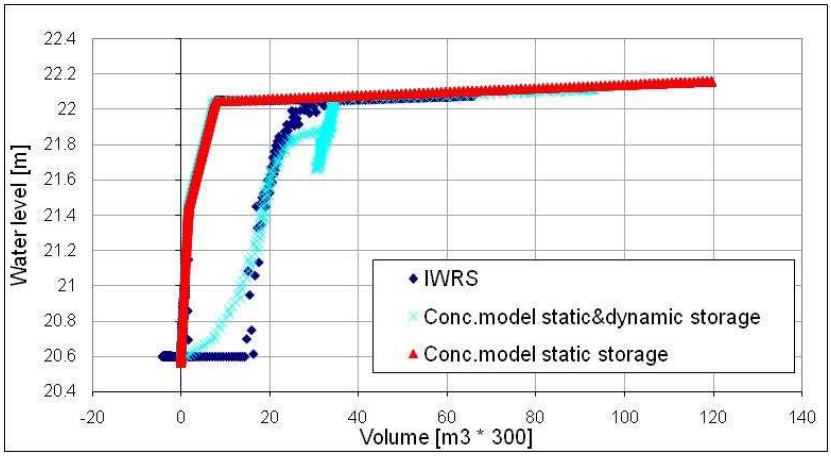


Figure 2.17: Calibration result applying approach (a) for the Leugebeek describing hlg. The hysteresis effect in the storage-volume relation has been modeled separating the total storage in static and dynamic storages.

relation is unique, a linear, non-linear, piece-wise affine or non-linear function was calibrated to that relation. Figure 2.15 shows for the same Demer reach as in Figure 2.13 the relation between the water level h3 (depending on the storage volume v3) and the downstream discharge. A unique relation was found which means the reservoir concept is applicable, thus both approach (a) and approach (b) could be implemented for this case. In Figure 2.15, the relation is not linear but a piecewise relation could be calibrated: a linear relation up to water level 20.2 m TAW, and a power relation for the higher water levels.

When the relation is not unique, but shows 'hysteresis effects' (see example in Figure 2.17), a more complex calibration method was needed. As shown in [115] based on hydrodynamic sewer system applications, hysteresis effects are to be explained by different storage - outflow relationships in the decreasing and increasing flanks of the flow hydrographs. These differences are to be explained by the differences in 'dynamic storage' in the system, and thus by the inflow discharges. In this case, the reservoir model is advanced separating the total storage in the river reach in static and dynamic storage parts. The static storage is identified as the lowest storage for a given outflow discharge (thus during the decreasing flanks of the flow hydrographs), while the dynamic storage is the difference between the total storage in the reach and the static storage identified. By plotting this dynamic storage at each time step versus the inflow discharge in the reach, and calibrating the identified relation by means of a linear, non-linear, piece-wise affine or non-linear model, a first reservoir submodel is obtained: the submodel to predict at each time step the dynamic storage based on the inflow discharge. When this dynamic storage is subtracted from the total storage (at each time step to be calculated considering water continuity) static storage predictions are obtained. Analyzing the relationship between this static storage and the outflow discharge, and fitting a function to this relation, completes the reservoir model-structure.

The 'hysteresis' fenomenon was seen for reaches of the 'Leugebeek' river (node 'lg') (see Figure 2.17). This is a river reach where part of the reach has water levels that are upstream controlled (steep hydraulic slope) and another part downstream controlled (mild hydraulic slope).

Both approaches (a) and (b) model the water levels in a 'lumped' way, which means that only 1 water level (at one single location along the reach) is considered. When water levels at more locations are required along the reach relations are determined between the levels at the new locations and the primary modeled water level. Because of its stability and mathematical elegance, preference has been given to approach (b).

In the conceptual model, model equations are solved based on finite differences between successive time steps. Multiple iterations per time step are avoided in order to have the model calculation times as limited as possible. A time step of 5 minutes is considered, which is coarse in comparison with the spatial resolution of the model (average distance between the calculation nodes). For this reason, the first version of the conceptual model was strongly affected by instabilities. Flow delay terms therefore

have been added to the model, making use of the linear reservoir model equation:

$$q_{out}(t) = \exp(-\frac{1}{k})\ q_{out}(t-1) + \ (1 - \exp(-\frac{1}{k})\ q_{in} \tag{2.43}$$

where q_{in} is the original flow (before delay) and q_{out} the flow after application of the delay equation.

2.4.3 Modeling of hydraulic structures

River systems are provided with structures in order to control the discharges and the water levels. There are many types of structures like e.g. pumps, lift stations, hydropower stations, culverts, orifices, weirs, and gates. The structures of major importance for flood regulation are the weirs and hydraulic structures with movable gates as these structures can be influenced directly by the controller. The control can be based on overflow or underflow. Hereafter, the control structures based on underflow will be called "vertical sluices" and the ones based on overflow "gated weirs". Both type of structures are present in the Demer basin and are therefore described in more detail in this subsection.

2.4.3.1 Underflow-vertical sluices

Gates are openings in hydraulic structures to permit the passage of water, usually in order to regulate the outflow. In this thesis the considered gates are of the sluice gate type. In Figure 2.18 sluice gates under different flow conditions, also called modes, are depicted. In Figure 2.18(a) the discharge through the gate is not completely under water and is therefore considered under free flow. In Figure 2.18(b) the discharge through the gate is under water and is therefore under submerged flow. Water level h_1 corresponds to the water level producing the discharge and is called the *head*. Water level h_2 is chosen at the *vena contracta*. At the *vena contracta* the streamlines are parallel [40]. U_g is the gate opening and can be controlled by the controller. The constant μ is a contraction coefficient. The discharge through the sluice gate can be determined by means of the Bernouilli equation. In [103] it was shown that this leads to the following equation for the discharge relation of a sluice gate:

$$Q = B_2 \mu U_g \sqrt{2g(h_1 - h_2)} \tag{2.44}$$

with B_2 the width of the cross section at water level h_2. For free flow the water level h_2 can be replaced by μU_g. In practice the observed discharge differs from equation (2.44) because of the simplified assumptions needed for the Bernouilli equation to be valid. Furthermore, the weir geometry, roughness and inflow conditions also affect the discharge characteristics [49]. Therefore, in practice a discharge coefficient C_D is added to equation (2.44):

$$Q = C_D B_2 \mu U_g \sqrt{2g(h_1 - h_2)}. \tag{2.45}$$

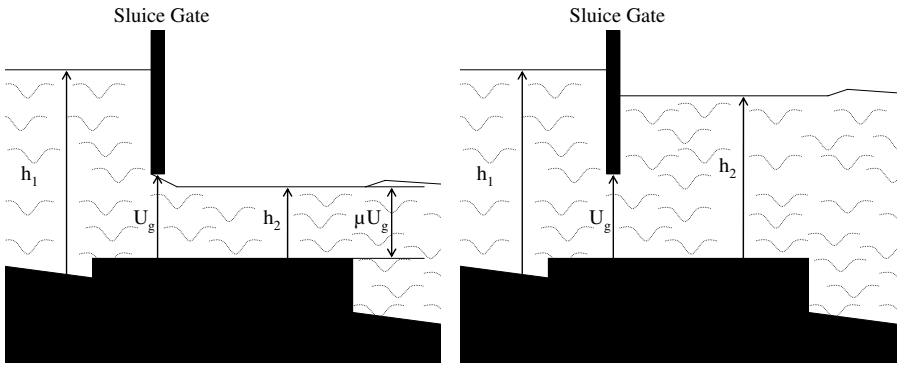
Note that the discharge coefficient C_D can be different for free flow conditions than for submerged flow conditions. The simulation data used to calibrate the conceptual

model is generated in Infoworks (full hydrodynamic model) by means of a similar type of equations. Therefore the value of C_d in the conceptual model is taken from the Infoworks model.

2.4.3.2 Overflow-gated weirs

A weir is a notch of regular form over which water flows. The term is also applied to the structure containing such a notch. Thus a weir may be a depression in the side of a tank, reservoir or channel, or it may be an overflow dam or other similar structure. The edge or surface over which the water flows is called the *crest*. The overflowing sheet of water is termed the nappe. A weir with a sharp upstream corner, or edge, is called a *sharp-crested* weir. Sharp-crested weirs are usually only used for measuring flowing water. All other weirs are classed as not sharp-crested. One type of not sharp-crested weirs of major importance in many hydraulic applications is the *broad-crested* weir. A broad-crested weir is a weir with a (nearly) horizontal crest sufficiently long in the direction of flow so that the nappe will be supported and hydrostatic pressures will be fully developed for at least a short distance. In Figure 2.19 a broad-crested weir is depicted. The water level upstream, height of the gate and the critical depth are denoted as respectively h_1, U_g and d_c. Note that if the nappe discharges into the air the weir is under free flow. If the discharge is partially under water the weir is under submerged flow.

The equation of the discharge over a broad-crested weir cannot be derived exactly. Most derivations found in literature approximate the real discharge over the crest. These derivations only take the effects of gravitational forces into account. Effects such as viscosity, surface tension, shape of the crest, ... are not included in the derivations. The approximate derivations are based on the Bernouilli equation. In [103] it was



(a) Free flow (b) Submerged flow

Figure 2.18: Sluice gates (images taken from [103])

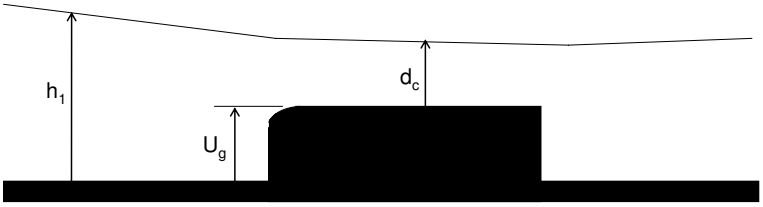


Figure 2.19: Flow over broad-crested weir (image taken from [103])

shown that after applying the Bernouilli equations following discharge relation can be derived for a broad-crested weir:

$$Q = g^{\frac{1}{2}} B_c \left(\frac{2}{3}(h_1 - U_g) \right)^{\frac{3}{2}} \tag{2.46}$$

with B_c the width of the cross section at the section where critical flow occurs. However, in a practical setting most weirs are short crested and equation (2.46) is not valid. For those structures a discharge coefficient C_D is added to equation (2.46) in order to compensate for the errors made by simplifying the hydraulics [103]:

$$Q = C_D g^{\frac{1}{2}} B_c \left(\frac{2}{3}(h_1 - U_g) \right)^{\frac{3}{2}}. \tag{2.47}$$

If the weir is submerged the water level downstream of the weir is so high that critical flow does not occur anymore above the weir. This leads to a reduction of the flow rate. In this case equation (2.47) is often applied but with a discharge coefficient different from the one used for free flow. Similar to the vertical sluices, the values of the discharge coefficients in the conceptual model were taken from the Infoworks model.

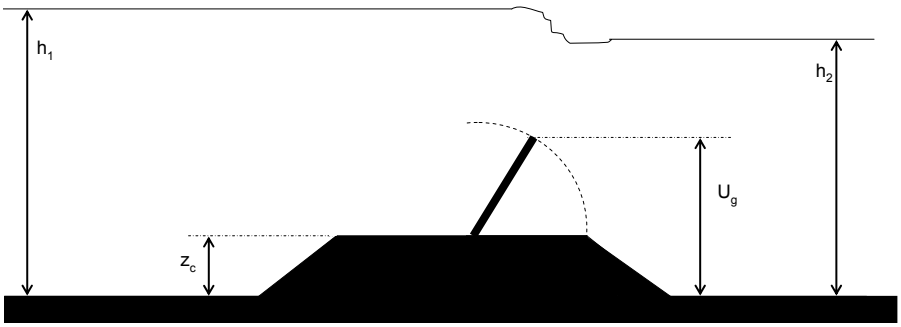


Figure 2.20: gated weir - gate control

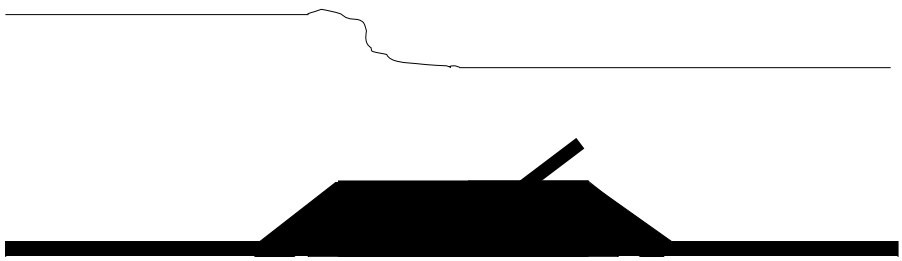


Figure 2.21: gated weir - throat control

A special weir of importance in this thesis is the gated weir. Most of the gates in the studied area of the Demer basin are of this type. The gated weir is depicted in Figure 2.20. A gated weir is a weir that rotates around its support with the channel bottom. Depending on the rotation angle of the weir, the discharge passing over the weir can be influenced. There are 2 important modes that can be distinguished. The first mode is depicted in Figure 2.20. This mode is called gate control and is a mode in which the discharge over the gate is determined directly by the gate itself. The flatter the position of the gate, the bigger discharge over the gate. However, if the gate is positioned at a very sharp angle the gate has no influence anymore over the discharge. This coincides with the second mode which is called throat control and is depicted in Figure 2.21. In this mode the discharge is influenced by the bottom elevation z_c. In the gate control mode the equations of the broad-crested weir can be used with some minor modifications. In the throat control mode these same equations can be used but with U_g replaced by z_c, again with some minor modifications. The exact equations for the throat control mode are

$$Q = 0.5 C_D B_c \sqrt{g} \left(\frac{2}{3} (h_1 - z_c) \right)^{1.5} \tag{2.48}$$

for free flow and

$$Q = 0.5 C_D B_c \sqrt{g} \left(\frac{2}{3} (h_1 - z_c) \right)^{1.5} f \tag{2.49}$$

for drowned flow, with f a factor to correct for drowned flow. For more information about these equations we refer to the InfoWorks manual [6, 7]. Note that by replacing U_g with z_c the discharge over the gate becomes independent of the gate position itself. As will be shown in Chapter 3 this leads to controllability problems.

2.4.4 Conceptual model validation

The identification of the conceptual model-structure and the calibration of its parameters, based on the methodology described above, has been based on the simulation

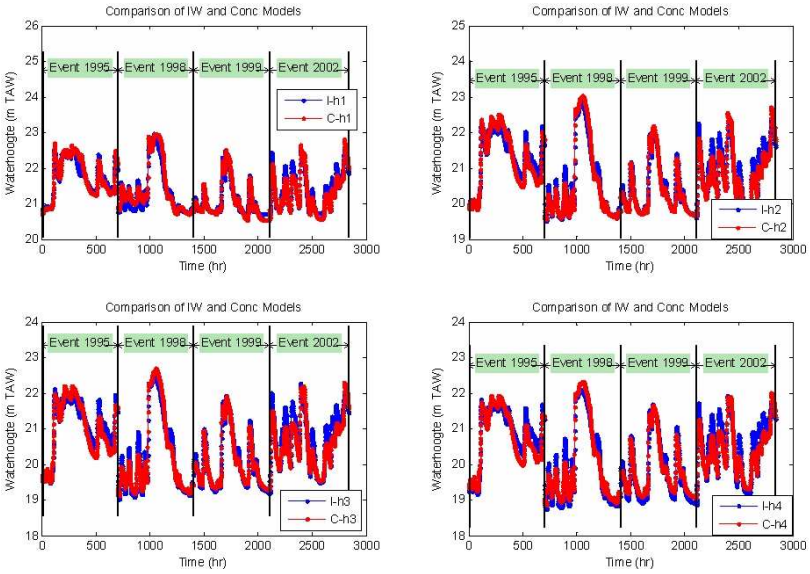


Figure 2.22: Comparison of the InfoWorks-RS (I) and conceptual model (C) results for the discharges and water levels at selected nodes; for the historical floods of 1995, 1998, 1999-2000 and 2002.(a)

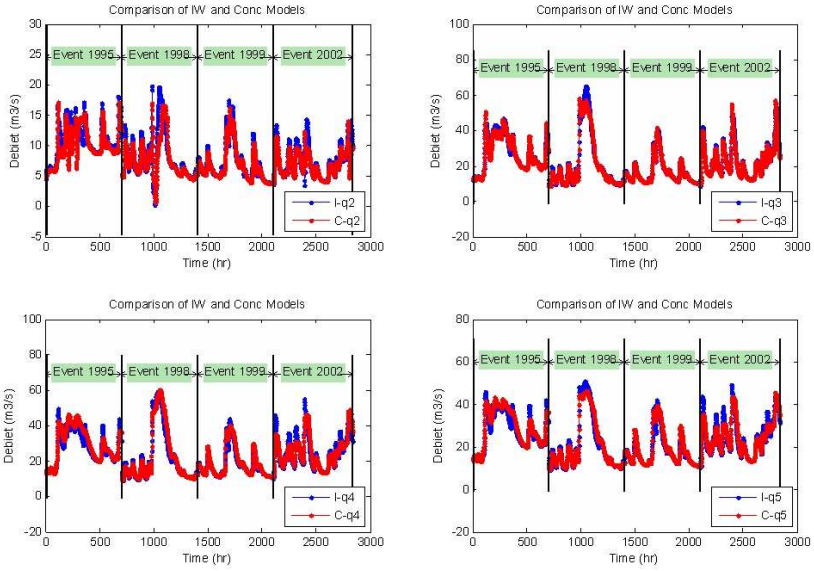


Figure 2.23: Comparison of the InfoWorks-RS (I) and conceptual model (C) results for the discharges and water levels at selected nodes; for the historical floods of 1995, 1998, 1999-2000 and 2002.(b)

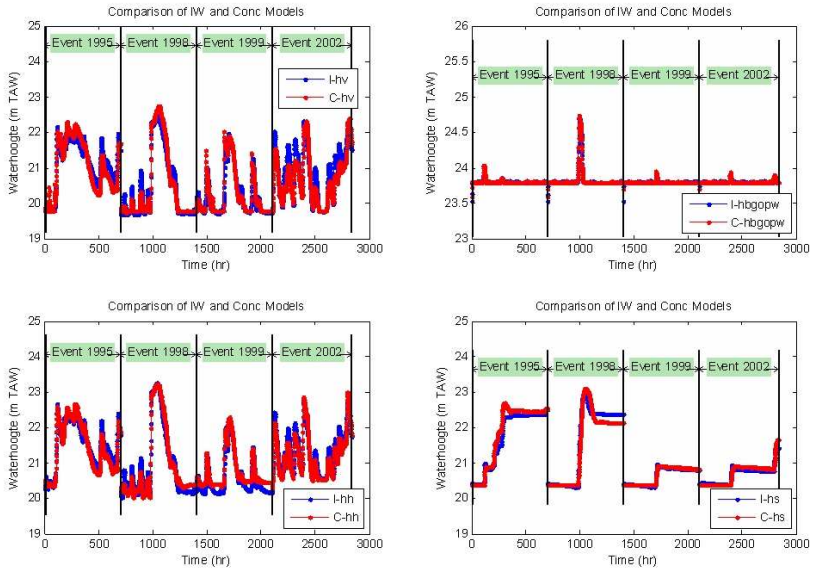


Figure 2.24: Comparison of the InfoWorks-RS (I) and conceptual model (C) results for the discharges and water levels at selected nodes; for the historical floods of 1995, 1998, 1999-2000 and 2002.(c)

results with the full hydrodynamic model for two historical high flow or flood events: the flood events of September 1998 and January 2002. Validation was done based on the full set of flood events during the past 15 years; being the flood events of 1995, 1998, $1999-2000$ and 2002. Figure 2.22-2.24 show some validation results, comparing the water level and discharge simulation results of the conceptual model with the one of the InfoWorks models. Both models have a time step of 5 minutes. The model output results shown in Figure 2.22-2.24 are aggregated at the hourly time step. In Table 2.1 a quantitative representation is given of the calibration process. The error e_r is defined as

$$e_r = 100\frac{y_c - y_{iw}}{y_{iw}}, \tag{2.50}$$

with y_c the simulation data of the conceptual model and y_{iw} the simulation data of the InfoWorks model. Note that the calibrated conceptual model has a calculation time 100 times shorter than the full hydrodynamic InfoWorks model.

	mean $\lvert e_r \rvert$ (%)	std $\lvert e_r \rvert$
hv	0.37	0.70
hbgopw	0.08	0.28
hh	0.96	0.92
hs	0.39	0.35
h1	0.66	0.54
h2	0.42	0.50
h3	0.66	0.67
h4	0.94	0.88

Table 2.1: The mean and the standard deviation of the absolute value of the error between the simulations results of the conceptual model and that of the InfoWorks model.

2.5 Conclusions

In this chapter an overview was given of the basic hydraulic concepts as well as a more detailed explanation of the 3 hydraulic conservation laws that form the basis for hydrodynamic water system modeling. A state of the art was given of the dynamic models that are used in practice for simulation and control purposes. Finite-difference based models are based on the conservation laws and simulate very accurately the behavior of river systems. Currently, in the Demer basin a prediction of 2 days ahead is performed based on rainfall predictions. This prediction is performed by the Wallingford software InfoWorks. This software package describes a full finite-difference based hydrodynamic model of the Demer basin. Based on the predictions obtained from this model the local water administration knows which areas will flood so evacuation can be planned in advance. The drawback of this model is that it requires a large amount of calculation time and is therefore mainly used for simulation purposes. For control purposes, models with a lower computational complexity are needed. For setpoint regulation the integrator-delay model is successfully used in practice. However, in this thesis setpoint regulation is not the main goal but flood prevention. During floods the complete nonlinear dynamics of the river system is excited and a simple integrator-delay model cannot be used. Therefore a reservoir type model was calibrated based on data obtained by simulating historical events with the InfoWorks software package. This conceptual model was calibrated on data obtained with the InfoWorks package by simulating an historical event that was not used for calibration of the model. This validation showed that the conceptual model gives an accurate enough description of the river dynamics to verify whether a model predictive controller can result to better flood regulation than the three-position controller. Moreover, compared with the full hydrodynamic InfoWorks model its calculation time is reduced with a factor of 100.

Chapter 3

Flooding Regulation with Model Based Predictive Control

In the previous chapter a conceptual nonlinear model was calibrated for the Demer basin. The model was calibrated and validated based on historical simulation data from the Demer basin. In this chapter this model will be controlled by a model predictive controller (MPC). The chapter starts by giving an overview of existing control strategies in Section 3.1. Next, in Section 3.2 a nonlinear MPC scheme is outlined for flooding regulation. The solution of the nonlinear MPC scheme is based on the optimization techniques introduced in Chapter 1. A strategy will be elaborated in order to deal with infeasibilities w. r. t. the flood levels in the basin and the local uncontrollability problem of the gates will be discussed and tackled. In Section 3.3 the performance of the outlined NMPC scheme will be tested by means of historical simulations and compared with that of the three-position controller. The experiments are twofold. In a first step the performance is tested on a simplified model only describing the upstream part of the river Demer. In a second step the controller is tested on the large-scale model representing the complete river Demer. Next, the robustness w. r. t. the uncertainty on the rainfall-runoff predictions is tested by historical simulations in Section 3.4. The chapter ends with conclusions in Section 3.5

3.1 State of the art

3.1.1 Controllers

Many controllers can be found in literature for the management of open water systems. The different types of controllers can all be categorized based on their water management related characteristics [93]. There are different possibilities to categorize

controllers. One way to classify controllers is based on general control theory and consists of following classes of controllers:

- Classical feedback and feedforward control
- Three-position control
- Optimal control
- Heuristic control
- Model predictive control

In the remainder of this section these different classes will be discussed in more detail as well as their use in practical management systems for the regulation of open water systems.

3.1.1.1 Classical feedback and feedforward control

In Fig. 3.1 a diagram of a classical feedback and feedforward controller for water system management is presented. The setpoint represents the desired water level. The objective of the controller is to steer the water levels as close as possible to the desired setpoints. In order to achieve this at each sampling time the different water levels are measured. This measured water level is then compared to the setpoint and the deviation is passed to the feedback controller. The feedback controller will then determine a control action that tries to remove this deviation. In water system management the feedback controller is typically a proportional integral (PI) controller [101]. The feedforward controller uses measurements of the disturbance (in this case rainfall) and an inverse model in order to counteract the effect the measured disturbance will have on the water level. In the ideal case this would lead to perfect setpoint regulation. In practice, however, there are uncertainties on the models and the measured disturbances which make perfect setpoint regulation with a feedforward controller impossible. In order to cancel out these uncertainties a feedforward controller is always combined with a feedback controller [94]. Successful applications of these techniques were reported in [29, 42, 78, 105].

3.1.1.2 Optimal control

The most common optimal controllers in water systems are based on the Linear Quadratic Regulator (LQR) theory. An LQR controller is a controller based on a quadratic criterion in which weighting matrices penalize the deviation of the inputs and the outputs from their steady-state value. The resulting input is a linear state feedback minimizing the quadratic criterion. Changing the weighting factors influences the response and robustness of the controller. Therefore, tuning of the weighting matrices is necessary in order to obtain a satisfactory control action. A drawback of this method is that constraints on the inputs and/or states cannot be taken into account. Works in which this method has been applied successfully can be found in [52, 83, 84].

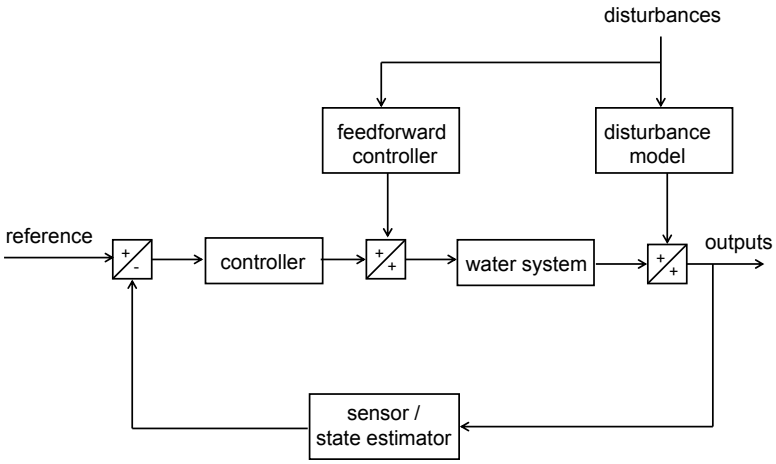


Figure 3.1: A diagram of a classical feedback and feedforward controller for water system management. [93]

3.1.1.3 Heuristic control

Heuristic control methods are based on a heuristic approach. Examples of these methods are control based on rule-of-thumb, neural networks control [114], fuzzy logic control [107, 108, 118] and genetic algorithm control [121]. These methods have to be tuned on a complete simulation model, or on the real system, since no mathematical tool can enable this, nor prove their performances. The dynamic behavior of the water system is seen as a black box, which is another drawback of these methods. Although quoted in literature these methods are hardly operational and too site specific.

3.1.1.4 Three-position controller

Three-position control [101] corresponds to a control mode that is based on some very simplistic rules and typically has the purpose to steer a water level to a desired setpoint. A three-position controller achieves this by reacting on each deviation from the setpoint by adjusting the gate position for a predetermined amount of time. A standard three-position controller consists of the following rules:

- The water level lies between desired upper and lower limit - no corrective action. The setpoint lies between the upper limit and lower limit.

- The water level exceeds the upper limit - the gate is lowered in order to lower the water level.

- The water level is lower than the lower limit - the gate is raised in order to raise

the water level.

The local water administration responsible for the water management of the Demer basin uses three-position controllers for the control of the gates. However, their controllers are more advanced than the standard three-position controller. During normal operation the purpose is to steer the water levels to their setpoints. This is done by standard three-position controllers. During periods of heavy rainfall the focus shifts towards flooding prevention. Therefore, the standard rules are replaced by new rules. These new rules determine the gate position based on the measurement of water levels in the water system. These rules are formulated by the local water administration based on many years of experience in regulating the Demer basin and can therefore be considered to be expert knowledge. These type of controllers have the advantage that the movement of the gate is limited, which is positive with respect to wear and tear of the gates. Another advantage is its straightforward implementation. Note that the controller used by the local water administration of the Demer is a combination of a feedback controller with a heuristic controller.

3.1.1.5 Model predictive control

In chapter 1 the principles of model predictive control were already outlined. A model predictive controller is a combination of a feedback controller, a feedforward controller and an optimal controller. In the control of open water systems MPC makes a prediction of the water levels based on the rain forecast and determines the control action that minimizes the cost function. The cost function is usually a quadratic criterion. An important advantage of MPC is that it can explicitly cope with constraints on the inputs and outputs of the system. Works in which MPC was successfully applied on open water systems are reported in [93, 116, 119].

3.1.2 MPC in practice

In figure 3.2 a schematic overview is depicted of a practical MPC control scheme for river control purposes. One can typically distinct a two level control hierarchy. In the top level one has the model predictive controller. Sensor measurements provide the MPC with information about the water levels. Based on that information the MPC controller determines the optimal trajectory for the discharges through the gates in order to steer the water levels as close as possible to their setpoints. The optimal discharges calculated by the MPC are then passed as setpoints to local controllers. In figure 3.2 PI controllers are chosen as local controllers but other controllers are also possible. The local controllers have the task to move the gates in order to steer the discharges through the gates as close as possible to the setpoints provided by the MPC. Note that the local controller usually operates at a much faster rate than the MPC.

The reason for using this two level control hierarchy is to make the *on-line* optimization problem of the MPC much easier. In the previous chapter it was already

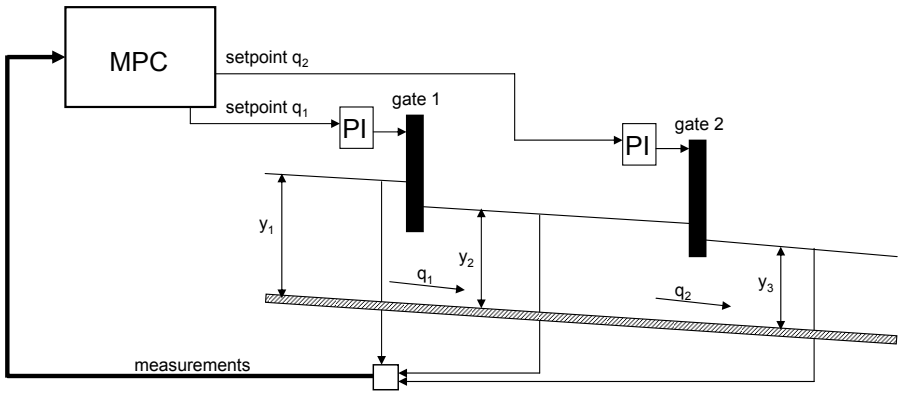


Figure 3.2: Practical MPC control scheme for river regulation

explained that one typically uses the integrator-delay model for practical river control applications. The integrator-delay model assumes a linear relationship between the water levels and the discharges. Therefore, if one uses the discharges as inputs the underlying model of the MPC is linear and therefore the *on-line* optimization problem the MPC needs to solve is a quadratic programming problem. If one would chose the gate openings as inputs the underlying model would be nonlinear and the MPC would need to solve *on-line* a nonlinear programming problem which is time-consuming at the one hand, but also introduces many local minima at the other hand and can therefore converge to a bad local minimum. Once the MPC passes the setpoints to the local controllers, it is up to the local controllers to determine the gate opening that leads to the desired discharge. This two level control hierarchy is also referred to as decentralized control.

In contrast to the existing river control applications which always concern setpoint regulation, in this thesis the main goal is to avoid flooding. In subsection 3.2.3 it will be outlined that in the latter case it is not straightforward to implement a decentralized control scheme. Therefore, in this thesis a centralized control scheme is implemented in which the water levels, discharges and volumes are considered as state variables and the gate openings as control inputs.

3.2 Nonlinear model predictive control

In the previous section a categorization of controllers was made based on control theory. The control strategy that seems the most suited for flooding regulation in the Demer basin is MPC because of following reasons:

- MPC solves at each sampling time an optimization problem. MPC can only be applied if this optimization problem can be solved within the sampling time. Because river systems have relatively slow dynamics the necessary sampling time is typically big enough in order to solve the optimization problem within the time constraint.

- MPC can cope with all the constraints that are present in a river system like physical upper and lower bounds of the gates and maximal gate movement. Also upper constraints on the water levels can be taken into account which is necessary for flooding prevention.

- By combining the rainfall predictions with the mathematical model of the river system MPC can make predictions of the future water levels and use this information for making better decisions w.r.t. flooding avoidance.

- During a flood event all the nonlinear dynamics of the river system are excited. So in order to make accurate predictions of the future water level it is necessary to have a nonlinear model of the river system. In literature [45, 56] there exist efficient nonlinear MPC schemes that can cope with nonlinear models.

- River systems are typically highly interactive multi-input-multi-output systems (MIMO). It is known that traditional control design for such kind of systems is very difficult because it makes use of relatively little information about the system. MPC, however, can effectively deal with MIMO systems.

As pointed out already, several works exist were MPC has been successfully applied for the regulation of water systems. However, the focus of these works is mainly steering the water levels to a desired setpoint. Therefore, in these works a linear model is used for making the predictions of the future states which gives rise to linear MPC (see chapter 1). For flooding regulation a simple linear model is not sufficient anymore as the complete nonlinear dynamics of the river system will be excited. Therefore a nonlinear MPC scheme is necessary. In the remainder of this section the nonlinear MPC scheme implemented in this thesis will be outlined in more detail.

3.2.1 Mathematical formulation

The process model used in this work is a nonlinear state space model described by

$$x_{k+1} = f(x_k, u_k, d_k) \tag{3.1}$$
$$y_k = Cx_k \tag{3.2}$$

with $x_k \in \mathbb{R}^{n_x}$ the state of the system at time step k, $u_k \in \mathbb{R}^{n_u}$ the input of the system consisting of the gate positions at time step k, $d_k \in \mathbb{R}^{n_d}$ the disturbance input containing the rainfall-runoff at time step k and the output $y_k \in \mathbb{R}^{n_o}$ containing the water levels at time step k. Note that the state x_k consists of the water levels, the discharges and the node volumes of the water system.

MPC tries to steer the future outputs to the reference value y_r while trying to keep the inputs as close as possible to the reference input u_r without violating constraints on the inputs and the states. MPC achieves this by optimizing on-line at each time step an optimal control problem. In the case of the nonlinear system (3.1) the online optimal control problem is a nonlinear programming problem defined as:

Optimization 3.1 (Nonlinear programming problem)**.**

$$\begin{aligned} \min_{\substack{x_{k+1},\ldots,x_{k+N_p} \\ y_{k+1},\ldots,y_{k+N_p} \\ u_k,\ldots,u_{k+N_c-1}}} \quad & \sum_{i=1}^{N_p} \left(y_{k+i} - y_r\right)^T Q \left(y_{k+i} - y_r\right) + \\ & \sum_{i=0}^{N_c-1} \left(u_{k+i} - u_r\right)^T R \left(u_{k+i} - u_r\right) \end{aligned} \tag{3.3}$$

subject to the following constraints for $i = 1,\ldots,N_p, j = 0,\ldots,N_c-1$*:*

$$x_k = \hat{x}_k \tag{3.4}$$
$$x_{k+i+1} = f(x_{k+i}, u_{k+i}, d_{k+i}) \tag{3.5}$$
$$y_{k+i} = Cx_{k+i} \tag{3.6}$$
$$u_{k+i} = u_{k+N_c-1}, \textit{for } i \geq N_c \tag{3.7}$$
$$\underline{y} \leq y_{k+i} \leq \bar{y} \tag{3.8}$$
$$\underline{u} \leq u_{k+j} \leq \bar{u} \tag{3.9}$$
$$\left\|u_{k+j+1} - u_{k+j}\right\|_\infty \leq \Delta_{max} \tag{3.10}$$

with N_p the prediction horizon, N_c the control horizon, Q and R positive-definite diagonal weighting matrices, $\underline{y}$ and $\bar{y}$ under and upper limits on the outputs, $\underline{u}$ and $\bar{u}$ under and upper limits on the inputs, and $\hat{x}_k$ the measured or estimated state at time step k. The prediction horizon is usually bigger than the control horizon. The inputs after the control horizon are all equal to u_{k+N_c-1}. Constraints (3.5) and (3.6) represent the nonlinear system equations. Constraint (3.8) represents the constraints bounding the different water levels in order to avoid flooding. Constraint (3.9) ensures the gates do not exceed their upper limit and constraint (3.10) ensures the maximal possible gate movement is not exceeded.

Optimization 3.1 is a constrained nonlinear programming problem due to the nonlinear equality constraint (3.5) representing the dynamical model of the water

system. Nonlinear programming problems are typically solved in an iterative way [91]. Suppose a point is described as $p = \left[\begin{array}{cccccc} u_k^T & \dots & u_{k+N_c-1}^T & x_{k+1}^T & \dots & x_{k+N_p}^T \end{array} \right]^T$, then first a starting point p_0 is chosen which in this thesis is typically a relatively good estimate of the (local) optimal solution of the nonlinear programming problem. Then a sequence of iterates $\{p_k\}_{k=0}^{\infty}$ with decreasing value for the cost function (3.3) are generated that terminate when either no more progress can be made, when it seems a solution point has been approximated with sufficient accuracy or when the available time (the sampling time) has expired.

In order to obtain such a sequence of improving iterates $\{p_k\}_{k=0}^{\infty}$ a nonlinear MPC scheme is used [45]. Basically, a nonlinear MPC scheme approximates the nonlinear behavior of the dynamical system locally by a linear time-variant system. A linear time-variant system has the following form:

$$x_{k+1} = A_k x_k + B_k u_k + d_k \tag{3.11}$$

$$y_k = C x_k \tag{3.12}$$

with A_k,B_k and C_k the system matrices of the system at time step k. Remark that the system matrices are time dependent which is why the system is called time variant. Also note that the disturbance input d_k is time variant and makes it possible to take future rain predictions into account. The nonlinear scheme starts by performing a simulation with the future optimal inputs obtained at the previous time step. This leads to a trajectory of future states. The future optimal inputs computed at time step $k-1$ are defined as

$$\begin{array}{cccccc} u_{k|k-1} & \dots & u_{k+N_c-1|k-1} & \dots & u_{k+N_c-1|k-1} \end{array} \tag{3.13}$$

with $u_{k+i|k}$ denoting the input at time step $k+i$ predicted at time step k. At time step k a simulation of the river model is done using the following inputs

$$\begin{array}{cccccc} u_{k|k-1} & \dots & u_{k+N_c-2|k-1} & u_{k+N_c-1|k-1} & \dots & u_{k+N_c-1|k-1} \end{array} . \tag{3.14}$$

This simulation gives rise to a sequence of future states

$$\begin{array}{ccc} x_{k+1}^0 & \dots & x_{k+N_p}^0 \end{array} . \tag{3.15}$$

Combining (3.14) and (3.15) and assuming for convenience that

$$\left[\begin{array}{ccc} (u_k^0)^T & \dots & (u_{k+N_c-1}^0)^T \end{array} \right] = \left[\begin{array}{cccc} u_{k|k-1}^T & \dots & u_{k+N_c-1|k-1}^T & \dots & u_{k+N_c-1|k-1}^T \end{array} \right] \tag{3.16}$$

an initial starting point

$$p_0 = \left[\begin{array}{cccccc} (u_k^0)^T & \dots & (u_{k+N_c-1}^0)^T & (x_{k+1}^0)^T & \dots & (x_{k+N_p}^0)^T \end{array} \right]^T \tag{3.17}$$

can be obtained that is a relatively good estimate of the (local) solution of the nonlinear programming problem defined in optimization 3.1. The nonlinear equality constraint (3.5) representing the nonlinear behavior of the system can now be linearized around the point p_0 leading to the following linear time-varying approximation of the behavior of the nonlinear model around the point p_0:

$$\begin{aligned} x_{k+j+1} = A_{k+j}(x_{k+j} - x^0_{k+j}) + \\ B_{k+j}(u_{k+j} - u^0_{k+j}) + x^0_{k+j+1} \end{aligned} \tag{3.18}$$

with x^0_{k+j} and u^0_{k+j} the simulated state and input at time step $k+j$ and $x^0_{k+j+1} = f(x^0_{k+j}, u^0_{k+j}, d_{k+j})$ and with the matrices A_k and B_k defined by:

$$A_{k+j}\,(r,s) = \frac{\partial f_r\,(k+j)}{\partial x_s\,(k+j)} \tag{3.19}$$

$$B_{k+j}\,(n,m) = \frac{\partial f_n\,(k+j)}{\partial u_m\,(k+j)}. \tag{3.20}$$

In this thesis the matrices A_{k+j} and B_{k+j} are obtained numerically by means of finite differences. The central difference scheme is the most accurate of all the finite difference schemes and is therefore used in this thesis. The matrices A_{k+j} and B_{k+j} can numerically be obtained from the central finite difference scheme as follows:

$$A_{k+j}(:,s) = \frac{f(x^0_{k+j} + \Delta x(s), u^0_{k+j}) - f(x^0_{k+j} - \Delta x(s), u^0_{k+j})}{2\delta} \tag{3.21}$$

$$B_{k+j}(:,s) = \frac{f(x^0_{k+j}, u^0_{k+j} + \Delta u(s)) - f(x^0_{k+j}, u^0_{k+j} - \Delta u(s))}{2\delta} \tag{3.22}$$

with $A_{k+j}(:,s), B_{k+j}(:,s)$ denoting the s-th column of A_{k+j}, B_{k+j}, $\Delta x(s)$ and $\Delta u(s)$ two column vectors containing only zeros with exception of the s-th element which is equal to a small perturbation δ. The linearized system (3.18) can be re-written into the form of (3.11) by noting that $d_k = x^0_{k+1} - A_k x^0_k - B_k u^0_k$. Also note that the rain predictions are taken into account implicitly due to the presence of the states x^0_{k+j+1}.

The obtained linear time-varying system (3.11) is an approximation of the nonlinear model. Note that this approximation is only valid within a certain region around the states and inputs around which the system is linearized. Replacing the nonlinear system equation (3.5) by the linear time-varying system (3.11) can be done according to 2 different schemes, namely the *uncondensed* and the *condensed* scheme. The most straightforward scheme is the uncondensed scheme in which the system states as well as the control inputs are both considered as optimization variables in the optimization scheme:

Optimization 3.2 (Uncondensed scheme)**.**

$$\min_{\substack{x_{k+1},\ldots,x_{k+N_p}\\ y_{k+1},\ldots,y_{k+N_p}\\ u_k,\ldots,u_{k+N_c-1}}} \sum_{i=1}^{N_p} (y_{k+i}-y_r)^T Q\,(y_{k+i}-y_r) + \sum_{i=0}^{N_c-1} (u_{k+i}-u_r)^T R\,(u_{k+i}-u_r) \tag{3.23}$$

subject to the following constraints for $i = 1,\ldots,N_p, j = 0,\ldots,N_c-1$

$$\begin{aligned}
x_{k+i+1} &= A_{k+i}x_{k+i} + B_{k+i}u_{k+i} + z_{k+i} && (3.24)\\
y_{k+i} &= Cx_{k+i} && (3.25)\\
u_{k+i} &= u_{k+N_c-1}, \textit{for } i \geq N_c && (3.26)\\
\underline{y} &\leq y_{k+i} \leq \bar{y} && (3.27)\\
\underline{u} &\leq u_{k+j} \leq \bar{u} && (3.28)\\
\|u_{k+j+1} - u_{k+j}\|_\infty &\leq \Delta_{max} && (3.29)\\
& && (3.30)
\end{aligned}$$

Note that the optimization program is now a QP.

The second scheme that can be used is the condensed scheme. In the condensed scheme the state variables are eliminated from the optimization scheme. The elimination of the states can be achieved as follows. From (3.11) it follows that

$$y(k+1) = CA_kx(k) + CB_ku(k) + Cd(k). \tag{3.31}$$

Taking (3.31) into account the next predicted output can be written as:

$$\begin{aligned}
y(k+2) &= CA_{k+1}x(k+1) + CB_{k+1}u(k+1) + Cd(k+1)\\
y(k+2) &= CA_{k+1}A_kx(k) + CA_{k+1}B_ku(k) + CA_{k+1}d(k) +\\
&\quad CB_{k+1}u(k+1) + Cd(k+1)
\end{aligned} \tag{3.32}$$

Remark that the predicted output only depends on the current state $x(k)$, the disturbance inputs $d(k+i)$ which are known in this work and the current and future control inputs $u(k+i)$ that have to be determined. Repeating this for all the other outputs, the predicted outputs can be written as follows:

$$
\begin{bmatrix} y_{k+1} \\ y_{k+2} \\ y_{k+3} \\ \dots \\ y_{k+N_p} \end{bmatrix} = \begin{bmatrix} CP_0 \\ CP_1 \\ CP_2 \\ \dots \\ CP_{N_p-1} \end{bmatrix} x_k +
$$

$$
\begin{bmatrix} CB_0 & 0 & 0 & \dots & 0 \\ CP_0B_0 & CB_1 & 0 & \dots & 0 \\ CP_1B_0 & CP_0B_1 & CB_2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ CP_{N_p-1}B_0 & CP_{N_p-2}B_1 & CP_{N_p-3}B_2 & \dots & CB_{N_p-1} \end{bmatrix} \begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ \dots \\ u_{k+N_c-1} \\ \dots \\ u_{k+N_c-1} \end{bmatrix} +
$$

$$
\begin{bmatrix} C & 0 & 0 & \dots & 0 \\ CP_0 & C & 0 & \dots & 0 \\ CP_1 & CP_0 & C & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ CP_{N_p-1} & CP_{N_p-2} & CP_{N_p-3} & \dots & C \end{bmatrix} \begin{bmatrix} d_k \\ d_{k+1} \\ d_{k+2} \\ \dots \\ d_{k+N_p-1} \end{bmatrix} \tag{3.33}
$$

with $P_i = \prod_{n=0}^{i} A_{k+i-n}$ and $B_i = B_{k+i}$.

This can be rewritten into the following shortened vector notation:

$$Y_p = Gx_k + H\mathbf{u} + J\mathbf{d}, \tag{3.34}$$

with

$$\mathbf{u} = \begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ \dots \\ u_{k+N_c-1} \\ \dots \\ u_{k+N_c-1} \end{bmatrix} \tag{3.35}$$

and

$$\mathbf{d} = \begin{bmatrix} d_k \\ d_{k+1} \\ d_{k+2} \\ \dots \\ d_{k+N_p-1} \end{bmatrix}. \tag{3.36}$$

Taking this into account the condensed scheme leads to following optimization problem:

Optimization 3.3 (Condensed scheme)**.**

$$\min_{\boldsymbol{u}} \boldsymbol{u}^T \left(H^T H + \hat{R}\right) \boldsymbol{u} + 2\left(x_k^T G^T \hat{Q} H + d^T J^T \hat{Q} H\right) \boldsymbol{u} \tag{3.37}$$

subject to the following constraints for $i = 1, \ldots, N_p$ *and* $j = 1, \ldots, N_c - 1$

$$\begin{aligned}
\hat{I}_{k+i}\boldsymbol{u} &= \hat{I}_{k+N_c-1}\boldsymbol{u}, \text{ for } i \geq N_c && (3.38)\\
\underline{y} &\leq \bar{I}_{k+i}\left(Gx_k + H\boldsymbol{u} + J\boldsymbol{d}\right) \leq \bar{y} && (3.39)\\
\hat{I}_{k+i}\boldsymbol{u} &= \hat{I}_{k+N_c-1}\boldsymbol{u}, \text{ for } i \geq N_c && (3.40)\\
\underline{u} &\leq \hat{I}_{k+j}\boldsymbol{u} \leq \bar{u} && (3.41)\\
\left\|\hat{I}_{k+j+1}\boldsymbol{u} - \hat{I}_{k+j}\boldsymbol{u}\right\|_\infty &\leq \Delta_{max} && (3.42)
\end{aligned}$$

with $\hat{I}_{k+i}$ the matrix that multiplied with $\mathbf{u}$ leads to u_{k+i}, $\bar{I}_{k+i}$ the matrix that multiplied with $Y_p = Gx_k + H\mathbf{u} + J\mathbf{d}$ leads to y_{k+i} and $\hat{Q}$ and $\hat{R}$ defined as

$$\hat{Q} = \begin{bmatrix} Q & 0 & 0 & \cdots \\ 0 & Q & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \ldots & Q \end{bmatrix}, \quad \hat{R} = \begin{bmatrix} R & 0 & 0 & \cdots \\ 0 & R & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \ldots & R \end{bmatrix}. \tag{3.43}$$

Optimization 3.2 and 3.3 are both QP's that are local approximations of the NLP defined in optimization 3.1. Both QP's can therefore be used in a SQP approach in order to find a local minimizer of the NLP in optimization 3.1 by generating a sequence of iterates $\{p_k\}_{k=0}^{\infty}$, with the first iterate p_0 defined by (3.17). Because the QP's are a local approximation of the NLP to be solved their solution does not guarantee to return an iterate with a lower value for the nonlinear cost function of the NLP. Therefore, in order ensure that the iterates from the resulting sequence $\{p_k\}_{k=0}^{\infty}$ have a decreasing value for the nonlinear cost function in each iteration a trust region or a line-search approach is needed (see chapter 1). In this thesis we opted for a line-search. The line-search used here is a simplification of the one discussed in chapter 1 and is performed as follows. Assume that at iteration i the system is linearized around point p_{i-1} and that the point p_i^* is obtained as the optimal solution of one of the two QP's from optimization 3.2 and 3.3. A line-search is then performed by searching the optimal value of α for which the point p defined as $p = p_{i-1} + \alpha(p_i^* - p_{i-1})$ has the lowest value for the cost function of the NLP of optimization 3.1. An approximated solution for α is obtained by varying α from 0 to 1 in steps of $\frac{1}{25}$ and taking that value of α for which the point p has the lowest value for the nonlinear cost function. Assume α^* to be the optimal value for α then the iterate $p_i = p_{i-1} + \alpha^*(p_i^* - p_{i-1})$. With this new point p_i a new simulation can be done leading to future states around which the nonlinear system is linearized again. A new QP is then solved which in combination with the line-search leads to the next iterate p_{i+1}. This process is repeated until convergence is obtained or until the available time has expired. Note that the step size of $\frac{1}{25}$ was determined by several offline experiments that showed that this step size was small enough to obtain a good solution after the line-search procedure and at the same time big enough to

guarantee a fast solution of the line-search problem. In the following algorithm the complete procedure is summarized:

Algorithm 3.1.

(i) At time step k initialize the inputs based on the solution of the previous time step $\left\{u_k^0, \ldots, u_{k+N_c-1}^0\right\}$.

(ii) Perform a simulation with the input sequence $\left\{u_k^0, \ldots, u_{k+N_c-1}^0\right\}$ and obtain the future states $\left\{x_{k+1}^0, \ldots, x_{k+N_p}^0\right\}$.

(iii) Perform a linearization around these inputs and future states and solve optimization (3.2).

(iv) Perform a line-search.

(v) If the difference (based on the ∞-norm) of the new optimal input sequence $\left\{u_k^*, u_{k+1}^*, \ldots, u_{k+N_c-1}^*\right\}$ with $\left\{u_k^0, \ldots, u_{k+N_c-1}^0\right\}$ is smaller than a specified tolerance then stop the algorithm, else replace $\left\{u_k^0, \ldots, u_{k+N_c-1}^0\right\}$ with $\left\{u_k^*, u_{k+1}^*, \ldots, u_{k+N_c-1}^*\right\}$ and go to step 2.

So after the solution of a sequence of quadratic programs algorithm 3.1 returns a local optimal solution to the initial nonlinear optimal control problem of optimization 3.1. After convergence of the algorithm the first input u_k^* of the obtained input sequence is applied to the real system. At the next sampling time measurements on the system are performed based on which the current state of the system is estimated by means of a state estimator and the optimization procedure is repeated.

Remark 3.1. *Note that in this thesis the uncondensed QP's are used for the SQP approach because they are better conditioned than the QP's resulting from the condensed QP's.*

Remark 3.2. *In this thesis the QP's of optimization (3.2) are solved by the MOSEK QP solver [5].*

Remark 3.3. *The objective of this thesis is to verify whether floods in the Demer basin can be further reduced by replacing the current three-position controller of the local water administration by a model based predictive controller. If the results in this thesis show that MPC is indeed a better alternative, the next step of the local water administration is to replace the three-position controller in use by INCA, an industrial MPC software package developed by IPCOS [8]. The model predictive controller implemented in this thesis is developed in such a way that it is straightforward to plug INCA into the developed code. This is the reason why in this thesis we implemented a complete SQP algorithm instead of using an existing SQP solver.*

3.2.2 Constraint and cost function strategy

As pointed out earlier there are two types of constraints, namely hard and soft constraints. Hard constraints are constraints that can never be violated and are typically physical limitations on the gate movements. On the other hand, soft constraints are constraints that can only be violated in case no solution can be obtained that satisfies them all. In water systems the upper limits on the water levels are considered as soft constraints. During periods of heavy rainfall the situation can occur where it is impossible to keep each water level under its corresponding flood level meaning that the constraints are too stringent. In order to obtain a feasible solution some of the soft constraints need to be relaxed or omitted from the optimization problem. In order to do this a strategy needs to be defined that determines which soft constraints need to be relaxed. Simultaneously there also needs to be a strategy for the cost function as the objective is to steer the water levels corresponding to the relaxed soft constraints back into the feasible region.

Another reason to incorporate a variable constraint and cost function strategy is that the objective of MPC depends on the state of the water system. More specifically, during periods of light rainfall the main goal of MPC is to steer the water levels to a specified reference level. However, during periods of heavy rainfall steering the water levels to reference levels becomes less important as the focus shifts to flood prevention. This change in objectives has as consequence that the constraints as well as the objective function of MPC needs to change during operation. In the remainder of this subsection the chosen constraint and cost function strategy is outlined in more detail.

At first, during normal regulation the objective of MPC is to steer the water levels to the reference levels that are provided by the local water administration. It is important the physical limitations on the gates are satisfied. The constraints on the water levels do not play an important role here as the rainfall is too low to cause floodings. Besides this it is also important that during this period of time the water reservoirs are emptied and reach the lowest level that is allowed. Any unnecessary volume inside the reservoirs can lead to unnecessary floodings in the future. Therefore during normal regulation the cost function is adjusted to steer the most important water levels to their reference levels and to empty the reservoirs as much as possible.

During periods of heavier rainfall the expert knowledge of the operators controlling the gates in the field is incorporated in the MPC. At first sight this approach might seem restrictive as MPC is not allowed to use all possible degrees of freedom that are available. However, this is a necessary step in order for the MPC to avoid flooding properly. The reason for this is that rainfall events causing floodings can last more than five days. This means that the prediction horizon N_p of MPC should be taken longer than five days in order to be able to properly avoid flooding. But the prediction horizon cannot be taken that long for the following reasons:

- The optimal control problem to be solved at each time step would be intractable

as the number of unknown variables would be too large.

- Rain predictions are not accurate enough to predict so much in advance. Typically rain predictions are only reliable to predict 2 days ahead and even then those predictions can still have a significant amount of uncertainty.

Therefore it is necessary to incorporate some predefined rules containing expert knowledge in order to prevent flooding. Within these rules it is the task of MPC to find the optimal control strategy. This expert knowledge contains the following rules:

(i) For each water level a guard level is defined by the local water administration. As long as the water levels do not violate their corresponding guard level, it is not allowed to fill the water reservoirs. The reason behind this is to keep the reservoirs as empty as possible for as long as possible. Only when MPC detects that a guard level will get violated within the prediction horizon it is allowed to use some of the storage capacity from the reservoirs to keep the water levels beneath their guard level. This can be achieved by adding the guard levels as constraints of the optimization problem and putting a high value in the cost function for deviations of the reservoir levels from their corresponding reference level. The exact value in the cost function is determined by several experiments on smaller past flood events.

(ii) Water reservoirs can be used to avoid violation of the guard levels. However, it is not allowed to use the complete storage volume available in the reservoirs. For each reservoir a safety limit is defined. Once this safety limit is reached the reservoirs may not be used anymore and the guard levels are allowed to be violated. The reason for this is to ensure there is enough storage volume available in case of further future rainfall. Therefore, in the optimization the guard levels are replaced by their flood level and constraints are added to ensure the reservoir levels stay beneath their safety level.

(iii) If it continues raining until the point where MPC cannot keep the water levels beneath their flood level within the prediction window, it is allowed to further fill the reservoirs until the water level in the reservoirs reach their corresponding flood level. In order to achieve this the safety limits are replaced in the optimization by the flood levels. By doing this MPC can use all the available storage volume in the reservoirs in order to keep all water levels beneath their flood level.

(iv) If the rainfall is really excessive then flooding is unavoidable and some or all water levels will violate their corresponding flood levels and the optimization proposed in step 3 will be infeasible. In order to be able to get a useful solution

from MPC soft constraints need to be removed from the optimization until a feasible solution is obtained. If a soft constraint on a specific water level is removed, the weight in the cost function corresponding to that water level is increased. This is done in order to minimize the soft constraint violation.

Note that the removal of soft constraints outlined in step 4 is based on a priority based strategy. First, different sets with different priorities are defined. Soft constraints that are more important than others are categorized into sets with higher priorities. Soft constraints of equal importance are categorized into the same priority set. If the MPC defined in step 3 turns out to be infeasible all the soft constraints belonging in the set with the lowest priority are removed from the optimization and in the cost function the weights of the corresponding water levels are increased. If the optimization still turns out to be infeasible the soft constraints from the second lowest priority set are removed and the corresponding weights are adjusted in the cost function. This procedure is repeated until a feasible solution is obtained.

Remark 3.4. *Note that the outlined strategy with guard levels and safety limits is useful because MPC works with a limited prediction horizon and this strategy ensures that at the end of the horizon there is always storage capacity available in the reservoirs to cope with rainfall beyond the horizon. However, a better alternative would be to add a terminal cost in the cost function. But in this thesis we opted to make use of the strategy of guard levels and safety limits because this was imposed as a constraint by the local water administration.*

3.2.3 Uncontrollable modes

In river control (irrigation channels) the discharges over the gates are typically chosen as control inputs. As a result MPC returns a sequence of future optimal discharge sequences. This sequence is passed as setpoints to local controllers that try to follow their setpoint as close as possible. The advantage of this approach is that the underlying system dynamics turn linear. In literature many examples can be found where this approach has been applied successfully [29, 84, 93]. All these examples concern cases where the rainfall is relatively small and where the goal is to steer the water levels to a given reference level. In this work however the main focus is flooding prevention and taking the discharges as inputs can pose severe difficulties. The discharge over a gate depends on the gate level and the water levels up- and downstream of the gate. This dependence is strongly nonlinear and by taking the discharges as inputs this nonlinearity is neglected. During periods of heavy rainfall these nonlinearities play a very important role and therefore if the discharges over the gates are considered as inputs the MPC might come up with future discharge trajectories that cannot be realized by the gates.

For this reason in this work the gate levels are chosen as control inputs. A difficulty that arises when taking the gate levels as control inputs is that the gates can become locally uncontrollable. This uncontrollability is related to the discharge equations

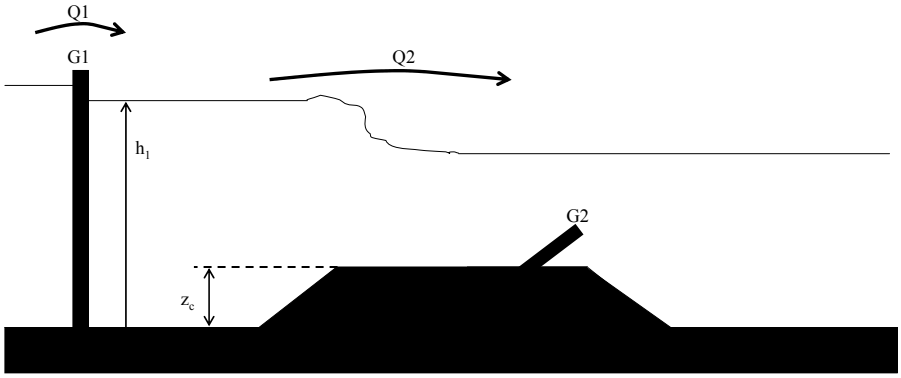


Figure 3.3: Two examples of gates that are in an uncontrollable mode.

determining the discharge over a gate. These equations consist of different modes. In some of these modes the discharge equation is independent of the gate level. If a gate reaches such a mode the gate becomes locally uncontrollable. Because MPC relies on local linear models this means that the controller is not able to control the gate anymore as it thinks that the gate has no influence on the water system.

In Fig. 3.3 two uncontrollable modes are depicted. Gate $G1$ is closed and therefore the discharge over $G1$ is equal to zero. This is the case for as long as the gate is closed which means that the discharge over a closed gate is independent of the gate level. During normal operation the gates controlling the flow to the flood reservoirs are typically in this mode. This can decrease the performance of the controller because controllability is only recovered whenever the water level upstream of the closed gate raises above the gate level. At the other hand, because of the sharp angle of gate $G2$ it lies in the throat control mode and therefore gate $G2$ locally does not influence the discharge going from upstream to downstream as can be seen from equations (2.48) and (2.49). Gate $G2$ is therefore uncontrollable. Gate $G2$ can only recover from this uncontrollability if it is raised. The uncontrollability problem can also be explained from a system theoretic point of view. Let h_{ups} be the water level upstream of the gate, h_{ds} the water level downstream of the gate, q the discharge over the gate and u_k the gate position at time step k. In one of the aforementioned uncontrollable modes the linearized system equations have the following form:

$$\begin{bmatrix} h_{ups}(k+1) \\ h_{ds}(k+1) \\ q(k+1) \end{bmatrix} = A \begin{bmatrix} h_{ups}(k) \\ h_{ds}(k) \\ q(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} u_k. \tag{3.44}$$

Note that all the elements of the B-matrix are zero and therefore the controllability matrix is not of full rank, which means that this linearized system is uncontrollable.

Losing the controllability of a gate can significantly decrease the performance of the controller. Therefore, the controller must try to steer each uncontrollable gate back into a controllable mode. As explained previously, at each time step a simulation is done in order to approximate the nonlinear system (3.1) by a linear time-varying system (3.11). At this point a check can be done to determine for each gate whether it will get uncontrollable within the time-window. This can be done by inspecting the columns of the B_k matrix in system equation (3.11). If a certain column p of B_k consists of only zeros, then the corresponding input $u_k\,(p)$ is uncontrollable at time step k. Now assume that the water levels up -and downstream of input $u_k\,(p)$ are equal to h_u and h_d respectively. If the gate is uncontrollable because it is completely closed the reference input u_r is chosen such that it is lower than the maximum of h_u and h_d. If the gate is uncontrollable because it is in the throat control mode then the reference u_r is chosen such that the gate is raised into a controllable mode.

In Fig. 3.4 a schematic overview is given of the complete nonlinear MPC scheme used in this thesis.

3.3 Experimental results

As mentioned in chapter 2 this study was divided in two parts. In a first phase a 'reduced area' around the Schulensmeer reservoir has been considered and a separate conceptual model was developed for this. In a second phase a much larger region of the

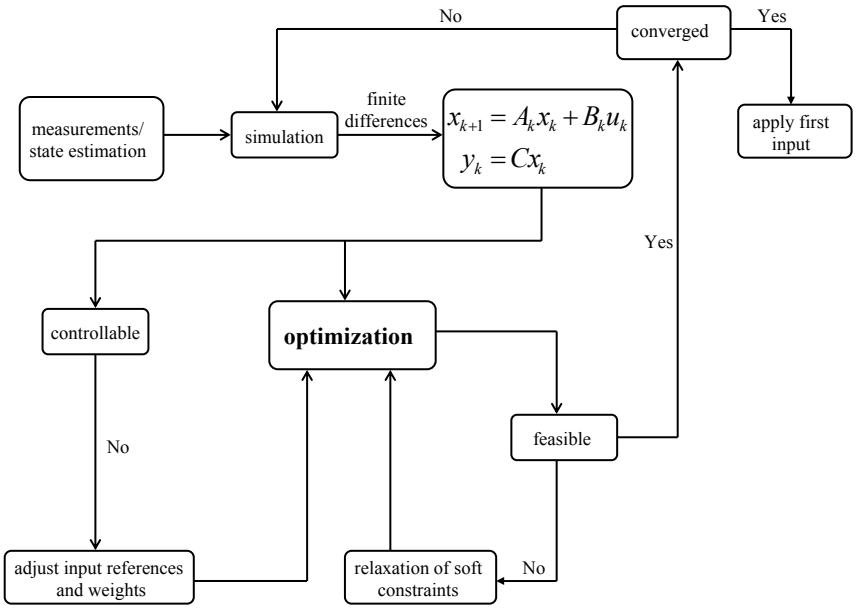


Figure 3.4: Schematic overview of the presented nonlinear NMPC scheme.

Demer basin was considered containing the two flood control reservoirs 'Webbekom' and 'Schulensmeer'. MPC was tested in both phases and compared with the current three-position controller of the local water administration. In the following these results are presented and discussed.

3.3.1 Performance indicators

In order to assess whether the performance of the nonlinear MPC is better than that of the three-position controller currently in use performance indicators are needed. The first performance indicator that is used in this thesis, is called the regulation cost. The regulation cost is related to a certain water level $h \in \mathbb{R}$ and is defined as:

$$J_{reg}(h) = \sum_{i=1}^{N_{sim}} (h_i - h_r)^T (h_i - h_r) \tag{3.45}$$

with N_{sim} the length of the simulation, h_i the water level h at time instant i and h_r the desired reference level for water level h. Basically, the regulation cost $J(h)$ indicates how much the water level h deviates from the desired reference level h_r. Since the main purpose of the controller is to avoid/decrease floodings a second indicator is needed which gives an indication of the amount of flooding that has occurred. This indicator will be called the flood cost and is defined as

$$J_f(h) = \sum_{i=1}^{N_{sim}} e_i^f(h) \tag{3.46}$$

with

$$e_i^f(h) = \begin{cases} (h_i - h_f)^T (h_i - h_f), \text{for } h_i \geq h_f \\ 0, \text{for } h_i \leq h_f \end{cases} \tag{3.47}$$

and h_f the flood level of water level h.

3.3.2 Experimental results on the 'reduced area' around Schulensmeer

The 'reduced area' considered here is depicted in Fig. 2.11. There are 3 water levels that need to be controlled, namely 'h_{opw}', 'h_s' and 'h_{afw}'. In order to achieve this MPC needs to control the 3 gates that are available, namely gates k_a, k_{k7} and k_d. During periods of moderate rainfall the following control objectives need to be achieved:

- Water level h_{opw} needs to be steered as close as possible to a desired reference level of 21.5 m.
- The flood reservoir represented by water level h_s should be kept as empty as possible in order to have as much as possible capacity available for flood prevention.

If the rainfall is excessive then the control objectives need to be changed and focus shifts towards flooding prevention. The new control objectives are as follows:

- Keep the water levels h_{afw} and h_s under their flood levels, respectively 22.75 m and 23.2 m.
- Keep water level h_{opw} under the guard level of 23 m. If h_{opw} violates this guard level extra storage capacity is available for flooding prevention by means of farm land. However, the local water administration needs to compensate the farmers for this. Therefore, it is preferable to avoid using this farm land.
- If h_{hopw} surpasses its guard level, then try to keep h_{opw} under its flood level of 23.2 m.

These control objectives need to be satisfied by suitable control actions. The control actions at their turn are also limited by physical limitations:

- Gates cannot move at a rate faster than 0.1 m/h.
- The gates have upper and lower limits which they cannot violate:
  - $20.00m \leq k_a \leq 22.50m$
  - $20.03m \leq k_{k7} \leq 23.00m$
  - $18.90m \leq k_d \leq 22.90m$

In order to compare the performance of MPC w.r.t. the performance of a three-position controller 2 simulations will be performed. The first simulation concerns the historical rainfall event of 1998. The rainfall event lead to one of the biggest floodings in the history of the Demer basin. Therefore, this event is suitable to draw conclusions w.r.t. the performance of MPC for flooding regulation. The second simulation concerns an artificial created rainfall event. The first part of this rainfall event coincides with the historical rainfall event of 1998. In the second part a second rainfall peak is inserted. The purpose of this event is to see whether MPC can deal better with 2 successive rainfall peaks than the three-position controller.

3.3.2.1 Simulation of historical event of 1998

In figure 3.5 the rainfall-runoff discharges of 1998 are depicted. These rainfall-runoff discharges are the disturbance inputs for our conceptual model. MPC uses this information as prediction of rainfall-runoff discharges in order to make predictions of the future states of the system. Note that in these simulations it is assumed that MPC has exact knowledge of the future rainfall-runoff predictions. In practice this is never the case as there is always uncertainty on the rainfall-runoff discharge predictions. The prediction and control horizon of MPC is chosen to be equal and is 30 hours. In Fig. 3.6 and Fig. 3.7 simulations are depicted with a three-position controller and a MPC controller respectively . Both simulations simulate the historical rainfall-runoff discharges of 1998. Comparing both figures leads to following observations:

- Before and after the big peak in the rainfall-runoff discharges MPC steers h_{opw} much closer to the reference level of $21.5m$ than the three-position controller.

- After the big peak MPC empties the reservoir (h_s) much earlier and faster than the three-position controller.

- MPC uses less capacity of the reservoir than the three-position controller. This means that MPC still has extra storage capacity available in the flood reservoir.

- MPC is capable to keep all water levels under their flood level. The simulation with the three-position controller shows that h_{afw} violates its flood level of $22.75m$. This is indicated in figure 3.6 by the shaded area.

- In both simulations the water level h_{opw} surpasses its warning level of $23m$. Therefore, a part of the farm land is used to avoid flooding. However, with MPC significantly less farm land is used than with the three-position controller.

- In figures 3.8 and 3.9 the control actions of the 3 gates are depicted respectively for the three-position controller and the MPC controller. It can be seen that for both controllers the gates do not violate their physical upper and lower limit. Though not shown on the picture the gates never move faster than $0.1m/hour$.

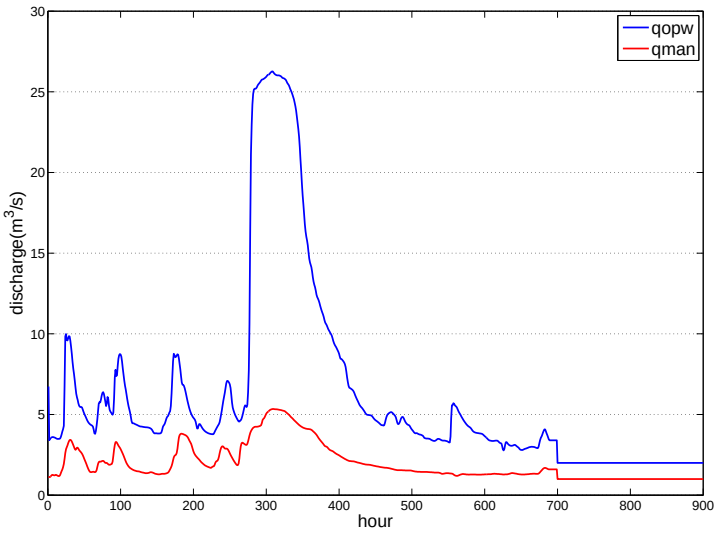


Figure 3.5: Historical rainfall-runoff discharges of the historical flood event of 1998

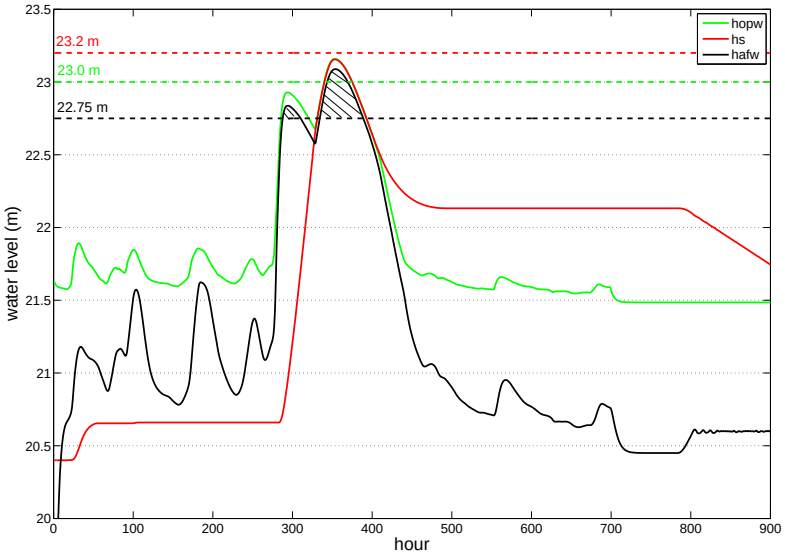


Figure 3.6: Regulation of the reduced area conceptual model with three-position controller

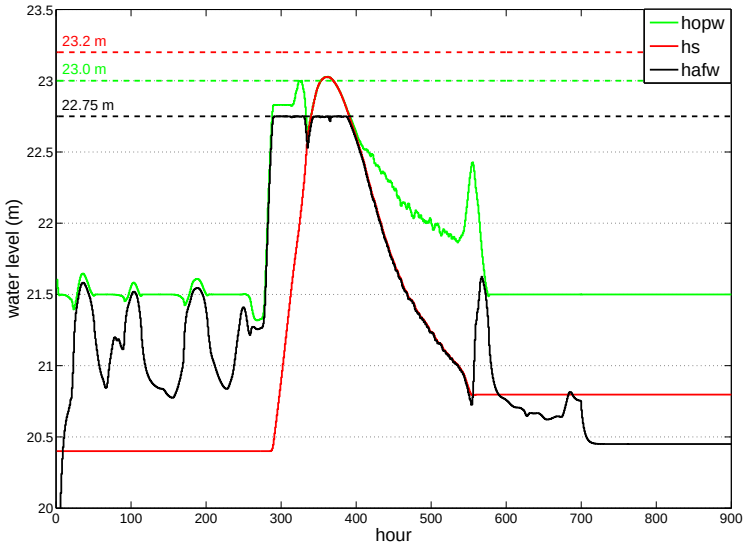


Figure 3.7: Regulation of the reduced area conceptual model with MPC

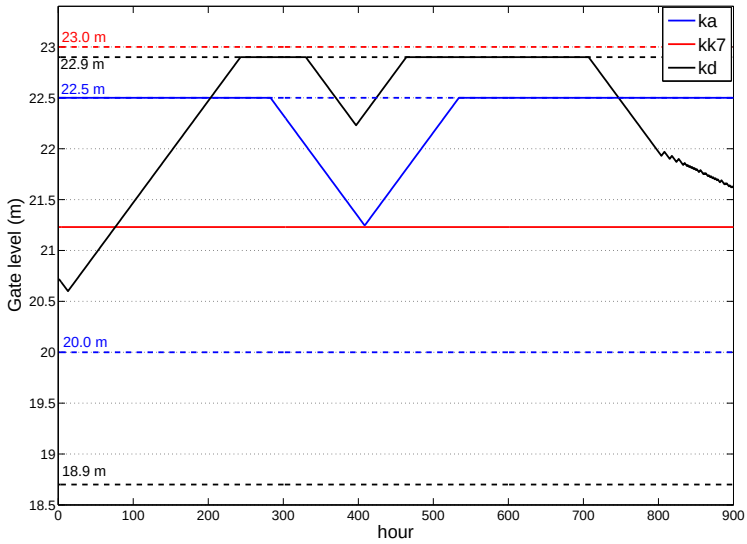


Figure 3.8: Gate positions for the reduced area conceptual model controlled by three-position controller

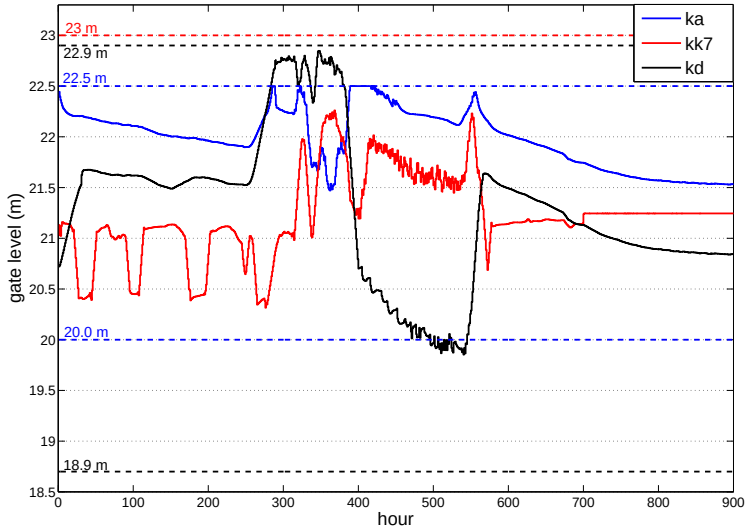


Figure 3.9: Gate positions for the reduced area conceptual model controlled by MPC

In order to make a quantitative comparison between both controllers, the regulation and flood cost is calculated and summarized in table 3.1. Note that the regulation cost is only calculated for the period between hour 1 and hour 250 and the period between hour 580 and hour 900. The reason for this is that between hour 250 and 580 a big peak in the rainfall-runoff discharges occur which makes the regulation cost during that period irrelevant. Also note that the flood cost of h_{opw} is not the real flood cost (as h_{opw} does not flood) but it is the cost associated to the violation of the warning level of $23m$. A backslash in the table means that there is no cost involved. The following observations can be made from the table:

- The regulation cost for h_{opw} is lower for MPC than for the three-position controller.
- The flood cost (w.r.t. the violation of the warning level) for h_{opw} is much lower for MPC than for the three-position controller. Therefore, less farm land is needed to avoid flooding with MPC.
- For MPC there is no flood cost for water level h_{afw} as MPC avoids flooding of h_{afw}. However, for the three-position controller h_{afw} floods and its flood cost is found in the table.

Both the figures as the tables show that MPC outperforms the three-position controller for the historical simulation of 1998. (Note: These results were published in [26]).

Cost simulation 1998	MPC	three-position controller
$log\left(J_{reg}(h_{opw})\right)$	-0.74	2.50
$log\left(J_f(h_{opw})\right)$	-5.01	-0.76
$log\left(J_f(h_s)\right)$	$\backslash$	$\backslash$
$log\left(J_f(h_{afw})\right)$	$\backslash$	1.60

Table 3.1: Comparison of regulation and flood cost of the three-position controller and MPC for the historical flood event of 1998.

3.3.2.2 Simulation with 2 successive rainfall peaks

In this simulation a fictitious rainfall-runoff event is created. The fictitious rainfall-runoff discharges are plotted in figure 3.10. The first 500 hours coincide with the rainfall-runoff discharges of the historical event of 1998. The second peak is fictitious and is 90% of the first peak. After the second peak the rainfall-runoff discharges are assumed to be constant. In figures 3.11 and 3.12 this event is simulated for the three-position controller and MPC respectively. The following observations can be made from these simulations:

- During periods of moderate rainfall-runoff discharges MPC steers water level h_{opw} much closer to the reference level of $21.5m$ than the three-position controller.

- After both rainfall-runoff discharges peaks it can be observed that MPC manages to empty the reservoir much faster than the three-position controller. Moreover, MPC uses less of the storage capacity available in the flood reservoir.

- MPC is capable of keeping h_{afw} under its flood level of $22.75m$. At the other hand, the simulation with the three-position controller shows that water level h_{afw} surpasses its flood level several times. The flooding is indicated on the figure with hatched areas.

- With MPC the water level h_{opw} only violates its warning level during the first peak. With the three-position controller h_{opw} violates its warning level during both peaks and therefore uses more farm land than the MPC controller.

- In figures 3.13 and 3.14 the control actions of the 3 gates are depicted respectively for the three-position controller and MPC. It can be seen from the figures that for both controllers the gates do not violate their physical upper and lower limit. Though difficult to see in the pictures the gates also do not move faster than $0.1m/hour$.

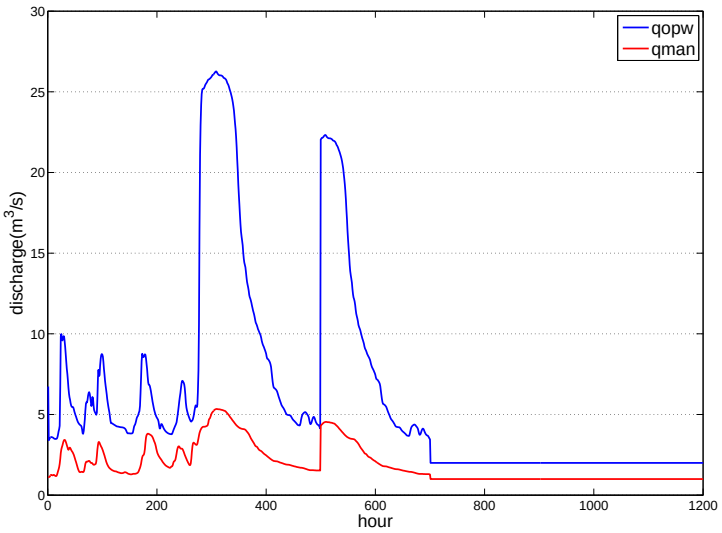


Figure 3.10: Fictitious rainfall-runoff discharges with 2 big rainfall-runoff peaks

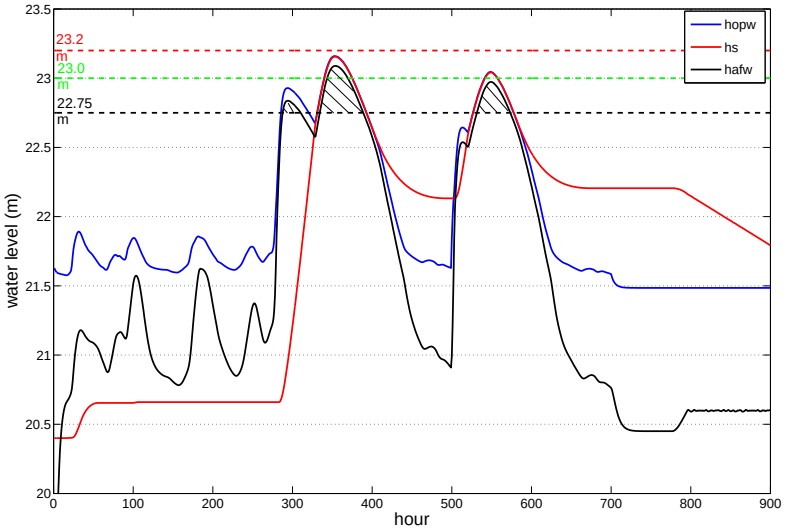


Figure 3.11: Regulation of the reduced area conceptual model with the three-position controller on fictitious event with 2 big rainfall-runoff peaks.

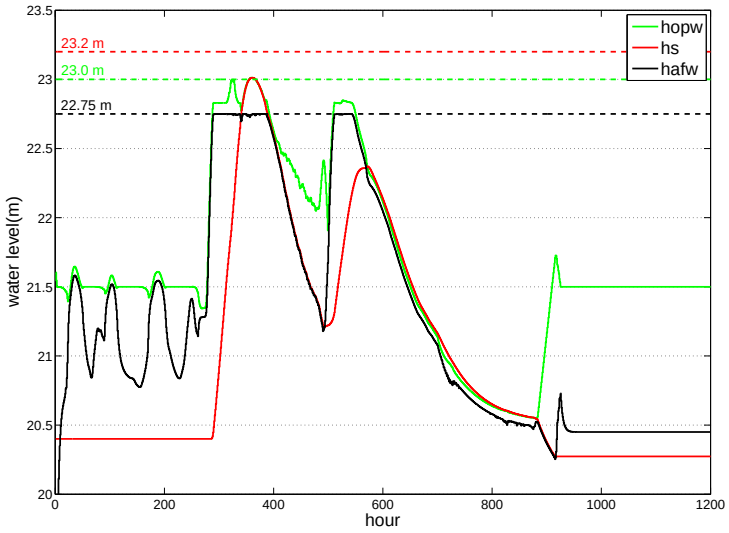


Figure 3.12: Regulation of the reduced area conceptual model with MPC on historical event with 2 big rainfall-runoff peaks.

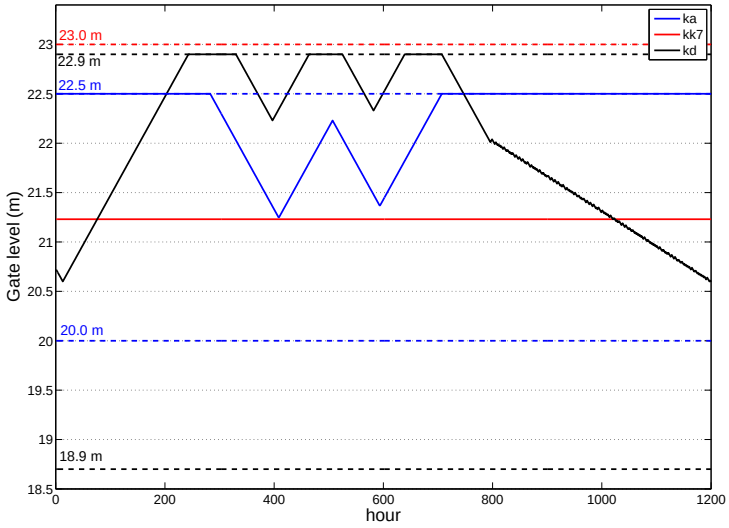


Figure 3.13: Gates with three-position controller for fictitious event with 2 big rainfall-runoff peaks.

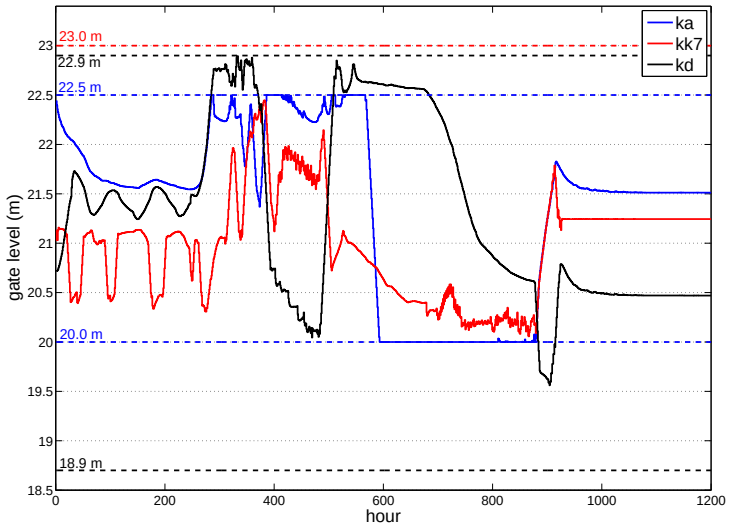


Figure 3.14: Gates with MPC for fictitious event with 2 big rainfall-runoff peaks.

In order to make a quantitative comparison between both controllers, the regulation and flood cost is calculated and summarized in table 3.2. Note that the regulation cost is only calculated for the period between hour 1 and hour 250 and the period between hour 920 and hour 1200; this because the regulation cost is irrelevant during the period of the big rainfall-runoff peak. Similar to the flood cost of h_{opw} for the historical event of 1998 the flood cost of water level h_{opw} is the cost associated to the violation of the warning level of $23m$. A backslash in the table again means that there is no cost involved. The following observations can be made from the table:

- The regulation cost for h_{opw} is lower for MPC than for the three-position controller.
- The flood cost (w.r.t. the violation of the warning level) for h_{opw} is much lower for MPC than for the three-position controller. Therefore, less farm land is needed to avoid flooding with MPC.
- For MPC there is no flood cost for water level h_{afw} as MPC avoids flooding of h_{afw}. However, for the three-position controller h_{afw} floods and its flood cost is found in the table.

Again these simulations show that MPC outperforms the three-position controller.

Cost simulation 2 peaks	MPC	three-position controller
$log\left(J_{reg}(h_{opw})\right)$	-0.64	2.44
$log\left(J_f(h_{opw})\right)$	-6.94	-0.72
$log\left(J_f(h_s)\right)$	\	\
$log\left(J_f(h_{afw})\right)$	\	1.45

Table 3.2: Comparison of regulation and flood cost of the three-position controller and MPC for the fictitious event with 2 big rainfall-runoff peaks.

3.3.3 Experimental results on the large area

The large area considered here corresponds to the area presented in Fig. 2.10. The objectives of the controller are summarized in the following:

- Steer h_{opw} and hbg_{opw} as close as possible to respectively 21.5 m and 23.8 m during normal operation.
- Gates cannot move at a rate faster than 0.1 m/hour.
- For each gate an upper and a lower limit is taken into consideration.
- The guard levels for the reservoirs h_s and h_w are equal to 23 m and 22 m.
- The safety limit for h_{opw} is equal to 23 m.

Table 3.3: Flooding levels in the Demer basin

water level	flood level (m)
h_{opw}	23.2
h_s	23.2
h_1	24.3
h_{vg}	22.52
hs_2	22.75
h_3	22.75
h_4	22.75
h_{zw}	21.6
h_h	23.6
h_2	22.7
h_3	22.9
h_v	23.5
h_w	22.4
h_{gl}	22
hbg_{opw}	24.8
h_{lg}	23
h_{bg}	24.2
h_{afw}	20.5
h_4	22.5
h_{zb}	21.5

- The flooding levels of all the water levels in the river system are shown in table 3.3.

Simulations were done with historical rainfall-runoff data from 1998. In fig. 3.15-3.19 a simulation with the three-position controller is presented. In Fig. 3.20-3.24 the same simulation is performed but with the NMPC scheme of fig. 3.4. In both figures the different flooding levels are indicated as horizontal lines. Comparing both results leads to following conclusions:

- The NMPC controller manages to steer h_{opw} much closer to its setpoint. The three-position controller regulates hbg_{opw} slightly closer to its setpoint than the NMPC controller.

- The three-position controller severely violates the flood level constraints for water levels h_{vg}, h_2 and h_{gl}. With the NMPC controller these water levels also violate their flood levels but the violation is not as severe and lasts for a much shorter period of time.

- The three-position controller severely violates the flood level constraints for water level h_w. With the NMPC controller this constraints is also violated but

only slightly and for a very short period of time.

- The three-position controller slightly violates the flood level constraints for water levels h_{zw} and h_{zb}. The NMPC controller does not violate these constraints.

- After the rainfall peak the NMPC controller manages to empty the reservoir h_s much faster than the three-position controller which is an advantage if a second rainfall peak succeeds the first one.

In tables 3.4 and 3.5 a quantitative comparison is made between both controllers and similar conclusions can be made as those following from the figures. From table 3.5 it follows that the reduction in flooding due to the NMPC controller is significant. The rainfall in 1998 was too severe for the NMPC controller to avoid flooding. Nevertheless, the NMPC controller outperforms the three-position controller.

Remark 3.5. *The QP's solved at each iteration were solved by the Matlab toolbox MOSEK [5]. The time needed to solve* 1 *QP was between* 10 *to* 15 *seconds. At maximum* 10 *QP iterations were allowed at each sampling time, which means that the maximum computation time was* 150 *seconds.*

Table 3.4: Comparison of the regulation cost of the 2 controllers during the periods before and after the flood event.

	three-position controller	NMPC
$J_{reg}\left(h_{opw}\right)$	16.08	0.45
$J_{reg}\left(hbg_{opw}\right)$	0.16	0.81

Table 3.5: Comparison of the flood cost of the 2 controllers.

	three-position controller	NMPC
$J_f\left(h_2\right)$	3.00	1.67
$J_f\left(h_w\right)$	1.48	0.01
$J_f\left(h_{gl}\right)$	9.47	1.78
$J_f\left(h_{vg}\right)$	13.02	2.87
$J_f\left(h_{zw}\right)$	9.8×10^{-4}	0.0×10^{-4}
$J_f\left(h_{zb}\right)$	0.01	0.00

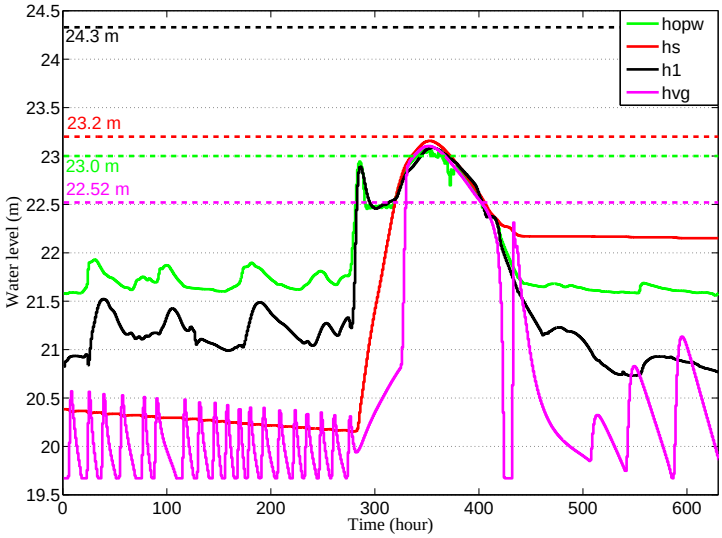


Figure 3.15: Regulation with three-position controller (1)

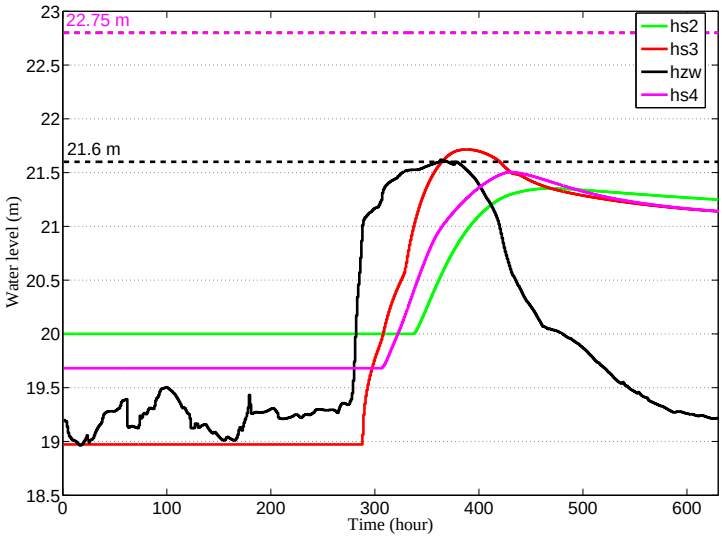


Figure 3.16: Regulation with three-position controller (2)

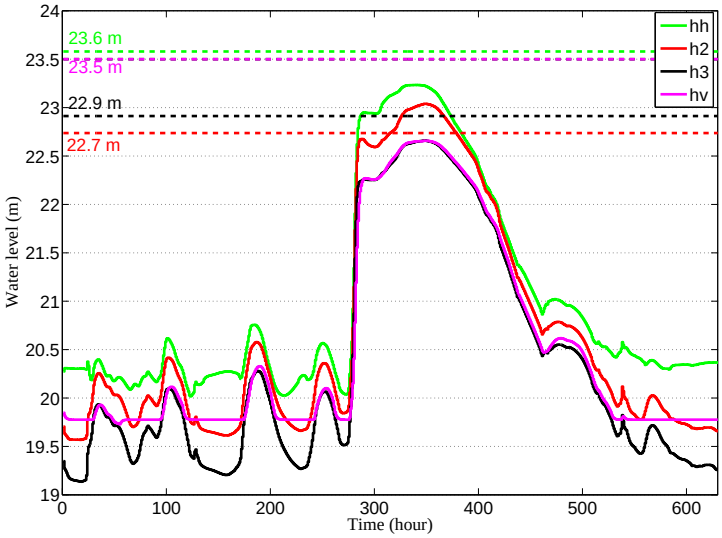


Figure 3.17: Regulation with three-position controller (3)

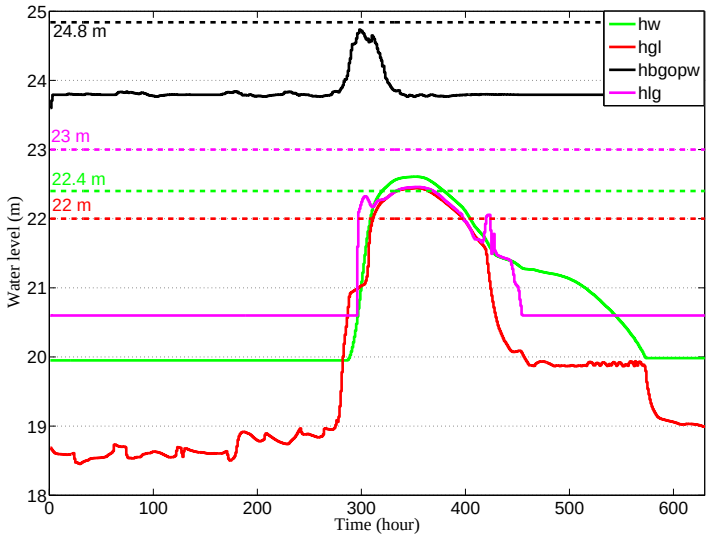


Figure 3.18: Regulation with three-position controller (4)

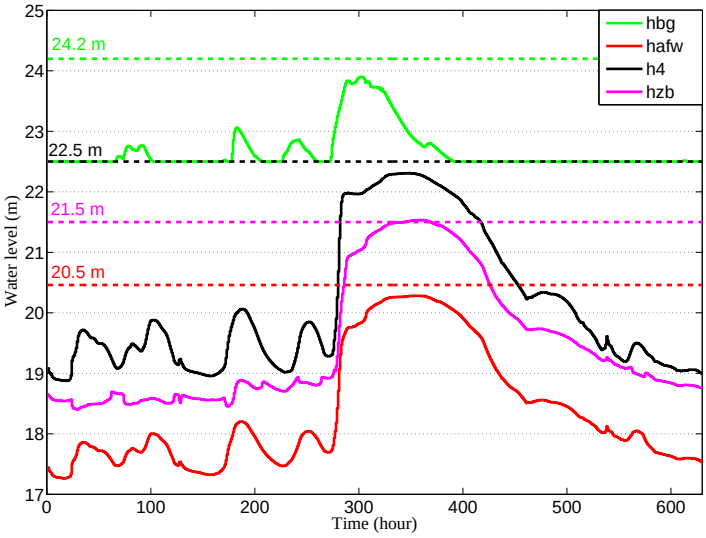


Figure 3.19: Regulation with three-position controller (5)

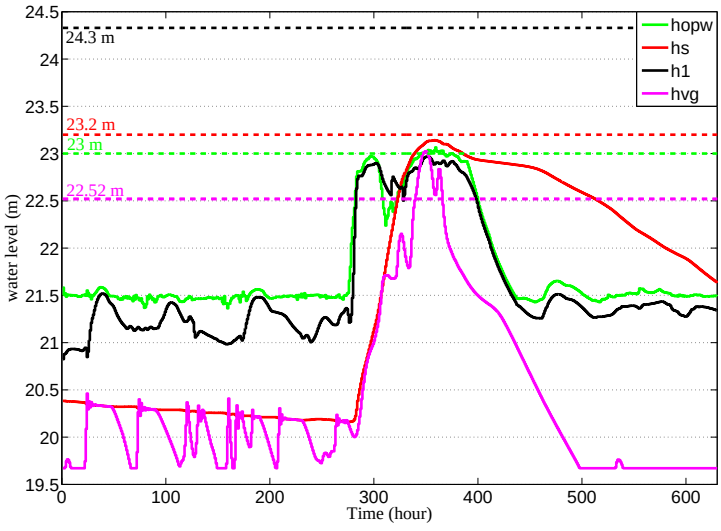


Figure 3.20: Regulation with NMPC controller under certain conditions (1)

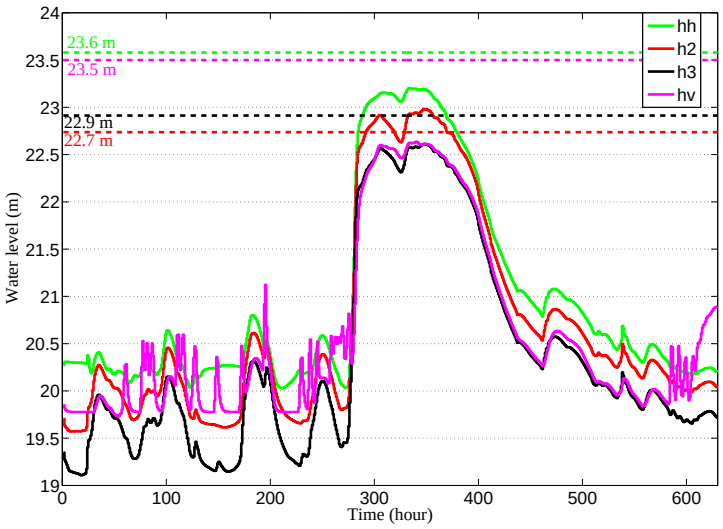


Figure 3.21: Regulation with NMPC controller under certain conditions (2)

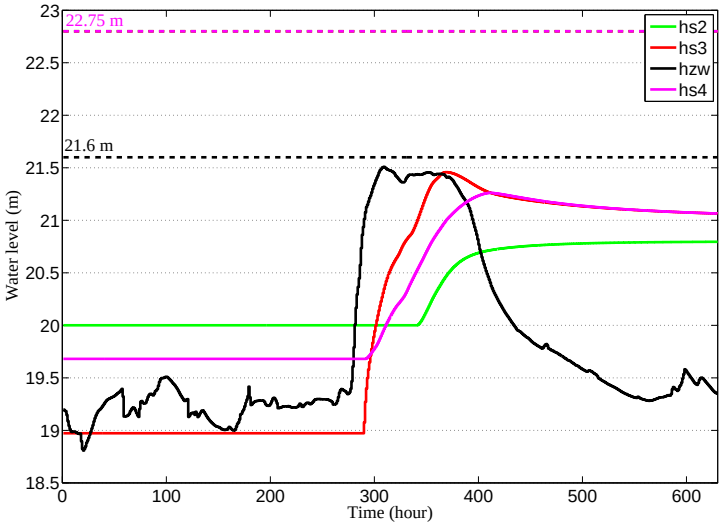


Figure 3.22: Regulation with NMPC controller under certain conditions (3)

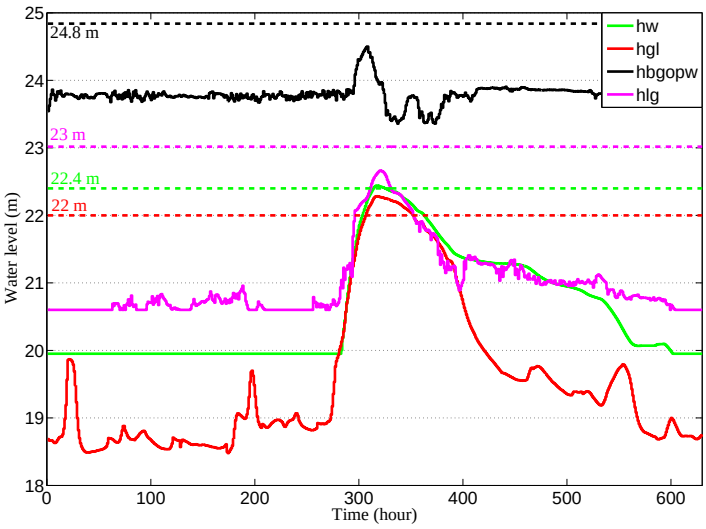


Figure 3.23: Regulation with NMPC controller under certain conditions (4)

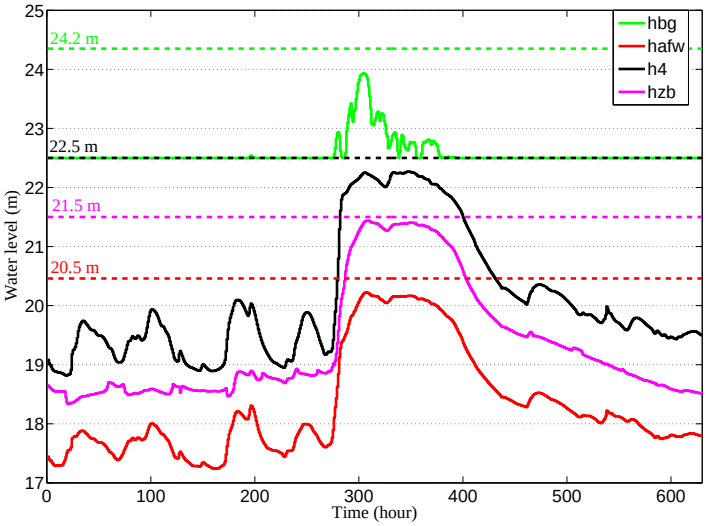


Figure 3.24: Regulation with NMPC controller under certain conditions (5)

3.4 Uncertainty

MPC uses rainfall-runoff predictions in order to make a prediction of the future water levels and based on these predictions a suitable control action is performed. An important point that needs to be considered is the fact that these rainfall-runoff predictions never coincide with the real rainfall-runoff. So the predictions made by MPC are always going to deviate from the real future water levels. However, one of the big advantages of MPC is that this control strategy is based on a receding horizon strategy. This means that at each sampling time the new state of the system is measured or estimated; based on this measured/estimated state a new prediction is made. Because of this state update that happens at each sampling time, MPC is inherently robust against the uncertainties in the rainfall prediction. In order to show this the simulation for MPC is going to be performed by adding a realistic amount of uncertainty to the rainfall-runoff discharges predictions.

The uncertainty of the rainfall-runoff discharge predictions varies within the prediction horizon. The further ahead in time the larger the uncertainty on the rainfall-runoff predictions. In these simulations it is assumed that the rainfall-runoff uncertainty at the current time step is 10% and that this uncertainty increases with 0.2% for each hour further ahead in the prediction. This means that if the prediction horizon is 30 hours then at the end of the prediction window the rainfall-runoff uncertainty is equal to 16%. These values were estimated by the hydraulic laboratory of the K.U.Leuven based on historical data by means of the techniques described in [111]. In this section it is assumed that the rainfall-runoff predictions used by MPC for calculating an optimal input sequence are an overestimation of the real rainfall-runoff of 1998. This is done

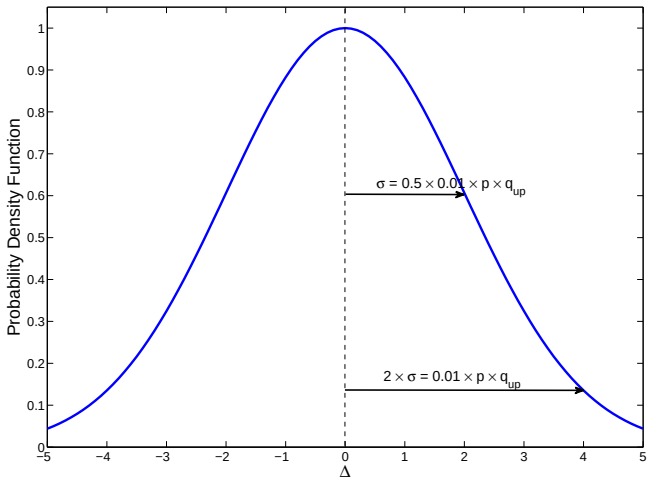


Figure 3.25: Gaussian uncertainty added to the predicted rainfall-runoff discharge

in the following way:

- Within the prediction horizon N the maximum uncertainty Δ_{max} is calculated as $p\%$ of the real rainfall-runoff discharge q_{up} of 1998. For time step k the percentage p is equal to 10%. For the next time steps it follows that the percentage $p(k+i)$ at time step $k+i$ is equal to $p(k+i-1)+0.2$.

- Based on this value for the maximum uncertainty Δ_{max} the uncertainty Δ is determined by performing a random sampling from a Gaussian distribution with mean 0 and standard deviation σ satisfying $2 \times \sigma = \Delta_{max}$. This distribution is plotted in figure 3.25. Note that it is not checked whether in real-life the Gaussian assumption is valid. However, because we only want to verify the robustness of the model predictive controller the exact distribution is not of importance.

- In order to ensure that an overestimation of q_{up} is obtained, the absolute value of Δ is added to q_{up}.

The result of this simulation with MPC is plotted in figures 3.26-3.30. Comparing these figures with the results of the three-position controller displayed in figures 3.15-3.19 the following conclusions can be derived:

- Despite the uncertainties in the rainfall predictions the NMPC controller steers the water level h_{opw} much closer to its setpoint than the three-position controller. On the other hand, the three-position controller regulates water level hbg_{opw} slightly closer to its setpoint than the NMPC controller.

- Both the three-position controller and the NMPC controller violate the flood level constraint for water level h_2. The amount of violation is comparable for both controllers.

- Both the three-position controller and the NMPC controller violate the flood level constraint for water level h_{vg}. However, the violation is significantly bigger for the three-position controller.

- The three-position controller significantly violates the flood level constraints for water levels h_{gl} and h_w. The NMPC controller does not violate these flood level constraints.

- The three-position controller slightly violates the flood level constraints for water levels h_{zw} and h_{zb}. The NMPC controller does not violate these flood level constraints.

In tables 3.6 and 3.7 a quantitative comparison is made between the three-position controller and the NMPC controller. From both tables it can be derived that despite the uncertainties in the rainfall prediction the NMPC controller still outperforms the three-position controller.

Table 3.6: Comparison of the regulation cost of the three-position controller with the NMPC controller under uncertain conditions during the periods before and after the flood event of 1998.

	three-position controller	NMPC
$J_{reg}\left(h_{opw}\right)$	16.08	2.00
$J_{reg}\left(hbg_{opw}\right)$	0.16	1.03

Table 3.7: Comparison of the flooding cost of the three-position controller and the NMPC controller under uncertain conditions for the historical flood event of 1998.

	three-position controller	NMPC
$J_f\left(h_2\right)$	3.00	3.09
$J_f\left(h_w\right)$	1.48	0.00
$J_f\left(h_{gl}\right)$	9.47	0.00
$J_f\left(h_{vg}\right)$	13.02	6.33
$J_f\left(h_{zw}\right)$	9.8×10^{-4}	0.0×10^{-4}
$J_f\left(h_{zb}\right)$	0.01	0.00

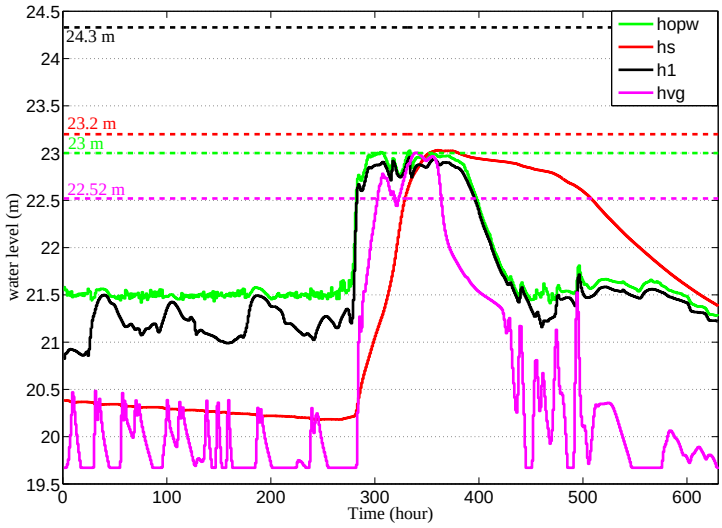


Figure 3.26: Regulation with NMPC controller under uncertain conditions (1)

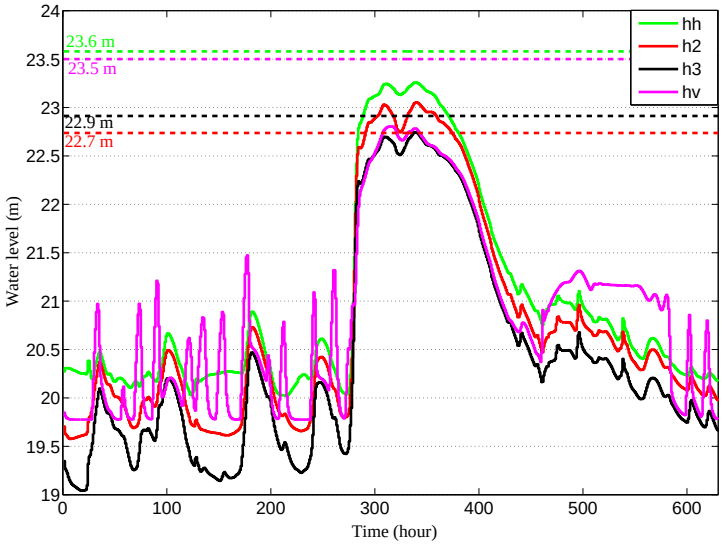


Figure 3.27: Regulation with NMPC controller under uncertain conditions (2)

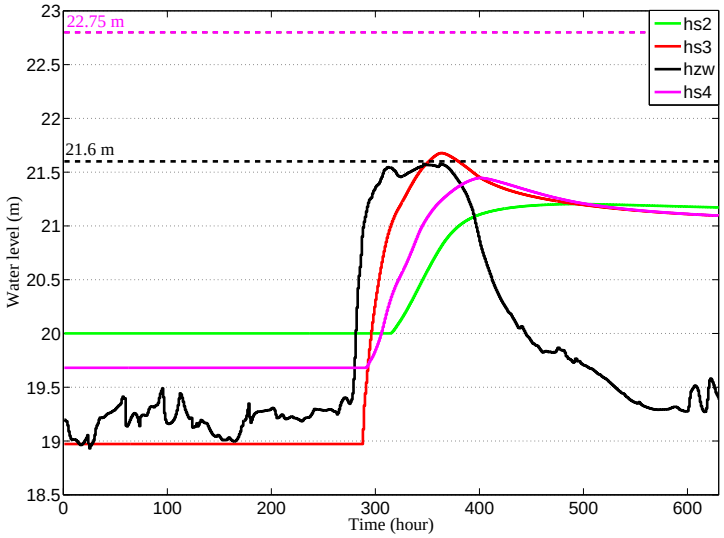


Figure 3.28: Regulation with NMPC controller under uncertain conditions (3)

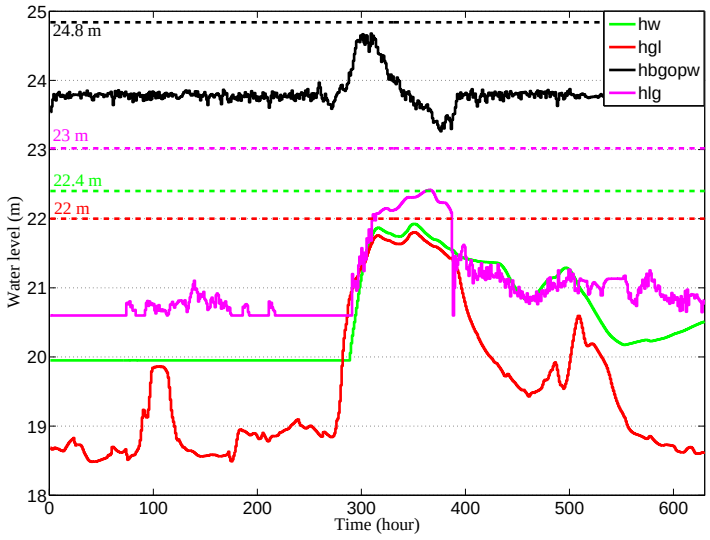


Figure 3.29: Regulation with NMPC controller under uncertain conditions (4)

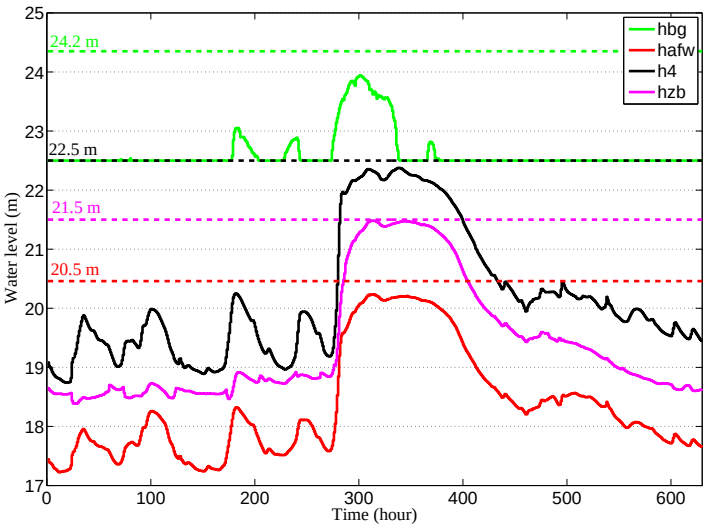


Figure 3.30: Regulation with NMPC controller under uncertain conditions (5)

3.5 Conclusions

A nonlinear MPC scheme is developed. The proposed nonlinear MPC scheme is a nonlinear optimization scheme that finds a local optimal solution by means of SQP. By solving a quadratic program combined with a line-search at each iteration a new solution is determined that has a value for the cost function lower or equal to that of the solution at the previous iteration. Besides that also a cost function and constraint strategy is outlined that has to cope with situations in which the optimization problem turns out to be unfeasible due to excessive rainfall. The proposed MPC scheme also deals with the uncontrollable modes of the gates by steering the gates outside of these modes.

This nonlinear MPC scheme is first tested on the 'reduced area' model and compared with the tree-position controller. The first simulation concerns the historical rainfall event of 1998. The simulation shows that the proposed MPC scheme satisfies all the physical constraints of the gates. Moreover MPC regulates h_{opw} and hbg_{opw} much closer to their corresponding set points. The MPC scheme avoids flooding of the 'reduced area' model and empties reservoir h_s much faster than the three-position controller. In a second simulation with the 'reduced area' model a fictitious rainfall event was created by adding a second rain peak at the end of the historical rainfall event of 1998. Again, the simulation shows that the MPC controller outperforms the three-position controller w.r.t. set point regulation as well as flood prevention.

In a next step the nonlinear MPC scheme is tested on the complete area. In a first

simulation the historical rainfall event of 1998 is simulated and it is assumed that the rainfall-runoff predictions are exact. This simulation shows that the MPC controller in general regulates the water levels closer to the set points than the three-position controller. It is also clear from the simulation that with the MPC scheme the amount of flooding decreases significantly compared with the simulations with the three-position controller. In a next simulation uncertainty was added to the rainfall-runoff predictions of 1998. Despite the uncertainties the simulation shows that MPC still achieves much better performance than the three-position controller w.r.t. set point regulation and flooding reduction.

Chapter 4

State Estimation

In the previous chapter a nonlinear MPC scheme was proposed for flooding regulation. In order for this scheme to be able to make predictions it needs to know the current state of the system. This state can be estimated based on water level measurements by a state estimator. In this chapter the implementation of such a state estimator is central. The chapter starts in Section 4.2 with an overview of different state estimators that can be used for processes with Gaussian noise. From all these possible state estimators it will be shown that a moving horizon estimator is the best choice for the application in this thesis. Next, in Section 4.3 an overview is given of the state of the art w.r.t. state estimation in river control applications. In Section 4.4 the nonlinear MHE scheme is outlined that is used in this thesis and in Section 4.5 experiments are performed in which the NMHE scheme is combined with the NMPC scheme and the performance is again compared with that of the three-position controller. The chapter ends with conclusions in Section 4.6.

4.1 Introduction

In the previous chapter a nonlinear MPC scheme was implemented and tested on the nonlinear Demer model for the historical rainfall-runoff data of 1998. In these simulations it was assumed that at each sampling time the full state of the system was exactly known. However, this is never the case in practice. In practice, it is not always possible to measure all the states. And even if possible, it is usually very costly to add a sensor for each state. Therefore, in control applications only a subset of the outputs is measured and based on those measurements a state estimation is performed. The state estimator determines the most probable value of the system state based on the given measurements and the assumed underlying model of the dynamic system. The estimated state is then passed to the controller in order to determine an appropriate control action.

In this chapter a moving horizon estimator for the nonlinear Demer model will be discussed. It will be discussed how such an estimator can be implemented for a nonlinear system. This estimator will then be combined with the NMPC scheme of Chapter 3 and tested on the nonlinear model for the historical rainfall runoff data of 1998. It will be assumed that only the water levels are measured by sensors. In order to verify the robustness of the estimator Gaussian measurement noise is added to the measurements. The result of this simulation will then be compared with that of the currently used three-position controller. It will be shown that the combination of MPC-MHE outperforms the three-position controller.

4.2 State estimators

A state estimator estimates the current state of the system based on past and current output measurements. In order to do this the state estimator needs an internal model of the underlying process it is trying to estimate. Estimators for linear systems typically use the following internal linear state space model

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{4.1}$$
$$y_k = Cx_k + v_k, \tag{4.2}$$

while estimators for nonlinear systems use the following nonlinear state space model

$$x_{k+1} = f(x_k, u_k) + w_k \tag{4.3}$$
$$y_k = Cx_k + v_k, \tag{4.4}$$

with $f(.,.)$ a nonlinear function. The vectors $w_k \in \mathbb{R}^{n_x}$ and $v_k \in \mathbb{R}^{n_o}$ represent process and measurement noise, respectively. In the simulations in this chapter the process and measurement noise added to the system, are assumed to be white with normal probability distributions described by

$$p(w) \sim N(0, Q) \tag{4.5}$$
$$p(v) \sim N(0, R) \tag{4.6}$$

with matrices Q and R covariance matrices. There exist several different types of estimators that can be used for models with Gaussian noise. In the remainder of this section the most relevant of them will be discussed in more detail.

Remark 4.1. *The main goal of this chapter is to test the robustness of the MPC scheme w. r. t. the extra uncertainty added by the estimation process. In order to verify the robustness of the MPC scheme, the exact noise distribution is not relevant. We have chosen to add Gaussian process and measurement noise in the simulations of this chapter because in many practical applications the noise can be assumed Gaussian.*

4.2.1 State estimators for processes with Gaussian noise

In this subsection, an overview is given of the most important state estimators that can be used for processes with Gaussian noise. There are many other state estimators available that can be used but they will not be discussed here.

4.2.1.1 Kalman filter

The Kalman filter is one of the greater discoveries in the history of statistical estimation theory [86]. Its most immediate applications have been for the control of complex dynamic systems. The Kalman filter has been succesfully applied for control purposes in areas as diverse as aerospace, marine navigation, nuclear power plants, manufacturing and many others. The Kalman filter has also been succesfully used for predicting the likely future courses of dynamic systems that people are not likely to control like the trajectories of celestial bodies, the prices of traded commodities or space weather [60].

The Kalman filter is only applicable to linear systems. Its internal model is described by equations (4.1)-(4.2). Based on this internal model, together with the current measurement and the normal probability distributions of the process noise and measurement noise the Kalman filter estimates the current state x_k by means of the following equation

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \left(y_k - \hat{y}_{k|k-1} \right) \tag{4.7}$$

with $K_k \in \mathbb{R}^{n \times m}$ and output

$$\hat{y}_{k|k-1} = C\hat{x}_{k|k-1}. \tag{4.8}$$

In order to obtain recursively an optimal estimator K_k, the first step of the procedure is to propagate ahead $x_{k-1|k-1}$ via (4.1)

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1}, \tag{4.9}$$

which is called the time-update step. Inserting (4.9) into (4.7) leads to the following state estimator form

$$\hat{x}_{k|k} = A\hat{x}_{k-1|k-1} + Bu_{k-1} + K_k \left(y_k - C \left(A\hat{x}_{k-1|k-1} + Bu_{k-1} \right) \right). \tag{4.10}$$

The value of K_k is then chosen such that it minimizes the mean square estimation error (for mathematical details see [28]).

4.2.1.2 Extended Kalman filter

A large class of estimation problems involve nonlinear models such as the ones described in (4.3)-(4.4). The standard Kalman filter cannot be applied for such type

of systems. A typical way to extend the standard Kalman for linear systems to the nonlinear case is by linearizing the nonlinear system. There are many ways to produce such linearized versions but the most commonly used version is the extended Kalman filter. The extended Kalman filter, though not precisely optimal, has been successfully applied to many nonlinear systems. The fundamental concept of this filter involves the notion that the true state is sufficiently close to the estimated state. In that case the error dynamics can be represented fairly accurately by a linearized first-order Taylor series expansion around the current estimate $\hat{x}_{k-1|k-1}$

$$f(x_{k-1}, u_{k-1}) \approx f(\hat{x}_{k-1|k-1}, u_{k-1}) + A_{k-1}(x_{k-1} - \hat{x}_{k-1|k-1}), \tag{4.11}$$

with

$$A_{k-1} = \frac{\partial f}{\partial x}\left(\hat{x}_{k-1|k-1}, u_{k-1}\right). \tag{4.12}$$

The nonlinear system equations (4.3)-(4.4) can then be approximated by the following linear time-varying system

$$\begin{aligned} x_k &= A_{k-1}x_{k-1} + \bar{u}_{k-1} + w_k & (4.13)\\ y_k &= Cx_k + v_k, & (4.14) \end{aligned}$$

with $\bar{u}_{k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}) - A_{k-1}\hat{x}_{k-1|k-1}$. This is a linear time-varying system and the state x_k can therefore be estimated by the standard Kalman filter for linear systems. Though the extended Kalman filter can give reasonable performance its main disadvantage is that the filter may quickly diverge if the initial estimate of the state is wrong or if the model is not accurate enough.

4.2.1.3 Ensemble Kalman filter

The ensemble Kalman filter (EnKF) [67] approximates the error covariance matrix based on a Monte Carlo approach. The starting point of the EnKF is an ensemble of state estimates that tries to capture the probability density function of the initial state. The ensemble of estimates is then propagated through the nonlinear model and the error covariance matrix of the actual state is approximated from the ensemble of estimates. Now, assume at time k we have an ensemble of forecasted states $X_{k|k-1}$ defined by

$$X_{k|k-1} = \left(\hat{x}^1_{k|k-1}, \ldots, \hat{x}^N_{k|k-1}\right), \tag{4.15}$$

where the superscript denotes the ensemble member. The EnKF then performs an ensemble of parallel data assimilation cycles for $i = 1, \ldots, n$ as follows:

$$\hat{x}^i_{k|k} = \hat{x}^i_{k|k-1} + K_k\left(y^i_k - C\hat{x}^i_{k|k-1}\right). \tag{4.16}$$

The value of K_k is calculated by means of the standard Kalman filter update with the approximated error covariance matrices. The error covariance matrices are

approximated from the ensemble $X_{k|k-1}$. An interesting advantage of the EnKF is that it is not necessary to explicitly form the error covariance matrices in the calculations of K_k. Therefore, the EnKF can be used in systems with very high state dimensions. Other advantages of the EnKF are that it is relatively simple to implement and well suited for highly nonlinear models. The disadvantage of the EnKF is that it is a sample-based estimator and therefore leads to a suboptimal solution.

4.2.1.4 Unscented Kalman filter

The unscented Kalman filter (UKF) is founded on the intuition that it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation [68]. First $2n_x + 1$ points, called sigma points, are drawn from the prior distribution of the random variable that needs to be estimated, with n_x the dimension of the state space. The set of $2n_x + 1$ sigma points with their corresponding weights is denoted as X_{k-1}:

$$X_{k-1} = \left\{ \left(x_{k-1}^j, W^j \right) | j = 0, \ldots, 2n_x \right\}. \tag{4.17}$$

The set of sigma points is then propagated through the nonlinear process model

$$x_k^{f,j} = f(x_{k-1}^j) \tag{4.18}$$

as well as through the observation model

$$y_{k-1}^{f,j} = C x_{k-1}^j. \tag{4.19}$$

The transformed points are used to compute the mean of the forecast value of x_k and the mean of the transformed observations:

$$x_k^f = \sum_{j=0}^{2n_x} W^j x_{k-1}^{f,j} \tag{4.20}$$

$$y_k^f = C x_k^f. \tag{4.21}$$

In a similar way, the transformed points are also used to compute the relevant covariance matrices of the transformed points and observations. The estimated state $\hat{x}_{k|k}$ is then determined as

$$\hat{x_{k|k}} = x_k^f + K_k \left(y_k - y_k^f \right). \tag{4.22}$$

The matrix K_k is calculated based on the updated covariance information of the transformed points and observations. For more details we refer to [120].

4.2.1.5 Moving horizon estimator

This state estimator was already introduced and briefly discussed in Chapter 1. Based on the past and current obtained output measurements $y_{k-N_e+1}, \ldots, y_k$, at time step k,

the moving horizon estimator (MHE) tries to estimate the current state of the system. The moving horizon estimator achieves this by solving *on-line* a constrained least squares problem. In order to keep this problem tractable the optimization problem is limited to the estimation of the states within a predefined estimation window. In section 4.4 the moving horizon estimator will be outlined in more detail.

4.2.2 Estimator choice

In the previous subsection an overview was given of several state estimators that have been successfully applied in many practical control applications with Gaussian process and measurement noise. In Figure 4.1 a visual overview is given of these estimators. The discussed estimators can be divided into those that are applicable for linear systems and those applicable for nonlinear systems. Note that the nonlinear state estimators can always be used for linear systems. However, this is typically avoided because the standard Kalman filter returns the optimal state estimation for linear systems with Gaussian noise. In the intersection of the linear and nonlinear state estimators there is the moving horizon estimator and the ensemble Kalman filter. Both types of estimators are used in linear as well as nonlinear systems.

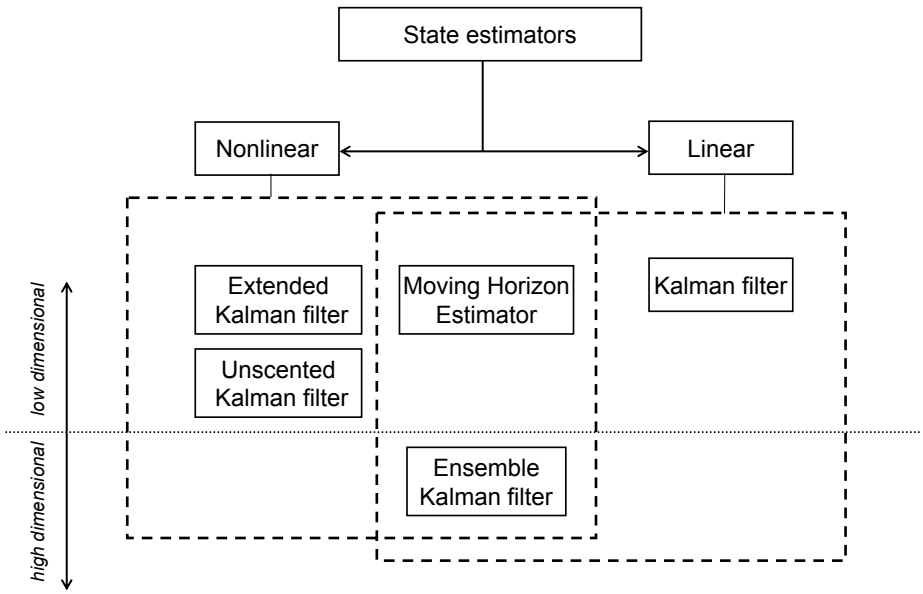


Figure 4.1: State estimators for processes with Gaussian process and measurement noise.

In order to estimate the states in the Demer basin a choice has to been made between one of these estimators. The standard Kalman filter cannot be used because the Demer model is highly nonlinear. The most straightforward choice would be the extended Kalman filter, but it is known that this filter diverges fast if the initial state estimate is wrong or if the model is not accurate enough. Since our model is only a simplification of reality this might lead to problems. A better choice would be the unscented Kalman filter or the ensemble Kalman filter. However, both estimators are based on sampling, which can lead to suboptimal solutions. Therefore, the best choice for this application is a MHE. A MHE can be used for both linear as nonlinear models and by a proper design of the cost function MHE usually leads to more optimal solutions than the sampling based estimators. Moreover, because MHE solves a constrained optimization problem, constraints on the states can be incorporated which can be important a priori knowledge that the other methods cannot benefit from. A possible drawback of MHE is its computational complexity. However, it was recently shown in [65] that by properly exploiting the structure of the optimization problem, MHE problems can be solved extremely fast.

4.3 State of the art

As already mentioned in chapter 3 all river control applications that can be found in literature concern setpoint regulation of several water levels along a canal. The canal is typically modeled as a series of pools. The purpose is usually to steer the water levels in each pool as close as possible to its setpoint. In some applications there are also lateral offtakes that act as disturbances on the system. The models used in these applications are linear. The use of linear models can be justified by the fact that the dynamics of the river system are linear for small deviations around the setpoint. Typically, the state vector consists of past water levels, the input vector of discharges over the gates and the output vector of the future water levels that need to be controlled. In order to be able to control these systems at each time step the state of the system must be known. In current applications there have been used 2 different approaches to deal with this problem:

- **Sensor measurements:** In this case one simply measures all the water levels in the system and assumes the measurement is noise-free and returns the real state information [61]. In order to avoid high frequency noise or to attenuate the amplification of the critical resonance wave, in some applications a low-pass filter is added [77, 92].

- **Kalman filtering:** In [93] the pools in the controlled canal were modeled more accurately by means of the Saint-Venant equations. Therefore, the state of each pool consisted of 5 water levels from which only 1 was measured, namely the water level at the downstream part of the pool. In order to estimate the unmeasured states a standard Kalman filter for linear systems was introduced to perform a state estimation.

Because of reasons, already explained in chapter 3, in this thesis the discharges are considered as state variables, not as inputs. Since the discharges are not measured in the Demer basin, state estimation is inevitable in this thesis. Also, because the underlying model in this thesis is highly nonlinear the standard Kalman filter cannot be used and a nonlinear state estimator is needed. Nonlinear state estimators have never been used in river control applications. Since all river control applications concern setpoint regulation and therefore the underlying process models are linear, nonlinear estimators have always been unnecessary. However, nonlinear estimators have been successfully used for streamflow forecasting in hydrologic models, especially the ensemble Kalman filter. For example, in [87] and [88] an ensemble Kalman filter was successfully implemented for the purpose of updating model states and boundary conditions. Similar nonlinear data assimilation techniques are also used in the full hydrodynamic Infoworks model used by the VMM in order to be able to accurately predict future floods in the Demer basin. But these works do not concern river control and are only limited towards accurate predictions of future states. Moreover, moving horizon estimation has never been used for data assimilation of river models, nor for state estimation purposes in river control applications. Therefore the results in this chapter are the first results in which a nonlinear moving horizon estimator is combined with a model predictive controller for flood regulation purposes.

4.4 Nonlinear moving horizon estimation

In the following the mathematical formulation of the nonlinear MHE scheme implemented in this thesis will be outlined in more detail. The process model is assumed to be a nonlinear state space model described by

$$x_{k+1} = f(x_k, u_k, d_k) + w_k \tag{4.23}$$
$$y_k = Cx_k + v_k \tag{4.24}$$

with $x_k \in \mathbb{R}^{n_x}$ the state of the system at time step k, $u_k \in \mathbb{R}^{n_u}$ the input of the system consisting of the gate positions at time step k, $d_k \in \mathbb{R}^{n_d}$ the disturbance input containing the rainfall-runoff at time step k and the output $y_k \in \mathbb{R}^{n_o}$ containing the water levels at time step k. The vector $w_k \in \mathbb{R}^{n_x}$ represents process noise and the vector $v_k \in \mathbb{R}^{n_o}$ measurement noise.

Optimization 4.1 (Nonlinear programming problem)**.**

$$\min_{\substack{x_{k-N_e+1},\ldots,x_k \\ w_{k-N_e+1},\ldots,w_{k-1} \\ v_{k-N_e+1},\ldots,v_k}} \begin{array}{l} \left(x^e_{k-N_e+1}\right)^T P^{-1}_{k-N_e+1|k-N_e} \left(x^e_{k-N_e+1}\right) + \\ \sum_{i=0}^{N_e-1} v^T_{k-i} R^{-1} v_{k-i} + \sum_{i=1}^{N_e-1} w^T_{k-i} Q^{-1} w_{k-i} \end{array} \tag{4.25}$$

$$\textit{s.t.} \quad x_{k+i+1} = f(x_{k+i}, u_{k+i}, d_{k+i}) + w_{k+i}, \quad i = -N_e, \ldots, -1 \tag{4.26}$$

$$v_{k+i} = y_{k+i} - Cx_{k+i}, \qquad i = -N_e + 1, \ldots, 0 \quad (4.27)$$

$$x_{k+i} \in \mathbb{X}, \qquad i = -N_e + 1, \ldots, 0 \quad (4.28)$$

with $x^e_{k-N_e+1} = x_{k-N_e+1} - x_{k-N_e+1|k-N_e}$, $x_{k-N_e+1|k-N_e}$ the estimate of the state x_{k-N_e+1} at time $k - N_e$ and N_e the horizon of the moving horizon estimator. The positive definite weighting matrices R^{-1}, Q^{-1} and $P^{-1}_{k-N_e+1|k-N_e}$ are quantitative measures of our confidence in the output model, the dynamic system model and the initial estimate $x^e_{k-N_e+1}$, respectively. At time k the measurements $\{y_{k-N_e+1,\ldots,y_k}\}$ appear explicitly in the moving horizon estimator. The statistics $x_{k-N_e+1|k-N_e}$ and $P_{k-N_e+1|k-N_e}$ represent the information about the states contained in the (previously discarded) measurements $\{y_1, \ldots, y_{k-N_e}\}$. Constraints (4.26) and (4.27) represent the system dynamics of the river model. Constraint (4.28) represents additional constraints on the parameters in order to obtain better and more realistic state estimates. Note that the variable y_{k+i} is assumed to be known as we assume in this thesis that all outputs (this is, the water levels) are measured.

The optimization problem described in optimization 4.1 is a nonlinear constrained programming problem due to the presence of the nonlinear system constraint (4.26). Similar to the nonlinear programming problem for MPC described in chapter 3 this programming problem needs to be solved in an iterative way by solving a sequence of QP's that locally approximate the nonlinear programming problem. This sequence of QP's leads to a sequence of iterates with a decreasing value for the cost function. When convergence is obtained a local solution of the nonlinear programming problem is obtained. In order to generate this sequence of QP's the nonlinear system equation (4.26) is approximated by the following linear time-variant system

$$x_{k+i+1} = A_{k+i}x_{k+i} + b_{k+i}\left(u_{k+i}, d_{k+i}\right) + w_{k+i}, \quad i = -N_e, \ldots, 0 \qquad (4.29)$$

with A_{k+i} the system matrix of the linearized system at time step $k+i$ and b_{k+i} a vector containing constant elements which are dependent on u_{k+i} and d_{k+i}. The system matrix and constant vector are time dependent in order to approximate the nonlinear system behavior as accurate as possible. In order to find such a linear time-varying system at time step k let us denote the estimated optimal state sequence at time step $k - 1$ as

$$x_{k-N_e+1|k-1} \quad \cdots \quad x_{k-2|k-1} \quad x_{k-1|k-1} \qquad (4.30)$$

with $x_{k-N_e+1|k-1}$ denoting the state at time step $k - N_e + 1$ estimated at time step $k - 1$. At time step k, in the MHE framework one tries to estimate the states $x_{k-N_e+2}, \ldots, x_{k-1}, x_k$. A good initial guess for the states $x_{k-N_e+2}, \ldots, x_{k-1}$ can be obtained from the solution at time step $k-1$ as $x_{k-N_e+2|k-1}, \ldots, x_{k-2|k-1}, x_{k-1|k-1}$. A good initial guess for the state x_k can be obtained through the nonlinear system equation (4.26) as

$$x_k = f(x_{k-1|k-1}, u_{k-1}, d_{k-1}) \qquad (4.31)$$

with u_{k-1} the input applied to the system at time step $k-1$ and d_{k-1} the rainfall at time step $k-1$. The nonlinear system equation (4.26) can now be linearized around the following initial guessed state sequence

$$x_{k-N_e+2|k-1} \quad \dots \quad x_{k-1|k-1} \quad f\left(x_{k-1|k-1}, u_{k-1}, d_{k-1}\right) \ , \tag{4.32}$$

which for convenience will be denoted in the sequel as

$$x^0_{k-N_e+2} \quad \dots \quad x^0_{k-1} \quad x^0_k \ . \tag{4.33}$$

This linearization procedure leads to the following linear time-varying system

$$x_{k+i+1} = A_{k+i}x_{k+i} + b_{k+i+1} + w_{k+i} \tag{4.34}$$

for $i = -N_e + 2, \dots, -1$ and with

$$A_{k+i}\left(r,s\right) = \left.\frac{\partial f_{k+i}\left(r\right)}{\partial x\left(s\right)}\right|_{x=x^0_{k+i}}, \tag{4.35}$$

$$f_{k+i} = f\left(x, u_{k+i}, d_{k+i}\right) \tag{4.36}$$

and

$$b_{k+i+1} = f\left(x^0_{k+i}, u_{k+i}, d_{k+i}\right) - A_{k+i}x^0_{k+i}. \tag{4.37}$$

Note that u_{k+i} and d_{k+1} coincide respectively with past inputs and past rainfall-runoff and are therefore known values. Similar to the nonlinear MPC scheme outlined in chapter 3 the system matrix A_{k+i} is computed numerically by means of the following central finite difference scheme:

$$A_{k+i}(:,s) = \frac{f\left(x^0_{k+i} + \Delta x(s), u_{k+i}, d_{k+i}\right) - f\left(x^0_{k+i} - \Delta x(s), u_{k+i}, d_{k+i}\right)}{2\delta} \tag{4.38}$$

with $A_{k+i}(:,s)$ denoting the s-th column of A_{k+i} and $\Delta x(s)$ a vector containing all zeros with exception of the s-th element which is equal to a small perturbation δ. In the nonlinear programming problem of optimization 4.1 the nonlinear system constraints (4.26) are replaced by the linear time-varying system of (4.34) which leads to the following optimization scheme:

Optimization 4.2 (Quadratic programming problem)**.**

$$\begin{aligned} \min_{\substack{x_{k-N_e+1},\dots,x_k \\ w_{k-N_e+1},\dots,w_{k-1} \\ v_{k-N_e+1},\dots,v_k}} \quad & \left(x^e_{k-N_e+1}\right)^T P^{-1}_{k-N_e+1|k-N_e}\left(x^e_{k-N_e+1}\right) + \\ & \sum_{i=0}^{N_e-1} v^T_{k-i}R^{-1}v_{k-i} + \sum_{i=1}^{N_e-1} w^T_{k-i}Q^{-1}w_{k-i} \end{aligned} \tag{4.39}$$

$$\textit{s.t.} \quad x_{k+i+1} = Ax_{k+i} + w_{k+i}, \quad i = -N_e, \dots, -1 \tag{4.40}$$

$$v_{k+i} = y_{k+i} - Cx_{k+i}, \qquad i = -N_e + 1, \ldots, 0 \tag{4.41}$$

$$x_{k+i} \in \mathbb{X}, \qquad i = -N_e + 1, \ldots, 0 \tag{4.42}$$

Note that optimization 4.2 is a QP. This QP is a local approximation of the nonlinear programming problem defined in optimization 4.1. The solution of this QP returns a new iterate. However, because the QP is only a local approximation of the nonlinear programming problem it is not ensured that the new iterate has a lower cost function value than the previous iterate. Therefore, in order to ensure the new iterate has a lower value for the cost function a line-search is implemented. Let p_j denote the iterate at iteration j and p^* the solution obtained from the QP defined in optimization 4.2 at iteration j. The line-search procedure determines the new iterate p_{j+1} by searching on the line defined by $p_{j+1} = p_j + \alpha(p^* - p_j)$, with $0 \leq \alpha \leq 1$. It is possible to find the optimal value for α but because this is too time consuming an approximate value of α is determined. In this thesis a simple line-search is implemented by evaluating the cost function of the iterate $p_{j+1} = p_j + \alpha(p^* - p_j)$ for several values of α going from $\alpha = 0$ to $\alpha = 1$ in steps of $1/25$.

Remark 4.2. *The most straightforward way to implement the nonlinear moving horizon estimator is to use a NLP solver. However, because of the big similarities between MPC and MHE we opted in this thesis to implement the complete SQP method ourselves by making some minor modifications to the MPC code that we developed in Chapter* 3*.*

4.5 Experimental results

In this section experiments are discussed in which a nonlinear MPC is combined with a nonlinear MHE in order to control the large area model of the Demer basin. The simulations are based on the historical rainfall-runoff data of 1998. It is assumed that all water levels are measured at each sampling time. The obtained measurements are assumed to be contaminated with measurement noise. The measurement noise is here assumed Gaussian with zero mean and a standard deviation of 2.5 cm.

4.5.1 Simulations without uncertainty on the rainfall-runoff predictions

In this simulation it is assumed that the rainfall-runoff predictions are free of uncertainties and therefore known exactly. The horizon of MPC is chosen as 30 hours. In order to reduce the calculation time of the estimation process, the shortest possible horizon for the moving horizon estimator was searched for that still lead to accurate estimation results. This was done by some off-line experiments and lead to an estimation window with a size of 8 hours. In contrast with the weighting matrices in the cost function of the model predictive controller, the weighting matrices P, R and Q in the cost function of the moving horizon estimator are chosen to be constant during the whole simulation and were also determined by some off-line experiments. The values of these matrices are summarized in table 4.1.

Table 4.1: The experimentally determined values of the weights in the cost function of the moving horizon estimator. The matrix I denotes the unitary matrix.

	MHE
P	$1000 \cdot I$
Q	$1000000 \cdot I$
R	$10 \cdot I$

In Figures 4.2-4.8 the value of several state variables during the simulation process are depicted. The blue lines coincide with the real values of the states and the red with the values of the states obtained from the moving horizon estimator. It can be seen from these figures that despite the measurement noise, the nonlinearity of the model and an estimation horizon of only 8 hours, the nonlinear moving horizon estimator very accurately estimates the states of the system. Therefore, in this simulation the model predictive controller based its predictions on very accurate state estimates. The results of the control actions provided by the model predictive controller can be seen in Figures 4.9-4.13. Comparing these results with those of the three-position controller depicted in chapter 3, the following conclusions can be made:

- The nonlinear model predictive controller manages to steer h_{opw} much closer to its setpoint. The three-position controller regulates hbg_{opw} slightly closer to its setpoint than the nonlinear model predictive controller. However, note that the three-position controller has an unfair advantage as it changes the gates at a sampling time of 5 minutes in contrast with the nonlinear model predictive controller which has a sampling time of 1 hour.

- Both controllers have a similar performance w.r.t. the flooding of water level h_{vg}. However, it was reported by the VMM that the flooding of water level h_{vg} did not cause any human or financial damage.

- The nonlinear model predictive controller slightly reduced the flooding of the water level h_2.

- With the three-position controller the flooding of water levels h_{gl} and h_w is severely. With the nonlinear model predictive controller only a very slight

Table 4.2: Comparison of the regulation cost of the 2 controllers during the periods before and after the flood event.

	three-position controller	NMPC
$J_{reg}\left(h_{opw}\right)$	16.08	0.2415
$J_{reg}\left(hbg_{opw}\right)$	0.16	0.2170

flooding of h_{gl} occurs for a very short period of time. NMPC manages to avoid flooding of water level h_w completely.

- With the three-position controller water levels h_{zw} and h_{zb} violate their flooding level slightly. The nonlinear model predictive controller manages to avoid flooding of both water levels.

- Both the three-position controller as the nonlinear model predictive controller manage to empty the reservoir h_s before the start of the big rainfall event. However, after the big rainfall event the nonlinear model predictive controller manages to empty the reservoir much faster than the three-position controller. This is important in case a second rainfall event would follow shortly after the first one.

These results are also summarized quantitatively in tables 4.2 and 4.3. The tables show the regulation and flooding cost, respectively. (For the definition of these costs we refer to chapter 3.)

Table 4.3: Comparison of the flooding cost of the 2 controllers.

	three-position controller	NMPC
$J_f(h_2)$	3.00	2.63
$J_f(h_w)$	1.48	0
$J_f(h_{gl})$	9.47	0.01
$J_f(h_{vg})$	13.02	13.18
$J_f(h_{zw})$	9.8×10^{-4}	0.0×10^{-4}
$J_f(h_{zb})$	0.01	0

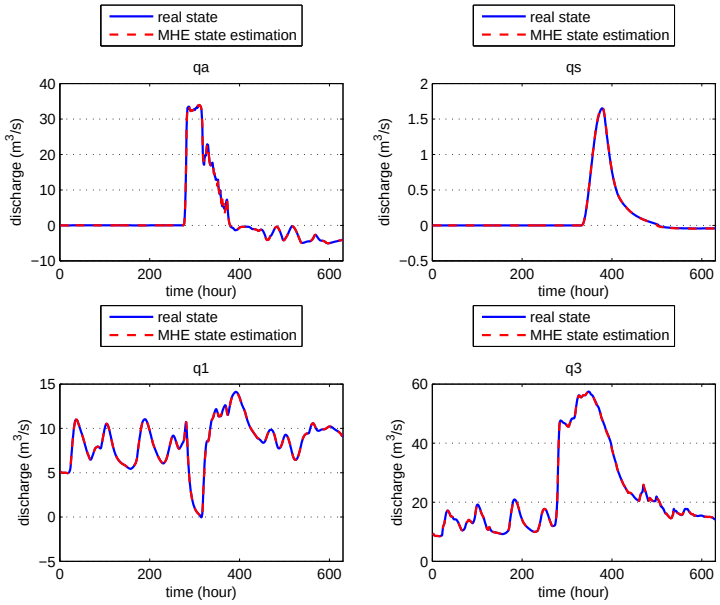


Figure 4.2: Estimation of the discharges without uncertainty on the rainfall-runoff predictions (1)

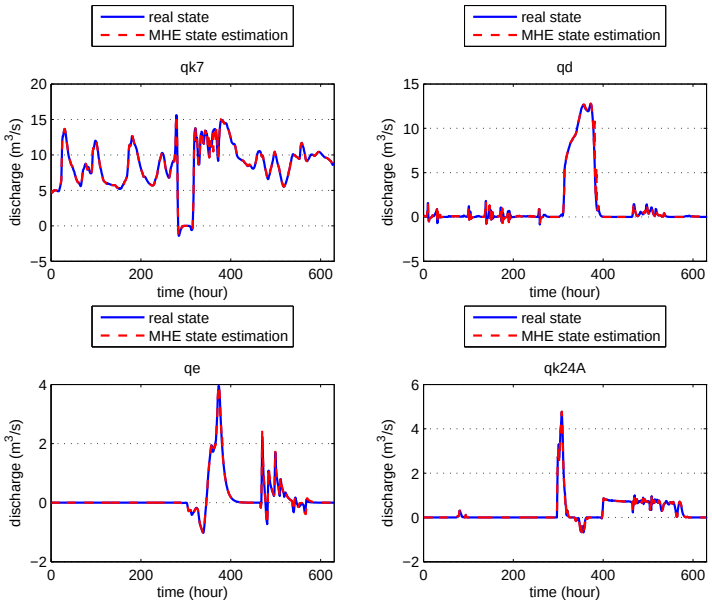


Figure 4.3: Estimation of the discharges without uncertainty on the rainfall-runoff predictions (2)

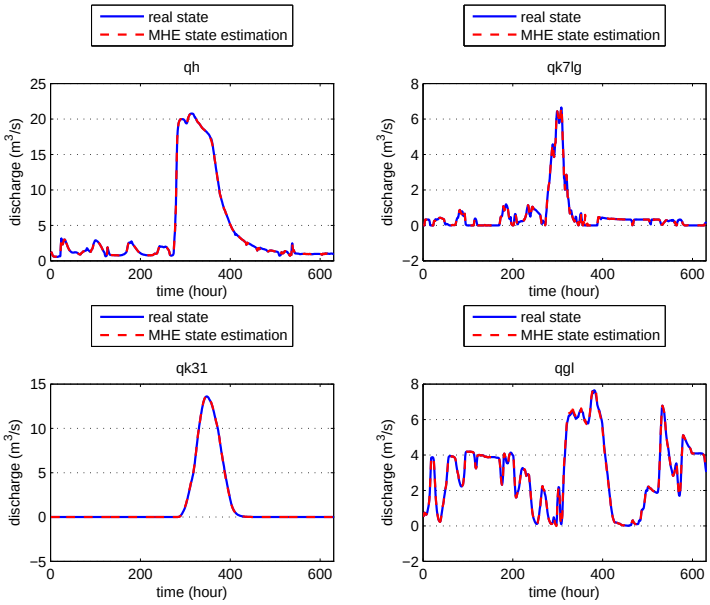


Figure 4.4: Estimation of the discharges without uncertainty on the rainfall-runoff predictions (3)

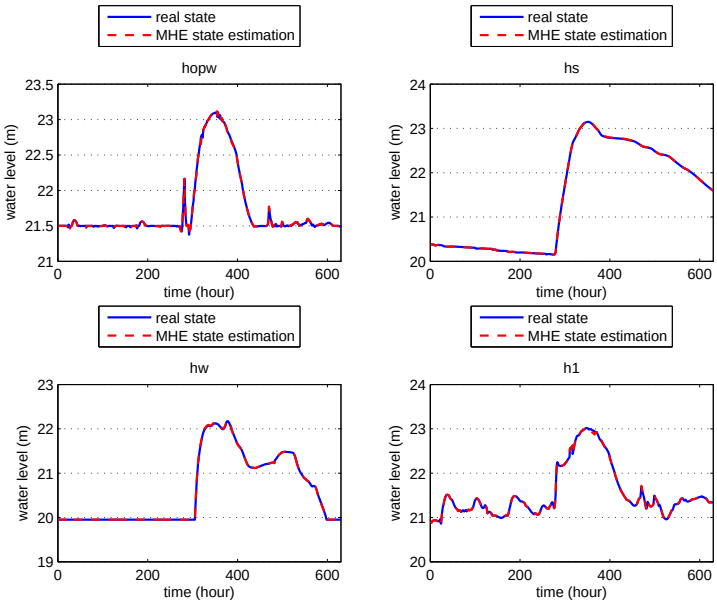


Figure 4.5: Estimation of the water levels without uncertainty on the rainfall-runoff predictions (1)

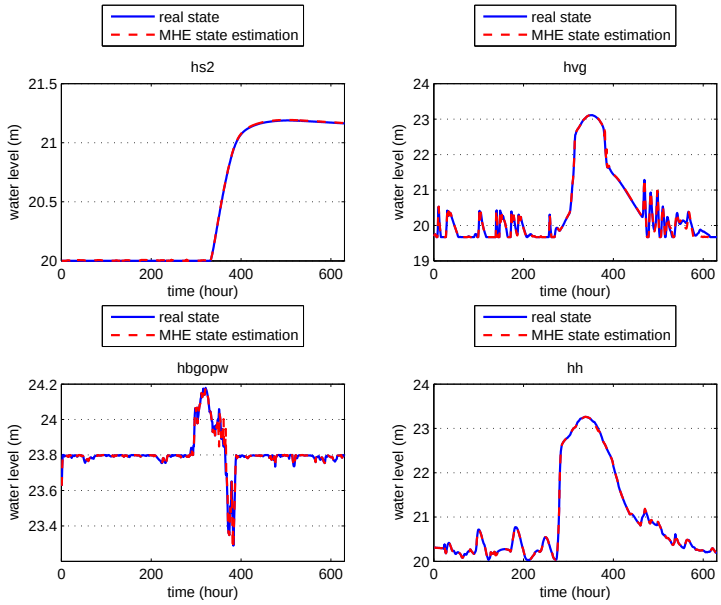


Figure 4.6: Estimation of the water levels without uncertainty on the rainfall-runoff predictions (2)

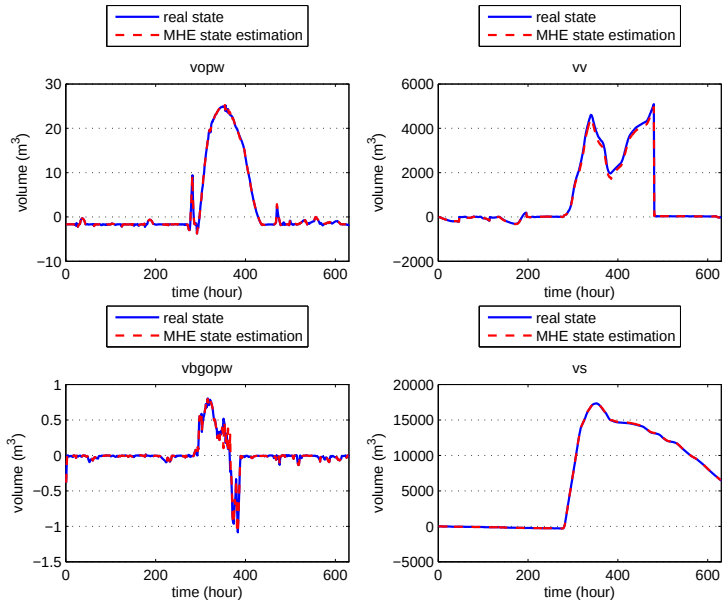


Figure 4.7: Estimation of the volumes without uncertainty on the rainfall-runoff predictions (1)

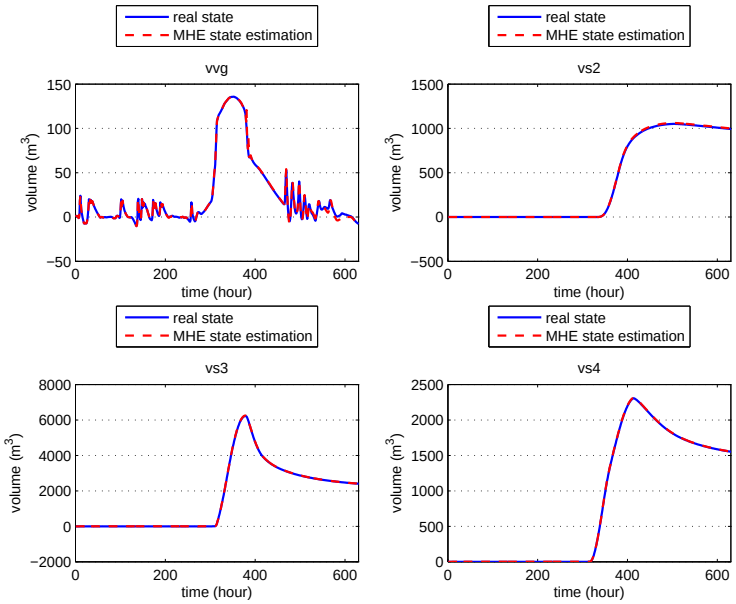


Figure 4.8: Estimation of the volumes without uncertainty on the rainfall-runoff predictions (2)

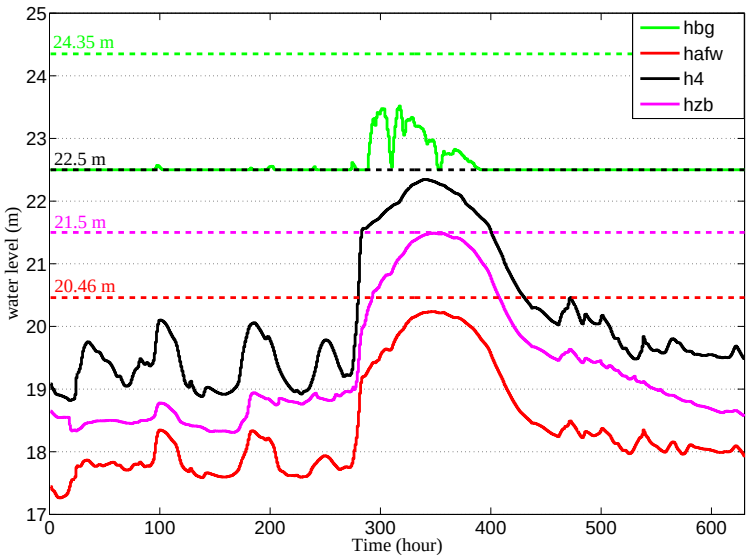


Figure 4.13: Regulation with the nonlinear model predictive controller and the nonlinear moving horizon state estimator without uncertainty on the rainfall-runoff predictions (5)

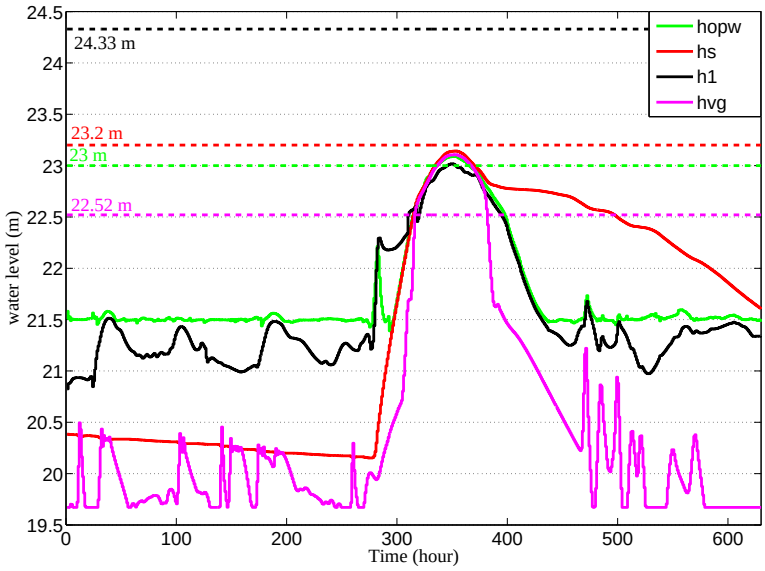


Figure 4.9: Regulation with the nonlinear model predictive controller and the nonlinear moving horizon state estimator without uncertainty on the rainfall-runoff predictions (1)

4.5.2 Simulations with Gaussian uncertainty on the rainfall-runoff predictions

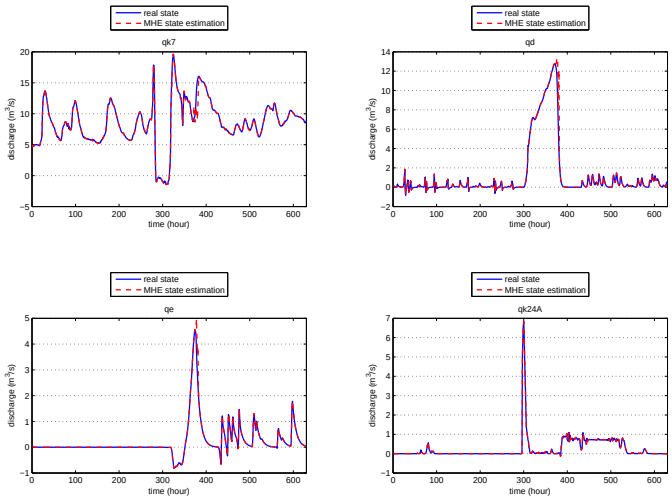


Figure 4.14: Estimation of the discharges with uncertainty on the rainfall-runoff predictions

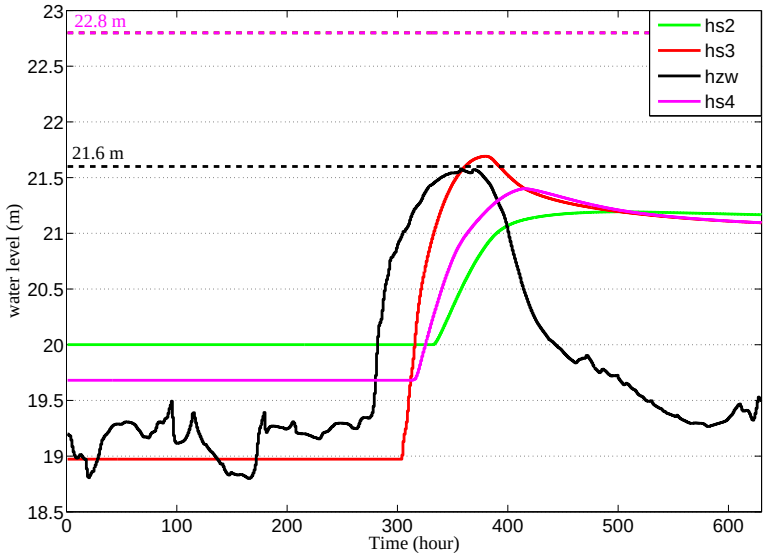


Figure 4.10: Regulation with the nonlinear model predictive controller and the nonlinear moving horizon state estimator without uncertainty on the rainfall-runoff predictions (2)

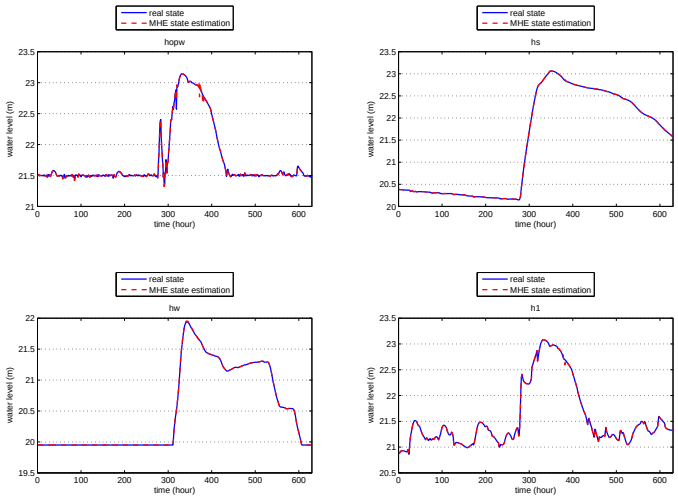


Figure 4.15: Estimation of the water levels with uncertainty on the rainfall-runoff predictions

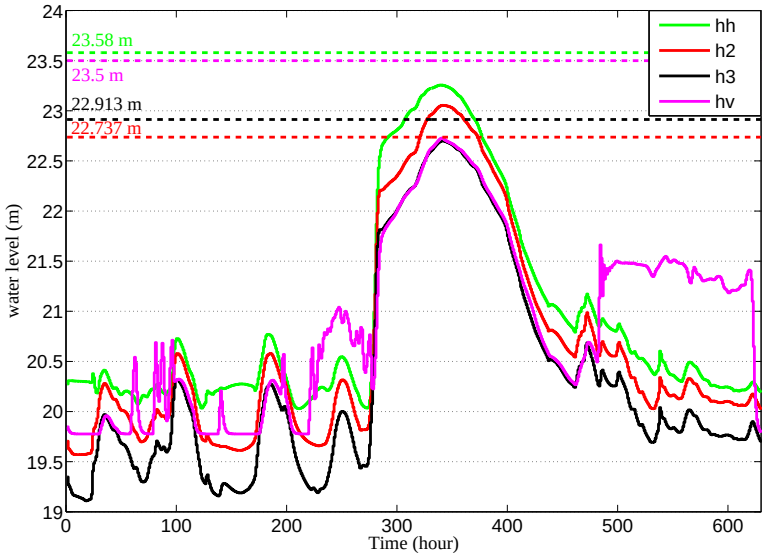


Figure 4.11: Regulation with the nonlinear model predictive controller and the nonlinear moving horizon state estimator without uncertainty on the rainfall-runoff predictions (3)

The simulations in this subsection are similar to the ones from the previous subsection with the only difference that the rainfall-runoff predictions are assumed to be uncertain. The uncertainty added to the rainfall-runoff prediction is the same as in the uncertain simulations performed in chapter 3. The parameters of the model predictive controller and the moving horizon estimator are exactly the same as those used in the previous subsection. In Figures 4.14 and 4.15 the states estimated by the moving horizon estimator are compared with the actual states of the system. It can be seen from both figures that the states estimates are still quite accurate. The estimates of the other states are not depicted but a similar level of accuracy is obtained.

In Figures 4.16-4.20 the simulation results of the NMPC scheme combined with the nonlinear moving horizon state estimator are depicted. These results are also

Table 4.4: Comparison of the regulation cost of the three-position controller with the NMPC-NMHE control strategy with uncertainty on the rainfall-runoff predictions

	three-position controller	NMPC-NMHE
$J_{reg}\,(h_{opw})$	16.08	0.3380
$J_{reg}\,(hbg_{opw})$	0.16	0.2160

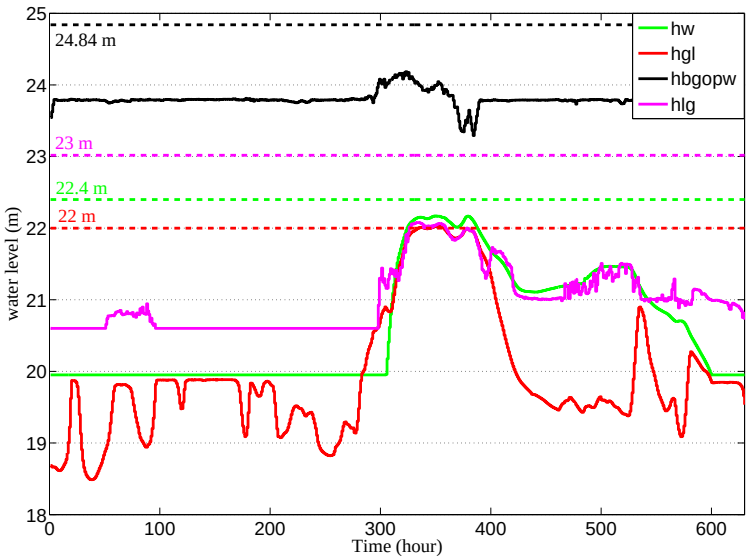


Figure 4.12: Regulation with the nonlinear model predictive controller and the nonlinear moving horizon state estimator without uncertainty on the rainfall-runoff predictions (4)

Table 4.5: Comparison of the flooding cost of the three-position controller with the NMPC-NMHE control strategy with uncertainty on the rainfall-runoff predictions.

	three-position controller	NMPC-NMHE
$J_f\,(h_2)$	3.00	3.92
$J_f\,(h_w)$	1.48	0
$J_f\,(h_{gl})$	9.47	0
$J_f\,(h_{vg})$	13.02	9.23
$J_f\,(h_{zw})$	9.8×10^{-4}	0.0×10^{-4}
$J_f\,(h_{zb})$	0.01	0

summarized in a quantitative way in tables 4.4 and 4.5. The following conclusions can be made from these results:

- Despite the uncertainties, water levels h_{opw} and hbg_{opw} are steered very close to their setpoints. The three-position controller manages to steer water level hbg_{opw} closer to the setpoint than the nonlinear model predictive controller but as already discussed the three-position controller works at a smaller sampling rate than the nonlinear model predictive controller.

- With the three-position controller there occurs severe flooding at the water levels h_w and h_{gl}. With the nonlinear model predictive controller these water levels do not flood. This is an important improvement.

- With the nonlinear model predictive controller water level h_{vg} floods significantly less than with the three-position controller.

- Water levels h_{zw} and h_{zb} slightly flood with the three-position controller. However, with the nonlinear model predictive controller these water levels do not flood.

- The simulations with NMPC show that water level h_2 floods a little bit more than with the three-position controller.

- Before the start of the big rainfall event both controllers manage to empty the reservoir h_s at an equal pace and towards a similar level. However, at the end of the big rainfall event the nonlinear model predictive controller empties the reservoir much faster than the three-position controller.

With exception of water level h_2 all water levels stay lower with the nonlinear model predictive controller than with the three-position controller. Moreover, the increase in water level h_2 is not very big. Therefore, one can conclude that despite the uncertainties introduced by the uncertain rainfall-runoff predictions and the estimation error, the combination NMPC-NMHE outperforms the three-position controller.

Remark 4.3. *The QP's solved at each iteration of the MHE scheme were solved by the Matlab toolbox MOSEK [5]. The time needed to solve* 1 *QP in each iteration of the moving horizon estimator was around* 8 *seconds. At maximum* 5 *QP iterations were allowed at each sampling time, which means that the maximum computation time for the moving horizon estimator was* 40 *seconds.*

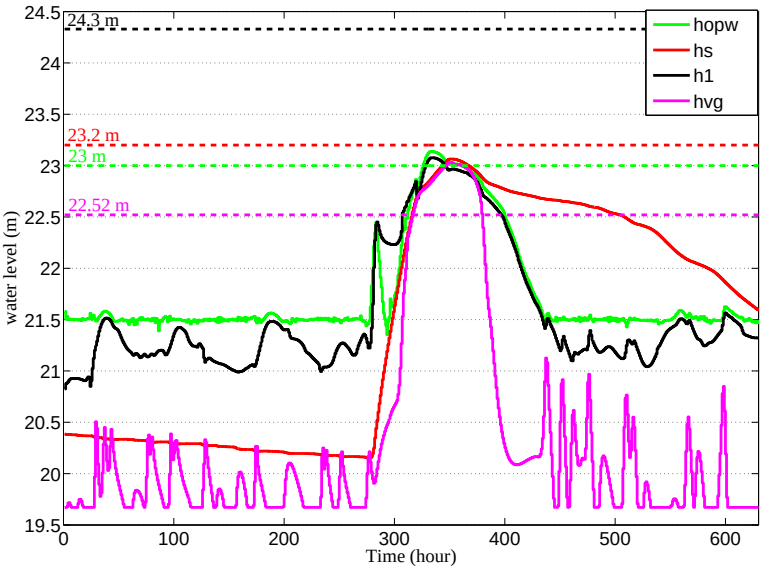


Figure 4.16: Regulation with the nonlinear model predictive controller and the nonlinear moving horizon state estimator with uncertainty on the rainfall-runoff predictions (1)

4.6 Conclusions

In this chapter an overview was first given of different state estimators that are used in practice for the state estimation of discrete-time models with Gaussian noise. Then, from all these estimators a moving horizon estimator was selected in order to estimate the states of the river system during the simulations. A MHE scheme in order to estimate the states of nonlinear systems was fully discussed. The proposed nonlinear MHE scheme was solved by a SQP method combined with a line-search procedure.

This NMHE scheme was combined with the NMPC scheme from chapter 3 in order to control the large area nonlinear model of the Demer basin. This was done by 2 simulations based on the historical rainfall-runoff of 1998. In the first simulation the rainfall-runoff predictions were assumed to perfectly known, while in the second simulation uncertainty was added to these predictions. It was assumed in both simulations that all water levels were measured and Gaussian measurement noise was added to the measured water levels. From the state estimates obtained from both simulations it could be seen that the nonlinear moving horizon estimator gave very accurate estimates of the different states in the river system. These estimates were used by the nonlinear model predictive controller as initial states for the prediction on basis of which the controller derived an optimal control strategy. The

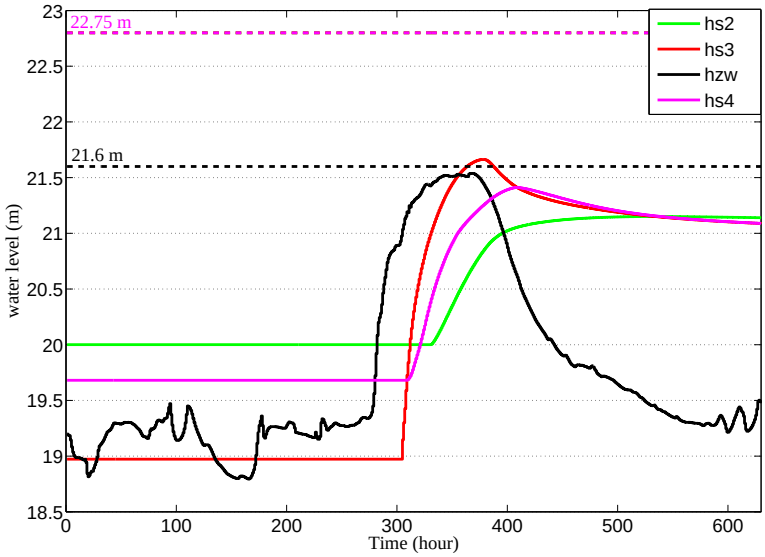


Figure 4.17: Regulation with the nonlinear model predictive controller and the nonlinear moving horizon state estimator with uncertainty on the rainfall-runoff predictions (2)

results of these simulations were compared to those obtained with the three-position controller in chapter 3. From this comparison it could be concluded that the combined NMHE-NMPC scheme still outperformed the three-position controller, despite the uncertainties added to the rainfall-runoff predictions and the state estimation error.

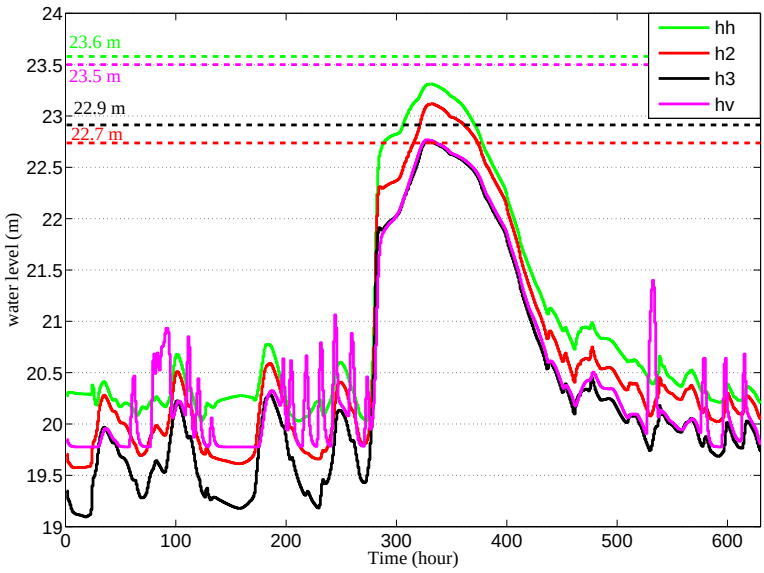


Figure 4.18: Regulation with the nonlinear model predictive controller and the nonlinear moving horizon state estimator with uncertainty on the rainfall-runoff predictions (3)

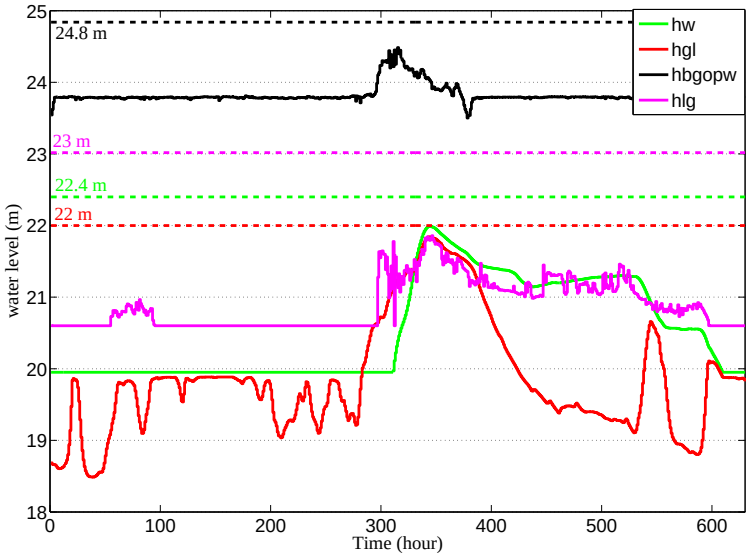


Figure 4.19: Regulation with the nonlinear model predictive controller and the nonlinear moving horizon state estimator with uncertainty on the rainfall-runoff predictions (4)

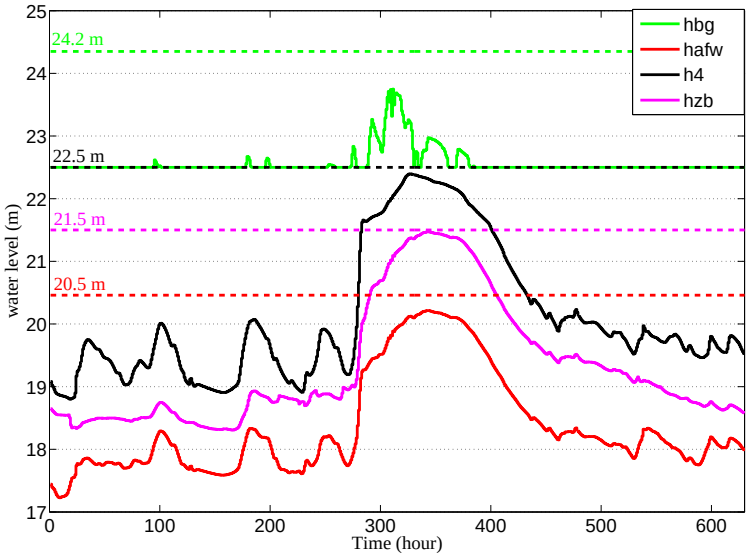


Figure 4.20: Regulation with the nonlinear model predictive controller and the nonlinear moving horizon state estimator with uncertainty on the rainfall-runoff predictions (5)

Chapter 5

Low-Complexity Controlled Invariant Polytopes

This chapter discusses invariants sets in control. First in Section 5.1 an overview is given of some basic definitions related to set invariance as well as an overview of the different types of invariant sets used in control theory. Then, in Section 5.2 the state of the art is given w. r. t. low-complexity polytopic invariant sets. In Section 5.3 a new algorithm is described for the efficient calculation of low-complexity polytopic invariant sets for uncertain linear systems and in Section 5.4 this method is extended to other problem classes. In Section 5.5 the validity of the proposed methods in this chapter are demonstrated by applying them on some practical examples. The chapter ends with conclusions in Section 5.7.

5.1 Set invariance

Set invariance is a concept that allows to tackle many problems in control theory. In this section the concept of set invariance will be outlined in more detail as well as its use for ensuring stability of the closed-loop system. The different families of invariant sets used in practice will be discussed together with the *pros* and *cons* of each class.

In this chapter we consider linear time-varying systems (LTV systems) defined as

$$x_{k+1} = \tilde{A}x_k + \tilde{B}u_k \tag{5.1}$$

with $\widetilde{A}$ and $\widetilde{B}$ uncertain, possibly time-varying belonging to the polyhedral uncertainty class

$$\Omega = \left\{ (\widetilde{A}, \widetilde{B}) = \sum_{i=1}^{n_p} \gamma_i (A_i, B_i) \,\middle|\, \gamma_i \geqslant 0, \sum_{i=1}^{n_p} \gamma_i = 1 \right\}. \tag{5.2}$$

For $n_p = 1$, system (5.2) represents a linear time-invariant system (LTI). Also note that LTV systems can be used to describe the behavior of certain nonlinear models. In this chapter it is assumed that the system is subject to the following linear input and state constraints:

$$x \in \mathbb{X} = \{x | \|Lx\|_\infty \leq 1\} \tag{5.3}$$
$$u \in \mathbb{U} = \{u | \|u\|_\infty \leq \bar{u}\}. \tag{5.4}$$

5.1.1 Definitions

Definition 5.1 (Stability, [36]). *The system 5.1 is stable if all its trajectories converge to zero as $t \to \infty$.*

Note that it can be shown that the LTV system (5.1) is stable if the linear systems $\{[A_i, B_i], i = 1, \ldots, n_p\}$ are stable. A special form of stability which is of importance in control theory is *quadratic stability*:

Definition 5.2 (Quadratic Stability, [36]). *The system (5.1) is called quadratically stable if there exists a quadratic function $V(\xi) = \xi^T P \xi, P \succ 0$ that decreases along every trajectory of the system.*

Definition 5.3 (Quadratic Controllability, [110]). *The LTV system defined by (5.1) is quadratically controllable if, for each $\alpha \geq 0$, it is quadratically stabilizable with the convergent rate α. If the feedback controller is linear, then system (5.1) is quadratically controllable via linear control.*

In order to guarantee stability for MPC the concept of set invariance plays an important role. *Robust positive invariance* is defined as:

Definition 5.4 (Robustly positively invariant set, [35]**).** *A set $\mathcal{S} \in \mathbb{R}^{n_x}$ is robustly positively invariant with respect to the system* (5.1) *for the linear feedback K if*

$$\left(\tilde{A} + \tilde{B}K\right) x \in \mathcal{S}, \qquad \forall x \in \mathcal{S}, \forall (\tilde{A}, \tilde{B}) \in \Omega. \tag{5.5}$$

A related concept is that of λ-contractive positively invariant sets and can be defined as follows:

Definition 5.5 (λ-contractive robustly positively invariant set, [34]**).** *A set $\mathcal{S} \in \mathbb{R}^{n_x}$ is a λ-contractive (with $\lambda \in (0, 1]$) robustly positively invariant set with respect to system* (5.1) *for the linear feedback K if*

$$\left(\tilde{A} + \tilde{B}K\right) x \in \lambda\mathcal{S}, \qquad \forall x \in \mathcal{S}, \forall (\tilde{A}, \tilde{B}) \in \Omega, \tag{5.6}$$

with the scalar multiplication of sets defined as $\lambda\mathcal{S} \triangleq \{\lambda x | x \in \mathcal{S}\}$ as in [128].

Another concept of importance is the concept of the *feasible set* which is defined as

Definition 5.6 (Feasible set). *In case of linear feedback K a set $\mathcal{S}$ is feasible w.r.t. the state constraints (5.3) and input constraints (5.4) if*

$$x \in \mathbb{X}, Kx \in \mathbb{U}, \qquad \forall x \in \mathcal{S}. \tag{5.7}$$

A related definition is that of the *admissible set*:

Definition 5.7 (**Admissable set).** *A set $\mathcal{S}$ is admissible with respect to* (5.1),(5.3),(5.4) *if $x(0) \in \mathcal{S}$ implies that all future states $x(k)$ and all future inputs $Kx(k)$ with $k \in \mathbb{N}$ satisfy constraint* (5.3)*:*

$$x(0) \in \mathcal{S} \quad \Rightarrow \quad x(i) \in \mathbb{X}, Kx(i) \in \mathbb{U} \qquad \forall (A(k), B(k)) \in \Omega, \quad k = 1, \ldots, i-1, \; i \in \mathbb{N}_0. \quad (5.8)$$

Note that an admissible set is not necessarily positively invariant. Admissibility is related to the size of a set (because $\mathbb{X}$ contains the origin in its interior, any sufficiently small neighborhood around the origin is an admissible set) whereas positive invariance is related to the shape of the set. Conversely, one can see that all feasible positively invariant sets are admissible.

Definition 5.8 (**Maximal admissible set** [59])**.** *The maximal admissible (MAS) set S for system* (5.1) *and constraints* (5.3),(5.4) *is defined as the set of all initial states for which the corresponding trajectories under the autonomous dynamics guarantee constraint satisfaction.*

5.1.2 Special families of invariant sets

In this subsection we present two important families of invariant sets which have been particularly successful in the solution of control problems. These are the class of ellipsoidal sets and the class of polyhedral sets. Both classes are important in control problems because they can both be expressed as convex constraints. Therefore, the *on-line* MPC problem to be solved stays a convex programming problem.

5.1.2.1 Ellipsoidal Sets

An ellipsoidal set can be represented as follows:

$$\xi = \left\{ x \in \mathbb{R}^n | x^T P x \leqslant 1 \right\} \quad (5.9)$$

with P a symmetric positive-definite matrix. Invariant ellipsoidal sets are used as terminal constraints in MPC for several reasons:

(i) Ellipsoidal terminal constraints can be represented as LMI's. Therefore, the programming problem to be solved *on-line* is a convex SDP which can be solved efficiently. Moreover, it is assured that the global optimal solution is obtained.

(ii) Ellipsoidal sets scale well with state dimension. They are completely determined by the square matrix P which in state dimension n is a $n \times n$-matrix.

(iii) Many efficient algorithms to calculate ellipsoidal invariant sets are available in literature [36].

Consider a linear feedback gain K and let us fix this gain. Invariance conditions for the ellipsoidal set ξ for system (5.1) under the given linear feedback gain K are given by following LMI ([36]):

$$(A_i + B_i K)^T P + P (A_i + B_i K) \prec 0, \forall i \in \{1, \ldots, n_p\}. \tag{5.10}$$

or equivalently, there exists $Q \succ 0$ such that

$$Q (A_i + B_i K)^T + (A_i + B_i K) Q \prec 0, \forall i \in \{1, \ldots, n_p\}. \tag{5.11}$$

Neither of these conditions is jointly convex in K and P or Q, but by a simple change of variables we can obtain an equivalent condition that is an LMI. Define $Y = KQ$, so that for $Q \succ 0$ we have $K = YQ^{-1}$. Substituting into (5.11) yields

$$A_i Q + Q A_i^T + B_i Y + Y^T B_i^T \prec 0, \forall i \in \{1, \ldots, n_p\}, \tag{5.12}$$

which is an LMI in Q and Y. Note that conditions (5.10)-(5.12) are conditions for quadratic stabilizability of system (5.1) and therefore if P or Q satisfy these conditions then following ellipsoids are invariant:

$$\varepsilon = \left\{ x | \, x^T P x \leq \alpha \right\} \tag{5.13}$$

or

$$\varepsilon = \left\{ x | \, x^T Q^{-1} x \leq \alpha \right\} \tag{5.14}$$

for each $\alpha \geq 0$.

5.1.2.2 Polyhedral Invariant Sets

Polyhedral invariant sets have been involved in the solution of control problems starting from the 70's. Their importance is due to the fact that they are often natural expressions of physical constraints on state and control variables. Furthermore, their shape is in some sense "more flexible" than that of the ellipsoids, leading to a better approximation of reachability sets and domain of attractions of dynamic systems. Moreover, when used as stabilizing constraints in a MPC framework they can be recast as a set of linear constraints which gives rise to a more efficient optimal control problem than when ellipsoidal stabilizing constraints are used. The drawback of polyhedral sets is that in general they have a more complex representation. A polyhedral set can be represented in the following form

$$\varphi = \{ x | \, Fx \leq \bar{1} \} \tag{5.15}$$

where F is an $r \times n$ matrix and $\bar{1} \in \mathbb{R}^r$ denotes a vector of the form

$$\bar{1} = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}^T, \tag{5.16}$$

and the inequality has to be thought componentwise. Polyhedral sets are very popular in control theory because of following properties:

(i) In [59] it was shown that for system (5.1) the MAS is a polyhedral set and a technique was provided to determine this set. In [96] and [97] this work was extended for uncertain systems and efficient algorithms were developed to determine the MAS.

(ii) Polyhedral sets can be expressed as a set of linear constraints and therefore the on-line optimization problem is typically a quadratic program which can be solved efficiently.

In order for a given polyhedral set to be invariant it needs to satisfy some invariance conditions. The following invariance conditions play a central role in this chapter.

Lemma 5.1. *The polyhedral set (5.15) defines an invariant set for the system (5.1) w.r.t. the state feedback K if and only if there exists a positive $\lambda \leq 1 (\lambda < 1)$, and an $r \times r$ nonnegative matrix H_i such that $\forall i \in \{1, \ldots, n_p\}$*

$$\begin{aligned} &H_i \bar{1} \leq \lambda \bar{1}, \\ &H_i F = F \left(A_i + B_i K \right). \end{aligned} \tag{5.17}$$

Note that these invariance conditions are based on following property (derived from Farkas' Lemma [66]).

Lemma 5.2. *Given two polyhedra $\varphi_1 = \{ x | \, F_1 x \leq g_1 \}$ and $\varphi_2 = \{ x | \, F_2 x \leq g_2 \}$ then $\varphi_1 \subseteq \varphi_2$ if there exists a non-negative matrix H such that $HF_1 = F_2$ and $Hg_1 \leq g_2$.*

5.1.2.3 Low-Complexity Polytopes

Polytopic sets are preferred over ellipsoidal sets because they can be expressed as linear constraints which lead to the *on-line* solution of a quadratic program (QP). Ellipsoidal sets are expressed as LMIs which lead to the *on-line* solution of a SDP which needs more calculation time to be solved than a QP. Moreover, polytopic sets allow us to exactly describe the MAS, the biggest possible invariant set and therefore gives rise to a bigger domain of attraction. An important drawback, however, is that for larger dimensional systems the number of constraints needed to describe the MAS typically increases exponentially, which makes the construction of these sets computationally intractable [97]. There exist two different approaches to tackle this combinatorial explosion of number of constraints needed to describe the MAS:

(i) The first approach starts from a full complexity invariant set (i.e. the MAS) which is then modified in order to reduce the number of constraints. This approach was adopted in [97].

(ii) The second approach consists of imposing a fixed low-complexity structure on the invariant set leading to a parallellotopic invariant inner approximation of the MAS. This approach was adopted in [46].

The latter approach is used in this chapter. In this chapter the low-complexity polytopes are assumed to have the following form [47]:

$$\varphi = \{x | \left\| Vx \right\|_\infty \leq \gamma\}, \gamma \geq 0 \tag{5.18}$$

with V a square matrix (not necessarily symmetric) and of full-rank. The polytope φ is defined by 2^n vertices via

$$v_j = \gamma V^{-1} s_j, j = 1, \ldots, 2^n \tag{5.19}$$

where $\{s_j\}, j \in \{1, \ldots, 2^n\}$ is the set of all possible n-dimensional vectors whose elements are ± 1. By ordering the indices so that $\{s_1, \ldots, s_n\}$ are linearly independent, the complete set of 2^n vertices can be parameterized in terms of the first n alone. For convenience such a set of n vertices will be referred to as primary vertices; all others can be expressed as linear combinations of $\{v_1, \ldots, v_n\}$ [47]:

$$v_j = \begin{bmatrix} v_1 & \ldots & v_n \end{bmatrix} \begin{bmatrix} s_1 & \ldots & s_n \end{bmatrix}^{-1} s_j, \quad j = n+1, \ldots, 2^{n-1}, \tag{5.20}$$

$$v_j = -v_{j-2^{n-1}}, \quad j = 2^{n-1}+1, \ldots, 2^n. \tag{5.21}$$

5.2 State of the art

Here we give a brief overview of the state of the art concerning the computation of invariant polytopes. We start with the computation of the MAS for linear systems. Then the focus will shift towards methods to calculate low-complexity polytopes for both LTI and LTV systems.

5.2.1 Low-complexity polytopes for the LTI case

Some of the early works concerning invariant low-complexity polytopes for LTI systems can be found in [30] and [32]. In these works the considered low-complexity polytopes were assumed to have the form as described in (5.18). In a more recent work [71] it was pointed out that these sets can be conservative whenever the input and state constraints are asymmetric due to their symmetrical shape. Therefore, in [71] the focus was shifted towards non-symmetric polytopes. The proposed non-symmetric polytopes have the following form:

$$\Pi(V, \alpha_1, \alpha_2) = \{x : -\alpha_1 \leq Vx \leq \alpha_2\}, \quad \alpha_1, \alpha_2 \geq 0 \quad rank(V) = n \tag{5.22}$$

$$\Pi(G, \alpha) = \{x : Vx \leq \alpha\}, \quad \alpha \geq 0 \quad rank(G) = n \tag{5.23}$$

which become equivalent for

$$G = \begin{bmatrix} V \\ -V \end{bmatrix}, \quad \alpha = \begin{bmatrix} \alpha_2 \\ \alpha_1 \end{bmatrix}. \tag{5.24}$$

Note that these types of polytopic set coincide with the symmetric polytopes described by (5.18) for $\alpha_1 = \alpha_2$. Therefore, the results obtained in [71] can be used to determine symmetric low-complexity polytopes for LTI systems. The strategy proposed in [71] is twofold. First, a method is described that allows to calculate an initial feasible invariant

polytope. In a next step an optimization procedure is outlined that allows to increase the volume of the initial set. The initialization is based on following lemma [32]

Lemma 5.3. *$\Pi(V, \alpha_1, \alpha_2)$ is positively invariant under the LTI $x_{k+1} = \phi x_k$ if and only if there exists a non-negative matrix H such that*

$$V\phi - HV = 0 \tag{5.25}$$

$$\begin{bmatrix} H^+ & -H^- \\ -H^- & H^+ \end{bmatrix} \begin{bmatrix} \alpha_2 \\ \alpha_1 \end{bmatrix} \leq \begin{bmatrix} \alpha_2 \\ \alpha_1 \end{bmatrix} \tag{5.26}$$

Based on this Lemma the following theorem was derived that offered the possibility to calculate an initial invariant set

Theorem 5.1. *Let K be such that ϕ has simple eigenvalues and let (λ_i, w_i) and $(\rho_i \pm j\varphi_i, a_i \pm jb_i)$ denote the real and complex eigenvalue/vector pairs. Then for $V = W^{-1}$, where W is the matrix comprising as columns the vectors w_i and a_i, b_i (each of the a_i, b_i taken only once), there exist α_1, α_2 such that $\Pi(V, \alpha_1, \alpha_2)$ is positively invariant under $x_{k+1} = Ax_k + Bu_k$ for $u_k = -Kx_k$ if and only if ϕ is Hurwitz with eigenvalues lying inside the complex plane defined by the square with vertices $\left\{1, e^{j\pi}, e^{\pi/2}, e^{j3\pi/2}\right\}$.*

So in order to determine an initial invariant set the eigenvalues $\lambda_1, \ldots, \lambda_n$ for the closed-loop system $\Phi = A + BK$ are chosen such that they lie in the square with vertices $\left\{1, e^{j\pi}, e^{j\pi/2}, e^{j3\pi/2}\right\}$. This can be achieved by means of poles placement. It then follows that the set $\varphi = \left\{x \middle| \left\|W^{-1}x\right\|_\infty \leq 1\right\}$ is invariant for the closed-loop system Φ. Note that the closed-loop system Φ can be written as

$$\Phi = WDW^{-1}, \quad D = \begin{bmatrix} diag\{\lambda_i\} & 0 \\ 0 & diag\{D_i\} \end{bmatrix}, \quad D_i = \begin{bmatrix} \rho_i & \sigma_i \\ -\sigma_i & \rho_i \end{bmatrix}. \tag{5.27}$$

Besides invariance, the set needs to be feasible w.r.t. the state (5.3) and input constraints (5.4). The following conditions are necessary conditions for invariance and feasibility of the set $\Pi(V, \alpha_1, \alpha_2)$ w.r.t. the constraints (5.3) and (5.4) under system $x_{k+1} = \Phi x_k$ for $u = -Kx$

$$\begin{bmatrix} D^+ & -D^- \\ -D^- & D^+ \end{bmatrix} \begin{bmatrix} \alpha_2 \\ \alpha_1 \end{bmatrix} \leq \begin{bmatrix} \alpha_2 \\ \alpha_1 \end{bmatrix} \tag{5.28}$$

$$\begin{bmatrix} \tilde{K}^+ & -\tilde{K}^- \\ -\tilde{K}^- & \tilde{K}^+ \end{bmatrix} \begin{bmatrix} \alpha_2 \\ \alpha_1 \end{bmatrix} \leq \begin{bmatrix} -\bar{u} \\ \bar{u} \end{bmatrix} \tag{5.29}$$

$$\begin{bmatrix} \tilde{L}^+ & -\tilde{L}^- \\ -\tilde{L}^- & \tilde{L}^+ \end{bmatrix} \begin{bmatrix} \alpha_2 \\ \alpha_1 \end{bmatrix} \leq \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} \tag{5.30}$$

with $\tilde{K} = KW, \tilde{L} = LW$ and $D^+ = \max\{D, 0\}, D^- = \min\{D, 0\}, \tilde{K}^+ = \max\{K, 0\}, \tilde{K}^- = \min\{K, 0\}$. Constraints (5.28)-(5.30) are linear in the parameters

α_1, α_2 which can therefore be deployed as degrees of freedom in order to maximize the volume of the initial invariant and feasible set. Now assume that

$$\begin{array}{c} \alpha = \begin{bmatrix} \alpha_2 \\ \alpha_1 \end{bmatrix}, \quad P = \begin{bmatrix} P_f \\ P_i \end{bmatrix}, \quad p = \begin{bmatrix} p_f \\ 0 \end{bmatrix}, \quad p_f = \begin{bmatrix} -\bar{u} \\ \bar{u} \end{bmatrix} \\ P_f = \begin{bmatrix} \tilde{K}^+ & -\tilde{K}^- \\ -\tilde{K}^- & \tilde{K}^+ \end{bmatrix}, \quad P_i = \begin{bmatrix} \tilde{D}^+ - I & -\tilde{D}^- \\ -\tilde{D}^- & \tilde{D}^+ - I \end{bmatrix} \end{array} \tag{5.31}$$

with the subscript indices f and i indicating conditions relating to feasibility and invariance, then the following optimization problem maximizes the volume of the initial invariant feasible set:

Optimization problem 5.1. *[Maximizing the volume of* $\Pi(V, \alpha_1, \alpha_2)$ *by solving a convex problem]* $\alpha = \arg\max_z \log(prod(z))$ *subject to*

$$Pz \leq p \tag{5.32}$$

$$z \geq 0 \tag{5.33}$$

where $prod(z)$ *denotes the product of the elements of* z*.*

Remark 5.1. *The facets of the maximal volume feasible set obtained from applying Optimization problem 5.1 are parallel to the facets of the initial invariant set. In order to see this let* $V_1, V_2, \ldots, V_n$ *denote the rows of the matrix* V *and* $\alpha_i(j)$ *the* j*-th element of* α_i*, then for an initial invariant set defined by* $\varphi_0 = \left\{x | \alpha_1^0 \leq Vx \leq \alpha_2^0\right\}$ *the constraints defining the set (5.18) can be written as*

$$\begin{array}{llll} V_1 x \leq \alpha_2^0(1), & \ldots, & V_n x \leq \alpha_2^0(n), \\ V_1 x \geq \alpha_1^0(1), & \ldots, & V_n x \geq \alpha_1^0(n) \end{array}$$

The optimization of Optimization problem 5.1 only changes the value of $\alpha_i^0(j)$ *and therefore the facets of the optimized set are parallel to the facets of the initial set.*

5.2.2 Low-complexity polytopes for the LTV case

Most of the work concerning low-complexity polytopes described by (5.18) deals with LTI systems. For LTV systems the literature is less elaborated. Some interesting results can be found in [73] where invariance of low-complexity polytope (5.18) is checked by the Perron-Frobenius norm of an appropriate matrix. However, the proposed conditions are nonlinear. A more recent and also more interesting work can be found in [47]. In [47] a method is outlined that allows to calculate maximal volume feasible invariant low-complexity polytopes for nonlinear systems. The methods in [47] are based on replacing the nonlinear model by a LTV system. Therefore, these results also apply for LTV systems. Because the results in [47] form the basis of the new results discussed in this chapter, in the sequel these results will be outlined in more detail.

Invariance and feasibility of the set (5.18) under the feedback law $u_k = Kx_k$ for the LTV system defined in (5.2) require

$$\|Vx_{k+1}\|_\infty - \|Vx_k\|_\infty \leq 0 \tag{5.34}$$
$$\|Kx_k\|_\infty \leq \bar{u} \tag{5.35}$$

In order to obtain better performance the invariance condition (5.34) can be strengthened to

$$\|Vx_{k+1}\|_\infty - \|Vx_k\|_\infty \leq -\|Cx_k\|_\infty . \tag{5.36}$$

Taking these invariance and feasibility conditions into account it was shown in [47] that the following nonlinear program defines the maximal feasible invariant low-complexity polytope:

Theorem 5.2.

$$\min_{v_j, w_j, j=1,\ldots,2^{n-1}} -\log\det\left(\begin{bmatrix} v_1 & \ldots & v_n \end{bmatrix}\right) \tag{5.37}$$

subject to the following inequality constraints invoked for $i = 1,\ldots,n_p, j = 1,\ldots,2^{n-1}$*:*

$$\|V(A_i v_j + B_i w_j)\|_\infty \leq \gamma - \|Cv_j\|_\infty \tag{5.38}$$
$$\|w_j\|_\infty \leq \bar{u} \tag{5.39}$$
$$\|Lv_j\|_\infty \leq 1 \tag{5.40}$$

where, for $j = n+1,\ldots,2^{n-1}$*,* v_j *is defined by (5.20)-(5.21) and* $w_j = \begin{bmatrix} w_1 & \ldots & w_n \end{bmatrix}\begin{bmatrix} s_1 & \ldots & s_n \end{bmatrix}^{-1} s_j$*. The associated linear feedback gain* K *can be recovered from the solution for* $\{v_j, w_j, j = 1,\ldots,n\}$ *via*

$$K = \begin{bmatrix} w_1 & \ldots & w_n \end{bmatrix}\begin{bmatrix} v_1 & \ldots & v_n \end{bmatrix}^{-1}. \tag{5.41}$$

Note that the number of optimization variables as well as the number of constraints increase exponentially with the state dimension. This is an important drawback which impedes the method to be used in high dimensions. Also note that the invariance condition (5.38) is nonconvex since it involves the product of V with terms that depend linearly on the variables $\{v_j\}$ and $\{w_j\}$. In [47] this constraint is formulated as a bilinear constraint by means of following lemma:

Lemma 5.4. *Condition (5.38) holds iff there exist* q_{ij} *with*

$$\begin{aligned} A_i v_j + B_i w_j &= \begin{bmatrix} v_1 & \ldots & v_{2^{n-1}} \end{bmatrix} q_{ij}, \\ \mathbf{1}^T |q_{ij}| &\leq 1 - \gamma^{-1}\|Cv_j\|_\infty \end{aligned} \tag{5.42}$$

for $i = 1,\ldots,n_p, j = 1,\ldots,2^{n-1}$ *(where* $\mathbf{1} = \begin{bmatrix} 1 & \ldots & 1 \end{bmatrix}^T$*).*

However, replacing condition (5.38) with (5.42) keeps the programming problem of Theorem 5.2 nonconvex. Therefore, in [47] the optimization is broken into a sequence of simpler problems. Each of these problems is concerned with optimizing only over a single primary vertex v_k, the remainder of the primary vertices being fixed at the values computed at the previous optimization. Thus let c_k denote the vector that is orthogonal to all except the kth primary vertex v_k:

$$c_k^T v_j = 0, \forall j \neq k, \tag{5.43}$$

then the optimization of Theorem 5.2 can be performed by solving a sequence of linear programs (LPs) as indicated below

Theorem 5.3. *The nonlinear program of Theorem 5.2 can be computed by solving the following LP successively for the individual vertices $v_k, k = 1, \ldots, n$:*

$$\min_{\substack{v_k, w_j, j=1,\ldots,n \\ Q_i, N_i, i=1,\ldots,n_p}} c_k^T v_k \tag{5.44}$$

subject to the following linear constraints invoked for $i = 1, \ldots, n_p, j = 1, \ldots, 2^{n-1}$:

$$\begin{aligned} A_i \begin{bmatrix} v_1 & \ldots & v_n \end{bmatrix} + B_i \begin{bmatrix} w_1 & \ldots & w_n \end{bmatrix} \\ = \begin{bmatrix} v_1 & \ldots & v_n \end{bmatrix} N_i \end{aligned} \tag{5.45}$$

$$\begin{aligned} N_i \begin{bmatrix} s_1 & \ldots & s_n \end{bmatrix}^{-1} \begin{bmatrix} s_1 & \ldots & s_{2^{n-1}} \end{bmatrix} \\ = \begin{bmatrix} s_1 & \ldots & s_n \end{bmatrix}^{-1} \begin{bmatrix} s_1 & \ldots & s_{2^{n-1}} \end{bmatrix} Q_i \end{aligned} \tag{5.46}$$

$$\boldsymbol{1}^T |Q_i e_j| \leq 1 - \gamma^{-1} \|C v_j\|_\infty \tag{5.47}$$

$$\|L v_j\|_\infty \leq 1 \tag{5.48}$$

$$\|w_j\|_\infty \leq \bar{u} \tag{5.49}$$

(e_j denotes the jth column of the identity matrix), where the kth row of N_i for $i = 1, \ldots, n_p$ and the primary vertices v_j for $j \neq k$ are constants defined by

$$e_k^T = e_k^T N_i^0, \tag{5.50}$$

$$v_j = v_j^0 \tag{5.51}$$

and where $\left\{v_j^0, j = 1, \ldots, n\right\}$ and $N_i^0, i = 1, \ldots, n_p$ are defined by the previously computed LP solution.

Though this sequence of LPs can be used in order to obtain a solution of the nonlinear program Theorem 5.2 the proposed approach suffers from several drawbacks:

- Similar to the nonlinear program outlined in Theorem 5.2 the number of constraints and optimization variables still scale exponentially with the state dimension and therefore the method cannot be used in practice for high state dimensions.

- In order to be able to start the sequence of LPs an initial invariant feasible set is needed. In the literature no methods exist to determine such an initial set. The only way to find such an initial set is to start from a random set and perform a local search until a set is found satisfying all the constraints in Theorem 5.2.

- In the examples of Section 5.5 it will be shown that the proposed sequence suffers from premature convergence leading to very conservative solutions.

5.3 Efficient computation of low-complexity polytopes

In this section a new method is discussed that does not suffer from the drawbacks of the methods discussed in Section 5.2. The method consists of two steps. In the first step an initial invariant feasible set is determined by means of a convex programming problem. In the second step the volume of this set is increased by a sequence of convex programs each increasing the volume of the set obtained from the previous iteration. This sequence is initiated by the initial invariant set obtained from the first step.

5.3.1 Invariance conditions

Lemma 5.2 states the Farkas' Lemma for general polytopes. In the special case of low-complexity polytopes this Lemma can be reduced as indicated in the following Lemma:

Lemma 5.5. *Given two polyhedra* $\varphi_1 = \{x| \left\|V_1 x\right\|_\infty \leq 1\}$ *and* $\varphi_2 = \{x| \left\|V_2 x\right\|_\infty \leq 1\}$ *then* $\varphi_1 \subseteq \varphi_2$ *if and only if there exist non-negative matrices* H_1 *and* H_2 *such that the following conditions are satisfied:*

$$(H_1 - H_2)V_1 = V_2 \tag{5.52}$$

$$\begin{bmatrix} H_1 & H_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{1} \\ \boldsymbol{1} \end{bmatrix} \leqslant \boldsymbol{1} \tag{5.53}$$

with $\boldsymbol{1}$ *a* n*-dimensional vector whose elements are all* 1.

Proof. The two sets can be rewritten as

$$\varphi_1 = \left\{ x \middle| \begin{bmatrix} V_1 \\ -V_1 \end{bmatrix} x \leqslant \begin{bmatrix} \boldsymbol{1} \\ \boldsymbol{1} \end{bmatrix} \right\} \tag{5.54}$$

$$\varphi_2 = \left\{ x \middle| \begin{bmatrix} V_2 \\ -V_2 \end{bmatrix} x \leqslant \begin{bmatrix} \boldsymbol{1} \\ \boldsymbol{1} \end{bmatrix} \right\}. \tag{5.55}$$

By Farkas' Lemma $\varphi_1 \subseteq \varphi_2$ if and only if the following conditions are satisfied:

$$\begin{bmatrix} H_1 & H_2 \\ H_3 & H_4 \end{bmatrix} \begin{bmatrix} V_1 \\ -V_1 \end{bmatrix} = \begin{bmatrix} V_2 \\ -V_2 \end{bmatrix} \tag{5.56}$$

$$\begin{bmatrix} H_1 & H_2 \\ H_3 & H_4 \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} \leqslant \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} \tag{5.57}$$

with $H_1, H_2, H_3, H_4 \geq 0$. Here the conditions on H_1, H_2 are identical to those on H_3, H_4 and (5.56) and (5.57) are therefore equivalent to (5.52) and (5.53).□

Combining this lemma with the invariance conditions for general polytopes defined in Lemma 5.1 leads to following invariance conditions for the low-complexity polytopes described by (5.18):

Lemma 5.6. *The set $\varphi = \{x| \left\|Vx\right\|_\infty \leq 1\}$ is invariant for the system $x_{k+1} = A_j x_k + B_j u_k$ under linear state feedback $u_k = Kx_k$ if and only if there exist non-negative matrices H_{1j} and H_{2j} such that the following conditions are satisfied for $j = 1, \ldots, n_p$:*

$$(H_{1j} - H_{2j})V = V(A_j - B_j K) \tag{5.58}$$

$$\begin{bmatrix} H_{1j} & H_{2j} \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} \leqslant \mathbf{1} \tag{5.59}$$

$$H_{1j}, H_{2j} \geq 0 \tag{5.60}$$

Proof.
Invariance of $\varphi = \{x| \left\|Vx\right\|_\infty \leq 1\}$ is equivalent to

$$\varphi \subseteq \left\{x|(\widetilde{A} + \widetilde{B}K)x \in \varphi, \forall(\widetilde{A}, \widetilde{B}) \in \Omega\right\}, \tag{5.61}$$

which is equivalent to $\varphi \subseteq \left\{x| \left\|V(A_j + B_j K)x\right\|_\infty \leq 1\right\}$ for $j = 1, \ldots, n_p$. By Lemma 5.5 this condition is satisfied if and only if (5.58)-(5.60) hold. □

5.3.2 Initialization

In order to determine an initial invariant set for the system (5.1) the following relation between 2- and ∞-norms will be used:

$$\forall x \in \mathbb{R}^n : \left\|Vx\right\|_\infty \leq \left\|Vx\right\|_2 \leq \sqrt{n} \left\|Vx\right\|_\infty . \tag{5.62}$$

The following theorem uses this property in order to determine an initial invariant set for the system (5.1).

Theorem 5.4. *The set $\varphi = \{\left\|Vx\right\|_\infty \leq \gamma\}$ is invariant for system (5.1) w.r.t. the state feedback K if the following condition is satisfied $\forall(\widetilde{A}, \widetilde{B}) \in \Omega$:*

$$\left\|V\widetilde{\phi}x_k\right\|_2 \leq (\frac{1}{\sqrt{n}}) \left\|Vx_k\right\|_2 \tag{5.63}$$

with $\widetilde{\phi} = (\widetilde{A} + \widetilde{B}K)$.

Proof. From (5.62) the following relationships follow

$$\left\|V\widetilde{\phi}x\right\|_\infty \leq \left\|V\widetilde{\phi}x\right\|_2 \leq \frac{1}{\sqrt{n}}\left\|Vx\right\|_2 \leq \left\|Vx\right\|_\infty \tag{5.64}$$

which proves the theorem. □

Remark 5.2. *In order to impose condition (5.63) for all $(\widetilde{A},\widetilde{B}) \in \Omega$ it is sufficient to impose the condition on the vertices of the polytopic uncertainty set Ω:*

$$\left\|V(A_i + B_iK)x\right\|_2 \leq (\frac{1}{\sqrt{n}})\left\|Vx\right\|_2, i = 1,\ldots,n_p. \tag{5.65}$$

In the case of $(\widetilde{A},\widetilde{B})$ belonging to the polytopic uncertainty class (5.2), the condition (5.63) can be re-written as the following set of LMIs

$$\phi_i^T P \phi_i \preceq \alpha^2 P,\ \ i = 1,\ldots,n_p \tag{5.66}$$

with $\alpha = 1/\sqrt{n}$, $P \succeq 0$ and $\phi_i = A_i + B_iK$. A matrix P satisfying these LMIs leads to a set $\varphi = \left\{x|\ \|P^{\frac{1}{2}}x\|_\infty \leq 1\right\}$ which is invariant for the uncertain system under the stabilizing feedback law K. A suitable P and K satisfying conditions (5.66) and constraints (5.3)-(5.4) can be found by solving the following optimization scheme ([36], [70], [72]):

Algorithm 5.1.

$$\min_{Q,X,Y} -\log\,\det(Q) \tag{5.67}$$

subject to

$$Q \succeq 0 \tag{5.68}$$

$$L(i,:)QL(i,:)^T \leq 1,\ i = 1,\ldots,n_c \tag{5.69}$$

$$\left[\begin{array}{cc} X & Y \\ Y^T & Q \end{array}\right] \succeq 0, \quad X(j,j) \leqslant \bar{u}^2\ ,\ j = 1,\ldots,n_u \tag{5.70}$$

and

$$\left[\begin{array}{cc} \alpha^2 Q & (A_iQ + B_iY)^T \\ A_iQ + B_iY & Q \end{array}\right] \succeq 0,\ j = 1,\ldots,n_p \tag{5.71}$$

By taking $P = \frac{1}{\alpha^2}Q^{-1}$ and $K = YQ^{-1}$, a feedback gain is obtained stabilizing the uncertain system (5.1) in such a way that the corresponding ∞-norm set φ is invariant and all the input and state constraints are satisfied.

Remark 5.3. *In case the system is linear time-invariant it can easily been shown that a necessary and sufficient condition for the optimization algorithm of Theorem 5.4 to be feasible is controllability of the LTI system. If the system is linear time-varying than a sufficient condition is that the LTV system is quadratically controllable.*

5.3.3 Volume maximization

Theorem 5.5. *The maximum volume feasible invariant low-complexity polytope is the solution of the nonlinear program, for $j = 1, \ldots, n_p$:*

$$\min_{V,K,H_{1j},H_{2j},H_3,\ldots,H_6} \log(\det(V)) \tag{5.72}$$

subject to the following constraints:

$$\begin{aligned}
(H_{1j} - H_{2j})V &= V(A_j - B_jK) && (5.73)\\
\begin{bmatrix} H_{1j} & H_{2j} \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} &\leqslant \mathbf{1} && (5.74)\\
(H_3 - H_4)V &= K && (5.75)\\
\begin{bmatrix} H_3 & H_4 \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} &\leqslant \bar{u}\mathbf{1} && (5.76)\\
(H_5 - H_6)V &= L && (5.77)\\
\begin{bmatrix} H_5 & H_6 \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} &\leqslant \mathbf{1} && (5.78)\\
H_{1j}, H_{2j}, H_3, H_4, H_5, H_6 &\geqslant 0 && (5.79)
\end{aligned}$$

Proof. Consider the following sets:

$$\begin{aligned}
\varphi_1 &= \{x \,|\, \|Vx\|_\infty \leqslant 1\} && (5.80)\\
\varphi_2 &= \{x \,|\, \|Kx\|_\infty \leqslant \bar{u}\} && (5.81)\\
\varphi_3 &= \{x \,|\, \|Lx\|_\infty \leqslant 1\} && (5.82)\\
\varphi_{4j} &= \left\{x \,|\, \|V(A_j - B_jK)x\|_\infty \leqslant 1\right\} && (5.83)
\end{aligned}$$

Conditions (5.73) and (5.74) are the invariance conditions as defined in Lemma 5.6. In order for the feedback K to satisfy the input constraints (5.4) the set φ_1 must satisfy $\varphi_1 \subseteq \varphi_2$. Applying Lemma 5.5 this results to conditions (5.75) and (5.76). In order for the set φ_1 to satisfy the state constraints (5.3) the set φ_1 must satisfy $\varphi_1 \subseteq \varphi_3$. Again, by applying Lemma 5.5 this leads to the conditions (5.77) and (5.78).□

Constraints (5.75) and (5.77) of Theorem 5.5 are bilinear constraints. By introducing the transformations $P = V^{-1}$ and $Q = KP$ these constraints can be made linear as given in the following theorem:

Theorem 5.6. *The optimization of (5.72) subject to (5.73)-(5.79) is equivalent to:*

$$\min_{P,Q,H_{1j},H_{2j},H_{3,\ldots,6},j=1,\ldots,n_p} -\log(\det(P)) \tag{5.84}$$

subject to the following equations for $j = 1, \ldots, n_p$:

$$P(H_{1j} - H_{2j}) = A_jP + B_jQ \tag{5.85}$$

$$H_3 - H_4 = Q \tag{5.86}$$

$$H_5 - H_6 = LP \tag{5.87}$$

$$H_{1j}, H_{2j}, H_{3,\dots,6} \geqslant 0 \tag{5.88}$$

$$\begin{bmatrix} H_{1j} & H_{2j} \\ H_3 & H_4 \\ H_5 & H_6 \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} \leqslant \begin{bmatrix} \mathbf{1} \\ \bar{u}\mathbf{1} \\ \mathbf{1} \end{bmatrix} \tag{5.89}$$

with

$$P = V^{-1} \tag{5.90}$$

and

$$Q = KP. \tag{5.91}$$

The corresponding stabilizing feedback K can be obtained as

$$K = QP^{-1} \tag{5.92}$$

Proof. The volume of the set (5.18) is optimized by maximizing the objective $\left|det(\begin{bmatrix} v_1 & \dots & v_n \end{bmatrix})\right|$, with $v_{1,\dots,n}$ the primary vertices of the set (5.18). This objective can be written as $\left|det(V^{-1}S)\right| = |det(P)|\,|det(S)| \sim |det(P)|$ (where S is a constant non-singular matrix with elements equal to ± 1), which leads to the objective (5.84). The invariance constraint (5.73) can easily be re-written as $(H_{1j} - H_{2j})P^{-1} = P^{-1}(A_j + B_jK)$ which can be re-written as (5.85) taking into account that $Q = KP$. A similar approach applied to (5.75) and (5.77) leads to conditions (5.86) and (5.87).□

Remark 5.4. *Note that the state constraints (5.87) and input constraints (5.86) in Theorem 5.6 are linear. In Theorem 5.5 these constraints are bilinear. This clearly shows the benefits of Theorem 5.6. However, the invariance conditions (5.85) are still bilinear and are therefore the reason that the optimization procedure in Theorem 5.6 is nonlinear.*

In comparison with the method described in Theorem 5.2 this method leads to a significant reduction of the number of variables and constraints. In Theorem 5.2 for a state space with dimension n the increase in variables and constraints is of order $n2^n$ while in this method the increase is of order n^2 which means that the optimization scheme described in Theorem 5.6 is scalable in high dimensions. Also note that this optimization scheme is nonlinear due to the cost function (5.84) and the n bilinear invariance constraints (5.85). This optimization scheme can be simplified by means of the concept of inverse-positive matrices [31].

Definition 5.9. *A matrix is inverse-positive if all the entries of its inverse are positive.*

A matrix A satisfying $A = \alpha I - D$ with $\alpha \geq 0$, D a nonnegative matrix and $\|D\|_\infty \leq \alpha$ is inverse-positive. The following theorem uses this concept in order to generate a sequence of convex programs leading to a sequence of feasible invariant sets with increasing volume.

Theorem 5.7. *The following iteration for $k = 0, \dots$:*

$$\min_{X_k,Q,R_1,R_2,H_3,\ldots,H_6,a,D} \left(-\log\det\left(P_k X_k\right)\right) \tag{5.93}$$

subject to the following convex constraints for $j = 1,\ldots,n_p$*:*

$$\begin{aligned}
P_k(R_{1j} - R_{2j}) &= A_j P_k X_k + B_j Q && (5.94)\\
H_3 - H_4 &= Q && (5.95)\\
H_5 - H_6 &= L P_k X_k && (5.96)\\
\begin{bmatrix} R_{1j} & R_{2j} \\ H_3 & H_4 \\ H_5 & H_6 \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} &\leqslant \begin{bmatrix} X_k\mathbf{1} \\ \bar{u}\mathbf{1} \\ \mathbf{1} \end{bmatrix} && (5.97)\\
R_{1j}, R_{2j} &\geq 0 && (5.98)\\
X_k &= aI - D && (5.99)\\
a &\geq 0 && (5.100)\\
\|D\|_\infty &\leq a && (5.101)\\
X_k &\succeq 0 && (5.102)\\
X_k &= X_k^T && (5.103)
\end{aligned}$$

with $P_0 = V_0^{-1}, X_0 = I, P_{k+1} = P_k X_k$ *and the unity matrix* I*, leads to a sequence of feasible invariant sets* $P_0, P_1, \ldots$ *with volume* $(P_{k+1}) \geq$ *volume* (P_k)*.*

Proof. Condition (5.94) follows from condition (5.85) by assuming $P_{k+1} = P_k X_k$ and $R_{ij} = X_k H_{ij}$. The first constraint of condition (5.97) coincides with the first constraint of condition (5.89). In order to see this note that the first constraint of (5.97) can be re-written as

$$R_{1j}\mathbf{1} + R_{2j}\mathbf{1} \leq X_k\mathbf{1}. \tag{5.104}$$

This can be re-written as

$$X_k^{-1}\left(R_{1j}\mathbf{1} + R_{2j}\mathbf{1}\right) \leq \mathbf{1} \tag{5.105}$$

due to condition (5.98) and the inverse-positiveness conditions (5.99)-(5.101). Because $R_{ij} = X_k H_{ij}$ if follows that this expression is the same as condition (5.89). Conditions (5.102) and (5.103) are necessary to ensure the cost function (5.93) is convex.

Also note that at iteration $k+1$ the set P_{k+1} obtained at iteration k still satisfies all the constraints imposed at iteration $k+1$ for $X_{k+1} = I$. This means that the set P_{k+1} obtained at iteration k is a feasible solution for the optimization at time step $k+1$. Therefore, this iterative sequence generates a sequence of invariant sets feasible w. r. t. the state and input constraints. This, in combination with the fact that P_{k+1} coincides with the set $\left\{x \middle| \left\|P_{k+1}^{-1}x\right\|_\infty \leq 1\right\}$ and the fact that at iteration $k+1$ the cost function (5.93) maximizes the volume of the set defined by P_{k+2}, ensure that the generated sequence of sets satisfies that the volume $(P_{k+2}) \geq$ the volume

(P_{k+1}). Therefore, the generated sequence consists of invariant and feasible sets with monotonically increasing volume. It follows that the sequence of feasible invariant sets generated as solutions of the convex program of this theorem is guaranteed to converge to a (possibly local) solution of the nonlinear program defined in Theorem 5.5. □

Remark 5.5. *For $D = \mathbf{0}$, the matrix X turns into a diagonal matrix. Note that in this case the facets of the resulting set are parallel to the facets of the initial invariant set. It can also easily been shown that the sequence converges after 1 iteration. Observe that in the case of a LTI system and X a diagonal matrix, the optimization problem of Theorem 5.7 coincides with the optimization of Optimization problem 5.1 with the only difference that in Theorem 5.7 the feedback gain K is also part of the optimization problem. Therefore Optimization problem 5.1 can be seen as a special case of Theorem 5.7.*

Remark 5.6. *By means of Theorem 5.1 an initial invariant and feasible set $\varphi = \{x| \left\|V_0 x\right\|_\infty \leq 1\}$ can be determined. The sequence defined by Theorem 5.7 can then be initialized by chosing $P_0 = V_0^{-1}$.*

5.4 Extensions

5.4.1 Linear systems with bounded disturbances

In this subsection the focus lies on extending the results from Section 5.3 to linear systems with bounded disturbances. Such systems are defined as follows:

$$x_{k+1} = Ax_k + Bu_k + d_k \tag{5.106}$$

with disturbance d_k bounded and belonging to the following uncertainty polytope

$$\Delta = \left\{ d = \sum_{i=1}^{n_d} \lambda_i d_i \,\middle|\, \lambda_i \geqslant 0, \sum_{i=1}^{n_d} \lambda_i = 1 \right\} \tag{5.107}$$

with d_i the vertices of the polytope Δ. Similar as the procedure for determining low-complexity polytopes for LTV system in Section 5.3, two steps are necessary for determining an initial invariant feasible low-complexity polytope for the system (5.106). The first step consists in finding an initial invariant feasible set. In order for a low-complexity polytope described by (5.18) to be invariant for the system (5.106) under the bounded disturbance described by Δ and linear feedback gain K the following invariance condition must be satisfied:

$$\left\|V\left((A+BK)v_i + d\right)\right\|_\infty \leq \left\|Vv_i\right\|_\infty, \forall i \in \{1, \ldots, 2^n\} \tag{5.108}$$

with v_i the vertices of the low-complexity polytope. Taking the triangle inequality into account following stricter (and therefore more conservative) invariance condition is obtained

$$\left\|V(A+BK)v_i\right\|_\infty + \left\|Vd_j\right\|_\infty \leq \left\|Vv_i\right\|_\infty, \forall i \in \{1, \ldots, 2^n\}, \forall j \in \{1, \ldots, n_d\} \tag{5.109}$$

which forms the basis of following lemma.

Lemma 5.7. *The low-complexity polytope defined by (5.18) is invariant for the linear system(5.106) under the bounded disturbance described by Δ and linear feedback gain K if following sufficient conditions are satisfied for $i = 1, \ldots, 2^n, j = 1, \ldots, n_d$:*

$$\|Vd_j\|_\infty \leq \beta \|Vv_i\|_\infty \tag{5.110}$$
$$\|V(A+BK)v_i\|_\infty \leq (1-\beta)\|Vv_i\|_\infty \tag{5.111}$$

Proof Imposing condition (5.110) and combining this with the invariance condition (5.109) leads to condition (5.111).□

Note that condition (5.111) is equivalent to $(1-\beta)$-contractiveness of the low-complexity polytope for the linear system without disturbances. The factor β is determined by the user by trial-and-error. The conditions provided by lemma 5.7 are not convex because V,K and the vertices v_i are unknown. By adding extra conservativeness the following theorem provides convex conditions that allow us to determine an initial invariant low-complexity polytope:

Theorem 5.8. *The set $\varphi = \{x | \, \|Vx\|_\infty \leq 1\}$ is invariant for the linear system (5.106) with disturbance d_j vertices of the disturbance set Δ under the linear feedback gain K if the following conditions are satisfied for $j = 1, \ldots, n_d$:*

$$\|Vd_j\|_2 \leq \beta \tag{5.112}$$
$$\|V(A+BK)x\|_2 \leq \left(\frac{(1-\beta)}{\sqrt{n}}\right)\|Vx\|_2 \tag{5.113}$$

Proof Condition (5.112) follows trivially from condition (5.110), property (5.22) and the fact that $\|Vv_i\|_\infty = 1$ as v_i is a vertex of the low-complexity polytope. From property (5.22) and condition (5.113) the following can be derived:

$$\|V(A+BK)x\|_\infty \leq \|V(A+BK)x\|_2 \leq \left(\frac{(1-\beta)}{\sqrt{n}}\right)\|Vx\|_2 \leq \|Vx\|_\infty \tag{5.114}$$

If condition 5.114 holds for each x then it also holds for $x = v_i, \forall i \in \{1, \ldots, 2^n\}$ and therefore following condition is satisfied

$$\|V(A+BK)v_i\|_\infty \leq \|Vv_i\|_\infty \tag{5.115}$$

which coincides exactly with the invariance condition (5.111) and therefore proves the theorem. □

Similar to algorithm 5.1 following Optimization problem can be derived from Theorem 5.8 in order to determine an invariant low-complexity polytope for the linear system with bounded disturbances that satisfies the input and state constraints defined by (5.4)-(5.3):

Optimization problem 5.2.

$$\min_{Q,X,Y} -\log\,\det(Q) \tag{5.116}$$

subject to

$$\begin{aligned} Q &\succeq 0 && (5.117)\\ L(i,:)QL(i,:)^T &\leq 1,\ i=1,\ldots,n_c && (5.118)\\ \begin{bmatrix} X & Y \\ Y^T & Q \end{bmatrix} &\succeq 0, && (5.119)\\ X(j,j) &\leqslant \bar{u}^2,\ j=1,\ldots,n_u && (5.120)\\ \begin{bmatrix} \beta & d_j^T \\ d_j & Q \end{bmatrix} &\succeq 0 && (5.121) \end{aligned}$$

and with

$$\begin{bmatrix} \alpha^2 Q & (A_iQ+B_iY)^T \\ A_iQ+B_iY & Q \end{bmatrix} \succeq 0,\ j=1,\ldots,n_p \tag{5.122}$$

and

$$\alpha = \left(\frac{(1-\beta)}{\sqrt{n}}\right) \tag{5.123}$$

By taking $P = Q^{-1}$ and $K = YQ^{-1}$, a feedback gain is obtained stabilizing the linear system with bounded disturbances in such a way that the corresponding low-complexity polytope defined by $\varphi = \left\{x \middle| \left\|P^{0.5}x\right\|_\infty \leq 1\right\}$ is invariant and feasible w.r.t. the input and state constraints.

In a second step the volume of this initial invariant set is increased. Following Theorem defines the maximal volume feasible invariant low-complexity polytope for linear systems with bounded disturbances:

Theorem 5.9. *The maximal volume low-complexity polytope defined by* $\varphi = \left\{x \middle| \left\|P^{-1}x\right\|_\infty \leq 1\right\}$ *invariant for the linear system (5.106) with bounded disturbance defined by (5.107) and feasible w.r.t. the input and state constraints (5.4)-(5.3) under the linear feedback gain* K *is the solution of the nonlinear program, for* $j = 1,\ldots,n_d$ *:*

$$\min_{H_{1,\ldots,6},V,P,\beta} -\log\det(P) \tag{5.124}$$

subject to the following constraints for $j = 1,\ldots,n_d$

$$\begin{aligned} P\,(H_1 - H_2) &= AP + BQ && (5.125)\\ H_3 - H_4 &= Q && (5.126)\\ H_5 - H_6 &= LP && (5.127) \end{aligned}$$

$$\begin{aligned}
\|V d_j\|_\infty &\leq \beta && (5.128)\\
H_{1,\ldots,6} &\geq 0 && (5.129)\\
\begin{bmatrix} H_1 & H_2 \end{bmatrix} \mathbf{1} &\leq \mathbf{1}(1-\beta) && (5.130)\\
\begin{bmatrix} H_3 & H_4 \\ H_5 & H_6 \end{bmatrix} \mathbf{1} &\leq \mathbf{1} && (5.131)\\
V &= P^{-1}. && (5.132)
\end{aligned}$$

The linear feedback K can be recovered as $K = QP^{-1}$.

Proof Condition (5.125) combined with condition (5.130) are conditions obtained from the Farkas' Lemma (5.2) and ensure the set φ is $(1-\beta)$-contractive. Therefore, together with condition (5.128) the conditions for disturbance invariance as defined in Lemma 5.7 are satisfied. The other conditions in the Theorem ensure the input and state constraints are satisfied and are similar to those of Theorem 5.6. □

Theorem 5.9 defines a nonlinear programming problem which can raise considerable computational demands. Following Theorem circumvents this by solving a sequence of convex programming problems leading to a sequence of feasible disturbance invariant sets with increasing volume:

Theorem 5.10. *The following iteration for $k = 1, \ldots$:*

$$\min_{X_k, Q, R_1, \ldots, R_4, H_3, \ldots, H_6, a, D} \left(-\log\det\left(P_{k-1}X_k\right)\right) \tag{5.133}$$

subject to the following convex constraints:

$$\begin{aligned}
P_{k-1}(R_1 - R_2) &= A_j P_{k-1} X_k + B_j Q && (5.134)\\
H_3 - H_4 &= Q && (5.135)\\
H_5 - H_6 &= L P_{k-1} X_k && (5.136)\\
R_3 - R_4 &= I && (5.137)\\
\begin{bmatrix} R_1 & R_2 \\ H_3 & H_4 \\ H_5 & H_6 \\ R_3 & R_4 \end{bmatrix} \mathbf{1} &\leqslant \begin{bmatrix} X_k\mathbf{1}(1-\beta_{k-1}) \\ \bar{u}\mathbf{1} \\ \mathbf{1} \\ X_k\mathbf{1} \end{bmatrix} && (5.138)\\
R_1, \ldots, R_4 &\geq 0 && (5.139)\\
X_k &= aI - D && (5.140)\\
a &\geq 0 && (5.141)\\
\|D\|_\infty &\leq a && (5.142)\\
X_k &\succeq 0 && (5.143)\\
X_k &= X_k^T && (5.144)
\end{aligned}$$

with $\beta_{k-1} = \max_j \left\| P_{k-1}^{-1} d_j \right\|_\infty$*, leads to a sequence of feasible disturbance invariant sets* $P_0^{-1}, P_1^{-1}, \ldots$ *satisfying* $P_k = P_{k-1} X_k$ *with volume* $(P_k) \geq$ *volume* (P_{k-1})*.*

Proof

The conditions of this Theorem can be derived from Theorem 5.9 by substituting P with P_k and parameterizing P_k as $P_k = P_{k-1}X_k$ in a similar way as done in Theorem 5.7. Condition (5.137) combined with the last condition of (5.138) ensures that condition (5.128) is satisfied. This can be seen as follows. Assume that we impose that the set $\varphi_{k-1} \subseteq \varphi_k$, then according to the Farkas' Lemma following conditions must be satisfied:

$$(H_7 - H_8)V_{k-1} = V_k \tag{5.145}$$

$$\begin{bmatrix} H_7 & H_8 \end{bmatrix} \mathbf{1} \leq \mathbf{1}. \tag{5.146}$$

From condition (5.145) the following can be derived:

$$P_k(H_7 - H_8) = P_{k-1}. \tag{5.147}$$

Parameterizing P_k as $P_k = P_{k-1}X_k$ and defining R_3 and R_4 as $R_3 = X_k H_7$ and $R_4 = X_k H_8$ condition (5.137) is obtained. Combined with the last condition of 5.128 this ensures that $\varphi_{k-1} \subseteq \varphi_k$. So if $\forall j \in \{1, \ldots, n_d\} : \|V_{k-1} d_j\| \leq \beta_{k-1}$ then if follows automatically that $\forall j \in \{1, \ldots, n_d\} : \|V_k d_j\| \leq \beta_{k-1}$ and therefore ensuring condition (5.128). Also note that if we chose $X_k = I$ then P_{k-1} is also a solution of the nonlinear program of the theorem and therefore this theorem leads to a sequence of disturbance invariant and feasible polytopes. Because the cost function (5.133) maximizes the volume of the low-complexity polytopes defined by $\varphi = \left\{ x \middle| \left\| P_k^{-1} x \leq 1 \right\|_\infty \right\}$ and because the set obtained at iteration $k-1$ is also feasible at iteration k, the generated sequence of disturbance invariant low-complexity polytopes is monotonically increasing in volume. □

Remark 5.7. *Note that the variable* β_{k-1} *decreases at each iteration because the volume of the sequence of sets increases by which* $\max_j \left\| P_{k-1}^{-1} d_j \right\|_\infty$ *decreases. Therefore, at each iteration the sets become less contractive until convergence of the sequence.*

Remark 5.8. *Note that the solution from Optimization problem 5.2 can be used to initialize the first set* P_0^{-1} *of the sequence.*

5.4.2 Low-complexity polytopes based on the L_1-norm

In this subsection the focus lies on LTV systems described by (5.1). The considered low-complexity polytopes are based on the L_1- norm and are described as

$$\varphi = \{ x \mid \|Vx\|_1 \leq 1 \}. \tag{5.148}$$

The set φ is completely determined by n linearly independent primary vertices $\{v_1, \ldots, v_n\}$ as follows:

$$v_i = -v_{i-n}, i = n+1, \ldots, 2n. \tag{5.149}$$

At each vertex the following relation holds

$$Vv_i = e_i, \qquad i = 1, \ldots, n \tag{5.150}$$
$$Vv_i = -e_{i-n}, \quad i = n+1, \ldots, 2n \tag{5.151}$$

with e_i the i-th column of the identity matrix. Therefore, following relation holds between V and the primary vertices

$$V = \left[\begin{array}{ccc} v_1 & \ldots & v_n \end{array} \right]^{-1}. \tag{5.152}$$

Observe that the number of vertices scales linearly with the state dimension in contrast to the low-complexity polytopes defined as in (5.18) where the number of vertices scales exponentially with the state dimension. Therefore, vertex based approaches as the one described in [47] scale linearly with the state dimension and can therefore be used in higher state dimensions. The first step consists in determining an initial invariant and feasible low-complexity described by (5.148). This can be achieved by the following relationship between 1- and 2-norms:

$$\|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\|x\|_2. \tag{5.153}$$

Following Lemma uses this relationship in order to determine an initial invariant low-complexity polytope described by (5.148):

Lemma 5.8. *The set (5.148) is invariant for the system (5.1) w.r.t. the state feedback K if the following condition is satisfied for $\forall(\tilde{A}, \tilde{B}) \in \Omega$:*

$$\left\| V\tilde{\phi}x \right\|_2 \leq \left(\frac{1}{\sqrt{n}} \right) \|Vx\|_2 \tag{5.154}$$

Proof
From (5.154) and (5.153) the following can be derived

$$\left\| V\tilde{\phi}x \right\|_1 \leq \left\| V\tilde{\phi}x \right\|_2 \leq \left(\frac{1}{\sqrt{n}} \right) \|Vx\|_2 \leq \|Vx\|_1 \tag{5.155}$$

which means that the set $\varphi = \{x | \, \|Vx\|_1 \leq 1\}$ is invariant.□

Note that condition (5.154) is exactly the same as the condition (5.63) for low-complexity polytopes based on the L_∞-norm. Therefore, Algorithm 5.1 can also be used for the calculation of an initial invariant feasible set based on the L_1-norm. In the second step a volume maximization is performed in order to increase the volume of the initial set. The following Theorem concerns a vertex-based optimization scheme in order to determine maximal volume feasible invariants sets of the form (5.148).

Theorem 5.11. *The maximum feasible invariant low-complexity polytope is the solution of the nonlinear program:*

$$\min_{v_1,\ldots,v_n,w_1,\ldots,w_n} -\log\left(\left|\det\left(\left[\begin{array}{ccc} v_1 & \ldots & v_n \end{array}\right]\right)\right|\right) \tag{5.156}$$

subject to the following equations for $j = 1, \ldots, n_p$*:*

$$\begin{aligned} A_j \left[\begin{array}{ccc} v_1 & \ldots & v_n \end{array}\right] + B_j \left[\begin{array}{ccc} w_1 & \ldots & w_n \end{array}\right] &= \\ \left[\begin{array}{cccccc} v_1 & \ldots & v_n & -v_1 & \ldots & -v_n \end{array}\right] Q_j & \end{aligned} \tag{5.157}$$

$$\|Lv_j\|_\infty \leq \mathbf{1} \tag{5.158}$$

$$\|w_i\|_\infty \leq \bar{u} \tag{5.159}$$

$$\left[\mathbf{1}^T \mathbf{1}^T\right] Q_j = \mathbf{1}^T \tag{5.160}$$

$$Q_j \geq 0 \tag{5.161}$$

The corresponding stabilizing feedback K can be obtained as

$$K = \left[\begin{array}{ccc} w_1 & \ldots & w_n \end{array}\right] \left[\begin{array}{ccc} v_1 & \ldots & v_n \end{array}\right]^{-1}. \tag{5.162}$$

Proof

Condition (5.157) together with (5.160)-(5.161) ensure the resulting set is invariant. To see this note that column p of constraint (5.157) can be re-written as $Av_p + Bw_p = \sum_{i=1}^{n} Q_j(i,p)v_i + \sum_{i=1}^{n} Q_j(n+i,p)(-v_i)$. Because (5.161) ensures $Q_j(i,p) \geq 0$ and (5.160) ensures $\sum_{i=1}^{2n} Q_j(i,p) = 1$ this means that $A_j v_p + B_j w_p \in \text{Co}\{v_1, \ldots, v_n, -v_1, \ldots, -v_n\}$ and the set is invariant. Conditions (5.158) and (5.159) ensure the state and input constraints are satisfied. Expression (5.162) follows trivially from $\left[\begin{array}{ccc} w_1 & \ldots & w_n \end{array}\right] = K \left[\begin{array}{ccc} v_1 & \ldots & v_n \end{array}\right]$. □

The program of Theorem 5.11 is nonlinear due to the nonlinear cost function (5.156) and the bilinear invariance constraints (5.157). However, by introducing some conservativeness it is possible to recast this problem into a convex programming problem as stated below.

Theorem 5.12. *The nonlinear program of Theorem 5.11 can be computed by solving the following convex program:*

$$\begin{array}{c} \min \\ v_1, \ldots, v_n, w_1, \ldots, w_n \\ Q_1, \ldots, Q_{n_p} \\ a_1, \ldots, a_n \end{array} \quad \sum_{i=1}^{n} -\log a_i \tag{5.163}$$

subject to the following linear constraints for $i = 1, \ldots, n, j = 1, \ldots, n_p$

$$A_j \left[\begin{array}{ccc} v_1 & \ldots & v_n \end{array}\right] + B_j \left[\begin{array}{ccc} w_1 & \ldots & w_n \end{array}\right] = \tag{5.164}$$

$$\begin{bmatrix} v_1^0 & \dots & v_n^0 & -v_1^0 & \dots & -v_n^0 \end{bmatrix} Q_j \tag{5.165}$$

$$\begin{bmatrix} v_1 & \dots & v_n \end{bmatrix} = \tag{5.166}$$

$$\begin{bmatrix} v_1^0 & \dots & v_n^0 \end{bmatrix} \begin{bmatrix} a_1 & \boldsymbol{0}^T & 0 \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ 0 & \boldsymbol{0}^T & a_n \end{bmatrix} \tag{5.167}$$

$$|w_i| \leqslant u \tag{5.168}$$

$$\begin{bmatrix} \boldsymbol{1}^T & \boldsymbol{1}^T \end{bmatrix} Q_j = \boldsymbol{1}^T \tag{5.169}$$

$$Q_j \geq 0 \tag{5.170}$$

$$a_1, \dots, a_n \geq 1 \tag{5.171}$$

with $\boldsymbol{0} = \begin{bmatrix} 0 & \dots & 0 \end{bmatrix}^T \in \mathbb{R}^{n-2}$ *and the robustly invariant feasible set* $\left\{v_1^0, \dots, v_n^0, -v_1^0, \dots, -v_n^0\right\}$ *computed at the previous iteration.*

Proof.

Similar to the proof of Theorem 5.11 conditions (5.165), (5.169)-(5.170) ensure for each $p = 1, \dots, 2n$:

$$A_j v_p + B_j w_p \in Co\left\{v_1^0, \dots, v_n^0, -v_1^0, \dots, -v_n^0\right\}. \tag{5.172}$$

In order for condition (5.172) to guarantee invariance it is necessary that

$$Co\left\{v_1^0, \dots, v_n^0, -v_1^0, \dots, -v_n^0\right\} \subseteq Co\left\{v_1, \dots, v_n, -v_1, \dots, -v_n\right\}. \tag{5.173}$$

This is ensured by condition (5.167). To see this note that (5.167) is equivalent to $\frac{1}{a_p} v_p = v_p^0$. Condition (5.171) ensures $1 \geq \frac{1}{a_p} \geq 0$. It can easily be shown that for each $p = 1, \dots, n$ there exist $\lambda_1^p, \lambda_2^p \geq 0$ such that $v_p^0 = \lambda_1^p v_p + \lambda_2^p(-v_p)$ with $\lambda_1^p - \lambda_2^p = \frac{1}{a_p}$ and $\lambda_1^p + \lambda_2^p = 1$. Therefore, it is ensured that $v_p^0 \in Co\left\{v_1, \dots, v_n, -v_1, \dots, -v_n\right\}$. Because this holds for all p it is ensured that $Co\left\{v_1^0, \dots, v_n^0, -v_1^0, \dots, -v_n^0\right\} \subseteq Co\left\{v_1, \dots, v_n, -v_1, \dots, -v_n\right\}$. Condition (5.168) ensures the input constraints are satisfied. From condition (5.167) follows that $\left|\det\left(\begin{bmatrix} v_1 & \dots & v_n \end{bmatrix}\right)\right| \sim \prod_{i=1}^{n} a_i$. Therefore the cost function (5.163) ensures the volume of the set $Co\left\{v_1, \dots, v_n, -v_1, \dots, -v_n\right\}$ is maximized. □

Remark 5.9. *The feedback gain K can be recovered from the solution via (5.162).*

5.4.3 Rate constraints

Constraints (5.3) and (5.4) are constraints on the states and inputs of the system. In some applications additional constraints on the rate at which the states and/or inputs vary, need to be taken into account. These constraints are called rate constraints. Assume the rate at which the states may vary is bounded by

$$\|x_{k+1} - x_k\|_\infty \leq \Delta_s, \tag{5.174}$$

then following theorem provides necessary conditions for the rate constraints to hold:

Theorem 5.13. *The rate constraint defined by (5.174) is satisfied for the LTV system (5.1) under the linear feedback K if following necessary conditions are satisfied*

$$\begin{aligned} (H_{1j} - H_{2j}) &= (A_j - I)P + B_j Q && (5.175)\\ \begin{bmatrix} H_{1j} & H_{2j} \end{bmatrix} \mathbf{1} &\leq \Delta_s \mathbf{1} && (5.176)\\ H_{1j}, H_{2j} &\geq 0 && (5.177)\\ & && (5.178) \end{aligned}$$

Proof

The difference $x_{k+1} - x_k$ can be expressed as $x_{k+1} - x_k = (A_j - I)x_k + B_j u_k$. In case of a state feedback gain K the difference becomes $x_{k+1} - x_k = (A_j - I + B_j K)x_k$. Now assume the set $\varphi_2 = \{x \mid \|(A - I + BK)x\|_\infty \leq \Delta_s\}$, then the rate constraint (5.174) is satisfied if the low-complexity polytope (5.18) is a subset of φ_2. By means of Lemma 5.5 this leads to the conditions

$$\begin{aligned} (H_{1j} - H_{2j})V &= A - I + BK && (5.179)\\ \begin{bmatrix} H_1 & H_2 \end{bmatrix} \mathbf{1} &\leq \Delta_s \mathbf{1} && (5.180)\\ H_{1j}, H_{2j} &\geq 0. && (5.181) \end{aligned}$$

Using the nonlinear transformation $P = V^{-1}$ and $Q = KP$ the invariance conditions (5.175)-(5.177) are obtained. □

Note that the invariance conditions (5.175)-(5.177) are linear in the unknowns. The feedback gain K can be recovered as $K = QP^{-1}$. Another interesting constraint that is often encountered in practical applications is a rate constraint on the inputs. Such a rate constraint can be defined as

$$\|K(x_{k+1} - x_k)\|_\infty \leq \Delta_u. \tag{5.182}$$

Following conditions for satisfaction of rate constraint (5.182) follow straightforward from Theorem 5.13:

$$\begin{aligned} (H_{1j} - H_{2j}) &= K(A_j - I)P + KB_j KP && (5.183)\\ \begin{bmatrix} H_{1j} & H_{2j} \end{bmatrix} \mathbf{1} &\leq \Delta_s \mathbf{1} && (5.184)\\ H_{1j}, H_{2j} &\geq 0. && (5.185) \end{aligned}$$

Note that these conditions are only linear for a fixed value of the feedback gain K.

5.5 Examples

Example 1 This example illustrates that the algorithms in this paper can lead to bigger feasible invariant sets than the methods described in [47]. The example deals with the

constrained control problem of a DC electric motor with independent excitation and is taken from [33]. The system matrices of the linear continuous-time system are given by:

$$A = \begin{bmatrix} -0.07 & -0.86\,(1+q_1) \\ 0.06\,(1+q_1) & -q_2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{5.186}$$

The uncertainty is defined by

$$Q = \{-0.2 \leqslant q_1 \leqslant 0.2, 0.0085 \leqslant q_2 \leqslant 0.5\}\,. \tag{5.187}$$

Note that this implies that the uncertainty polytope Ω consists of 4 linear systems that define its vertices. The state constraints are given by

$$\begin{bmatrix} -10 \\ -10 \end{bmatrix} \leqslant x \leqslant \begin{bmatrix} 10 \\ 10 \end{bmatrix} \tag{5.188}$$

and the input constraints by

$$-10 \leqslant u \leqslant 10. \tag{5.189}$$

This system is then discretized by the Euler discretization scheme using a sampling time $T_s = 0.1s$. The methods described in this chapter are applied on this discretized LTV system.

The set S_0 in Figure 5.1 is the initial invariant feasible set obtained using Algorithm 5.1. The primary vertices of this set are given by

$$v_1 = \begin{bmatrix} 0.598 \\ 2.049 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 0.071 \\ -1.380 \end{bmatrix} \tag{5.190}$$

and it is invariant (see different paths of the vertices in the figure) for the linear feedback gain

$$K = \begin{bmatrix} 15.475 & -7.020 \end{bmatrix}. \tag{5.191}$$

The set S_1 is the resulting set after increasing the volume of S_0 with the sequence of convex programs described in Theorem 5.7. The set S_1 consists of the primary vertices

$$v_1 = \begin{bmatrix} 10 \\ 4.092 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 8.663 \\ -2.756 \end{bmatrix} \tag{5.192}$$

and is invariant (see different paths of the vertices in the figure) for the linear feedback gain

$$K = \begin{bmatrix} 0.253 & -2.833 \end{bmatrix}. \tag{5.193}$$

Note that the iterative method described in Theorem 5.3 (which is the method described in [47]) is not capable of increasing the volume of the set S_0. Therefore, the method described in this work outperforms the one described in Theorem 5.3.

Example 2

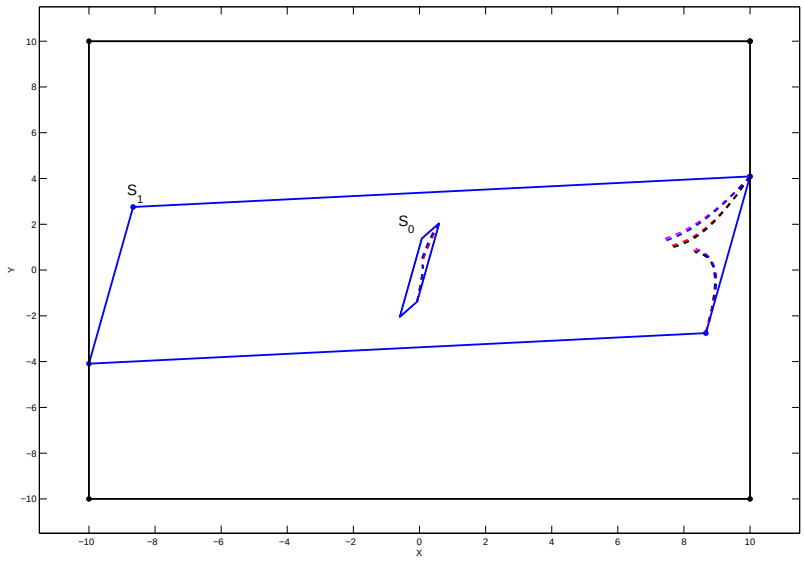


Figure 5.1: Feasible invariant polytopic sets for Example 1. The dashed lines indicate the trajectories of the vertices for several realizations of the uncertain A-matrix.

In this example an invariant set for a linear system with bounded disturbances is calculated. The linear system is a discrete-time state space model and is taken from [109]. The model is described as follows:

$$x_{k+1} = \begin{bmatrix} 1.1 & 1 \\ 0 & 1.3 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k + d_k \tag{5.194}$$

with the input constraint

$$\mathbb{U} = \{u | \, \|u\|_\infty \leq 1\} \tag{5.195}$$

and the disturbance bounded by $\|d\|_\infty \leq 0.1$. The set S_0 in Figure 5.2 is an initial invariant set for this system and is obtained by means of Optimization problem 5.2:

$$\varphi = \left\{ x \, \middle| \, \left\| \begin{bmatrix} 3.825 & -0.309 \\ -0.309 & 0.969 \end{bmatrix} x \right\|_\infty \leqslant 1 \right\} \tag{5.196}$$

This set is invariant (see different paths of the vertices) under the following linear feedback gain K:

$$K = \begin{bmatrix} -0.921 & -0.935 \end{bmatrix}. \tag{5.197}$$

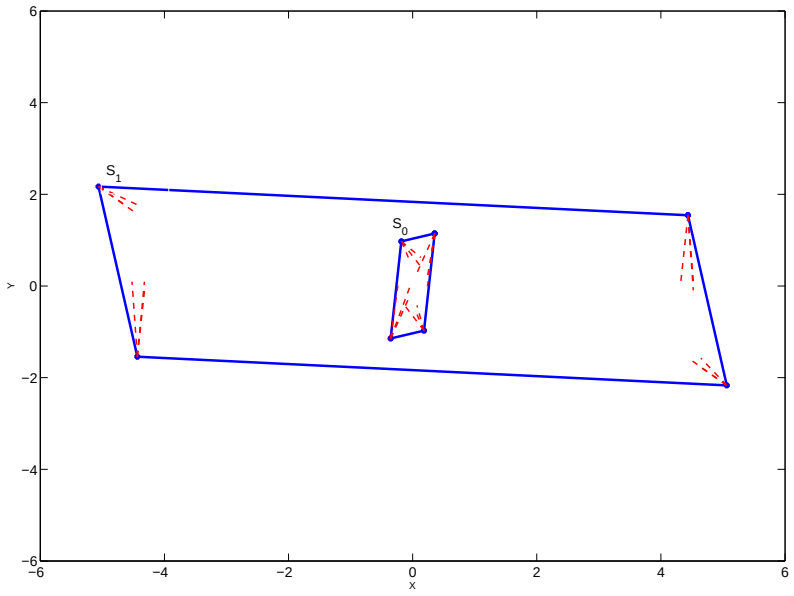


Figure 5.2: Feasible invariant polytopic sets for Example 2. The dashed lines coincide with the closed-loop trajectories of the vertices for several realizations of the disturbance.

Applying Theorem 5.7 with the set S_0 as initial set leads to the maximal feasible invariant set S_1 defined by:

$$\varphi = \left\{ x \left| \left\| \begin{bmatrix} 0.213 & 0.036 \\ 0.114 & 1.128 \end{bmatrix} x \right\|_\infty \leqslant 1 \right. \right\} \tag{5.198}$$

The set is invariant and feasible under the following linear feedback gain K:

$$K = \begin{bmatrix} -0.148 & -0.872 \end{bmatrix}. \tag{5.199}$$

Example 3

LTI system

This example illustrates the advantage in computation time and memory usage of the methods in this paper compared with the method described in Theorem 5.3 due to the reduction in numbers of the optimization variables and constraints. The example consists of the control of the temperature profile of an one-dimensional bar [12]. In order to control the temperature profile 3 inputs are present which are the temperature of the bar at 3 locations (see [12]). The model consists of parabolic Partial Differential

Equations (PDE's). These PDE's are simulated by discretization in spatial and temporal domain. By dividing the one-dimensional bar into 400 sections (spatial discretization) and taking a sampling time of 1 sec (temporal discretization) a linear system was obtained in [12] consisting of 398 states. In order to reduce this complexity a model reduction was performed by means of Proper Orthogonal Decomposition (POD) [12] leading to following reduced order model:

$$A_1 = \begin{bmatrix} 0.858 & -0.053 & 0.021 & 0.193 & -0.029 & 0.027 \\ -0.053 & 0.716 & 0.039 & 0.051 & 0.179 & 0.162 \\ 0.021 & 0.039 & 0.698 & 0.031 & 0.135 & -0.206 \\ 0.193 & 0.051 & 0.031 & 0.567 & 0.053 & -0.014 \\ -0.029 & 0.179 & 0.135 & 0.053 & 0.658 & -0.031 \\ 0.027 & 0.162 & -0.206 & -0.014 & -0.031 & 0.504 \end{bmatrix}, \tag{5.200}$$

$$B_1 = \begin{bmatrix} -0.438 & -1.607 & -0.474 \\ -0.074 & -1.479 & 1.389 \\ -1.647 & 1.011 & 0.204 \\ 0.866 & 1.869 & 0.821 \\ 0.633 & 0.004 & -1.364 \\ -0.997 & 1.185 & -0.554 \end{bmatrix} \tag{5.201}$$

subject to the following input constraints

$$\overline{u} = [1, 1, 1]^T, \quad \underline{u} = [-1, -1, -1]^T \ . \tag{5.202}$$

An initial feasible invariant set is obtained using the LMI of Algorithm 5.1. This set is then used as starting point for the optimization schemes of Theorem 5.7 and the procedures of Theorem 5.3 and Optimization problem 5.1. The results of the different optimization schemes implemented on a P4-2.8 GHz PC with 3 GB RAM are summarized in table 5.1. From table 5.1 it can be seen that the optimization scheme of Theorem 5.7 needs significantly less memory and computation time than the one described in Theorem 5.3 and therefore outperforms the method of Theorem 5.3. Also note that for a dimension higher than 6 it is not possible to use the method of Theorem 5.3 due to the excessive memory requirements. The method described in Theorem 5.3 also results in a much larger set than the method described in Optimization problem 5.1. The reason for this is that in Optimization problem 5.1 the feedback matrix K is not considered to be an optimization variable and because in the optimization of Optimization problem 5.1 only parallel movements of the facets are considered.

Remark 5.10. *To solve the programming problems in this example we used CVX, a package for specifying and solving convex programs [63,64].*

LTV system

We now assume that there is uncertainty on system (5.200)-(5.201). This uncertainty is described by following LTV system:

Table 5.1: *Optimality and computational load for Example 3.*

Algorithm	$\lvert det(P)\rvert$	variables	time(s)
Algorithm 5.1	2.204×10^3	63	2.58
Optimization problem 5.1	1.35×10^4	6	2.27
Theorem 5.3	7.056×10^4	16514	2410.56
Theorem 5.7	5.721×10^6	235	21.11

$$
\begin{aligned}
&A_1\\
&B_1\\
&A_2 = A_1 + 0.25I\\
&B_2 = B_1\\
&A_3 = A_1 - 0.25I\\
&B_3 = B_1
\end{aligned}
$$

with I the unitary matrix, and the matrices A_1 and B_1 defined by (5.200) and (5.201). The following matrix defines the initial invariant feasible set obtained after applying Algorithm 5.1:

$$
V = \begin{bmatrix}
3.88 & -0.85 & 0.59 & 1.87 & -0.26 & -0.40\\
-0.85 & 2.86 & 1.20 & 0.42 & 2.49 & 0.38\\
0.59 & 1.20 & 2.40 & 0.93 & 2.08 & -0.93\\
1.87 & 0.42 & 0.93 & 1.27 & 0.80 & -0.27\\
-0.26 & 2.49 & 2.08 & 0.80 & 2.90 & -0.29\\
-0.40 & 0.38 & -0.93 & -0.27 & -0.29 & 0.70
\end{bmatrix} \tag{5.203}
$$

with the volume proportional to $\left|det\left(V^{-1}\right)\right| = 2.35 \times 10^3$. The corresponding feedback matrix is

$$
K = \begin{bmatrix}
0.71 & 0.40 & 0.98 & 0.59 & 0.78 & -0.38\\
0.85 & 0.07 & -0.02 & 0.44 & 0.44 & 0.11\\
0.71 & -1.09 & -0.43 & 0.03 & 0.03 & -0.13
\end{bmatrix}. \tag{5.204}
$$

After application of the convex sequence described in Theorem 5.7 the robustly invariant set is obtained defined by following matrix:

$$
V = \begin{bmatrix}
2.79 & -0.57 & 0.46 & 1.36 & -0.14 & -0.3\\
-0.35 & 1.46 & 0.73 & 0.28 & 1.33 & 0.13\\
0.17 & 0.75 & 1.1 & 0.42 & 1.08 & -0.35\\
1.07 & 0.33 & 0.49 & 0.75 & 0.49 & -0.09\\
-0.04 & 0.97 & 0.81 & 0.34 & 1.13 & -0.11\\
0.01 & 0.41 & -0.14 & 0.08 & 0.18 & 0.26
\end{bmatrix} \tag{5.205}
$$

which has a volume proportional to $\left|det\left(V^{-1}\right)\right| = 4.35 \times 10^5$. The corresponding feedback matrix is

$$K = \begin{bmatrix} 0.63 & 0.37 & 0.67 & 0.50 & 0.59 & -0.20 \\ 1.01 & -0.09 & 0.08 & 0.50 & -0.02 & -0.01 \\ 0.58 & -0.53 & -0.16 & 0.15 & -0.42 & -0.07 \end{bmatrix}. \tag{5.206}$$

5.6 Robust feedback control for a river system

In this section a robust controller will be designed for the conceptual model describing the dynamics of the upstream part of the Demer river and compared with a standard LQR controller. The model of the upstream part is depicted in Figure 2.11. Because the purpose of this example is to obtain improved setpoint regulation which occurs during periods of moderate rainfall, the model can be simplified. The gates A and D are closed during periods of moderate rainfall and therefore q_a, q_d, v_s, and h_s can be omitted. The resulting system consists of 6 states, 2 disturbance inputs and the gate K_7. The goal of the controller is to steer h_{opw} to a reference level of 21.5 m. In order to design such a controller a steady state needs to be determined. This was done by trial and error and the following suitable steady state was obtained:

$$\begin{aligned} q_{opw} &= 6.85 m^3/s \\ q_{man} &= 1.5 m^3/s \\ h_{opw} &= 21.50 m \\ h_{afw} &= 21.25 m \\ q_{k7} &= 8.35 m^3/s \\ q_{afw} &= 8.35 m^3/s \\ k_a &= 22.50 m \\ k_d &= 22.50 m \\ k_{k7} &= 21 m \end{aligned} \tag{5.207}$$

For this steady state the system was linearized and a LQR controller was determined with weighting matrices $Q = 1000I$ and $R = I$. In a next step a simulation was performed in which at each sampling time a perturbation was added to the rainfall-runoff variables q_{opw} and q_{man}. Because of these perturbations the state of the system deviated from the steady state and new linearizations were made. The purpose of these linearizations was to capture the nonlinear behavior of the model by means of an LTV system. For this LTV system a robust feedback controller was determined by means of the algorithms discussed in this chapter. Note that 3 linearized models were used for the LTV system. In order to compare the performance of both controllers a simulation was done in which a constant disturbance was added to the steady state values of the rainfall-runoff variables. The result of this simulation is depicted in Figure 5.3. It can be seen that improved setpoint regulation is obtained due to the robust feedback controller.

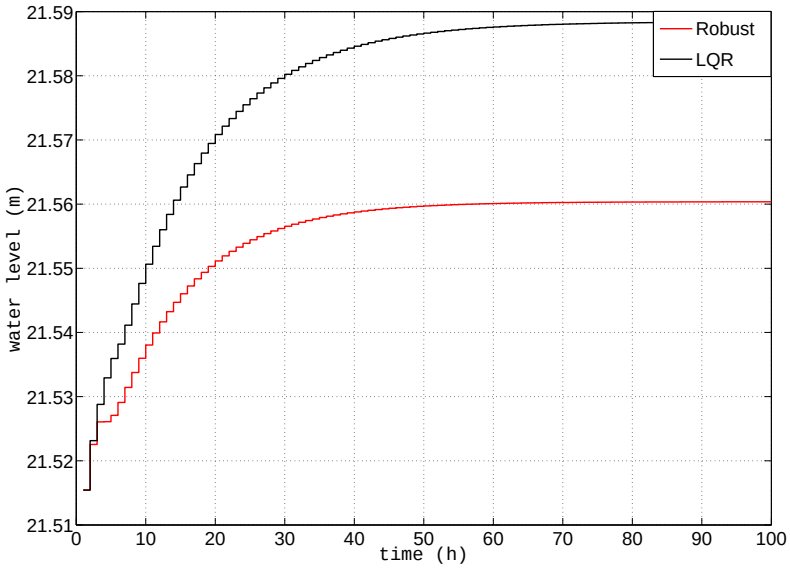


Figure 5.3: Comparison of setpoint regulation with LQR and a robust controller.

5.7 Conclusions

In this chapter new methods were proposed to calculate low-complexity control invariant sets for linear uncertain systems subject to symmetric state and input constraints. The proposed methods consisted of two steps. In the first step an initial invariant set was determined for the uncertain system by means of an LMI. Feasibility of the LMI could only be guaranteed for uncertain systems that were quadratically controllable. In the second step, this set was then used in order to initiate a sequence of convex programs which at the end of each iteration ensured a new invariant set with a volume bigger than that of the invariant set at the beginning of the iteration. In contrast to other methods existing in literature the newly proposed sequence of convex programs scaled linearly with the state dimension. The proposed methods were tested on several practical problems. It was shown that the proposed methods were much faster than existing methods and at the same time also less conservative for uncertain linear systems. This was reflected in the bigger resulting invariant sets. In a final simulation an LQR and a robust controller was designed for setpoint regulation of the upstream part of the Demer. The robust controller was designed with the new techniques outlined in this chapter. By means of a simulation it was shown that the robust controller resulted in improved performance w. r. t. setpoint regulation.

Chapter 6

Conclusions and Future Research

6.1 Conclusions

6.1.1 General

In this thesis the author investigated a control strategy for flooding regulation of the river Demer. The proposed control strategy was the model predictive control (MPC) strategy. The main goal of this thesis was to verify whether MPC is a better alternative w.r.t. flooding regulation than the currently three-position controller in use. In order to make this comparison first a satisfactory MPC scheme needed to be worked out and implemented. In order to develop such a scheme several subgoals needed to be solved:

- *conceptual model:* In order for MPC to predict the impact of control actions on the future states of the system a model is necessary that accurately represents the dynamics of the river Demer.
- *nonlinear MPC:* A nonlinear MPC scheme is needed in order to cope with the nonlinear prediction model. Besides that the MPC scheme needs to deal with difficulties like uncontrollable modes of the gate equations and infeasibility of the constraints during heavy rainfall.
- *nonlinear MHE:* MPC can only make predictions if the current states of the system are measured. However, in practice only a subset of the states is measured. The unmeasured states need to be estimated based on past and current measurements. Since the underlying model is nonlinear, this state estimation is performed by a nonlinear moving horizon estimator.

In this thesis each of these subgoals were tackled and solved in Chapters $2, 3$ and 4. A comparison was also made between the performance of MPC and that of

the three-position controller. Besides flooding regulation the author of this thesis also investigated new mathematical techniques for the efficient calculation of low-complexity controlled invariant sets for uncertain linear time-varying systems. These techniques were then used for improved setpoint regulation of the upstream part of the Demer river. These results were discussed in Chapter 5.

6.1.2 Flooding regulation of the river Demer

6.1.2.1 Conceptual model

In Chapter 2 a conceptual model was derived for the Demer basin. The conceptual model was a simplification of the real hydraulic model of the river Demer. It was assumed that this conceptual model received the rainfall-runoff data from the hydrologic model. The conceptual model was based on different river modeling concepts. Flood control reservoirs and large water storage areas were modeled as simple storage nodes, maintaining water continuity. A continuous relationship was implemented between the storage volume and the mean water level. Long river reaches were modeled by means of the water surface concept. In this concept one starts at the downstream location along the reach and considers water level changes to upstream. A nonlinear relationship is then assumed between the discharge and the water level based on the equation of Manning. The parameters of this model were calibrated based on simulation results with the full hydrodynamic InfoWorks model. This calibration process was based on the flood events of September 1998 and January 2002. Validation was done based on the full set of flood events during the past 15 years; being the flood event of $1995, 1998, 1999 - 2000$ and 2002. The resulting model was a good approximation of the full hydrodynamic Infoworks model and in the remainder chapters it was therefore assumed that the conceptual model was a perfect representation of the real dynamics of the Demer river.

6.1.2.2 Nonlinear MPC

In Chapter 3 a MPC scheme was outlined for flooding regulation of the Demer. The first difficulty that was overcome, was the nonlinearity of the conceptual model of the Demer. This was achieved by the implementation of a nonlinear MPC scheme. The proposed nonlinear MPC scheme was initiated by an initial guess of the optimal future input trajectory based on the optimal solution obtained at the previous time step. Based on this simulation the nonlinear dynamics of the model were linearized around the predicted trajectories which lead to the approximation of the nonlinear model by a linear time-varying system. The optimization problem was then solved with the nonlinear system dynamics replaced by local linear time-varying approximations of the nonlinear model. The resulting optimization problem was a QP that could be solved efficiently. Since the linear time-varying approximations were only valid approximations not too far away from the points around which was linearized, a line-search was performed in order to guarantee the new input trajectories to let the cost function decrease. These more optimal inputs were then used to perform a new simulation in order to obtain new linear time-varying approximations of the nonlinear

system dynamics and the procedure was repeated until convergence. The second difficulty that arose was the uncontrollability problem of the gated weirs in the river Demer. If a gated weir gets into the throat control mode the discharge over the gate is independent of the gate position and therefore the gate is uncontrollable in this mode. A method was proposed to overcome this method by enforcing the gate to leave this mode via a proper modification of the objective function. The third difficulty that was overcome was the constraint handling during heavy rainfall. During periods of heavy rainfall not all constraints could be satisfied. Therefore a constraint strategy was proposed that removed constraints with less priority until a feasible solution could be obtained. At the same time the cost related to the water levels corresponding with the removed constraints, was modified in order to minimize the violation of the removed constraints.

This nonlinear MPC scheme was then tested on the conceptual model for the historical flood of 1998 and its performance was compared with that of the three-position controller. Two different tests were performed. In the first test the rainfall-runoff predictions used by MPC were assumed to coincide exactly with the real rainfall-runoff from 1998. In a second test uncertainty was assumed on these rainfall-runoff predictions in order to assess the robustness of the proposed MPC strategy. The regulation cost and the flood cost of the simulations under perfect knowledge of future rainfall-runoff predictions showed that NMPC outperformed the three-position controller. NMPC only performed worse than the three-position controller for the regulation of water level hbg_{opw}. However, comparing the three-position controller with the nonlinear model predictive controller for water level hbg_{opw} is a bit unfair as the three-position controller regulates with a sampling time of 5 minutes, while the nonlinear model predictive controller regulates at a much larger sampling time of 1 hour. The second simulation based on uncertain rainfall-runoff predictions showed that despite the added uncertainty the nonlinear model predictive controller performed significantly better than the three-position controller. However, the performance of the nonlinear model predictive controller decreased compared to the NMPC simulations in which the rainfall-runoff predictions were exactly known.

6.1.2.3 Nonlinear MHE

In Chapter 4 a MHE scheme was outlined for the state estimation of the states in the river system. In order for the model predictive controller to predict the future states it needs to know the current state of the system. This is provided by the state estimator. It was pointed out that all the works in literature concerning river regulation deal about steering the water levels to the desired setpoint. This is achieved by assuming a simple linear model and therefore a linear Kalman filter has always been used for river control applications. However, the purpose in this thesis was to develop a control scheme for flooding regulation and therefore a nonlinear model needed to be assumed. In contrast to existing work concerning river regulation, it was not possible to use a simple linear Kalman filter but a more advanced estimator was needed that could deal with the nonlinearities of the river model and with constraints imposed on the estimated states.

An estimator that could deal with these challenges was a nonlinear moving horizon estimator and therefore in Chapter 4 such an estimator was outlined in detail. In order to test the proposed NMHE scheme two simulations were performed based on the historical rainfall-runoff data of 1998 in which the model predictive controller obtained the state estimates from the nonlinear moving horizon estimator. Similar to the experiments performed in Chapter 3 the first simulation assumed perfect knowledge of the future rainfall-runoff predictions, while the second simulation assumed uncertainty in the rainfall-runoff predictions. In the simulations it was also assumed that all water levels were measured. In order to assess the robustness of the nonlinear moving horizon estimator Gaussian noise was added to the water level measurements. The simulations showed that despite the uncertainty added to the water level measurements and despite the nonlinear model, the nonlinear moving horizon estimator was capable of obtaining very accurate state estimates. The simulations also showed that the combined NMPC-NMHE scheme outperformed the three-position controller, despite the extra uncertainty coming from the state estimation process. Note that the river model used in Chapter 4 was slightly different than the one used in Chapter 3. In order to obtain good estimates with the NMHE scheme the model needed to be smoothed. This smoothing process did not affect the accuracy of the model. As a consequence of this smoothing process the simulation results of the control scheme were different than those obtained in Chapter 3.

6.1.3 Low-complexity controlled invariant sets

In Chapter 5 a method was discussed that allowed to efficiently calculate low-complexity controlled invariant sets for linear systems with polytopic uncertainty description. These types of set had the advantage that for a system with n state dimension they could be described $2n$ linear constraints, which makes the set scalable with the state dimension. As discussed in Chapter 5 methods exist to calculate low-complexity polytopes but they are too conservative or scale badly with the state dimension. Therefore, a new method was outlined that was less conservative than the already existing methods but at the same time scaled well with the state dimension.

The proposed method consisted of two steps. In the first step, a method was discussed that allowed to calculate an initial invariant low-complexity polytope by solving a convex LMI. Such convex invariance conditions for low-complexity polytopic sets could not be found in literature and are therefore new. However, guarantees for the existence of a solution for the convex LMI could only be given for linear uncertain systems that were quadratically controllable. In the second step the initial invariant set obtained from the first step was used in order to initiate a sequence of convex programs that iteratively increased the volume of the invariant set until convergence. In order to test the efficiency of the proposed method, the method was applied to several real-life examples. The results of these examples showed that the newly proposed algorithm outperformed all existing methods found in literature and this w. r. t. volume size of the invariant set as well as computation time. This method was then used for determining

a linear feedback gain for setpoint regulation of the upstream part of the Demer river. The obtained linear feedback gain resulted in better setpoint regulation compared to the design of a standard LQR controller for a linearization of the nonlinear model around a steady state.

6.2 Future research

6.2.1 Flooding regulation

Several interesting future research topics can be identified in the area of flooding regulation. Some of them are summarized in the following:

(i) In this thesis a conceptual model was derived that approximated the dynamic behavior of the river Demer. Based on this conceptual model a NMPC and

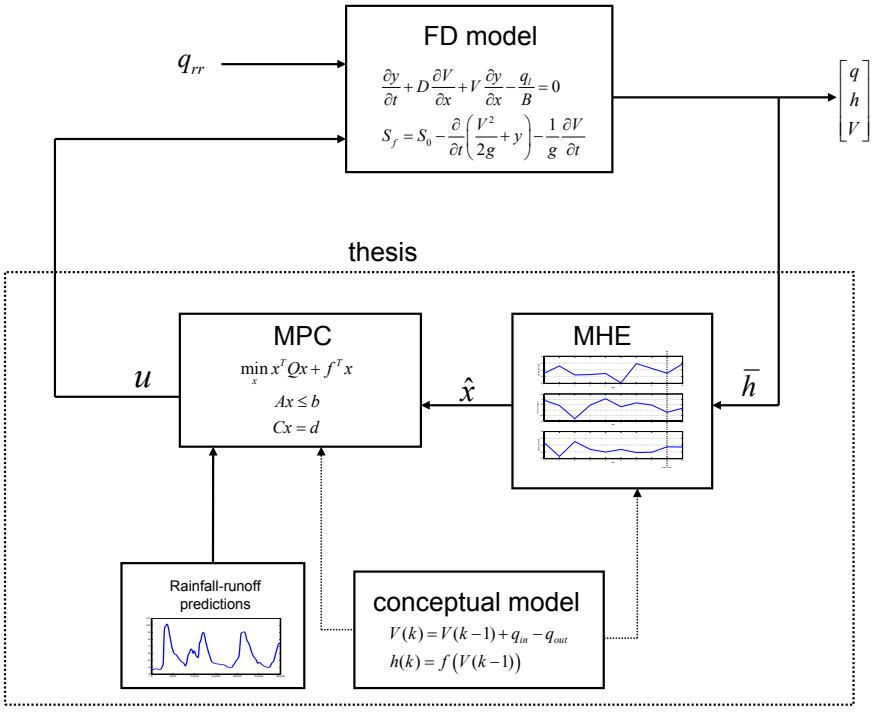


Figure 6.1: Future scheme for testing the practical validity of the MPC-MHE scheme developed in this thesis.

NMHE scheme were proposed and tested. In order to verify the robustness of both the NMPC and NMHE scheme uncertainties were added to the rainfall-runoff predictions and the water level measurements. These tests showed that the proposed schemes were indeed robust w.r.t. the added uncertainties and resulted in an improvement when compared with the results obtained with the three-position controller. However, in these tests it was assumed that the conceptual model was a perfect representation of the reality, which is not the case. Future research should extend the tests in this thesis and verify whether the proposed control scheme is also robust w.r.t. the uncertainty of the conceptual model. This can be done by performing the test depicted in figure 6.1. In this figure the different components implemented and tested in this thesis are located inside the dashed box. In order to assess whether the proposed scheme is robust w.r.t. model uncertainties a test should be performed where the optimal inputs calculated by the NMPC scheme are applied to the finite-difference InfoWorks model. If the results of this test demonstrate that the proposed control scheme still outperforms the three-position controller, then the control scheme will be used by the local water administration as a decision support tool.

(ii) In this thesis the focus was to keep the water levels as much as possible under their corresponding flood levels. This is reflected in the design of the cost function of the model predictive controller. In order to obtain better flooding regulation an interesting future research topic is to extend the simplified model with a simplified model of the flood map of the Demer basin. This would allow the model predictive controller to take real financial damage into account and therefore make more optimal decisions.

(iii) The VMM is only responsible for a subset of all river basins in Belgium. Other river basins are the responsibility of other water administrations. The VMM cannot base the control decisions on its own river basins only as this could lead to flood reduction in its own river basins at the expense of severe floods in the other river basins. Therefore, information must be exchanged between the several water administrations and this information must be used for control decision purposes. This can be achieved by implementing model predictive controllers for each water administration and make them interact with each other. Such a control scheme is called a multi-agent MPC scheme and has already been applied in practice for other applications (e.g. for power networks in [89]). An interesting future research topic would therefore be to test such a multi-agent MPC scheme for flooding regulation.

6.2.2 Low-complexity controlled invariant sets

In this thesis low-complexity controlled invariant polytopic sets were discussed for uncertain linear systems. The discussed sets were described by the following equation:

$$\varphi = \{x | \, \|Vx\|_\infty \leq 1\} \tag{6.1}$$

with $x \in \mathbb{R}^n$ and $V \in \mathbb{R}^{n \times n}$. The procedure describing the efficient calculation of invariant sets described by (6.1) consisted of two steps. In the first step an initial invariant set was calculated by the solution of a convex program (LMI). However, guarantees could only be given for quadratically controllable uncertain systems. An interesting topic of research is the determination of new convex invariance conditions for the proposed invariant sets that can give guarantees to a larger class of uncertain systems. Another interesting topic of research is to extend the results discussed in this thesis to more general low-complexity polytopes with a non-square matrix $V \in \mathbb{V}^{m \times n}$ and $m > n$. Sets of this type typically lead to bigger invariant sets than those described by a square matrix V and at the same time they still scale well with the state dimension.

Bibliography

[1] http://www.lin.vlaanderen.be/awz/waterstanden/hydra/hic_demerbekken.htm.

[2] http://asmslibrary.editme.com.

[3] http://www.overstromingsvoorspeller.be.

[4] Wetenschappelijke ondersteuning van het waterbeheer. De dienstverlening van het Hydrologisch Informatiecentrum (HIC). http://www.waterstanden.be/publicaties/hic-brochures/hic-brochure.pdf/at_download/file.

[5] Mosek 6.0 MATLAB optimization toolbox users manual. http://www.mosek.com/fileadmin/products/6_0/tools/doc/html/toolbox/index.html.

[6] Wallingford Software and Halcrow (UK). FloodWorks, version 9.5, 2008.

[7] Wallingford Software and Halcrow (UK). InfoWorks-RS, version 8.5, 2008.

[8] http://www.ipcos.be.

[9] De digitale Demer. Een nieuw en krachtig instrument voor waterpeilbeheer. Ministerie van de Vlaamse gemeenschap Std., december 2003. http://www.lin.vlaanderen.be/awz/waterstanden/hydra/publicaties/demer-brochure.pdf.

[10] De digitale dender. Een nieuw en krachtig instrument voor waterpeilbeheer. De dienstverlening van het Hydrologisch Informatiecentrum (HIC). http://www.waterstanden.be/publicaties/hic-brochures/dender-brochure.pdf.

[11] *Neural control of nonlinear non-minimum phase dynamical systems*. in the Proceedings of the Internation Conference on Artificial Neural Nets and Genetic Algorithms, France, 1995.

[12] M. Agudelo, J. Espinosa, and B. De Moor. Application of POD and predictive control techniques to the control of the temperature profile of an one-dimensional bar. *Proceedings of XII Latin-American Congress on Automatic Control (CLCA2006)*, page CDROM, 2006.

[13] M. Agudelo, J. Espinosa, and J. Vandewalle. Control of the synthesis section of a urea plant by means of an MPC controller. *Computer Aided Chemical Engineering*, 21:1305–1310, 2006.

[14] A.C. Antoulas. *Approximation of large-scale dynamical systems*. SIAM, Philadelphia, 2005.

[15] C. Aurora, L. Magni, P. Colombo, F. Pretolani, and G. Villa. Predictive control of thermal power plants. *International Journal of Robust and Nonlinear Control*, 14(4):415–433, 2004.

[16] Pluymers. B. *Robust model based predictive control - an invariant set approach.* PhD thesis, KULeuven, 200.

[17] D. Balogun, O.S. Hubbard, and J.J. De Vries. Automatic control of canal flow using linear quadratic regulator theory. *Journal of Hydraulic Engineering*, 114(1):75–102, 1988.

[18] H. Bang and O. Choong-Seok. Predictive control for the attitude maneuver of a flexible spacecraft. *Aerospace Science and Technology*, 8(5):443–452, 2004.

[19] J.R. Banga, E. Balsa-Canto, C.G. Moles, and A.A. Alonso. Improving food processing using modern optimization methods. *Trends in Food Science & Technology*, 14(4):131–144, 2003.

[20] T. Barjas Blanco, M. Cannon, and B. De Moor. On efficient computation of low-complexity controlled invariant sets for uncertain linear systems. *International Journal of Control*, 83(7):1339–1346, 2010.

[21] T. Barjas Blanco and B. De Moor. Polytopic invariant sets for continuous-time systems. In *Proceedings of the European Control Conference 2007*, pages 5087–5093, 2007.

[22] T. Barjas Blanco and B. De Moor. Robust MPC based on symmetric low-complexity polytopes. In *the 17th Mediterranean Conference on Control and Automation (MED09), Thessaloniki, Greece*, page CDROM, june 2009.

[23] T. Barjas Blanco and B. De Moor. Robustly invariant polytopes based on the L1-norm. In *the 28th IASTED International Conference on Modelling, Identification and Control (MIC2009), Innsbruck, Austria*, page CDROM, February 2009.

[24] T. Barjas Blanco, P. Willems, J. Berlamont, B. De Moor, K. Cauwenberghs, S. Rombauts, and F. Raymaekers. Real-time sturing van wachtbekkens. *Water*, 25:42–47, 2008.

[25] T. Barjas Blanco, P. Willems, P. Chiang, K. Cauwenbergh, B. De Moor, and J. Berlamont. *Intelligent Infrastructures*, chapter Flood Regulation by means of Model Predictive Control. Springer, 2010.

[26] T. Barjas Blanco, P. Willems, B. De Moor, and J. Berlamont. Flooding prevention of the Demer using model predictive control. In *IFAC World Congress, Seoul, S-Korea*, volume 17, page CDROM, 2008.

[27] T. Barjas Blanco, P. Willems, P. Po-Kuan Chiang, J. Berlamont, and B. De Moor. Flood regulation using nonlinear model predictive control. *Control Engineering Practice*. Accepted for publication.

[28] O. Barrero. *Data assimilation in magnetohydrodynamics systems using Kalman filtering*. PhD thesis, K.U.Leuven, 2005.

[29] E. Bautista, A.J. Clemmens, and R.J. Strand. Salt river project canal automation pilot project: Simulation tests. *Journal of Irrigation and Drainage Engineering*, 132(2):143–152, 2006.

[30] A. Benzaouia and C. Burgat. Regulator problem for a class of linear systems with constrained control. *Systems and Control Letters*, 10(5):357–363, 1988.

[31] A. Berman and R. Plemmons. *Nonnegative matrices in the mathematical sciences*. SIAM, 1994.

[32] G. Bitsoris. On the positive invariance of polyhedral sets for discrete-time systems. *Systems and Control Letters*, 11(3):243–248, 1988.

[33] F. Blanchini. Constrained control for uncertain linear systems. *Journal of Optimization Theory and Applications*, 71:465–484, 1991.

[34] F. Blanchini. Ultimate boundedness control for uncertain discrete-time systems via set-induced lyapunov functions. *IEEE Transactions on Automatic Control*, 39(2):428–433, 1994.

[35] F. Blanchini. Set invariance in control - a survey. *Automatica*, 35:1747–1767, 1999.

[36] S. Boyd, L. El Ghaoui, Feron E., and V. Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[37] S. Boyd, E. Feron, V. Balakrishnan, and L. El Ghaoui. Control system analysis and synthesis via linear matrix inequalities. In *Proceedings of the American Control Conference, San Francisco*, volume 2, pages 2147–2154, 1993.

[38] S. Boyd, E. Feron, V. Balakrishnan, and L. El Ghaoui. History of linear matrix inequalities in control theory. In *Proceedings of the American Control Conference, Baltimore*, volume 1, pages 31–34, 1994.

[39] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[40] E.F. Brater, H.W. King, J.E. Lindell, and C.Y. Wei. *Handbook of hydraulics*. McGraw-Hill Professional Publishing, 1996.

[41] M. Breckpot, T. Barjas Blanco, and B. De Moor. Flood control of rivers with model predictive control – proof of concept based on the river Demer in Belgium. In *the 2010 American Control Conference*, pages 2983–2988, 2010.

[42] T. Brian and A.J. Clemmens. Performance of historic downstream canal control algorithms on ASCE test canal 1. *Journal of Irrigation and Drainage Engineering*, 128(6):365–375, 2002.

[43] T. Wahlin Brian and J. Clemmens Albert. Automatic downstream water-level feedback control of branching canal networks. *Journal of Irrigation and Drainage Engineering*, 132(3):198–207, 2006.

[44] T. Wahlin Brian and A.M. ASCE. Performance of model predictive control on ASCE test canal 1. *Journal of Irrigation and Drainage Engineering*, 130(3):227–238, 2004.

[45] E.F. Camacho and C. Bordons. Nonlinear model predictive control: an introductory survey. *International workshop on assessment and future directions of NMPC, Freudenstadt-Lauterbad, Germany*, 2005.

[46] M. Cannon, V. Deshmukh, and B. Kouvaritakis. Nonlinear model predictive control with polytopic invariant sets. *Automatica*, 39:487–494, 2003.

[47] M. Cannon, V. Deshmukh, and B. and Kouvaritakis. Nonlinear model predictive control with polytopic invariant sets. *Automatica*, 39:1487–1494, 2003.

[48] Alain C. Carelli and Mauricio B. de Souza Jr. Stochastic and deterministic performance assessment of PID and MPC controllers: Application to a hydrotreater reactor. In Caludio Augusto Oller do Nascimento Rita Maria de Brito Alves and Jr. Evaristo Chalbaud Biscaia, editors, *10th International Symposium on Process Systems Engineering: Part A*, volume 27 of *Computer Aided Chemical Engineering*, pages 1635 – 1640. Elsevier, 2009.

[49] Hubert Chanson. *The hydraulics of open channel flow: an introduction*. Elsevier, 1999.

[50] M.H. Chaudhry. *Open channel flow*. Springer, 2008.

[51] V.T. Chow, D.R. Maidment, and L.W. Mays. *Applied hydrology*. McGraw-Hill International Editions, 1988.

[52] A.J. Clemmens and J. Schuurmans. Simple optimal downstream feedback canal controllers: theory. *Journal of Irrigation and Drainage Engineering*, 130(1):26–34, 2004.

[53] A.J. Clemmens and B.T. Wahlin. Simple optimal downstream feedback canal controllers: ASCE test case results. *Journal of Irrigation and Drainage Engineering*, 130(1):35–46, 2004.

[54] A.R. Conn, I.M. Gould, and P.L. Toint. *Trust-region methods*. SIAM Series on Optimization, 2000.

[55] G.B. Dantzig. *Linear programming and extensions*. Princeton University Press, 1963.

[56] M. Diehl, H.J. Ferreau, and N. Haverbeke. Efficient numerical methods for nonlinear MPC and moving horizon estimation. *Proceedings of NMPC 08 workshop in Pavia*, 2008.

[57] K. Euren and E. Weyer. System identification of open water channels with undershot and overshot gates. *Control Engineering Practice*, 15:813–824, 2007.

[58] Gene F. Franklin, David J. Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems*. Prentice Hall PTR, 2001.

[59] G. Gilbert and K. T. Tan. Linear systems with state and control constraints: The theory and application of maximal output admissable sets. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, 1991.

[60] S. Gillijns. *Kalman filtering techniques for system inversion and data assimilation*. PhD thesis, K.U.Leuven, 2007.

[61] M. Gomez, J. Rodellar, and J.A. Mantecon. Predictive control method for decentralized operation of irrigation canals. *Applied Mathematical Modelling*, 26:1039–1056, 2002.

[62] G.C. Goodwin, M.M. Seron, and J.A. De Dona. *Constrained control and estimation: an optimisation approach*. Springer, 2005.

[63] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. http://stanford.edu/~boyd/graph_dcp.html.

[64] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx, July 2010.

[65] N. Haverbeke, M. Diehl, and B. De Moor. A structure exploiting interior-point method for moving horizon estimation. In *IEEE Conference on Decision and Control*, pages 1273–1278, 2009.

[66] J. C. Hennet. Une extension du lemme de Farkas et son application au probleme de regulation lineaire sous constraints. *Comptes Rendus des Seances de l'Academie des Sciences Paris*, 308(Serie I):415–419, 1989.

[67] P. Houtekamer and H. L. Mitchell. Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*, 126:796–811, 1998.

[68] S.J. Julier and K. Uhlmann. Unscented filtering and nonlinear estimation. In *Proceedings of the IEEE*, volume 92, pages 401–422, 2004.

[69] H. K. Khalil. *Nonlinear Systems*. Prentice-Hall, 2nd edition, 1996.

[70] M. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32:1361–1379, 1996.

[71] B. Kouvaritakis, M. Cannon, A. Karas, B. Rohal-Ilkiv, and C. Belavy. Asymmetric constraints with polytopic sets in MPC with application to coupled tanks system. *International Journal of Robust and Nonlinear Control*, 14:341–353, 2004.

[72] Y. Lee, M. Cannon, and B. Kouvaritakis. Extended invariance and its use in model predictive control. *Automatica*, 41:2163–2169, 2005.

[73] Y.I. Lee and Kouvaritakis B. Robust receding horizon predictive control for systems with uncertain dynamics and input saturation. *Automatica*, 36:1497–1504, 2000.

[74] X. Litrico and V. Fromion. Simplified modeling of irrigation canals for controller design. *Journal of Irrigation and Drainage Engineering*, 130(5):373–383, 2004.

[75] X. Litrico and V. Fromion. Tuning of robust distant downstream PI controllers for an irrigation canal pool. i: Theory. *Journal of Irrigation and Drainage Engineering*, 132(4):359–368, 2006.

[76] X. Litrico, V. Fromion, and J.-P. Baume. Tuning of robust distant downstream PI controllers for an irrigation canal pool. ii: Implementation issues. *Journal of Irrigation and Drainage Engineering*, 132(4):369–379, 2006.

[77] X. Litrico, V. Fromion, J.-P. Baume, C. Arranja, and M. Rijo. Experimental validation of a methodology to control irrigation canals based on Saint-Venant equations. *Control Engineering Practice*, 13:1425–1437, 2005.

[78] X. Litrico, P.-O. Malaterre, J.-P. Baume, P.-Y. Vion, and Ribot-Bruno J. Automatic tuning of PI controllers for an irrigation canal pool. *Journal of Irrigation and Drainage Engineering*, 133(1):27–37, 2007.

[79] L. Ljung. *System Identification: Theory for the User (2nd Edition)*. Prentice Hall PTR, 1998.

[80] D.G. Luenberger. *Linear and nonlinear programming, 2nd edition*. Addison-Wesley, 1984.

[81] A.M. Lyapunov. Problème général de la stabilité du mouvement. In *volume 17 of Annals of Mathematics Studies*. Princeton University Press, Princeton, 1947.

[82] P.O. Malaterre. *Modélisation, analyse et commande optimal LQR d'un canal d'irrigation*. PhD thesis, Ecole nationale du génie rural des eaux et des forêts, Paris, 1994.

[83] P.O. Malaterre and J. Rodellar. Design and comparison of multivariable optimal and predictive control on a 2-pool model. *International workshop on regulation of irrigation canals*, 1997.

[84] M. Marinaki and M. Papageorgiou. *Optimal Real-Time Control of Sewer Networks*. Springer, 2005.

[85] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: stability and optimality. *Automatica*, 36:789–814, 2000.

[86] S. Mohinder, S. Grewal, and A.P. Andrews. *Kalman filtering: theory and practice using matlab*. Wiley, 2001.

[87] H. Moradkhani, S. Sorooshian, H.V. Gupta, and P.R. Houser. Dual state-parameter estimation of hydrological models using ensemble Kalman filter. *Advances in Water Resources*, 28:135–147, 2005.

[88] J.C. Neal, P.M. Atkinson, and C.W. Hutton. Flood inundation model updating using an ensemble Kalman filter and spatially distributed measurements. *Journal of Hydrology*, 336:401–415, 2007.

[89] R. Negenborn. *Multi-agent model predictive control with applications to power networks*. PhD thesis, T.U.Delft, 2007.

[90] Y.E. Nesterov and A.S. Nemirovski. *Interior point polynomial algorithms in convex programming*. SIAM Publications, Philadelphia, USA, 1994.

[91] J. Nocedal and S.J. Wright. *Numerical optimization*. Springer Series in Operations Research, 1999.

[92] S.K. Ooi and E. Weyer. Control design for an irrigation channel from physical data. *Control Engineering Practice*, 16:1132–1150, 2008.

[93] P.J. van Overloop. *Model Predictive Control of Open Water Systems.* PhD thesis, T.U.Delft, Faculty of Civil Engineering and Geosciences, Department of Water Management, Delft, The Netherlands., 2006.

[94] P.J. van Overloop, A. Mursi-Batt, K.J. van Heeringen, and R.A.H.A. Thabet. Real-time-control of water quantity and quality in a re-use of drainage water project. In *Proceedings of the first Asian Regional ICID Conference*, 2001.

[95] Gianni Di Pillo and F. Giannessi. *Nonlinear Optimization and Related Topics*. Springer-Verlag New York, Inc., 2000.

[96] B. Pluymers, J. A. Rossiter, J. A. K. Suykens, and B. De Moor. Efficient computation of polyhedral invariant sets for LPV systems and application to robust MPC. In *American Control Conference*, volume 2, pages 804–809, 2005.

[97] B. Pluymers, J. A. K. Suykens, and B. De Moor. Construction of reduced complexity polyhedral invariant sets for LPV systems using linear programming. Technical report, KULeuven, ESAT, 2006.

[98] J.M. Reddy. Local optimal control of irrigation canals. *Journal of Irrigation and Drainage Engineering*, 116(5):616–631, 1990.

[99] R. Rivas Perez, V. Feliu Batlle, and L. Sanchez Rodriguez. Robust system identification of an irrigation main canal. *Advances in Water Resources*, 30:1785–1796, 2007.

[100] D.G. Robertson and H.J. Lee. A least squares formulation for state estimation. *Journal of Process Control*, 5(4):291–299, 1995.

[101] D. C. Rogers and J. Goussard. Canal control algorithms currently in use. *Journal of Irrigation Drainage Engineering*, 124(1):11–15, 1998.

[102] J.A. Rossiter. *Model Predictive Control: A Practical Approach.* CRC press, 2000.

[103] J. Schuurmans. *Control of water levels in open channels.* PhD thesis, Delft University of Technology, The Netherlands, 1997.

[104] J. Schuurmans, A.J. Clemmens, S. Dijkstra, A. Hof, and R. Brouwer. Modelling of irrigation and drainage canals for controller design. *Journal of Irrigation and Drainage Engineering*, 125(6):338–344, 1999.

[105] J. Schuurmans, W. Schuurmans, H. Berger, M. Meulenberg, and R. Brouwer. Control of water levels in the Meuse river. *Journal of Irrigation and Drainage Engineering*, 123(3):180–184, 1997.

[106] A. Singh, J.F. Forbes, P.J. Vermeer, and S.S. Woo. Model-based real-time optimization of automotive gasoline blending operations. *Journal of Process Control*, 10(1):43–58, 2000.

[107] B.L. Stringam and G.P. Merkley. Field application of a fuzzy controller for an irrigation canal in roosevelt utah. *International workshop on the regulation of irrigation canals: state of the art research and applications, RIC97, Marrakech (Morocco)*, pages 349–354, 1997.

[108] B.L. Stringam and G.P. Merkley. Fuzzy controller simulation for local downstream water level control in canals. *International workshop on the regulation of irrigation canals: state of the art research and applications, RIC97, Marrakech (Morocco)*, pages 342–348, 1997.

[109] D. Sui, Le Feng, M. Hovd, and Ong C. J. Decomposition principle in model predictive control for linear systems with bounded disturbances. *Automatica*, 45:1917–1922, 2009.

[110] S.-M. Swei, T. Iwasaki, and M. Corless. Quadratic controllability, strong controllability, and a related output feedback property. *SIAM Journal on Control and Optimization*, 39(5):1373–1390, 2000.

[111] L. Timbe. *River flooding analysis using quasi-2D hydraulic modelling and geospatial data*. PhD thesis, Faculty of Engineering, K.U.Leuven, Leuven, Belgium, 2007.

[112] M.J. Todd. The many facets of linear programming. *Mathematical Programming Series B*, 91:417–436, 2002.

[113] A. Toudeft. Combining neural and adaptive controllers for a non-minimum phase varying time delay system. *Journées Hispano-Françaises: Systèmes Intelligents & Contrôle Avancé*. Barcelona (Spain), Université Polytechnique de Catalogne, November 12-l3, 1996.

[114] A. Toudeft and P. Gallinari. Neural and adaptive controllers for a non-minimum phase varying time-delay system. *Artificial Intelligence in Engineering*, 11:431–439, 1997.

[115] G. Vaes. *The influence of rainfall and model simplification on combined sewer system design*. PhD thesis, Katholieke Universiteit Leuven, 1999.

[116] P. van Overloop, S. Weijs, and S. Dijkstra. Multiple model predictive control on a drainage canal system. *Control Engineering Practice*, 16(5):531–540, 2008.

[117] P.J. van Overloop, J. Schuurmans, R. Brouwer, C.M. Burt, and M. ASCE. Multiple-model optimization of proportional integral controllers on canals. *Journal of Irrigation and Drainage Engineering*, 131(2):190–196, 2005.

[118] B. Voron and A.-P. Bouillot. Application of the fuzzy set theory to the control of a large canal. *International workshop on the regulation of irrigation canals: state of the art research and applications, RIC97, Marrakech (Morocco)*, pages 317–331, 1997.

[119] B.T. Wahlin. Performance of model predictive control on ASCE test canal 1. *Journal of Irrigation and Drainage Engineering*, 130(3):227–238, 2004.

[120] E. Wan and R. van der Merwe. *The Unscented Kalman Filter*. Wiley, 2001.

[121] R. Wardlaw and K. Bhaktikul. Application of a genetic algorithm for water allocation in an irrigation system. *Irrigation and Drainage*, 50(2):159–170, 2001.

[122] E. Weyer. System identification of an open water channel. *Control Engineering Practice*, 9:1289–1299, 2001.

[123] D.C. Wilcox. *Basic fluid mechanics (fourth edition)*. DCW Industries, 2010.

[124] P. Willems. *Probabilistic immision modelling of receiving surface waters*. PhD thesis, Katholieke Universiteit Leuven, 2000.

[125] P. Willems, T. Barjas Blanco, J. Berlamont, and B. De Moor. Evaluation of river flood regulation by means of model predictive control. In *4th International Symposium on Flood Defence (ISFD4)*, page CDROM, 2008.

[126] P. Willems, T. Barjas-Blanco, P. Chiang, K. Cauwenberghs, J. Berlamont, and B. De Moor. Real-time control of urban flooding. In *International Conference on Water and Urban Development Paradigms: Towards an integration of engineering, design and management approaches; CRC Press, Taylor and Francis Group*, pages 265–270, 2008.

[127] Chiang P. Cauwenberghs K. Berlamont J. De Moor B. Willems P., Barjas Blanco T. Real-time control of urban flooding. *in Water & Urban Development Paradigms: Towards an integration of engineering, design and management approaches., (Feyen J., Shannon K., and Neville M. , eds.), CRC Press, Taylor & Francis Group (London, UK)*, pages 265–270, 2008.

[128] G. M. Ziegler. *Lectures on polytopes*. Graduate Texts in Mathematics, Springer, 1995.

Curriculum Vitae

Toni Barjas Blanco was born October 24, 1979 in Genk, Belgium. He received his M.Eng. degree (magna cum laude) in electromechanical engineering from the Katholieke Hogeschool Limburg, Belgium in July 2001 and the M.S. degree (cum laude) in computer science engineering from the Katholieke Universiteit Leuven (K.U.Leuven), Belgium, in 2004. Since 2004 he has been pursuing a Ph.D. under the supervision of Prof. B. De Moor at the Department of Electrotechnical Engineering, K.U.Leuven, Belgium. During his Ph.D. research he extensively cooperated with the Hydraulics Laboratory of the K.U.Leuven, Belgium. Mr. Barjas Blanco was funded by the VMM (Flemish Environment Agency) from 2004 until 2009. His research interest include control and optimization with special emphasis on river regulation.

Publications by the author

Journal papers

- T. Barjas Blanco, M. Cannon and B. De Moor. On efficient computation of low-complexity controlled invariant sets for uncertain linear systems. *International Journal of Control*, 83(7):1339–1346, 2010.
- T. Barjas Blanco, P. Willems, P. Po-Kuan Chiang, J. Berlamont and B. De Moor. Flood regulation using nonlinear model predictive control. *Accepted for publication in Control Engineering Practice*, 2010.
- T. Barjas Blanco, P. Willems, J. Berlamont, B. De Moor, K. Cauwenberghs, S. Rombauts and F. Raymaekers. Real-time sturing van wachtbekkens. *Water*, 25:42–47, 2008.

International Conference papers

- R. Dhulst, K. Verhaegen, T. Barjas Blanco, J. A. K. Suykens, J. Driesen, J. Vandewalle and R. Belmans Teaching in green power production and prediction of electricity consumption. In *Proceedings of the IEEE Power Engineering Society (IEEE-PES), San Francisco, USA*, 1:157-163, 2005.
- T. Barjas Blanco and B. De Moor Polytopic invariant sets for continuous-time systems In *Proceedings of the European Control Conference 2007 (ECC07), Kos, Greece*, pages 5087–5093, 2007.
- T. Barjas Blanco, P. Willems, J. Berlamont and B. De Moor Flood prevention of the Demer river using model predictive control In *Proceedings of the 17th IFAC World Congress, Seoul, South Korea*, pages 3629–3634, 2008.
- P. Willems, T. Barjas Blanco, J. Berlamont and B. De Moor Evaluation of river flood regulation by means of model predictive control In *Proceedings of the 4th International Symposium on Flood Defence (ISFD4), Toronto, Canada*, pages CD, 2008.
- P. Willems, T. Barjas Blanco, P. Po-Kuan Chiang, K. Cauwenberghs, J. Berlamont and B. De Moor Real-time control of urban flooding In *Water & Urban*

Development Paradigms: Towards an integration of engineering, design and management approaches., (J. Feyen, K. Shannon and M. Neville, eds.), CRC Press, Taylor & Francis Group (London, UK), pages 265-270, 2008.

- T. Barjas Blanco and B. De Moor Robustly invariant polytopes based on the L1-norm In *Proceedings of the 28th IASTED International Conference on Modelling, Identification and Control (MIC2009), Innsbruck, Austria*, pages 36–50, 2009.

- T. Barjas Blanco and B. De Moor Robust MPC based on symmetric low-complexity polytopes In *Proceedings of the 17th Mediterranean Conference on Control and Automation (MED09), Thessaloniki, Greece*, pages CD, 2009.

- M. Breckpot, T. Barjas Blanco and B. De Moor Flood control of rivers with model predictive control – proof of concept based on the river Demer in Belgium In *Proceedings of the 2010 American Control Conference (ACC2010), Baltimore, USA*, pages 2983-2988, 2010.

- M. Breckpot, T. Barjas Blanco and B. De Moor Flood control of rivers with model predictive control and moving horizon estimation *Accepted for publication in Proceedings of the 49th IEEE Conference on Decision and Control (CDC2010)*, 2010.

Book Chapters

- T. Barjas Blanco, P. Willems, P. Po-Kuan Chiang, K. Cauwenberghs, B. De Moor and J. Berlamont. Flood regulation by means of model predictive control. In Chapter *Intelligent Infrastructures*, (R. R. Negenborn, Z. Lukszo, and J. Hellendoorn, eds.), Springer, pages 407-436, 2009.

Internal Reports

- T. Barjas Blanco and B. De Moor. General polytopic invariant sets Internal Report 08-44, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2008