Model predictive control of open water systems with mobile operators

R.C. Kassing

Irrigation canal, Mchinji. Credit: FISD/IPS



Delft Center for Systems and Control

# Model predictive control of open water systems with mobile operators

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Systems and Control at Delft University of Technology

R.C. Kassing

June 22, 2018

Faculty of Mechanical, Maritime and Materials Engineering (3mE)  $\cdot$  Delft University of Technology





Copyright © Delft Center for Systems and Control (DCSC) All rights reserved.

# Abstract

In this master thesis, the recently introduced Mobile Model Predictive Control (MoMPC) approach for open water systems with uncertain dynamics is discussed, where there are no sensors or actuators installed in the system that would allow for a fully automatic operation. MoMPC is a configuration of Model Predictive Control (MPC) that explicitly incorporates the role of a mobile operator travelling between the points of interest, i.e., nodes, of the system as instructed by a remote centralised controller. The operator provides the controller with up-to-date measurements from the locations visited and acts as the actuator as required by the remote controller.

In this research, four areas of improvement of MoMPC from literature are explored and some possible solutions are proposed, resulting in a new method, called Multiple-Action Mobile Model Predictive Control (MaMoMPC).

First, the MoMPC approach is generalised to open water systems described as a network, wherein for each node a unique set of actions is possible, e.g., at some nodes only actuation is possible, while at others both measuring and actuation is possible.

Secondly, in MoMPC, controlling the system is only allowed until a predefined control horizon, after which there is often still some setpoint water level error present, which is penalised until the end of the prediction. Cyclic control is proposed to include some simplified estimate of future control in the MPC optimisation problem past the control horizon, without introducing extra computational burden. By including cyclic control the future effort to drive the water levels to the setpoints is better represented in the prediction, improving system performance.

The third area of improvement consists of the consideration of the limitations of the mobile operators in the optimisation problem. Until now, the human operators were assumed to be able to work continuously without requiring breaks. An extension that keeps track of the energy levels of the human operators is proposed, which can be used by the controller to schedule breaks for the human operators.

Finally, another shortcoming in the MoMPC approaches from literature is the discrepancy between the predicted state of the system and the actual state. Depending on the number of operators available the measuring and actuating actions will be sparse in time. Furthermore, the system is subjected to external disturbances and will always have some modelling errors. As a result, there is some uncertainty on the predicted state of the system. This uncertainty about the system can become large when some measurement locations are not visited regularly. Moreover, the uncertainty about the predicted state of the system may result in reduced system performance and constraint violations. To ensure the predicted system state does not drift too far from the actual state, the information gathering capabilities of the system have to be augmented. To that end, three methods to weigh the measurement frequency are proposed.

To evaluate, a case study is performed on a realistic numerical model of the Dez main irrigation canal in Iran. In the first part of the case study, the system performance when adding cyclic control to the Time Instant Optimisation Mobile Model Predictive Control (TIO-MoMPC) approach from literature is evaluated. Including cyclic control improved the reference tracking performance during a scenario without noise with statistical significance. In the second part of the case study, noise is added to the numerical model and the MaMoMPC approach with uncertainty weighing methods and cyclic control is evaluated. The results show that the addition of the uncertainty weighing methods yields enhanced disturbance rejection and reference tracking performance.

# **Table of Contents**

Acknowledgements
------------------

•	:	:	
v	•	•	

1	Intro	oduction	1
	1-1	Open water systems	2
	1-2	Automation of open water systems	3
	1-3	Control of open water systems	3
		1-3-1 Model-based control	4
		1-3-2 Human-operated control	5
	1-4	Problem statement	5
	1-5	Thesis outline	5
2	Мос	lel Predictive Control of Open Water Systems	7
	2-1	Modelling open water systems	7
		2-1-1 Canal reaches	8
		2-1-2 Control structures	10
		2-1-3 Disturbances	11
		2-1-4 Model of an open water system	11
	2-2	Mobile MPC	23
		2-2-1 Mobile MPC for irrigation canals	23
		2-2-2 Optimisation problem formulation	24
		2-2-3 Results	25
	2-3	Mobile TIO-MPC	26
		2-3-1 Optimisation problem formulation	27
		2-3-2 Results	30
	2-4	Conclusions on Model Predictive Control of Open Water Systems	31

R.C. Kassing

3	Mul	tiple-A	ction Mobile Model Predictive Control	33
	3-1	Netwo	rk of an open water system	33
	3-2	Contro	ol algorithm	34
		3-2-1	Operator's schedule	35
		3-2-2	State estimation prediction	39
		3-2-3	Cyclic control	39
		3-2-4	Uncertainty weighing methods	47
		3-2-5	Operator's energy level	49
	3-3	Optim	isation problem formulation	52
	3-4	Solving	g the optimisation problem	53
	3-5	Conclu	isions on Multiple-Action Mobile Model Predictive Control	53
4	Case	e Study	: Cyclic Control and Uncertainty Weighing Methods	55
	4-1	Case s	tudy and MaMoMPC details	56
		4-1-1	Dez main irrigation canal, Iran	56
		4-1-2	MaMoMPC settings	57
		4-1-3	Operator settings	58
		4-1-4	Optimisation problem solver	58
		4-1-5	Offtake schedules	59
		4-1-6	Method used for comparison	60
	4-2	Case s	tudy part I: cyclic control	61
		4-2-1	Fixing the cyclic paths	61
		4-2-2	Simulation issues	62
		4-2-3	Results	62
	4-3	Case s	tudy part II: network uncertainty minimisation	64
		4-3-1	Process noise	66
		4-3-2	Measurement frequency cost	66
		4-3-3	Results	66
	4-4	Conclu	isions on Case Study: Cyclic Control and Uncertainty Weighing Methods .	70
5	Con	clusion	s and Recommendations	71
	5-1	Conclu	isions	71
	5-2	Kecom	Are time delays worth the added complexity?	72
		5-2-1	Are time delays worth the added complexity?	(2
		5-2-2	Is measuring only water levels enough?	73
		5-2-3	How to improve the reliability of the controller?	73
		5-2-4	is distributed control required for larger systems?	74
-				

## A Model Predictive Control

Master of Science Thesis

В	Irrigation Canals Parameters	77
	B-1 Dez main irrigation canal, Iran	77
	B-1-1 Layout	77
	B-1-2 Control structures	78
	B-2 Arizona west-main canal, United States of America	80
	B-2-1 Layout	80
	B-2-2 Control structures	81
С	Quadratic Programming Problem Solving	83
D	Genetic Algorithm Functions	85
	Glossary	93
	List of Acronyms	93

v

# Acknowledgements

This document is a part of my Master of Science graduation thesis.

I would like to thank my supervisor, prof.dr.ir. B. De Schutter, for his guidance during the thesis process. His insights and feedback were invaluable for the creation of this report. Furthermore, I would like to thank dr.ir. E. Abraham for our useful discussions and helping me see the real-world aspects of water management more clear.

I would also like to thank my family. My parents for supporting me morally and financially during my time in Delft, my brother for the insightful and fun discussions on any topic, and my girlfriend, Kristi, for the continuous support.

Lastly, I would like to thank my friends from DSKV IJzersterk, for the needed distractions during times of stress and the motivation to stay disciplined and work hard to achieve my goals.

Delft, June 22, 2018 R.C. Kassing

"L'acqua che tocchi de' fiumi è l'ultima di quella che andò e la prima di quella che viene. Cosí il tempo presente."

In rivers, the water that you touch is the last of what has passed and the first of that which comes; so with present time.

— Leonardo da Vinci

# Chapter 1

# Introduction

The importance of water for sustaining human life cannot be understated. It is rooted in every day life through drinking, recreation, transport, agriculture, and energy production. Water systems are available in the form of water bodies such as seas, lakes, and reservoirs which are connected to natural rivers and man-made canals. The flows of water can be manipulated through shaping rivers, building canals, and operation of structures, such as pumps and gates. Local control inputs include filling or draining of water reservoirs and actuation of pumps or locks. When controlling a large open water system all of these local water systems with their local control inputs need to be taken into account.

The importance of an efficient water management system is increasing world-wide, in particular due to higher sea levels, increase of rain during the spring season, and drier summers. About 70 percent of the freshwater supply is used for irrigation of lands that produce 40 percent of the world's food [26]. Too high water levels (floods, water spillage) and too low water levels (decreased crop yield, irrigation issues) should be avoided, while minimising the cost of the control actions. The evolution of the local water levels depends on what happens over a much larger region, which can even extend beyond nations. However, water is usually managed in a relatively small region by local water management organisations. The current lack of coordination of localised control results in suboptimal water delivery and loss of water. More efficient water management with less risks and costs can be obtained by coordination of the local water management actions, and by also including predictions of future rain fall, future droughts, and future arrival of water flows from other water systems. These predictions can be based on hydraulic models driven by weather predictions and data from a network of sensors. This coordination is complex, as some of the local requirements may sometimes be conflicting, which requires a multi-constraint and multi-objective control task to be solved. Thus, new intelligent, multi-agent model-based predictive control approaches for water management have to be developed. In this thesis, the control approaches for open water transport systems with human operators are investigated. These approaches will need to satisfy the basic requirements and service levels to perform adequate water management.

Another issue that these approaches will need to overcome is that they cannot rely on a fully automatic control operation, as it is unrealistic in less developed countries because humans are involved in the actuation process, in taking measurements, or in the system dynamics; in addition, fully automatic control operation is often too expensive. On the other hand, having fully manual control operation can compromise the performance by limitations of the operators to oversee the many interacting components of the water system. This gives incentive in providing a link between modern control methods and human-operated control systems. A few approaches [28, 37] have been proposed to answer the question of how to integrate humans into the control problem. However, how to prioritise the locations for the operators to take measurement, in order to keep track of the state of the open water system, is still an open question.

# 1-1 Open water systems

The term open water system covers a broad spectrum of large-scale water systems. Generally, they are characterised by their large size and consist of multiple interacting bodies of water. Furthermore, open water systems are subjected to various meteorological influences, which can have a significant impact due to the vast size of open water systems. Two interesting open water systems from a control perspective are irrigation canals and drainage systems, as actuators and sensors are available throughout the systems and the water demands change over time, while the system is disturbed by external influences. Both irrigation canals and drainage systems and the pumps.

Controlling irrigation canals is of vital importance as the irrigation sector claims about 70 percent of the withdrawals of freshwater worldwide [26]. Irrigation water uses are: regulating the salinity of the soil, improving the soil texture, and providing enough water for growing crops. Adequate irrigation can result in crop yields that are two to four times greater compared to rain-fed farming and it currently provides 40 percent of the worlds food from approximately 20 percent of all agricultural land [26].

The water from irrigation canals to be delivered has some specifications:

- Accurate delivery of water: to avoid spillage, the right amount of water should arrive at the right time;
- Flexible water delivery, as the water delivery requirements of the farmers change;
- Safe and robust management of the water, as floods are dangerous and inefficient;
- Low operational costs: managing and maintaining the delivery of water, which is important from an economical point of view.

To deliver the water to the users a water distribution network with several features is used. Control structures, such as gates and pumps, are used to control the water flows and thereby distribute the water over the canal network. There is a wide variety of control structures, such as sluices, pumps, and weirs. Most sluices and weirs can be categorised as some kind of overshot gate or undershot gate [29]. Furthermore, the offtake structures of the irrigation canal network are used for taking the water out of the canal, i.e., onto the land or into specific secondary canals. The canal sections located between two control or offtake structures are called reaches.

For the control structures and offtake structures to function properly they require an upstream water level that is sufficiently high. There are three ways to obtain a high enough water level:

- Adjusting the upstream flow rate,
- Backing up the water level by use of a downstream control structure,
- Using a combination of the two methods mentioned above.

Meeting these water requirements is not a trivial problem. The water arrives at the offtake structures with some delay, due to hydrodynamics. Furthermore, the structures cannot be set completely accurately, since the measurements have limited accuracy and the control structures are not operated continuously. This results in spillage of water and the supplied water not matching the demand. If the demand for water was static, this error could be corrected by simply integrating the mismatch. However, the water requirements are unpredictable due to environmental factors, such as rainfall, droughts, and the working schedules of the farmers. Therefore, more sophisticated control methods are needed to meet the water requirements.

# 1-2 Automation of open water systems

The development of automatic control for open water systems started with registering the water level using automatic water level sensors. Next, the measurements from multiple sensors became available at a central location, using communication lines to relay the measurements. Meanwhile, information on the disturbances such as rain forecasts have become available and increasingly more accurate. The structures used to guide the water such as pumps and gates were also automated by electric motors and automatic switching off and on of pumps. When a water system has advanced to this level of automation, it is capable of being controlled by a central computer. However, such a system depends heavily on communication lines, which are prone to failures. For this reason, local control techniques are still used today. In developing countries, labour is inexpensive, so structures are manually operated by humans since installing sensors and actuators is too costly.

# **1-3** Control of open water systems

Although various control techniques have been designed for open water systems, not all of them have been implemented in practice [14, 19, 32, 38]. Instead, they are tested on accurate hydrodynamical models to illustrate applicability. In [22], regulation methods for irrigation canals are classified based on considered variables, logic of control, design method, and field implementation. A classification based on control theory is also possible [35]:

• Feedforward control: the earliest control implementations were based on feedforward control, as accurate models were available that could be used for inverse modelling and

feedback control was not yet a technique known to water control engineers. However, the standalone feedforward controllers are not able to track setpoints to zero error, as the inverse model can never be perfectly representative of the water system and the measurements and expected disturbances have some errors.

- Feedback control: the introduction of feedback control led to the first successful implementations that were able to regulate water levels close to setpoints, see [22, 23]. In [30], a simple water level controller is proposed that consists of a proportional-integral-based controller for feedback control to correct for measured deviations from the setpoint, and a feedforward controller based on inversion of the dynamic model of the canal system that uses an estimate of the disturbance to counter the influence of the disturbance on the water level.
- Optimal control: optimal controllers minimise an objective function using an optimisation algorithm. In this objective function, weights are assigned to the square of the error in tracking the reference and to the square of the change in control input. The weights are chosen by their relative importance and are tuned for optimal performance of the system. Setting higher weights on the error in reference tracking will result in faster reference tracking at the cost of increased control input magnitude and frequency of updates. On the other hand, putting more weight on the change in control action results in the tracking error increasing, but the control actions are applied more smoothly and with less magnitude. See [44] for a field tested centralised linear-quadratic regulator.
- Heuristic control: in these methods the dynamical behaviour of the water system is seen as a black box. Examples of these methods are fuzzy-logic control [41], rule-based control [11], and neural networks [33]. Since the dynamical behaviour of open water systems has been researched extensively in the last decades, these methods miss out on system understanding and are not applied to a large extent.

## 1-3-1 Model-based control

To describe the relationship between the many actuators and sensors of the water system a model is required. The states of the model as well as the multiple inputs to the water system, through gates and pumps, are limited by physical constraints. Furthermore, the water system is also bound to socioeconomic developments, such as changing irrigation demands by farmers. In addition, the dynamics of open water systems typically involve long time lags. Hence, a more advanced control technique than classical feedforward and feedback is required. An advanced control technique capable of dealing with these requirements is Model Predictive Control (MPC). In MPC, the benefits of the control methods such as feedforward and feedback control can be used, while also including explicit constraints on the states and inputs of the system and explicit performance measures.

Over the last two decades, MPC has been successfully implemented in the process industry [5], power networks [39], and road traffic networks [8]. For water management, MPC has been shown to be superior to conventional local control techniques like PI control [38].

### 1-3-2 Human-operated control

Irrigation canals are used to provide water for food production, drinking, washing, and many more applications all over the world. Often, the canals are located in harsh environments that result in wear and tear and thus malfunctioning of operating equipment, which sometimes even gets stolen by passers-by. For this reason and the fact that equipment can be very expensive to buy and maintain, the managers of irrigation canals often resort to manual operation, where one or multiple human operators travel between the gates and adjust gates' positions based on subjective judgement. The human operator only takes into account local information about the gate position and water levels; so the resulting performance is far from the global optimum that could be achieved with central, automatic operation of the whole system. This motivates the design of control methods that integrate humans as sensors and actuators. One of the promising methods is Mobile Model Predictive Control (MoMPC) introduced in [20, 37]. MoMPC is a configuration of MPC that explicitly integrates a human operator in the control problem. The operator travels between the gates of an irrigation canal and communicates with the central controller using a mobile device.

# 1-4 Problem statement

The MoMPC approaches from literature [20, 28, 37] rely on operators to provide the controller with up-to-date measurements from the locations visited and to act as the actuators as required by the central controller. At every new location the operator visits, he/she communicates the new measurements to the control centre, where they are used by the centralised controller to solve the optimisation problem. The central controller returns the control action that is to be implemented at the current location and provides the operator with the next location to travel to and the desired time to reach the destination. In literature [28, 37], this next location is only based on the cost associated with implementing a control action. However, taking measurements should have some priority too, since not measuring at locations for a long time can introduce substantial error between the real and the predicted state. This may lead to suboptimal operator route choices and can cause instability in the system and constraint violations. Therefore, the problem statement for this master thesis is:

How to efficiently extend the mobile model predictive control scheme by introducing methods to prioritise measuring at certain locations of the open water system, in order to decrease the uncertainty about the systems' state and increase the closedloop reference tracking performance?

## **1-5** Thesis outline

This master thesis is organised as follows. In Chapter 2, the modelling of open water systems is discussed in detail and two MPC approaches from literature for controlling an irrigation canal with human operators are presented. In Chapter 3, a new control algorithm for open water systems with mobile operators is presented, called Multiple-Action Mobile Model Predictive Control (MaMoMPC). This new control algorithm allows the controller to decide on

what actions to complete at each location visited in the open water system. Furthermore, 'cyclic control' is proposed to improve the reference tracking performance of the algorithm. Moreover, three methods to weigh the measurement frequency at each location are introduced. After that, an energy recharging framework that ensures the human operators get adequate breaks is presented. In Chapter 4, the extensions proposed in the MaMoMPC algorithm from Chapter 3 are tested in a case study. Note that each of these chapters ends with a conclusion. Finally, in Chapter 5, the conclusions and recommendations are presented.

# Chapter 2

# Model Predictive Control of Open Water Systems

This chapter focusses on two main topics regarding previous work on control of open water systems. First, the modelling of open water systems is discussed in detail. Secondly, the Model Predictive Control (MPC) approaches to controlling an irrigation canal with mobile (human) operators are presented. The Mobile Model Predictive Control (MoMPC) method is a derivative of MPC; it is assumed that the reader is familiar with the basic concept of MPC. If the reader is not familiar, a short summary can be found in Appendix A. If more detailed information about MPC is needed; it is recommended to read [4, 5]. At the end of the chapter, a conclusion is presented about the modelling of open water systems and the mobile operator MPC approaches to controlling them.

# 2-1 Modelling open water systems

The management objective of MPC for open water systems is to keep the water levels as close to the setpoints as possible. To that end, a suitable model of the relevant processes of the water system is needed. The models need to be set up to contain the most relevant dynamics of the water system for this regulating of the water levels. The most relevant processes are the behaviour of the water movement in the open channel: maintaining certain water levels at various locations and the water flows that influence these water levels. Control structures are used to manipulate the water flows, through which the controller can achieve the management objective. Achieving this objective is not straightforward, as there are varying inflows and outflows that disturb the water system. By modelling all these parts of the water system (canal reaches, structures, disturbances, controller) a model predictive controller can predict the future water levels and flows that are the result of the disturbances and the control actions.

In Section 2-1-1, a simple approach to modelling a canal reach is presented. Next, in Section 2-1-2, the models of control structures that are commonly used in open water systems are presented. In Section 2-1-3, a short introduction to the disturbances acting on the open

water system is given. Then, in Section 2-1-4, two approaches to incorporating the control structures in the model of an irrigation canal are presented, respectively a flow control and a gate control approach. Finally, the shortcomings and advantages of these two modelling approaches are discussed.

## 2-1-1 Canal reaches

The flow of water in irrigation canals is driven by gravity. Irrigation canals are cascadeconnected networks with reaches separated by movable gates to transport the water downstream to the farmers from a reservoir or pumping station. In Figure 2.1, a schematic drawing of an irrigation system is depicted.



Figure 2.1: Irrigation system built up out of reaches. The blue arrows indicate offtake flows.

The dynamic behaviour of water flow can be accurately described by the nonlinear De Saint-Venant partial differential equations [6, 18]. As the discretised De Saint-Venant equations have a large computational burden, they are difficult to use in real-time applications. Instead, simplified models are used. One of these simplified models is the Integrator Delay (ID) model, proposed in [29]. The ID model is based on observations of the dynamics of water movements: in reaches that are completely affected by backwater (water collecting due to a downstream structure) and are short enough, resonance can occur. Moreover, the resulting resonance waves hardly deform when travelling through the reach. On the other hand, when reaches are long and steep, resonance waves quickly dampen out and the flow of water is not influenced much by waves bouncing off the structure downstream; therefore, it can be assumed that waves travel downstream only. Based on these insights in the dynamics of open water a simple model is formulated: an integrator for the backwater part of the reaches and a time delay for the wave of water to travel from the upstream part of the reach to the downstream structure. In Figure 2.2, the physical interpretation of the characteristics of the ID model is depicted. The model can be described by a discrete-time difference equation for the downstream water level:

$$h(k+1) = h(k) + \frac{T_{\rm s}}{A_{\rm s}}Q_{\rm in}(k - k_{\rm delay}) - \frac{T_{\rm s}}{A_{\rm s}}Q_{\rm out}(k) + \frac{T_{\rm s}}{A_{\rm s}}Q_{\rm d}(k),$$
(2.1)

where h(k) is the current water level,  $A_s$  is the storage area of the reservoir,  $T_s$  is the sampling time of the model,  $Q_{in}(k - k_{delay})$  is the delayed flow to the reservoir,  $Q_{out}(k)$  is the outflow

from the reservoir, and  $Q_{\rm d}(k)$  is the disturbance inflow to the reservoir due to, e.g., offtakes or rainfall.



Figure 2.2: Integrator delay model of a single canal reach

**Remark.** The time between the water level changing at the upstream part of the reach and the resulting water level change at the downstream of the reach is defined as the delay time. For a model predictive controller to take into consideration the full effect of the control actions these delays need to be accounted for. The prediction horizon is usually set as the summed delay times of all reaches, so that the effect of a control action upstream in the canal on the water level downstream can be predicted and accounted for.

The ID model captures the basic dynamics, such as the delay time and the basic frequency. The most beneficial quality of the model is that it is simple; it scales linearly with the number of reaches. Because of its low computational burden and linearity, the ID model is often used as internal model in applications of MPC [25, 34]. However, the ID model describes the low frequency behaviour accurately, but does not include all the resonance modes. Thus, for long, steep, and shallow irrigation canals the ID model is representative, especially since for these types of canals the flow conditions, i.e., the discharge rate of the canal, do not have a large influence on the storage area and flow transport times. However, the fraction  $\frac{T_s}{A_s}$  and flow delay time steps  $k_{\text{delay}}$  from (2.1) are included in the ID model as constants. Consequently, a disadvantage is that the model cannot account for the parameter changes when the flow conditions change significantly. Therefore, the predefined delay time and storage area are only valid when the canal is operating at certain percentages of the maximum discharge rate. If the model predictive controller does not take into account the relevant dynamics of the real water system, the closed-loop control of the open water system can become unstable. In particular, in the ID model the resonance frequency is not included, which causes a discrepancy between the internal model and the open water system for certain excitations of the system. As the canals used in this thesis have steep and long reaches and are operated at high discharge rates none of these methods are required. However, there are several ways to resolve the issues of resonance waves playing a dominant role: low-pass filtering [29], time-variant linear models [35], and higher order models [17, 36, 38].

## 2-1-2 Control structures

The structures need to be modelled as part of the entire water system for them to be included in the model predictive controllers. The behaviour of the control structures can be described by analytical discharge relations based on the Bernoulli equation describing the conservation of energy.

**Remark.** For controllers with a feedback part, such as PI controllers and model predictive controllers, modelling errors of the structures do not pose big (instability) problems. This is because the controller will continue to adjust the gates, until the objective is achieved. However, as actuation may be sparse in time for the case of humans operating the gates, the modelling errors of the structures might pose bigger problems.

### **Overshot** gates

An overshot gate is a control structure that backs up water, by forcing water to flow over the crest of the gate, see Figure 2.3. The energy at the upstream side of the structure is the sum of two components: the potential energy of the water level above the crest and the kinetic energy in the velocity of the flow. The kinetic energy at the upstream side is often neglected, as the velocity is low, due to the large wetted area on the upstream side and measuring the flow velocity in canal reaches is generally not done.



Figure 2.3: Overshot gate

The flow formula for the overshot gate is expressed as [35]:

$$Q(k) = \frac{2}{3}c_{\rm w} \cdot W_{\rm g} \sqrt{\frac{2}{3}g} \cdot (h_1(k) - h_{\rm cr}(k))^{\frac{3}{2}}, \qquad (2.2)$$

where Q represents the flow (m<sup>3</sup>/s) over the crest of the structure,  $c_{\rm w}$  is a lateral contraction coefficient that describes the way the stream lines of the flow change in the interaction with the structure,  $W_{\rm g}$  is the width (m) of the gate,  $h_1$  is the upstream water level (m),  $h_{\rm cr}$  is the crest height (m), and k is the time step index.

The flow can be increased or decreased by changing the crest level  $h_{\rm cr}$ ; the increase or decrease in flow will then scale with a power 3/2. There is also a natural feedback loop in place: if  $h_{\rm cr}$ is constant and the upstream water level  $h_1$  increases, the flow over the crest will increase more than proportionally, decreasing the upstream water level.

### **Undershot** gates

In an undershot gate, as the name suggests, the water has to flow under the gate; which is lowered into the water. There are two types of undershot gates: free-flowing and submerged gates, see Figure 2.4.

### R.C. Kassing



Figure 2.4: Free-flowing (left) and submerged (right) undershot gate

The formula for a free-flowing undershot gate is given by [35]:

$$Q(k) = c_{\rm w} \cdot W_{\rm g} \cdot \mu_{\rm g} \cdot h_{\rm g}(k) \sqrt{2g \cdot (h_1(k) - (h_{\rm cr} + \mu_{\rm g} \cdot h_{\rm g}(k))))}, \qquad (2.3)$$

where  $\mu_{\rm g}$  is the contraction coefficient,  $h_{\rm cr}$  is the crest level,  $h_{\rm g}$  is the gate opening from the crest level to the bottom of the gate,  $h_1$  is the upstream water level, and  $h_2$  is the downstream water level.

An advantage of undershot gates is that they are well suited to control the downstream water level precisely, as the flow increases less than proportionally with the gate opening. However, from this scaling factor a disadvantage follows as well: the gate is not well suited to control the upstream water level.

### 2-1-3 Disturbances

Open water systems are significantly influenced by disturbances, due to their vast size and continuous exposure to uncertain meteorological forces. Disturbances can cause instability in the system; so having an accurate disturbance model to counter them is important. Some examples of meteorological disturbances acting on open water systems are rainfall and drought, while other disturbances are the water withdrawal by the farmers also known as offtakes, and the inaccuracy of the human operators implementing the control actions. For some of these disturbances a prediction can be available, e.g., if the people in charge of the open water system have agreements with the farmers for the timing and amounts of water withdrawals from the canal. However, there is some uncertainty on the exact timing and amount of the water withdrawal. Moreover, rainfall predictions can be included. However, these rainfall predictions are only sufficiently reliable for the short term (about a few hours to a day). Nonetheless, by including a prediction of the offtake disturbance and rainfall the system performance can be improved.

### 2-1-4 Model of an open water system

### State space notation for flow control

To be able to directly minimise water level errors in the MPC cost function the water level state from (2.1) is rewritten to a water level error state:

$$e(k+1) = e(k) + \frac{T_{\rm s}}{A_{\rm s}}Q_{\rm in}(k) - \frac{T_{\rm s}}{A_{\rm s}}Q_{\rm out}(k) - \frac{T_{\rm s}}{A_{\rm s}}Q_{\rm d}(k), \qquad (2.4)$$

Master of Science Thesis

R.C. Kassing

where

$$e(k) = h_{\rm SP} - h(k),$$
 (2.5)

where  $h_{\rm SP}$  denotes the water level setpoint.

The actuation of the canals will be sparse in time, due to the required presence of a mobile operator to operate a gate. Therefore, controlling the increment of control input is preferable; in order to keep the gate flow constant in between actuation instants. By introducing the change in error as an additional state, the input can be written in terms of flow increments [35]:

$$\Delta e(k+1) = e(k+1) - e(k).$$
(2.6)

Inserting (2.4) into (2.6) yields:

$$\Delta e(k+1) = e(k) - e(k-1) + \frac{T_{\rm s}}{A_{\rm s}}(Q_{\rm in}(k) - Q_{\rm in}(k-1))$$
(2.7)

$$-\frac{T_{\rm s}}{A_{\rm s}}(Q_{\rm out}(k) - Q_{\rm out}(k-1)) - \frac{T_{\rm s}}{A_{\rm s}}(Q_{\rm d}(k) - Q_{\rm d}(k-1)).$$

Define:

$$\Delta e(k) = e(k) - e(k-1),$$
(2.8)

$$\Delta Q_{\rm in}(k) = Q_{\rm in}(k) - Q_{\rm in}(k-1), \qquad (2.9)$$

$$\Delta Q_{\text{out}}(k) = Q_{\text{out}}(k) - Q_{\text{out}}(k-1), \qquad (2.10)$$

$$\Delta Q_{\rm d}(k) = Q_{\rm d}(k) - Q_{\rm d}(k-1). \tag{2.11}$$

Finally, (2.4) and (2.7) can be rewritten in terms of flow increments using (2.8)–(2.11) to obtain the state space description for a single canal reach:

$$\begin{bmatrix} e(k+1) \\ \Delta e(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e(k) \\ \Delta e(k) \end{bmatrix} + \begin{bmatrix} \frac{T_{\rm s}}{A_{\rm s}} \\ \frac{T_{\rm s}}{A_{\rm s}} \end{bmatrix} \Delta Q_{\rm in}(k) + \begin{bmatrix} -\frac{T_{\rm s}}{A_{\rm s}} \\ -\frac{T_{\rm s}}{A_{\rm s}} \end{bmatrix} \Delta Q_{\rm out}(k) + \begin{bmatrix} -\frac{T_{\rm s}}{A_{\rm s}} \\ -\frac{T_{\rm s}}{A_{\rm s}} \end{bmatrix} \Delta Q_{\rm d}(k).$$

$$(2.12)$$

To describe an open water system, several of these canal reaches will need to be connected. The water flowing from one canal reach to another will have some transport delay time. That can be included in the system description by introducing water flow transport states. Consider an example irrigation canal with two reaches, as depicted in Figure 2.5.

The flow transport delay times from the head gate to the first reach and from the second gate to the second reach are assumed to be, respectively, 800 and 150 seconds. Next, the model of the system is discretised with a time step of  $T_{\rm s} = 300$  seconds; the transport delays are rounded up to, respectively, 3 and 1 time steps.

Next, this example irrigation system can be described by the following state space description:

$$\mathbf{x}_{\text{ex,flow}}(k+1) = \mathbf{A}_{\text{ex,flow}}\mathbf{x}_{\text{ex,flow}}(k) + \mathbf{B}_{\mathbf{u},\text{ex,flow}}\mathbf{u}_{\text{ex,flow}}(k) + \mathbf{B}_{\mathbf{d},\text{ex,flow}}\mathbf{d}_{\text{ex,flow}}(k), \qquad (2.13)$$
$$\mathbf{y}_{\text{ex,flow}}(k) = \mathbf{C}_{\text{ex,flow}}(k), \qquad (2.14)$$

R.C. Kassing

Master of Science Thesis



**Figure 2.5:** An example irrigation canal consisting of two reaches with an upstream reservoir providing water to the first reach.

where

where the 'measured' states are the water level errors. Note that the water levels are actually measured and are subsequently subtracted from water level setpoint to retrieve the water level error.

Master of Science Thesis

R.C. Kassing

#### State space notation for gate control

The flow-controlled system (2.12) from the previous section assumes that the control flow through a gate stays constant after actuation of the gate. This assumption only holds if a local flow controller is available at the gate; if not, then the control flow is dependent on the upstream water level and the gate opening, as described by (2.3) for an undershot gate.

To include this nonlinear behaviour in the internal model, the undershot gate flow equation (2.3) needs to be linearised; this can be achieved by applying a first-order Taylor expansion [35, 42]. The first-order Taylor series expansion around the last known values of the states,  $h_{\rm g}(k)$  and  $h_1(k)$ , is applied to (2.3). The resulting linear equation for undershot gate free-flow is:

$$Q(h_1 + \Delta h_1, h_g + \Delta h_g) = Q(h_1, h_g) + \Delta h_1 \frac{\partial Q}{\partial h_1}(h_1, h_g) + \Delta h_g \frac{\partial Q}{\partial h_g}(h_1, h_g), \qquad (2.15)$$

where

$$\frac{\partial Q}{\partial h_1} = \frac{g \cdot c_{\mathbf{w}} \cdot W_{\mathbf{g}} \cdot \mu_{\mathbf{g}} \cdot h_{\mathbf{g}}(k)}{\sqrt{2g \cdot (h_1(k) - (h_{\mathbf{cr}} + \mu_{\mathbf{g}} \cdot h_{\mathbf{g}}(k)))}},\tag{2.16}$$

$$\frac{\partial Q}{\partial h_{g}} = c_{w} \cdot W_{g} \cdot \mu_{g} \sqrt{2g \cdot (h_{1}(k) - (h_{cr} + \mu_{g} \cdot h_{g}(k)))} - \frac{g \cdot c_{w} \cdot W_{g} \cdot \mu_{g}^{2}(h_{g}(k) - h_{cr})}{\sqrt{2g \cdot (h_{1}(k) - (h_{cr} + \mu_{g} \cdot h_{g}(k)))}},$$

$$(2.17)$$

where the partial derivatives are coefficients that can be obtained from the last closed-loop result. Next, by combining (2.4), (2.8), (2.10), and (2.15), the following equation can be derived:

$$\Delta Q(k) = Q(k) - Q(k-1) = C_{\rm e} \cdot \Delta e(k) + C_{\rm u} \cdot \Delta u(k), \qquad (2.18)$$

where for ease of notation the following definitions are used:

$$C_{\rm e}(k) = \frac{\partial Q(k)}{\partial h_1(k)},\tag{2.19}$$

$$C_{\rm u}(k) = \frac{\partial Q(k)}{\partial h_{\rm g}(k)}.$$
(2.20)

Finally, by expressing the control action in terms of the change in gate opening  $\Delta u$ , a state space model with linearised gate equations for a single canal reach is obtained:

$$\begin{bmatrix} e(k+1) \\ \Delta e(k+1) \end{bmatrix} = \begin{bmatrix} 1 & (1-C_{e} \cdot \frac{T_{s}}{A_{s}}) \\ 0 & (1-C_{e} \cdot \frac{T_{s}}{A_{s}}) \end{bmatrix} \begin{bmatrix} e(k) \\ \Delta e(k) \end{bmatrix} + \begin{bmatrix} \frac{T_{s}}{A_{s}} \\ \frac{T_{s}}{A_{s}} \end{bmatrix} \Delta Q_{in}(k)$$
$$+ \begin{bmatrix} -C_{u} \cdot \frac{T_{s}}{A_{s}} \\ -C_{u} \cdot \frac{T_{s}}{A_{s}} \end{bmatrix} \Delta u(k) + \begin{bmatrix} -\frac{T_{s}}{A_{s}} \\ -\frac{T_{s}}{A_{s}} \end{bmatrix} \Delta Q_{d}(k).$$
(2.21)

Now, the example irrigation system from Figure 2.5 can be described by the following gate-controlled state space model:

$$\mathbf{x}_{\text{ex,gate}}(k+1) = \mathbf{A}_{\text{ex,gate}}\mathbf{x}_{\text{ex,gate}}(k) + \mathbf{B}_{\mathbf{u},\text{ex,gate}}\mathbf{u}_{\text{ex,gate}}(k) + \mathbf{B}_{\mathbf{d},\text{ex,gate}}\mathbf{d}_{\text{ex,gate}}(k), \quad (2.22)$$
$$\mathbf{y}_{\text{ex,gate}}(k) = \mathbf{C}_{\text{ex,gate}}(k), \quad (2.23)$$

R.C. Kassing

Master of Science Thesis

where

where the 'measured' states are the water level errors. Note that the water levels are actually measured and are subsequently subtracted from water level setpoint to get the water level error.

#### Comparing gate control to flow control

For some rural irrigation canals there might not be a flow controller present at the gates, due to the cost and trouble of maintenance. As a result, these gates are likely operated by human operators. Therefore, it may be beneficial to control the gate opening instead of the gate flow, due to being able to counter the disturbances more accurately by better anticipation of the effect of the control action [42]. Moreover, for these types of gates without flow controller the human operators will need to be instructed what gate settings to implement. Furthermore, by including gate control in the internal model it is possible to put constraints on the gate openings and the change of the gate openings [16].

In order to compare the difference in water level prediction between the gate-controlled and flow-controlled internal models, a real irrigation canal is modelled: the west-main irrigation canal located in Arizona, United States of America. The Arizona west-main canal consists of eight canal reaches interconnected by seven undershot gates and the water is supplied by a head gate. Furthermore, the canal has a total length of 10 km and a maximum discharge capacity of 2.8 m<sup>3</sup>/s at the head gate. The rest of the parameters of the canal can be found in Appendix B-2. The canal is discretised with a sampling time step of 240 seconds. Furthermore, none of the gates (other than the head gate) are equipped with local flow controllers. An artificial flow control signal is created and converted into gate settings for use in the gate-controlled model, see Figure 2.6.



**Figure 2.6:** The control signals in terms of flow and gate opening for the Arizona west-main canal. Note that the head gate will be flow-controlled in the gate-controlled internal model.

Next, the undershot gate equation (2.3) is linearised for each gate at the start of the simulation to insert into the gate-controlled model. Then, the trajectories of the water level errors are computed using open-loop simulations of the flow-controlled model, gate-controlled model, and nonlinear gate-controlled model. Note that neither disturbances nor noise are added to the models. The resulting water level error predictions are depicted in Figure 2.7.

The linearised gate-controlled model accurately approximates the nonlinear model for the first half of the prediction, in the second half the water level predictions start to diverge, see Figure 2.7. This diverging of the prediction of the linearised gate-controlled model can be completely attributed to the accumulation of the linearisation error. The effect of water level on the gate flow is not included in the flow-controlled model. Therefore, the flow-controlled model wrongfully predicts that the water levels are at rest when the gates are not actuated.

The modelling of the real system will not be perfect. In order to evaluate the effect of the model imperfections, some white Gaussian noise with a standard deviation of  $0.001 \text{ m}^3/\text{s}$  is added as external disturbance to the nonlinear model, see Figure 2.8. As a result, the water level predictions of the linearised gate-controlled model diverge faster from that of the nonlinear gate-controlled model than in Figure 2.7. This emphasises the importance of accurately capturing the disturbance dynamics acting on the system and measuring frequently.

Note that for reach 8, the nonlinear prediction diverges from the linearised prediction after just a few time steps, see Figure 2.8. This is troublesome, as there is no gate present at



**Figure 2.7:** The predicted water level trajectories of the flow-controlled model, linearised gatecontrolled model, and nonlinear gate-controlled model when the Arizona west-main canal is subjected to the control signal from Figure 2.6.

reach 8 to directly control the water level. The only way to reduce the water level error in the last reach is by adjusting the flow of the gate at reach 7, which will have some delay before arriving at reach 8. Moreover, a persistent mismatch in flow and water level error is to be expected as this is the last reach: the excess water of the upstream reaches, caused by the linearisation errors, measurement errors, and process noise, will collect here. If this mismatch is relatively large and forms quickly, it will likely destabilise the model predictive controller. Therefore, if the last reach does not have a gate its water level errors are not taken into account in the cost of the optimal control problem, for the purpose of this thesis.

#### Gate and flow constraints

In the next few paragraphs, we investigate the lower and upper bound on the gate opening and gate flow that are required to stay within the physical limitations of operating the irrigation canal, as well as the limitations of the applicability of the nonlinear flow equation (2.3).

### Upper bound on the gate openings

In order to determine the upper bound on the gate opening, the effect of the water level and the gate opening on the gate flow is examined. The second gate of the Dez main canal is used for illustrative purposes. Accordingly, for a small set of 10 upstream water levels ranging from no water in the reservoir to a water level above the setpoint (4.63 m), the gate opening



**Figure 2.8:** The predicted water level trajectories of the flow-controlled model, linearised gate-controlled model, and nonlinear gate-controlled model when the Arizona west-main canal is subjected to the control signal from Figure 2.6, along with white Gaussian noise with a standard deviation of 0.001  $\text{m}^3$ /s on the disturbance.

is varied from fully closed to fully opened and the resulting flows are plotted in Figure 2.9. The gate parameters can be found in Table B-3.

When opening the gate above the water level, the square root term in (2.3) can become negative and the real part of the flow goes to zero. Therefore, it is unacceptable to allow the controller to open the gate more than  $\frac{h_1(k)-h_{\rm cr}}{\mu_{\rm g}}$ . Furthermore, the flow decreases after opening the gate a certain amount and to achieve each flow there are two possible gate openings. This decrease in flow after exceeding a certain gate opening does not occur in the real setting, as the water freely flows beneath the gate to the next reach, when the gate is lifted out of the water. Moreover, when linearising the gate flow in the negative slope part of the curve the controller will wrongfully assume the gate flow increases by decreasing the gate opening. For these reasons, it is investigated at what gate opening the maximum flow occurs when the upstream reservoir water level is constant; in order to constraint the gate opening to this maximum.

Consider the free-flowing undershot gate equation (2.3). The goal is to find the gate opening that corresponds to the maximum flow of the function for the water level  $h_1(k)$  at time step k. First, the gate crest level,  $h_{cr}$ , is subtracted from the relative water level  $h_1(k)$  to get the



**Figure 2.9:** Flow as a function of the gate opening for the second gate of the Dez main canal for 10 different water levels.

effective water level of the upstream reservoir:

$$h_{\rm eff}(k) = h_1(k) - h_{\rm cr}.$$
 (2.24)

The water level  $h_{\text{eff}}(k)$  is assumed a constant, that is retrieved from the last closed-loop result or measurement. To find the maxima of (2.3), the derivative with respect to the gate opening  $h_{\text{g}}(k)$  needs to be calculated. This is already accomplished in the gate linearisation part of Section 2-1-4. Next, inserting (2.24) into the partial derivative from (2.17) and setting equal to zero:

$$c_{\rm w} \cdot W_{\rm g} \cdot \mu_{\rm g} \sqrt{2g \cdot (h_{\rm eff}(k) - \mu_{\rm g} \cdot h_{\rm g}(k))} = \frac{g \cdot c_{\rm w} \cdot W_{\rm g} \cdot \mu_{\rm g}^2 \cdot h_{\rm g}(k)}{\sqrt{2g \cdot (h_{\rm eff}(k) - \mu_{\rm g} \cdot h_{\rm g}(k)))}}.$$
 (2.25)

Then, simplifying with the knowledge that  $c_{\rm w}$ ,  $W_{\rm g}$ ,  $\mu_{\rm g}$ , g, and  $h_{\rm eff}(k)$  are positive constants:

$$c_{\rm w} \cdot W_{\rm g} \cdot \mu_{\rm g} \cdot (2g \cdot (h_{\rm eff}(k) - \mu_{\rm g} \cdot h_{\rm g}(k))) = g \cdot c_{\rm w} \cdot W_{\rm g} \cdot \mu_{\rm g}^2 \cdot h_{\rm g}(k),$$
  

$$2 \cdot (h_{\rm eff}(k) - \mu_{\rm g} \cdot h_{\rm g}(k)) = \mu_{\rm g} \cdot h_{\rm g}(k),$$
  

$$h_{\rm g}(k) = \frac{2}{3} \cdot \frac{h_{\rm eff}(k)}{\mu_{\rm g}}.$$
(2.26)

Now, a second derivative test is needed in order to verify that the points on the line from (2.26) are maxima. The second derivative of (2.3) is:

$$\frac{\partial^2 Q}{\partial^2 h_{\rm g}} = -\frac{\sqrt{2}}{4} \cdot \frac{W_{\rm g} \cdot c_{\rm w} \cdot g^2 \cdot \mu_{\rm g}^2 \cdot (4h_{\rm eff}(k) - 3h_{\rm g}(k) \cdot \mu_{\rm g})}{(g \cdot (h_{\rm eff}(k) - h_{\rm g}(k) \cdot \mu_{\rm g}))^{\frac{3}{2}}}.$$
(2.27)

Finally, inserting (2.26) into (2.27):

$$\frac{\partial^2 Q}{\partial^2 h_{\rm g}} \left( \frac{\partial Q}{\partial h_{\rm g}} = 0 \right) = -\frac{3 \cdot \sqrt{3}}{\sqrt{2}} \cdot \frac{W_{\rm g} \cdot c_{\rm w} \cdot g^2 \cdot \mu_{\rm g}^2 \cdot h_{\rm eff}(k)}{(g \cdot h_{\rm eff}(k))^{\frac{3}{2}}}.$$
(2.28)

Master of Science Thesis

R.C. Kassing

The second partial derivative from (2.28) is always negative, since a fraction of polynomials with positive constants is multiplied with a negative constant. Therefore, all the points on the line from (2.26) are maxima of (2.3), when the water level is assumed a positive constant.

Knowing the second partial derivative is always negative, it can be concluded that the constraint on the maximum height of each individual gate depends on the upstream reservoir water level, the gate crest level, the contraction coefficient, and maximum opening height of the gate as follows:

$$h_{\rm g,max,constraint}(k) = \min(\frac{2}{3} \cdot \frac{h_1(k) - h_{\rm cr}}{\mu_{\rm g}}, h_{\rm g,max}), \qquad (2.29)$$

where  $h_{g,max}$  is defined as the physical limitation on the gate opening. These constraints on the gate opening can be converted to flow constraints for use in flow-controlled MPC by converting (2.29) to flow using (2.3).

Note that as there may be some discrepancy between the water levels the internal model predicts and actual water levels, the upper bound should be lowered to be conservative. Furthermore, the linearisation of the flow curve is a poor approximation when linearising around the maximum flow region of the curve.

As the constraint in (2.29) depends on the water level in the upstream reservoir of the gate, it is time-variant. If it is undesirable to include time-variant constraints, due to limitations on computational resources, it is also possible to set a time-invariant maximum gate opening constraint for the whole prediction based on information obtained from the last closed-loop result or measurement. Another option is to use the previous open-loop trajectory to create a prediction of the time-variant constraints and then extrapolate the last known value of the predicted constraints until the prediction horizon [35].

#### Lower bound on gate flow

When a gate is completely closed, the gate flow is zero, see Figure 2.9. However, if the water level becomes higher than the crest of the weirs besides the gate, see Figure 2.10, then the weirs plus the gate act as an overshot gate and there is some overshot flow.



Figure 2.10: Front view of an undershot gate and the surrounding weirs in an irrigation canal.

For flow-controlled MPC to account for this, a minimum gate flow constraint can be added. However, in the case of gate-controlled MPC this is not possible, as the input is in terms of change of gate opening. A possible solution is to add an extra gate overflow variable to the description of each reach with a gate, and to set this variable equal to the amount of expected water overflow or to add the expected overflow to the scheduled disturbance of the downstream reach. Another solution is to set the minimum gate opening such that it matches the expected overflow. However, that comes with an important issue, as the linearised gate equations are dependent on the upstream water level, which will change gate flow over the prediction as the upstream water level changes. For the scope of this thesis, the overflow constraint of the gates are not included as the irrigation canals will not run close to maximum flow capacity in the experiments.

#### Upper bound on change in gate opening

The flow through a gate depends non-linearly on the gate opening and the water level, see (2.3) and Figure 2.9. Before changing gates' settings, the water level is first measured by the operator. Therefore, the linearisation error of the expected gate flow caused by the discrepancy between expected water level and actual water level of the upstream reservoir can be neglected. However, if the gate opening is changed by a significant amount the linearisation will be a poor approximation of the resulting gate flow. This becomes apparent when plotting the nonlinear flow as a function of the gate opening alongside the linear approximation, see Figure 2.11.



**Figure 2.11:** In the left subplot, the nonlinear flow and linearised flow as a function of the change in gate opening are plotted when the water level is at the setpoint and the gate is linearised around the opening of 0.5m. In the right subplot, the corresponding flow error as a function of the change in gate opening is plotted. Both of these plots involve the third gate of the Arizona west-main canal.

The approximation error of the flow will cause an offset on the expected water level error of the gate the operator is visiting and the flow towards the next gate. To limit these errors, the change in gate opening needs to be restricted. A possible method for restricting the change in gate opening is to find the maximum change in gate opening (in both closing,  $\Delta u_{\text{close,constraint}}$ , and opening directions,  $\Delta u_{\text{open,constraint}}$ ) for each gate that respect some predefined linearisation error boundary. For example, when setting the maximum flow error to 20% the resulting maximum change in opening can be found by drawing a horizontal line at height 20% flow error and finding the intersections with the flow error curve, see Figure 2.12.



**Figure 2.12:** For the third gate of the Arizona west-main irrigation canal the lower and upper bound of the change in gate opening that still respect the 20% flow error tolerance are, respectively,  $\Delta u_{\text{close,constraint}} = -0.2888 \text{ m}$  and  $\Delta u_{\text{open,constraint}} = 0.4255 \text{ m}$ .

In Figure 2.12,  $\Delta u_{\text{close,constraint}}$  and  $\Delta u_{\text{open,constraint}}$  are found by solving:

$$\epsilon_{\rm flow} = \frac{C_{\rm u} \cdot \Delta u_{\rm constraint} + Q_0 - Q_{\rm nonlinear}}{Q_{\rm nonlinear}}, \qquad (2.30)$$

s.t. 
$$\Delta u_{\text{constraint}} \ge -h_{\text{g}}(k),$$
 (2.31)

$$\Delta u_{\text{constraint}} \le h_{\text{g,max,constraint}} - h_{\text{g}}(k), \qquad (2.32)$$

where

$$Q_{0} = c_{\rm w} \cdot W_{\rm g} \cdot \mu_{\rm g} \cdot h_{\rm g}(k) \sqrt{2g \cdot (h_{1}(k) - (h_{\rm cr} + \mu_{\rm g} \cdot h_{\rm g}(k)))}, \qquad (2.33)$$

$$Q_{\text{nonlinear}} = c_{\text{w}} \cdot W_{\text{g}} \cdot \mu_{\text{g}} \cdot h_{\text{g},\Delta u}(k) \sqrt{2g \cdot (h_1(k) - (h_{\text{cr}} + \mu_{\text{g}} \cdot h_{\text{g},\Delta u}(k)))}, \qquad (2.34)$$

$$h_{g,\Delta u}(k) = h_g(k) + \Delta u_{\text{constraint}}, \qquad (2.35)$$

where  $\epsilon_{\text{flow}} > 0$  is the flow error tolerance (%) and  $Q_0$  represents the flow of the gate without actuation. Furthermore, solving (2.30)–(2.32) for  $\Delta u_{\text{constraint}}$  will return two unique solutions: a negative and a positive one. The negative and positive solutions correspond, respectively, to the variables  $\Delta u_{\text{close,constraint}}$  and  $\Delta u_{\text{opening,constraint}}$ . Note that (2.30)–(2.32) will always have two unique solutions, as the horizontal  $\epsilon_{\text{flow}}$  line intersects twice with the flow error parabola when  $\epsilon_{\text{flow}} > 0$ .

Next, the change in gate opening of each gate can be limited to the calculated lower and upper bound for the entire prediction window. Moreover, these bounds can be recalculated before every MPC iteration using the predicted or measured states of the current time step. Note that if a gate is actuated more than once in the prediction, the total change in gate opening of a gate might exceed the allowable linearisation error  $\epsilon_{\text{flow}}$ . Nonetheless, as the gate equation is linearised with every measurement and the model predictive controller only implements the first control time step of every MPC iteration, the effect on the performance will likely not be significant. However, if the occurrence of this linearisation error does have
a significant negative effect on the system performance, a possible solution is to constrain the sum of changes in gate opening for each gate over the duration of the prediction window. The investigation of this effect on the performance of the system is not part of the scope of this thesis.

# 2-2 Mobile MPC

In [20, 37], a novel human-in-the-loop method, MoMPC, is proposed; named after the mobility of the human operator in the control setting. The authors aim to find a middle ground between state-of-the-art control methods and human operations. The control system requires regular actions from the human operator. The human operator has no decision in these actions: the control system relies on the operator to implement the control actions and to perform measurements. It is assumed that there are only a few human operators available relative to the number of subsystems, which implies that both the sensing and actuating processes have a sparse nature. This will result in reduced performance compared to fully automatic methods. Controlling the system without the benefits of automated sensors and actuators that provide regular information and control actions is an interesting challenge for the new human-involved controller design.

## 2-2-1 Mobile MPC for irrigation canals

In MoMPC, an operator is employed to travel a sequence of gates to take measurements and to implement control actions. When an operator arrives at a location, the operator measures the water level and communicates this to the central controller using a mobile device, which triggers the central controller. The central controller then uses this measurement to update the corresponding states of the internal model and computes an optimal schedule of gates to visit and corresponding control actions to implement and communicates this back to the mobile device of the operator. In Figure 2.13, the control configuration is depicted.

#### 2-2-1-1 Configuration of Mobile MPC

In a standard centralised model predictive controller, the manipulated variables are optimised until the control horizon,  $N_c$ , to form an optimal control signal. For example, an irrigation system with three gates with, respectively, the control actions  $u_1(k)$ ,  $u_2(k)$ , and  $u_3(k)$  has the optimal control signal:

$$\mathbf{u}^{*}(k:k+N_{c}) = (u_{1}^{*}(k), u_{2}^{*}(k), u_{3}^{*}(k), u_{1}^{*}(k+1), u_{2}^{*}(k+1), u_{3}^{*}(k+1), \dots, u_{1}^{*}(k+N_{c}), u_{2}^{*}(k+N_{c}), u_{3}^{*}(k+N_{c}))^{T}.$$
(2.36)

In MoMPC, the operator travels along some optimal route and only when an operator is present at a gate and actuating, the corresponding control input of that gate in the control sequence is non-zero. For example, if there is only one operator, the travel time between each gate is one time step, and the optimal route turns out to be  $3 \rightarrow 2 \rightarrow 1$ , then the output of MoMPC would be:

$$\mathbf{u}^{*}(k:k+N_{c}) = (0,0,u_{3}^{*}(k),\underbrace{0,0,0}_{\text{travel time}},0,u_{2}^{*}(k+2),0,\underbrace{0,0,0}_{\text{travel time}},u_{1}^{*}(k+4),0,0)^{T}.$$
(2.37)

Master of Science Thesis

R.C. Kassing



Figure 2.13: Configuration of MoMPC for an irrigation canal

Hence, the non-zero elements of the control sequence determine both the route the operator must follow and the actions to implement along the route. The times at which an operator can be present at the gates are constrained by the fact that the operator can only be at one location at a time, has some travel time between each pair of locations, and requires some time to implement a control action. As a result, the control sequence has a lot less variables to be optimised compared to conventional MPC. On the other hand, the computational burden grows, because many different routes have to be explored and their corresponding control sequences have to be optimised to evaluate the system performance associated with the routes considered. Note that this, similar to travelling salesman problems, is a combinatorial problem. As a result, the search space grows much faster than that of conventional MPC as an optimal routing problem is introduced into the optimisation problem.

## 2-2-2 Optimisation problem formulation

The irrigation canal consists of N reaches and of the same number of gates, and can be described by a linear discrete-time model:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_{\mathbf{u}}\mathbf{u}(k) + \mathbf{B}_{\mathbf{d}}\mathbf{d}(k), \qquad (2.38)$$

where  $\mathbf{x}(k)$  represents the state,  $\mathbf{u}(k)$  represents the control input,  $\mathbf{d}(k)$  represents the external disturbance input, and  $\mathbf{A}$ ,  $\mathbf{B}_{\mathbf{u}}$  and  $\mathbf{B}_{\mathbf{d}}$  are matrices of appropriate dimensions. The state vector  $\mathbf{x}(k)$  contains information about the water level in each relative to the respective setpoints at step k and the incoming flows as a result of the control inputs.

Let  $N_{\rm p}$  be the prediction horizon, where  $N_{\rm c} \leq N_{\rm p}$  and let  $N_{\rm s}$  be the number of gates to be scheduled for the operator at step k starting at the current location  $i_{\rm current}(k) \in \{1, \ldots, N\}$ . Furthermore, a matrix  $\mathbf{M} \in \mathbb{N}^{N \times N}$  is defined for the travel time between each location, such that  $\mathbf{M}_{i,i} = 0$  and  $\mathbf{M}_{i,j}$  denotes the travel time between gate *i* and gate *j*. Moreover, the time needed at each gate to implement all the activities is  $T_{\rm o}$ , which is assumed identical for all gates. The human operator interactions with the system are modelled as delayed control actions in the optimisation problem. According to the authors of [37], this is a simplification that is justified by the system's slow and stable dynamics, which mitigates the uncertainty regarding the precise implementation moment of the control action.

The optimal control problem in MoMPC consists of solving the following the mixed-integer programming problem at each control time step [20, 28, 37]:

$$\min_{\tilde{\mathbf{U}},\mathbf{p}(k)} \sum_{j=0}^{N_{\rm p}-1} (\mathbf{x}^T (k+j+1|k) \mathbf{Q} \mathbf{x} (k+j+1|k)$$
(2.39)

$$+ \mathbf{u}^{T}(k+j|k)\mathbf{R}\mathbf{u}(k+j|k)), \qquad (2.40)$$

s.t. 
$$\mathbf{x}(k+j+1|k) \in \mathcal{X}$$
, for  $j = 0, \dots, N_p - 1$ , (2.41)

$$\mathbf{u}(k+j|k) \in \mathcal{U}, \text{ for } j = 0, \dots, N_{\rm p} - 1,$$
 (2.42)

$$\mathbf{p}(k) \in \mathcal{P}_{i_{\text{current}}(k), N_{\text{s}}, T_{\text{o}}, M}(k), \tag{2.43}$$

$$a(\mathbf{p}(k), k+j|k) \neq i \Rightarrow u_i(k+j|k) = 0, \ i \in \{2, \dots, N\},$$
  
for  $i = 0$   $N = 1$ 
  
(2.44)

$$\mathbf{x}(k+j+1|k) = \mathbf{A}\mathbf{x}(k+j|k) + \mathbf{B}_{\mathbf{u}}\mathbf{u}(k+j|k) + \mathbf{B}_{\mathbf{d}}\mathbf{d}(k+j|k),$$
(2.45)  
for  $j = 0, \dots, N_{\mathbf{p}} - 1,$ 

where  $\mathbf{Q}$  is a positive semi-definite weighing matrix and  $\mathbf{R}$  is a positive definite weighing matrix,  $\tilde{\mathbf{U}}(k) = (\mathbf{u}^T(k|k), \dots, \mathbf{u}^T(k+N_{\rm p}-1|k))^T$  is the input sequence, with  $\mathbf{u}(k+j|k) = (u_1(k+j|k), \dots, u_N(k+j|k))^T$ . The sets  $\mathcal{X}$  and  $\mathcal{U}$  are the sets of closed convex operational constraints on the states and input, respectively. Furthermore,  $p(k) \in \mathbb{N}^{N_{\rm s}}$  denotes the optimised path at time step k, which is a subset of all paths of length  $N_{\rm s}$  starting from  $i_{\rm current}$ , which is the first element of p(k). Moreover,  $\mathcal{P}_{i_{\rm current}(k),N_{\rm s},T_{\rm o},M}(k)$  denotes a set of allowed paths of length  $N_{\rm s}$  that have  $i_{\rm current}$  as origin, given the duration  $T_{\rm o}$  and the travel times  $\mathbf{M}$  with repetitions allowed. It is important to note that the path variable is defined in such a way that when the has operator finished the work in one location, he/she is sent to the next location without delay. In addition,  $a(\mathbf{p}(k), k+j|k)$ , denotes the availability of the operator at time step k + j when travelling a path  $\mathbf{p}(k)$ . The availability function returns 0 if the operator is available at prediction step k + j, results in  $u_i(k+j|k) = 0$ . Otherwise, the input only has to satisfy the constraints set in (2.42).

## 2-2-3 Results

The algorithm is tested on a model of the Dez main irrigation canal in Iran. Information about this irrigation canal can be found in Appendix B-1. In the test case, some noise is added to the water level measurements, gate positions, and offtake schedule. Furthermore, the water level measurements of the operators are the only source of information the central controller has to update the internal model states.

The head gate flow is controlled using standard MPC with a control time step  $T_c$ . Thus, the head gate flow is coordinated with the solution of MoMPC when an operator appears at a local site at an integer multiple of the control time step  $T_c$ .

The internal model used is the flow-controlled model from Section 2-1-4. Furthermore, when the operators implement a flow change at the scheduled gates it is assumed that a local flow controller is present to keep the flow of the gates at this new reference, until updated by another visit of an operator.

To solve the mixed-integer problem from (2.39)-(2.45), an exhaustive search approach is performed. The non-zero indices of the control signal are determined by the path that the operator takes and the times when he/she arrives. By fixing the path the mixed-integer problem from (2.39)-(2.45) is transformed into a Quadratic Programming (QP) problem. Next, by looping through all the possible paths an operator can take and calculating the associated minimum cost of these paths, the global optimum to (2.39)-(2.45) can be found. However, as the number of possible paths scales with the power  $N_{\rm s}$ , the number of paths and associated QP problems renders using exhaustive search intractable for large-scale systems with multiple operators and larger values of  $N_{\rm s}$ .

The proposed MoMPC scheme performs better than the uncoordinated local PI controllers, but slightly worse than the nominal centralised MPC, which could be regarded an upper bound on performance [37].

# 2-3 Mobile TIO-MPC

One of the weaknesses of MoMPC presented in Section 2-2 is that the water levels are not maintained around the setpoint during the implementation of a control action, which can result in less effective water offtake flows to the users. There are two limitations that cause this problem:

- 1. The open water system is discretised, so the evolution of the system states is only predicted for whole sampling steps and control actions take place at integer multiples of the control time step. Therefore, the controller has no knowledge on what happens in between sampling steps. This can be detrimental to the system performance, as the flow delays that are assumed in the ID model will likely not match those of the real open water system. Consequently, the water level deviates temporarily from the setpoint, until the operator adjusts the gate. For a more in detail explanation about this disadvantage, we refer to [28].
- 2. The second limitation arises from the restrictions on the controller on deciding when the gate should be opened, as the operator travel times are fixed. Accordingly, the only way for the controller to delay actuation at a certain gate is by sending the operator to visit other gates first. Therefore, the capabilities of the controller to synchronise with system dynamics are inadequate.

The first limitation can be reduced by making the sampling and control time steps smaller. However, the controller is then still not able to consistently synchronise with the arrival of the water flow, due to the fixed travel times. This gives motivation to develop an improved algorithm, where the precise time instants of the operator's action are determined by the controller and coordinated with the system's dynamics. In [28], some aspects of Time Instant Optimisation Model Predictive Control (TIO-MPC) are added to the MoMPC approach,

which is called Time Instant Optimisation Mobile Model Predictive Control (TIO-MoMPC). The TIO-MPC approach was first proposed in traffic control [8], and was later applied to water systems [9, 34]. In TIO-MPC, the allowed number of switches of a discrete element, e.g., a pump or a gate, over the entire control window is set a priori. Instead of having the controller decide whether to switch the pump off/on for each time step, the continuous-time instants of the pre-determined switching are optimised. Therefore, the number of optimisation variables can be reduced significantly, provided the number of switching instants is low. However, this is not of much use to MoMPC as the number of switching instants is already determined through the path (actuation can only take place if an operator is at a gate). Moreover, the gates are not similar to on/off switching pumps; the gate flow is increased by opening a gate a certain real-valued amount. The aspect of TIO-MPC that is useful for MoMPC is the introduction of time instants. These time instants determine when the operator should measure or actuate at a certain gate, bounded by the minimal travel times and implementation times. In MoMPC [20, 37], these time instants are fixed to the minimal time required for travelling and implementation of the control actions. By optimising the delays on measurements and actuations, through the time instants, the system performance can be improved, as the actuation actions can be synchronised with the system dynamics. For this to happen, a continuous-time modelling approach is needed to define the (actuation) time instants as real-valued. In [28], the details of how this would be achieved are not discussed and in the case study the algorithm is discretised. In addition to including time delays, the authors from [28] include a penalty on the number of location changes on a specific path that the operator needs to take, in order to minimise the workload of the human operator.

## 2-3-1 Optimisation problem formulation

Using so called continuous sampled-data MPC [15, 21, 31] a continuous-time model of the system is used, but measurements are taken from the system and new control actions are applied at consecutive integer multiples of the control time step  $T_{\rm c}$  only (control and sampling time steps are equal for MoMPC), see Figure 2.14. The operator measures and actuates the system at integer multiples of the control time step. After taking a measurement at the continuous activation time  $t_{\rm a}$ , the operator communicates the measured state to the model predictive controller using a mobile device. Next, the model predictive controller predicts the future states of the system using a continuous-time model, the measurements, and an offtake schedule. These predictions are used to find the optimal operator schedule that optimises the state of the system, while taking into account the constraints. Once the optimal schedule is found, the time instants for the operator to interact with the system are rounded to integer multiples of the control time step. This is depicted as a C2D (continuous-time to discrete-time) box in Figure 2.14. Finally, the schedule is communicated to the operator and the operator implements the first scheduled control action on the system, travels to the next location, and measures the system states at that next location; in order to close the control loop. By using a continuous-time model, a more accurate prediction of the effect of actuation can be made and real-valued optimisation variables can be used.

The continuous-time model of a canal is described by:

$$\dot{\mathbf{x}}_{c}(t_{a}) = \mathbf{A}_{c}\mathbf{x}_{c}(t_{a}) + \mathbf{B}_{c,\mathbf{u}}\mathbf{u}_{c}(t_{a}) + \mathbf{B}_{c,\mathbf{d}}\mathbf{d}_{c}(t_{a}), \qquad (2.46)$$

Master of Science Thesis

R.C. Kassing



Figure 2.14: Schematic illustrating the control scheme used in continuous sampled-data MPC.

where a subscript 'c' is added to the variables of the continuous-time model to distinguish them from the discrete-time model presented in (2.38). Moreover,  $t_a$  indicates the continuous activation time at which the operator communicates a measurement of the central controller. Note that the delay between measuring and subsequent communication to the central is assumed negligible.

Equivalently to the discrete case, the control input is written as a collection of N elements:

$$\mathbf{u}_{c}(t_{a}) = (u_{c,1}(t_{a}), \dots, u_{c,N}(t_{a}))^{T},$$
 (2.47)

for N reaches with each one gate in the irrigation canal. Furthermore, the path variable is defined as:

$$\mathbf{p}(t_{\mathbf{a}}) = (p_1(t_{\mathbf{a}}), \dots, p_{N_{\mathbf{s}}}(t_{\mathbf{a}}))^T, \ p_\ell(t_{\mathbf{a}}) \in \{1, \dots, N\},$$
(2.48)

which specifies the sequence of gate indices the operator needs to travel starting from the origin gate  $p_1(t_a) = i_{current}$ . Note that the indices of path variable  $\mathbf{p}(t_a)$  are allowed to be repeated, as it can be worthwhile to visit certain locations multiple times. To reduce to computational burden, the search space is reduced by limiting the maximum distance between consecutive locations:

$$|p_{\ell+1}(t_{\rm a}) - p_{\ell}(t_{\rm a})| \le N_{\rm limit}, \text{ for } \ell = 1, \dots, N_{\rm s} - 1.$$
 (2.49)

The time instants when the operator should arrive at the  $N_s$  gates to apply the new gate settings are denoted by:

$$\mathbf{T}(t_{\rm a}) = (T_1(t_{\rm a}), \dots, T_{N_{\rm s}}(t_{\rm a}))^T, \ T_\ell(t_{\rm a}) \in \mathbb{R},$$
(2.50)

where  $T_1(t_a) = t_a$  is fixed, due to the first element of the path always being  $p_1(t_a) = i_{\text{current}}$ . Furthermore, the control actions to be implemented by the human operator are denoted:

$$\mathbf{u}^{\text{operator}}(t_{a}) = (u_{1}^{\text{operator}}(t_{a}), \dots, u_{N_{s}}^{\text{operator}}(t_{a}))^{T}, \ u_{\ell}^{\text{operator}}(t_{a}) \in \mathbb{R},$$
(2.51)

R.C. Kassing

Master of Science Thesis

where  $u_1^{\text{operator}}(t_a)$  specifies the actions to be applied at gate  $i_{\text{current}}$ , which are to be determined by the controller at activation time  $t_a$ . Next, let  $N_c$  be the control horizon, with  $N_c < N_p$ . The entries of  $\mathbf{T}(t_a)$  and the corresponding entries of  $\mathbf{u}^{\text{operator}}(t_a)$  that are outside of the control window  $[t_a, t_a + N_c T_c]$  are not considered for the control signal. This allows the controller to schedule fewer switching instants than  $N_s$  within the prediction.

The path, time instants, and operator actions are used to parameterize the control input  $\tilde{\mathbf{U}}_{c}(t_{a})$ , which denotes the trajectories of the control input  $\mathbf{u}_{c}(t_{a})$  for the whole control window  $\tau \in [t_{a}, t_{a} + N_{c}T_{c}]$ :

$$u_{\mathrm{c},i}(\tau|t_{\mathrm{a}}) = \begin{cases} u_{\ell}^{\mathrm{operator}}(t_{\mathrm{a}})\delta(\tau - (T_{\ell}(t_{\mathrm{a}}) + T_{\mathrm{d}})) & \text{if } i = p_{\ell}(t_{\mathrm{a}}), \\ 0 & \text{otherwise,} \end{cases}$$
(2.52)

where  $\delta$  denotes the Dirac impulse function,  $T_{\rm d} \in \mathbb{R}$  the time delay between time of measurement and implementation of the control action, and  $\tau \in [t_{\rm a}, t_{\rm a} + N_{\rm c}T_{\rm C}]$  a time instant in the control window.

Using the path variable  $\mathbf{p}(t_a)$ , the number of location changes that are scheduled for the operator in the control window can be determined. The number of location changes is minimised as it is a metric for the amount of workload the operator faces on a predicted path. Let  $n_{N_c}(t_a)$  denote the number of gates the operator has to visit in the control window. The number of gates the operators has to visit can be determined by solving a constrained integer programming problem:

$$n_{N_{\rm c}}(t_{\rm a}) = \arg\max_{\ell} \ell, \qquad (2.53)$$

s.t. 
$$T_{\ell} \le t_{\rm a} + N_{\rm c} T_{\rm c},$$
 (2.54)

$$2 \le \ell \le N_{\rm s}$$
, where  $\ell \in \mathbb{N}$ . (2.55)

To ensure a minimal state update frequency, it is assumed that  $n_{N_c}(t_a) \ge 2$ . Consequently, the additional workload penalty  $J_{\text{operator}}(t_a)$  is denoted as:

$$J_{\text{operator}}(t_{\text{a}}) = \sum_{s=1}^{n_{N_{\text{c}}}-1} \mathbb{1}_{p_{s+1}(t_{\text{a}}) \neq p_{s}(t_{\text{a}})},$$
(2.56)

and serves to minimise the number of location change for the operator. The indicator function,  $\mathbb{1}_A$ , is defined as:

$$\mathbb{1}_{A} = \begin{cases}
1 & \text{if } A \text{ is true,} \\
0 & \text{otherwise.}
\end{cases}$$
(2.57)

Moreover, the cost function is reformulated for the continuous-time setting:

$$J_{\text{MoMPC}}(t_{\text{a}}) = \int_{t}^{t_{\text{a}}+N_{\text{p}}T_{\text{c}}} (\mathbf{x}_{\text{c}}^{T}(\tau|t_{\text{a}})\mathbf{Q}\mathbf{x}_{\text{c}}(\tau|t_{\text{a}}) + \mathbf{u}_{\text{c}}^{T}(\tau|t_{\text{a}})\mathbf{R}\mathbf{u}c(\tau|t_{\text{a}}))d\tau.$$
(2.58)

The optimal control problem that is to be solved at each new location visit of the operator is

Master of Science Thesis

R.C. Kassing

then:

$$\min_{\mathbf{p}(t_{a}),\mathbf{T}(t_{a}),\mathbf{u}^{\text{operator}}(t_{a})} J_{\text{MoMPC}}(t_{a}) + w J_{\text{operator}},$$
(2.59)

s.t. 
$$\mathbf{x}_{c}(\tau|t_{a}) \in \mathcal{X}, \ \forall \tau \in [t_{a}, t_{a} + N_{p}T_{c}],$$
 (2.60)

$$\mathbf{u}_{c}(\tau|t_{a}) \in \mathcal{U}, \ \forall \tau \in [t_{a}, t_{a} + N_{p}T_{c}],$$
(2.61)

$$\dot{\mathbf{x}}_{c}(t_{a}) = \mathbf{A}_{c}\mathbf{x}_{c}(\tau|t_{a}) + \mathbf{B}_{c,\mathbf{u}}\mathbf{u}_{c}(\tau|t_{a}) + \mathbf{B}_{c,\mathbf{d}}\mathbf{d}_{c}(\tau|t_{a}), \ \forall \tau \in [t_{a}, t_{a} + N_{p}T_{c}],$$
(2.62)

$$T_{\ell+1}(t_{\rm a}) \ge T_{\ell}(t_{\rm a}) + T_{\rm o} + T_{\rm d} + M_{p_{\ell}(t_{\rm a}), p_{\ell+1}(t_{\rm a})}, \text{ for } \ell = 1, \dots, N_{\rm s} - 1,$$
 (2.63)

$$T_1(t_a) = t_a, \ p_1(t_a) = i_{\text{current}}(t_a),$$
 (2.64)

$$T_2(t_a) \le t_a + N_c T_c,$$
 (2.65)

$$|p_{\ell+1}(t_{\rm a}) - p_{\ell}(t_{\rm a})| \le N_{\rm limit}, \text{ for } \ell = 1, \dots, N_{\rm s} - 1,$$
(2.66)

$$u_{\mathrm{c},i}(\tau|t_{\mathrm{a}}) = \begin{cases} u_{\ell}^{\mathrm{operator}}(t_{\mathrm{a}})\delta(\tau - (T_{\ell}(t_{\mathrm{a}}) + T_{\mathrm{d}})) & \text{if } i = p_{\ell}(t_{\mathrm{a}}), \\ 0 & \text{otherwise,} \end{cases}$$
(2.67)

where w is positive parameter that indicates the relative importance of the workload of the operator against the system performance  $J_{\text{MoMPC}}$ . At every new location the operator visits, he/she communicates the new measurements to the control centre, where they are used by the controller to solve the optimisation problem (2.59)–(2.67). The controller returns the control action  $u_1^{\text{operator}}(t_a)$ , that needs to be implemented at the current location  $i_{\text{current}}$  at time  $t_a + T_d$  and provides the operator with the next location to travel to and the time  $T_2(t_a)$  at which the operator is expected to measure at that next location.

## 2-3-2 Results

The TIO-MoMPC method is evaluated on the same irrigation canal as the MoMPC method from Section 2-2-3: the Dez main canal in Iran. However, no noise is added to the system and perfect estimates of the states are assumed to be available at every time step.

The head gate is controlled by standard MPC with control time step  $T_c$  and the head gate flow is coordinated with the solution to the TIO-MoMPC problem. Moreover, the continuous sampled-data model (2.59)–(2.67) is approximated by the discrete-time, flow-controlled model from Section 2-1-4, with a control time step of  $T_c = 5$  minutes. Furthermore, it is assumed that a local flow controller is present at each gate to maintain the gate flow at the reference value set by the operator.

To solve the mixed-integer nonlinear programming problem (2.59)-(2.67) a Genetic Algorithm (GA) is used to search through the possible paths and time instants. For TIO-MoMPC, exhaustive search is intractable as the number of possible routes is very big, due to the introduction of time instants to the optimisation problem.

The TIO-MoMPC method outperforms the MoMPC method due to the added degree of freedom of delaying actuation to synchronise with the system dynamics [28]. Moreover, the workload of the operator is decreased by penalising the number of location changes along the path of the operator.

# 2-4 Conclusions on Model Predictive Control of Open Water Systems

In the first half of this chapter, the subsystems of open water systems that are part of the closed-loop or affect the water system at its outer limits have been discussed. The subsystems have been formalised in mathematical models that are interconnected to be used as internal models in MPC. Two types of models have been presented and compared: gate-controlled and flow-controlled ID models. The flow-controlled model is only representative for the water system, if local flow controllers are available to keep the flow constant in between actuation instants of the human operator. On the other hand, the gate-controlled model approximates the nonlinear behaviour of a water system without local flow controllers accurately.

Another important conclusion is that the linear ID model is only representative for a certain type of open water system, namely those not sensitive to resonance waves, i.e., irrigation canals with long and steep reaches. Moreover, the accuracy of the ID model depends on the deviation from the working point; if the flow conditions of the canal change, the internal model parameters, such as the storage area and delay times, may have a mismatch with the real system.

The human scheduling approach discussed in Section 2-2 is a potential solution to the irrigation problems that developing countries face. The approach is able to schedule multiple operators to maintain optimal water levels, while both sensing and actuating are sparse. Thus, no equipment needs to be installed and maintained at the local sites, instead the operator uses a mobile device. Then, in [28] a continuous-time model is used to more accurately predict the effect of inputs on the evolution of the water system. Moreover, an extra degree of freedom is added to MoMPC: the controller can decide at what time the operator should update the gate settings at a specific location. This allows the controller to schedule the implementation instants of flow changes such that the arrival of water flow is better synchronised with the water level dynamics of the canal. As a result, the system performance is improved.

In [28, 37], a mathematical model of the Dez main canal in Iran is used to test the control algorithms. This irrigation canal has a linear layout, is resistant to resonance waves (long and steep reaches), and all of its parameters, such as reach dimensions and control structure characteristics, are available, see Section B-1. Therefore, this irrigation canal will also be used in the case study of this master thesis.

Model Predictive Control of Open Water Systems

\_\_\_\_\_

# Chapter 3

# Multiple-Action Mobile Model Predictive Control

This chapter presents a new control algorithm for open water systems with mobile operators, called Multiple-Action Mobile Model Predictive Control (MaMoMPC). Similarly to the previous work from literature presented in Chapter 2, operators interact with the central controller by sending measurements and receiving instructions on where to go to next to measure and implement control actions. The central controller optimises the state of the system based on the latest information provided by the operators, while taking into account the constraints. However, in MaMoMPC the operator can perform multiple activities at each node and the algorithm allows for more activities than just measuring and actuating. Furthermore, cyclic control is included in the optimal control problem to improve the reference tracking performance of the algorithm. Moreover, it is attempted to account for the limitations caused by the sparse nature of the measuring process by providing incentives to the controller to measure frequently at all locations in the system. To that end, three methods are presented to weigh the state uncertainty at each node in the system. Next, an energy level approach is proposed that keeps track of the fatigue of each operator to ensure the operators get adequate breaks. Finally, the complete optimisation problem is presented and a conclusion is presented about the new MaMoMPC algorithm.

# 3-1 Network of an open water system

Consider an arbitrary, uncertain, and large-scale open water system that has multiple operators that travel between locations in the system to take measurements and implement control actions. The system can be described by a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ . Here,  $\mathcal{V}$  is the set of nodes that can be used to operate and measure the system locally and  $\mathcal{E}$  is the set of edges of the graph that can be used by the operators to travel between nodes  $v_i$  and  $v_j$ , if there is a direct link, such that  $(v_i, v_j) \in \mathcal{E}$ . Furthermore, there are three sets to which each node can belong:

1.  $\mathcal{V}_{\text{meas}}$ , the set of nodes where measurements can be taken,

- 2.  $\mathcal{V}_{\text{control}}$ , the set of nodes where control actions can be implemented,
- 3. and  $\mathcal{V}_{charge}$ , the set of nodes where the human operators can take a break.

Note that the sets are not necessarily disjoint, so each node can belong to multiple sets. This allows the controller to decide on which types of tasks the operator needs to perform at the local site.

**Remark.** The problem of controlling a networked system by operators travelling between the nodes has similarities to a multiple travelling salesman problem. However, in our case not all of the nodes need to be visited, nodes can be visited multiple times, and the operators do not have to return to the origin nodes.

The model of the open water system consisting of  $N_{\text{node}}$  nodes can be described by a continuoustime state space description:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}_{\mathbf{u}}\mathbf{u}(t) + \mathbf{B}_{\mathbf{d}}\mathbf{d}(t) + \mathbf{w}(t), \qquad (3.1)$$

$$\mathbf{y}(t) = \mathbf{H}(t)\mathbf{x}(t) + \mathbf{v}(t), \tag{3.2}$$

where  $\mathbf{x}(t) \in \mathbb{R}^n$  denotes the state,  $\mathbf{u}(t) \in \mathbb{R}^m$  denotes the input,  $\mathbf{d}(t) \in \mathbb{R}^r$  denotes the (predicted or known) exogenous input,  $\mathbf{w}(t) \in \mathbb{R}^n$  denotes the unknown process noise,  $\mathbf{y}(t) \in \mathbb{R}^{p(t)}$ denotes the measured output, and  $\mathbf{v}(t) \in \mathbb{R}^{p(t)}$  is the unknown measurements noise. Furthermore,  $\mathbf{A}$ ,  $\mathbf{B}_{\mathbf{u}}$ ,  $\mathbf{B}_{\mathbf{d}}$ , and  $\mathbf{H}(t)$  are system matrices of suitable dimensions. Note that  $\mathbf{H}(t)$  is time-varying in the number of rows and the number of entries as the number of measurements depends on the number of operators sending measurements to the controller at that time.

# 3-2 Control algorithm

In the Mobile Model Predictive Control (MoMPC) implementation from [20, 37], the arrival and actuation time instants follow directly from the fixed travelling times between any two locations. As an improvement, [28] suggested using a time instant approach, inspired by Time Instant Optimisation Model Predictive Control (TIO-MPC), to synchronise the human operator's actions with the system's dynamics by allowing the controller to optimise the delays of the actuation instants in the operator schedules. Therefore, introducing TIO-MPC to MoMPC will increase the computational burden, as two extra optimisation variables are added for each gate visit (a measuring and an actuation time instant). Moreover, this should result in improved performance, as the time delays that follow from the time instants can be set to zero to obtain the minimal travelling time approach from [20, 37]. Note that it is assumed that there is no delay between taking measurements, sending them to the controller, and receiving back instruction from the controller.

To define the time instants from (3.7) as real-valued variables a continuous-time model is used. Similarly to [28], so called sampled-data Model Predictive Control (MPC) [15, 21, 31] is adopted, in which a continuous-time model of the system is used, but measurements are taken from the system and new control actions are applied at consecutive sampling times only. The control schematic and elaboration of this approach can, respectively, be found in Figure 2.14 and Section 2-3-1. Furthermore, the details of how this continuous sampleddata MPC would work in practice are not part of this thesis. Accordingly, for the case study the continuous sampled-data model is approximated by the gate-controlled version of the discrete-time models presented in Section 2-1-4.

In the next subsections, a new control algorithm is presented that expands on the Time Instant Optimisation Mobile Model Predictive Control (TIO-MoMPC) approach from [28]. This new mobile MPC approach is called MaMoMPC, where for brevity 'Time Instant Optimisation' is omitted from the name. The control algorithm distinguishes itself from [20, 28, 37] by allowing the controller to schedule more than one action at a node, by including cyclic control, and by incorporating breaks for the human operators. Furthermore, three methods are proposed to weigh and reduce the uncertainty about the system's state in MaMoMPC. In a similar way to the approaches presented in Chapter 2, the head gate is controlled using conventional MPC with control time step  $T_c$ . Therefore, the head gate flow is coordinated with the solution to the MaMoMPC problem.

### 3-2-1 Operator's schedule

Consider a network with  $N_{op} \geq 1$  operators indexed by  $j \in \mathcal{O} = \{1, \ldots, N_{op}\}$ . Assume that at the continuous activation time  $t_a$  some operators take the measurements from their present location and communicate them to the controller. The controller receives the measurements, updates the corresponding internal model states, and calculates some optimal schedule and control actions for each operator, which are communicated back to the operators' mobile devices (only to the operators that just measured). Furthermore, also at time  $t_a$ , the operators that are not measuring at a location may be involved in some type of activity. An operator status function for each operator j is defined as:

$$st_j(t_a) = \begin{cases} 1 & \text{if operator } j \text{ is travelling at time } t_a, \\ 0 & \text{otherwise.} \end{cases}$$
(3.3)

It is not possible to communicate with operator that are travelling, so the next scheduled location of their route is fixed. The operators that are completing activities at a location are allowed  $T_{\text{finish}}$  time units to finish the activities before travelling to the next gate. The schedule that is received after measuring can be completely different from the one received previously.

Next, the path variable  $\mathbf{p}_j(t_a)$  for operator j is defined, which contains  $N_s$  consecutive indices of nodes to be visited by the operator:

$$\mathbf{p}_{i}(t_{\mathrm{a}}) = (p_{1,j}(t_{\mathrm{a}}), \dots, p_{N_{\mathrm{s}},j}(t_{\mathrm{a}})), \text{ where } p_{\ell,j}(t_{\mathrm{a}}) \in \mathcal{V}.$$

$$(3.4)$$

The first node in the path is the current operator position  $p_{1,j}(t_a) = v_{\text{current}}$  if  $st_j(t_a) = 0$ . Otherwise, if the operator is travelling,  $st_j(t_a) = 1$ , the first node in the path is the node the operator is travelling to:  $p_{1,j}(t_a) = p_{2,j}(t_{a,\text{prev}})$ , where  $t_{a,\text{prev}}$  indicates the preceding activation time of the controller. The elements  $p_{\ell,j}(t_a)$  of the path variable  $\mathbf{p}_j(t_a)$  may appear more than once in the sequence, as it may be beneficial for an operator j to inspect and/or actuate a subset of possible locations multiple times. Moreover, for the controller to schedule multiple activities for one node it is allowed to visit the same node more than once in succession. Along with the path an activity schedule is provided to the operator, containing the activities to complete for each location:

$$\mathbf{a}_{j}(t_{\mathbf{a}}) = (a_{1,j}(t_{\mathbf{a}}), \dots, a_{N_{\mathbf{s}},j}(t_{\mathbf{a}})), \quad \text{where } a_{\ell,j}(t_{\mathbf{a}}) \in \mathcal{A},$$
  
$$\mathcal{A} = \{ \operatorname{act}_{1}, \operatorname{act}_{2}, \operatorname{act}_{3} \},$$
(3.5)

where

$$act_1$$
 : measure at location,(3.6) $act_2$  : operate at location,(3.6) $act_3$  : recharge at location.

Depending on the system, any other types of activities can be added to the set of activities  $\mathcal{A}$ . Furthermore, the types of activities that can be performed at each scheduled location depend on the subsets of  $\mathcal{V}$  the corresponding nodes belongs to. Moreover, the amount of time it takes to perform the measuring, subsequent communication to the central controller, and receiving of the control action (if included in the activities for that node) is assumed to be negligible. The other two activities, implementing a received control action and recharging, take a certain amount of time to complete. The duration to implement a control action is assumed to be constant and is denoted by  $T_{\text{control}}$  and is predefined. On the other hand, the duration the operator spends recharging  $T_j^{\text{recharge}}(t_a)$ , i.e., having a break, can be dynamic and depends on his/her energy level and recharging speed, see Section 3-2-5. Note that for our case, humans are considered and break times are often predefined. However, this energy recharging is introduced as a general description that can also be used to schedule the recharging of the (electric) vehicles the operators drive in to travel around the open water system.

In this thesis, we assume that at every location visited the operator will first measure (if possible at that location), in order to keep the central controller updated with an accurate prediction of the system. However, the number of control actions to implement depends on the activity schedule from (3.5). Therefore, the path and activity schedule are first determined and subsequently the time instants at which the measuring, actuation, and recharging should take place. This nested approach will be elaborated on in Section 4-1-4.

The vector of  $N_s$  time instants at which operator j should commence the activities at the consecutive locations of the path  $\mathbf{p}_i(t_a)$  is defined as:

$$\mathbf{T}_{j}^{\mathrm{act}}(t_{\mathrm{a}}) = (T_{1,j}^{\mathrm{act}}(t_{\mathrm{a}}), \dots, T_{N_{\mathrm{s}},j}^{\mathrm{act}}(t_{\mathrm{a}})), T_{\ell,j}^{\mathrm{act}}(t_{\mathrm{a}}) \in \mathbb{R}.$$
(3.7)

Similarly to the first element of the vector  $\mathbf{p}_j(t_a)$ , if  $\mathrm{st}_j(t_a) = 0$ , the first element of the vector  $\mathbf{T}_j^{\mathrm{act}}(t_a)$  is fixed to the current time:  $T_{1,j}^{\mathrm{act}}(t_a) = t_a$ . Furthermore, if  $\mathrm{st}_j(t_a) = 1$  (operator travelling), the first element of the vector  $\mathbf{T}_j^{\mathrm{act}}(t_a)$  is set to the arrival time of the operator at the gate it is travelling to. Note that this time cannot be changed, as no communication is possible with travelling operators.

The  $N_{j,\text{control}}(t_a)$  control actions to implement (how much to open the gate) are dependent on operator j's activity schedule. Moreover, the vector of  $N_{j,\text{control}}(t_a)$  control actions to be executed by operator j at the prescribed locations at times  $T_{\ell,j}^{\text{act}}(t_a)$ , where  $a_{\ell,j}(t_a) = \text{act}_2$  are denoted by:

$$\mathbf{u}_{j}^{\mathrm{op}}(t_{\mathrm{a}}) = (u_{1,j}^{\mathrm{op}}(t_{\mathrm{a}}), \dots, u_{N_{j,\mathrm{control}}(t_{\mathrm{a}}),j}^{\mathrm{op}}(t_{\mathrm{a}})), \quad \mathrm{where} \ u_{\ell,j}^{\mathrm{op}}(t_{\mathrm{a}}) \in \mathbb{R}.$$
(3.8)

R.C. Kassing

Master of Science Thesis

The control action to apply at the current position and all the other locations in the path vector are computed given the most up-to-date measurements provided by the operators.

There is a fixed distance between any pair of nodes  $v_i$  and  $v_j$ . For sake of simplicity, the operators are assumed to travel with a constant speed over the whole open water system. From the knowledge of the constant speed between each node of the network, the travel time between each pair of nodes is assumed fixed and can be calculated. This travel time between pair of nodes  $(v_i, v_j)$  is denoted by  $T_{\text{travel}}(v_i, v_j)$ .

The number of possible paths, activities, and control actions grows exponentially fast with  $N_{\rm s}$ ; in order to limit the computational burden, it is common in MPC to only evaluate the control variables until a predefined control horizon. After this control horizon the evolution of the system is still evaluated and included in the objective function. To avoid confusion with the standard terminology of MPC, the control horizon  $N_{\rm c}$  is split into two definitions: variable control horizon and fixed control horizon. The variable control horizon,  $N_{\rm c,schedule} \leq N_{\rm p}$ , indicates the length of the control time window in which the controller can freely schedule the operator to visit a maximum of  $N_{\rm s}$  gates to perform activities, see Figure 3.1b. This means that any entries of  $\mathbf{T}_{j}^{\text{act}}(t_{\text{a}})$  that surpass the variable control horizon will be ignored in the optimisation problem, along with the corresponding entries in  $\mathbf{p}_i(t_a)$ ,  $\mathbf{a}_i(t_a)$ , and  $\mathbf{u}_i^{\text{op}}(t_a)$ . The fixed control horizon indicates the length of the time window, starting at the current time instant until when the control sequence can have non-zero elements, which is set equal to the prediction horizon  $N_p$  for MaMoMPC. In [20, 28, 37], the gate settings were not changed during the fixed control window. However, in MaMoMPC the non-zero control sequence elements during the fixed control window are determined by cyclic control, see Section 3-2-3. Note that the evolution of the system states, e.g., water level errors, is predicted until the prediction horizon, see Figure 3.1a.

Using the variables  $\mathbf{p}_j(t_a)$ ,  $\mathbf{a}_j(t_a)$ ,  $\mathbf{T}_j^{\text{act}}(t_a)$ , and  $\mathbf{u}_j^{\text{op}}(t_a)$ , for all  $j \in \mathcal{O}$  the control signal  $\tilde{\mathbf{U}}(t_a)$  is constructed. This  $\tilde{\mathbf{U}}(t_a)$  denotes the trajectories of the control input  $\mathbf{u}(t)$  for the whole duration  $[t_a, t_a + N_p T_c]$  of the control window. As the controller can only freely schedule the gates to be visited and actuated within the variable control window, the non-zero part of  $\tilde{\mathbf{U}}(t_a)$  during  $\tau \in [t_a, t_a + N_{\text{c,schedule}} T_c]$  is defined as:

$$u_i(\tau|t_{\mathbf{a}}) = \begin{cases} u_{\ell,j}^{\mathrm{op}}(t_{\mathbf{a}})\delta(\tau - T_{\ell,j}^{\mathrm{u}}(t_{\mathbf{a}})) & \text{if } v_i = p_{\ell,j}(t_{\mathbf{a}}), \\ 0 & \text{otherwise,} \end{cases}$$
(3.9)

where  $\delta$  denotes the Dirac impulse function.

In this thesis, we assume that the operators are able to implement the control actions at the exact times provided by the controller. In reality, the time of implementation by the operator includes some uncertainty as it depends on factors such as their workload and fatigue. However, for systems with slow dynamics the uncertainty in time of control action implementation may be mitigated. Nevertheless, the operator may have some inaccuracy in implementing the new settings of the gate, adding some uncertainty to the position and subsequent flow of the gate. The uncertainty in time of control action implementation will not be addressed in this thesis, but must not be ignored in real-world applications.

Master of Science Thesis



**Figure 3.1:** The MPC prediction window is split into a variable control window (green) and a fixed control window (blue). In Figure 3.1b, the predicted operator schedule is depicted; the controller is free to schedule the operator until the variable control horizon. Furthermore, in Figure 3.1a the predicted evolution of the water level errors is depicted. Moreover, this evolution of the water level errors during the whole prediction window is evaluated in the MPC objective function (3.10).

### 3-2-2 State estimation prediction

In the system dynamics from (3.1) and (3.2), noise terms  $\mathbf{w}(t)$  and  $\mathbf{v}(t)$  are present. These noise terms cause the expected states from the internal MPC model to drift from the states of the real system. Building an observer to estimate the real states is not trivial, as it must be able to deal with fusing measurements coming from a subset of the operators in the network obtained at unevenly spaced sampling steps related to operators arriving at the gates. Moreover, the output/measurement matrix of the system is time-varying, since at every activation time there may be a different number of operators available at the gates to take measurements. As a result of having only those sparse measurements from operators to update the system states, there is some discrepancy between the predicted state of the system (by the internal model) and the real state of the open water system. Therefore, not measuring at certain locations for an extended period of time can cause instability in the system, constraint violation, and reduced system performance.

In this thesis, observers are not considered to estimate the states. Instead, when a water level measurement is received by the central controller, the corresponding water level error state in the internal model state vector  $\mathbf{x}(t)$  (3.1) is directly updated by the unfiltered measurement. This is based on the assumption that the local measurements will not have much measuring noise on them. However, not measuring at a location for a prolonged period of time will have a large effect on the discrepancy between the estimated water level (from the internal model) and the actual water level of that location, as external influences disturb the system. Therefore, instead of attempting to built and include a difficult observer in the optimisation problem the measuring frequency of some parts of system is increased. To ensure the predicted water level states do not drift too far from the actual states, the information gathering capabilities of the system will have to be augmented, see Section 3-2-4.

With the internal model state estimates for the whole prediction window, the cost function  $J_{\text{MoMPC}}$  that characterises the reference tracking performance of the system is defined. It is desired to penalise both positive and negative deviations of the error levels and control inputs. Furthermore, higher deviations should be penalised more than proportionally, to prioritise driving high deviations back to the setpoint. Therefore, a quadratic objective function is chosen:

$$J_{\text{MoMPC}}(t_{\text{a}}) = \int_{t_{\text{a}}}^{t_{\text{a}}+N_{\text{p}}T_{\text{c}}} (\hat{x}^{T}(\tau|t_{\text{a}})\mathbf{Q}\hat{x}(\tau|t_{\text{a}}) + u^{T}(\tau|t_{\text{a}})\mathbf{R}u(\tau|t_{\text{a}}))\mathrm{d}\tau, \qquad (3.10)$$

where  $\mathbf{Q}$  is a positive semi-definite matrix indicating the weight on the water level errors and  $\mathbf{R}$  is a positive definite matrix indicating the weight on the change in flow increment of the head gate and change in gate opening of the other gates. Note that the objective function (3.10) is equivalent to those used in [20, 28, 37].

## 3-2-3 Cyclic control

The model predictive controller uses (3.1) to predict the evolution of the system during the prediction window. Depending on whether the water system has typically long transport delays, it is common to compute the control actions for a shorter control window and to evaluate their impact during the larger prediction window. As open water systems have large

water transport delays, the control window is indeed often limited, see [14, 37]. However, not accounting for the possible control that takes place after the control horizon can result in the controller scheduling paths and control actions that are optimal for the short term, but detrimental to the long term, closed-loop, performance. Heuristic methods can be used to bridge the control gap from the end of the control window until the end of the prediction window. These heuristic methods should preferably not increase the computational burden significantly, while still providing realistic control after the control horizon. The goal of introducing these heuristics is to improve the reference tracking performance of the model predictive controller by inserting some estimates of future inputs in the prediction.

#### Cyclic control for MoMPC

One such heuristic to include control actions after the control horizon without introducing extra optimisation parameters is cyclic control. As described in Section 3-2-1 and Figure 3.1, the controller optimises the control actions during a variable control window. In cyclic control, the control actions that the operator optimises during the duration of the variable control window are repeated until the end of the fixed control horizon. Moreover, as the to-be-optimised control actions are simply repeated no extra computational burden is added. However, in the case of MoMPC the times at which the control signal is non-zero is determined by the operator path and the optimal control actions to implement at these locations in the path are calculated by solving a Quadratic Programming (QP) problem. The cyclic control equivalent for MaMoMPC is some repetition of an operator path that the controller is likely to schedule in the fixed control window. This will enlarge the QP problem to solve for each operator path, as the cyclic path introduces extra control actions to include in the control signal  $\tilde{\mathbf{U}}(t_a)$ . However, QP problems can be solved efficiently by commercially available solvers like CPLEX, see Appendix C. Furthermore, by writing the cost function (3.10) as a function of input some optimisation variables can be eliminated to reduce the size of the QP problem [35].

#### Searching for a cyclic path for linear canals

To find a high-performance path to use as cyclic path the linear, cascaded configuration of irrigation canals is considered. As described in Section 2-1-1, irrigation canals consist of multiple reaches that are interconnected by adjustable gates which are used to control the flow from the upstream reach to the downstream reach. Moreover, the water is supplied by the head gate and the water flows downstream driven by gravity. Consider an irrigation canal with eight gates. This canal has a water level error due to water shortage at reach 3, and there is no water level error present at any of the other gates. Then, the head gate, gate 2, gate 3, and gate 4 will have to be adjusted in sequence to allow water from the head gate to supply the required water to reach 3. On the other hand, if reach 3 has a positive excess of water gate 4 will need to be adjusted along with the remaining downstream gates 5, 7, and 8 to flush the excess water out of the system. From this we create the hypothesis that to drive the water levels to the setpoints some combination of 'downstream' operator paths can be used. However, it is not known beforehand which reaches will have a shortage of water and which will have a surplus. Therefore, visiting each subsequent gate starting from gate 2 until the most downstream gate might be a viable solution. This complete downstream path will result in each gate being adjusted once and allows for the flushing of excess water and providing the required water of each reach. Furthermore, as the water flows in the same direction as the operator travels there is a possibility of synchronising the arrival of the water

with that of the operator. Moreover, because of the linear shape of the canal this complete downstream path is the solution to the travelling salesman problem of visiting each gate of the canal starting from gate 2. Therefore, this downstream path ensures some update frequency. However, visiting all of the gates is inefficient when only a few gates need to be visited (e.g., excess water at the most downstream gate).

To support the hypothesis that downstream operator paths are efficient, the behaviour of the controller from [28] is studied in a closed-loop simulation without measurement noise or process noise. The continuous sampled-data model is approximated by a discrete-time model of the west-main irrigation canal in Arizona, United States of America. More information about this irrigation can be found in Appendix B-2. Furthermore, the sampling and control time steps are set to 240 seconds and we consider only one operator to be available. The settings of the model predictive controller are:  $N_{\rm p} = 60$ ,  $N_{\rm c,schedule} = 44$ ,  $N_{\rm s} = 5$  gates,  $N_{\text{limit}} = 7 \text{ gates}, T_{\text{o}} = 0 \text{ seconds}, T_{\text{control}} = 0 \text{ seconds}, T_{\text{c}} = 240 \text{ seconds}, Q_{\text{error}} = 100,$ and  $R_{\Delta \text{flow}} = 0.05$ . Furthermore, the weighing matrix **Q** from (3.10) has value  $Q_{\text{error}}$  on the diagonal indices which correspond to water level error states of  $\mathbf{x}(k)$ . Moreover, the weighting matrix **R** from (3.10) has value  $R_{\Delta \text{flow}}$  on the diagonal. These are representative MPC settings for the Arizona west-main irrigation canal. To solve the Mixed-Integer Nonlinear Programming (MINLP) problem Genetic Algorithm (GA) is used with a stopping condition of 200 seconds of CPU time on the computer described in Section 4-1-4. The initial offtakes of the system at the start of the simulation are 0.2, 0.4, 0.2, 0.3, 0.3, and 0.2  $m^3/s$  at, respectively, reaches 1–4, 6, and 7. As a test scenario a step of  $-0.2 \text{ m}^3/\text{s}$  is added to all the offtake flows after 3 hours. Next, the controller is tested in the closed-loop simulation and the results are depicted in Figure 3.2.

By scheduling the operator to travel the downstream path, highlighted in red in Figure 3.2c, all of the gate settings are updated and the water level setpoints are tracked closely. Moreover, as the water flows downstream from the head gate to the last gate, this path is able to synchronise with the system dynamics (letting part of the delayed incoming flow through for each gate as the flow arrives). However, as the internal model used in [28] is a flow model with local flow controllers, this operator routing behaviour may not be representative for a gate-controlled internal model, which is used in the MaMoMPC algorithm presented in this chapter. Therefore, the internal flow-controlled model is changed to a gate-controlled one. Furthermore, the gate equations are linearised at every time step, the local flow controllers removed, and the plant flows are updated at every time step using the nonlinear flow equations. Moreover, the states are only updated by measurements of the water levels by the operators. There will be some mismatch between the linearised predicted flow and the nonlinear flow when adjusting the opening of a gate. Therefore, a constraint is added to restrict the maximum change in gate opening. Restricting the change in gate opening too much will result in the controller needing to schedule multiple visits to a gate to change the gates flow to the desired flow, which is inefficient. On the other hand, allowing the gate opening to be changed too much results in instability in the system, due to the mismatch in predicted flow and actual flow. After consideration of this trade-off, the change in gate opening is restricted to 20% of the maximum gate opening. The MPC settings are identical to those used in the aforementioned flow-controlled simulation, except for the cost on change in gate opening; that cost is set to  $R_{\Delta \text{flow}} = 0.1$ . The resulting water level errors, gate flows, and operator path are depicted in Figure 3.3.



**Figure 3.2:** To compensate for a scheduled step change on the offtakes the flow-controlled model predictive controller schedules the operator to travel to certain gates in the canal to adjust the gates to the desired flows. In Figures 3.2a, 3.2b, and 3.2c, the water level errors, gate flows, and operator path are depicted, respectively. Highlighted in red in Figure 3.2c is the part of the operator path during which the largest changes in gate settings are carried out the operator.



**Figure 3.3:** To compensate for a scheduled step change on the offtakes the gate-controlled controller schedules the operator to travel to certain gates in the canal to adjust the gate flows. In Figures 3.3a, 3.3b, and 3.3c, the water level errors, gate flows, and operator path are depicted, respectively. The controller schedules the operator to travel to the first four gates first, see the green highlighted path in Figure 3.3c. Next, the controller schedules the operator to take a route from the top of the canal downstream adjusting the settings of each subsequent gate, see the red highlighted path in Figure 3.3c.

As a result of the small storage area of the first three reaches of the canal, the flows of the gates of these reaches will have to be adjusted more than those with bigger storage areas (reach 4, 6, and 7). Because of the constraint on the change in gate opening the controller is unable update all of the gate openings to the desired settings in one visit. Therefore, the operator first visits gates 2–5 to adjust the flow by the maximum amount (path in green) and then visits all the gates (path in red) to achieve the desired flows.

In the gate-controlled system, it is harder to control the water levels to the setpoints, due to the restrictions on gate change and the linearisation errors of the gates. These effects are not present in the flow-controlled simulation as the plant and MPC internal model are identical. Furthermore, note that there is some steady state error for the water level of reach 1. This is due to the discrepancy between the state of the internal model of the MPC and the nonlinear model, due to the linearisation errors. Since the water level error of reach 1 is zero in the internal model, the controller has no incentive to schedule a visit for reach 1. This emphasises the importance of measuring frequently at every location in the irrigation system.

**Remark.** Although the downstream routing behaviour has been observed in this section using a single scenario on the Arizona west-main canal, the same route scheduling behaviour is found when controlling the Dez main canal (also in the operator routes in the case study results presented in Chapter 4) and throughout the experiments performed during this thesis. The main reason for choosing the Arizona west-main canal is that the search space is much smaller, as there are only six gates the operator can visit. This will make it easier for the controller to find more optimal solutions and illustrate the use of the downstream path.

To summarise and extend on the aforementioned properties of the downstream path:

- All the gates are visited; the operator starts at the top of the canal (gate 2) and visits each subsequent downstream gate. This allows for the controller to adjust the settings of all the gates. However, visiting all the gates in the complete downstream path may be inefficient, as some gates may not require actuation depending on the water shortages and surpluses of the reaches.
- The downstream path is the solution to the travelling salesman problem of starting at gate 2, visiting all the other gates once, and returning to gate 2. Therefore, it is the optimal time-efficient solution of visiting all the gates when starting at gate 2.
- The TIO-MoMPC algorithm is observed to schedule (partial) downstream paths to update the gate flows to compensate for a change in offtake flows. This behaviour is observed in both the gate-controlled and flow-controlled modelling approach.
- The downstream path can be used to update all of the gate flows, while synchronising with the water flowing downstream that originates from the head gate.

Because of these properties the downstream path is used in the cyclic path definitions in the next paragraphs.

## Formalising the cyclic path

The number of times the cyclic path can be travelled after the variable control horizon depends

on the fixed control horizon, the number of operators, travel time of the operators between each pair of gates, and the number of gates to visit in the path. Furthermore, it may not be necessary to visit every gate of the downstream path. Therefore, optimisation variables can be added that indicate at which gate the cyclic path should start and at which downstream gate to end the cyclic path at. Moreover, optimisation variables can be added that indicate the start of each repetition of the cyclic path. This allows the controller to synchronise the start of each cyclic path with the flow changes in the offtake schedule. Then the controller can update the flow of each gate to meet the changed offtake flows.

First, which part of the downstream path to schedule for a cyclic path for each operator j has to be determined:

$$p_j^{\text{cycle}}(t_{\mathbf{a}}) = (v_j^{\text{start}}(t_{\mathbf{a}}), \dots, v_j^{\text{end}}(t_{\mathbf{a}})), \text{ where } p^{\text{cycle}}(t_{\mathbf{a}}) \in \mathbb{N}^{N_{\text{node}}^{\text{cycle}}},$$
(3.11)

where  $N_{\text{node}}^{\text{cycle}}$  is the number of nodes in path  $p_j^{\text{cycle}}(t_a)$ ,  $v_j^{\text{start}}(t_a)$  and  $v_j^{\text{end}}(t_a)$  are optimisation variables that indicate, respectively, the starting node index and ending node index of the cyclic path for each operator to travel. Accordingly, every node with an index between  $v_j^{\text{start}}(t_a)$  and  $v_j^{\text{end}}(t_a)$  is visited subsequently starting from the lowest node index (corresponding to the most upstream location) until the one with the highest node index (corresponding to the most downstream location). Furthermore, the number of nodes to visit in each cycle  $N_{\text{node}}^{\text{cycle}}$  for operator j is equal to:

$$N_{\text{node}}^{\text{cycle}} = v_j^{\text{end}}(t_{\text{a}}) - v_j^{\text{start}}(t_{\text{a}}) + 1.$$
(3.12)

To ensure a downstream path is created and that the cyclic path consists of more than one location visit a constraint is created:

$$v_j^{\text{end}}(t_a) > v_j^{\text{start}}(t_a). \tag{3.13}$$

Next, the number of cyclic paths,  $N_{\text{cycle}}$ , to schedule for each operator during the fixed control horizon is established. Then, the initiation time instants of the cyclic paths for operator j are:

$$\mathbf{T}_{j}^{\text{cycle}}(t_{\mathrm{a}}) = (T_{1,j}^{\text{cycle}}(t_{\mathrm{a}}), \dots, T_{N_{\text{cycle}},j}^{\text{cycle}}(t_{\mathrm{a}})), \text{ where } T_{i,j}^{\text{cycle}}(t_{\mathrm{a}}) \in \mathbb{R},$$
(3.14)

subject to the constraints:

$$T_{i,j}^{\text{cycle}}(t_{\mathrm{a}}) > T_{\mathrm{c,schedule}}, \text{ for } z = 1, \dots, N_{\mathrm{cycle}},$$

$$(3.15)$$

$$T_{i+1,j}^{\text{cycle}}(t_{a}) \ge T_{i,j}^{\text{cycle}}(t_{a}) + \sum_{i=v_{j}^{\text{start}}(t_{a})}^{v_{j}^{\text{cycle}}(t_{a})-1} (T_{\text{travel}}(v_{i}, v_{i+1}) + T_{\text{finish}} + T_{\text{control}})$$

$$+ T_{\text{travel}}(v_{j}^{\text{end}}(t_{a}), v_{j}^{\text{start}}(t_{a})), \text{ for } i = 1, \dots, N_{\text{cycle}} - 1.$$
(3.16)

The constraint (3.16) ensures that operator j has enough time to finish the whole cyclic path and return to the top of the canal. Next, the path that operator j travels after the operator schedule horizon is:

$$\mathbf{p}_{j}^{\text{cyclic}} = (\underbrace{p_{j}^{\text{cycle}}(t_{a}), \dots, p_{j}^{\text{cycle}}(t_{a})}_{N_{\text{cycle}} \text{ elements}}).$$
(3.17)

Master of Science Thesis

R.C. Kassing

Furthermore, the time instants at which operator j actuates the  $\mathbf{p}_{j}^{\text{cyclic}}$  gates are:

$$\mathbf{T}_{j}^{\text{act,cyclic}}(t_{a}) = (T_{1,j}^{\text{act,cyclic}}(t_{a}), \dots, T_{N_{\text{node}}^{\text{cycle}} \cdot N_{\text{cycle}}, j}^{\text{act,cyclic}}(t_{a})), \\ = (T_{1,j}^{\text{cycle}}(t_{a}) + T_{\text{cum}}, \dots, T_{N_{\text{cycle}}, j}^{\text{cycle}}(t_{a}) + T_{\text{cum}}), \text{ where } T_{\ell,j}^{\text{act,cyclic}}(t_{a}) \in \mathbb{R},$$

$$(3.18)$$

where

$$T_{\rm cum} = \begin{bmatrix} 0 \\ T_{\rm d,1} \\ \vdots \\ T_{\rm d,v_j^{\rm end}(t_{\rm a})-1} \end{bmatrix}^T, \ T_{\rm d,k} = \sum_{i=1}^k (T_{\rm travel}(v_i, v_{i+1}) + (k-1) \cdot (T_{\rm control} + T_{\rm finish})).$$
(3.19)

Moreover, the control actions to apply at the time instants are:

$$\mathbf{u}_{j}^{\text{op,cyclic}}(t_{a}) = (u_{1,j}^{\text{op,cyclic}}(t_{a}), \dots, u_{N_{\text{node}}^{\text{cycle}} \cdot N_{\text{cycle}}, j}^{\text{op,cyclic}}(t_{a})), \text{ where } u_{\ell,j}^{\text{op,cyclic}}(t_{a}) \in \mathbb{R},$$

$$\text{for } \ell = 1, \dots, N_{\text{node}} \cdot N_{\text{cycle}}.$$

$$(3.20)$$

Finally, (3.17), (3.18), and (3.20) can be used to construct the trajectories of the control input during the fixed control window. This results in the following relationship for  $\tau \in [N_{c,schedule}T_c, N_pT_c]$ :

$$u_{i}^{\text{cyclic}}(\tau|t_{a}) = \begin{cases} u_{\ell,j}^{\text{op,cyclic}}(t_{a})\delta(\tau - T_{\ell,j}^{\text{act,cyclic}}(t_{a})) & \text{if } v_{i} = p_{\ell}^{\text{cycle}}(t_{a}) \text{ and } v_{i} \in \mathcal{V}_{\text{control}}, \\ 0 & \text{otherwise,} \end{cases}$$
(3.21)

where  $\delta$  denotes the Dirac impulse function. Since in (3.21)  $\tau$  is only defined until the fixed control horizon, the entries of  $\mathbf{T}_{j}^{\text{act,cycle}}(t_{a})$  (and corresponding inputs in  $\mathbf{u}_{j}^{\text{op,cyclic}}(t_{a})$ ) that exceed the fixed control horizon are simply ignored in the optimisation problem. This provides the model predictive controller with the freedom of removing cyclic paths or parts of cyclic paths to decrease the open-loop cost. Finally, the control signal from (3.9) is appended with (3.21) to include cyclic control in the control input trajectory  $\tilde{\mathbf{U}}(t_{a})$  of the control algorithm.

#### Finding cyclic paths for complex open water systems

For cascaded open water systems forming a single canal like the Arizona west-main irrigation canal in the United States of America and the Dez main canal in Iran, the optimal cyclic path is found in this section by looking at the travelling salesman solution in terms of minimising the operator travel time to visit all the gates and by examining the closed-loop MPC behaviour to a scheduled step on the offtake flows. However, this cyclic path may not hold up when multiple operators are involved and the network is more complex, i.e., the irrigation canal diverges into multiple small canals, as finding near optimal solutions using TIO-MoMPC is then much harder (the search space is much bigger).

For more complex and large scale systems it might be beneficial to limit each operator to a certain working region; in order to reduce the search space and to avoid sending operators on paths with large travel times. The cyclic paths of these more complex water systems will likely still consist of having the operator travel along with the water flow, visiting each

consecutive downstream gate. As this allows for synchronisation with the system dynamics and adjusting the flow of each gate. However, as the canals split, extra operators might be needed for each fork of the canal to visit each downstream gate, depending on the quantity and speed of the water flow in each fork.

## 3-2-4 Uncertainty weighing methods

Depending on the number of operators available, the measuring and actuating actions will be sparse in time. Furthermore, the system is subjected to noise and will always have some modelling errors. As a result, there is some uncertainty on the predictions of the system states. This uncertainty about the system states can grow large when the locations are not visited sufficiently regularly. Moreover, the uncertainty about the expected system state may result in reduced system performance and constraint violation. To ensure the expected state does not drift too far from the actual state, the information gathering capabilities of the system have to be augmented. Three methods to assign some weight to the network uncertainty are proposed:

- Static measurement attraction using the time elapsed since the last measurement, where each measurement node is equally attractive.
- Static measurement attraction using the time elapsed since the last measurement, where measurement nodes can have different attraction levels. Furthermore, a soft constraint is added on the maximum attraction level of the nodes.
- Having the operators always travel a cyclic path to ensure a certain update frequency at each gate.

#### Static measurement attraction using the time elapsed since the last measurement

Measuring regularly at every location in the system can be achieved by making nodes that have not been measured for some time more 'attractive' [7]. To that end, for each measurement node  $v_i \in \mathcal{V}_{\text{meas}}$  the time of last measurement  $t_i^{\text{last}}(\tau|t_a)$  is tracked. This is used to retrieve the time elapsed since the last measurement  $T_i^{\text{elapsed}}(\tau|t_a)$ , which is then multiplied with a certain factor  $\alpha_{\text{loc},i}$  to get an attraction level for each reach *i*. This attraction level is included in the objective function for  $\tau = [t_a, t_a + N_{\text{c,schedule}}T_c]$ :

$$J_{\text{meas}}(t_{\text{a}}) = \sum_{i=1}^{|\mathcal{V}_{\text{meas}}|} \int_{t_{\text{a}}}^{t_{\text{a}}+N_{\text{c,schedule}}T_{\text{c}}} \alpha_{\text{loc},i} T_{i}^{\text{elapsed}}(\tau|t_{\text{a}}) \mathrm{d}\tau, \qquad (3.22)$$

where

$$T_i^{\text{elapsed}}(\tau|t_{\mathbf{a}}) = \begin{cases} 0 & \text{if } \tau = T_{\ell,j}^{\text{act}}, \ p_{\ell,j}(t_{\mathbf{a}}) = v_i, \text{ and } a_{\ell,j}(t_{\mathbf{a}}) = \text{act}_1, \\ \tau - t_i^{\text{last}}(\tau|t_{\mathbf{a}}) & \text{otherwise}, \end{cases}$$
(3.23)

where  $t_i^{\text{last}}(\tau|t_a)$  is initialised as  $t_i^{\text{last}}(0|0) = 0$  and is updated to  $t_i^{\text{last}}(\tau|t_a) = \tau$  whenever for any  $j, \tau = T_{\ell,j}^{\text{act}}(t_a), p_{\ell,j}(t_a) = v_i$ , and  $a_{\ell,j}(t_a) = \text{act}_1$ ; otherwise the previously assigned value of  $t_i^{\text{last}}(\tau|t_a)$  is kept. Moreover, at every new activation of the controller the initial

Master of Science Thesis

value of  $t_i^{\text{last}}(t_a|t_a)$  is updated with the last measurement time from the previous activation:  $t_i^{\text{last}}(t_a|t_a) \leftarrow t_i^{\text{last}}(t_a|t_{a,\text{prev}})$ , in which  $t_{a,\text{prev}}$  denotes the time of the activation immediately before.

The factor  $\alpha_{\text{loc}}$  will determine the prioritisation of scheduling certain locations of the system. There are many possible ways to choose  $\alpha_{\text{loc}}$  and methods to update the factor after each MPC iteration. However, for the purpose of this thesis, the  $\alpha_{\text{loc}}$  values will stay constant during both the MPC prediction and the closed-loop simulations.

Consider the case that the system is at rest. If the system is at rest the controller should not schedule delays as there are no flows to synchronise with. Moreover, the nodes purely used for actuation are not visited as no actuation is needed. In other words, cost function (3.10) is negligible compared to cost function (3.22). The controller will then fully commit to minimising the time since last measurement multiplied by  $\alpha_{loc}$  for each reach. As time since last measurement is penalised quadratically due to the integral, the controller will likely schedule the operators to visit the reaches with the highest attraction levels first. If the factors  $\alpha_{loc}$  are the same for every reach the update frequency of each reach should be approximately the same (due to the quadratic penalty). If one of the reaches has a factor  $\alpha_{loc}$  twice as big as the other reaches, it will result in it being visited approximately twice as often as the attraction level increases twice as fast. Using this deduction the factors  $\alpha_{loc}$  can be determined based on the relative preferred update frequency.

By adding this new measurement attraction objective function (3.22) to the reference tracking objective function from (3.10) a new objective function is defined:

$$J_{\text{MoMPC}}(t_{\text{a}}) + w \cdot J_{\text{meas}}(t_{\text{a}}) \tag{3.24}$$

where w indicates the relative weight of minimising measurement attraction versus minimising the water level errors and control input.

**Remark.** It is difficult to assign weight w in (3.24). Clearly, the system performance can be improved by having a better estimate of the (future) state of the system. Nonetheless, quantifying this trade-off between (3.10) and (3.22) is difficult, if not impossible as the value of measuring depends on many stochastic processes acting on the system, and measuring only improves the reference tracking performance indirectly. However, the Pareto front between the measurement attractiveness and the combined input and error cost could be investigated to offer some more insight in the trade-off for system performance.

#### Method I: each node equally attractive

This is the most basic attraction method that ensures all water level states in the system are updated equally frequently. The factor  $\alpha_{loc}$  is equal for all reaches:

$$\alpha_{\mathrm{loc},i} = \eta, \ \forall \ i, \tag{3.25}$$

where  $\eta > 0$ . This method ensures all the nodes in  $\mathcal{V}_{\text{meas}}$  are measured equally regularly.

#### Method II: measurement attraction based on system knowledge

If information about the system is available regarding local dynamics or disturbances, the local sites that are more prone to external influences or those that require more actuation in the system can be prioritised. For example, consider an irrigation canal with two reaches and two gates. The first gate has an offtake right before it that is known to withdraw water unscheduled from the reach, while the second gate has no offtakes or other external influences acting on it. In that case, the system performance can be improved by measuring at the first gate more often, i.e., increase  $\alpha_{loc,1}$  relative to  $\alpha_{loc,2}$ , as the uncertainty of the first gate is higher than that of the second gate.

From the system dynamics, it follows that the water level error due to an external flow disturbance on a reach is determined by the storage area of a reach, see (2.4). If a reach has a small storage area relative to the other reaches, it will get higher water level errors than reaches with bigger storage areas for the same (disturbance) flow. So from expert knowledge on the expected disturbance and process noise at each gate, the storage area of each gate and relative location of the reaches in the canal  $\alpha_{loc}$  can be determined for each location.

In addition to including measurement attraction directly into the objective function, a constraint on the upper bound of the time elapsed since the last measurement is added for Method II. This has some similarities to sensor scheduling, see [3, 40]. Consider the constraint:

$$T_i^{\text{elapsed}}(\tau|t_a) \le \text{UB}_{\text{time}},$$
 (3.26)

where  $UB_{time}$  is an upper bound on the time elapsed since the last measurement. This constraint will ensure that the nodes have to be visited within a certain period. Without this constraint the controller may decide to delay the measurement to achieve a better open-loop objective value. However, if the bound is chosen too strict, it may result in infeasibility. Therefore, the constraint is implemented as a soft constraint. Nonetheless, choosing the bound violation penalty too strict may still result in poor scheduling behaviour, as it will force the controller to send the operator to visit the gate with the largest soft constraint violation each time, disregarding system performance. Choosing the bound too loose will result in it having no effect on the system performance, as there is already incentive through the objective function (3.22) to visit locations regularly.

#### Method III: fixing the path to the cyclic path

The cyclic downstream path from Section 3-2-3 visits all the gates with the same frequency, which results in frequent state updates of all the system water levels. Moreover, the delay of the flows can be synchronised with the arrival of the operator. Therefore, the third method is to set the path schedule  $\mathbf{p}_j(\mathbf{t}_a)$  equal to the cyclic path for each operator. However, the controller can still decide on the activities and implementation times for each location in the path. So, measurement attractiveness is still added in the objective function to ensure that the controller does not schedule long delays as that may result in not measuring frequently enough at each location, which can have a negative effect on the system performance. Another advantage of this method is that by fixing the path the search space of the controller is reduced significantly, allowing the controller to evaluate more permutations of the activity schedule for the fixed cyclic path.

## 3-2-5 Operator's energy level

Consider an energy level  $c_j(\tau|t_a)$  that represents the tiredness of operator j, the maximum energy level being 100% and no minimum. This energy level drains as the operator trav-

els between nodes, performs activities at these nodes, and waits at the nodes for instructions. Human operators are used to operate the canal; therefore, each node is a recharge node ( $\mathcal{V}_{charge} = \mathcal{V}$ ), as humans can just eat a sandwich in their vehicles. The rate at which the energy level depletes or recharges at every time  $\tau \in [t_a, t_a + N_{c,schedule}T_c]$  is defined as:

$$\dot{c}_{j}(\tau|t_{a}) = \begin{cases} \dot{c}_{j,\text{travel}} & \text{if } T_{\ell,j}^{\text{act}} - T_{\text{travel}}(p_{\ell-1,j}(t_{a}), p_{\ell,j}(t_{a})) \leq \tau \leq T_{\ell,j}^{\text{act}}, \\ \dot{c}_{j,\text{control}} & \text{if } T_{\ell,j}^{\text{act}} \leq \tau \leq T_{\ell,j}^{\text{act}} + T_{\text{control}} \text{ and } a_{\ell,j}(t_{a}) = \text{act}_{2}, \\ \dot{c}_{j,\text{recharge}} & \text{if } c_{j}(\tau|t_{a}) < 100\%, \ T_{\ell,j}^{\text{act}} \leq \tau \leq T_{\ell,j}^{\text{act}} + T_{j,\text{recharge}}(t_{a}), \\ & \text{and } a_{\ell,j}(t_{a}) = \text{act}_{3}, \\ 0 & \text{if } c_{j}(\tau|t_{a}) = 100\%, \\ \dot{c}_{j,\text{idle}} & \text{otherwise}, \end{cases}$$
(3.27)

where  $\dot{c}_{j,\text{control}} > 0$ ,  $\dot{c}_{j,\text{recharge}} < 0$ ,  $\dot{c}_{j,\text{travel}} > 0$ , and  $\dot{c}_{j,\text{idle}} > 0$ ; all of which can be personalised for each operator, i.e., the discharge rate for travelling can be set higher than that of implementing a control action at a node.

Using the (dis)charge rates  $\dot{c}_j(\tau|t_a)$  the operator energy level trajectory over the variable control window  $\tau \in [t_a, t_a + N_{c,schedule}T_c]$  can be calculated:

$$c_j(\tau|t_{\mathbf{a}}) = c_j(t_{\mathbf{a}}|t_{\mathbf{a}}) - \int_{t_{\mathbf{a}}}^{\tau} \dot{c}_j(\tau_{\mathrm{int}}|t_{\mathbf{a}}) \mathrm{d}\tau_{\mathrm{int}}, \qquad (3.28)$$

where  $c_j(t_a|t_a)$  is the energy level of the operator at the activation time. The energy level of each operator is initialised as  $c_j(t_a|t_a) = 100\%$  at the start of the first measurement by the operator. In subsequent activations of the controller, the energy level at the start of the MPC iteration is set equal to the prediction from the previous activation  $c_j(t_a|t_a) \leftarrow c_j(t_a|t_{a,\text{prev}})$ .

Adding a cost on the energy level of the operator to the objective function is undesirable, as it is difficult to assign a meaningful weight to it: how much system performance is gained by scheduling a break for an operator? However, not allowing for breaks for the operators will violate the European Union human working rights provisions [2]:

#### 1. Duration of the break

Recital 5 of the Directive [1] states that rest periods, to which breaks pertain, must be expressed in units of time, i.e., in days, hours and/or fractions thereof, and that workers must be granted 'adequate' breaks.

The Commission therefore considers that the rest breaks to which workers must be entitled must be clearly defined in units of time and that, although the duration of the break must be defined by collective agreement or national legislation, excessively short breaks would be contrary to the Directive's provisions.

#### 2. Timing of the break

Similarly, although the Directive [1] leaves it to collective agreements or legislation to define the terms under which the break is granted, the break should effectively allow workers to rest during their working day where the latter is longer than 6 hours. Its timing should therefore be adapted to the workers' schedule and it should take place at the latest after 6 hours. Moreover, exhausted operators may be less accurate in implementing the scheduled actions and the operators may become less agreeable, over time, to follow the controller's instructions. Therefore, a lower bound is added on the energy level of each operator:

$$c_j(\tau|t_a) \ge c_{\min,constraint}, \ \forall \tau \in [t_a, t_a + N_{c,schedule}T_c].$$
 (3.29)

This is enforced as a hard constraint, which will ensure the controller schedules breaks for the operators. Note that infeasibility will not occur, as the human operator is able to take a break at any location and the operators always arrive on the scheduled time (operator delays are not considered in this thesis).

The  $N_{j,\text{charge}}(t_a)$  times the operator recharges depends on each operator j's activity schedule. Furthermore, the vector of  $N_{j,\text{charge}}(t_a)$  recharging durations to be followed by operator j at the prescribed locations at times  $T_{\ell,j}^{\text{act}}(t_a)$  where  $a_{\ell,j}(t_a) = \text{act}_3$  are denoted by:

$$\mathbf{T}_{j}^{\text{recharge}}(t_{a}) = (T_{1,j}^{\text{recharge}}(t_{a}), \dots, T_{N_{j,\text{charge}}(t_{a}),j}^{\text{recharge}}(t_{a})), \text{ where } T_{r,j}^{\text{recharge}}(t_{a}) \in \mathbb{R},$$
(3.30)

where the recharge times  $T_{r,j}^{\text{recharge}}(t_{a})$  are bounded by:

$$0 \leq T_{r,j}^{\text{recharge}}(t_{a}) \leq \frac{100\% - c_{j}(T_{r,j}^{\text{act}_{3}}(t_{a}))}{\dot{c}_{j,\text{recharge}}},$$
for  $r = 1, \dots, N_{j,\text{charge}}(t_{a}),$ 

$$(3.31)$$

where  $T_{r,i}^{\text{act}_3}(t_a)$  is the time instant when the  $r^{\text{th}}$  recharging is to be initiated.

**Remark.** To limit the number of optimisation variables, extra constraints can be added to the recharging of the operators. Instead of having the controller decide on how much and when to recharge the operator, the operator is recharged to 100% whenever his/her charge is lower than some predefined threshold (the operators that are travelling while hitting this threshold commence their break when they arrive at the scheduled location). Furthermore, this way of scheduling breaks for the operator more closely resembles the real-world scenario in which human workers have constraints for how long they can be active until requiring a break (by law).

**Remark.** In this thesis, we only consider human operators. However, consider the case that the operators drive in (electric) vehicles. Then, these vehicles will need to be recharged or refuelled during the operation of the canal. The energy recharging formulation presented in this section can then be used to include the recharging or refuelling of these vehicles in the optimal scheduling problem.

Master of Science Thesis

# 3-3 Optimisation problem formulation

Whenever the controller is activated, by receiving a measurement from the operator, the following optimal control problem is solved:

$$\min_{\tilde{\mathbf{U}}(t_{a})} \quad J_{\text{MoMPC}}(t_{a}) + w \cdot J_{\text{meas}}(t_{a}), \tag{3.32}$$

subject to:

$$\hat{x}(\tau|t_{\rm a}) \in \mathcal{X}, \ \forall \tau \in [t_{\rm a}, t_{\rm a} + N_{\rm p}T_{\rm c}], \tag{3.33}$$

$$u_{\ell,j}^{\text{op}}(t_{\mathbf{a}}) \in \mathcal{U}, \text{ for } \ell = 1, \dots, N_{j,\text{control}}, \ j \in \mathcal{O},$$

$$(3.34)$$

$$T_{\ell+1,j}^{act} \geq \begin{cases} T_{\ell,j}^{act} & \text{if } p_{\ell+1,j} = p_{\ell,j} \text{ and } a_{\ell,j} = \operatorname{act}_{1}, \\ T_{\ell,j}^{act} + T_{\text{control}} & \text{if } p_{\ell+1,j} = p_{\ell,j} \text{ and } a_{\ell,j} = \operatorname{act}_{2}, \\ T_{\ell,j}^{act} + T_{\text{recharge}} & \text{if } p_{\ell+1,j} = p_{\ell,j} \text{ and } a_{\ell,j} = \operatorname{act}_{3}, \\ T_{\ell,j}^{act} + T_{\text{travel}} & \text{if } p_{\ell+1,j} \neq p_{\ell,j} \text{ and } a_{\ell,j} = \operatorname{act}_{1}, \\ T_{\ell,j}^{act} + T_{\text{travel}} + T_{\text{control}} & \text{if } p_{\ell+1,j} \neq p_{\ell,j} \text{ and } a_{\ell,j} = \operatorname{act}_{2}, \\ T_{\ell,j}^{act} + T_{\text{travel}} + T_{\text{recharge}} & \text{if } p_{\ell+1,j} \neq p_{\ell,j} \text{ and } a_{\ell,j} = \operatorname{act}_{3}, \\ \text{for } \ell = 1, \dots, N_8 - 1, \ p_{\ell,i} \in \mathcal{V}, \ j \in \mathcal{O}, \end{cases}$$

$$(3.35)$$

$$a_{1,j}^{\text{act}} = \operatorname{act}_{1}, \text{ if } \operatorname{st}_{j}(t_{a}) = 0, \qquad (3.36)$$

$$a_{\ell+1,j}^{\text{act}} = \operatorname{act}_{1}, \text{ if } p_{\ell+1,j} \neq p_{\ell,j} \text{ and } p_{\ell+1,j} \in \mathcal{V}_{\text{meas}}, \text{ for } \ell = 1, \dots, N_{\text{s}} - 1, \ j \in \mathcal{O}, \qquad (3.37)$$

$$T_{\ell,j}^{\text{act}} < N_{\text{c,schedule}} T_{\text{c}} \text{ only for the first occurrence of } \ell > 1, \qquad (3.38)$$

$$\text{where } p_{\ell,j} \neq p_{1,j} \text{ and } a_{\ell,j}^{\text{act}} = \operatorname{act}_{1},$$

$$a_{\ell,j} \neq \begin{cases} \operatorname{act}_{1} & \text{if } p_{\ell,j} \notin \mathcal{V}_{\text{meas}}, \\ \operatorname{act}_{2} & \text{if } p_{\ell,j} \notin \mathcal{V}_{\text{control}}, \\ \operatorname{act}_{3} & \text{if } p_{\ell,j} \notin \mathcal{V}_{\text{charge}}, \\ \text{for } \ell = 1, \dots, N_{\text{s}}, \ p_{\ell,j} \in \mathcal{V}, \ j \in \mathcal{O}, \end{cases}$$

$$(3.39)$$

system equations: (3.1) and (3.2), (3.40)schedule definitions: (3.3)-(3.9), (3.41)

be bjective function definitions: 
$$(3.10)$$
 and  $(3.22)$ – $(3.26)$ ,  $(3.42)$ 

cyclic path definitions: (3.11)-(3.21), (3.43)

and energy level definitions: (3.27)-(3.31), (3.44)

where, for conciseness, the activation time dependence  $(t_a)$  is excluded from the constraints,  $T_{\text{travel}}(p_{\ell+1,j}, p_{\ell,j}) + T_{\text{finish}}$  is reduced to  $T_{\text{travel}}$ , and  $T_{r,j}^{\text{recharge}}(t_a)$  to  $T_{\text{recharge}}$ . Furthermore, w is a positive weighing parameter and  $\mathcal{X}$  and  $\mathcal{U}$  are closed convex constraints on, respectively, the states and the inputs. See (2.29)-(2.32) for the constraints on the input. Constraint (3.35) ensures that the controller respects the minimal travel time between locations and allows the operator time to finish the scheduled activities. Constraints (3.36) and (3.37) require that the operators first measure at a location before performing other activities. Furthermore, constraint (3.38) enforces that the controller is updated with a new measurement from each operator within the variable control window. Finally, constraint (3.39) checks for each activity in the schedule whether the activity is allowed at the scheduled location.

# 3-4 Solving the optimisation problem

The optimisation problem (3.32)–(3.44) is a MINLP problem with many nonlinear (equality) constraints. Because of these nonlinear constraints generic MINLP problem solvers have difficulty in finding feasible solutions. Therefore, one of the most suitable solvers is a GA, wherein the constraints can be taken into account directly within the fitness function and permutations, crossover, and mutation of the population, so that the population of the genetic algorithm is and remains feasible and the solver does not get stuck in infeasible regions. Furthermore, the search space of the optimisation problem can be greatly reduced by limiting the number of allowed scheduled activities at the nodes and the number of activities in set  $\mathcal{A}$ . By limiting the activities to only measuring and actuating at each node to once per visit (in that order) and ignoring the measurement attraction and energy level equations the minimal travel and completion times of the activities the algorithm from [20, 37] is obtained.

The MaMoMPC algorithm presented in this chapter contains both integer (e.g., paths and activities) and real-valued (e.g., time instants) optimisation variables. All of these optimisation variables together result in a control signal that has some effect on the water levels and flows of the system. The path and activity schedule determine the number of switching instants at which each gate opening can be altered. Furthermore, the control inputs and time instants, respectively, determine the magnitude and timing of the gate openings. In [8, 9, 34], TIO-MPC was used to convert a mixed-integer problem into a real-valued problem. This reduces the computational burden by defining a priori how many switching instants take place and allows the controller to determine the precise time instant to switch a control structure on/off. However, this is not as straightforward for our control algorithm as the number of switching instants of each gate depends on the path and activity schedule. Moreover, at a switching instant a gate is not turned on/off, but opened by a to-be-optimised amount. Furthermore, although the time instants can be defined as real-valued by utilising continuous sampled-data MPC the optimisation problem is still of the mixed-integer type as the path and activity schedule consist of integer optimisation variables.

# 3-5 Conclusions on Multiple-Action Mobile Model Predictive Control

In this chapter, the MaMoMPC algorithm for open water systems with mobile operators have been presented. The operator schedule from previous work [28] has been extended with an activity schedule, which can contain any type of activity depending on the specifications of the open water system.

Unlike previous work, MaMoMPC only requires water level measurements and does not assume that flow controllers are present to keep the flow at each gate constant between actuation times. Moreover, the water level measurements are directly used to update the knowledge about the system. A linearised gate-controlled model has been included to make more accurate predictions about the evolution of the system states.

Next, cyclic control, a heuristic to provide realistic control after the control horizon, has been presented. By including cyclic control the controller has some prediction on what gates the

operator will visit in the future, improving closed-loop performance. A candidate cyclic path has been found that consists of the operator following the water from the head gate to the downstream, adjusting each consecutive gate. Then, a formalisation of cyclic control has been presented, in which some optimisation variables have been introduced.

After that, the information gathering capabilities of the controller have been enhanced by making measurements at certain locations in the open water system more attractive. This has been achieved by keeping track of the time elapsed since the last measurement at each location and multiplying this with some attraction factor that can be adjusted based on the prioritisation of some locations in the system. Three methods to weight the uncertainty of the state of measurement locations in MaMoMPC have been presented.

Then, tracking the energy level of each operator has been proposed. This energy level represents the fatigue of the operator, caused by working on the open water system. The lower the energy level the higher the fatigue of the operator. By allowing breaks for the operator and having the controller schedule them the system performance can be improved, as well as the well-being of the human operators.

Finally, the optimisation problem of MaMoMPC has been presented. The problem is an MINLP problem with nonlinear constraints, which is difficult to solve. Likely some simplifications and assumptions will need to be made to make the algorithm tractable for real-world applications. However, the presented MaMoMPC approach applies to a general class of open water systems and can be simplified based on the specificities of the application.

# Chapter 4

# Case Study: Cyclic Control and Uncertainty Weighing Methods

In this chapter, a case study on the Dez main irrigation canal is performed to analyse some of the features of the proposed Multiple-Action Mobile Model Predictive Control (MaMoMPC) algorithm. The case study is split into two parts: part I (no noise acting on the system) and part II (noise acting on parts of the system). Part I focusses on investigating the effect of cyclic control from Section 3-2-3 on the reference tracking performance. No uncertainty weighing method are added to this first part of the case study. Therefore, adding noise can result in instability as the discrepancy between the predicted states and real system states can become too large. Accordingly, the case study is split in a part with noise and a part without noise. Part II of the case study investigates the effect of including the network uncertainty weighing methods from Section 3-2-4 on the performance of the system. In part II, noise is added to the offtake schedules, offtake flows, measurements, and control actions. Note that the irrigation canal, corresponding model, initial conditions, and offtake schedules are identical for both parts of the case study.

In Section 4-1, the settings of the case study are elaborated on. First, the Dez irrigation canal and its corresponding (internal) model that are used in the case study are presented. Then, the assumptions that are made to simplify the algorithm from Chapter 3 are discussed. Furthermore, the constraints, offtake schedules, and initial (operator) settings of the canal are presented. Next, the solver for the resulting Mixed-Integer Nonlinear Programming (MINLP) optimisation problem is presented. The a posteriori cost functions that are used to compare the performance are then presented along with the statistical methods used for the comparisons. The noiseless part of the case study about cyclic control can be found in Section 4-2 and the second part of the case study about measurement uncertainty in Section 4-3. Finally, conclusions are presented about the settings and simplifications used for MaMoMPC and the case study presented in this chapter.

Master of Science Thesis

# 4-1 Case study and MaMoMPC details

# 4-1-1 Dez main irrigation canal, Iran

The Dez irrigation canal is located in the south-west of Iran, near the city of Dezful. It was designed to transport water from the large dam on the Dez river to the irrigated areas in the north of Khuzestan province. For this thesis, the 45 km long section of the west-main canal is considered, which consists of 13 reaches with a head gate maximum discharge capacity of 157 m<sup>3</sup>/s. All of the information about the hydraulic structures and dimensions of the canal have been acquired from the water authority of the Khuzestan Province [14, 37] and can be found in Appendix B-1.

## **Control structures**

The head gate of the canal can be operated continuously and is assumed to follow the reference flow set by the model predictive controller without error. In reality, the other gates, located at the end of each canal reach, have local flow controllers present. However, for the scope of this thesis, the local flow controllers are removed, as this more closely resembles a rural irrigation canal for which the managing authority of the canal would not have the resources available to either install and maintain local flow controllers or to equip the canal with sensors and actuators (other than the head gate).

The last reach of the Dez main irrigation canal does not have a control gate present. As discussed at the end of Section 2-1-4, this can result in instability problems; due to linearisation errors and process noise. Therefore, the last reach is not considered in the Model Predictive Control (MPC) formulation nor in the case study.

## Mathematical model

The Dez main irrigation canal is modelled by interconnecting the reaches using the nonlinear undershot gate equation (2.3) and the Integrator Delay (ID) modelling approach; the gate parameters, ID parameters, and reach dimensions can be found in Appendix B-1. For the internal model of MaMoMPC the gate equations are linearised, resulting in the linear gate-controlled model from Section 2-1-4.

The nonlinear water dynamics of the irrigation canal are included neither in the internal model nor in the plant model. When an irrigation canal has short or flat reaches and is operated at low discharge rates, resonance waves form and have a substantial influence on the system dynamics. This can result in a mismatch between the model and real system, and the closed-loop system can become unstable. However, the Dez main irrigation canal consists of long and steep reaches and the canal will be operated at medium discharge rates, see Section 4-1-5. Therefore, no actions need to be taken to attenuate possible resonance waves [14]. However, should such action be required several methods are proposed in [35] to deal with the resonance wave problem.

When a water level measurement is received by the central controller, the corresponding water level error state in the internal model is directly updated by the unfiltered measurement. Moreover, at the start of each MPC iteration the gate linearisation coefficients used in the internal model are recomputed using the internal (estimated) water level errors of each reach at the current time step.

## 4-1-2 MaMoMPC settings

The MaMoMPC algorithm presented in the previous chapter is very general. Information about the irrigation canal is assumed to be available to simplify the resulting model predictive controller and to make it more tractable:

- Recharging for operators is not considered in the case study; there are only two possible operator activities on the irrigation canal: measuring and actuating. Moreover, the last reach is not considered as it has no gate (see Section 2-1-4 for the explanation). Therefore, the operator can measure and actuate at every reach in the canal,  $\mathcal{V}_{\text{meas}} = \mathcal{V}_{\text{control}} = \mathcal{V}$ .
- The continuous sampled-data model (3.32)–(3.44) is approximated by the discretetime flow-controlled model from Section 2-1-4 using sampling and control time steps of  $T_{\rm c} = 300$  seconds.
- With each gate visit the operator will first measure the water level, send the measurement to the controller, and receive back the control action to apply at the present gate and the next gate to travel to. Note that if the computational time required for the central controller is small compared to the speed of system dynamics (time it takes for water to flow from reach to reach), this always makes sense. Therefore, the notation presented in Chapter 3 can be simplified. Note that the controller can schedule the operator to measure the same node again. Then, the operator will have to measure again at the designated time and subsequently implement the received control action.

These simplifications result in the MaMoMPC algorithm being reduced to the Time Instant Optimisation Mobile Model Predictive Control (TIO-MoMPC) algorithm from [28] (without the operator workload cost), but with the addition of cyclic control and uncertainty weighing methods.

Moreover, some constraints are set on the input (change in gate opening for gates 2–12, change in discharge rate for the head gate):

- The maximum change in gate opening is set to 20% of the maximum gate opening for each actuation occurrence.
- The maximum and minimum water flow references of the head gate are set to respectively, 157 m<sup>3</sup>/s and 0 m<sup>3</sup>/s.
- The maximum gate opening of gate i (m) is set to:

$$\max\left(\min\left(0.9 \cdot \frac{2}{3} \cdot \frac{h_{1,i}(k) - h_{\mathrm{cr},i}}{\mu_{\mathrm{g},i}}, h_{\mathrm{g},\mathrm{max},\mathrm{i}}\right), h_{\mathrm{g},i}(k)\right)$$
(4.1)

where k is the current time step and i is the gate index. This constraint is based on the constraint formulation from (2.29). Furthermore, the constant 0.9 is to have a 10% margin on the linearisation error of the flow curve, see Section 2-1-4. Moreover, the max function is to ensure feasibility when the current gate opening exceeds the constraint for the next MPC iteration.

• The minimum gate opening of each gate is set to 0 m.

# 4-1-3 Operator settings

Two operators are available to operate the canal. One of the operators starts with measuring at the top of the canal, i.e., at the second gate, while the other one starts measuring at the bottom of the canal, i.e., gate 13. Furthermore, the working areas of both of the operators consist of all of the gates. Moreover, the operators are assumed to travel with a constant speed of 30 km/h between all locations. There are two main arguments for choosing two operators:

- The first reason is to investigate the closed-loop scheduling behaviour when multiple operators are available that can travel to every position in the canal. In [37], the operator working areas are split. Therefore, it is interesting to investigate what the operator paths will look like when the operators have no restrictions on their working area.
- The size of the search space increases rapidly with the number of operators. If the search space is very large, it may be difficult for the MINLP solver to find close to optimal results within the allocated CPU time. Therefore, the number of operators will need to be limited.

# 4-1-4 Optimisation problem solver

To solve the MINLP problem a hierarchical two-layer approach is used as shown in Figure 4.1. The top layer consists of a mixed-integer Genetic Algorithm (GA) [24] in which the genomes of the individuals encode the path schedule of each operator and the corresponding time instants of the measuring and actuation activities. First, a population is created that satisfies the constraints (3.33)-(3.44), see Algorithm 1. Then, the fitness of each individual is calculated by evaluating the fitness function for each genome in the population. With the path and corresponding actuation instants fixed by the genome, a Quadratic Programming (QP) problem is solved to find the optimal inputs for the actuation instants. Using this optimal control signal the water level errors of the system are predicted and a water level error and input cost are calculated. Moreover, by keeping track of the time elapsed since the last measurement the network uncertainty cost can be calculated as well. Finally, the cost on input, water level errors, and network uncertainty is weighed and summed for each individual and communicated to the GA solver. The GA utilises these fitness values of the population to create a new population, wherein the genome of some individuals is randomly mutated and new children are made by combining the genomes of the parents from the previous population. The fitness of new and mutated individuals is then calculated and the population for the next generation is created. This is repeated until some stopping condition is reached.

All of the simulations are performed using MATLAB R2017b with IBM ILOG CPLEX Optimisation Studio V12.7.1 to solve the QP problems on a 64-bit Linux computer with 64GB RAM and eight Intel Xeon CPU-E5-1620 v3 @3.50GHz processors. Note that MATLAB has its own QP solver. However, the CPLEX QP solver is approximately 4 times faster on average than MATLAB's QP solver for a set of 10000 representative problems, see Figure C.2 in Appendix C.


Figure 4.1: Schematic of the GA approach used to solve the optimisation problem.

The standard integer GA solver from MATLAB requires a lot of computational resources to create a feasible initial population, due to the many nonlinear (equality) constraints. Moreover, when creating new populations by use of mutation and crossover functions the constraints are often violated; resulting over time in populations consisting of mostly infeasible individuals. Therefore, a custom population creation, mutation, and crossover function has been written in MATLAB, see Appendix D.

In the case study, the GA stopping conditions are: lowest cost did not decrease in five subsequent generations or the CPU time exceeds 200 seconds. The 200 seconds is chosen to evaluate a large part of the search space, while still being able to run enough simulations during the thesis to have significance in the results. Furthermore, 200 seconds is less than the sampling time. Therefore, 200 seconds is a realistic time in which the solver needs to solve the MINLP problem in the real implementation. If the lowest cost did not decrease over five generations, the GA is restarted with the top 10% of the previous GA population and the rest of the population is newly generated using the population creation function. Moreover, the time that was spent in the previous GA iteration is subtracted from the 200 seconds CPU time limit.

### 4-1-5 Offtake schedules

At the start of the simulation the head gate supplies a water flow of  $76.325 \text{ m}^3/\text{s}$  to the canal. This means the canal initially runs at 48% of its maximum flow capacity. Therefore, the medium-flow ID model parameters from Table B-2 are used for the case study.

At the end of every reach an offtake structure is present that withdraws water at a certain discharge rate. A disturbance is added on all these offtake flows after five hours, see Figure 4.2. This offtake schedule is representative for the Dez main irrigation canal and is retrieved from [28, 37]. After five hours the canal runs at 58% of the maximum discharge rate capacity. Therefore, the ID model parameters for medium flow are representative for the whole simulation.



Figure 4.2: Offtake profile used in the case study.

#### 4-1-6 Method used for comparison

To compare the performance of the approaches in the case study, the controllers run several simulations in which the system is disturbed by the offtake schedule from Section 4-1-5. The simulations of part I and part II of the case study are run, respectively, 10 times and 15 times. Moreover, the performance is quantified by defining an a posteriori cost for the closed-loop simulation result. This a posteriori reference tracking cost is the same as the discrete-time version of objective function (3.10) applied over the duration of simulation:

$$J_{\mathrm{xu}} = \sum_{k=1}^{N_{\mathrm{f}}} (\mathbf{x}^{T}(k)\mathbf{Q}\mathbf{x}(k) + \mathbf{u}^{T}(k)\mathbf{R}\mathbf{u}(k))$$
(4.2)

where  $N_{\rm f} = 288$  indicates the total number of simulation steps, corresponding to 24 hours for the case study. The weighing matrix **Q** is set to 100 for the diagonal indices corresponding to water level errors; the rest of the entries of **Q** are zero. Furthermore, the weighing matrix **R** is set to 0.01 for the diagonal indices corresponding to the weight on the flow change of the head gate and set to 0.1 on the other diagonal indices, which correspond to the weight on gate opening changes; the rest of the entries of **R** are zero.

A statistical test is performed to compare the a posteriori costs of the different approaches. The tests of the case study will have a small sample size of 10 to 15 simulation results. Moreover, it is unknown and difficult to guess beforehand how the results of the simulations will be distributed. Therefore, it is decided to perform a two-sample Welch test [43]. This test is used to test the hypothesis that two populations have equal means or, alternatively, that one of the population means is greater than or equal to the other. Moreover, it is a version of the student's t-test that assumes that the two populations are normally distributed with unequal variances and unequal sample sizes [43]. However, the two-sample Welch test can deal with small sample sizes and has robustness properties against non-normality of the distribution of the data [12, 13, 27]. The null hypothesis used for the case study is that the a posteriori performance of one of the methods has a mean greater than or equal to the mean

of the a posteriori performance of one of the other methods. Furthermore, the one-tailed significance level has been set to 0.05.

## 4-2 Case study part I: cyclic control

In Section 3-2-3, cyclic control has been proposed to improve the setpoint tracking performance of MaMoMPC. There are many optimisation variables for the cyclic path of each operator that need to be optimised during each MPC iteration. This will result in a very large search space, which requires long solving times for the GA solver to converge. If the search space is disproportionately large relative to the allowed solving time, the uncertainty on the solution found by the stochastic GA is large too, e.g., if you let GA optimise the exact same scenario 10 times you can get 10 completely different performances. That is problematic for the case study as the goal is to compare the performance of the methods for a certain scenario within a reasonable simulation time. Therefore, all of the optimisation variables of the cyclic path are fixed beforehand so the GA solver only has to evaluate operator schedules. The performance of the MaMoMPC gate-controlled controller with cyclic control, but without uncertainty weighing methods, is compared to the performance of the TIO-MoMPC method from [28] (the internal model of [28] is changed to a gate-controlled equivalent).

The MaMoMPC and TIO-MoMPC settings for this case study are:  $N_{\rm p} = 84$ ,  $N_{\rm c,schedule} = 36$ ,  $N_{\rm s} = 8$  gates,  $N_{\rm limit} = 12$  gates,  $T_{\rm o} = 0$  seconds,  $T_{\rm c} = 300$  seconds. The number of gates in the operator schedule  $N_{\rm s} = 8$  is chosen such that controller is likely to consider all of the gates (12 in total) within the variable control window (two operators corresponds to a maximum of 16 gate visits within the variable control window). The variable control horizon  $N_{\rm c,schedule} = 36$  corresponds to 3 hours and is chosen such that it exceeds the maximum travelling time between nodes (21 time steps), while still having a margin for the controller to schedule a few other gate visits. Moreover, the variable control horizon is chosen large enough to fit many of the possible path schedules consisting of  $N_{\rm s} = 8$  gates (58.1% of them to be exact). The prediction horizon,  $N_{\rm p} = 84$ , corresponds to 7 hours and is chosen such that two complete downstream paths can be completed in the fixed control horizon, see Figure 4.3. Finally, a sampling and control time step of 300 seconds are typical for the Dez main irrigation canal, see [14, 28, 37].

#### 4-2-1 Fixing the cyclic paths

Each of the operators performs one cyclic path after the variable control horizon. Furthermore, both of these cyclic paths consist of visiting all of the 12 reaches and the arrival and subsequent actuation at the gates is synchronised with the incoming flow if possible (note that this is not possible if it takes the operator longer to travel to the location than it takes for the water to flow there). Moreover, the start of the cyclic path of the first operator is delayed by the time it takes for the water to flow from the head gate to gate 2. Furthermore, the second operator starts when the first operator is at approximately half of its cyclic route. The cyclic paths the operators take after the variable control horizon are depicted in Figure 4.3. Note that this fixing of the cyclic paths will result in some loss of optimality. However, it still provides the controller with a prediction of what routes the operators travel after the variable control horizon, while not introducing much extra computational burden.



**Figure 4.3:** The operator routes after the variable control horizon are estimated by the fixed paths depicted in the fixed control window. Furthermore, the activation time step  $k_a$  is the discrete-time variant of the activation time  $t_a$ .

#### 4-2-2 Simulation issues

As mentioned throughout the chapters: instability can occur when the discrepancy between the predicted states and the actual systems' states becomes too big. In this first part of the case study, the internal model is the linearised gate-controlled model and the connected 'real model', i.e., the plant, an ID model with nonlinear flow equation (2.3) to interconnect the reaches. Therefore, there will be some discrepancy between the internal model and the plant (after actuation). As long as the algorithm visits every reach frequently this is not necessarily an issue, as the corresponding model state are updated by the measurements and the controller schedules a change in the gate flows accordingly. Moreover, in the offtake schedule a flow change is scheduled for each offtake at the end of the reaches, so the controller will eventually schedule an operator to change gate settings at those locations at least once. However, after 10-12 hours in the simulation, the internal model error states of some reaches are zero and thus the controller will no longer schedule operator visits for those reaches. If there is a flow mismatch at a gate that is no longer visited, the discrepancy between predicted and actual states will keep growing over time, resulting in instability.

To still be able to use the simulations for comparison, the a posteriori cost (4.2) is only evaluated for 10 hours ( $N_{\rm f} = 120$ ), in which no instability occurred in any of the simulations. Moreover, at the 10 hour mark the steady state errors were close to zero. Consequently, the first 10 hours of the simulations are representative of the controllers' performances.

#### 4-2-3 Results

The simulations have been run 10 times for both of the methods: MaMoMPC with cyclic control and TIO-MoMPC, see the beeswarm plot (point distributions where horizontal jitter has been added to the data points to avoid overlap) in Figure 4.4. Outliers have been defined

as data points outside the  $\pm 2.7$  times standard deviation interval, i.e., 99.3 percent coverage, if the data are normally distributed. Furthermore, the outliers have been omitted from the statistical tests.



**Figure 4.4:** Beeswarm plot of the a posteriori reference tracking cost  $J_{xu}$  obtained from 10 simulations of each of the approach.

From the beeswarm plot in Figure 4.4 it becomes clear that the a posteriori cost of TIO-MoMPC has a larger spread than that of MaMoMPC. Moreover, note that four of the simulation results were identical for MaMoMPC.

To compare the means of the two methods a two-sample Welch test is used [43]. The null hypothesis used is that the mean of MaMoMPC is greater than or equal to that of TIO-MoMPC. The test statistic t is computed as:

$$t = \frac{\bar{g}_1 - \bar{g}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} = -4.3762,$$
(4.3)

where  $\bar{g}_i$  is the mean,  $s_i$  is the standard deviation, and  $n_i$  is the sample size of group  $i \in \{1, 2\}$  (group 1 corresponds to MaMoMPC and group 2 to TIO-MoMPC). Next, the number of degrees of freedom are calculated:

$$df = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{1}{n_1 - 1}\left(\frac{s_1^2}{n_1}\right)^2 + \frac{1}{n_2 - 1}\left(\frac{s_2^2}{n_2}\right)^2} = 9.7373.$$
(4.4)

Finally, using the student's t cumulative distribution function, the test statistic, and the degrees of freedom the (one-tailed) statistical significance is calculated as  $7.38 \times 10^{-4} < 0.05$ . Therefore, we can reject the null hypothesis that the mean of MaMoMPC is greater than that of TIO-MoMPC. In other words, the mean a posteriori cost of MaMoMPC with cyclic control is smaller than that of TIO-MoMPC with statistical significance. Furthermore, the average a posteriori cost for MaMoMPC with cyclic control was approximately 3.5 times lower than that of TIO-MoMPC.

The settings and algorithm of MaMoMPC and TIO-MoMPC for this first part of the case study are identical, except for the inclusion of cyclic control. Therefore, we can conclude that

Master of Science Thesis

including this particular cyclic path improves the a posteriori reference tracking performance with statistical significance. Nonetheless, the performance is evaluated on a single disturbance scenario, while using certain MPC settings and a heuristic solving approach, i.e., a GA, which is given limited CPU time. Note that the offtake schedule used this case study is also used in previous literature [28, 37] for the same canal. However, to verify our results the algorithms (and cyclic path) should be evaluated using other disturbance scenarios and different irrigation canals.

As the simulations use a single disturbance set the spread of the results is due to the stochastic nature of the GA solver. Therefore, a valid question is whether the difference in performance of the two methods can be attributed to the performance of the solver (whether the solver is biased towards one of the methods). The GA solver is able to evaluate the cost of approximately 40000 and 55000 solutions to the MINLP problem for MaMoMPC and TIO-MoMPC, respectively, during the 200 seconds of CPU time. This difference in the number of cost evaluations can be attributed to the enlarged QP problem when including cyclic control, see Appendix C. Accordingly, even though more solutions to the MINLP problem are evaluated for TIO-MoMPC, MaMoMPC performs better. Note that it is intractable to go through all the possible operator schedules (approximately 10<sup>18</sup>). Therefore, it is difficult if not impossible to say how much of the performance difference is due to the difference in method and how much can be attributed to the behaviour of the solver.

## 4-3 Case study part II: network uncertainty minimisation

In Section 3-2-4, three network uncertainty weighing methods are proposed to decrease the discrepancy between the estimated water levels (from the internal model) and the actual water level of the reaches in the canal. In this case study, the proposed uncertainty weighing methods are combined with the cyclic control MaMoMPC approach from Section 4-1-2 to create four uncertainty weighing approaches:

- Method A: MaMoMPC with static measurement attraction using the time elapsed since the last measurement, where each measurement node is equally attractive. This is the most simple, measurement attraction method. If no knowledge is available about the disturbances acting on the system, it makes sense to just set the attraction factors equal for all locations. Therefore, the attraction factor  $\alpha_{loc}$  is set equal to 1 for all reaches.
- Method B: MaMoMPC with static measurement attraction using the time elapsed since the last measurement, where measurement nodes can have different attraction factors. For this method the knowledge about the system, external disturbances, and offtake schedule is used to assign an individual  $\alpha_{\text{loc},i}$  for each reach *i*. The offtake flow and the offtake flow change at reach 4 is much bigger than that at the other reaches. This means that gate 5 will have to be adjusted more drastically than the other gates, which increases the linearisation errors as was discussed in Section 2-1-4. Moreover, the standard deviation of the noise on the offtake flow at reach 4 is bigger than that of the noise at the other reaches. However, these extra uncertainties do not only influence reach 4, but also the surrounding reaches. The upstream gate at reach 3 is used to offset the water level errors at reach 4 and the linearisation errors at gate 5 will result in water level errors at reach 5. Based on this knowledge, the update frequency of reaches 3, 4, and 5 is

set 50% higher than that of the other reaches. Note that the  $\alpha_{\text{loc}}$  values are normalised, so that  $\sum_{i=1}^{12} \alpha_{\text{loc},i} = 12$  for both methods A and B; in order to not alter the weight on the time elapsed since the last measurement (3.22) relative to the weight on input and errors (3.10). Furthermore, a soft constraint (3.26) is added on the update frequency:

$$T_{i}^{\text{elapsed}}(k) \leq \begin{cases} 6000 \text{ seconds} & \text{if } i \in \{3, 4, 5\}, \\ 9000 \text{ seconds} & \text{else}, \end{cases}$$
for  $k \in \{k_{\text{a}}, k_{\text{a}} + 1, \dots, k_{\text{a}} + N_{\text{c,schedule}}\}.$ 

$$(4.5)$$

- Method C: MaMoMPC in which the path schedule is fixed to a downstream cyclic path to ensure a certain update frequency at each gate. The optimiser still has the freedom to optimise measuring and actuation instants to improve system performance. In this method, the path  $p_j(t)$  is defined beforehand, as the downstream path of visiting all the gates subsequently starting from gate 2 and ending at gate 12. However, the measuring and actuation time instants are still optimisation variables. This gives the controller the freedom to still delay measurements and actuations to synchronise better with incoming flows or scheduled changes in offtakes. However, the delays are detrimental to the measurement frequency. Therefore, the measurement attractiveness cost (3.22) is still included in the cost function of this method. Although the operator paths are known beforehand for the whole prediction horizon, the location visits that exceed the variable control horizon are removed from the optimisation problem. This allows for a more fair comparison with the other methods.
- Method D: a version of MaMoMPC in which the path and time instants are fixed based on a downstream path beforehand. Method D restricts the freedom of the controller even more than method C by freezing the measurement and actuation time instants to the minimal time required that follows from the travel time between locations. This results in fixed paths, fixed time instants, and fixed activity schedules; so the toplayer GA solver has nothing to optimise. The only optimisation required is that of the control inputs the operators have to implement at the locations in the fixed path. To compute the optimal control inputs a QP problem needs to be solved, which can be done efficiently. Therefore, the required solving time for this solver is negligible compared to the other methods. Note that because a single disturbance set is used for all of the simulations and no stochastic solver is involved in method D, this method only requires a single simulation (the solver is deterministic).

These four methods are compared to each other and to method E: MaMoMPC with cyclic control from Section 4-1-2 without measurement attraction methods (so the controller solely schedules operators to track the water level setpoints). However, as uncertainties act on several parts of the system the predicted water levels can diverge from the actual water levels over time. If this results in a negative water level or too high water level (higher than the maximum gate opening), the nonlinear undershot flow equation from (2.3) no longer holds and the simulation will be terminated. Method E is used to evaluate the system performance when no measurement attraction methods are included, while the system is subjected to external disturbances.

The MaMoMPC and TIO-MoMPC settings for this case study are:  $N_{\rm p} = 84$ ,  $N_{\rm c,schedule} = 36$ ,  $N_{\rm s} = 8$  gates,  $N_{\rm limit} = 12$  gates,  $T_{\rm o} = 0$  seconds,  $T_{\rm c} = 300$  seconds. The number of gates

in the operator schedule  $N_{\rm s} = 8$  is chosen such that controller is likely to consider all of the gates (12 in total) within the variable control window (two operators corresponds to a maximum of 16 gate visits within the variable control window). The variable control horizon  $N_{\rm c,schedule} = 36$  corresponds to 3 hours and is chosen such that it exceeds the maximum travelling time between nodes (21 time steps), while still having a margin for the controller to schedule a few other gate visits. Moreover, the variable control horizon is chosen such that more than half of the possible operator paths (58.1% of the possible paths to be exact) fit in variable control window (assuming no delays are added). The prediction horizon,  $N_{\rm p} = 84$ , corresponds to 7 hours and is chosen such that two complete downstream paths can be completed in the fixed control horizon, see Figure 4.3. Finally, a sampling and control time step of 300 seconds are typical for the Dez main irrigation canal, see [14, 28, 37].

#### 4-3-1 Process noise

The reason for introducing uncertainty weighing methods is to decrease the uncertainty on the water level predictions, thus improving closed-loop reference tracking. Therefore, to evaluate whether the proposed methods actually reduce the uncertainty, some noise is inserted in the simulations. Zero-mean Gaussian noise has been added to the water-level measurements, control gate positions (when an operator changes the settings), and timing of implementing the offtake flow changes with a standard deviation of 0.01 m, 0.001 m, and 900 seconds, respectively. The same noise characteristics are also used in the case study of [37]. Moreover, zero-mean Gaussian noise is added to the offtake flows:  $0.005 \text{ m}^3/\text{s}$  on the offtake of reach 4 and  $0.001 \text{ m}^3/\text{s}$  on all the other offtakes. The uncertainty on the offtake of reach 4 is set higher, because it has a significantly higher offtake flow than the other reaches. To reduce the uncertainty on the simulation results exactly the same disturbance set is used for each simulation.

#### 4-3-2 Measurement frequency cost

The a posteriori reference tracking cost from (4.2) does not provide insight into the controllers succession in measuring frequently at the reaches. Therefore, a second a posteriori cost function is created. This second a posteriori cost function is the same as the discrete-time version of objective function (3.22) applied over the duration of simulation:

$$J_{\rm mf} = \sum_{i=1}^{12} \sum_{k=1}^{N_{\rm f}} (\alpha_{{\rm loc},i} T_i^{\rm elapsed}(k))$$
(4.6)

where  $N_{\rm f}$  (set to 288) indicates the total number of simulation steps, corresponding to 24 hours for the case study.

#### 4-3-3 Results

The simulations have been run 15 times for methods A, B, C, and E. Method D only requires one simulation as the outcome is deterministic (GA is not used and a single disturbance set is used for each simulation). The resulting a posteriori reference tracking cost  $J_{xu}$  has been depicted in the logarithmic beeswarm plot in Figure 4.5. Furthermore, the resulting a posteriori measurement frequency cost  $J_{\rm mf}$  has been depicted in the beeswarm plot in Figure 4.6. Note that five of the simulations of method E were terminated, due to negative water levels or too high water levels. These five simulations have been omitted from the statistical tests, as well as the beeswarm plots. Outliers have been defined as data points outside the  $\pm 2.7$  times standard deviation interval, i.e., 99.3 percent coverage, if the data are normally distributed. Furthermore, the outliers are omitted from the statistical tests.



**Figure 4.5:** Logarithmic beeswarm plot of the a posteriori reference tracking cost  $J_{xu}$  obtained from 15 simulations of the five methods.



**Figure 4.6:** Beeswarm plot of the a posteriori measurement frequency cost  $J_{mf}$  obtained from 15 simulations of the five methods.

As method D produces deterministic results no statistical test is performed to compare it to other methods. Nonetheless, both of the a posteriori costs of method D are lower than the means of the a posteriori costs of all the other methods. Furthermore, only one simulation of method A resulted in a lower  $J_{\rm xu}$  cost and method D had the lowest  $J_{\rm mf}$  cost of all the simulations performed. The a posteriori costs of each of the methods (other than method D) have been compared to that of the other ones using the two-sample Welch test [43]. The resulting probabilities of accepting the null hypothesis (one of the methods in the most left column has a greater mean a posteriori cost than one of the methods in the top row) for  $J_{\rm xu}$  and  $J_{\rm mf}$  can, respectively, be found in Tables 4-1 and 4-2.

All of the methods that included an uncertainty weighing method were able to track the

**Table 4-1:** Probabilities of accepting the null hypothesis for each method pair for the a posteriori cost  $J_{xu}$ . Note the symmetry along the diagonal: changing the order of the comparison affects the sign of the results, but does not affect the magnitude of the results.

Method	А	В	С	Е
А	_	0.0023	0.9027	0.0016
В	0.9977	_	0.9985	0.0029
С	0.0973	0.0015	—	0.0015
Е	0.9984	0.9971	0.9985	_

**Table 4-2:** Probabilities of accepting the null hypothesis for each method pair for the a posteriori cost  $J_{mf}$ . Note the symmetry along the diagonal: changing the order of the comparison affects the sign of the results, but does not affect the magnitude of the results.

Method	А	В	С	E
А	_	$6.73 \times 10^{-12}$	1.00	$4.80 \times 10^{-4}$
В	1.00	_	1.00	$1.40 \times 10^{-3}$
С	$4.85 \times 10^{-5}$	$7.68 \times 10^{-13}$	_	$3.33 \times 10^{-4}$
E	1.00	1.00	1.00	_

water level setpoints without destabilising the system. In five of the simulations, in which the system is controlled by method E, instability occurred and the simulations were terminated. This emphasises the importance of including some sort of uncertainty weighing method to control the stochastic system to the setpoints.

Method A performed significantly (significance level 0.0023 < 0.05) better on both a posteriori costs than method B. An explanation is that the choice of attraction factors was suboptimal in method B; the system performance decreased because some parts of the open water system were not measured and actuated frequently enough. However, the assigned attraction factors and thresholds used in method B did not have the desired effect: reaches 3, 4, and 5 were only measured  $30.42 \pm 6.7\%$  more often than the other reaches over all 15 simulations (so not 50% more, as was intended). This is likely because the controller decided to delay measuring at reaches 3, 4, and 5 to reduce the water level error at one of the other reaches. These results show that the system is sensitive to the choice of local attraction factors and makes a trade-off between measuring and reference tracking. Moreover, the total number of measurements per simulation was about 8% (12 measurements) lower for method B compared to method A (significance level 0.001 < 0.05). This suggests that increasing the measurement frequency at reaches 3, 4, and 5 comes at the cost of less efficient operator routing.

For both of the a posteriori costs method C results in a significantly better performance than method B (significance level 0.0015 < 0.05 for  $J_{xu}$  and significance level  $7.68 \times 10^{-13} < 0.05$ for  $J_{mf}$ ). Moreover, method C has a significantly lower mean a posteriori measurement frequency cost than method B (significance level  $4.85 \times 10^{-5} < 0.05$ ), but not a lower mean a posteriori reference tracking cost (significance level  $0.0973 \leq 0.05$ ). An explanation is that the cyclic path used is the same path as the one that method C schedules for the operators (although the algorithm can decide to delay measurement and actuation activities). Therefore, the prediction of the control actions in the fixed control window is more accurate for method C than for methods A and B, explaining the difference in performance. To verify this several simulations using different cyclic control paths could be performed.

The most heuristic method, method D, had the lowest a posteriori costs of all the methods. This is unexpected as the controller has more possibilities/freedom to reduce the open-loop cost through tuning the operators' schedules in the other methods. We provide four possible explanations why the most heuristic method performs the best:

- The variable control horizon is set to small relative to the prediction horizon. When examining the open-loop costs in a similar way to the a posteriori costs from (4.2)and (4.6) the open-loop costs found during simulations of method D are indeed higher than that of the other methods. Therefore, although the open-loop cost improves by allowing the controller to decide on more parameters of the schedule, the closed-loop (a posteriori) cost is worse. We believe that this is due to the open-loop cost not being representative of the system performance, as a result of the MPC settings. Methods A, B, C, and E minimise the open-loop cost by tuning the operator schedule during the variable control window. If the variable control horizon is chosen too short (relative to the prediction horizon), the controllers will tend towards selecting operator schedules that focus on reducing the biggest errors during the variable control window as quickly as possible to minimise the accumulating water level error cost over the longer prediction window. This results in suboptimal 'aggressive' control, wherein the controller wrongfully assumes that control is no longer possible after the variable control horizon. This would explain why the open-loop cost of method D is worse than the other methods while its closed-loop performance is better.
- Cyclic control is included in all of the methods in part II of the case study to provide the controllers with some estimate of the future control actions. Nonetheless, this estimate is the most accurate for method D, as that method actually schedules the cyclic operator path. Methods A, B, and E, do not actually schedule cyclic paths in the closed-loop, due to the moving horizon approach of MPC. However, in method D the operators are forced to always take the same route (identical to the cyclic path). Therefore, the cyclic control that is used is a much better estimate for method D than for any of the other methods, explaining the difference in performance. To verify this several simulations using different cyclic control paths could be performed.
- The introduction of the delays decreases the measurement frequency, which increases the network uncertainty. This explanation is related to the limitations of the variable control window and number of gates to schedule for each operator during this time window. Methods A, B, C, and E optimise the operator schedule during the variable control window; in this operator schedule some delays can be added to the actuation and measurement activities to synchronise gate openings with the system dynamics to improve (open-loop) performance. However, each scheduled delay will result in the measurement frequency of the locations going down compared to method D (which has no delays). Moreover, the uncertainty about the states of the system grows over time. Therefore, methods A, B, C, and E will have a larger discrepancy between the predicted and actual systems' states decreasing the system performance compared to method D. To verify this explanation, the possibility of delaying actuation and measurement activities could be removed from the methods.

• A final explanation is that the solver is unable to find adequate solutions for methods A, B, C, and E. However, seeing that the open-loop cost is already lower for all of the methods (compared to method D) this seems unlikely. Nonetheless, this could be verified by repeating the simulations with more allocated CPU time for each MPC iteration and monitoring the open-loop costs and the a posteriori costs.

These explanations or a combination of them provide some perspective on why method D performs the best. More research and simulations are needed to gain a better understanding of the performances and (dis)advantages of the various uncertainty weighing methods proposed.

# 4-4 Conclusions on Case Study: Cyclic Control and Uncertainty Weighing Methods

In this chapter, the MaMoMPC algorithm has been adjusted for the Dez main irrigation canal in Iran. The continuous sampled-data model has been approximated by the discretetime flow-controlled model from the Chapter 3 using a sampling and control time step of 300 seconds. Moreover, some assumptions and simplifications have been made on the order of activities, types of activities, and number of operators. However, energy recharging has not been considered in this chapter. A custom GA solver has been used to find solutions to the MINLP problem. Furthermore, the performance of the algorithms has been tested by using a representative offtake schedule from previous literature [28, 37] and comparing the algorithm performance on two a posteriori cost functions. A two-part case study has been presented.

The first part of the case study focussed on the effect on the reference tracking performance of including cyclic control in MaMoMPC and compared the performance to that of TIO-MoMPC. Accordingly, the cyclic path the operator takes is fixed and not subject to optimisation. This has been done in order to reduce the uncertainty on the results, by reducing the search space of the MINLP problem. The results show that MaMoMPC with cyclic control has statistically significant better performance (significance level  $7.38 \times 10^{-4} < 0.05$ ) than TIO-MoMPC for the deterministic offtake schedule scenario of this case study. Note that the scheduled paths of the operators match the downstream path computed in Section 3-2-3, which supports the use of this downstream path as cyclic path.

In part II of the case study, four methods to weigh the measurement uncertainty have been compared to each other and to MaMoMPC without such measurement uncertainty method. For this second part of the case study, a second a posteriori cost has been introduced to weigh the measurement frequency performance. Moreover, process noise has been introduced to have a more realistic scenario and to compare the performance of the different uncertainty weighing methods. The results show that all of the uncertainty weighing methods proposed have better measurement frequency and reference tracking performance than not including such a method (significance level  $5.30 \times 10^{-4} < 0.05$ ). Moreover, the choice of the measurement attraction factor  $\alpha_{\rm loc}$  has a large influence on the system performance. The best performing method is a heuristic that schedules the operators to visit all of the reaches subsequently using the downstream path and minimal travelling times. Some explanations are given as to why the heuristic method outperforms the methods that provide the controller with more freedom.

Chapter 5

# **Conclusions and Recommendations**

In this chapter, the conclusions and recommendations derived from the research presented in this thesis are given.

# 5-1 Conclusions

This thesis extends the work in [20, 28, 37] by proposing a more general algorithm, called Multiple-Action Mobile Model Predictive Control (MaMoMPC), see Chapter 3. In MaMoMPC, activity schedules, cyclic control, uncertainty weighing methods, and energy levels have been proposed to efficiently extend the control scheme from [28] and prioritise measuring at certain locations of the open water system, while taking into account the breaks of human operators. Moreover, MaMoMPC takes a more realistic approach towards modelling the open water systems, as operators only measure the water levels of the reaches and no local flow controllers are assumed to be available at each gate to maintain the flow at a reference value set by the human operator. The MaMoMPC optimisation problem in its most general form is very difficult to solve due to the many integer optimisation variables and nonlinear constraints. The optimisation problem can be reduced using knowledge about the system it is applied to; e.g., for some irrigation canals there may be only two activities: measuring and actuation. Therefore, the usefulness of MaMoMPC mostly comes from its ability to reduce the control problem to fit the characteristics of the open water system it is applied to.

The MaMoMPC algorithm has been evaluated on a case study of the Dez main canal. In the first part of the case study, the effect of including cyclic control has been tested. The controller with cyclic control has an a posteriori cost that is  $3.5 \times$  lower than that of the same controller without cyclic control. In the second part of the case study, the uncertainty weighing methods are tested on simulations with stochastic disturbances. The results show that including the uncertainty weighing methods significantly improves the a posteriori cost, by measuring frequently at all locations in the canal the prediction of the internal model becomes more accurate and as a result, the reference tracking performance of the system improves.

Master of Science Thesis

By keeping track of the energy levels of the operators the controller is able to account for the breaks of the operators and adjust the operator schedules accordingly. This is an important aspect of the real setting in which humans will require breaks (by law). Moreover, the energy level framework presented in Section 3-2-5 can be used to keep track of the charge or remaining fuel of the vehicles that the operators use to travel to the different locations. This gives the controller the possibility to schedule recharging/refuelling activities for the vehicles while taking into account system performance. The energy level framework is not implemented in the case study.

## 5-2 Recommendations

Although the results of MaMoMPC are promising, various challenges remain. To begin with, humans are modelled solely as delays in the control problem; so more human-related aspects, such as inaccuracies in the following and implementation of the orders received from the central controller, will need be added to the control design. Moreover, the resulting impact of having this uncertainty about the times at which the human operators actually implement their activities on the performance and stability is an important issue that needs to be considered. To verify the performance of cyclic control more research is needed, in which cyclic control is tested on more realistic (dynamical) offtake schedules with realistic external disturbances. Furthermore, the uncertainty weighing methods proposed in this thesis were very basic and assigning attraction factors to indicate which location are more important to visit than others is non-trivial and requires more research. In the next sections, recommendations are given on a few other areas of research that are worth investigating in future work.

### 5-2-1 Are time delays worth the added complexity?

In [28], it was proposed to introduce time delays to the measuring and actuation activities to allow the controller to synchronise opening the gates with the system dynamics. However, the superiority of the algorithm from [28] over that of [20, 37] was shown using a noiseless scenario in which the plant and the internal model were identical. In this thesis, stochastic disturbances have been introduced to the system, which caused the predicted states to drift quickly from the actual states. Therefore, not measuring for a prolonged period of time will harm the system performance and may result in constraint violations. Does the increase of performance through synchronisation with the system dynamics outweigh the decreased frequency of measuring and actuating of the system? Looking at the results from Section 4-3-3, the heuristic method without time delays had the lowest a posteriori costs, suggesting that by removing the delays the performance increases. Note that by removing the time delays from the optimisation problem, the size of the search space can be decreased significantly, as well as the number of nonlinear constraints. However, synchronisation of the opening of the gates with system dynamics may be very important for the real system. Therefore, simulations need to be performed on more realistic mathematical models to test whether time delays improve the system performance significantly.

### 5-2-2 Is measuring only water levels enough?

The MaMoMPC algorithm uses solely measurements of the water levels to control the system. In the case study, this information has been shown to be sufficient to track the water level setpoints. However, this will have to be evaluated on more realistic models of the irrigation canal, such as a nonlinear SOBEK [10] model that uses the De Saint-Venant equations to model the open water system. Moreover, the effect of meteorological disturbances, such as rainfall, on the system performance and stability will have to evaluated.

Note that whenever the height of a gate is changed by an operator (including some operator error), the linearised coefficients of that gate in the internal model are updated using the actual gate height; this is done in order to prevent opening the gate past its limits and in order to reduce the linearisation error. In reality, the exact gate height is not known until an operator measures it and communicates it to the central controller. The effect this assumption has on the performance (and feasibility) of the system must be investigated.

If water level measurements are not sufficient to control the open water system, we recommend investigating the effect of including measurements of the local water flow. The nonlinear undershot gate flow equation (2.3) is linearised for use in the linear internal model. Therefore, when the height of a gate is changed there is some error between the expected flow of the internal model and the actual flow of the plant. Moreover, external disturbances will act on the system that affect the water flow, e.g., rainfall and unexpected changes in offtake flows. By having more accurate estimates/measurements of the water flow available, the prediction of the water levels is more accurate. Additionally, it is interesting to investigate whether measuring the flows yields enough information to regulate the water level of the last reach (where no gate is present) to the setpoint.

In this thesis, we have assumed the standard deviation of the (measurement) noise acting on the system to be relatively small, allowing us to use the operator measurements directly to update the predicted systems' state. However, if the small noise assumption does not hold, an observer can be built to estimate the system states. Nonetheless, building such an observer is challenging in this case, as it must be able to deal with fusing measurements coming from a subset of the operators in the open water system, that have been obtained at unevenly spaced sampling steps related to operators arriving at the gates.

### 5-2-3 How to improve the reliability of the controller?

Another challenging topic is the feasibility of solving the optimisation problem for certain realtime applications. In the case study, the a posteriori performance results had some spread on them, which was to be expected as the Genetic Algorithm (GA) solver uses a stochastic approach to sample solutions. However, the magnitude of the spread on results is likely due to the size and complexity of the search space being too large to adequately sample within the allocated solving time of the algorithm. Assigning more time to the controller will help, but may not be possible in the real-time control setting of the open water system. Therefore, it is interesting to investigate what happens to the spread of the results when the search space complexity and size are reduced. We recommend investigating the effect on the (spread of) performance when reducing the size of the search space by adding

- constraints on paths. As observed in the case study in Section 3-2-3 and throughout experiments, the controllers tend to schedule the operators to travel (partial) downstream paths. These paths allow for synchronisation of the opening of the gates with the arrival of the flow, and for efficiently adjusting the necessary gates. With this knowledge, the size of the search space can be reduced significantly, by imposing some rules on what is a 'valid' path. For example, a valid path could be defined as one that contains at least a partial downstream path of three locations. In that case,  $4 \rightarrow 3 \rightarrow 2 \rightarrow 12$  is valid, but  $3 \rightarrow 5 \rightarrow 1 \rightarrow 6$  is not. By adding these rules on what is a valid path the size of the search space can be reduced without overly affecting the control performance.
- The size of the search space can also by reduced by reducing the operator working areas. The number of possible paths (without considering delays) to schedule for an operator j scales as  $w_j^{N_s}$ , where  $w_j$  is the number of gates operator j is allowed to work at and  $N_s$  the number of gates to schedule for each operator in each Model Predictive Control (MPC) iteration. By limiting the working areas, the number of possible paths can be reduced drastically. However, assigning optimal operator working areas is not straightforward, as the division of the operators depends on the desired frequency of measuring and actuation, which is hard to estimate and time dependent.

Moreover, as discussed in Section 5-2-1, by removing the time delays from the optimisation problem the size of the search space can be decreased significantly, as well as the number of nonlinear constraints.

### 5-2-4 Is distributed control required for larger systems?

If the open water system is very large or branches into many smaller canals, it is likely to be intractable to control with a central controller using the approaches from [20, 28, 37] and would require distributed control. However, the cyclic control heuristics proposed in this thesis might offer a simple solution. Consider a complex/large irrigation canal that is divided in linear sections. Moreover, for each of these sections an operator is assigned that travels along a cyclic (downstream) path to regulate the water flow of that particular section. By choosing the cyclic paths such that the presence of an operator and the arrival of water flow are synchronised the complete irrigation canal can be regulated efficiently. This would only require solving a Quadratic Programming (QP) problem during each MPC iteration, which can be done computationally efficient, see Appendix C.

# Appendix A

# **Model Predictive Control**

Model Predictive Control (MPC), which is also known as receding-horizon control or movinghorizon control is a model-based online optimal control approach. MPC is not a specific control strategy, but more of a design methodology. This design methodology contains the following components:

- Internal model: explicit use of a process model to predict the evolution of the system states and outputs over time, as a result of the inputs, the disturbances, and the past states.
- Objective function: this function describes the quality of the control input to the system over the prediction window. It is build up from giving weighted penalties to certain (future) states, inputs, and/or outputs of the system. All weighted penalties together, which are often conflicting, form the objective function.
- Constraints: these put bounds on the solution space of the controller. These limitations can be due to physical limitations of the system, such as pump capacity, or operational specifications, such as the water level within a reach.
- Optimisation: calculation of a control signal that minimises a certain objective function, while taking the constraints into account.
- Receding horizon: control the system in a receding strategy: only the first control action is implemented; the prediction horizon is shifted one time step to the future and the optimisation problem is reformulated and solved.

In Figure A.1, the MPC strategy is illustrated. New information about the process becomes available and a MPC iteration is started at activation time step  $k_{\rm a}$ , in order to find an optimal control action to achieve the desired system state:

1. The evolution of the outputs are predicted during a prediction window, using the internal model. The predicted outputs  $y(k_{\rm a} + k|k_{\rm a})$  for  $k = 1, \ldots, N_{\rm p}$ , depend on the known



Figure A.1: MPC strategy

past inputs and outputs, and the future control actions  $u(k_{\rm a} + k|k_{\rm a})$  for  $k = 0, \ldots, N_{\rm c}$ , during the control window, where  $N_{\rm p}$  represents the prediction horizon and  $N_{\rm c}$  the control horizon.

- 2. The optimal future control signals are calculated by minimising the objective function, that penalises the reference tracking error r(k) y(k) and the control effort u(k).
- 3. The optimal control action at the current time  $u(k_{\rm a}|k_{\rm a})$  is applied to the process and the remainder of the control signal is discarded.

# Appendix B

# **Irrigation Canals Parameters**

In this appendix, the layout, control structures, and canal parameters are presented for the Dez main irrigation canal in Iran and the Arizona west-main irrigation canal in the United States of America in, respectively, Sections B-1 and B-2. Moreover, some background information is given on their location and their use.

As discussed in Section 2-1-2, the flow through the gates is dependent on the water level of the upstream reservoir. To mitigate this dependency the gates of the Dez main canal and Arizona west-main canal are equipped with local flow controllers which control the opening of the gate in order to maintain the water flow at the reference value. If the local flow controller control loops are fast enough, the water levels of the upstream reaches do not have an effect on the downstream water flow and water levels [29]. The only unintended coupling of the adjacent reaches is then through the constraints on the minimum and maximum flow each gate can supply, as this still depends on the upstream water level of each gate. However, for rural irrigation canals this local flow control equipment may not always be available or is not adequately maintained. Therefore, for some parts of this thesis the flow controllers are dismissed in order to have a more realistic representation of rural irrigation canals.

## B-1 Dez main irrigation canal, Iran

The Dez irrigation canal is located in the south-west of Iran, near the city of Dezful. It was designed to transport water from the large dam on the Dez river to the irrigated areas in the north of Khuzestan province. For this thesis, the 45 km long section of the west-main canal is considered, which consists of 13 reaches. Furthermore, the head gate has a maximum discharge capacity of 157 m<sup>3</sup>/s. All of the information about the hydraulic structures and dimensions of the canal have been acquired from the water authority of the Khuzestan Province [14, 37].

### B-1-1 Layout

The west-main Dez canal consists of 13 long and steep reaches in series that stretch a total of 45 km, see Figure B.1. The dimensions of the reaches and their water level setpoints can

be found in Table B-1. Furthermore, the surface area of the backwater part and the delay time of the uniform flow part that are used in the Integrator Delay (ID) model are dependent on the discharge rate the canal is operating in. These ID parameters have been identified for 10% (low flow), 50% (medium flow), and 80% (high flow) of the maximum discharge rate of the west-main Dez canal, see Table B-2.



Figure B.1: Longitudinal profile of the west-main Dez canal.

Reach	Length [m]	Width (bottom) [m]	Channel depth [m]	Setpoint [m+MSL]
1	6219	12	4.62	115.05
2	1933	12	4.62	112.80
3	3718	10	3.75	110.17
4	3906	10	3.75	108.62
5	2934	5	3.66	103.11
6	4670	5	3.66	101.20
7	3110	5	3.66	97.70
8	2240	5	3.66	95.95
9	3405	5	3.49	93.10
10	3820	5	3.15	90.42
11	2520	4	2.89	89.25
12	2874	4	2.68	86.80
13	2468	4	2.68	83.30

Table B-1: reach dimensions and water level setpoints of the west-main Dez canal.

### B-1-2 Control structures

At the downstream side of every reach a so called check structure, i.e., gate, is located that can be controlled, except for the last reach. All of the check structures are free-flowing undershot gates. Moreover, offtake structures are located at certain locations in the reaches

Reach	Low flow		Medium flow		High flow	
	Surface area	Delay time	Surface area	Delay time	Surface area	Delay time
	$(\times 10^5)  [m^2]$	$[\mathbf{s}]$	$(\times 10^5) \ [m^2]$	$[\mathbf{s}]$	$(\times 10^5) \ [m^2]$	$[\mathbf{s}]$
1	3.3415	1080	6.1851	900	0.9318	660
2	0.7376	120	0.9143	60	1.0952	60
3	1.8687	660	0.6041	540	0.8554	360
4	4.7051	720	1.062	540	3.7060	60
5	1.5237	240	1.8141	180	1.7095	360
6	0.6056	720	1.9239	660	0.7786	720
7	0.8390	540	0.8935	540	0.6661	420
8	0.7556	360	1.0595	360	0.8904	360
9	1.0049	420	1.3072	360	0.8671	360
10	1.8355	240	0.4290	540	0.4897	600
11	0.3492	300	0.3673	480	0.4032	240
12	0.4087	660	0.3971	540	0.3820	600
13	0.2971	420	0.3044	480	0.3884	420

Table B-2: reach discharge dependent characteristics of the west-main Dez canal.

for farmers to withdraw water from for irrigation. In this thesis, the offtake structures are not modelled. Instead, the farmers are assumed to just withdrawal water at certain flow rate from the downstream end of each reach. The check structure characteristics can be found in Table B-3. Note that the head gate is not modelled, as it is assumed there is a local flow controller present to follow the reference value set by the model predictive controller.

Gate	Gate crest	Gate width	Maximum	Contraction	Lateral contraction
	elevation	[m]	gate opening	coefficient	coefficient
	[m+MSL]		[m]	[-]	[-]
2	110.42	14.4	4.00	0.78	1
3	110.38	22.0	3.78	0.78	1
4	106.42	13.5	5.12	0.78	1
5	106.42	15.5	3.57	0.78	1
6	99.86	7.4	4.72	0.78	1
7	97.95	7.4	4.27	0.78	1
8	94.45	7.4	4.27	0.78	1
9	92.70	6.8	4.02	0.78	1
10	89.85	6.0	4.27	0.78	1
11	87.17	5.6	4.25	0.78	1
12	87.00	6.8	4.25	0.78	1
13	84.55	6.0	3.25	0.78	1

Table B-3: check structure characteristics of the west-main Dez canal.

## B-2 Arizona west-main canal, United States of America

The Maricopa-Stanfield Irrigation and Drainage District is situated south of Phoenix, Arizona, United States of America. It was established in 1962 to provide irrigation water for agricultural use and provides the required water for almost 87000 acres. One of the main canals for this district is the Santa Rosa canal. For this thesis the 10 km long section of the west-main canal part of the Santa Rosa canal is considered, which consists of eight reaches in series. Furthermore, the head gate has a maximum discharge capacity of 2.8 m<sup>3</sup>/s. All of the information about the hydraulic structures and dimensions of the canal have been acquired from [29, 35, 42].

#### B-2-1 Layout

The Arizona west-main canal is a steep canal consisting of eight reaches with a total elevation drop of 40m over a total length of 10 km, see Figure B.2. The dimensions of the reaches and their water level setpoints can be found in Table B-4. Furthermore, the surface area of the backwater part and the delay time of the uniform flow part that are used in the ID model are dependent on the discharge rate the canal is operating in. These ID parameters have been identified for 10% (low flow), 50% (medium flow), and 80% (high flow) of the maximum discharge rate of the west-main canal, see Table B-5.



Figure B.2: Longitudinal profile of the west-main canal in Arizona.

The backwater surface area of reach 5 is relatively small compared to the other reaches, see Table B-5. Consequently, it is much more sensitive to flow disturbances and other external influences. A local high performance PI controller is available at gate 6, so any flow manipulation is passed almost unaffected [35]. Therefore, reach 5 can be viewed as a transport

Reach	Length [m]	Width (bottom) [m]	Channel depth [m]	Setpoint [m+MSL]
1	118	1.22	1.22	416.23
2	1203	1.22	1.22	406.53
3	423	1.22	1.22	403.42
4	809	1.22	1.22	399.07
5	1954	1.22	1.22	-
6	1669	1.22	1.22	385.53
7	1617	0.61	1.22	381.20
8	1691	0.61	1.22	376.78

Table B-4: reach dimensions and water level setpoints of the west-main canal in Arizona.

Table B-5: reach discharge dependent characteristics of the west-main canal in Arizona.

Reach	Low flow		Medium flow		High flow	
	Surface area	Delay time	Surface area	Delay time	Surface area	Delay time
	$(\times 10^3)  [m^2]$	$[\mathbf{s}]$	$(\times 10^3)  [m^2]$	$[\mathbf{s}]$	$(\times 10^3)  [m^2]$	$[\mathbf{s}]$
1	0.397	0	0.379	0	0.343	0
2	0.653	534	0.600	360	0.450	288
3	0.503	120	0.493	90	0.240	78
4	1.630	162	1.621	72	1.506	60
5	0.171	1152	0.240	828	0.248	702
6	1.614	792	1.385	648	0.878	540
7	2.000	540	1.385	576	1.266	504
8	1.241	1008	1.319	954	1.263	720

reach to convey water towards reach 6. In the model predictive controller, reaches 5 and 6 are considered to be combined into one reach with a storage area equal to the storage area of reach 6 and a delay time equal to the summation of the delay times of reaches 5 and 6.

### **B-2-2** Control structures

At the downstream side of every reach a so called check structure, i.e., gate, is located that can be controlled, except for the last reach. All of the check structures are free-flowing undershot gates. Moreover, offtake structures are located at certain locations in the reaches for farmers to withdraw water from for irrigation. In this thesis, the offtake structures are not modelled. Instead, the farmers are assumed to just withdrawal water at certain flow rate from the downstream end of each reach. The check structure characteristics can be found in Table B-6. Note that the head gate is not modelled, as it is assumed there is a local flow controller present to follow the reference value set by the model predictive controller.

Table B-6: check structure characteristics of the west-main canal in Arizona.

\_\_\_\_\_

Gate	Gate crest	Gate width	Maximum	Contraction	Lateral contraction
	elevation	[m]	gate opening	coefficient	coefficient
	[m+MSL]		[m]	[-]	[-]
2	415.30	1.524	0.991	0.661	1
3	405.60	1.524	0.991	0.662	1
4	402.57	1.524	0.914	0.666	1
5	398.07	1.219	1.067	0.659	1
7	384.75	1.219	0.838	0.654	1
8	380.34	0.610	0.914	0.695	1

# Appendix C

# Quadratic Programming Problem Solving

As mentioned multiple times throughout the thesis report the Mixed-Integer Nonlinear Programming (MINLP) can be reduced to a Quadratic Programming (QP) by choosing a schedule for each operator. This 'choosing' of the schedules is done in [28] using a Genetic Algorithm (GA) and in [37] with an exhaustive search in order to find solutions to the global MINLP problem. Therefore, a large number of QP problems need to be solved to find good solutions to the MINLP problem. Moreover, the number of schedules that can be evaluated per unit of time heavily depends on the solving time of the QP problems. Accordingly, introducing cyclic control increases the solving time of the QP problems. To investigate how much cyclic control increases the QP problem solving times, 10000 operator schedules are generated for the offtake scenario from Section 4-1-5 and the resulting QP problems are solved with and without the fixed cyclic control from Figure 4.3 in Section 4-2-1. Furthermore, the Model Predictive Control (MPC) settings and algorithm are those used in the first case study, see Section 4-2. Moreover, IBM ILOG CPLEX Optimisation Studio V12.7.1 is used to solve all the QP problems. The resulting solving times are depicted in the violin plots in Figure C.1. Cyclic control increased the QP solving time by a factor 1.5 on average. As a result, the number of schedules that can be evaluated using the GA solver per unit of time is decreased by roughly a third. Note that although cyclic control increases the solve time by half, this extra computational burden is negligible compared to the increase in computational burden of the global MINLP problem when enlarging the number of gates in the operator schedule  $N_{\rm s}$ .

Using the same aforementioned settings the solving time of MATLABs QP solver is compared to that of the CPLEX QP solver for a set of 10000 representative problems. The CPLEX QP solver is approximately 4 times faster on average than MATLAB's QP solver for a set of 10000 representative problems, see the violin plots in Figure C.2.



**Figure C.1:** Violin plots for the solving times of 10000 representative QP problems with cyclic control and without cyclic control. The surface area of both of the distributions is equal; the width of the distribution indicates the relative number of data points at that particular calculation time. Furthermore, the green square indicates the median and the red cross the mean.



**Figure C.2:** Violin plots for the solving times of 10000 representative QP problems with MAT-LAB's QP solver and CPLEX' QP solver. The surface area of both of the distributions is equal; the width of the distribution indicates the relative number of data points at that particular calculation time. Furthermore, the green square indicates the median and the red cross the mean.

# Appendix D

# **Genetic Algorithm Functions**

The creation function from Algorithm 1 is used to create the initial population. In this thesis, the settings: popsize = 5000, nprev = 500, ndelay = 1500, Ns = 8, and n = 2 are used. Note that two complete downstream paths are included in the population, one without delays and one with delays that synchronise the operator arrival with the arrival of the water flow. These downstream paths are added to ensure the selected path at the end of the Model Predictive Control (MPC) iteration performs at least as good as the downstream path. Once an initial population is created the cost of the individuals in the population is evaluated and a new population is created using the mutation function from Algorithm 2 and the crossover function from Algorithm 3. The mutation, crossover, and elite fractions have been set to 15%, 80%, 5%, respectively, for all simulations performed during the thesis.

```
input : population size psize, number of operators n, operator paths scheduled by the previous MPC iteration pathprev, current operator positions pos, number of paths that are generated using the previously scheduled paths nprev, number of paths that get random delays added to them ndelay
```

 $output: {\rm Population} \ pop$ 

```
for i \leftarrow 1 to psize do
   if i < nprev then
       // Append on the previously scheduled operator path
       path \leftarrow [pathprev randpath(1)];
       // No delays are added to these operator paths
       delays \leftarrow 0;
   else if i == nprev + 1 then
       // Schedule downstream path
       path \leftarrow downstreampath(pos);
       // No delays are added to these operator paths
       delays \leftarrow 0;
   else if i == nprev + 2 then
       // Schedule downstream path
       path \leftarrow downstreampath(pos);
       Delays are added to the downstream operator paths to synchronise the actuation
        with the water flow;
   else
       // Generate a random path
       path \leftarrow randpath(Ns);
       if i \leq nprev + 2 + ndelay then
          Random delays are added, similar to the mutation function;
       else
          // No delays are added to these operator paths
          delays \leftarrow 0;
       \mathbf{end}
   end
   pop(i) \leftarrow createschedule(path, delays)
end
```

Algorithm 1: Function used to create a population for use in the custom GA

input : parents p1 and p2, number of operators n, number of gates in route Ns output: child that contains parts of the operator schedules from both parents p1 and p2

```
for j \leftarrow 1 to n do
```

 $\mathbf{end}$ 

Algorithm 2: Mutation function used for the custom GA

```
input : parent p, number of operators n, number of gates in route Ns output: child that is a mutated copy of a parent p
```

```
// Randomly select one of the operators
j \leftarrow \text{randi}(n);
// Select the schedule of operator j
s \leftarrow p(j);
// Select a time instant to delay
ind \leftarrow randi(Ns \cdot 2 + 1);
// Select a (uniform) random number between 0 \ {\rm and} \ 100
roll \leftarrow randi(100);
if roll \leq 10 then
   // Reset the delay of the time instant to zero
   s \leftarrow reset(s);
else if roll \leq 25 then
   // Add one time step of delay
   s(ind) \leftarrow s(ind) + 1;
else if roll \leq 35 then
   s(ind) \leftarrow s(ind) + 2;
else if roll < 45 then
   s(ind) \leftarrow s(ind) + 3;
else if roll \leq 50 then
   s(ind) \leftarrow s(ind) + 4;
else
   // Switch two random gates in the schedule
   s \leftarrow switch(s):
   // Check if the operator schedule violates the minimum travelling time
   if infeasible(s) then
       // Repair the operator schedule by adjusting the time instants
       s \leftarrow repair(s);
   \mathbf{end}
end
// Create mutated child
child \leftarrow p;
\texttt{child}(j) \leftarrow \mathsf{s};
                Algorithm 3: Crossover function used for the custom GA
```

# Bibliography

- Directive 2003/88/EC of the European Parliament and of the Council. Official Journal of the European Union, L 299:9–19, November 2003.
- [2] Interpretative Communication on Directive 2003/88/EC of the European Parliament and of the Council concerning certain aspects of the organisation of working time. Official Journal of the European Union, C 165:24–25, May 2017.
- [3] D.J. Antunes, W.P.M.H Heemels, J.P. Hespanha, and C.J. Silvestre. Scheduling measurements and controls over networkspart I: Rollout strategies for protocol design. In *Proceedings of the American Control Conference*, pages 2036–2041, 2012.
- [4] E.F. Camacho and C.B. Alba. Model Predictive Control. Springer Science & Business Media, 2013.
- [5] E.F. Camacho and C. Bordons. Model Predictive Control in the Process Industry. Springer Science & Business Media, 2012.
- [6] V.T. Chow. Open-Channel Hydraulics. New York: McGraw-Hill Book Co. Inc., 1959.
- [7] Z. Cong, B. De Schutter, M. Burger, and R. Babuška. Monitoring of traffic networks using mobile sensors. In *Proceedings of the 17th International Conference on Intelligent Transportation Systems*, pages 792–797, 2014.
- [8] B. De Schutter. Optimal traffic light control for a single intersection. In Proceedings of the American Control Conference, volume 3, pages 2195–2199, 1999.
- [9] B. Dekens, A.D. Sadowska, P.-J. van Overloop, D. Schwanenberg, and B. De Schutter. Gradient-based hybrid model predictive control using time instant optimization for Dutch regional water systems. In *Proceedings of the European Control Conference*, pages 1343– 1348, 2014.
- [10] Deltares. SOBEK manual and technical reference. 2017.
- [11] J. Domingues, D. Valério, and J.S. da Costa. Rule-based fractional control of an irrigation canal. Journal of Computational and Nonlinear Dynamics, 6(2):024503–024503–5, 2010.

- [12] M.W. Fagerland and L. Sandvik. Performance of five two-sample location tests for skewed distributions with unequal variances. Contemporary Clinical Trials, 30(5):490–496, 2009.
- [13] M.P. Fay and M.A. Proschan. Wilcoxon-Mann-Whitney or t-test? on assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics Surveys*, 4:1, 2010.
- [14] F. Fele, J.M. Maestre, S.M Hashemy, D.M. de la Peña, and E.F. Camacho. Coalitional model predictive control of an irrigation canal. *Journal of Process Control*, 24(4):314–325, 2014.
- [15] R. Findeisen, F. Raff, and F. Allgöwer. Sampled-data nonlinear model predictive control for constrained continuous time systems. In Advanced Strategies in Control Systems with Input and Output Constraints, volume 346 of Lecture Notes in Control and Information Sciences, pages 221–249. Springer Berlin Heidelberg, 2007.
- [16] K. Horváth, E. Galvis, M.G. Valentín, and J.R. Benedé. Is it better to use gate opening as control variable than discharge to control irrigation canals? *Journal of Irrigation and Drainage Engineering*, 141(3):04014054, 2014.
- [17] X. Litrico and V. Fromion. Analytical approximation of open-channel flow for controller design. Applied Mathematical Modelling, 28(7):677–695, 2004.
- [18] 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(11):1425–1437, 2005.
- [19] J.M. Maestre, L. Raso, P.-J. van Overloop, and B. De Schutter. Distributed tree-based model predictive control on a drainage water system. *Journal of Hydroinformatics*, 15(2):335–347, 2013.
- [20] J.M. Maestre, P.-J. van Overloop, M. Hashemy, A.D. Sadowska, and E.F. Camacho. Mobile model predictive control for irrigation canals. In *Proceedings of the IEEE Conference* on Decision and Control, pages 4881–4886, Los Angeles, CA, 2014.
- [21] L. Magni and R. Scattolini. Model predictive control of continuous-time nonlinear systems with piecewise constant control. *IEEE Transactions on Automatic Control*, 49(6):900–906, 2004.
- [22] P.-O. Malaterre. Regulation of irrigation canals. Journal of Irrigation and Drainage Engineering, 9(4):297–327, 1995.
- [23] P.-O. Malaterre and J.-P. Baume. Modeling and regulation of irrigation canals: existing applications and ongoing researches. In *Proceedings of the IEEE International Conference* on Systems, Man, and Cybernetics, volume 4, pages 3850–3855, 1998.
- [24] M. Mitchell. An Introduction to Genetic Algorithms. MIT press, Cambridge, MA, 1998.
- [25] R.R. Negenborn, P.-J. van Overloop, T. Keviczky, and B. De Schutter. Distributed model predictive control of irrigation canals. *Networks and Heterogeneous Media*, 4(2):359–380, 2009.

R.C. Kassing

- [26] J. Renner. Area equipped for irrigation at record levels, but expansion slows. volume 20 of Vital Signs, pages 62–65. Island Press, Washington, DC, 2013.
- [27] G.D. Ruxton. The unequal variance t-test is an underused alternative to student's t-test and the Mann–Whitney U test. *Behavioral Ecology*, 17(4):688–690, 2006.
- [28] A.D. Sadowska, P.-J. van Overloop, J.M. Maestre, and B. De Schutter. Human-inthe-loop control of an irrigation canal using time instant optimization model predictive control. In *Proceedings of the European Control Conference*, pages 3274–3279, 2015.
- [29] J. Schuurmans. Control of Water Levels in Open Channels. PhD thesis, Delft University of Technology, Delft, October 1997.
- [30] J. Schuurmans, A. Hof, S. Dijkstra, O.H. Bosgra, and R. Brouwer. Simple water level controller for irrigation and drainage canals. *Journal of Irrigation and Drainage Engineering*, 125(4):189–195, 1999.
- [31] P. Sopasakis, P. Patrinos, and H. Sarimveis. MPC for sampled-data linear systems: Guaranteeing constraint satisfaction in continuous-time. *IEEE Transactions on Automatic Control*, 59(4):1088–1093, 2014.
- [32] X. Tian, R.R. Negenborn, P.-J. van Overloop, J.M. Maestre, A.D. Sadowska, and N.C. van de Giesen. Efficient multi-scenario model predictive control for water resources management with ensemble streamflow forecasts. *Advances in Water Resources*, 2017.
- [33] A. Toudeft and P. Gallinari. Neural and adaptive controllers for a non-minimum phase varying time-delay system. *Artificial Intelligence in Engineering*, 11(4):431–439, 1997.
- [34] H. van Ekeren, R.R. Negenborn, P.-J. van Overloop, and B. De Schutter. Hybrid model predictive control using time-instant optimization for the Rhine-Meuse delta. In *Proceed*ings of the IEEE International Conference on Networking, Sensing and Control, pages 215–220, 2011.
- [35] P.-J. van Overloop. Model Predictive Control on Open Water Systems. PhD thesis, Delft University of Technology, Delft, June 2006.
- [36] P.-J. van Overloop, K. Horváth, and B.E. Aydin. Model predictive control based on an integrator resonance model applied to an open water channel. *Control Engineering Practice*, 27:54–60, 2014.
- [37] P.-J. van Overloop, J.M. Maestre, A.D. Sadowska, E.F. Camacho, and B. De Schutter. Human-in-the-loop model predictive control of an irrigation canal [applications of control]. *IEEE Transactions on Control Systems Technology*, 35(4):19–29, 2015.
- [38] P.-J. van Overloop, S. Weijs, and S. Dijkstra. Multiple model predictive control on a drainage canal system. *Control Engineering Practice*, 16(5):531–540, 2008.
- [39] A.N. Venkat, I.A. Hiskens, J.B. Rawlings, and S.J. Wright. Distributed MPC strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, 16(6):1192–1206, 2008.

- [40] M.P. Vitus, W. Zhang, A. Abate, J. Hu, and C.J. Tomlin. On sensor scheduling of linear dynamical systems with error bounds. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 1318–1323, 2010.
- [41] B. Voron and A.-P. Bouillot. Application of the fuzzy set theory to the control of a large canal. In International Workshop on the Regulation of Irrigation Canals: State of the Art Research and Applications, Marrakech, Morocco, pages 317–331, 1997.
- [42] R. Wagemaker. Model predictive control on irrigation canals. Master's thesis, Delft University of Technology, 2005.
- [43] B.L. Welch. The generalization of student's problem when several different population variances are involved. *Biometrika*, 34(1/2):28–35, 1947.
- [44] E. Weyer. Control of irrigation channels. IEEE Transactions on Control Systems Technology, 16(4):664–675, 2008.

# Glossary

# List of Acronyms

$\mathbf{GA}$	Genetic Algorithm
ID	Integrator Delay
MaMoMPC	Multiple-Action Mobile Model Predictive Control
MINLP	Mixed-Integer Nonlinear Programming
MoMPC	Mobile Model Predictive Control
MPC	Model Predictive Control
$\mathbf{QP}$	Quadratic Programming
TIO-MoMPC	Time Instant Optimisation Mobile Model Predictive Control
TIO-MPC	Time Instant Optimisation Model Predictive Control