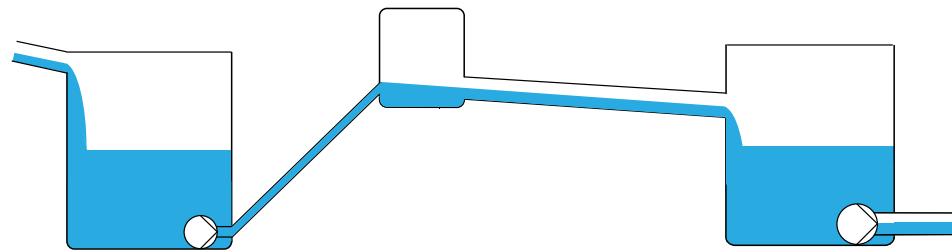


PROJECT OF GROUP 20GR831

Model Predictive Control of Combined Sewer Overflows in Urban Drainage Networks

EIGHTH SEMESTER OF CONTROL AND AUTOMATION



AALBORG UNIVERSITY

2020



Title:

Model Predictive Control of
Combined Sewer Overflows
in Urban Drainage Networks

Project:

P8-project

Period of Project:

February 3rd 2020 - May 27th 2020

Project Group:

20gr831

Members:

Adis Hodžić
Casper Knudsen
David Michalik
Frederik Rasmussen
Jakob Poulsen

Supervisors:

Carsten Skovmose Kallesøe
Jan Dimon Bendtsen
Krisztian Mark Balla
Jorge Val Ledesma

ECTS points: 15

Printed exemplary: 0

Page number: 102

Ended: July 3, 2021

Abstract

The main focus of this report has been the design of an optimal controller for handling large quantities of water in urban drainage networks (UDNs). More specifically, the objective has been to design a controller to prevent Combined Sewer Overflow (CSO) in the case of heavy rain events. After a contextual analysis of sewage systems and waste water treatment plants, the report covers the theory, design and simulation of a real time controller in form of an Model Predictive Controller (MPC). Two numerical approaches to discretizing the Saint-Venant equations, the Euler method and the Preissmann scheme, have been compared to see which method is best suited for creating a model of UDNs. These models were implemented along side a cost function, constraints and disturbances in the MPC. A simulation of a UDN network is carried out in a Python environment using the PYSWMM Python package. Simulations were run with the three different models used for prediction. While the optimization problems created as part of the MPC were feasible, the models in their current form did not have a satisfactory prediction accuracy. From a simple-gain-performance test of the controllers, no definitive conclusion can be made. For the models to provide satisfactory results, more work will have to be done to fit the models onto a real world system. It is finally, concluded that a considerable amount of work has been done towards a performance comparison between the three MPC implementations using different prediction models, however, additional work is required for decisive results.

Table of Contents

1	Introduction	3
1.1	Introduction	3
1.2	Case Description	3
1.3	Project Scope and Initial Problem Formulation	4
2	Contextual Analysis	5
2.1	Brief History of Sanitation and Sewer Systems	5
2.2	Urban Drainage Networks	7
2.3	Sewers As a Chemical and Microbial Reactor	9
2.4	How WWTPs Work	11
2.5	Conclusion	12
3	Real time Control of Urban Drainage Networks	13
3.1	RTC Design Considerations	13
3.2	Optimal Control	14
3.3	Conclusion	17
4	Modeling of Urban Drainage Networks	19
4.1	Pipe model	19
4.2	Retention Tank	25
4.3	Pumps	26
4.4	Conclusion	26
5	Requirements, Limitations and System Overview	29
5.1	Project Objectives	29
5.2	Proposed Network Topology And Disturbance Models	30
5.3	Control Objectives	32
5.4	Technical Requirements	32
5.5	Method of Evaluation	33
5.6	Summary	33
6	Model Discretization and Linearization	35
6.1	Characteristics of the Saint-Venant Equations	35
6.2	Model Simplification	37
6.3	Model Discretization Using the Euler Scheme	43
6.4	Model Discretization Using the Preissmann Scheme	47
6.5	Conclusion	50
7	System and Controller Model	51
7.1	State Space Models	51
7.2	Model Predictive Controller	55

7.3	System Parameterization	59
7.4	Conclusion	61
8	Simulation Design and System Testing	63
8.1	Simulation Building Blocks	64
8.2	EPA SWMM	65
8.3	PYSWMM	66
8.4	MPC Implementation Using CasADi	68
8.5	Disturbance Implementation	71
8.6	System Testing	73
8.7	Test Results	73
8.8	Result Analysis	76
9	Global Conclusion	79
9.1	Discussion	79
9.2	Conclusion	80
	Bibliography	83
	A Grundfos Project Proposal	87
	B Derivation of the Euler Model	91

Introduction

1

1.1 Introduction

In spite of being under ground and out of sight urban drainage networks (UDNs) play an important role for urban infrastructure. UDNs are crucial to ensuring public health and have been in use since ancient time. Modern UDNs lead the contaminated water to a wastewater treatment plant (WWTP) for safe treatment. However, a problem occurs when sewage exceeds a WWTP capacity, as sewage has to be allowed to overflow. This sewage overflow is released into the environment where it poses numerous risk to people and the environment. There are multiple ways to decrease the likelihood of CSOs. The most straightforward way to decrease likelihood of CSOs is to add capacity to the wastewater treatment plant or to create a retention tank to be used as a buffer. These solutions can however be very expensive to employ, and the price scales with the scale of the construction. Another way to reduce overflow is to optimally use the existing storage, and thus less infrastructure has to be constructed. This can be achieved by optimal control. The most common optimal controller used within UDNs is the Model Predictive Controller. The goal of this report is to develop such a controller.

1.2 Case Description

As part of the master's education, "Control and Automation" at Aalborg University, a 2nd semester project has been proposed by Grundfos Holding A/S. The title of the project proposal is "Model Predictive Control of Combined Sewer Overflows in Urban Drainage Networks" see Appendix A.

Students from Aalborg University have previously, in collaboration with Grundfos, conducted research in control of Urban Drainage Networks, see: *Model Predictive Control of a Sewer System* by J. N. Pedersen and T. H. Pilgaard,[1] and *Laboratory Emulation and Control of a Sewer System with Storage* by L. S. Mestre and P. K. Murugesan [2].

This project is motivated by two main points. First, growing population density in European cities and second, increasing rainfall due to changing weather conditions. Both place an increasing load on already established Urban Drainage Networks (UDNs). This increase in domestic wastewater and rainwater can lead to Sewer Overflows, which is a detriment to both the environment and human health, see *Subsection 2.2.1 Effects of Combined Sewer Overflows*.

Though the objective of the project is to develop an optimal control strategy, it is not the only way to reduce the probability of sewer overflows. A straighter forward, but

more expensive, approach is increasing the capacity of drainage networks, by for example creating retention tanks [3]. In general, UDNs have been constructed in two main ways either as combined sewers, where sewage and rainwater are mixed, or separated sewer systems, where only the sewage is taken to a Wastewater Treatment Plant (WWTP). The focus of the project is mainly on the combined sewer systems, since overflows in this kind of sewer has the potential to be the most devastating.

The project proposal presents a sewer topology to be controlled, shown on *Figure 1.1*. This topology will be the starting point of the project. The next section presents the scope of the project and the initial problem formulation.

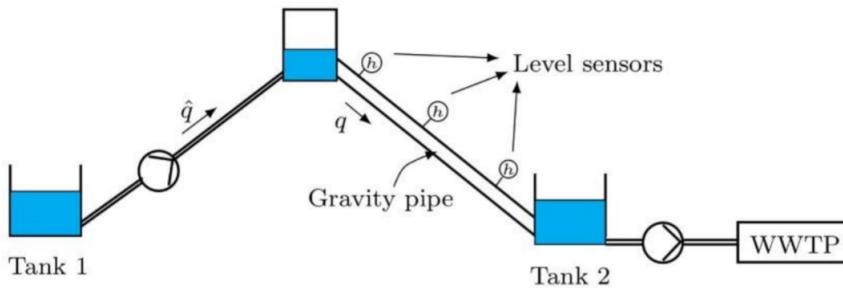


Figure 1.1: Topology presented by Grundfos

1.3 Project Scope and Initial Problem Formulation

The scope of the project: It is desired to explore optimal control with physical constraints. Based on the project proposal we have determined that the most important parameter for which the control should be optimized is sewer overflow. This leads us the initial problem formulation, knowing we have chosen our main control objective to be overflow prevention:

How can we develop a controller that prevents combined sewer overflows in urban drainage networks, in the case of heavy rain events?

Contextual Analysis 2

Modern day sewer systems perform a variety of different functions, ranging from wastewater transportation and storm water routing to ensuring adequate sanitation. The sewer system can be defined as a network of pipes, wells, tanks and other elements such as pump stations which are built to create a coherent flow of excess sewage away from densely populated areas. Specifically, in the modern world they are built to deliver such wastewater to Wastewater Treatment Plants (WWTP). The transportation is performed in different ways depending on the physical properties of the locations. In countries such as the Netherlands, pressurized systems are a requirement, but in more mountainous regions like Barcelona, gravity pipes-based systems are more appropriate. Sewer systems have been utilized during most of human history. Archaeological sites all around the world suggest that complex sewer systems were broadly built and maintained in most major civilizations. The following sections introduce the history of sewerage, how urban drainage networks operate today and to some of the issues experienced with these modern sewer systems.

2.1 Brief History of Sanitation and Sewer Systems

The history of sanitation and sewer systems date back to at least the ancient Mesopotamian civilization. The Mesopotamians (ca. 4000 BC) introduced clay sewer piping systems, which are currently estimated to be one of the earliest occurrences of sanitation. These systems were built to transport domestic waste away from major populations. However, rather than treating the wastewater in specific treatment plants the wastewater was often used for plant irrigation or was directly led into nearby rivers [4].

In Asia, the Indus Valley civilization (ca. 2600-1900 BC) were the first ones to implement more complicated sewerage systems. Excavation sites show that during the era of this civilization, each house had its own private lavatory facilities which were connected to a central sewer system of the city, see *Figure 2.1*. Additionally, this centralized sewer system was covered, which was unlike other early civilizations [5].

In Ancient Europe, the Minoan civilization (ca. 2700 – 1100 BC) on the island of Crete, Greece, were the first ones to implement complex underground clay pipes for wastewater transportation and excess stormwater disposal. They also used such systems to provide fresh water for the capital of Knossos. Additionally, the sewer systems were often flushed with clean water [6].

The Mayans (250 - 600 AD), have also utilized complex water and sanitation networks. They were the first ones to implement a pressurized water-based system. In the city of Palenque Mexico, they also rerouted streams to underground aqueducts to reduce the

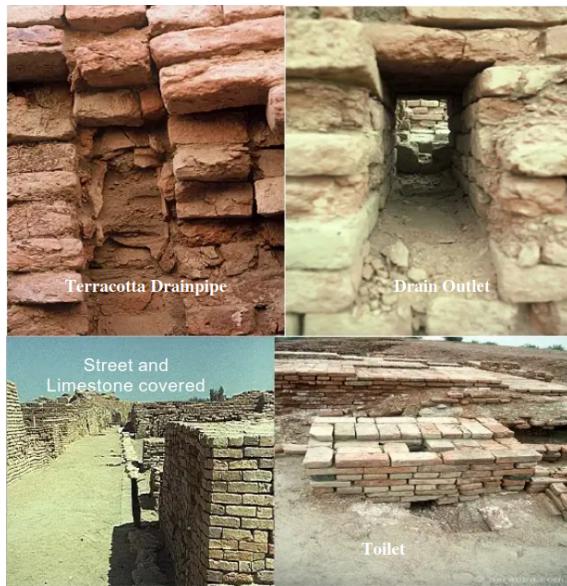


Figure 2.1: Sewer system of the Indus Valley civilization [5]

surface area occupied by water networks [7].

During the time of the Roman Empire (ca. 30 BC - 400 AD), vast and very complicated water and sewer networks have been built around the empire. Even today, much of the aqueduct systems are standing. In Rome, for example, some parts of the original sewer networks are still in operation as part of today's wastewater networks. Unfortunately, after the fall of the empire and during the middle ages most of the engineering solutions and complex underground systems were forgotten and were not used at all [8, p. 36], [9].

Only during the industrial revolution had major cities in Europe begun to adopt new sewer systems. Due to the population boom, and the lack of adequate sanitation, most cities often experienced widespread disease outbreaks. Some cities have used the ancient sewer networks, however, many across Europe did not, and oftentimes the wastewater was directly disposed onto streets and public areas. Public pressure and proper funding have compelled the local governments to act and to build new sewer and sanitation systems. First implementations and enforcement of legislation to curb pollution has also been adopted [9].

A good example is the City of London and its late development of sanitation systems. London largely neglected the implementation of sewer systems until the industrial revolution. Sewage was generally discarded on the streets and into the river Thames. Pollution increased to a degree, where the Thames was described as a black, sticky and smelly sludge. The city has often experienced massive outbreaks of various deadly diseases. It was not until the physician John Snow's discoveries, that diseases were linked to contaminated water. Before his time, the scientific consensus still blamed the so-called "foul-air" for disease transmission. Using a dot distribution map (see *Figure 2.2*) and statistical tools, John Snow has identified areas of London where cholera outbreaks were common. He determined that the disease had to have spread through contaminated water. When the areas which experienced the outbreaks had the water supply isolated, the diseases vanished soon after [9].

Once widespread adoption of new water networks had begun, most of the cities have experienced vast reduction in water-borne disease related fatalities. In some cities in the United Kingdom, such as Liverpool, the average life expectancy nearly doubled.

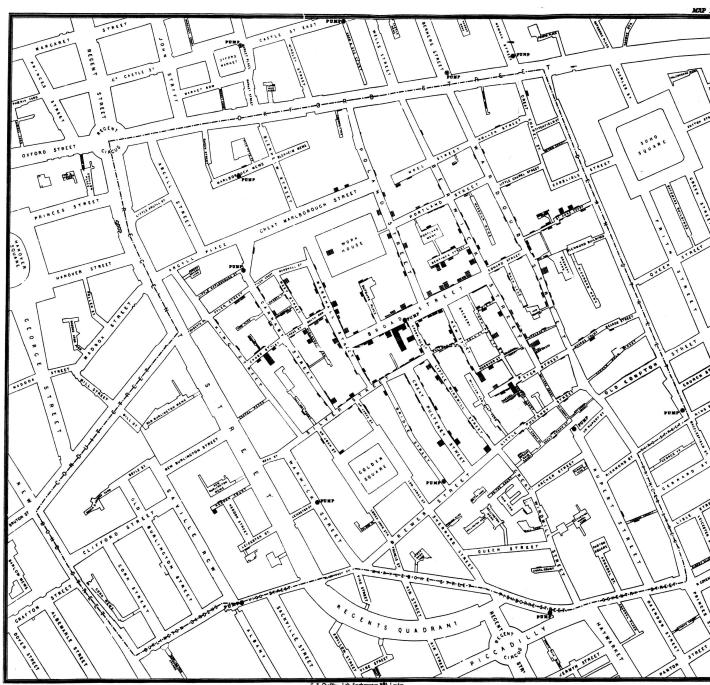


Figure 2.2: Dot distribution map of London cholera outbreak by John Snow [10]

2.2 Urban Drainage Networks

Modern sanitation systems must transport three main types of discharge. Domestic waste, rainwater and industrial discharge. In the 20th and 21st century there has been an increase in the volume of all three types of discharges. This can be contributed to several factors, mainly to global warming and growing urbanization. Today, most of the industrialized world has extensive and complicated sewer networks. Most sewers built since the industrial revolution are all combined sewer systems. This means that, the storm water from rainfall, industrial discharge and domestic waste are all conveyed by a shared pipe system. These sewers then carry the wastewater and rainwater mixture to the WWTPs. After treatment, the water is discharged into the environment, such as rivers or the sea [11]. Though most existing sewers are combined sewer systems, close to all of the newly built sewer systems are designed to be split systems. Split sewer systems carry the domestic wastewater and industrial discharge in a separate network from the storm water. This is because a costly issue arises with the combined sewer systems during heavy rainfall. During heavy rainfall, there is a possibility that the capacity of wastewater treatment plants is reached due to the sudden increase in wastewater load. When the treatment capacity of the WWTPs are reached, the combined wastewater is discharged into the environment without treatment, as shown in *Figure 2.3*. This is called a combined sewer overflow (CSO). CSOs can cause substantial property damage as well as increase the risk of various disease outbreaks. The effects of CSOs are further elaborated in *Subsection 2.2.1 Effects of Combined Sewer*

Overflows.

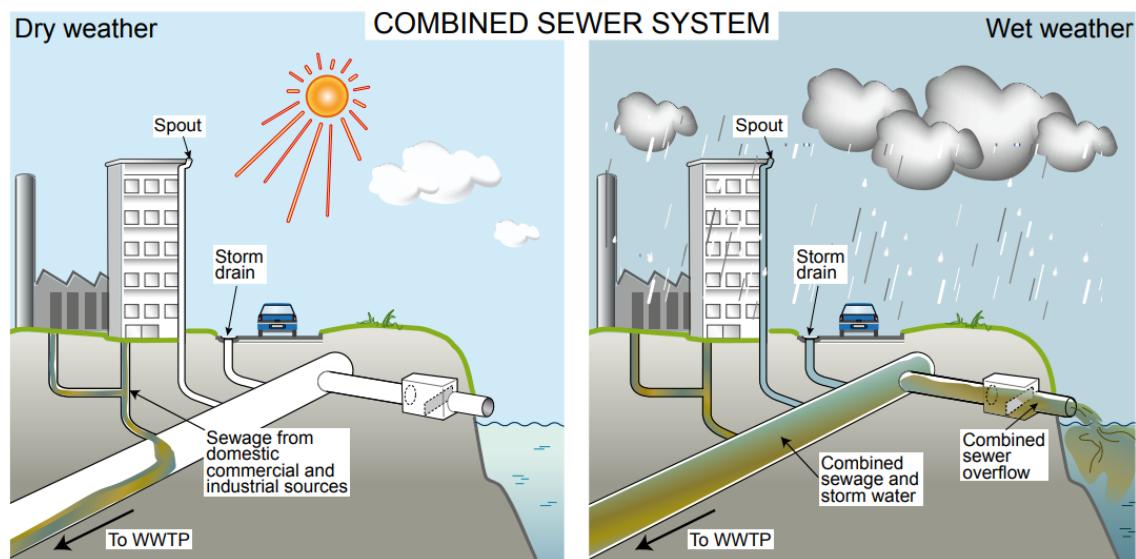


Figure 2.3: Combined sewer overflow as a result of heavy rainfall [11]

The problem of CSOs develop because the WWTPs maximum intake capacity. WWTPs require a specific and delicate bacterial culture to dissolve the sewage. In the treatment tanks during cases of heavy rainfall, the WWTP can be overloaded, leading to the bacterial balance to break down. Tipping this balance can be very expensive, since these bacterial cultures often take several weeks to redevelop. How WWTPs work will be explained in Section 2.4.

There are many suggestions to mitigate the frequency of occurrence and effect of the CSOs. For example, Grundfos proposes to build retention tanks, shown in *Figure 2.4* on the next page, as additions into current systems, which could slow down and offset the sudden load increase effects of rainfall.

Other proposals suggest increasing the intake capacity of the WWTPs or to reduce the intake of storm water into the system. Solutions like this combined with modern control approaches could cheaply mitigate the CSO problem.

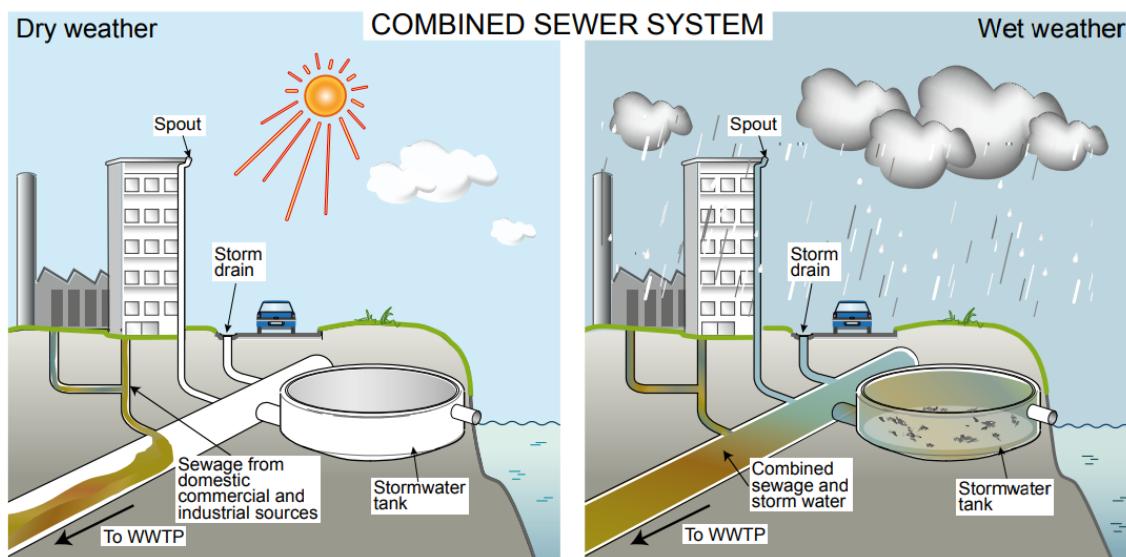


Figure 2.4: Combined sewer overflow prevented as a result of a retention tank [11]

2.2.1 Effects of Combined Sewer Overflows

When sewage overflow occurs, it contaminates public places and waters. The processing of ecosystems is at risk, and people can become ill when exposed to untreated sewage when e.g. drinking from a contaminated community water supply or eating seafood from contaminated water. Additionally, the environmental impacts of sewage include harmful algal blooms, habitat degradation, floating debris and species which depend on fragile ecosystems. Furthermore, combined sewage water from residents and industrial waste, may contain disease-causing pathogens. Bacteria, viruses, worms and protozoa can cause life-threatening illnesses such as Hepatitis B, cholera and dysentery. Lesser threatening illnesses occurs as well such as stomach flu and upper respiratory infections [12].

Sewage overflow impacts the economic activity as well. Fishing industries must slow down or cease completely on contaminated areas of water. This results in factories close to the water lowering their production and consider relocating, thus removing jobs from the area [13].

In addition to economic damage, property damage is an issue as well. Sewer overflows damages driveways, landscaping and outside possessions. Inside the properties, spills can damage foundations, electrical and gas appliances that are typically located in basements. The costs of repair damage and disinfection of properties from floods can be expensive. Communities pay huge amounts of money to clean up and repair overflow damage to sewer infrastructure [12].

2.3 Sewers As a Chemical and Microbial Reactor

Nowadays, the problem of sanitation and environmental effects in sewers are solved by combined wastewater conveyance followed by treatment. However, it is important to note that the treatment of wastewater already begins in the sewers.

Typically, wastewater from domestic households contains between 100,000 to 1,000,000 microorganisms per millimeter before it enters the WWTP. These microorganisms mostly originate from soil and sanitary wastes. They are regarded as a natural living part of the organic waste in the sewers. Their presence in sewers are important, because they degrade and transform wastewater components. Activities of microorganisms in sewers can be important for a biological wastewater treatment plant, since they can help ease the load which WWTPs have to process. One could argue that a biological wastewater treatment plant is dependent on the knowledge of the activities of the microorganisms, to expect a certain load of treatment that is required.

The most significant microorganisms for waste are arguably bacteria, fungi and protozoa. Bacteria have the primary role of degrading the wastewater compounds. They can reproduce rapidly if they are in contact with water. They feed on the waste by absorbing it through their cell wall, turning it into sediment solids. On the other hand protozoa acts like a predator for the bacteria and keeps the population under control [14]. Fungi also degrade wastewater, like bacteria, but can cause problems since fungal growth can expand exponentially and affect the quality of the wastewater [15].

Therefore, the understanding of requirements for optimal environments in sewers are important, as well as the unfavorable conditions in a sewer. Microbial transformations and thus biochemical transformations characterize the sewer environment in terms of wastewater quality. The formation of such low molecular organics can be enhanced by applying certain sewer design characteristics. Wastewater parameters such as pH, temperature, active biomass and biodegradability of the organic matter are all important for the wastewater to have a healthy environment for microorganisms and formation thereof [16].

The way sewers are constructed lead to mainly two conditions, either aerobic or anaerobic. The conditions depend on availability of electron acceptors for a given redox reaction there are in the wastewater. Presence of dissolved oxygen (O_2) and sulphate (SO_4^{2-}) leads to products from redox reactions, which are water (H_2O) and hydrogen sulfide (H_2S) thus aerobic and anaerobic conditions respectively. The aerobic condition is dominant, and the anaerobic condition can only occur when dissolved oxygen is absent. Aerobic conditions lead to degradation of easily biodegradable organic matter. In anaerobic conditions hydrogen sulfide causes toxicity in the sewers, corrosion and bad odor.

The design characteristics of the sewers influence which of the two conditions are in the sewer. The aerobic condition exists in partly filled gravity pipes and aerated pressure sewers. Anaerobic conditions exist in full-flowing gravity pipes and pressure sewers [16]. An overview of when aerobic and anaerobic conditions occur, can be seen on in Table 2.1.

Condition	External electron acceptor	Sewer system characteristics
Aerobic	+ Oxygen	Partly filled gravity pipes Aerated pressure pipes
Anaerobic	- Oxygen + Sulfate + Carbondioxid	Full-flowing gravity pipes Low slope gravity pipes Pressure pipes

Table 2.1: Electron acceptors and corresponding conditions for microbial redox processes [16].

2.4 How WWTPs Work

Wastewater treatment plants' main objective is to break down organic matter and restore the oxygen content of treated water so it can be reused. They have four sets of operations, preliminary-, primary-, secondary- and sludge treatments.

In the preliminary phase the WWTPs remove rather big objects like tree branches and garbage in form of cans, plastic bottles and diapers. In many plants the rate of water inflow can be regulated, such that sand, stones and glass can settle out. The basins function is to hold sewage until its ready for treatment, and to handle overflows. The primary treatment collects the wastewater into large basins and sedimentation tanks. Scrapers will collect solid matter and direct it to hoppers connected to the sludge treatment equipment. Smaller particles will settle out due to gravity. The secondary treatment phase adds beneficial microorganisms to break down organic matter into sludge by aerating and agitating the wastewater in secondary basins. The sludge treatment is the final phase and treats the remaining water, biosolids or sludge. Organic waste from heaver grits gets separated due to gravity. The remaining sludge gets centrifuged and fed to digesting tanks containing anaerobic bacteria. The plant gets power from the methane that these tanks produce. The final product of the sludge can partially be deodorized and plowed into soil as fertilizer. The remaining wastewater have phosphorus, nitrogen and other nutrients removed and then returned to the water supply [17]. In *Figure 2.5* a picture of a WWTP is shown.

WWTPs can only treat a finite amount of wastewater at a time. As mentioned, the plant has basins to store water until the wastewater is ready for treatment. However, if a lot



Figure 2.5: Illustration of a WWTP [18]

of wastewater enter the basins in a short amount of time, the WWTP must reduce the quality of the treatment and let water enter the WWTP to prevent overflow. In worst case the WWTP can overflow, because of the WWTP cannot discharge the treatment water fast enough. Therefore, it's important for the UDNs to help smoothing the inflow to the WWTP to prevent large variations of inflow [19].

2.5 Conclusion

Sewer systems have been utilized during the history of most major civilizations. All over the world, from ancient Europe and Asia to South America, these sewer systems played a crucial role in keeping disease away and providing clean water to cities. During the middle ages, the sewer systems have been neglected which led to the resurgence of many waterborne diseases. To resolve these issues, local governments had to build new and extensive sanitation systems. Such systems form the basis for today's sewer networks. Due to the old design paradigms and climate change induced extreme weather phenomena, combined sewer overflows today pose a serious threat to the environment and health of the population. Thus, overcoming issues during increasingly unpredictable weather patterns with CSOs is critical.

Real time Control of Urban Drainage Networks 3

A case study of the sewer system in Flanders Belgium reckons that the price of establishing new infrastructure to decrease the yearly overflow volume, in the form of retention tanks, would be at least twice as expensive as implementing solutions using RTC [3]. Since this is a case study of a specific region, the result may not be directly transferable to the other regional sewer systems, but the paper points towards a trend in the way CSOs can be prevented. As stated in the conclusion of the paper:

“... RTC [has] showed to be the most cost-effective measure, expressed as highest overflow volume reduction (ecological gain) versus annual equivalent cost. Though being a sole case study, it is believed to reveal an important trend with regard to potential implementations of RTC strategies as optimization measure; ...” - G. Dirckx, M. Schütze, S. Kroll, Ch. Thoeye, G. De Gueldre, and B. Van De Steene [3]

As the field of “Control of urban drainage systems” is shared between a number of engineering disciplines, not limited to control engineering, the definition of Real Time Control is stated to avoid confusion. Real time control is typically by having a digital controller change states of one or more actuators based on sensor data in order to ensure that a system stays at specific setpoint. This definition is a restatement of the one given in: *Model predictive control of urban drainage systems: A review and perspective towards smart real-time water management* [20]. Though not explicitly stated in the definition: Real time controllers are designed, so they take the dynamics of a system into consideration.

Modern sewer infrastructure is controlled, but not based on the dynamic behavior of the sewer whole system. Often, actuators are set to fixed positions or on/off controlled by some local condition such as water level. RTC allows for an increase in utilization of existing infrastructure, by presenting more complicated control solutions.

3.1 RTC Design Considerations

The scope of an RTC system tend to be a function of the sewer systems complexity. The benefits of an RTC system tend grow as bigger sewer systems are considered. For control of sewer systems, it is typical to distinguish between two control layers: Global, also called integrated, control and local control [21]. Both require local equipment, while global control requires additional equipment for data transfer between a local control location and the global control center.

An example of a global control structure is shown in *Figure 3.1*. When implementing global

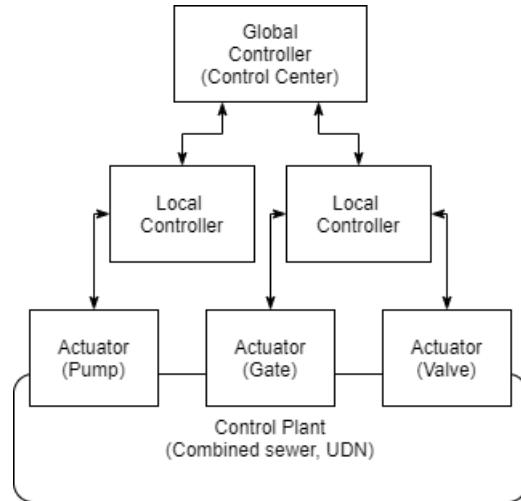


Figure 3.1: Example of control topology in UDNs

control, the global controller is usually built on top of mostly pre-existing infrastructure. This is because most sewers already have some form of simple local control. An example can be a motor connected to a sewer gate, the local controller can be designed to run the motor until the gate is open or closed. The local controllers would have to be updated in order to handle the input from the global controller but would mostly function as they did before global control was introduced.

As stated in *Section 1.2 Case Description* the focus of the project is control of sewer systems, we will therefore not concern ourselves with the control of local actuator dynamics. Instead we will focus on global control, which the practice of optimal control is applicable. Thus, it is desired to investigate the potential of optimal control, to solve the initial problem formulation.

3.2 Optimal Control

Optimal control is a control strategy used for designing a control feedback by solving an optimization problem. Consider a system $\frac{dx(t)}{dt} = f(x(t), u(t), t)$ with an input vector u and a state vector x . The control objective is defined such that it minimizes a cost function given as:

$$\mathcal{J}(x, u, t) = \Phi(x(0), x(T), t) + \int_0^T \mathcal{L}(x(t), u(t), t) dt \quad (3.1)$$

Where states and inputs can be constrained by $x(t) \in X$ and $u(t) \in U$, and $\mathcal{L}(x(t), u(t), t)$ is a function which is never negative. This is a general formulation of the optimal control and it is suitable for describing any practical problem [22]. Unfortunately, finding solutions for such a general, highly nonlinear problem is practically difficult.

A wide array of well-developed mathematical tools for interpreting linear systems has been developed. Therefore, it is common to model dynamic systems as linear. The linear system behavior often results in a choice, which is to cast \mathcal{L} and Φ as quadratic functions

as this result not in a non-linear control problem but rather in a quadratic control problem. According to [22], most of the control problems are solved under these assumptions. As the assumptions allow for a linear control law to be obtained by solving an ordinary differential equation (the Riccati equation). In this section we are going to present the two most commonly used optimal controllers, with quadratic cost functions and linear or no constraints.

3.2.1 Linear-Quadratic Regulator

A Linear-Quadratic Regulator (LQR) is an optimal controller. The dynamic system is described as a set of linear differential equations set up as a state space model. LQR is a way of designing the full-state-feedback for a state space model on the form $\dot{x} = Ax + Bu$. Where the feedback, u , is on the form: $u = -K(x - x_{ref})$. resulting a system dynamic on the form:

$$\dot{x} = (A - BK)x + BKx_{ref} \quad (3.2)$$

Where K is the full-state-feedback. With LQR a certain value or cost, is set for each input and state and a optimization problem is solved based on the those values. The formulation of cost is done using a cost function. The cost function is described with a quadratic equation based on the error of the states and the magnitude of the control action. In matrix form the cost function is written as:

$$L = \frac{1}{2}x^T Qx + \frac{1}{2}u^T Ru \quad (3.3)$$

Where the Q matrix is used to weigh or put a cost to the state errors. The R matrix is used to weigh or put a cost to the control input to the actuators. This allows for tuning based on design requirements in a sensible manner. In the case where there is an actual cost related to parameters, these can be applied almost directly, however some parameters that might still be important in some cases such as convenience or comfort can be hard to assign a real cost to. It can also be difficult to adjust the cost if it is not directly proportional to a parameter.

The optimal solution to the gain K can be found using the Riccati equation, or more specifically for the matrix case, the *matrix algebraic Riccati equation* (MARE) which has to be satisfied.

$$PA + A^T P + Q - PBR^{-1}B^T P = 0 \quad (3.4)$$

Where the solution P is used in the feedback equation:

$$K = R^{-1}B^T P \quad (3.5)$$

This will often give multiple solutions, but usually only one is stable and should be chosen. Which will then be the optimal full-state-feedback based on the given cost function.

Thus, a simple step by step solution can be followed to design feedback using LQR:

1. Derive A and B from the model of the system
2. Choose appropriate Q and R cost matrices
3. Solve MARE for the system
4. Compute feedback gain K and select the stable version

5. Check if it behaves as intended
6. If the system does not behave as intended, go to the second step and adjust Q and R matrices
7. If system behaves as intended the LQR is done

3.2.2 Model Predictive Control (MPC)

A predictive controller uses a system model to predict the system's future behavior. The period for which the model is used for predictions is called the *prediction horizon* (H_p).

The future behavior is a result of the input trajectory $\hat{u}(k + i|k)$ ($i = 0, 1, 2, \dots, H_p - 1$), where current time is denoted with a time step k . The notation $\hat{u}(k + i|k)$ indicates that the input trajectory depends on the observations of the states at time k [22]. The idea is to choose the input trajectory which promises the best predicted response with respect to some functional requirements or objectives. In other words, the control input is the solution of the optimization problem.

Most often, the *least squares* method is used to find the optimal trajectory. Namely, the sum of the squares of the errors $\sum_{i=0}^{H_p-1} [r(k+i|k) - \hat{y}(k+i|k)]^2$, or the quadratic function (Equation (3.3)) is minimized over the prediction horizon [22].

Once the optimal input trajectory is chosen, only the first element is applied as the actual input to the system $u(k) = \hat{u}(k|k)$. Then the prediction horizon is moved one sample step forward and the procedure of finding the optimal solution is repeated [22]. Parameters of the MPC are shown on *Figure 3.2*.

Often, other control objectives along with the minimum output error are defined and each objective is assigned with a cost. Together they form the cost function (\mathcal{J}). Therefore, this function is minimized with respect to predicted state trajectory to minimize the cost. The optimization is defined as follows:

$$\min_{\hat{u}(n|n), \dots, \hat{u}(n+H_p-1|n)} \mathcal{J}(y, u, \dots, j) \triangleq \sum_{j=1}^{\Gamma} \sum_{i=0}^{H_p-1} \|F_j(i)\|_{Q(i,j)}^2 \quad (3.6)$$

where:

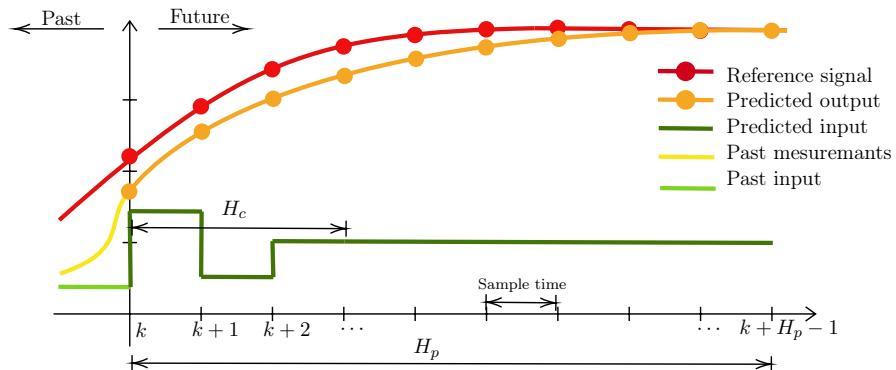


Figure 3.2: Discrete time MPC

$\ x\ _Q^2$	is the quadratic form.
H_p	is the prediction horizon.
\mathcal{F}	is the control objective.
\mathcal{Q}	is the cost of each objective.
Γ	is the number of objectives.

Furthermore, the control strategy is subject to various operational constraints (the rate of change of control input, saturation) and physical constraints which define limits of the actuators and/or states. Satisfying these constraints while solving the optimization problem is a unique advantage of MPC. Usually constraints are in form of linear inequalities such as:

$$\hat{u}_{min} \leq \hat{u}(n+i|n) \leq \hat{u}_{max}, \quad \forall i = 0, \dots, T_n, \quad T_n \leq H_p, \quad (3.7)$$

$$|\hat{u}(n+i|n) - \hat{u}(n+i-1|n)| \leq \Delta \hat{u}_{max}, \quad \forall i = 0, \dots, T_n, \quad T_n \leq H_p, \quad (3.8)$$

The biggest difference from an LQR controller is that MPC optimizes in a receding prediction window. The MPC updates the optimization problem with the current state measurements at each time step, while the LQR optimizes trajectory from the initial states over the entire horizon. This allows the MPC to have the ability to anticipate future act of disturbances and handle the migration of non-linear system from its linearized operating points. These are two advantages over the LQR method.

3.3 Conclusion

The objective is to design a controller which helps to mitigate sewer overflows. Of the two presented control approaches, Model Predictive Control (MPC) seem to have the most potential to do mitigate sewer overflows. Therefore, it is chosen to have the development of this kind of controller as the focus for the rest of the report. This choice is motivated specifically by the fact that an MPC has the ability to anticipate future act of disturbances, as this will help the system prepare for major rain events.

Modeling of Urban Drainage Networks 4

The internal MPC model describes the dynamic behavior of the urban drainage network. A UDN consists of various passive elements for storing and transport of sewage such as: gravity pipes, retention tanks and catchment areas. The most common used actuators in these networks are pumps, gates and weirs which provide or alter the flow of the sewage. In this chapter, modelling of each element comprising the proposed network from *Section 1.2 Case Description* is described. This chapter is based on two main sources *Control-Oriented Modeling of Thermodynamic Systems*[23] by *T. S. Pedersen and P. Andersen* and *Applied Hydrology*[24] by *V. T. Chow, D. R. Maidment, and L. W. Mays*

4.1 Pipe model

Modeling fluid flow in a pipe is often done using the Saint-Venant equations. These equations are also known as the shallow water equations since they are derived based on an assumption which is valid for shallow water channels. That assumption is: that the density of the fluid in the channel is constant perpendicular to the flow direction.

The term "Saint-Venant equations" cover two specific equations, the conservation of mass and the conservation of momentum. We present these two equations respectively in *Subsection 4.1.2 Conservation of Mass* and *Subsection 4.1.3 Conservation of Momentum* and summarize them in *Subsection 4.1.5 Final Expressions for the Pipe Model*. Both equations are based on Reynolds transport theorem which is presented in the coming subsection.

4.1.1 Reynolds Transport Theorem

In Lagrangian approach to mechanics, Newtons laws are applied to a particle and the focus is on the movement of said particle. This approach is however inconvenient when working with fluids, as it is often considered a non-solid system of particles. Instead of concerning with the movement of a single particle, it is preferred to focus on a control volume through which fluid passes. This is called the Eulerian approach to mechanics. [24,p. 20-24]

Eulerian mechanics separates properties of fluid into two categories: intensive property b and the extensive property B . These are related by $\frac{dB}{dM} = \beta$. Here the parameter B describes some fluid system parameter, for example mass, momentum or energy, while the parameter β describes the same property but per unit mass M .

The Reynolds transport theorem describes physical laws that are normally applied in the

Lagrangian sense and "translate" them for use in a fluid system. On *Figure 4.1 a and b*, a particle system is shown traveling through a pipe. We want to express the change in property B of the particle system using a control volume. [24,p. 20-24] The change in the particle system parameter B during the time interval Δt can be defined as:

$$\frac{dB_{sys}}{dt} = \lim_{\Delta t \rightarrow 0} \left[\frac{B_{t+\Delta t} - B_t}{\Delta t} \right] \quad (4.1)$$

At time t we assume the particle system coincides with our control volume, shown by volume I and II on *Figure 4.1 c.*

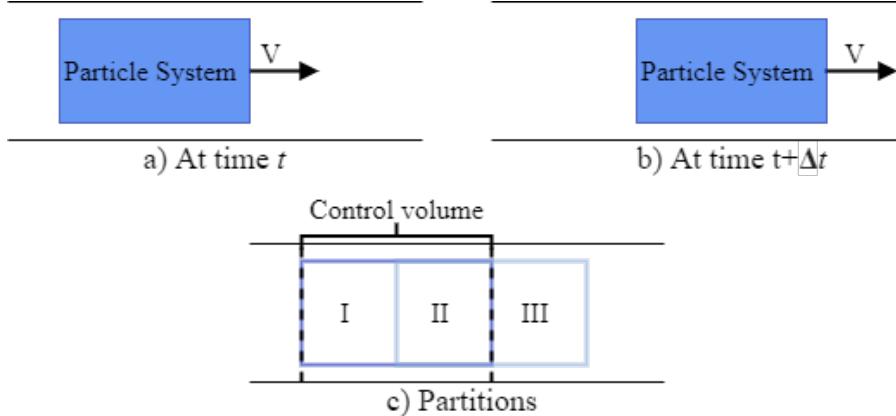


Figure 4.1: Illustration of a pipe with Particle system and Control volume

Since the particle system is traveling at some velocity V , we assume that the particle system makes up area II and III on *Figure 4.1 c* at time $t + \Delta t$ [23]. This is expressed as:

$$\frac{dB_{sys}}{dt} = \lim_{\Delta t \rightarrow 0} \left[\frac{(B_{II} + B_{III})_{t+\Delta t} - (B_I + B_{II})_t}{\Delta t} \right] \quad (4.2)$$

We want to express (4.2) using a term for the property B remaining in the control volume as the particle systems travel during the interval Δt :

$$\frac{dB_{sys}}{dt} = \lim_{\Delta t \rightarrow 0} \left[\frac{B_{II,t+\Delta t} - B_{II,t}}{\Delta t} \right] + \lim_{\Delta t \rightarrow 0} \left[\frac{B_{III,t+\Delta t} - B_{I,t}}{\Delta t} \right] \quad (4.3)$$

as $\Delta t \rightarrow 0$ the II region coincides with the control volume and the first term in (4.3) becomes the derivative of the parameter B stored in the control volume.

$$\lim_{\Delta t \rightarrow 0} \left[\frac{B_{II,t+\Delta t} - B_{II,t}}{\Delta t} \right] = \frac{d(B_{CV}(t))}{dt}$$

While the second term from (4.3) becomes the net flow out of our control volume, where b is B pr second:

$$\lim_{\Delta t \rightarrow 0} \left[\frac{B_{III,t+\Delta t} - B_{I,t}}{\Delta t} \right] = -b_{in}(t) + b_{out}$$

Combining these results we can restate (4.3) as [23]:

$$\frac{d(B_{sys}(t))}{dt} = \frac{d(B_{CV}(t))}{dt} - b_{in}(t) + b_{out}(t) \quad (4.4)$$

Or more formally in terms of intensive and extensive properties, which is known as Reynolds transport equation. [24, p. 20-24]

$$\underbrace{\frac{dB_{sys}}{dt}}_{\text{Time rate of change of } B \text{ in the system}} = \underbrace{\frac{d}{dt} \iiint_{c.v.} \beta \rho dV}_{\text{Time rate of change of } B \text{ in CV}} + \underbrace{\iint_{c.s.} \beta \rho \bar{V} \bullet d\bar{A}}_{\text{Net output flow of } B} \quad (4.5)$$

A deviation of Reynolds Transport theorem has now been done, though a more complete derivation exists. It has been chosen to derive this version, since it is considered to lend the best understanding to the theorem. A more complete deviation can be found in the book: Applied Hydrology [24, p. 20-24]

4.1.2 Conservation of Mass

In the previous section Reynolds transport equation was derived and in this section we choose to express the extensive property as mass, $B = M$

$$\frac{d(M_{sys}(t))}{dt} = \frac{d(M_{CV}(t))}{dt} - m_{in}(t) + m_{out}(t) \quad (4.6)$$

where M_{CV} is the mass in the control volume, m_{in} and m_{out} are the mass flows in and out of the control volume respectively. According to the conservation of mass principle, a closed system does not change its mass over time. That can be expressed as:

$$\begin{aligned} \frac{d(M_{sys}(t))}{dt} &= 0 \\ \Downarrow \\ \frac{d(M_{CV}(t))}{dt} &= m_{in}(t) - m_{out}(t) \end{aligned}$$

Until now, the assumption presented in *Section 4.1 Pipe model*; that the density changed in the x direction, has not yet been considered. To do this we substitute $m(t) = m(x, t)$ making (4.6) into what is known as a distributed parameter description.

It is possible to express a control volume of length Δx using an integral shown in equation (4.1.2), where A is the cross-sectional area of the pipe and ρ is the density of the fluid in the pipe:

$$\frac{d(M_{CV}(t))}{dt} = \frac{d}{dt} \left(\int_x^{x+\Delta x} A(\xi, t) \rho(\xi, t) d\xi \right)$$

which for any infinitesimal control volume $\Delta x = \partial x$ yields:

$$\begin{aligned} \frac{\partial[A(\xi, t)\rho(\xi, t)]}{\partial t} \partial x &= m(x, t) - m(x + dx, t) \\ \Downarrow \\ \frac{\partial[A(x, t)\rho(x, t)]}{\partial t} &= \frac{m(x, t) - m(x + dx, t)}{\partial x} \end{aligned} \quad (4.7)$$

letting $\partial x \rightarrow 0$ (4.7) yields a partial differential equation, known as the one dimensional conservation of mass equation, shown below:

$$\frac{\partial[A(x, t)\rho(x, t)]}{\partial t} = -\frac{\partial m(x, t)}{\partial x} \quad (4.8)$$

The one dimensional conservation of mass equation is the first of the two Saint-Venant equations.

4.1.3 Conservation of Momentum

Previously, the Reynolds transport equation was derived and in this section we choose to express the extensive property as momentum, $B = p$. While still adhering to the assumption that density changed in the x direction. This yields an expression for the dynamics of the fluid in a control volume:

$$\frac{dp_{sys}}{dt} = \frac{d(p_{CV}(t))}{dt} - F_{in}(t) + F_{out}(t) \quad (4.9)$$

where p_{CV} is the momentum stored in the control volume F_{in} and F_{out} are the moment added to the system by in and out flows respectively. If outside forces, such as gravity, is acting on the system while it travels through the control volume, the change in momentum can be expressed as:

$$\begin{aligned} \frac{dp_{sys}}{dt} &= \sum F \\ &\Downarrow \\ \sum F &= \frac{d(p_{CV}(t))}{dt} - F_{in}(t) + F_{out}(t) \end{aligned} \quad (4.10)$$

In the previous section, the net mass outflow was expressed from the control volume as:

$$-\frac{\partial m(x, t)}{\partial x} = m(x, t) - m(x + \Delta x, t) \quad (4.11)$$

So, the expression for momentum added form net outflow can be expanded using, Newtons second law. It states that $F = ma$ or equivalently $p = mv$ yielding:

$$\begin{aligned} -F_{in}(t) + F_{out}(t) &= -v(x, t)m(x, t) + v(x + \Delta x, t)m(x + \Delta x, t) \\ &= -v(x, t)m(x, t) + v(x, t)m(x, t) + \frac{\partial v(x, t)m(x, t)}{\partial x} \Delta x \\ &= \frac{\partial}{\partial x} [v(x, t)m(x, t)] \Delta x \\ &= \frac{\partial}{\partial x} \left[\frac{m(x, t)^2}{A(x, t)\rho(x, t)} \right] \Delta x \end{aligned} \quad (4.12)$$

Where $v(x, t)$ is the velocity of the fluid in the control volume. Expressing the momentum stored in the control volume using Newtons second law as well, yields:

$$\begin{aligned} \frac{dp_{CV}(t)}{dt} &= \frac{dM_{CV}(t)}{dt} v(x, t) \\ &= \frac{\partial}{\partial t} \left[\rho(x, t)A(x, t)\Delta x \frac{m(x, t)}{\rho(x, t)A(x, t)} \right] \\ &= \frac{\partial m(x, t)}{\partial t} \Delta x \end{aligned} \quad (4.13)$$

Substituting (4.12) and (4.13) into (4.10) yields the second of the two Saint Venant equations:

$$0 = -\sum F + \frac{\partial m(x, t)}{\partial t} \Delta x + \frac{\partial}{\partial x} \left[\frac{m(x, t)^2}{A(x, t)\rho(x, t)} \right] \Delta x \quad (4.14)$$

also known as the one dimensional equation for conservation of momentum.

4.1.4 Force Effects

Without expanding the $\sum F$ term in Equation (4.14) the equation is not of much use in modeling a pipe. Therefore, the term $\sum F$ is expanded. Equation (4.14) is used to model flow of sewage in a pipe, where the following external hydrodynamic effects can be expected to affect the flow:

1. Pressure
2. Gravity
3. Pipe surface friction
4. Air friction
5. Contraction/expansion

Three hydrodynamic effects dominate the behavior of the flow, these are pressure, gravity and pipe surface friction. The effect of air friction and contraction/expansion are negligible. We therefore choose only to expand on pressure, gravity and pipe surface friction in this section.

Pressure

The net pressure force exerted on the fluid in the control volume is going to be a result of unbalanced pressure on the left and right side of the control volume ($F_{pl} - F_{pr}$) and the pressure change along the sides of the control volume F_{pc} . [24,p. 277] Meaning:

$$F_p = F_{pl} - F_{pr} + F_{pc} \quad (4.15)$$

The unbalanced pressure is shown in *Figure 4.2 a*. If we first consider the pressure on the left side of the control volume, we can express the hydrostatic pressure as:

$$F_{pl} = \int_0^H \rho(x, t) g(h - W) B dW \quad (4.16)$$

Where $\rho(x, t)$ is the density of the fluid in the pipe and g is the gravitational acceleration. Shown in *Figure 4.2 b*, dW is an infinitesimal slice of the pipe, B is the width of the slice, h is diameter of the pipe and W is the height of the slice in the pipe. Note that all units shown in *Figure 4.2 b* are functions of the x direction though they will not be denoted as such in the coming equations.

From the pressure on the left side the pressure exerted on the right side of the volume can be expressed using a pressure gradient:

$$F_{pr} = F_{pl} + \frac{\partial F_{pl}}{\partial x} \Delta x \quad (4.17)$$

Which can be expanded upon by using Leibniz rule for differentiation:

$$\begin{aligned} \frac{\partial F_{pl}}{\partial x} &= \int_0^h \rho(x, t) g \frac{\partial h}{\partial x} B dW + \int_0^h \rho(x, t) g(h - W) \frac{\partial B}{\partial x} dW \\ &= \rho(x, t) g A(x, t) \frac{\partial h}{\partial x} + \int_0^h \rho(x, t) g(h - W) \frac{\partial B}{\partial x} dW \end{aligned} \quad (4.18)$$

The pressure change along the sides of the control volume is related to the change in the diameter of the pipe and is expressed as:

$$F_{pc} = \left[\int_0^H \rho(x, t) g(h - W) \frac{\partial B}{\partial x} dW \right] dx \quad (4.19)$$

Restating equation (4.16) in terms of Equations (4.17), (4.18) and (4.19) yields:

$$F_p = -\rho(x, t)gA(x, t)\frac{\partial h}{\partial x}\Delta x \quad (4.20)$$

Which is the total pressure effect on the control volume.

Gravity

Given a volume of fluid $A \cdot dx$, such as our control volume, we can express its weight as $\rho(x, t) \cdot g \cdot A \cdot dx$, assuming a level pipe. In the case of an inclined pipe we get the lateral force component as shown below:

$$F_g = M_{CV} \cdot g \cdot \sin(\theta) = \rho(x, t)A(x, t) \cdot \Delta x \cdot g \cdot \sin(\theta) \quad (4.21)$$

Where θ is the inclination of the pipe, as shown in *Figure 4.2 A.* [24, p. 276]. When working with pipes $\sin(\theta)$ is often represented by S_b which is called the "bed slope" so (4.21) becomes:

$$F_g \approx \rho(x, t)A(x, t) \cdot \Delta x \cdot g \cdot S_b \quad (4.22)$$

Friction

The friction forces are created by the stress along the inside of the pipe. For our control volume the friction can be written as:

$$F_{fp} = -S_f \cdot M_{CV} = -S_f \cdot \rho(x, t)A(x, t)\Delta x \cdot g \quad (4.23)$$

Where S_f is known as the friction slope [24, p. 276]. The friction slope is described as the "*The rate at which energy is lost along a given length of channel*" and is often calculated

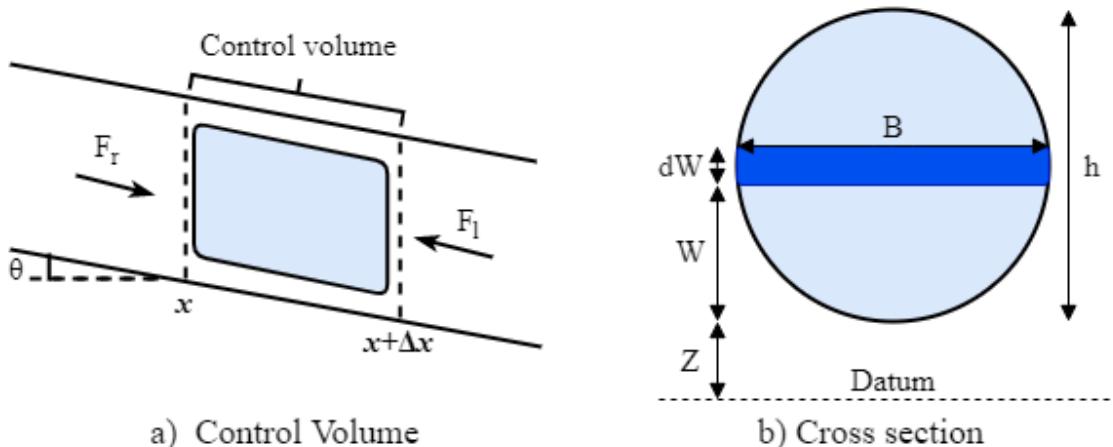


Figure 4.2: Cross-section of sewer pipe filled with water [24]. Note all sizes shown in b) are a function of x

using empirical equations such as the Manning equation or the Darcy–Weisbach equation. Since it describes a ratio of loss it has no unit, but it is till often described using $\frac{m}{m}$

4.1.5 Final Expressions for the Pipe Model

Restating the Saint-Venant equations as applied to the model the mass balance is expressed as:

$$\frac{\partial[A(x,t)\rho(x,t)]}{\partial t} = -\frac{\partial m(x,t)}{\partial x} \quad (4.24)$$

and expanding the expression for conservation of momentum, presented in (4.14), the sum of forces is expressed as:

$$\begin{aligned} \sum F &= F_{pres} + F_{grav} + F_{fric} \\ &= (g \rho(x,t)A(x,t)\Delta x)(-S_f + S_b - \frac{\partial h}{\partial x}) \end{aligned} \quad (4.25)$$

inserting this into (4.14) and removing the common t term Δx the conservation of momentum is stated:

$$0 = -(g \rho(x,t)A(x,t))(-S_f + S_b - \frac{\partial h(x)}{\partial x}) + \frac{\partial m(x,t)}{\partial t} + \frac{\partial}{\partial x} \left(\frac{m(x,t)^2}{A(x,t)\rho(x,t)} \right) \quad (4.26)$$

In *Section 6.2 Model Simplification*, the meaning behind each term will be expanded upon, and in *Subsection 6.2.3 Linearization of the Diffusion Wave Model* a simplified form of the Saint Venant equations will be used to create a model to be used in a MPC.

4.2 Retention Tank

Retention tanks in sewers functions as a storage of wastewater that is not yet ready to be sent further into the sewer system. This could for example be due to a WWTP has reached its treating capacity. The mass in the tank can be expressed by the conservation of mass explained in *Subsection 4.1.2 Conservation of Mass*.

The change of mass in a system is expressed in Equation (4.6). Extending the m_{out} term from Equation (4.6) to include tank overflow, the mass balance can be expressed as:

$$\frac{d(M_{CV}(t))}{dt} = m_{in}(t) - (m_{out}(t) + m_{of}(t)) \quad (4.27)$$

where m_{of} is the mass flow that flows over the edge of the tank and $m_{out}(t)$ is now only the mass leaving through the output pipe. Rewriting the equation such that $M_{CV} = \rho Ah$ and $m = \rho Q$, Equation (4.27) can be expressed as the change of height instead of mass. The expression can be seen in Equation (4.28).

$$\begin{aligned} \frac{d\rho Ah(t)}{dt} &= \rho Q_{in}(t) - \rho Q_{out}(t) - \rho Q_{of}(t) \\ \rho A \frac{dh(t)}{dt} &= \rho(Q_{in}(t) - Q_{out}(t) - Q_{of}(t)) \\ \frac{dh(t)}{dt} &= \frac{1}{A}(Q_{in}(t) - Q_{out}(t) - Q_{of}(t)) \end{aligned} \quad (4.28)$$

where:

h	is the level of the liquid.	[m]
ρ	is the density of the fluid.	[kg/m³]
A	is the horizontal cross-sectional area of the retention tank.	[m²]
Q	is the flow.	[m³/s]

An illustration of the retention tank can be seen on *Figure 4.3*.

This model assumes that turbulence in the input- and output flow is neglected, and that the density is uniform, meaning the mixing from the input flow happens instantly.

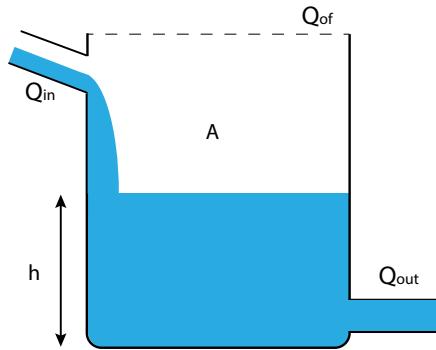


Figure 4.3: Retention tank model

4.3 Pumps

In some pipes and retention tanks the flow can be controlled by a pump. Ideally the pumps should be modelled to predict its behavior, this can be done in great detail [25]. However, the pumps usually have local controllers and work on a very different timescale than the sewer system as a whole. Therefore it is assumed that the pump will practically follow the reference and thus the pumps can be interpreted as a flow provider. This is also the case when pumps are controlled with an on/off controller similar to PWM. Since the pump can be viewed as a flow provider a simple constant or linear model can be used.

$$Q_{\text{pump}}(n) = \text{constant}$$

or

$$Q_{\text{pump}}(n) = x(n) \cdot G \quad (4.29)$$

$$(4.30)$$

Where G is the a constant gain.

4.4 Conclusion

This chapter presented a detailed non-linear mathematical model of the flow in a gravity sewer pipe. This model together with hydraulic models of other sewer elements is used in high-fidelity simulation environments for a UDN analysis and design (for example MIKE URBAN and SWMM). However, these models are too complex for implementing in the MPC design. Instead, much simpler and linear models are used in a standard MPC. It is a

challenging task to find an appropriate simplified model which captures all the important dynamics of the UDN system, since better accuracy reflects directly in the quality of the optimization procedure [20, p. 304]. The simplified models are described in *Chapter 6 Model Discretization and Linearization*.

Requirements, Limitations and System Overview

5

This chapter is going to present the project objectives, the proposed network topology and the technical and functional requirements of the MPC which will be designed. Furthermore a disturbance profile is also presented here in conjunction with the planned test scenarios. Technical decisions made by the group during the research phase are also implemented in the requirement specifications. Where it is possible, the tractability of the requirements is stated. In the end, the final problem formulation statement is presented.

5.1 Project Objectives

The project scope was introduced in *Section 1.3 Project Scope and Initial Problem Formulation*, where the group decided to pursue the creation of a optimal controller with a single initial objective to control an Urban Drainage Network. Based on the initial research presented in previous chapters, and the project proposal, this section introduces the major goals the rest of the project.

- Design a network topology which can mimic real life like operation by including various storage elements and actuators, such as:
 - Retention tanks
 - Gravity pipes
 - Pumps
 - Outfalls to a WWTP
- Model the network topology using the modeling techniques outlined in *Chapter 4 Modeling of Urban Drainage Networks*.
- Design an MPC with the following control objectives:
 1. Reduce the effect of Combined Sewer Overflows
 2. Smoothen inflow to WWTP
 3. Ensure equal usage of storage in the network
- Construct a disturbance dataset on which the model can be tested on, such as:
 - A dataset based around domestic and/or industrial waste in the sewer systems.
 - A forecast prediction for rainfall.

Furthermore, to qualitatively and quantitatively test the performance of the designed MPC, a comparative analysis needs to be made between a conventional local control scheme, on/off control, to prove whether the proposed MPC is an improvement over the status quo. Therefore, these project objectives are created to guide the implementation and testing phase of the project:

1. Create a simulation environment in which the comparative analysis can be executed.
This could be achieved in the following environments:

- Matlab & Simulink
- EPA SWMM and python
- AAU Modular WaterLab

2. Create a testing scenario, which incorporates inflow disturbance into the proposed network topology. Then use the testing scenarios, to compare the following control schemes:

- A local controller
- Model Predictive Controller with different prediction models

In the following sections, the proposed network topology and the disturbance is further elaborated on.

5.2 Proposed Network Topology And Disturbance Models

The original network proposed in the *Chapter A Grundfos Project Proposal* consists of two retention tanks, two pumps, a manhole, a gravity pipe and an inflow to a WWTP. The proposal is missing specific dimensions, such as length and depth of the tanks and gravity pipe. It was originally intended that the controller is implemented at Aalborg university's Modular WaterLab, however, due the lockdown caused by the Covid-19, this is not possible. Instead using the proposal an arbitrary network is created. Thus, the dimensions specified in the following section are only meant to replicate a real-life system by using approximate measurements which are considered close to real life proportions.

The network topology consists of two tanks, T_1 and T_2 , with one pump each, a pressure pipe and a gravity pipe. They have a squared bottom with a surface area of 150 m^2 and 200 m^2 and a height with 3 m and 2 m respectively. The gravity pipe is assumed to be 200 m long. From the beginning of the pipe until the end of the pipe, the channel descends 10 m . Using these dimensions of the pipe, the slope becomes $S_b = 0.05$. The setup can be seen in *Figure 5.1*.

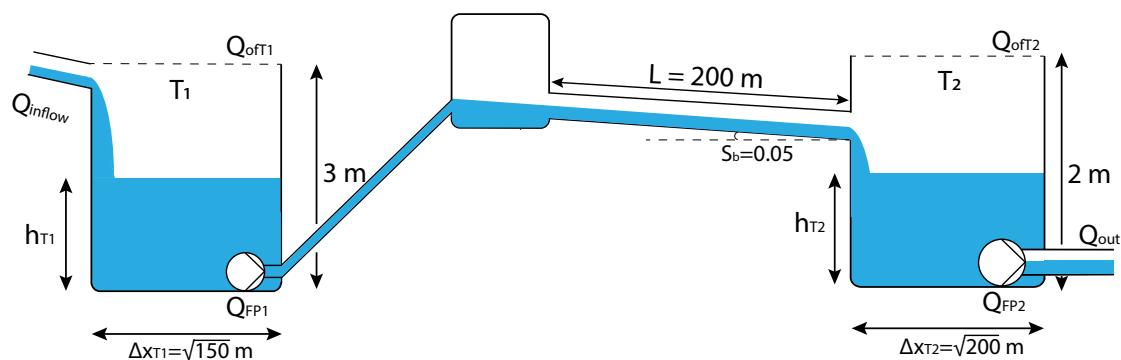


Figure 5.1: Topology of AAU water-lab

5.2.1 Disturbances models

To create a test scenario, inflow disturbances must be defined. These disturbances can be, domestic or industrial waste and rainfall. It is possible to model these inflows using various modelling software, however, in the case of this project catchment dynamics are neglected. Thus, we have chosen to focus only on making an inflow dataset which resembles the flow of wastewater coming from residential areas and rainfall which comes from manholes. Data of wastewater from a residential- and industrial area in Fredericia together with data of measured rainfall will be scaled to fit the topology in the simulation. The data of the wastewater can be seen in *Figure 5.2(A)*. It can be seen that flow from residential- and industrial waste are the most intense during the mornings and evenings, and lowest during the nights. The period of the rainfall has been chosen to be the week from the 5th of August until the 11th of August, since sudden irregularities occur. Furthermore, the month of August was chosen because it was read to be the month with most rain of 2019. The data of the rainfall can be seen in *Figure 5.2(B)*.

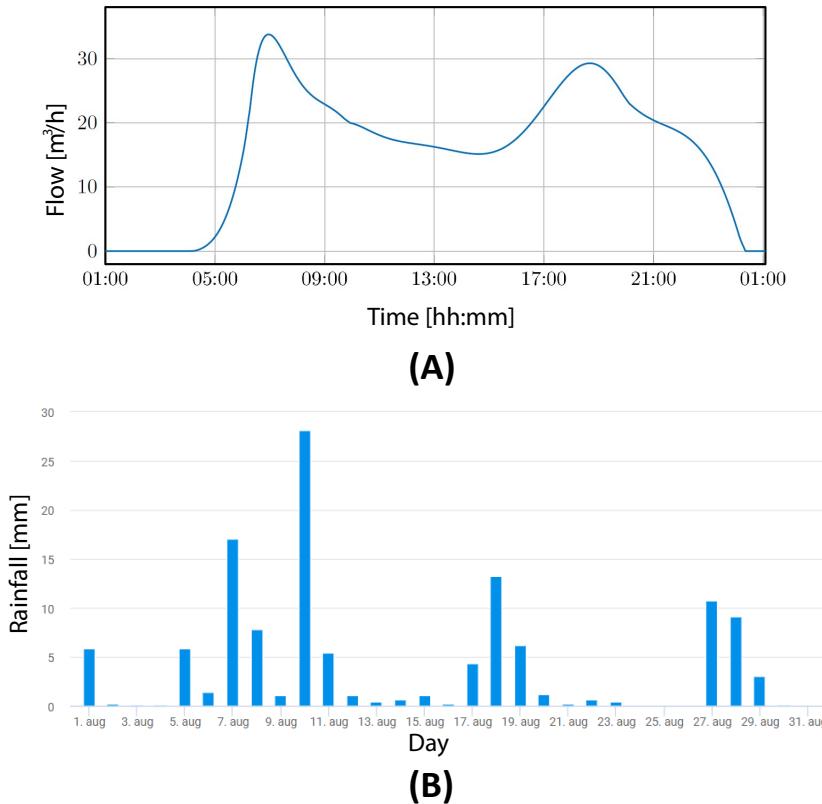


Figure 5.2: (A) shows the daily flow from a residential- and industrial area in Fredericia [1] and (B) shows the rainfall in Aalborg Kommune in August 2019 [26]

In order for the disturbance models to be used in the evaluation of the arbitrary network we select to we elect to normalize them with respect to the simulation time step $\Delta t = 12.65\text{s}$ and gain $11 \frac{\text{m}^3}{\text{s}}$. A gain factor can then later be applied to both the model of rainfall and wast water form residential area.

5.3 Control Objectives

To better get an overview during the design process, the control objectives are specified and explained. They are as follows in prioritised order:

1. Minimize CSO
2. Minimize and limit the change in inflow to the WWTP
3. Equal use of storage capacity
4. Minimize the cost of operation

The control objectives are going to be used to setup the cost function and the constraints. This cost can be based on an estimate of the economic cost of each of the objectives or used as weight to obtain the desired behavior of the MPC.

Minimizing the CSO is the main objective. As detailed in *Subsection 2.2.1 Effects of Combined Sewer Overflows*, it is expensive and highly undesired to have CSO in the system. This will have a very high cost and will have the largest weight, as it is the primary objective of the system. The second control objective is related to the fact that there is a considerable cost to changing the condition in the WWTP, therefore, a cost should be added to changing this input rate. The WWTP can handle large amounts of waste water, but as described in *Section 2.4 How WWTPs Work* there is a limit to how much a WWTP can process in a certain amount of time. Therefore, it is necessary to limit the amount of waste water which can flow into the WWTP.

The third control objective is to ensure that the retention tank will be used equally. There will be placed a cost onto different tank levels, and to storing water in general. Finally, there will be a minor cost associated with running the pumps as they have a cost in upkeep. However, this cost will likely be negligible compared to the others.

5.4 Technical Requirements

Since this project is built as a theoretical scenario, **the technical requirements are only presented to be used as a representation of what real-world requirements would look like.** This means that these requirements presented below are based on functionality requirements, but are disclosed with arbitrary values. In a real-world implementation, these requirements could be used a basis for design, as an implementation of a MPC could be done with measured values for the proposed system. The requirements are set up with placeholder values X and Y. These requirements will not be used for this project.

The following technical requirements could be set for a real system:

- **X amount of CSO at Y conditions:** This is a requirement that would be made to ensure that the system is sufficient for the application given conditions such as the expected wastewater in the context of the given system.
- **X WWTP delta flow at y conditions:** This requirement will ensure that the system can keep the change of flowrate into the WWTP within the limits in the given context.

- **X tank storage difference under y conditions:** This requirement will ensure that controller keep the water in the tanks within the expected levels in the context of the given system, under certain circumstances.
- **X cost of operation under Y conditions:** This requirement would ensure that the price of operation within the expectations for the given system under some circumstances.
- **The MPC should be X amount cheaper according to the cost functions than a controller with no prediction:** This requirement covers all the control objectives and is meant to ensure that the MPC performs better than traditional control with no prediction.

Of the technical requirements above, only the last requirement can be tested in a meaningful way without a specific system. This is because it is a comparison of different controllers on an arbitrary system. The relative performance can therefore be qualitatively evaluated.

5.5 Method of Evaluation

Since this project has been set up using a theoretical scenario, it will be difficult to formulate proper technical requirements to assess whether the system fulfills its purpose of reducing CSOs. Instead, the controllers' design will be compared to a more traditional local and non-predictive controller on the described system. The system contains similar dynamics to a real-world sewer system and the performance of the controllers on the system should be comparable to a real-life scenario. Based on this, the controllers' performance can be compared to the different control objectives and thus, the designs can be assessed based on these results. To properly test the control objective performances of the controllers, different test scenarios should be set up to evaluate the controllers under circumstances that highlight the different control objectives. This is because the different control objectives will have different relative impact depending on the scenario.

5.6 Summary

Thus the project objectives has been set. The control objectives has been set, with CSO prevention being the primary objective. The network topology has been established, it includes two tanks and a gravity pipe. The disturbances has been made from a combination of rainfall data and domestic waste data. The technical requirements will not be used to evaluate the performance. Instead, a comparison will be made between a local controller and MPC's using the two models.

Model Discretization and Linearization 6

This chapter is going to present discretization methods chosen for the pipe and the tank. The chapter begins with a short introduction about the classification of PDEs, as the full Saint-Venant equation is a PDE. Then the SV equation model is simplified, linearized and two different methods of discretization are derived. These two schemes are the Preissmann scheme and a Euler method. Furthermore, the Euler method will be applied to the discretization of the tank.

6.1 Characteristics of the Saint-Venant Equations

Partial differential equations can be categorized in three distinct types, which are analogous to the hyperbolic, parabolic and elliptic conic sections. Similar properties of PDEs can be determined to the conic sections, such as continuity, discontinuity and smoothness. As an example, elliptic PDEs generally have smooth solutions, since an ellipse describes a smooth and rounded object. Whereas, the hyperbole is a discontinuous object, so hyperbolic PDEs can describe discontinuous phenomena, such as shock waves. [27, p. 28]

Therefore, understanding the class of PDEs can help the formulation of a well-posed solution problem, with well-defined initial and boundary conditions [27, p. 25]. This in turn can guide the selection of an adequate numerical method to obtain smooth and unique solutions. Generally, the classification is based on the number of real roots of the discriminant. In this context, the classification of the Saint-Venant equations are of interest. The classification is derived in [28, p. 166] using the method of characteristics, and it is not presented in this report. The resulting two real roots of the discriminant classifies the Saint-Venant equations as hyperbolic partial differential equations. Since, hyperbolic PDEs generally describe wave phenomena, where, information travels at a finite speed in the system, no information is propagated until a wave arrives at another point of interest. This is also intuitively clear since the SV equations describe the 1D wave motion of fluids in channels.

6.1.1 Solving the Saint-Venant Equations

There are no analytical solutions for the Saint-Venant equations. For this reason numerical solutions are required [28, p. 301]. For hyperbolic PDEs, the finite difference method is widely used for discretization, especially in open channel modelling. There are several variations of the finite difference method, such as: Delft Hydraulic Laboratory scheme, Vasiliev scheme, Gunaratman-Perkins scheme, Preissmann scheme [28, p. 302].

These schemes are implicit, meaning they find a solution by solving an equation where the current state of the system and the state later are involved. On the other hand, in explicit schemes, the future state of a system is calculated only from the current state. All the mentioned schemes use the non-staggered grid, which means that in each node the values of both functions Q and h are calculated. The implicit four-point scheme is used to illustrate the grid, which is also called the box scheme, and can be seen in *Figure 6.1*.

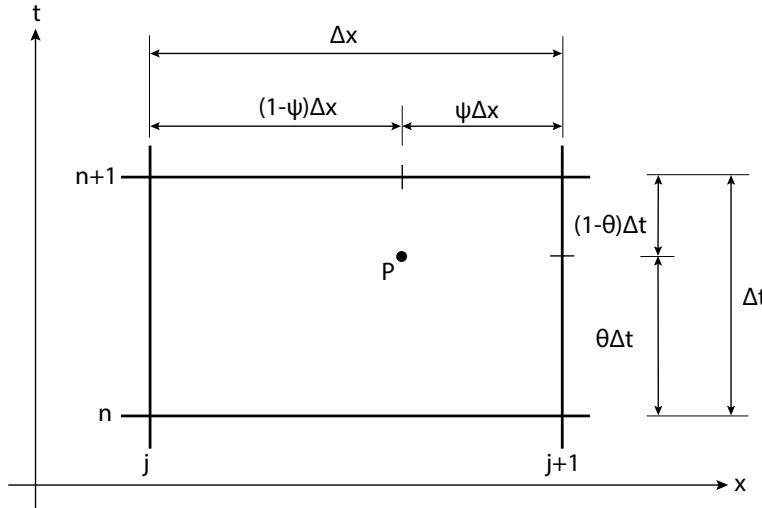


Figure 6.1: Grid points for the box scheme. Nodes are shown in each intersection of the lines.

The derivatives are approximated at point P inside the mesh. The following formulas describe the derivatives [28, p. 219]:

$$\frac{\partial f}{\partial t} \Big|_P = \psi \frac{f_j^{n+1} - f_j^n}{\Delta t} + (1 - \psi) \frac{f_{j+1}^{n+1} - f_{j+1}^n}{\Delta t} \quad (6.1)$$

$$\frac{\partial f}{\partial x} \Big|_P = \theta \frac{f_{j+1}^{n+1} - f_j^{n+1}}{\Delta x} + (1 - \theta) \frac{f_{j+1}^n - f_j^n}{\Delta x} \quad (6.2)$$

where:

f	is a scalar function.
n	is the index of time level.
j	is the index of cross-section.
Δt	is the time step.
Δx	is the spatial mesh dimension.
ψ	is a weighting parameter which ranges $\langle 0, 1 \rangle$.
θ	is a weighting parameter which ranges $\langle 0, 1 \rangle$.

Among the various numerical methods using the box scheme, the Preissmann scheme is accepted as the most robust. Some of the advantages is that it is complete stable, so it does not require limiting of the time step value. Furthermore it gives the exact solution of properly chosen values of Δt and Δx [28, p. 302]. It has been compared to the other

methods like Lax and Vasiliev scheme, where the Preissmann scheme provided the closest agreement with measured free-surface profile. However, it is important to note that this comparison was conducted on ogee spillways and not in sewer channel conditions [29]. In the following subsection the Preissmann scheme is described in more detail.

6.1.2 Preissmann Scheme

With the Preissmann scheme the approximation is carried out with P in the middle of the interval Δx , corresponding to $\psi = 0.5$. The point P can then only move along the t axis in a way controlled by the parameter θ . This parameter will be evaluated upon in *Subsection 6.4.1 Stability and Precision of the Preissmann Scheme*. The Preissmann grid scheme can be seen in *Figure 6.2*.

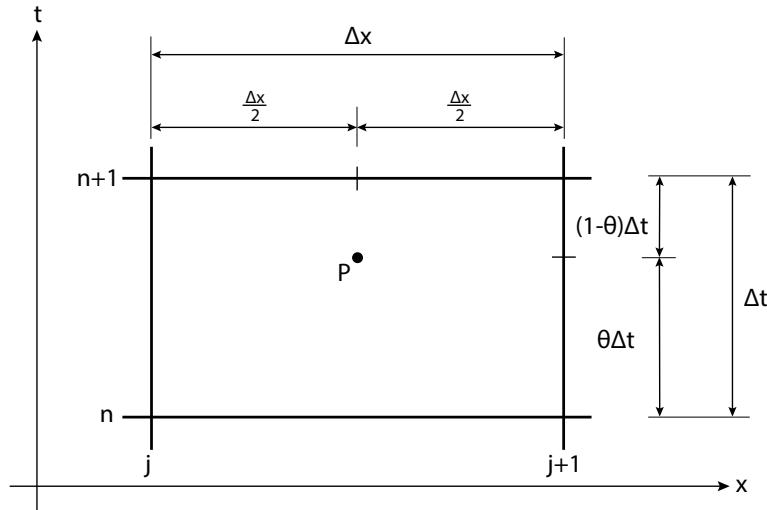


Figure 6.2: Grid points for the Preissmann scheme.

For the Preissmann scheme an arbitrary function, $f_P(x, t)$ at point P is approximated in Equation (6.3).

$$f_P \approx \frac{1}{2} \left(\theta f_j^{n+1} + (1 - \theta) f_j^n \right) + \frac{1}{2} \left(\theta f_{j+1}^{n+1} + (1 - \theta) f_{j+1}^n \right) \quad (6.3)$$

The numerical approximation for the derivatives of time and cross-section are from Equation (6.1) and Equation (6.2) with $\psi = 0.5$ are then [28, p. 304]:

$$\frac{\partial f}{\partial t} \Big|_P \approx \frac{1}{2} \left(\frac{f_j^{n+1} - f_j^n}{\Delta t} + \frac{f_{j+1}^{n+1} - f_{j+1}^n}{\Delta t} \right) \quad (6.4)$$

$$\frac{\partial f}{\partial x} \Big|_P \approx \theta \frac{f_{j+1}^{n+1} - f_j^{n+1}}{\Delta x} + (1 - \theta) \frac{f_{j+1}^n - f_j^n}{\Delta x} \quad (6.5)$$

6.2 Model Simplification

The Saint-Venant equations provide a full description of the mass and momentum conservation in distributed routing contexts and in channels. These equations are derived

in Subsection 4.1.2 Conservation of Mass and Subsection 4.1.3 Conservation of Momentum. They are presented here again:

$$\frac{\partial[A(x,t)\rho(x,t)]}{\partial t} = -\frac{\partial m(x,t)}{\partial x} \quad (6.6)$$

$$0 = -(g \cdot \rho(x,t)A(x,t))(-S_f + S_b - \frac{\partial H(x)}{\partial x}) + \frac{\partial m(x,t)}{\partial t} + \frac{\partial}{\partial x} \left(\frac{m(x,t)^2}{A(x,t)\rho(x,t)} \right) \quad (6.7)$$

The following section will go through the terms of the SV equations.

6.2.1 Volume Flow Expression and Disturbance Inflow

Before deriving the simplified models, it is useful to reformulate the equations (6.6) and (6.7) in terms of volume flow $Q = \frac{A(x,t)}{\rho}$. Equation (6.6) can be rewritten as such by moving the terms to the LHS:

$$\frac{\partial[A(x,t)\rho]}{\partial t} + \frac{\partial m(x,t)}{\partial x} = 0 \quad (6.8)$$

Note, that for the equation to be reformulated, it is necessary to assume that the density, $\rho(x,t)$ does not change in space or in time as it is assumed to be constant. In open channels it is often the procedure. Thus, dividing both sides of the equation by ρ the volume flow is substituted:

$$\frac{\partial A(x,t)}{\partial t} + \frac{\partial Q}{\partial x} = 0 \quad (6.9)$$

The conservation of momentum describes the relation between the net out flow from the control volume and the current mass in the control volume. Thus, it is possible now to add disturbance inflow $q(x,t)$ to the RHS of (6.10):

$$\frac{\partial A(x,t)}{\partial t} + \frac{\partial Q}{\partial x} = q(x,t) \quad (6.10)$$

The assumption on the density ρ is also required to express (6.7) in terms of volume flow. Moving everything to the LHS and multiplying by -1 yields:

$$-(g \cdot \rho \cdot A(x,t)) \left(-S_f + S_b - \frac{\partial H(x)}{\partial x} \right) + \frac{\partial m(x,t)}{\partial t} + \frac{\partial}{\partial x} \left(\frac{m(x,t)^2}{A(x,t)\rho} \right) = 0 \quad (6.11)$$

Expanding on parenthesis then gives:

$$-(g \cdot \rho \cdot A(x,t))S_f + (g \cdot \rho \cdot A(x,t))S_b - (g \cdot \rho \cdot A(x,t))\frac{\partial H(x)}{\partial x} + \frac{\partial m(x,t)}{\partial t} + \frac{\partial}{\partial x} \left(\frac{m(x,t)^2}{A(x,t)\rho} \right) = 0 \quad (6.12)$$

Dividing both sides by $(\rho \cdot A(x, t))$ then gives us the final equation in terms of volume flow:

$$\underbrace{\frac{1}{A(x, t)} \frac{\partial Q(x, t)}{\partial t}}_{\text{Local acceleration term}} + \underbrace{\frac{1}{A(x, t)} \frac{\partial}{\partial x} \left(\frac{Q(x, t)^2}{A(x, t)} \right)}_{\text{Convective acceleration term}} + \underbrace{g \frac{\partial h(x, t)}{\partial x}}_{\text{Pressure force term}} + \underbrace{g(S_f - S_b)}_{\text{Friction force term - Gravity force term}} = 0 \quad (6.13)$$

This form of the momentum equation is often called the dynamic wave model. There are two acceleration and velocity related terms in the dynamic wave model. These are, the local acceleration term which represents the change in momentum due to change in velocity over time and the convective acceleration term which represents the change in velocity along the channel. There are also three force terms. These are, the pressure force term which relates the change in water depth along the channel and the gravity and friction force terms which relate to the bed and the friction slope of the channel.

The model in the complete form requires a variety of precise initial conditions about the flow and the channel. These conditions are often difficult to obtain. Furthermore, in most cases using simplified models could reduce computational complexity, while still providing adequate approximations of the true flow [28, p. 367]. As concluded in [28, p. 367, 368], during the evaluation of the magnitude of the terms in the Saint-Venant equations, impact of the terms can differ up to two orders of magnitude in open channel modelling. More precisely, this means the pressure, friction and the gravity term has the most influence on the system and it is common to disregard the other terms [28, p. 368], [24, p. 281].

Therefore, neglecting the local and convective acceleration terms reduces the model to the diffusion wave. Additionally neglecting the pressure term gives the kinematic wave model. Both diffusion wave and kinematic models are often used in various distributed routing contexts. Only the diffusive wave is evaluated further upon in this report. Generally, the mass conservation model is not simplified further [24, p. 281].

6.2.2 Diffusion Wave Model

The diffusion wave model was first introduced in 1951 by Hayami [28, p. 368]. It has been successfully used to model sewer systems in RTC control related applications, as well as in overland flood simulations [30] [31].

As stated above it is the original continuity equation reduced by neglecting the acceleration terms and assuming that the gravity, friction and pressure terms govern the flow:

$$\frac{\partial A(x, t)}{\partial t} + \frac{\partial Q(x, t)}{\partial x} = 0 \quad (6.14)$$

$$\frac{\partial h(x, t)}{\partial x} + S_f - S_b = 0 \quad (6.15)$$

The momentum conservation equation reduces to a parabolic ordinary differential equation from a hyperbolic PDE. Additionally, this allows for other ODE only discretization

methods to be applied, such as Runge-Kutta methods, instead of the finite difference method previously discussed. It is also possible to solve the diffusion wave equations with analytical methods, if the celerity (characteristic velocity, or speed) and diffusion is assumed to be constant.

The remaining pressure term allows for the modeling of the backwater effect. Backwater occurs when a channel is carrying subcritical flow, meaning the flow velocity is smaller than the wave velocity. Subcritical flow is often called "slow-flow". This relation is measured with the Froude number. Intuitively, the backwater effect happens when the depth of the fluid in the channel downstream starts increasing, creating an increase upstream. It is categorized into two classes, where on one hand it is induced by lateral inflow or changes in pipe discharge capacity (change in diameter or decrease of slope) and on the other hand it can be induced by a critical depth boundary conditions. Critical depth boundary conditions in this case mean that, at the outfall of the pipe if there are subcritical conditions, where backwater can occur. This happens when the outfall is submerged, e.g.: the pipe outfalls into a filled tank. This also means that the end of the pipe is under pressure, hence the pressure term in the diffusion wave model. An example a backwater effect occurring can be seen in *Figure 6.3*.

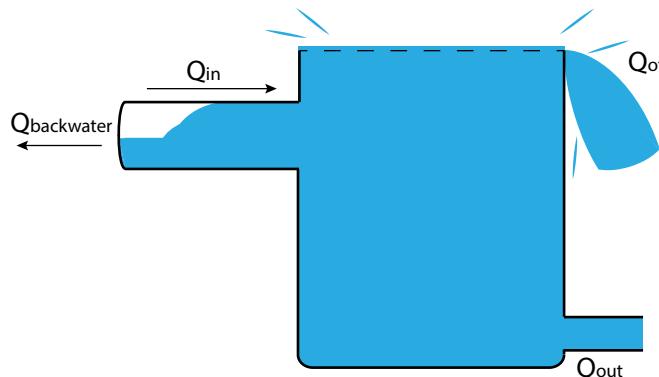


Figure 6.3: Illustration of a backwater effect occurring due to a filled tank

In this project, due to the proposed network topology we elect not to consider the later inflow related backwater effect. Only pressure related backwater is taken into account by a submerged pipe inflow into a tank. However, it is important to note here, that during subcritical flows in long channels with low slope, [30] has found that the convective acceleration term from the dynamic model has a significant effect. For this reason, in [30], has added the convective acceleration term. In the context of this report, this term is not included, and the diffusion wave model is used. Expanding the model with the convective acceleration term increases the mathematical complexity, so for the sake of simplicity it has been decided to omit this term.

6.2.3 Linearization of the Diffusion Wave Model

When solving differential equations, linearizing the model is crucial for implementing it into a linear state space form.

Beginning with the continuity equation of the diffusion wave model expressed in Equation

(6.14), $A(x, t)$ describes the wetted cross-sectional area of the channel. The rate of change of the area is depending on the shape of the channel. On *Figure 6.4*, two sketches are shown where the channel is assumed to have the shape of either a rectangular- or a circular shape.

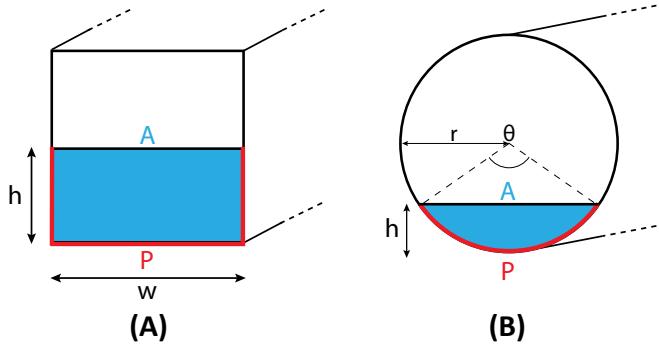


Figure 6.4: Sketch of a rectangular channel, (A), and circular channel, (B).

Assuming the channel is rectangular, (A), the area can be expressed in a simple linear equation as a function of h .

$$A(h) = wh \quad (6.16)$$

where w is the width of the channel and is assumed a constant. If the pipe is assumed to be circular, (B), the expression of the area expressed in Equation (6.17).

$$A(\theta) = \frac{r^2(\theta(h) - \sin(\theta(h)))}{2} \quad (6.17)$$

where r is the radius of the circular pipe and is assumed to be constant and $\theta(h) = 2\cos^{-1}\left(\frac{r-h}{r}\right)$.

Notice that the circular channel, (B), consists totally of three sinusoidal functions. This makes the rectangular channel, (A), the preferable model to work with, since it is already linear.

Since the width of the rectangular channel is constant the continuity equation is rewritten in Equation (6.18).

$$w \frac{\partial h(x, t)}{\partial t} + \frac{\partial Q(x, t)}{\partial x} = 0 \quad (6.18)$$

Now looking at the momentum equation in Equation (6.15), the term S_f denotes the friction loss in the channel. This friction term can be expressed with empirical equations, either the Darcy–Weisbach [32] or the Manning equation relating water height to the flow. The Manning equation is used in open, water channel design with small slopes expressed as:

$$Q = Av = AR_h^{2/3} \frac{1}{n} \sqrt{S_b} \quad (6.19)$$

where:

Q	is volume flow.	$\left[\frac{\text{m}^3}{\text{s}}\right]$
v	is the cross-sectional mean velocity.	$\left[\frac{\text{m}}{\text{s}}\right]$
A	is the cross-sectional area of the flow.	$[\text{m}^2]$
R_h	is hydraulic radius.	[m]
n	is Manning roughness coefficient.	[.]
S_b	is the slope of the channel.	[.]

The hydraulic radius is expressed as $R_h = \frac{A}{P}$.

Unlike the Manning equation, the Darcy-Weisbach equation can be used for pipe flows as well, where the flow is pressurized and surface pressure is non-uniform along the channel (for example accounting for the backwater flow). The friction term can be expressed as a function of the height and the flow using the Darcy-Weisbach equation. Note that $h = h(x, t)$ and $Q = Q(x, t)$.

$$S_f(Q, h) = f \frac{P(h)}{4A(h)^3} \frac{|Q|Q}{2g} \quad (6.20)$$

where:

h	is the height of the liquid as a function of x and t .	$[\text{m}]$
Q	is the flow of the liquid as a function of x and t .	$\left[\frac{\text{m}^3}{\text{s}}\right]$
$P(h)$	is the wetted perimeter as a function of the height of the liquid.	[m]
$A(h)$	is the cross-sectional area of the liquid as a function of the height of the liquid.	$[\text{m}^2]$
f	is Darcy-Weisbach friction factor for turbulent flow.	[.]
g	is the gravitational acceleration.	$\left[\frac{\text{m}^2}{\text{s}}$

By assuming a rectangular shape of the channel, the wetted cross-sectional area and wetted perimeter becomes: $A(h) = wh$ and $P(h) = w + 2h$.

$$S_f(Q, h) = f \frac{w + 2h}{4(wh)^3} \frac{|Q|Q}{2g} \quad (6.21)$$

A new constant is introduced: $\alpha = \frac{f}{2g}$. The S_f -function is not a linear function, since it contains the terms h^3 and $|Q|Q$. $S_f(Q, h)$ is linearized around the operating points Q_{op}, h_{op} . This is approximated with a Taylor expansion where only the first order derivatives are used. The expression can be seen in Equation (6.22).

$$\begin{aligned}
S_f(Q, h) &\approx \\
S_f((Q)_{op}, (h)_{op}) + \frac{\partial S_f(Q, h)}{\partial Q} \Big|_{op} (Q - Q_{op}) + \frac{\partial S_f(Q, h)}{\partial h} \Big|_{op} (h - h_{op}) & \quad (6.22) \\
&\approx \underbrace{\alpha \frac{w + 2h_{op}}{4w^3h_{op}^3} |Q_{op}| Q_{op} - \alpha \frac{w + 2h_{op}}{4w^3h_{op}^3} \frac{2Q_{op}^3}{|Q_{op}|} - \alpha \frac{-4h_{op} - 3w}{2w^3h_{op}^3} |Q_{op}| Q_{op}}_{\gamma_1(Q_{op}, h_{op})} \\
&\quad + \underbrace{\alpha \frac{w + 2h_{op}}{4w^3h_{op}^3} \frac{2Q_{op}^2}{|Q_{op}|} Q}_{\gamma_2(Q_{op}, h_{op})} + \underbrace{\alpha \frac{-4h_{op} - 3w}{2w^3h_{op}^4} |Q_{op}| Q_{op} h}_{\gamma_3(Q_{op}, h_{op})} \quad (6.23)
\end{aligned}$$

In Equation (6.24) it can be seen that the new expression for $S_f(Q, h)$ is linear.

$$S_f(Q, h) \approx \gamma_1 + \gamma_2 Q + \gamma_3 h \quad (6.24)$$

The linearized Saint-Venant equations of the diffusion wave model from 6.14 and 6.15 can be seen in 6.25 and 6.26.

$$w \frac{\partial h}{\partial t} + \frac{\partial Q}{\partial x} = 0 \quad (6.25)$$

$$\frac{\partial h}{\partial x} + \gamma_1 + \gamma_2 Q + \gamma_3 h - S_b = 0 \quad (6.26)$$

6.3 Model Discretization Using the Euler Scheme

The Euler discretization method will be applied to two different system components, the pipe and the tank. Though the Euler method is well known, a short overview of the concept is given here.

The Euler method is the simplest version of the Runge-Kutta method, used to numerically approximate a derivative in ODEs. Given a point x_j , for which the derivative can be approximated, Euler proposes estimating the derivative by using a point before x_{j-1} or after x_{j+1} and dividing by the distance between them Δx . This is often denoted as:

$$\begin{aligned}
\frac{d y_j}{dx} &\approx \frac{y_{j+1} - y_j}{\Delta x} \\
&\text{or} \\
\frac{d y_j}{dx} &\approx \frac{y_j - y_{j-1}}{\Delta x} \quad (6.27)
\end{aligned}$$

Known respectively as forward and backward Euler approximations, each useful in different cases.

6.3.1 Euler Pipe Model

As explained in *Section 4.1 Pipe model*, the two Saint-Venant equations describe the behaviour of fluid as it is flowing through a pipe. Often a longer piece of pipe is divided

into sections, because conditions such as flow and water height change along the pipe, see *Figure 6.5 a*. Thinking of this in terms of the Euler approximation, if the two points are the ends of the pipe section, the shorter the pipe section the better the approximation.

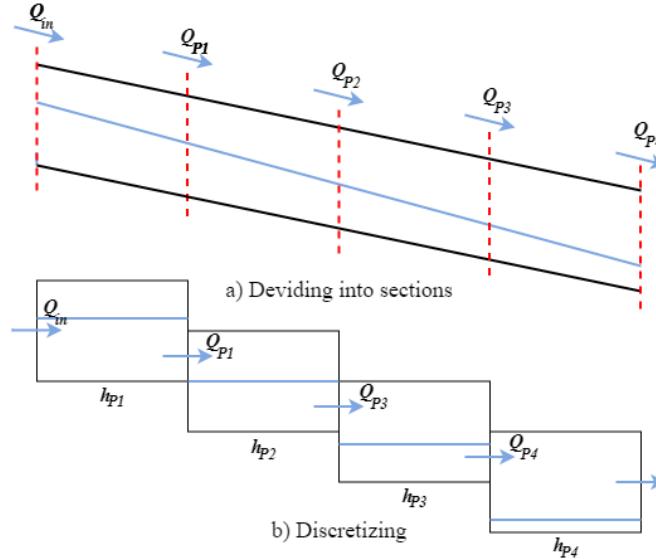


Figure 6.5: Partitioning and discretization of pipe as part of Euler approximation

The full non-linear SV equations are not useful for control which is why they are linearized in (6.25) and (6.26), and now discretized. The goal of the discretization is to construct a discrete state space representation of the pipe, where the state is the height of fluid in every section of the pipe.

The first step is to consider the flow into and out of a pipe section, this is described by (6.25). A forward Euler approximation can be applied to describe the change in fluid level at the next time step:

$$\begin{aligned}
 w \frac{\partial h}{\partial t} + \frac{\partial Q}{\partial x} &= 0 \\
 \Downarrow \\
 w \frac{h_j^{n+1} - h_j^n}{\Delta t} + \frac{\partial Q}{\partial x} &= 0 \\
 \Updownarrow \\
 h_j^{n+1} &= \frac{\Delta t}{w} \left(-\frac{\partial Q}{\partial x} \right) + h_j^n
 \end{aligned} \tag{6.28}$$

As shown in *Figure 6.5 b*, the fluid height in a pipe section is dependent on the pipe section outflow Q_k and its inflow Q_{k-1} . This makes the Backward Euler the obvious choice for

approximating the change in flow:

$$\begin{aligned}
 h_j^{n+1} &= \frac{\Delta t}{w} \left(-\frac{\partial Q}{\partial x} \right) + h_j^n \\
 &\Downarrow \\
 h_j^{n+1} &= \frac{\Delta t}{w} \left(\frac{-(Q_j^n - Q_{j-1}^n)}{\Delta x_b} \right) + h_j^n \\
 &= \frac{\Delta t}{w \cdot \Delta x_b} (Q_{j-1}^n - Q_j^n) + h_j^n
 \end{aligned} \tag{6.29}$$

Equation (6.26) describes the flow through the pipe so the equation will be discretized for each pipe section. The forward Euler approach is used to approximate the change in height spatial-wise:

$$\begin{aligned}
 \frac{\partial h}{\partial x} + \gamma_1 + \gamma_2 Q + \gamma_3 h - S_b &= 0 \\
 &\Downarrow \\
 \frac{h_{j+1}^n - h_j^n}{\Delta x_f} + \gamma_1 + \gamma_2 Q_j^n + \gamma_3 h_j^n - S_b &= 0 \\
 &\Updownarrow \\
 Q_j^n &= \frac{1}{\gamma_2} \left(h_j^n \left(\frac{1}{\Delta x_f} - \gamma_3 \right) - h_{j+1}^n \frac{1}{\Delta x_f} + S_b - \gamma_1 \right) \\
 &= h_j^n \left(\frac{1}{\Delta x_f \gamma_2} - \frac{\gamma_3}{\gamma_2} \right) - h_{j+1}^n \frac{1}{\Delta x_f \gamma_2} + \frac{S_b - \gamma_1}{\gamma_2}
 \end{aligned} \tag{6.30}$$

$$\tag{6.31}$$

The flow between each pipe section, have now been described. This yields a partial state space description, which include the flow between each pipe section. The state space representation is shown in Equation (6.32).

$$\begin{bmatrix} h_{j-1}^{n+1} \\ h_j^{n+1} \\ h_{j+1}^{n+1} \end{bmatrix} = \begin{bmatrix} \left(\frac{\alpha \gamma_3}{\gamma_2} - \frac{\alpha}{\Delta x_f \gamma_2} \right) & \left(1 - \frac{2\alpha}{\Delta x_f \gamma_2} + \frac{\alpha \gamma_3}{\gamma_2} \right) & \frac{\alpha}{\Delta x_f \gamma_2} \end{bmatrix} \begin{bmatrix} h_{j-1}^n \\ h_j^n \\ h_{j+1}^n \end{bmatrix} \tag{6.32}$$

To have a full state space representation, the flow at each end of the pipe needs to be described. The flows at the ends of the pipe are called boundary conditions. The boundary conditions are dependent on what the pipe is connected to, such as pumps or tanks. Pumps are often modeled as constant flow providers or control inputs, while the boundary conditions from pipe to tank can be modeled in multiple ways. Two ways of this boundary conditions will be presented in the following subsection.

6.3.2 Euler Tank Model

In *Section 4.2 Retention Tank* the mass balance for a tank was presented as (4.3), stated again below:

$$\frac{dh(t)}{dt} = \frac{1}{A} (Q_{in}(t) - Q_{out}(t) - Q_{of}(t))$$

Applying the forward Euler approximation yields a result that is close to identical to (6.29):

$$\begin{aligned} \frac{h_j^{n+1} - h_j^n}{\Delta t} &= \frac{1}{A}(Q_{in}^n - Q_{out}^n - Q_{of}^n) \\ &\Downarrow \\ h_j^{n+1} &= \frac{\Delta t}{A}(Q_{in}^n - Q_{out}^n - Q_{of}^n) + h_j^n \end{aligned} \quad (6.33)$$

The only difference between (6.33) and the equivalent pipe equation is Q_{of} which is an additional outflow. This can intuitively be understood as if it were modelled as "another" pipe section. Note that the tank has three boundary conditions, the inflow Q_{in}^n , the outflow Q_{out}^n and the overflow Q_{of}^n . A partial state space description can be made for the tank, this description will depend on how the different flows are handled. More on this in 7.1.

6.3.3 Boundary Conditions

From the previous subsections it is known that boundary conditions for the pipe and the tank are needed. Two different approaches to modeling boundary conditions will be presented below, making the total number of boundary conditions presented in this report three. Note, the third boundary condition is the one related to a pipe or tank being connected to a pump.

Free flow or weir flow can be used as a model for inflow and outflow from a tank and outflow from a pipe [33, p. 161]. The expression for free flow is not dependent on the height of fluid downstream:

$$Q_j^n = C_d \sqrt{2g} \cdot (h_j^n)^{3/2} \quad (6.34)$$

Where C_d is a discharge coefficient and g is the gravitational acceleration. Note that, Equation (6.34) would typically be used to describe flow when downstream backwater effects are negligible or non-existent. An example is illustrated in *Figure 6.6 a* where the outflow from a pipe goes into the top of a tank, while overflow leaves at a lower point.

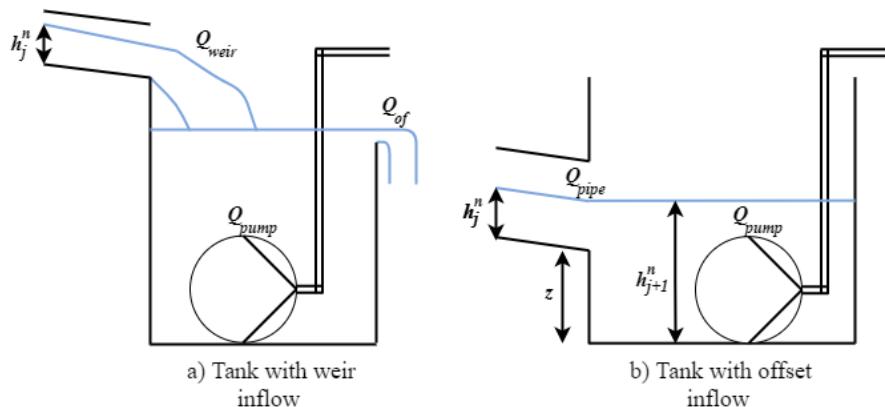


Figure 6.6: Two tanks inputs presented as boundary conditions

To use (6.34) as a boundary condition it is first necessary to linearize it. The first order Taylor approximation, same approach used in *Subsection 6.2.3 Linearization of the Diffusion Wave Model*, will again be used to linearize (6.34).

$$\begin{aligned}
Q_j^n &\approx C_d \sqrt{2g} \cdot h_{op}^{3/2} + \frac{d}{dh_j^n} \left[C_d \sqrt{2g} \cdot (h_j^n)^{3/2} \right] \Big|_{h_{op}} (h_j^n - h_{op}) \\
&\Downarrow \\
Q_{weir} &= \underbrace{\frac{3C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \cdot h_j^n}_{\mu_2} - \underbrace{\frac{C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2}}_{\mu_1} \\
&= \mu_1 + \mu_2 \cdot h_j^n
\end{aligned} \tag{6.35}$$

With Equation (6.35) the two possible boundary conditions have been described.

Offset flow is an idea based on the observation that was made previously: the mass balance of a pipe and a tank are very alike. A proposed modified form of (6.31), when a pipe outflow is connected to a tank inflow. Adding a term for the relative height difference between the bottom of the pipe Z and the bottom of the tank results in the following equation:

$$Q_{pipe}^n = \frac{1}{\gamma_2} \left[h_{pipe}^n \left(\frac{1}{\Delta x_f} - \gamma_3 \right) - (h_{tank}^n - Z) \frac{1}{\Delta x_f} + S_b - \gamma_1 \right] \tag{6.36}$$

The setup is illustrated on *Figure 6.6 b*. The idea is to allow for backwater from the tank to the pipe, in cases where such backwater is relevant. The approximation results in an additional constant term.

6.4 Model Discretization Using the Preissmann Scheme

In Section 6.2.2, the Saint-Venant equations are presented in a simplified form of a diffusion wave function. Here they are presented in Equation (6.37) and Equation (6.38).

$$w \frac{\partial h}{\partial t} + \frac{\partial Q}{\partial x} = 0 \tag{6.37}$$

$$\frac{\partial h}{\partial x} + \gamma_1 + \gamma_2 Q + \gamma_3 h - S_b = 0 \tag{6.38}$$

The numerical solution to the simplified Saint-Venant equations using the Preissmann scheme can be seen in Equation (6.39) and Equation (6.40).

$$\frac{w}{2} \left(\frac{h_j^{n+1} - h_j^n}{\Delta t} + \frac{h_{j+1}^{n+1} - h_{j+1}^n}{\Delta t} \right) + \theta \frac{Q_{j+1}^{n+1} - Q_j^{n+1}}{\Delta x} + (1 - \theta) \frac{Q_{j+1}^n - Q_j^n}{\Delta x} = 0 \tag{6.39}$$

$$\theta \frac{h_{j+1}^{n+1} - h_j^{n+1}}{\Delta x} + (1 - \theta) \frac{h_{j+1}^n - h_j^n}{\Delta x} + \gamma_1 + \gamma_2 Q + \gamma_3 h - S_b = 0 \tag{6.40}$$

The Preissmann approximation in Equation (6.3) is inserted in Q and h in Equation (6.40):

$$\begin{aligned} & \theta \frac{h_{j+1}^{n+1} - h_j^{n+1}}{\Delta x} + (1-\theta) \frac{h_{j+1}^n - h_j^n}{\Delta x} + \gamma_1 + \gamma_2 \underbrace{\frac{1}{2} \left((1-\theta)(Q_j^n + Q_{j+1}^n) + \theta(Q_j^{n+1} + Q_{j+1}^{n+1}) \right)}_{Q(x,t)} \\ & + \gamma_3 \underbrace{\frac{1}{2} \left((1-\theta)(h_j^n + h_{j+1}^n) + \theta(h_j^{n+1} + h_{j+1}^{n+1}) \right)}_{h(x,t)} - S_b = 0 \end{aligned} \quad (6.41)$$

Equation (6.39) and Equation (6.41) is being factorized and isolated, such that the future terms of Q and h are on the left hand side of the equation. This is done to easier put it into a state space form.

$$\frac{w}{2\Delta t}(h_j^{n+1} + h_{j+1}^{n+1}) + \frac{\theta}{\Delta x}(Q_j^{n+1} - Q_{j+1}^{n+1}) = \frac{w}{2\Delta t}(h_j^n + h_{j+1}^n) + \frac{(1-\theta)}{\Delta x}(Q_j^n - Q_{j+1}^n) \quad (6.42)$$

$$\begin{aligned} & \left(\frac{\gamma_3\theta}{2} - \frac{\theta}{\Delta x} \right) h_j^{n+1} + \left(\frac{\gamma_3\theta}{2} + \frac{\theta}{\Delta x} \right) h_{j+1}^{n+1} + \frac{\gamma_2\theta}{2}(Q_j^{n+1} + Q_{j+1}^{n+1}) \\ & = \left(\frac{(1-\theta)}{\Delta x} - \frac{\gamma_3(1-\theta)}{2} \right) h_j^n - \left(\frac{(1-\theta)}{\Delta x} + \frac{\gamma_3(1-\theta)}{2} \right) h_{j+1}^n - \frac{\gamma_2(1-\theta)}{2}(Q_j^n + Q_{j+1}^n) - \gamma_1 + S_b \end{aligned} \quad (6.43)$$

Lastly the equations are put in a state space form.

$$\begin{aligned} & \begin{bmatrix} \frac{w}{2\Delta t} & -\frac{\theta}{\Delta x} & \frac{w}{2\Delta t} & \frac{\theta}{\Delta x} \\ \left(\frac{\gamma_3\theta}{2} - \frac{\theta}{\Delta x} \right) & \frac{\gamma_2\theta}{2} & \left(\frac{\gamma_3\theta}{2} + \frac{\theta}{\Delta x} \right) & \frac{\gamma_2\theta}{2} \end{bmatrix} \begin{bmatrix} h_j^{n+1} \\ Q_j^{n+1} \\ h_{j+1}^{n+1} \\ Q_{j+1}^{n+1} \end{bmatrix} \\ & = \begin{bmatrix} \frac{w}{2\Delta t} & \frac{(1-\theta)}{\Delta x} & \frac{w}{2\Delta t} & -\frac{(1-\theta)}{\Delta x} \\ \left(\frac{(1-\theta)}{\Delta x} - \frac{\gamma_3(1-\theta)}{2} \right) & -\frac{\gamma_2(1-\theta)}{2} & -\left(\frac{(1-\theta)}{\Delta x} + \frac{\gamma_3(1-\theta)}{2} \right) & -\frac{\gamma_2(1-\theta)}{2} \end{bmatrix} \begin{bmatrix} h_j^n \\ Q_j^n \\ h_{j+1}^n \\ Q_{j+1}^n \end{bmatrix} + \begin{bmatrix} 0 \\ -\gamma_1 + S_b \end{bmatrix} \end{aligned} \quad (6.44)$$

Since the Preissmann scheme is dependent on future terms there can only be established $2(M - 1)$ equations with $2M$ unknown variables, where M denotes the number of sections there are in the channel. This means that two boundary equations are necessary to solve the system of equations.

The boundary equations include what the predicted flow of the water is coming into and out of the channel, and what the predicted height of the water is coming into and out of the channel. To establish the boundary equations, defining the beginning and end of the channel is necessary. The system is solved for $0 \leq x \leq L$, where x is the spatial coordinate and L is the length of the channel.

$$\delta_0 h_1^{n+1} + (1 - \delta_0) Q_1^{n+1} = \delta_0 h_0(t_{n+1}) + (1 - \delta_0) Q_0(t_{n+1}) \quad (6.45)$$

$$\delta_L h_M^{n+1} + (1 - \delta_L) Q_L^{n+1} = \delta_L h_L(t_{n+1}) + (1 - \delta_L) Q_L(t_{n+1}) \quad (6.46)$$

where:

$h_0(t_{n+1})$	is the predicted height of the water into the channel at the next time step.
$Q_0(t_{n+1})$	is the predicted inflow of water into the channel at the next time step.
$h_L(t_{n+1})$	is the predicted height of water at the end of the channel at the next time step.
$Q_L(t_{n+1})$	is the predicted outflow of water at the end of the channel at the next time step.
δ_0	is an integer number which can be either 0 or 1.
δ_L	is an integer number which can be either 0 or 1.

Note that for the beginning and end of the channel only one predicted term is necessary to solve the system of equations. To have the predicted term to be height or flow, δ is set to 1 or 0 respectively [28, p. 305].

An illustration of the channel and its divided sections can be seen in *Figure 6.7*, where $Q_0(t_{n+1})$ and $h_L(t_{n+1})$ is shown.

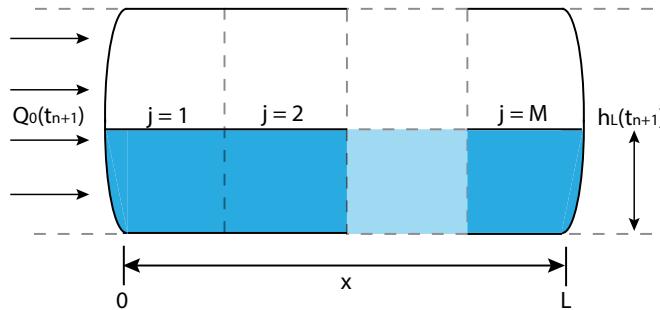


Figure 6.7: Illustration of a channel divided into sections

With the boundary equations defined the system of equations can be seen below.

$$\left. \begin{array}{l} F_1 \left(h_1^{n+1}, Q_1^{n+1} \right) = \delta_0 h_1^{n+1} + (1 - \delta_0) Q_1^{n+1} - \delta_0 h_0(t_{n+1}) - (1 - \delta_0) Q_0(t_{n+1}) = 0 \\ F_{2j-1} \left(h_j^{n+1}, Q_j^{n+1}, h_{j+1}^{n+1}, Q_{j+1}^{n+1} \right) = 0 \\ F_{2j} \left(h_j^{n+1}, Q_j^{n+1}, h_{j+1}^{n+1}, Q_{j+1}^{n+1} \right) = 0 \\ \vdots \\ F_{2M} \left(h_M^{n+1}, Q_M^{n+1} \right) = \delta_L h_M^{n+1} + (1 - \delta_L) Q_M^{n+1} - \delta_L h_L(t_{n+1}) - (1 - \delta_L) Q_L(t_{n+1}) = 0 \end{array} \right\} \quad j = 1, \dots, M-1$$

The general form of the Preissmann scheme with boundary equations will produce a $2M \times 2M$ matrix with a diagonal pattern produced by the Saint-Venant equations with $1 \times 2M$ states.

$$\left[\begin{array}{cccc} \bullet & \bullet & & \\ \bullet & \bullet & \bullet & \bullet \\ \ddots & \ddots & \ddots & \ddots \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{array} \right] \left[\begin{array}{c} h_j^{n+1} \\ Q_j^{n+1} \\ h_{j+1}^{n+1} \\ Q_{j+1}^{n+1} \\ h_{j+2}^{n+1} \\ Q_{j+2}^{n+1} \\ \vdots \\ h_{M-1}^{n+1} \\ Q_{M-1}^{n+1} \\ h_M^{n+1} \\ Q_M^{n+1} \end{array} \right] = \left[\begin{array}{cccc} \bullet & \bullet & & \\ \bullet & \bullet & \bullet & \bullet \\ \ddots & \ddots & \ddots & \ddots \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{array} \right] \left[\begin{array}{c} h_j^n \\ Q_j^n \\ h_{j+1}^n \\ Q_{j+1}^n \\ h_{j+2}^n \\ Q_{j+2}^n \\ \vdots \\ h_{M-1}^n \\ Q_{M-1}^n \\ h_M^n \\ Q_M^n \end{array} \right]$$

6.4.1 Stability and Precision of the Preissmann Scheme

The Preissmann scheme applied for solving the Saint-Venant equation is stable for any Courant number which is defined in (6.47).

$$C_r = \frac{\sqrt{g \cdot \bar{H} \Delta t}}{\Delta x} \quad (6.47)$$

On the condition that:

$$\theta \geq 0.5 \quad (6.48)$$

where C_r is the dimensionless Courant number and \bar{H} is the average height of the liquid in the channel. Based on Romuald Szymkiewicz in [28, p. 312], the Preissmann scheme provides an exact solution when $C_r = 1$ and $\theta = 0.5$. This results in a sharp discontinuity in the solution and is unrealistic in a true sewer realization. To compensate for this, some numerical diffusion is introduced into the solution. By setting $C_r = 0.5$ and $\theta \approx 0.67$ as [28, p. 313] suggests, the results of calculations should be rather smooth without "too strong smoothing."

6.5 Conclusion

In this chapter, two linear models for the gravity pipe have been developed. The Saint-Venant equations are numerically solved using two different discretization methods. All network elements are now in the linear form which makes it suitable for standard state space representation of the entire system and linear MPC design. This is further elaborated in the next chapter.

System and Controller

Model

7

In the previous chapter components of a urban drainage system was linearized and discretized. In this chapter we are going to use the new description of these components to create two linearized system models to be used in an MPC design. Furthermore, the structure of an MPC is going to be expanded upon, for later use in a software implementation. We conclude the chapter with a section on model parameterization for the system model to be used in simulation, alongside MPC.

7.1 State Space Models

In *Subsection 6.3.1 Euler Pipe Model* it was mentioned that a dividing a long piece of pipe into smaller sections, provided increased precision for our linearized models. In *Figure 5.1* the pipe connecting tank numbered T_2 two and pump numbered $FP1$ one is viewed as a single piece. To increase precision we elect to divide the pipe into five equally sized sections. As shown on *Figure 7.1*.

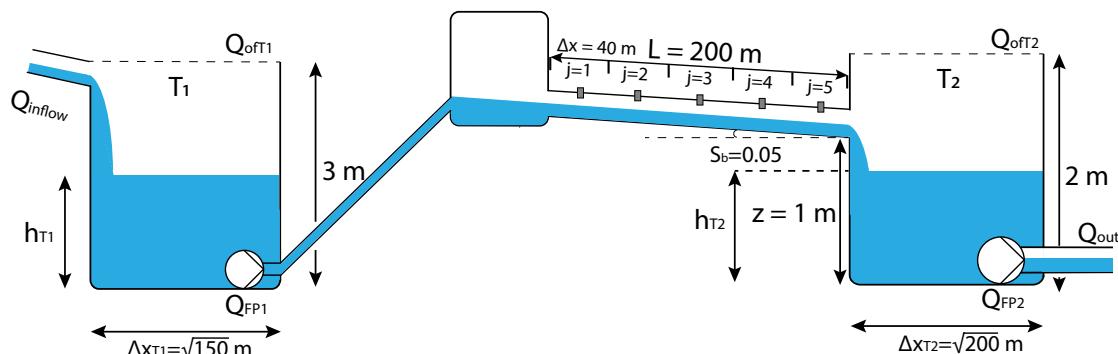


Figure 7.1: Topology of AAU water-lab

From this "divided" system two different models are developed, to be used in a MPC. Both linerized models can be represented on the form:

$$x(n+1) = Ax(n) + Bu(n) + B_d u_d(n) + \mathcal{C}_{\Delta x} \quad (7.1)$$

Where A is the system matrix, B is the input matrix and $\mathcal{C}_{\Delta x}$ is a vector resulting in a constant change. Furthermore the inflow of sewage into the models dictated in the requirements, is included using the $u_d(n)$. Normally state space model also include a C and D matrices but as we assume $C = I$ and $D = 0$ these become redundant.

7.1.1 Euler state space model

A model is created using the system components discretized by use of the Euler method, described in *Section 6.3 Model Discretization Using the Euler Scheme*.

The first tank is modeled using the Euler tank model and the five pipe sections are modeled using the Euler pipe model. The last tank can be modeled in two ways dependant on at which height the pipe enters the tank, see *Subsection 6.3.3 Boundary Conditions*. As the pipe enters the tank at a $1m$ offset, shown on figure 7.1, the logical choice of boundary condition would be the offset flow. However due to the fact that we are approaching the design of this MPC as a comparison study we elect to create two different models each employing a different boundary condition. The model using the offset boundary condition is expressed as:

$$\begin{aligned}
 \underbrace{\begin{bmatrix} h_{T1}^{n+1} \\ h_1^{n+1} \\ h_2^{n+1} \\ h_3^{n+1} \\ h_4^{n+1} \\ h_5^{n+1} \\ h_{T2}^{n+1} \end{bmatrix}}_{x(n+1)} &= \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \left(1 - \frac{\alpha_P}{\Delta x_P \gamma_2} + \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \frac{\alpha_P}{\Delta x_P \gamma_2} & 0 & 0 & 0 \\ 0 & \left(\frac{\alpha_P}{\Delta x_P \gamma_2} - \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \left(1 - \frac{2\alpha_P}{\Delta x_P \gamma_2} + \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \frac{\alpha_P}{\Delta x_P \gamma_2} & 0 & 0 \\ 0 & 0 & \left(\frac{\alpha_P}{\Delta x_P \gamma_2} - \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \left(1 - \frac{2\alpha_P}{\Delta x_P \gamma_2} + \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \frac{\alpha_P}{\Delta x_P \gamma_2} & 0 \\ 0 & 0 & 0 & \left(\frac{\alpha_P}{\Delta x_P \gamma_2} - \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \left(1 - \frac{2\alpha_P}{\Delta x_P \gamma_2} + \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \frac{\alpha_P}{\Delta x_P \gamma_2} \\ 0 & 0 & 0 & 0 & \left(\frac{\alpha_P}{\Delta x_P \gamma_2} - \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \left(1 - \frac{\alpha_P}{\Delta x_P \gamma_2} - \frac{\alpha_T_2}{\Delta x_P \gamma_2} + \frac{\alpha_P \gamma_3}{\gamma_2}\right) \\ 0 & 0 & 0 & 0 & 0 & \left(\frac{\alpha_P}{\Delta x_P \gamma_2} - \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \left(1 - \frac{\alpha_P}{\Delta x_P \gamma_2}\right) \end{bmatrix}}_A \underbrace{\begin{bmatrix} h_{T1}^n \\ h_1^n \\ h_2^n \\ h_3^n \\ h_4^n \\ h_5^n \\ h_{T2}^n \end{bmatrix}}_{x(n)} \\
 + \underbrace{\begin{bmatrix} -\alpha_{T1} & 0 & -\alpha_{T1} & 0 \\ \alpha_P & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -\alpha_{T2} & 0 & -\alpha_{T2} \end{bmatrix}}_B \underbrace{\begin{bmatrix} Q_{FP1}^n \\ Q_{FP2}^n \\ Q_{offT1}^n \\ Q_{offT2}^n \end{bmatrix}}_{u(n)} &+ \underbrace{\begin{bmatrix} \alpha_{T1} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{B_d} \underbrace{\begin{bmatrix} Q_{inflow} \\ u_d(n) \end{bmatrix}}_{\tilde{u}_d} + \underbrace{\begin{bmatrix} 0 \\ -\alpha_P \frac{(S_b - \gamma_1)}{\gamma_2} \\ 0 \\ 0 \\ 0 \\ -\frac{Z \alpha_P}{\Delta x_{T2} \cdot \gamma_2} \\ -\frac{Z \alpha_P}{\Delta x_{T2} \cdot \gamma_2} + \alpha_P \frac{(S_b - \gamma_1)}{\gamma_2} \end{bmatrix}}_{C_{\Delta x}} \quad (7.2)
 \end{aligned}$$

and the model using the free flow boundary condition is expressed as:

$$\begin{aligned}
 \underbrace{\begin{bmatrix} h_{T1}^{n+1} \\ h_1^{n+1} \\ h_2^{n+1} \\ h_3^{n+1} \\ h_4^{n+1} \\ h_5^{n+1} \\ h_{T2}^{n+1} \end{bmatrix}}_{x(n+1)} &= \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \left(1 - \frac{\alpha_P}{\Delta x_P \gamma_2} + \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \frac{\alpha_P}{\Delta x_P \gamma_2} & 0 & 0 & 0 \\ 0 & \left(\frac{\alpha_P}{\Delta x_P \gamma_2} - \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \left(1 - \frac{2\alpha_P}{\Delta x_P \gamma_2} + \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \frac{\alpha_P}{\Delta x_P \gamma_2} & 0 & 0 \\ 0 & 0 & \left(\frac{\alpha_P}{\Delta x_P \gamma_2} - \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \left(1 - \frac{2\alpha_P}{\Delta x_P \gamma_2} + \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \frac{\alpha_P}{\Delta x_P \gamma_2} & 0 \\ 0 & 0 & 0 & \left(\frac{\alpha_P}{\Delta x_P \gamma_2} - \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \left(1 - \frac{2\alpha_P}{\Delta x_P \gamma_2} + \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \frac{\alpha_P}{\Delta x_P \gamma_2} \\ 0 & 0 & 0 & 0 & \left(\frac{\alpha_P}{\Delta x_P \gamma_2} - \frac{\alpha_P \gamma_3}{\gamma_2}\right) & \left(-\alpha_P \mu_2 - \frac{\alpha_P}{\Delta x_P \gamma_2}\right) \\ 0 & 0 & 0 & 0 & 0 & \alpha_P \mu_2 \end{bmatrix}}_A \underbrace{\begin{bmatrix} h_{T1}^n \\ h_1^n \\ h_2^n \\ h_3^n \\ h_4^n \\ h_5^n \\ h_{T2}^n \end{bmatrix}}_{x(n)} \\
 + \underbrace{\begin{bmatrix} -\alpha_{T1} & 0 & -\alpha_{T1} & 0 \\ \alpha_P & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -\alpha_{T2} & 0 & -\alpha_{T2} \end{bmatrix}}_B \underbrace{\begin{bmatrix} Q_{FP1}^n \\ Q_{FP2}^n \\ Q_{offT1}^n \\ Q_{offT2}^n \end{bmatrix}}_{u(n)} &+ \underbrace{\begin{bmatrix} \alpha_{T1} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{B_d} \underbrace{\begin{bmatrix} Q_{inflow} \\ u_d(n) \end{bmatrix}}_{\tilde{u}_d} + \underbrace{\begin{bmatrix} 0 \\ -\alpha_P \frac{(S_b - \gamma_1)}{\gamma_2} \\ 0 \\ 0 \\ 0 \\ \alpha_P \frac{(S_b - \gamma_1)}{\gamma_2} - \alpha_P \mu_1 \end{bmatrix}}_{C_{\Delta x}} \quad (7.3)
 \end{aligned}$$

where:

h	is the height of the liquid as a function of x and t .	[m]
Q	is the flow of the liquid as a function of x and t .	[$\frac{m^3}{s}$]
n	is the temporal index .	[.]
j	is the spatial index.	[.]
α	is a flow to height conversion factor of a given section.	[$\frac{m}{t}$]
Δx	is the spatial length of a given section.	[m]
γ	is a constant based on the OPs Q_{OP} and h_{OP} .	[.]
Z	is the height where the gravity pipe intersects with second tank, $T2$.	[m]
S_b	is the slope of the gravity pipe.	[.]
μ_1	is a constant flow created by the flow boundary condition.	[$\frac{m^3}{s}$]
μ_2	is a factor created by the free flow boundary condition.	[$\frac{m^3}{s}$]

Note: $u(n)$ includes the two terms Q_{ofT1}^n and Q_{ofT2}^n in addition to the pump control variables Q_{FP1}^n and Q_{FP2}^n . This makes placing a penalty on the CSO in our cost function possible. More on this in *Subsection 3.2.2 Model Predictive Control (MPC)*. Including the overflow as a control variable intuitively makes sense as the tanks height can be expressed as:

$$h_j^{n+1} = \frac{\Delta t}{A} (Q_{in}^n - Q_{FPx}^n - Q_{of}^n) + h_j^n \quad (7.4)$$

Another observation can be made in relation to the factor α it is simply the conversion factor between flow and height. Whether α has a positive or negative sign, is based on whether the flow is going into the tank or out of the tank.

7.1.2 Preissmann state space model

The Preissmann scheme is likewise put up in a state space model, however this model deviates from the standard by being expressed as Equation (7.5).

$$\epsilon x(n+1) = \epsilon (Ax(n) + Bu(n) + B_d u_d(n) + C_{\Delta}x) \quad (7.5)$$

We have chosen to express it this way as it relates closely to Equation (6.44).

The state space model for the full system can be seen in Equation (7.6). The pipe sections are described using Equation (6.44). Like the last tank, T_2 , is considered as another section of the gravity pipe, so the boundary condition, Equation (6.46), is applied to tank T_2 (δ_0 and δ_L is set to 0).

$$\begin{aligned}
 & \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{w}{2\Delta t} & -\frac{\theta}{\Delta x} & \frac{w}{2\Delta t} & \frac{\theta}{\Delta x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \left(\gamma_3 \frac{\theta}{2} - \frac{\theta}{\Delta x}\right) & \frac{\gamma_2 \theta}{2} & \left(\gamma_3 \frac{\theta}{2} + \frac{\theta}{\Delta x}\right) & \frac{\gamma_2 \theta}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{w}{2\Delta t} & -\frac{\theta}{\Delta x} & \frac{w}{2\Delta t} & \frac{\theta}{\Delta x} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \left(\gamma_3 \frac{\theta}{2} - \frac{\theta}{\Delta x}\right) & \frac{\gamma_2 \theta}{2} & \left(\gamma_3 \frac{\theta}{2} + \frac{\theta}{\Delta x}\right) & \frac{\gamma_2 \theta}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{w}{2\Delta t} & -\frac{\theta}{\Delta x} & \frac{w}{2\Delta t} & \frac{\theta}{\Delta x} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \left(\gamma_3 \frac{\theta}{2} - \frac{\theta}{\Delta x}\right) & \frac{\gamma_2 \theta}{2} & \left(\gamma_3 \frac{\theta}{2} + \frac{\theta}{\Delta x}\right) & \frac{\gamma_2 \theta}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{w}{2\Delta t} & -\frac{\theta}{\Delta x} & \frac{w}{2\Delta t} & \frac{\theta}{\Delta x} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \left(\gamma_3 \frac{\theta}{2} - \frac{\theta}{\Delta x}\right) & \frac{\gamma_2 \theta}{2} & \left(\gamma_3 \frac{\theta}{2} + \frac{\theta}{\Delta x}\right) & \frac{\gamma_2 \theta}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{w}{2\Delta t} & \frac{\theta}{\Delta x T_2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\Lambda} x(n+1) \\
 & = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{w}{2\Delta t} & \frac{1-\theta}{\Delta x} & \frac{w}{2\Delta t} & -\frac{1-\theta}{\Delta x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \left(\frac{1-\theta}{\Delta x} - \gamma_3 \frac{1-\theta}{2}\right) & -\gamma_2 \frac{1-\theta}{2} & -\left(\frac{1-\theta}{\Delta x} + \gamma_3 \frac{1-\theta}{2}\right) & -\gamma_2 \frac{1-\theta}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{w}{2\Delta t} & \frac{1-\theta}{\Delta x} & \frac{w}{2\Delta t} & -\frac{1-\theta}{\Delta x} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \left(\frac{1-\theta}{\Delta x} - \gamma_3 \frac{1-\theta}{2}\right) & -\gamma_2 \frac{1-\theta}{2} & -\left(\frac{1-\theta}{\Delta x} + \gamma_3 \frac{1-\theta}{2}\right) & -\gamma_2 \frac{1-\theta}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{w}{2\Delta t} & \frac{1-\theta}{\Delta x} & \frac{w}{2\Delta t} & -\frac{1-\theta}{\Delta x} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \left(\frac{1-\theta}{\Delta x} - \gamma_3 \frac{1-\theta}{2}\right) & -\gamma_2 \frac{1-\theta}{2} & -\left(\frac{1-\theta}{\Delta x} + \gamma_3 \frac{1-\theta}{2}\right) & -\gamma_2 \frac{1-\theta}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{w}{2\Delta t} & \frac{1-\theta}{\Delta x} & \frac{w}{2\Delta t} & -\frac{1-\theta}{\Delta x} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \left(\frac{1-\theta}{\Delta x} - \gamma_3 \frac{1-\theta}{2}\right) & -\gamma_2 \frac{1-\theta}{2} & -\left(\frac{1-\theta}{\Delta x} + \gamma_3 \frac{1-\theta}{2}\right) & -\gamma_2 \frac{1-\theta}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{w}{2\Delta t} & \frac{1-\theta}{\Delta x T_2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{A} x(n) \\
 & + \underbrace{\begin{bmatrix} -\alpha_{T_1} & 0 & -\alpha_{T_1} & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\alpha_{T_2} \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} Q_{FP1}(t_{n+1}) \\ Q_{FP2}(t_{n+1}) \\ Q_{offT1}^n \\ Q_{offT2}^n \end{bmatrix}}_{u(n)} + \underbrace{\begin{bmatrix} \alpha_{T_1} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{d(n)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ -\gamma_1 + S_b \\ -z \frac{w}{2\Delta t} \\ -\gamma_1 + S_b + z \left(\frac{1-\theta}{\Delta x T_2} + \gamma_3 \frac{1-\theta}{2} \right) \\ 0 \end{bmatrix}}_{OP} \\
 \end{aligned} \quad (7.6)$$

In the next section a the structure of a MPC utilizing the now developed models will be presented.

7.2 Model Predictive Controller

It should be noted that MPC notation and terminology varies a lot between authors. We have chosen to follow the notation of *J.M. Maciejowski* presented in the book *Predictive Control With Constraints* [22]. Maciejowski uses a "hat" notation to denote predictions: $\hat{x}(k|n)$ where the variable x is predicted to have a certain value at time k and n is the time when the prediction is made.

As presented in *Subsection 3.2.2 Model Predictive Control (MPC)* the main function of an MPC is to create a input trajectory \hat{u} and apply it, the order in which the prediction is made and applied is important to note in order to avoid confusion. At every time step the following operations are executed in order:

1. Obtain plant measurement $x(n)$, assuming $C = I$
2. Compute required input trajectory $\hat{u}(n|n), \hat{u}(n+1|n), \dots, \hat{u}(n+H_u|n)$, where H_u is known as the control horizon
3. Apply $u(n) = \hat{u}(n+1|n)$ to the plant

The quadratic optimization problem which is solved in order to find the predicted input trajectory can be stated as:

$$\min_{\hat{u}(n|n), \dots, \hat{u}(n+H_p-1|n)} \mathcal{J}(y, u, \dots, i) \triangleq \sum_{j=1}^{\Gamma} \sum_{i=0}^{H_p-1} \|F_j(i)\|_{Q(i,j)}^2 \quad (7.7)$$

Where $\|x\|_Q^2 = x^T Q x$ is the quadratic form, H_p is the prediction horizon, \mathcal{F} is the control objective, Q is the cost of each objective and Γ is the number of objectives. This optimization problem will be subject to the linear inequalities constraints:

$$\begin{aligned} \hat{u}_{min} &\leq \hat{u}(n+i|n) \leq \hat{u}_{max}, \quad \forall i = 0, \dots, T_n, \quad T_n \leq H_p, \\ \Delta \hat{u}_{min} &\leq |\hat{u}(n+i|n) - \hat{u}(n+i-1|n)| \leq \Delta \hat{u}_{max}, \quad \forall i = 0, \dots, T_k, \quad T_n \leq H_p, \quad (7.8) \\ \hat{x}_{min} &\leq \hat{x}(n+i|n) \leq \hat{x}_{max}, \quad \forall i = 0, \dots, T_n, \quad T_n \leq H_p, \end{aligned}$$

Where $\hat{u}(n+i|n)$ is predicted control inputs at time i , $\Delta \hat{u}(n+i|n)$ is the predicted change in control inputs from time $i-1$ to i and $\hat{x}(n+i|n)$ is the predicted state at time i .

Though these constraints often arise from physical limitations of a system, such as a pumps maximum speed or a tanks height, they are limitations on our **optimization problem** not on the actual physical system. Therefore the physical system might exceed these bounds if the system model is imprecise or sensors make wrong measurements due to noise.

7.2.1 Cost Function

From *Chapter 5 Requirements, Limitations and System Overview* we have control objectives: Prevent CSO, smoothen WWTP inflow and use available storage equally. In order for a Model predictive controller to take these objectives into account we have formulate them by use of a cost function. A common way of stating a general quadratic

cost function is:

$$\mathcal{J}(y, u, \dots, j) \triangleq \sum_{j=1}^{\Gamma} \sum_{i=0}^{H_p-1} \|F_j(i)\|_{Q(j,i)}^2 \quad (7.9)$$

The most important of our operational control objectives is preventing CSO. By including overflow, into our model as a input parameter it can be penalized in our cost function by the therm $\hat{u}(i)$. To see how the overflow is included in our model go to *Section 7.1 State Space Models*, where overflow is included as Q_{oFT1} and Q_{oFT2} . The second most important control objective is smoothing WWTP inflow, this can be done by placing a cost to the change in actuator output $\Delta\hat{u}(i)$. The least important control objective is ensuring equal use of storage space, the MPC is only expected to try and use storage space equally during periods with low water inflow. Equal storage usage shall be achieved by penalising $\hat{E}(i)$ which is a state deviation from a reference. This yields the cost function:

$$V(n) = \sum_{i=0}^{H_p-1} \|\hat{E}(n+i|n)\|_{Q(i)}^2 + \sum_{i=0}^{H_u-1} \|\Delta\hat{u}(n+i|n)\|_{R(i)}^2 + \sum_{i=0}^{H_u-1} \|\hat{u}(n+i|n)\|_{S(i)}^2 \quad (7.10)$$

Where, H_p is our prediction horizon, H_u is our control horizon. One thing to take note of is that initially (7.10) does not seem to comply with the cost function presented in the optimization problem stated as (7.7). This is because $H_u \neq H_p$. However since the cost $Q(i)$, $R(i)$, $S(i)$ can be changed at any future time i we can simply select the cost $R(i)$, $S(i) = 0$ for $t > H_u$, assuming $H_u \leq H_p$.

In order to actually calculate the cost for a real system we need to define the therm: $\hat{E}(i)$. Assuming a linear system with $C = I$ and $D = 0$ and input disturbance, the system model used in the MPC can be stated as:

$$\hat{x}(n+i|n) = A\hat{x}(n+i-1|n) + B\hat{u}(n+i-1|n) + B_d\hat{u}_d(n+i-1|n)\mathcal{C}_{\Delta x}$$

where $\mathcal{C}_{\Delta x}$ is a constant change in \hat{x} see *Section 7.1 State Space Models*, Then we can express the error $\hat{E}(j)$ as:

$$\begin{aligned} \hat{E}(n+i|n) &= r(n+i) - \hat{x}(n+i|n) \\ &= r(n+i) - (A\hat{x}(n+i-1|n) + B\hat{u}(n+i-1|n) \\ &\quad + B_d\hat{u}_d(n+i-1|n) + \mathcal{C}_{\Delta x}) \end{aligned} \quad (7.11)$$

where r is the reference. As the $\hat{E}(n+i|n)$ is used to ensure equal storage, r is set as the states at which the MPC should seek to keep the system at.

7.2.2 Lifting and Optimization

Though we have stated our initial optimization problem as (7.7) most standard optimization frameworks needs a quadratic problem stated on its standard Quadratic Programming form [22]:

$$\min_{\theta} \frac{1}{2} \theta^T \Phi \theta + \phi^T \theta \quad (7.12)$$

subject to

$$\Omega \theta \leq M$$

Luckily (7.7) can be formulated into such a problem, using a process called lifting. The idea behind the lifting process is to express each sum in a cost function (7.7) as a matrix vector product, and to express the constraints as a matrix inequality. Until now we have assumed that the function for which we wanted to optimize is \hat{u} , however choosing $\Delta\hat{u}$ instead is beneficial. This is because choosing $\Delta\hat{u}$ allows us to express the cost function in terms past/known parameters and predicted parameters. Keep this in mind as we progress.

The lifting process is quite lengthy, and for linear systems with no disturbance and simple constraints like (7.8), trivial. Some optimization frameworks like the one used in this report Casadi, can even perform the process. If the reader is unfamiliar with the concept of lifting, consult *Predictive Control With Constraints* [22,p. 75-77 and 81-84]. As the cost function (7.10) deviates from the "standard" by including disturbance and constant change $\mathcal{C}_{\Delta x}$ we will show the lifting process only for the cost function. The lifting of constraints will not be shown.

Lifting the cost function involves expressing the sum of quadratic terms as vectors. Before this is the variable The rewritten cost function is shown below:

$$V(n) = \|\mathcal{T}(n) - \mathcal{X}(n)\|_{\mathcal{Q}}^2 + \|\Delta\mathcal{U}(n)\|_{\mathcal{R}}^2 + \|\Delta\mathcal{U}(n) - \mathcal{U}(n)_{pre}\|_{\mathcal{S}}^2 \quad (7.13)$$

Where the known parameters are:

$$\mathcal{T}(n) = \begin{bmatrix} r(n+1|n) \\ r(n+2|n) \\ \vdots \\ r(n+H_p|n) \end{bmatrix}, \quad \mathcal{U}_{pre}(n) = \begin{bmatrix} u(n-1) \\ u(n-1) \\ \vdots \\ u(n-1) \end{bmatrix}$$

the predicted parameters are:

$$\mathcal{X}(n) = \begin{bmatrix} \hat{x}(n+1|n) \\ \hat{x}(n+2|n) \\ \vdots \\ \hat{x}(n+H_p|n) \end{bmatrix}, \quad \Delta\mathcal{U}(n) = \begin{bmatrix} \Delta\hat{u}(n+0|n) \\ \Delta\hat{u}(n+1|n) \\ \vdots \\ \Delta\hat{u}(n+H_u-1|n) \end{bmatrix} \quad (7.14)$$

and weight matrices are::

$$\mathcal{Q} = \begin{bmatrix} Q(1) & 0 & \cdots & 0 \\ 0 & Q(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & Q(H_p) \end{bmatrix}, \quad \mathcal{R} = \begin{bmatrix} R(0) & 0 & \cdots & 0 \\ 0 & R(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & R(H_u-1) \end{bmatrix}$$

$$\mathcal{S} = \begin{bmatrix} S(0) & 0 & \cdots & 0 \\ 0 & S(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & S(H_u-1) \end{bmatrix}$$

It is possible to express the predicted states $\mathcal{X}(n)$ in terms of known values, model parameters and $\Delta\hat{U}(n)$. This results in the only "unknown" in our system being the

value for which we want to optimize $\Delta\hat{U}(n)$. Below the predicted states are shown:

$$\mathcal{X}(n) = \Psi x(n) + \Upsilon u(n-1) + \Theta \Delta \mathcal{U}(n) + \Upsilon_d u(n-1) + \Theta_d \Delta \mathcal{U}_d(n) + \Lambda \mathcal{C}_{\Delta x} \quad (7.15)$$

And express using lifted matrices:

$$\begin{aligned} \begin{bmatrix} \hat{x}(n+1|n) \\ \vdots \\ \hat{x}(n+H_u|n) \\ \hat{x}(n+H_u+1|n) \\ \vdots \\ \hat{x}(n+H_p|n) \end{bmatrix} &= \begin{bmatrix} A \\ \vdots \\ A^{H_u} \\ A^{H_u+1} \\ \vdots \\ A^{H_p} \end{bmatrix} x(n) + \begin{bmatrix} B \\ \vdots \\ \sum_{i=0}^{H_u-1} A^i B \\ \sum_{i=0}^{H_u} A^i B \\ \vdots \\ \sum_{i=0}^{H_p-1} A^i B \end{bmatrix} u(n-1) \quad (7.16) \\ + \begin{bmatrix} B & \cdots & 0 \\ AB+B & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{H_u-1} A^i B & \cdots & B \\ \sum_{i=0}^{H_u} A^i B & \cdots & AB+B \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{H_p-1} A^i B & \cdots & \sum_{i=0}^{H_p-H_u} A^i B \end{bmatrix} &\begin{bmatrix} \Delta\hat{u}(n|n) \\ \vdots \\ \Delta\hat{u}(n+H_u-1|n) \end{bmatrix} + \begin{bmatrix} B_d \\ \vdots \\ \sum_{i=0}^{H_u-1} A^i B_d \\ \sum_{i=0}^{H_u} A^i B_d \\ \vdots \\ \sum_{i=0}^{H_p-1} A^i B_d \end{bmatrix} u_d(n-1) \\ + \begin{bmatrix} B_d & \cdots & 0 \\ AB_d+B_d & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{H_u-1} A^i B_d & \cdots & B_d \\ \sum_{i=0}^{H_u} A^i B_d & \cdots & AB_d+B_d \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{H_p-1} A^i B_d & \cdots & \sum_{i=0}^{H_p-H_u} A^i B_d \end{bmatrix} &\begin{bmatrix} \Delta\hat{u}_d(n|n) \\ \vdots \\ \Delta\hat{u}_d(n+H_u-1|n) \end{bmatrix} + \begin{bmatrix} I \\ A+I \\ A^2+A+I \\ \vdots \\ \sum_{i=0}^{H_p-1} A^i I \end{bmatrix} \mathcal{C}_{\Delta x} \end{aligned}$$

With all terms in the cost function now including $\Delta\hat{U}(n)$ we can restate the cost function on the form shown in (7.12). We start by inserting the expression for $\mathcal{X}(n)$ into the matrix cost function (7.13) yielding:

$$\begin{aligned} V(n) &= \|\mathcal{T}(n) - (\Psi x(n) + \Upsilon u(n-1) + \Theta \Delta \mathcal{U}(n) + \Upsilon_d u(n-1) + \Theta_d \Delta \mathcal{U}_d(n) + \Lambda \mathcal{C}_{\Delta x})\|_Q^2 \\ &\quad + \|\Delta \mathcal{U}(n)\|_{\mathcal{R}}^2 + \|\Delta \mathcal{U}(n) - \mathcal{U}(n)_{pre}\|_{\mathcal{S}}^2 \end{aligned}$$

which if we make the substitution with:

$$\mathcal{E}(n) = -\mathcal{T}(n) + \Psi x(n) + \Upsilon u(n-1) + \Upsilon_d u(n-1) + \Theta_d \Delta \mathcal{U}_d(n) + \Lambda \mathcal{C}_{\Delta x}$$

Where $\mathcal{E}(n)$ is known as the tracking error, yields:

$$V(n) = \|\Theta \Delta \mathcal{U}(n) - \mathcal{E}(n)\|_Q^2 + \|\Delta \mathcal{U}(n)\|_{\mathcal{R}}^2 + \|\Delta \mathcal{U}(n) - \mathcal{U}(n)_{pre}\|_{\mathcal{S}}^2 \quad (7.17)$$

Furthermore substituting the definition of $\| - \|_{F_j(i)}^2$ in allows for the restatement:

$$V(n) = [\Theta \Delta \mathcal{U}(n) - \mathcal{E}(n)]^T \mathcal{Q} [\Theta \Delta \mathcal{U}(n) - \mathcal{E}(n)] + [\Delta \mathcal{U}(n)]^T \mathcal{R} [\Delta \mathcal{U}(n)] \\ + [\Delta \mathcal{U}(n) - \mathcal{U}(n)_{pre}]^T \mathcal{S} [\Delta \mathcal{U}(n) - \mathcal{U}(n)_{pre}]$$

$$V(n) = \Delta \mathcal{U}(n)^T [\Theta^T \mathcal{Q} \Theta + \mathcal{R} + \mathcal{S}] \Delta \mathcal{U}(n) + \mathcal{U}(n)_{pre}^T \mathcal{S} \mathcal{U}(n)_{pre} + \mathcal{E}(n)^T \mathcal{Q} \mathcal{E}(n) \\ - \Delta \mathcal{U}(n)^T [\mathcal{S} \mathcal{U}(n)_{pre} + \Theta^T \mathcal{Q} \mathcal{E}(n)]$$

$$V(n) = \Delta \mathcal{U}(n)^T [\Theta^T \mathcal{Q} \Theta + \mathcal{R} + \mathcal{S}] \Delta \mathcal{U}(n) + \mathcal{U}(n)_{pre}^T \mathcal{S} \mathcal{U}(n)_{pre} + \mathcal{E}(n)^T \mathcal{Q} \mathcal{E}(n) \\ - 2 \Delta \mathcal{U}(n)^T [\mathcal{S} \mathcal{U}(n)_{pre} + \Theta^T \mathcal{Q} \mathcal{E}(n)]$$

Which after a substitution of:

$$\Phi = \Theta^T \mathcal{Q} \Theta + \mathcal{R} + \mathcal{S} \\ \phi = 2[\mathcal{S} \mathcal{U}(n)_{pre} + \Theta^T \mathcal{Q} \mathcal{E}(n)]$$

yields a quadratic problem on its standard form.

$$V(n) = \Delta \mathcal{U}(n)^T \Phi \Delta \mathcal{U}(n) - \Delta \mathcal{U}(n)^T \phi + const \quad (7.18)$$

The MPC realized as part of the project was created using the optimization framework Casadi. A overview of the MPC code will be given in *Section 8.4 MPC Implementation Using CasADi* alongside a general overview of the simulation in which the MPC was used.

7.3 System Parameterization

With a system model and the structure of the MPC described we need to determine at which operation point the models should operate. Two operating points has to be chosen, the flow operating point Q_{op} and the height operating point h_{op} .

The linearized diffusion wave Saint Venant equation Equation (6.26), has a terms dependant on the operating points. These are: $\gamma_1(Q_{op}, h_{op})$, $\gamma_2(Q_{op}, h_{op})$ and $\gamma_3(Q_{op}, h_{op})$. These arise from the linearization of the friction term $S_f(Q, h)$ seen in Equation (6.21). To determine the operating points the friction term $S_f(Q, h)$ is plotted in *Figure 7.2*. Since the maximum height of the pipe is 1 m, the height value ranges from 0 m to 1 m. Meanwhile the two pumps maximum flow value is $2 \frac{\text{m}^3}{\text{s}}$ so the value ranges from $0 \frac{\text{m}^3}{\text{s}}$ til $2 \frac{\text{m}^3}{\text{s}}$. The plots have been rotated, to get a better view of how the relation between the height and flow is.

In *Figure 7.2 b* the interval for both variables are set to 0.2-1, to get a better overview of S_f . From ad hoc simulations with the Euler model, an average height in the gravity pipe was read to be approximately 0.256 m, under heavy rain events. This height is chosen to be the operating point, h_{op} . The Manning equation is used to find the flow operating point.

$$Q_{op} = AR_{h_{op}}^{2/3} \frac{1}{n} \sqrt{S_b} = 1.446 \frac{m^3}{s} \quad (7.19)$$

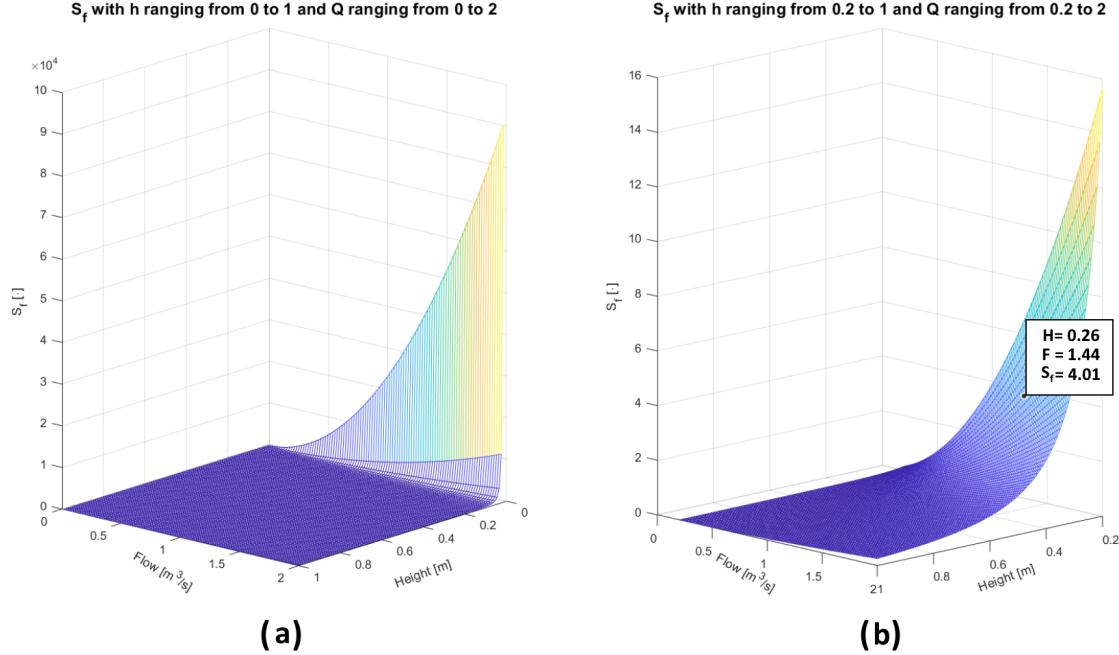


Figure 7.2: 3D mesh of $S_f(Q, h)$ at different ranges of h and Q .

The operating point can be seen in *Figure 7.2 b*. The operating points will be used for both the Preissmann scheme and the Euler scheme, furthermore the operating points will also be used in the free flow boundary condition.

Now that the operating points have been evaluated, the f term in Equation (6.21) will be evaluated. In the equation $\alpha = \frac{f}{2g}$ is present. The value f is the Darcy-Weisbach friction factor. The expression for the friction factor can be seen below in Equation (7.20) [34].

$$f = 0.25 \left[\log \left(\frac{\varepsilon}{3.7w} + \frac{5.74}{Re^{0.9}} \right) \right]^{-2} \quad (7.20)$$

where:

w	is the width of the gravity pipe.	[m]
ε	is the roughness factor of the pipe.	[·]
Re	is Reynolds number.	[·]

The roughness coefficient is a factor, that describes the roughness of a material. The factor can be found in a table for a given material. The gravity pipe is assumed to consist of smooth concrete, where factor is found to be 0.025. Reynolds number is used to predict whether the flow is laminar or turbulent. If $Re < 2100$ the flow is considered laminar, if $2100 \leq Re \leq 4000$ the flow is transitional to a turbulent flow, and $Re > 4000$ the flow is considered turbulent [35]. The formula for Reynolds number can be seen below in Equation (7.21).

$$Re = \frac{4 \cdot Q}{\pi \cdot w \cdot \vartheta} \quad (7.21)$$

where Q is the flow and ϑ is the kinematic viscosity for a given liquid. The expression for kinematic viscosity for water can be seen below.

$$\vartheta = \frac{1.792 \cdot 10^{-6}}{1 + \left(\frac{T}{25}\right)^{1.165}} \Bigg|_{T=10^\circ\text{C}} = 1.33 \cdot 10^{-6} \frac{\text{Pa}}{\text{s}} \quad (7.22)$$

where T is the temperature of the water. The temperature is assumed constant and set to $T=10^\circ\text{C}$. By using these values Reynolds number equals $Re = 1.39 \cdot 10^6$ when $Q = Q_{op}$ and the Darcy-Weisbach friction term becomes $f = 0.053$ [36] [34].

Regarding the Preissmann scheme, *Equation (6.47) Stability and Precision of the Preissmann Scheme* is used to determine the sampling time. By setting $\theta = 0.67$, $C_r = 0.5$ and $\bar{H} = h_{op}$ the sampling time is determined in Equation (7.23).

$$\Delta t = \frac{C_r \Delta x}{\sqrt{gh_{op}}} = 12.62 \text{ s} \quad (7.23)$$

The sampling time for the Euler model is set to 12.62 s as well, to give the two models the same conditions to compare against.

In the table below is an overview of the different values used in the state space models.

Δt	Δx_{T1}	Δx_{T2}	Δx	w	S_b	z
12.620 s	$\sqrt{150}$ m	$\sqrt{200}$ m	40 m	1m	0.05	1 m
f	h_{op}	Q_{op}	γ_1	γ_2	γ_3	θ
0.053	0.256 m	$1.446 \frac{\text{m}^3}{\text{s}}$	0.562	0.178	-2.702	0.67

7.4 Conclusion

In this section, two Euler state space models are formed alongside with the Preissmann state space representation. Also a complete description of the MPC is presented, enabling an implementation of the cost function and lifting process, suitable for a quadratic programming solver. System parameterization is performed in this chapter, where the operating point for the friction term $S_f(Q, h)$ is evaluated.

Simulation Design and System Testing

8

This chapter is going to present the design of the simulation software, the system tests and their corresponding results. It begins with an introduction to the simulation design. Afterwards, each component of the simulation is described. Starting with the implementation of the network topology in EPA SWMM and PY-SWMM, the implementation of the MPC using CasADi and finally the implementation of the disturbances. At the end, the specific test scenarios and their test results are discussed.

Note that, the presented software is written in Python and other Python related tools and packages. Thus, the decision was made to put the simulation code of the project on GitHub as a public repository [37].

8.1 Simulation Building Blocks

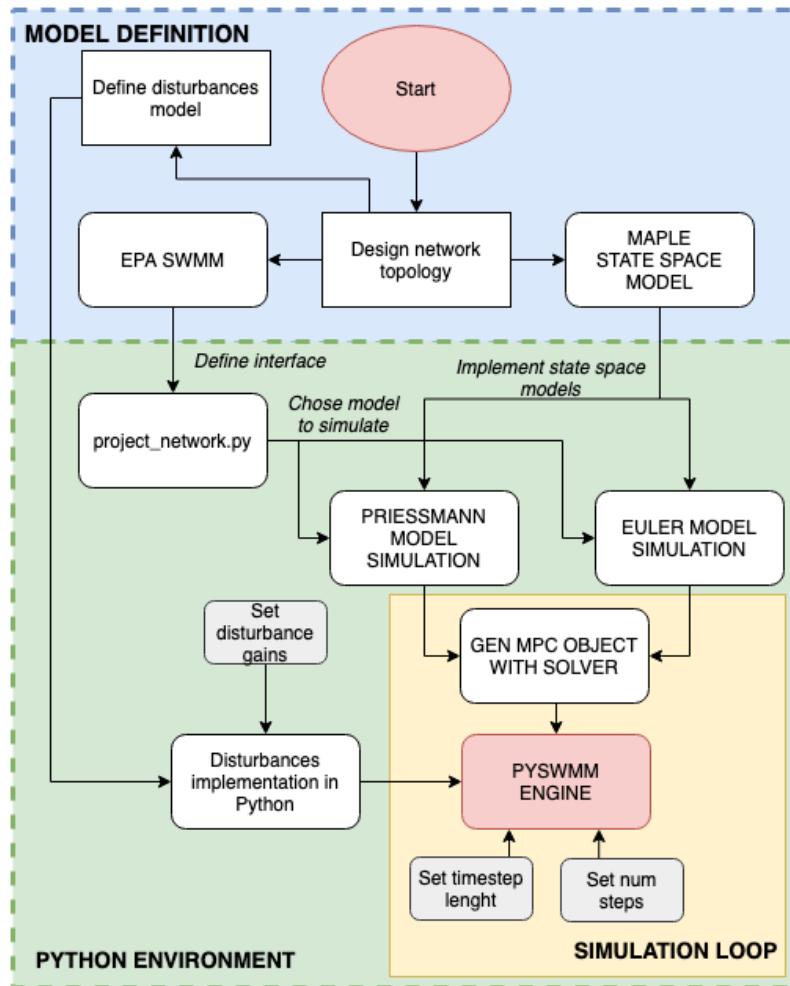


Figure 8.1: Simulation Model

In this section an overview of the simulation design is presented, see *Figure 8.1*. The figure depicts the flow of the simulation all the way from the design of the network topology to running the simulation engine. It also shows the connections between the different software used in the simulation. These are namely, EPA SWMM, PYSWMM (in Python), and Maple. Furthermore, it also shows how the simulation loop is created and where the MPC is solved. The MPC is solved using CasADi and it is described in detail in *Section 8.4 MPC Implementation Using CasADi*.

Furthermore, the figure breaks up the process into two major blocks, *Model Definition* and *Python Environment*. This is because, the Python environment and the model definition are directly linked to other software outputs, e.g.: from Maple it is possible to extract the state space models for both the Euler and Preissmann discretization. Subsequently these are put into the Python simulation environment.

The following sections are going to describe the major simulation components in more detail.

8.2 EPA SWMM

The network topology described in *Section 5.2 Proposed Network Topology And Disturbance Models* is implemented in the EPA Storm Water Management System (SWMM, or EPA SWMM) software [38]. It is a free to use, open source modeling tool which is used to predict runoff quantity and quality from drainage systems. It is created and maintained by the United States Environmental protection Agency (EPA) [38]. It has easy to use, pre-made models for the pumps, storage units, outfalls and junctions. The project's network is constructed using these network elements. The network topology in EPA SWMM can be seen in *Figure 8.2*.

To acquire level measurements along the gravity pipe, junctions are added at equal distances corresponding to the discretized sections. In EPA SWMM, junctions usually represent manholes, thus adding manholes with no-inflow, allows for level measurements without affecting the model downstream. The gravity pipe slope and the junctions can be seen in *Figure 8.3*. There are also three tanks in the model, the settings can be seen in *Table (8.1)*

The pumps are implemented as constant flow providers. This means that, in the EPA SWMM, the pumps have a pump curve which is equal to 1 Cubic Meter per Second (CMS) at timestamp 1. Thus, this implementation allows for a controller design where the output is in terms of flow. The pump curve implementation is shown in *Figure 8.4*. All the pumps in the system apply this pump curve.

Furthermore, as mentioned in *Section 7.1 State Space Models*, there are no catchment areas modelled in the simulation and the water inflow is directly going into T_3 . This is where the disturbances are added into the model as well, see *Section 8.5 Disturbance Implementation*.

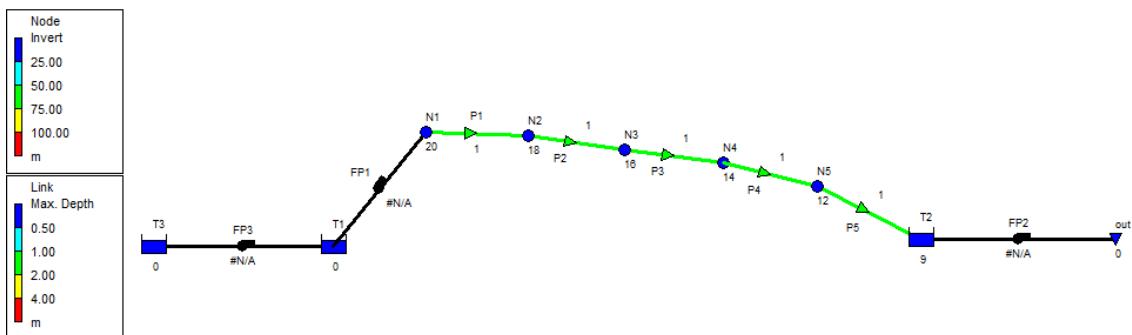


Figure 8.2: Topology in EPA SWMM. The T_{1-3} are tanks, FP_{1-3} are flow-providers, or pumps. P_{1-5} are the discrete pipe sections. N_{1-5} are junctions to measure the height within the discrete sections. Furthermore, out is the outfall.

Node/Property	Initial depth	Max depth	Invert. Elev.	Coefficient
Tank 1 (T1)	0	3	9	150
Tank 2 (T2)	2	2	9	150
Tank 3 (T3)	950	1000	0	1000

Table 8.1: Tank settings in EPA SWMM

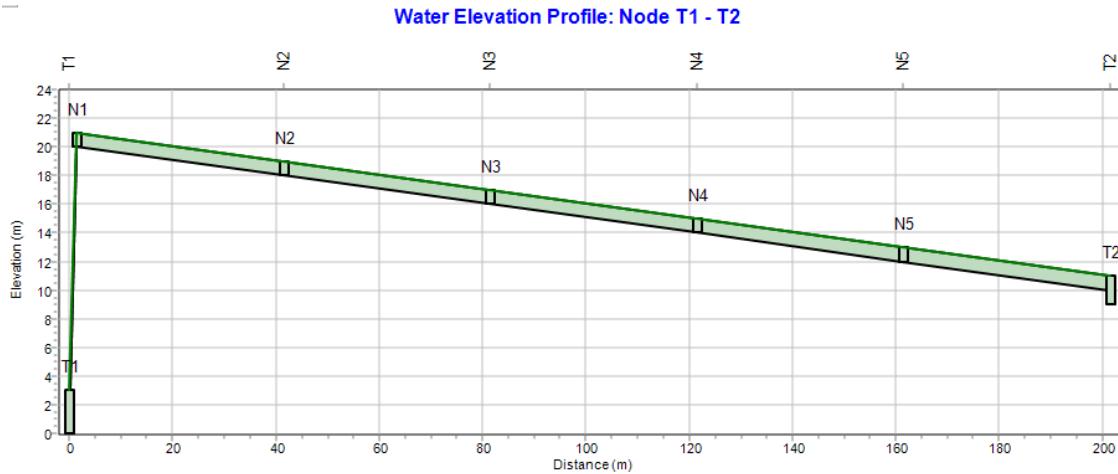


Figure 8.3: Gravity pipe slope in EPA SWMM

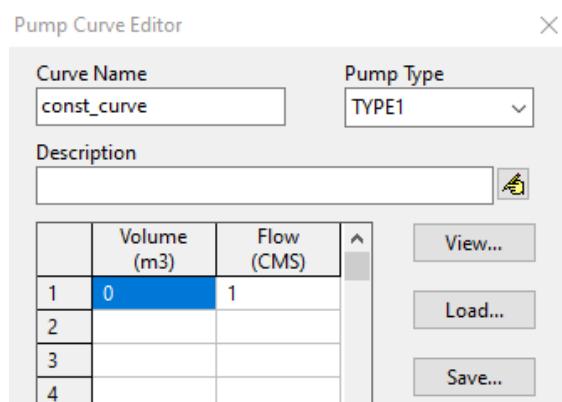


Figure 8.4: Pump curve as constant flow provider

The WWTP is not modelled in this topology and instead an outfall is added to the end of the model, where no backwater effects are present which could affect the system upstream.

The EPA SWMM simulation engine is set to use the complete Dynamic Wave model. Further settings inside the Dynamic Wave are left at default. The time step size and the length of the simulation is determined inside the Python environment. They are described in the subsequent sections. This exported network topology can also be accessed from the GitHub repository [37], inside the folder *epa_networks/project_network/*.

8.3 PYSWMM

EPA SWMM also has an open source python interface developed by [39]. Its purpose is to utilize the modeling interface and simulation engine of EPA SWMM and connect it to the extensive Python scientific simulation and computing environments. In this project, this interface was extensively utilized and the entirety of the MPC and this simulation is implemented in Python. This section describes how this interface is created and how it is possible to utilize EPA SWMM along with Python.

8.3.1 PYSWMM Interface to EPA SWMM

To interface with the topology created in the EPA SWMM, it is required to declare an instance of the PYSWMM interface by providing a path to the network topology. This is done in Snippet (8.7). The *time_step* size is also declared. The units by default are in seconds. Calling the function *step_advance()* with the specific time step, will forward the simulation, in this case, by 60 seconds.

Once the simulation instance has been created, it is possible to load in the interfaces to the pumps, tanks and the junctions. This is done by creating instances of the pumps and tanks:

Furthermore, to get the model states from the simulation engine, the corresponding attributes needs to be accessed. Note, that in this case depth represents the level measurement. In this example, a dummy MPC step functions also provided to show, how it is possible to solve the objective function at each timestep of the simulation. This is done in the following way:

A basic example of the PYSWMM interface is included in the project repository, [37], in the folder *example_projects/PySWMM working example/*. Note, this is only a simple ON/OFF control example on a different topology than the final project topology. This is meant as a guide for others to begin using PYSWMM with other control schemes.

```

1  from pyswmm import Simulation
2
3  network_name = "epa_networks/project_network/project_network.inp"
4  # [seconds]
5  time_step = 60
6
7  with Simulation(network_name) as sim:
8
9      for step in sim
10          sim.step_advance(time_step)

```

Figure 8.5: Loading the project_network model .inp file into PYSWMM

```

1 ...
2 with Simulation(network_name) as sim:
3
4     pump1 = Links(sim)["FP1"]
5     tank1 = Nodes(sim)["T1"]
6     junction_n1 = Nodes(sim)["N1"]
7
8     for step in sim
9         sim.step_advance(time_step)

```

Figure 8.6: Creating instances of the pump and tank objects

```

1 ...
2 for step in sim:
3     states = [
4         tank1.depth,
5         junction_n1.depth
6     ]
7
8     # dummy example of the optimization step solve
9     mpc.step(states, disturbances)
10
11     sim.step_advance(time_step)

```

Figure 8.7: Accessing states of simulation engine

8.4 MPC Implementation Using CasADi

As described in *Subsection 7.2.2 Lifting and Optimization*, the control problem can be lifted as a quadratic optimization problem. Such a problem can be solved by many different software packages. For this project, CasADi has been selected, since it can be used through the Python interface to solve the optimization problem. [40]

The MPC optimization problem must be formed in a way that the solver can interpret. This means, that the optimization problem is generally formulated as:

$$\begin{aligned}
 & \underset{x}{\text{minimize}} : \quad \frac{1}{2}x^T Hx + g^T x \\
 & \text{subject to:} \quad x_{lb} \leq x \leq x_{ub} \\
 & \quad \quad \quad a_{lb} \leq Ax \leq a_{ub}
 \end{aligned} \tag{8.1}$$

However, it is simpler and easier to provide the optimization problem in the following form:

$$\begin{aligned}
 & \underset{x}{\text{minimize}} : \quad F(x) \\
 & \text{subject to:} \quad lbx \leq x \leq ubx \\
 & \quad \quad \quad lbg \leq G \leq ubg
 \end{aligned} \tag{8.2}$$

$F(x)$	is cost function	$[.]$
x	is the optimization variable	$[.]$
G	is the constraint functions	$[.]$
lbx	is the lower bounds for the optimization variable	$[.]$
ubx	is the upper bounds for the optimization variable	$[.]$
lbg	is the lower bounds for the constraints	$[.]$
ubg	is the upper bounds for the constraints	$[.]$

This form of the optimization problem is reshaped into the form in Equation (8.1) by the Python interface. Where, all the variables from the simplified form must be passed to the CasADi solver. With these values, a solver object is created which can be used to solve the stated optimization problem with different inputs.

8.4.1 Generating the solver object

To illustrate how the above process is performed in CasADI, a working example is presented in Snippet (8.8). This example is directly from the project code and can be found in the repository, [37], in the *controller/mpc.py* file. This snippet shows how CasADI inputs the defined optimization problem and generates the solver. A more detailed breakdown of the code is explained in the following subsections.

Optimization Variable

Starting at *line 15*, the optimization variable is defined. For the current MPC use case, the optimization variable will only contain the control inputs. Additionally, these control inputs also contain the overflow variables, as the overflow has been modelled to be an expensive control input (see 7.2.1).

Input Variables

From *line 7* the input variables are defined. The *input_variables* is a CasADI vector containing variables which can be used in the equations that the solver uses for the cost and the constrain functions. It contains variables, such as: initial conditions for states, control input, the reference and disturbance.

Cost Function

The cost function is defined in *line 41*. It is defined according to *Subsection 7.2.1 Cost Function*. It assigns a cost to states, the control input and the slew rate of the control inputs. The cost function uses the predicted states which are calculated in *lines 22 - 27*. It also uses the previously defined change in control input and control inputs. The control inputs is defined as a function of the initial control input and the changes in control input. It is constructed in *lines 32 - 35*.

Constraint Function

The CasADI solver has inbuilt constraints for the optimization variable. In this specific case, it is the control input change. Therefore, the remaining constraints are the control input and the states. The constraint function, which is given to CasADI, only must express the values which should be constrained. The bounds for those values are given to the solver when it is used to solve the optimization problem.

Therefore, the control input and states, have to be expressed as functions of the input variables and the optimization variable. The control input and the states are already expressed as such and are concatenated and forwarded the solver in *line 36*.

8.4.2 Using the Solver Object

Once the solver has been created, it can be called with specific inputs. For the purpose of this project, the MPC solver has been wrapped in a class that handles all the declarations, definitions, the optimization problem and the variables required for the MPC. The code can be seen in Snippet (8.9).

```

1 def gen_mpc_solver(A, B, Hu, Hp, Q, R, B_d=None, S=None, operating_point=None):
2 ...
3
4 ...
5 # Declaring and parting out input variables
6 # Input = [x0, u_prev, ref, ud_prev, ud]
7 input_variables = ca.SX.sym('i', delta_disturbances_input_index_end, 1)
8 x0 = input_variables[0:initial_state_index_end, :]
9 u_prev = input_variables[initial_state_index_end:prev_control_input_index_end, :]
10 ref = input_variables[prev_control_input_index_end:reference_index_end, :]
11 ud_prev = input_variables[reference_index_end:prev_disturbance_index_end, :]
12 dud = input_variables[prev_disturbance_index_end:delta_disturbances_input_index_end,:]
13
14 # Declaring solver outputs
15 x = ca.SX.sym('x', number_of_inputs * Hu, 1)
16 du = x[:number_of_inputs * Hu]
17 ...
18
19 ...
20 # To formulate a MPC optimization problem we need to describe:
21 # predicted_states = Z = psi x(k) + upsilon u(k-1) + Theta dU(x) + upsilon ud(k-1) + Theta dUD(x)
22 psi = gen_psi(A, Hp)
23 upsilon = gen_upsilon(A, B, Hp)
24 theta = gen_theta(upsilon, B, Hu)
25 upsilon_d = gen_upsilon(A, B_d, Hp)
26 theta_d = gen_theta(upsilon_d, B_d, Hp)
27 predicted_states = gen_predicted_states(psi, x0, upsilon, u_prev, theta, du, upsilon_d,
28                                         ud_prev, theta_d, dud, op=operating_point)
29
30 # Setup constraints
31 # construct U from dU
32 U = ca.SX.ones(du.size1())
33 for i in range(0, number_of_inputs):
34     U[i:number_of_inputs] = ca.cumsum(du[i:number_of_inputs])
35 U = U + ca.repmat(u_prev, Hu, 1)
36 constraints = ca.vertcat(predicted_states, U)
37
38 # Cost function:
39 # Cost = (Z - T)' * Q * (Z - T) + dU' * R * dU
40 error = predicted_states - ref # e = (Z - T)
41 quadratic_cost = error.T @ Q @ error \
42                 + du.T @ R @ du \
43                 + U.T @ S @ U
44 # Setup Solver
45 ...
46
47 ...
48 quadratic_problem = {'x': du, 'p': input_variables, 'f': quadratic_cost, 'g': constraints}
49 mpc_solver = ca.qpsol('mpc_solver', 'qpoases', quadratic_problem, opts)
50
51 return mpc_solver

```

Figure 8.8: MPC optimization problem implemented in Python using CasADI

This function is used to solve the MPC optimization problem for the next time step. The input variables are generated containing the initial states, initial control input, the reference and the disturbance. The upper and lower bounds for the constraints are also specified when the solver is called. Once the problem is solved, the optimal control inputs are in the resulting class instance. The next control input can then be applied as a control input to the system.

8.5 Disturbance Implementation

The disturbance model previously described in *Subsection 5.2.1 Disturbances models* is also implemented into Python. The dataset proposed has two columns of data, domestic/industrial waste and rain. The disturbance model is implemented into Python as a class. An example configuration can be seen in the Snippet (8.13). It shows that, it is possible to select whether to use either of the two disturbances. Furthermore, gains are also implemented independently for both datasets.

There are functions exposed to the simulation engine to get the k -th disturbance and the disturbances corresponding to the length of the prediction horizon. Note, that if the prediction horizon is pointing outside of the dataset, the disturbance is treated as zero. This essentially mimics operation when neither of the disturbances are active. This API extension can be seen in the Snippet (8.11).

Furthermore, the disturbances are also provided to the EPA SWMM engine. This is done through the PYSWMM interface, rather than through the GUI of the EPA SWMM. To replicate the direct inflow of the combined disturbances into $T1$, an additional pump, $FP3$, and tank, $T3$, is added into the topology, see *Figure 8.12*. This pump also acts as a constant flow provider. $T3$, is initially filled with water at a relatively high depth. By opening the pump from $T3$ to $T1$ inside the PYSWMM simulation, it is possible to directly use the disturbances fed into MPC. Since EPA SWMM uses the complete Dynamic Wave model, the solution with the same disturbances will differ. Thus, this implementation allows for a single dataset interface to the simulation. This is also highlighted in Snippet (8.13).

```

1  def step(self, initial_state, u_prev, ref=None, prev_disturbance=None, disturbance=None):
2      ...
3          # saving variables in object and handling optional inputs
4      ...
5          self.solver_input = mpc.gen_solver_input(self.initial_state, self.u_prev, self.ref,
6                                              self.prev_disturbance, self.disturbance)
6          self.result = self.solver(p=self.solver_input, lbg=self.lbg, ubg=self.ubg,
7                                    lbx=self.lbx, ubx=self.ubx)
8
9          self.dU = self.result['x']
10         ...
11             # saving results and logging values
12         ...

```

Figure 8.9: Using the generated solver to get the optimal control input

```

1 from util_scripts import disturbance_reader
2
3 # Configure the disturbance
4 # If you do not wish to use any gains on the data,
5 # set either rain_gain or poop_gain to 1.
6 # It also possible to select which disturbance you want to use
7 disturb_config = {
8     "disturbance_data_name": "data/disturbance_data/hour_poop_rain.csv",
9     "use_rain": True,
10    "use_waste": True,
11    "rain_gain": 1,
12    "waste_gain": 1,
13    # "use_random": False
14 }
15
16 disturb_manager = disturbance_reader.Disturbance(disturb_config)

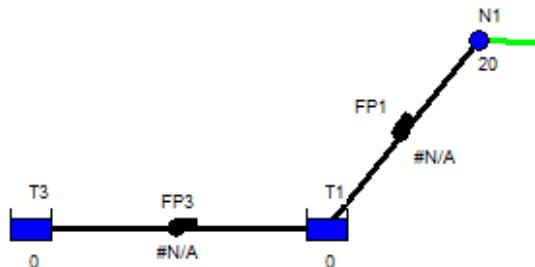
```

Figure 8.10: Configuring the disturbance interface

```

1 # idx is equivalent to k inside the simulation
2 for idx, step in enumerate(sim):
3
4     future_delta_disturb = disturb_manager.get_k_delta_disturbance(idx, prediction_horizon)
5     prev_disturb = disturb_manager.get_k_disturbance(idx - 1, 1)
6     current_disturb = disturb_manager.get_k_disturbance(idx, 1)
7
8     disturb_pump.target_setting = current_disturb

```

Figure 8.11: Accessing the k -th disturbance, and disturbances towards the prediction horizon.**Figure 8.12:** Implementation of the disturbance model with addition of T_3 and FP_3

8.6 System Testing

The test will be conducted as a comparison between an on/off controller and the designed MPC controller. The test will make the following assumptions:

- $FP1$ will have a maximum throughput of $2 \frac{m^3}{s}$ and a slew rate of $1 \frac{m^3}{s^2}$
- $FP2$ will have a maximum throughput of $1 \frac{m^3}{s}$ and a slew rate of $0.5 \frac{m^3}{s^2}$
- The inflow disturbance into the tank numbered $T1$ will be normalized with respect to the time step $\Delta t = 12.65s$ and gain $1 \frac{m^3}{s}$. A gain factor can then later be applied to both the model of rainfall and wast water form residential area. They are named G_{rain} and G_{res} respectively.

A baseline for the MPC will be established by increasing the inflow gain factor G_{rain} while ensuring a ratio of $\frac{G_{rain}}{G_{res}} = 5$, until overflow happens in a system controlled by the on/off controller. Here after the MPC will be exposed to an inflow of the same gain factor. Since more than one system model will be designed the MPCs using different system models will be ranked according to their ability to handle overflow. Note: As the MPC using the Preissmann model was unable to run results, no results are included.

The implementation of the on/off controller is shown below:

```

1  class PumpSetting(Enum):
2      CLOSED = 0
3      OPEN = 2
4      PUMP_2_OPEN = 1
5
6
7  def local_controller(tank, type, min_depth_tank):
8
9      if type == "t1":
10          if tank.depth >= min_depth_tank:
11              return PumpSetting.OPEN.value
12          else:
13              return PumpSetting.CLOSED.value
14
15      elif type == "t2":
16          if tank.depth >= min_depth_tank:
17              return PumpSetting.PUMP_2_OPEN.value
18          else:
19              return PumpSetting.CLOSED.value
20 }
```

Figure 8.13: Implementation of On/OFF-controller used in system test

8.7 Test Results

In this section a overview of the test results are shown. The MPC was run with a $H_p = 80$ and $H_u = 50$ to allow for anticipation of the heavy rain events. The baseline gain results from the on/off controller was found to be $G_{rain} = 5.6$ and $G_{res} = 1.12$.

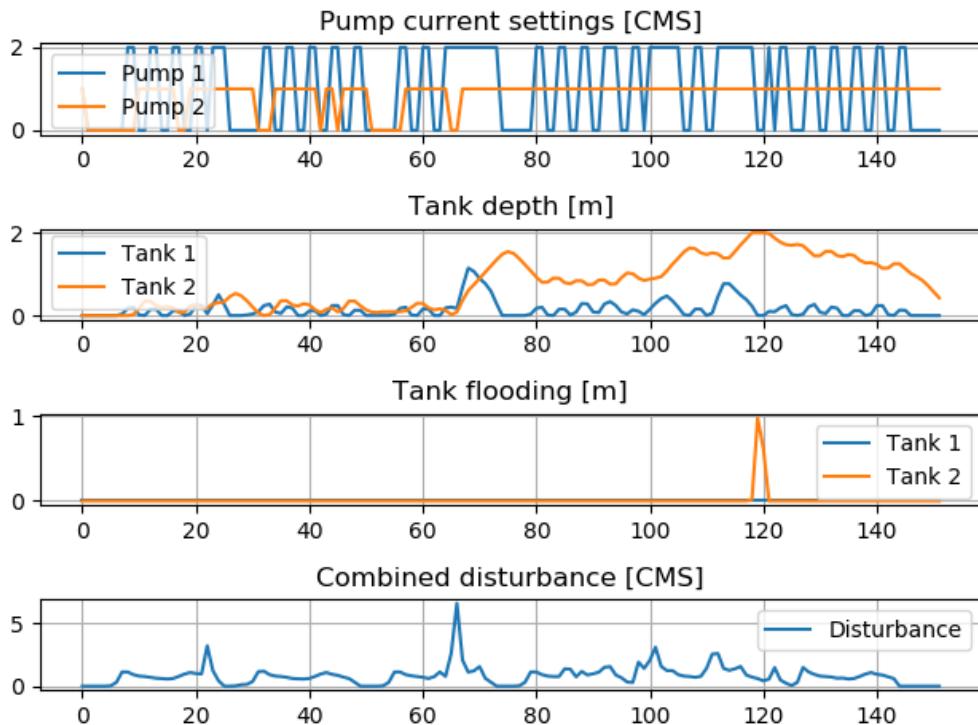


Figure 8.14: System behaviour when controlled by two local on/off controllers

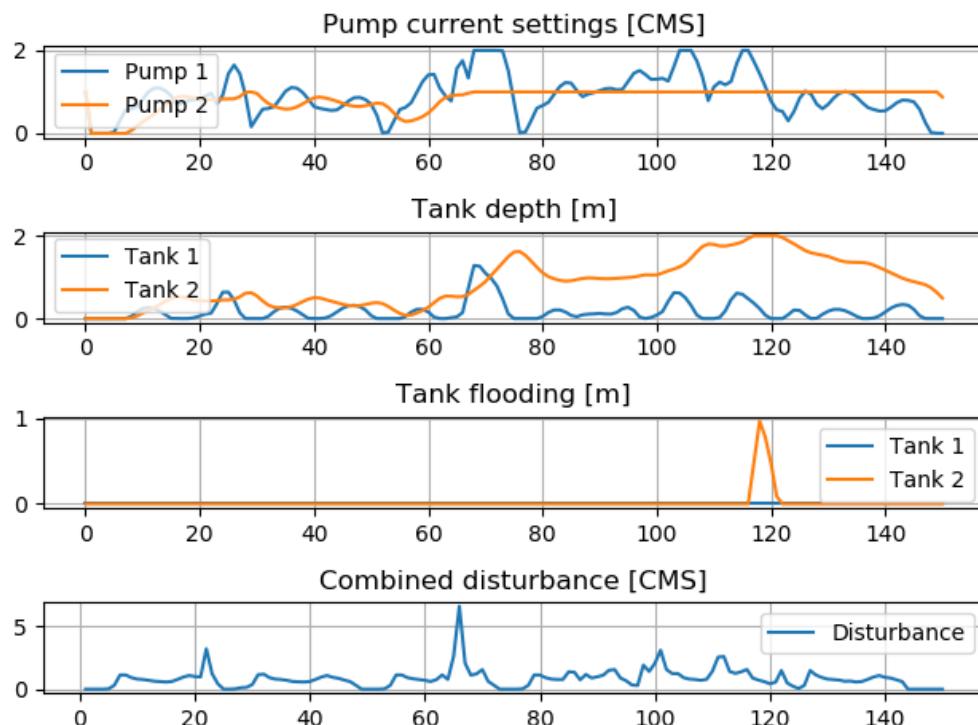


Figure 8.15: System behaviour when controlled by MPC using the backflow model

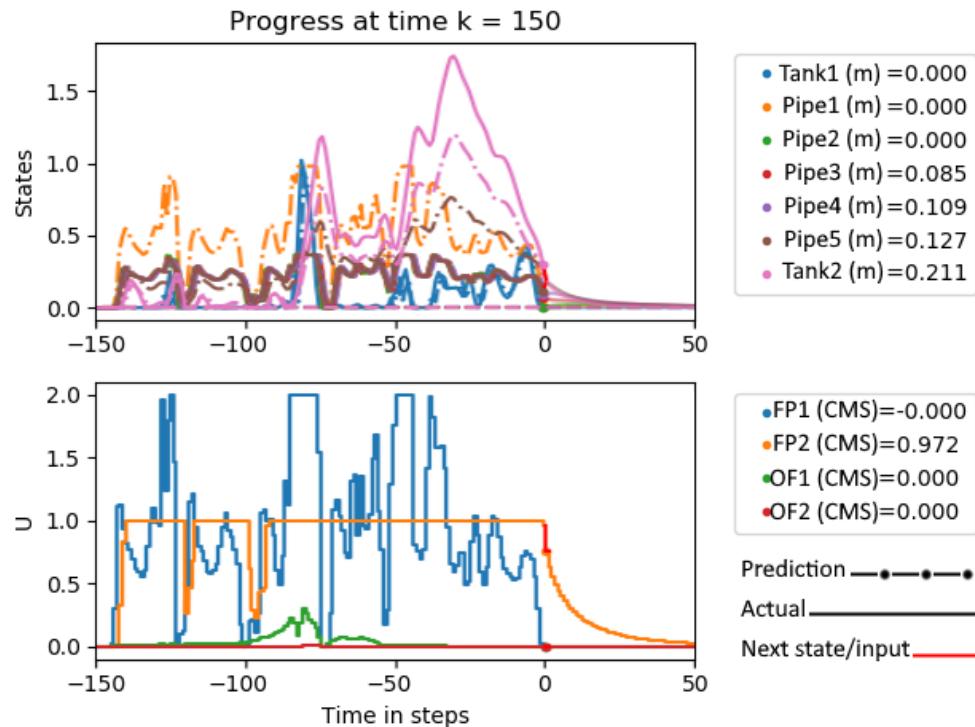


Figure 8.16: Predictive behaviour of the MPC using the backflow model

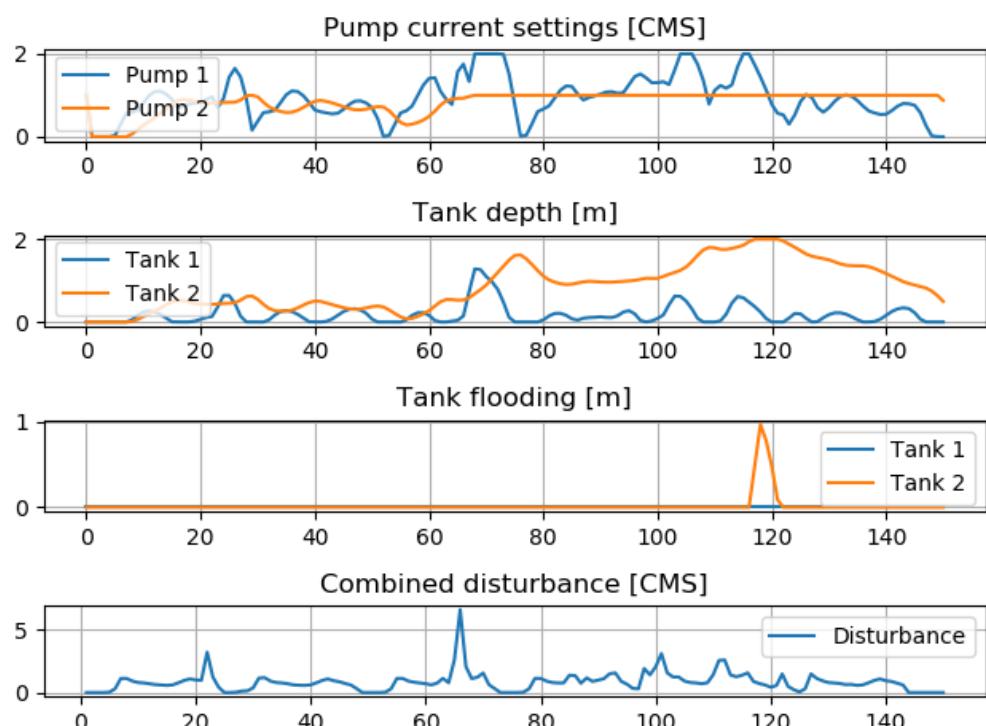


Figure 8.17: System behaviour when controlled by MPC using the free flow model

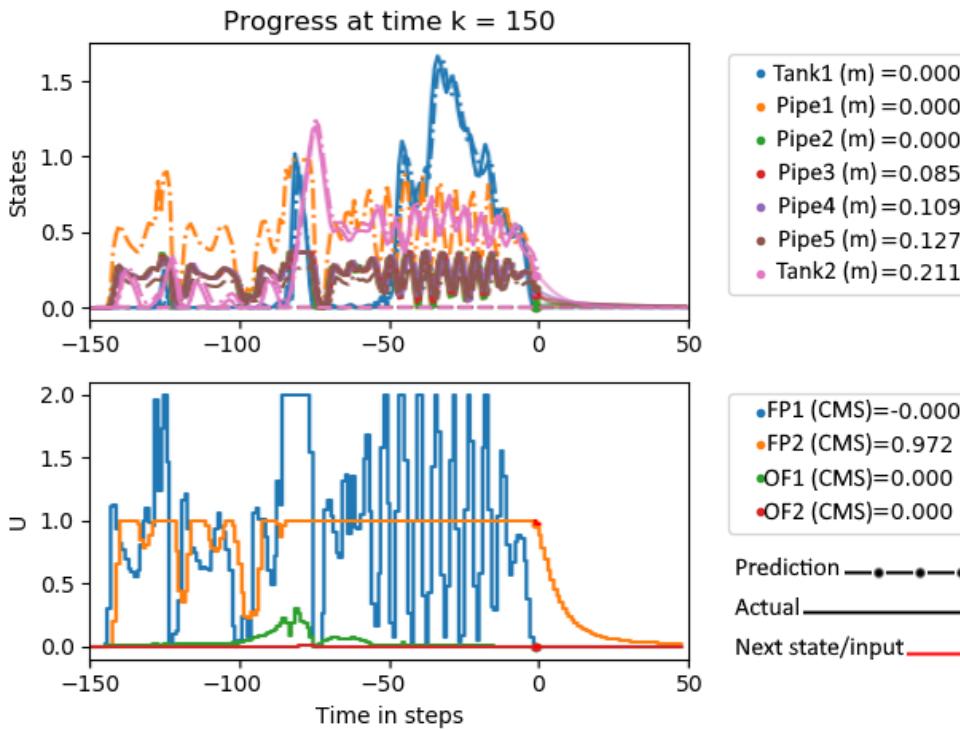


Figure 8.18: Predictive behaviour of the MPC using the free flow model

8.8 Result Analysis

As the tests made so far is insufficient to conclude on the performance of an MPC controller using either the Euler model or the Preissmann implemented in this project. The analysis will instead discuss the progress of the development of the models and the MPC based on the test results. The models are not yet considered fully implemented as they have not yet been fully adjusted to fit the system we are trying to predict. The simulation is therefore not giving results that is considered representative of the potential performance of these models.

One problem seems to be with the system we are trying to model, some of the dimensions seems to have unusual proportions compared to what would be expected in a real system. First of all, the slope of the gravity pipe is unusually high, this will cause faster dynamics in an already small-scale system which will cause the timescale to be very different from any real implementation of similar wastewater systems. Additionally, the retention tank size compared to the pipe section is very small which creates some unusual dynamics.

The models could not be tested with the operating point flows as they would make the optimization problems infeasible. This seems to be because the states have hard constraints which is also usually advised against. To solve this the constraints should be softened by adding slack variables similar to the overflow, but simply to keep the problem feasible and without any physical representation. This would solve one problem that was found in a test that whenever a state with constant outflow caused by the operating point would reach 0, it would be impossible to add more to that state to counter the outflow and keep the state within the constraint. However decent results were obtained with the Euler model if the

constant flow was omitted and the problem was feasible even with the hard constraints. However, with the constant flow and more adjustments to the model better results are expected.

The Preissmann model was not able to be properly tested as the model does not behave as expected. This should be resolved with tweaking but there has not been time for that. This problem and others could have been avoided with testing the models along the development. The models could have been tested by simply propagating the model with known inputs and check the behavior. However, this was not done.

Global Conclusion

9

9.1 Discussion

9.1.1 CSO as a Control Input

In this project the CSO was implemented as a control input with a high cost. The cost was a quadratic cost on the CSO outflow. This led to some unintended behavior because the MPC would try to distribute the CSO as if it expected it a high flow for a short time has a much higher cost than a long time with low flow. This still leads to the intended behavior of avoiding overflow, since the cost is still higher than pumping out as much water as possible. In a real scenario it would never be desired to have CSO unless the tank is full. A potential solution, that has not yet been tested, is also proposed.

The proposed solution is to add a cost to the integral of the CSO flow. This would add a quadratic cost to combined CSO volume. While the raw cost would be less accurate, it would incentivize the MPC to do the overflow as late as possible which would only happen when there would be actual overflow. This would better represent the real-world objective.

9.1.2 Continues testing during development

While some tests were done during development, for most of the code, the actual models were not tested this period. This resulted in the models having issues when they were tested for the final implementation. The code was tested as it was developed to make sure it behaved as expected. The models could have been tested without being implemented in the MPC to see if it behaved as expected. Testing alongside the development could have ensured that the models were ready for the final implementation. Additionally, it is easier to identify and then correct problems if the models were tested alone, as there are fewer things that could become an issue. In general, module tests are preferred to "big-bang" tests.

9.1.3 Extendable and flexible implementation of code in python

A significant amount of time has been spent to make the code easier to extend in the future. The individual modules can be used as standalone, such as the MPC math functions and the MPC Object. The simulation framework has been designed to be easily extendable. This will make it possible for potential future work that uses parts of or the whole implementation. This is also a result the decision to implement it in Python as opposed to something like MATLAB. Python is free for everyone, it can interface with far more software packages and such as other optimization solvers or simulation packages. Additionally, it was a simpler task to interface with the EPA SWMM software with Python

that with MATLAB or Simulink. It is overall considered as a good decision to implement the project in Python.

9.1.4 Future Work

This project has done a lot of the first steps towards making a comparison between different models for an MPC to control a wastewater system. However, there are still some important things that must be done to properly do a comparison, as well as some things that would be nice to further develop. First, the models should be adjusted to behave as expected. To fully implement the models with the constant flow it would be needed to add soft constraints to the states of the model. Then once the models behave as expected, it would be interesting to see the behavior of the MPC with the CSO cost being on the integral of the flow. Furthermore, it would be desired to have the constraints implemented as LMIs since this is the traditional way of doing it and it could be used to say something about robustness.

9.2 Conclusion

This project sought to develop an MPC and compare the performance using three different models. First the initial case was set up. Then a short analysis on the topic of controlling sewer networks were done. After which, the requirements and goals was set for the project and the product. Which were to test the performance of the three models, the Euler model, using free flow boundary conditions, the Euler model, using offset flow boundary conditions, and the Preissmann model. The performance should be compared to a local control with no prediction. The main control objective is minimizing the CSO, with some other secondary objectives.

The MPC and the models were developed separately. They were decided to be implemented as state space models. Therefore, the model was discretized and linearized and set up as state space models. The MPC was implemented in Python using CasADi and is designed to include disturbance and constant change to the prediction. The cost function where implemented to have costs for error on the states compared to a reference, to the input and to the slew rate of the input. And the constraints where implemented for the control input, the slew rate of the control input and the predicted states.

The expected tests for the performance of the system were not made as the initial testing showed problems with the prediction models. It is concluded that models were not ready for the performance comparison and more work is needed. However, the tests show that the simulations with the MPC using both Euler models run as expected.

Since the tests were not conducted, we cannot evaluate the performance of our controller. However, most of our project objectives were fulfilled. A network topology was designed which strives to mimic real life operation, using retention tanks, gravity pipes, pumps and output to a WWTP. We have made models using the Preissmann and Euler scheme. We have designed an MPC with the defined control objectives two of which behaves as expected. And we have created a disturbance dataset using real domestic waste and rainfall data which can be used for our simulation. This has all been implemented in Python, using CasADi for the problem solving. And interfacing with EPA SWMM for the simulation.

Additionally, the implementation has been made in a way that it is highly modular and straightforward to extend.

The group is very satisfied with the work done in this project, although there was no time for evaluation of the models and adequate performance testing, the rest of the project objectives were achieved.

Bibliography

- [1] Jacob Naundrup Pedersen and Thomas Holm Pilgaard. Model predictive control of a sewer system. Master's thesis, Aalborg University - Section for Automation and Control, 2018.
- [2] Llorenç Salleras Mestre and Pravin Karthick Murugesan. Laboratory emulation and control of a sewer system with storage. Master's thesis, Aalborg University - Section for Automation and Control, 2012.
- [3] G. Dirckx, M. Schütze, S. Kroll, Ch. Thoeye1, G. De Gueldre, and B. Van De Steene. Rtc versus static solutions to mitigate cso's impact. In *12nd International Conference on Urban Drainage, Porto Alegre/Brazil*, September 2011.
- [4] Trash and toilets in mesopotamia: Sanitation and early urbanism. <http://www.asor.org/anetoday/2016/04/trash-and-toilets-in-mesopotamia-sanitation-and-early-urbanism/>. Accessed: 2020-21-02.
- [5] Saifullah Khan. *Sanitation and wastewater technologies in Harappa/Indus valley civilization (ca. 2600-1900 BC)*. Academia.edu.
- [6] Emmanuel DialynasEmmanuel Dialynas. *Evolution of sanitation and wastewater technologies through the centuries - Edition: 1*. IWA Publishing, 2014.
- [7] Christopher J.Duffy Kirk D.French. Prehispanic water pressure: A new world first. *Journal of Archaeological Science*, 37(5):1027–1032, 2010.
- [8] Peter J. Aicher. *Guide to the Aqueducts of Ancient Rome*. Bolchazy-Carducci Publishers, 1 Jan 1995.
- [9] Water supply and sanitation services in modern europe: developments in 19th-20th centuries. https://www.academia.edu/34533153/Water_supply_and_sanitation_services_in_modern_Europe_developments_in_19th-20th_centuries. Accessed: 2020-13-03.
- [10] Dot distribution map london cholera outbreak from john snow. http://matrix.msu.edu/~johnsnow/images/online_companion/chapter_images/fig12-5.jpg. Accessed: 2020-13-03.
- [11] Optimised design of stormwater tanks. <https://www.grundfos.com/content/dam/Global%20Site/Market%20areas/Water%20utility/Downloads/Brochures/design%20of%20Stormwater%20Tanks.pdf>. Accessed: 2020-13-03.
- [12] Why control sanitary sewer overflows? https://www3.epa.gov/npdes/pubs/sso_casestudy_control.pdf. Accessed: 2020-17-02.

- [13] How bad is sewage overflow for properties and the environment? <https://www.mwremediation.com/blog/how-bad-is-sewage-overflow-for-properties-and-the-environment.html>. Accessed: 2020-17-02.
- [14] Biological components of wastewater. <https://water.me.vccs.edu/courses/ENV149/biological.htm>. Accessed: 2020-20-02.
- [15] Fungal problems in wastewater treatment works. <https://www.aquaenviro.co.uk/fungal-problems-in-wastewater-treatment-works/>. Accessed: 2020-20-02.
- [16] Thorkild Hvitved-Jacobsen, Jes Vollertsen, and Asbjørn Haaning Nielsen. *Sewer processes: microbial and chemical process engineering of sewer networks*. CRC press, 2013.
- [17] How does a waste water treatment plant work? <https://sciencing.com/waste-water-treatment-plant-work-4896800.html>. Accessed: 2020-25-05.
- [18] Go-ahead for conversion first two dutch wastewater treatment plants to revolutionary modular verdygo concept. <http://www.dutchwatersector.com/news/go-ahead-for-conversion-first-two-dutch-wastewater-treatment-plants-to-revolutionary-modular>.
- [19] Camilla Molander. Influence of excessive water on wastewater treatment performance. Master's thesis, Chalmers University of Technology, 2015.
- [20] Nadia Schou Vorndran Lund, Anne Katrine Vinther Falk, Morten Borup, Henrik Madsen, and Peter Steen Mikkelsen. Model predictive control of urban drainage systems: A review and perspective towards smart real-time water management. In *CRITICAL REVIEWS IN ENVIRONMENTAL SCIENCE AND TECHNOLOGY 2018, VOL. 48*, 2018.
- [21] T. Beenenken, V. Erbe, A. Messmer, C. Reder, R. Rohlfing, M. Scheer, M. Schuetze, B. Schumacher, M. Weilandt, and M. Weyand. Real time control (rtc) of urban drainage systems – a discussion of the additional efforts compared to conventionally operated systems. *Urban Water Journal*, 10(5):293–299, 2013.
- [22] J.M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, England., 2002.
- [23] Tom S.Pedersen and Palle Andersen. Control-oriented modeling of thermodynamic systems. 2016.
- [24] Ven Te Chow, David R. Madment, and Larry W. Mays. *Applied Hydrology*. Publication Services, 1987.
- [25] Gordana Janevska. Mathematical modeling of pump system. 09 2013.
- [26] Aalborg kommune august 2019 nedbør (mm). <http://www.dmi.dk/vejrarkiv/>.
- [27] Walter A. Strauss. *Partial Differential Equations, An introduction*. John Wiley & Sons, Inc, 2008.
- [28] Romuald Szymkiewicz. *Numerical Modeling in Open Channel Hydraulics*, volume 83. 2001.

- [29] MR Shaghaghi and N Talebbeydokhti. Comparison of several finite difference schemes in flow modeling over an ogee spillway. In *Advances in Water Resources and Hydraulic Engineering*, pages 1725–1730. Springer, 2009.
- [30] Emeric Dequidt Jean-Pierre Villeneuve Sophie Duchesne, Alain Mailhot. Mathematical modeling of sewers under surcharge for real time control of combined sewer overflows. 3(4):241–252, 2001.
- [31] Victor M. Ponce. Diffusion wave modeling of catchment dynamics. http://uon.sdsu.edu/diffusion_wave_modeling_of_catchment_dynamics.html. Accessed: 2020-05-05.
- [32] Carsten Skovmose Kallesøe. Canal flow (saint venant equations). 2020.
- [33] Xavier Litrico and Vincent Fromion. *Modeling and Control of Hydrosystems*. Springer, 2009.
- [34] Lesson 25. general equation for head loss-darcy equation. <http://ecoursesonline.iasri.res.in/mod/page/view.php?id=1084>. Accessed: 2020-25-05.
- [35] Classification of flows, laminar and turbulent flows. https://www-mdp.eng.cam.ac.uk/web/library/enginfo/aero/thermal_dvd_only/aero/fprops/pipeflow/node8.html. Accessed: 2020-23-05.
- [36] What is the reynolds number? <https://www.simscale.com/docs/simwiki/numerics-background/what-is-the-reynolds-number/>. Accessed: 2020-25-05.
- [37] A python project of a model predictive controller (mpc) for urban drainage networks (udns) to mitigate combined sewer overflows (csos). https://github.com/dmicha16/waterlab_mpc. Accessed: 2020-26-05.
- [38] Storm water management model (swmm). <https://www.epa.gov/water-research/storm-water-management-model-swmm>. Accessed: 2020-26-05.
- [39] Bryant E. McDonnell, Katherine M. Ratliff, Michael E. Tryby, Jennifer Jia Xin Wu, and Abhiram Mullapudi. PySWMM: The Python Interface to Stormwater Management Model (SWMM), April 2020.
- [40] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

Grundfos Project Proposal

A

Model Predictive Control of Combined Sewer Overflows in Urban Drainage Networks

Contacts:

Krisztian M. Balla (Industrial PhD at AAU & Grundfos A/S, e-mail: kmb@es.aau.dk)
 Carsten S. Kallesøe (Professor at AAU & Chief engineer at Grundfos A/S, e-mail: csk@es.aau.dk)
 Jan Dimon Bendtsen (Assoc. Professor at AAU, e-mail: dimon@es.aau.dk)

1. Background

Combined sewers carry domestic wastewater and rain runoff towards treatment plants, where the sewage is treated before it is discharged to the environment [1]. Real-Time Control (RTC) of these networks is a challenging task since the system is characterized by large-scale dimensions, non- linear dynamics, and significant time-delays. Besides, Urban Drainage Networks (UDNs) are increasingly being pushed to their capacity limits due to changing weather conditions, resulting in increased amount and more frequent heavy rain events, which potentially can lead to Combined Sewer Overflows (CSO). Such overflows have negative consequences for both the environment and human health.

The objective of the regulation of UDNs is to minimize the effects of external disturbances (rain infiltration, domestic sewage and groundwater infiltration), over a system that under normal operation would not require any action. In Model Predictive Control (MPC), prediction of the partly unpredictable disturbances such as rain-runoff from catchments, and the prediction of forecastable periodic disturbances such as domestic waste, are required to anticipate future states and decisions [2].

2. Objectives

An optimal control strategy for controlling sewage flow in a UDN is to be developed in this project. The project can include the following tasks:

1. Understanding the control problem in Urban Drainage.
2. Simplified modeling of UDNs, based on mass- and energy balances.
3. Identification of model parameters based on level measurements in the network.
4. Design of a model based optimal controller, including physical constraints and single- or multiple operational objectives. The operational objectives consist of: overflow prevention, smoothing of treatment plant inlet flow, equal use of available storage [3].
5. Test and implementation of the algorithms on a simulation model and on the Modular Water Laboratory at Aalborg University.

3. Laboratory Setup

The topology of the network considered in the project is shown in Fig. 1. Urban drainage systems typically consist of many storage elements such as gravity pipes, retention tanks, catchment areas and one or several outlet points leading to the treatment plant. The most common actuators in these types of networks are pumps and gates. In this project, networks with retention tanks are considered, where the stored sewage volumes are controlled by pumping stations. Hence, the regulated variable is flow, provided by locally controlled variable-speed pumps.

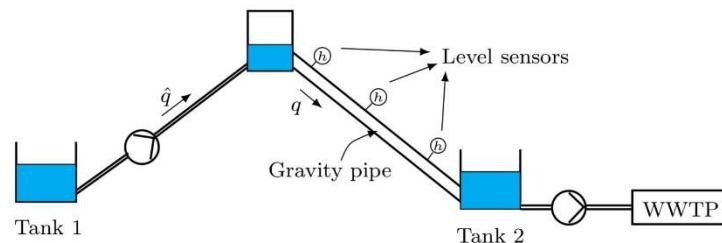


Figure 1 - Network topology.

It is furthermore expected to test the developed control algorithms in the modular water lab at Aalborg University. The test setup will consist of at least one retention tank (with overflow capability) and might consist of several gravity pipes.



Figure 2- Test setup.

The laboratory work consists of the following tasks:

1. Configuring the test setup, if necessary, small commissioning tasks.
2. Collect level, and if needed, flow data from the network for system identification.
3. Interfacing control through the Simulink interface.
4. Implementation and real-time test of initial static on-off controllers.
5. Implementation and real-time test of the developed MPC algorithms.

Moreover, the presented project is proposed in collaboration with the IoT super project, currently running in the ES department. The Wireless project is responsible for collecting data from the modules via wireless simulating the distribution in a real infrastructure.

4. References

- [1] M. R. Schütze, D. Butler, and M. B. Beck, *Modelling, Simulation and Control of Urban Wastewater Systems*. Springer, 2nd ed., 2002
- [2] V. Puig, G. Cembrano, J. Romera, J. Quevedo, B. Aznar, G. Ramón, and J. Cabot, "Predictive optimal control of sewer networks using CORAL tool: Application to Riera Blanca catchment in Barcelona," *Water Science and Technology*, vol. 60, no. 4, pp. 869–878, 2009.
- [3] N. S. V. Lund, A. K. V. Falk, M. Borup, H. Madsen, and P. Steen Mikkelsen, "Model predictive control of urban drainage systems: A review and perspective towards smart real-time water management," *Critical Reviews in Environmental Science and Technology*, vol. 48, no. 3, pp. 279–339, 2018.

Derivation of the Euler Model B

▼ Pipe To pipe flow - Check

Basis equations

$$h_j^{n+1} = \alpha \cdot (Q_{j-1}^n - Q_j^n) + h_j^n$$

$$Q_j^n = \frac{1}{\gamma_2} \cdot \left(-h_{j+1}^n \cdot \frac{1}{\Delta x_f} + h_j^n \cdot \left(\frac{1}{\Delta x_f} - \gamma_3 \right) + S_b - \gamma_1 \right)$$

Flows:

$$Q_{j-1}^n = \left(-h_j^n \cdot \frac{1}{\Delta x_f \cdot \gamma_2} + h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f \cdot \gamma_2} - \frac{\gamma_3}{\gamma_2} \right) + \frac{S_b - \gamma_1}{\gamma_2} \right)$$

$$-Q_j^n = \left(h_{j+1}^n \cdot \frac{1}{\Delta x_f \cdot \gamma_2} - h_j^n \cdot \left(\frac{1}{\Delta x_f \cdot \gamma_2} - \frac{\gamma_3}{\gamma_2} \right) - \frac{S_b - \gamma_1}{\gamma_2} \right)$$

Collected description

$$h_j^{n+1} = \alpha \cdot (Q_{j-1}^n - Q_j^n) + h_j^n$$

$$h_j^{n+1} = \alpha \cdot \left(-h_j^n \cdot \frac{1}{\Delta x_f \cdot \gamma_2} + h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f \cdot \gamma_2} - \frac{\gamma_3}{\gamma_2} \right) + \frac{S_b - \gamma_1}{\gamma_2} + h_{j+1}^n \cdot \frac{1}{\Delta x_f \cdot \gamma_2} - h_j^n \cdot \left(\frac{1}{\Delta x_f \cdot \gamma_2} - \frac{\gamma_3}{\gamma_2} \right) - \frac{S_b - \gamma_1}{\gamma_2} \right) + h_j^n$$

$$h_j^{n+1} = \alpha \cdot \left(h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f \cdot \gamma_2} - \frac{\gamma_3}{\gamma_2} \right) + \frac{S_b - \gamma_1}{\gamma_2} + h_{j+1}^n \cdot \frac{1}{\Delta x_f \cdot \gamma_2} + h_j^n \cdot \left(\frac{-2}{\Delta x_f \cdot \gamma_2} + \frac{\gamma_3}{\gamma_2} \right) - \frac{S_b - \gamma_1}{\gamma_2} \right) + h_j^n$$

$$h_j^{n+1} = \alpha \cdot \left(h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f \cdot \gamma_2} - \frac{\gamma_3}{\gamma_2} \right) + h_{j+1}^n \cdot \frac{1}{\Delta x_f \cdot \gamma_2} + h_j^n \cdot \left(\frac{-2 \cdot \alpha}{\Delta x_f \cdot \gamma_2} + \alpha \cdot \frac{\gamma_3}{\gamma_2} \right) \right) + h_j^n$$

$$h_j^{n+1} = h_{j-1}^n \cdot \left(\frac{\alpha}{\Delta x_f \cdot \gamma_2} - \alpha \cdot \frac{\gamma_3}{\gamma_2} \right) + h_{j+1}^n \cdot \frac{\alpha}{\Delta x_f \cdot \gamma_2} + h_j^n \cdot \left(\frac{-2 \cdot \alpha}{\Delta x_f \cdot \gamma_2} + \alpha \cdot \frac{\gamma_3}{\gamma_2} + 1 \right)$$

$$h_j^{n+1} = h_{j-1}^n \cdot \left(\frac{\alpha}{\Delta x_f \cdot \gamma_2} - \alpha \cdot \frac{\gamma_3}{\gamma_2} \right) + h_j^n \cdot \left(1 + \alpha \cdot \frac{\gamma_3}{\gamma_2} - \frac{2 \cdot \alpha}{\Delta x_f \cdot \gamma_2} \right) + h_{j+1}^n \cdot \frac{\alpha}{\Delta x_f \cdot \gamma_2}$$

▼ Free flow linerization - Check

$$Q_j^n = C_d \cdot \sqrt{2 \cdot g} \cdot (h_j)^{\frac{3}{2}}$$

$$Q_{FF} \approx Q_j^n(h_{op}) + \frac{d}{dh_j} \left(Q_j^n(h_j) \right) \Big|_{h_{op}} \cdot (h_j - h_{op})$$

$$Q_{FF} \approx \left(C_d \cdot \sqrt{2 \cdot g} \cdot (h_{op})^{\frac{3}{2}} \right) + \frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \cdot (h_j - h_{op})$$

$$Q_{FF} \approx$$

$$\left(C_d \cdot \sqrt{2 \cdot g} \cdot (h_{op})^{\frac{3}{2}} \right) + \frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \cdot h_j - \frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \cdot h_{op}$$

$$= \frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \cdot h_j - \frac{C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2}$$

$$Q_{FF} \approx \frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \cdot h_j - \frac{C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2}$$

▼ Pipe 5 height - with Backflow

Basis equations

$$h_j^{n+1} = \alpha \cdot (Q_{j-1}^n - Q_j^n) + h_j^n$$

Inflow:

$$Q_j^n = \frac{1}{\gamma_2} \cdot \left(h_j^n \cdot \left(\frac{1}{\Delta x_f} - \gamma_3 \right) - h_{j+1}^n \cdot \frac{1}{\Delta x_f} + S_b - \gamma_1 \right)$$

⇓ Ajust to correct Q

$$Q_{j-1}^n = \frac{1}{\gamma_2} \cdot \left(h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f} - \gamma_3 \right) - h_j^n \cdot \frac{1}{\Delta x_f} + S_b - \gamma_1 \right)$$

outflow:

$$Q_j^n = \frac{1}{\gamma_2} \cdot \left(h_j^n \cdot \left(\frac{1}{\Delta x_{f2}} - \gamma_3 \right) - (h_{j+1}^n - z) \cdot \frac{1}{\Delta x_{f2}} + S_b - \gamma_1 \right)$$

Collect

$$h_j^{n+1} = \alpha \cdot \left(\frac{1}{\gamma_2} \cdot \left(h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f} - \gamma_3 \right) - h_j^n \cdot \frac{1}{\Delta x_f} + S_b - \gamma_1 \right) - \frac{1}{\gamma_2} \cdot \left(h_j^n \cdot \left(\frac{1}{\Delta x_{f2}} - \gamma_3 \right) - (h_{j+1}^n - z) \cdot \frac{1}{\Delta x_{f2}} + S_b - \gamma_1 \right) \right) + h_j^n$$

$$h_j^{n+1} = \alpha \cdot \left(\left(h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f \cdot \gamma_2} - \frac{\gamma_3}{\gamma_2} \right) - h_j^n \cdot \frac{1}{\Delta x_f \cdot \gamma_2} + \frac{S_b - \gamma_l}{\gamma_2} \right) - \left(h_j^n \cdot \left(\frac{1}{\Delta x_{f2} \cdot \gamma_2} - \frac{\gamma_3}{\gamma_2} \right) - (h_{j+1}^n - z) \cdot \frac{1}{\Delta x_{f2} \cdot \gamma_2} + \frac{S_b - \gamma_l}{\gamma_2} \right) \right) + h_j^n$$

$$h_j^{n+1} = \alpha \cdot \left(h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f \cdot \gamma_2} - \frac{\gamma_3}{\gamma_2} \right) - h_j^n \cdot \frac{1}{\Delta x_f \cdot \gamma_2} + \frac{S_b - \gamma_l}{\gamma_2} + h_j^n \cdot \left(\frac{\gamma_3}{\gamma_2} - \frac{1}{\Delta x_{f2} \cdot \gamma_2} \right) + (h_{j+1}^n - z) \cdot \frac{1}{\Delta x_{f2} \cdot \gamma_2} - \frac{S_b - \gamma_l}{\gamma_2} \right) + h_j^n$$

$$h_j^{n+1} = h_{j-1}^n \cdot \left(\frac{\alpha}{\Delta x_f \cdot \gamma_2} - \frac{\alpha \cdot \gamma_3}{\gamma_2} \right) - h_j^n \cdot \frac{\alpha}{\Delta x_f \cdot \gamma_2} + \frac{\alpha \cdot (S_b - \gamma_l)}{\gamma_2} + h_j^n \cdot \left(\frac{\alpha \cdot \gamma_3}{\gamma_2} - \frac{\alpha}{\Delta x_{f2} \cdot \gamma_2} \right) + (h_{j+1}^n - z) \cdot \frac{\alpha}{\Delta x_{f2} \cdot \gamma_2} - \frac{\alpha \cdot (S_b - \gamma_l)}{\gamma_2} + h_j^n$$

$$h_j^{n+1} = h_{j-1}^n \cdot \left(\frac{\alpha}{\Delta x_f \cdot \gamma_2} - \frac{\alpha \cdot \gamma_3}{\gamma_2} \right) + h_j^n \cdot \left(1 + \frac{\alpha \cdot \gamma_3}{\gamma_2} - \frac{\alpha}{\Delta x_{f2} \cdot \gamma_2} - \frac{\alpha}{\Delta x_f \cdot \gamma_2} \right) + h_{j+1}^n \cdot \frac{\alpha}{\Delta x_{f2} \cdot \gamma_2} - z \cdot \frac{\alpha}{\Delta x_{f2} \cdot \gamma_2}$$

Tank 2 height - with Backflow

Basis equations

$$h_j^{n+1} = \alpha \cdot (Q_{j-1}^n - Q_j^n) + h_j^n$$

Inflow:

$$Q_j^n = \frac{1}{\gamma_2} \cdot \left(h_j^n \cdot \left(\frac{1}{\Delta x_f} - \gamma_3 \right) - (h_{j+1}^n - z) \cdot \frac{1}{\Delta x_f} + S_b - \gamma_l \right)$$

↓ Ajust to correct Q

$$Q_{j-1}^n = \frac{1}{\gamma_2} \cdot \left(h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f} - \gamma_3 \right) - (h_j^n - z) \cdot \frac{1}{\Delta x_f} + S_b - \gamma_l \right)$$

outflow:

$$Q_j^n = Q_{FP2}^n$$

Collect

$$h_j^{n+1} = \alpha \cdot \left(\frac{1}{\gamma_2} \cdot \left(h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f} - \gamma_3 \right) - (h_j^n - z) \cdot \frac{1}{\Delta x_f} + S_b - \gamma_l \right) - Q_{FP2}^n \right) + h_j^n$$

$$h_j^{n+1} = \alpha \cdot \left(h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f \cdot \gamma_2} - \frac{\gamma_3}{\gamma_2} \right) - (h_j^n - z) \cdot \frac{1}{\Delta x_f \cdot \gamma_2} + \frac{S_b - \gamma_l}{\gamma_2} - Q_{FP2}^n \right) + h_j^n$$

$$h_j^{n+1} = h_{j-1}^n \cdot \left(\frac{\alpha}{\Delta x_f \cdot \gamma_2} - \frac{\alpha \cdot \gamma_3}{\gamma_2} \right) - (h_j^n - z) \cdot \frac{\alpha}{\Delta x_f \cdot \gamma_2} + \alpha \cdot \frac{(S_b - \gamma_l)}{\gamma_2} - \alpha \cdot Q_{FP2}^n + h_j^n$$

$$h_j^{n+1} = h_{j-1}^n \cdot \left(\frac{\alpha}{\Delta x_f \cdot \gamma_2} - \frac{\alpha \cdot \gamma_3}{\gamma_2} \right) - h_j^n \cdot \frac{\alpha}{\Delta x_f \cdot \gamma_2} + z \cdot \frac{\alpha}{\Delta x_f \cdot \gamma_2} + \alpha \cdot \frac{(S_b - \gamma_l)}{\gamma_2} - \alpha \cdot Q_{FP2}^n + h_j^n$$

$$h_j^{n+1} = h_{j-1}^n \cdot \left(\frac{\alpha}{\Delta x_f \cdot \gamma_2} - \frac{\alpha \cdot \gamma_3}{\gamma_2} \right) + h_j^n \cdot \left(1 - \frac{\alpha}{\Delta x_f \cdot \gamma_2} \right) + z \cdot \frac{\alpha}{\Delta x_f \cdot \gamma_2} + \alpha \cdot \frac{(S_b - \gamma_l)}{\gamma_2} - \alpha \cdot Q_{FP2}^n$$

Pipe 1 height - Ignoring Backflow and with Backflow

restart

$$h_j^{n+1} = \alpha \cdot (Q_{j-1}^n - Q_j^n) + h_j^n$$

Inflow:

$$Q_{j-1}^n = Q_{FPI}^n$$

Outflow:

$$Q_j^n = \frac{1}{\gamma_2} \cdot \left(h_j^n \cdot \left(\frac{1}{\Delta x_f} - \gamma_3 \right) - h_{j+1}^n \cdot \frac{1}{\Delta x_f} + S_b - \gamma_l \right)$$

Collect

$$h_j^{n+1} = \alpha \cdot \left(Q_{FPI}^n - \frac{1}{\gamma_2} \cdot \left(h_j^n \cdot \left(\frac{1}{\Delta x_f} - \gamma_3 \right) - h_{j+1}^n \cdot \frac{1}{\Delta x_f} + S_b - \gamma_l \right) \right) + h_j^n$$

$$h_j^{n+1} = \alpha \cdot \left(Q_{FPI}^n - h_j^n \cdot \left(\frac{1}{\Delta x_f \cdot \gamma_2} - \frac{\gamma_3}{\gamma_2} \right) + h_{j+1}^n \cdot \frac{1}{\Delta x_f \cdot \gamma_2} - \frac{S_b - \gamma_l}{\gamma_2} \right) + h_j^n$$

$$h_j^{n+1} = \alpha \cdot Q_{FPI}^n - h_j^n \cdot \left(\frac{\alpha}{\Delta x_f \cdot \gamma_2} - \frac{\alpha \cdot \gamma_3}{\gamma_2} \right) + h_{j+1}^n \cdot \frac{\alpha}{\Delta x_f \cdot \gamma_2} - \alpha \cdot \frac{(S_b - \gamma_l)}{\gamma_2} + h_j^n$$

$$h_j^{n+1} =$$

$$\alpha Q_{FPI}^n + h_j^n \cdot \left(1 - \frac{\alpha}{\Delta x_f \cdot \gamma_2} + \frac{\alpha \cdot \gamma_3}{\gamma_2} \right) + h_{j+1}^n \cdot \frac{\alpha}{\Delta x_f \cdot \gamma_2} - \alpha \cdot \frac{(S_b - \gamma_l)}{\gamma_2}$$

Pipe 5 height - Ignoring Backflow

Basis equations

$$h_j^{n+1} = \alpha \cdot (Q_{j-1}^n - Q_j^n) + h_j^n$$

Inflow:

$$Q_j^n = \frac{1}{\gamma_2} \cdot \left(h_j^n \cdot \left(\frac{1}{\Delta x_f} - \gamma_3 \right) - h_{j+1}^n \cdot \frac{1}{\Delta x_f} + S_b - \gamma_l \right)$$

↓ Adjust to correct Q

$$Q_{j-1}^n = \frac{1}{\gamma_2} \cdot \left(h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f} - \gamma_3 \right) - h_j^n \cdot \frac{1}{\Delta x_f} + S_b - \gamma_l \right)$$

outflow:

$$Q_j^n = \frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \cdot h_j - \frac{C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2}$$

Collect

$$h_j^{n+1} = \alpha \cdot \left(\left(\frac{1}{\gamma_2} \cdot \left(h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f} - \gamma_3 \right) - h_j^n \cdot \frac{1}{\Delta x_f} + S_b - \gamma_l \right) \right) - \left(\frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \cdot h_j \right. \right. \\ \left. \left. - \frac{C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2} \right) \right) + h_j^n$$

$$h_j^{n+1} =$$

$$\alpha \cdot \left(\left(h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f \cdot \gamma_2} - \frac{\gamma_3}{\gamma_2} \right) - h_j^n \cdot \frac{1}{\Delta x_f \cdot \gamma_2} + \frac{S_b - \gamma_l}{\gamma_2} \right) - \left(\frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \cdot h_j^n \right. \right. \\ \left. \left. - \frac{C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2} \right) \right) + h_j^n$$

$$h_j^{n+1} =$$

$$\alpha \cdot \left(h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f \cdot \gamma_2} - \frac{\gamma_3}{\gamma_2} \right) - h_j^n \cdot \frac{1}{\Delta x_f \cdot \gamma_2} + \frac{S_b - \gamma_l}{\gamma_2} - \frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \cdot h_j^n \right. \\ \left. + \frac{C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2} \right) + h_j^n$$

$$h_j^{n+1} = \alpha \cdot \left(h_{j-1}^n \cdot \left(\frac{1}{\Delta x_f \cdot \gamma_2} - \frac{\gamma_3}{\gamma_2} \right) - h_j^n \cdot \frac{1}{\Delta x_f \cdot \gamma_2} + \frac{S_b - \gamma_l}{\gamma_2} - \frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \cdot h_j^n + \frac{C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2} \right) + h_j^n$$

$$h_j^{n+1} = \frac{h_{j-1}^n \alpha}{\Delta x_f \gamma_2} - \frac{h_{j-1}^n \alpha \gamma_3}{\gamma_2} - \frac{\alpha h_j^n}{\gamma_2 \Delta x_f} + \frac{\alpha S_b}{\gamma_2} - \frac{\alpha \gamma_l}{\gamma_2} - \frac{3 \alpha C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}} h_j^n}{2} + \frac{\alpha C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2} + h_j^n$$

$$h_j^{n+1} = h_{j-1}^n \cdot \left(\frac{\alpha}{\Delta x_f \gamma_2} - \frac{\alpha \gamma_3}{\gamma_2} \right) + h_j^n \cdot \left(1 - \frac{\alpha}{\gamma_2 \Delta x_f} - \frac{3 \alpha C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \right) + \alpha \cdot \left(\frac{S_b - \gamma_l}{\gamma_2} + \frac{C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2} \right)$$

Replace with notation

$$\mu_1 := -\frac{C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2} = -\frac{1}{2} C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}$$

$$\mu_2 := \frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} = \frac{3}{2} C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}$$

$$h_j^{n+1} = h_{j-1}^n \cdot \left(\frac{\alpha}{\Delta x_f \gamma_2} - \frac{\alpha \gamma_3}{\gamma_2} \right) + h_j^n \cdot \left(1 - \frac{\alpha}{\gamma_2 \Delta x_f} - \alpha \cdot \mu_2 \right) + \alpha \cdot \left(\frac{S_b - \gamma_l}{\gamma_2} - \mu_1 \right)$$

▼ Tank 2 height - Ignoring Backflow

Basis equations

$$h_j^{n+1} = \alpha \cdot (Q_{j-1}^n - Q_j^n) + h_j^n$$

Inflow:

$$Q_j^n = \frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \cdot h_j^n - \frac{C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2}$$

↓ Ajust to correect Q

$$Q_{j-1}^n = \frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \cdot h_{j-1}^n - \frac{C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2}$$

outflow:

$$Q_j^n = Q_{FP2}^n$$

Collect

$$h_j^{n+1} = \alpha \cdot \left(\frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \cdot h_{j-1}^n - \frac{C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2} - Q_{FP2}^n \right) + h_j^n$$

$$h_j^{n+1} = \alpha \cdot \frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} \cdot h_{j-1}^n - \alpha \cdot \frac{C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2} - \alpha \cdot Q_{FP2}^n + h_j^n$$

Replace with notation

$$\mu_1 := -\frac{C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}}{2} = -\frac{1}{2} C_d \sqrt{2} \sqrt{g} h_{op}^{3/2}$$

$$\mu_2 := \frac{3 C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}}{2} = \frac{3}{2} C_d \sqrt{2} \sqrt{g} \sqrt{h_{op}}$$

$$h_j^{n+1} = \alpha \cdot \mu_2 \cdot h_{j-1}^n + \alpha \cdot \mu_1 - \alpha \cdot Q_{FP2}^n + h_j^n$$