# Stochastic Model Predictive Control

of Combined Sewer Overflows
in Sanitation Networks

Adis Hodžić, Casper Knudsen

Control and Automation, CA10-931, 2021

Master Thesis

AALBORG UNIVERSITY
STUDENT REPORT

**Nine and Tenth Semester | The Technical Faculty**
**of IT and Design**
Control and Automation
Fredrik Bajers Vej 7
9000 Aalborg

**Title:**

Stochastic Model Predictive Control of Combined Sewer Overflows in Sanitation Networks

**Period of Project:**

September 1 st 2020 - June 03th 2021

**Project Group:**

CA10-931

**Members:**

Adis Hodžić
Casper Knudsen

**Supervisors:**

Carsten Skovmose Kallesøe
Krisztian Mark Balla
Rahul Misra

**ECTS points: 50**
**Printed exemplary: 0**
**Page number: 126**
**Ended: June 3, 2021**



**Abstract**

Sanitation networks are vital infrastructure in modern society. They are used for transporting wastewater and rainwater from cities to treatment facilities, where wastewater is treated before being released into the environment. Most countries still use combined sanitation networks where wastewater and rainwater are transported in a single pipe. This leaves the combined sanitation network prone to overflow in the event of heavy rainfall. A solution to minimizing the overflow is Real Time Control (RTC). A popular state-of-the-art RTC method used to anticipate and minimize overflows is standard Model Predictive Control (MPC). However, the standard formulation of the MPC does face challenges when dealing with the uncertainty caused by the inflow disturbances, i.e., the weather forecasts. In order to better handle the uncertainties, we propose an extended model predictive framework called Chance-Constrained MPC (CC-MPC). First, the nominal multi-objective MPC is formulated to deal with the challenges in the sanitation network. Then, the framework is extended to our stochastic MPC formulation. Two controllers are compared in a laboratory emulation of the network subsystem that we call the Two Tank Topology. Gravity pipe elements determine the primary dynamics that define the transport of wastewater through a network. Both controller frameworks require a model that can capture gravity pipe dynamics in order to predict overflow. Therefore, we developed a linear Diffusion Wave model based on the discretized Saint-Venant partial differential equations. The model is validated through a data-driven parameter estimation framework. Identification is conducted in a real network simulation and in the real-life experimental setup created in AAU Smart Water Lab.

# Preface

Foremost, we would like to express our sincere gratitude to our primary supervisors Carsten Skovmose Kallesøe and Krisztian Mark Balla, for their patience, motivation, and willingness to teach two coming control engineers the ropes. Their guidance has helped us countless times through the writing of this thesis. We would also like to thank our supervisor Rahul Misra, for his encouragement and insightful comments.

Special recognition is given to Krisztian Mark Balla, who found the time to go above and beyond our wildest expectations on an almost daily basis. Whether we need help with lab work or 10 pm math problems, we could count on Krisztian. He found time to do this despite working on writing his Ph.D.

Aalborg University, June 3, 2021

Adis Hodžić
<ahodi19@student.aau.dk>

Casper Knudsen
<ckn16@student.aau.dk>

To get access to the code we develop during the thesis, scan the QR-code or go to:
`https://github.com/TheCasperKnudsen/Master_Thesis_2021_Results_MPC_for_Sanitation_Networks`

# Table of Contents

# Introduction 1

Sanitation networks are a cornerstone of modern civilization. The objective of sanitation networks is to handle routing and treatment of domestic wastewater discharge, industrial wastewater discharge and rainwater. A sanitation network typically consists of elements such as pipes, pumps, gates, and tanks used to transport or route wastewater and rain from wastewater producers to a Waste Water Treatment Plant (WWTP). In modern sanitation networks, discharge can be routed separately from rainwater. A system routing discharge and rainwater together is called a combined sanitation network, while a network that routes discharge separately is called a separated sanitation network [1]. In figure *Figure 1.1* an illustration of a combined sanitation network is shown.

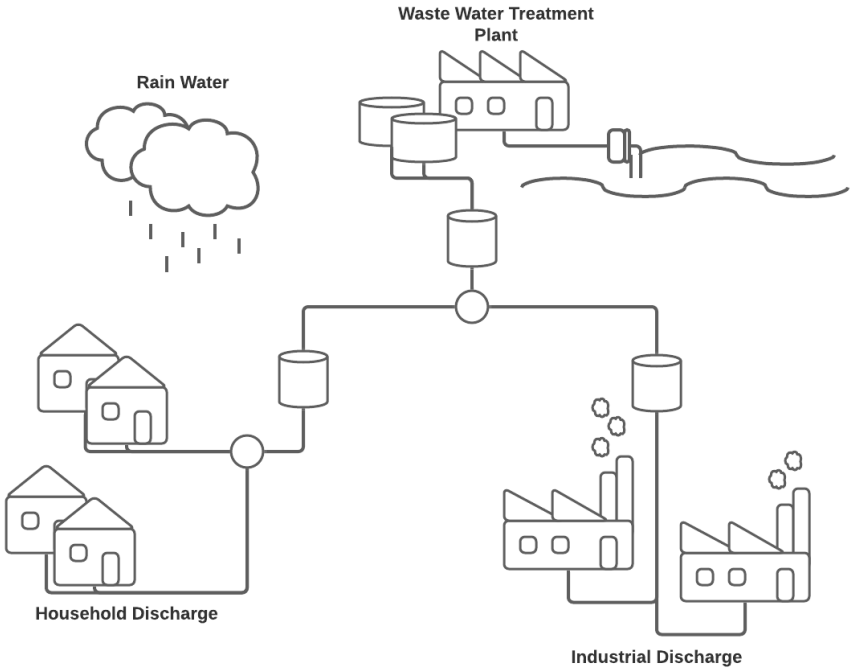



**Figure 1.1: Illustration of a combined sanitation network.**

During the later part of the 20th and 21st centuries, there has been an increase in rainfall and wastewater production. These increases can be attributed to three main factors: increased industrialization, increased urbanization, and climate change [1]. As most countries rely on centuries-old, now under dimensioned, combined sanitation networks, an increased amount of rainwater and discharge has led to a capacity problem. This is partially because it is difficult to quickly increase the capacity of a wastewater treatment plant during a heavy rain event, as treatment of wastewater is a biological process [2]. This

capacity problem can cause the network to overflow in the event of heavy rainfall. Though controlled overflows can happen, by discharging untreated waste into the sea or rivers, there is a possibility for untreated waste to be dispelled onto private or public property [3]. Overflow has a cost whether wastewater ends up on private property or in the environment [4].

The capacity problem manifests itself in two significant ways: in the treatment of discharge and the routing of discharge. Two approaches have been taken to solve these problems. The first approach focuses on increasing the treatment capacity as well as the routing capacity, while the other approach has been splitting the routing of sewage and rainwater into separate networks [5]. Both approaches are expensive as they require the construction of new infrastructure [6].

As a potential budget alternative to the aforementioned approaches, which result in the construction of new infrastructure, research efforts have been put towards optimal control using existing but not fully exploited infrastructure. One of these approaches is called Model Predictive Control (MPC) [6].

MPC of sanitation networks is a fairly young field, with researchers from multiple backgrounds contributing to it [1], [7]. Both deterministic MPC approaches [8] and stochastic MPC approaches have been presented [9]. Throughout the thesis, we further extend the understanding of how MPC approaches can be used in real-world combined sanitation networks.

*This thesis is dedicated to making two contributions to the field of Model Predictive Control (MPC) for combined sanitation networks:*

- The main contribution is made by presenting a performance comparison of two MPCs. Specifically, a Chance Constrained Model Predictive Controller (CC-MPC) and a standard Model Predictive Controller (MPC). The comparison is conducted as a scaled test in the AAU Smart Water Lab [10]. The performance comparison considers three criteria in prioritized order:
  1. Minimize overflow in a network.
  2. Avoid fast varying flow to a Waste Water Treatment Plant.
  3. Balance wastewater along available storage.
- The second contribution is made by presenting a novel, control-oriented, data-driven, linear diffusion wave model used for modeling a commonly found structure in real-world sanitation networks. We call this structure the Two Tank Topology.

The thesis is concerned with Linear, time discrete MPCs, under the influence of stochastic disturbance. The controllers are formulated as quadratic optimization problems, with deterministic constraints, and in the case of the CC-MPC, including chance constraints. Hence standard solvers can be utilized to solve the MPC problem efficiently.

Note that special attention has been paid to ensure that both contributions are applicable to real-world combined sanitation networks throughout the thesis. For this reason, our identification strategy uses data collected from level sensors, which are more cost-efficient and more robust to failure than typical flow sensors.

As model development has applications outside MPC, models have been developed for

sanitation networks for a long time, though the field of MPC for sanitation networks is fairly young [1]. Since model development has been done with different applications in mind, a number of approaches have been explored. The approaches can roughly be divided into heuristic approaches [11], and physics-based approaches [12], [13], [14]. While we considered using an existing model in our exploration of MPC, we ultimately developed our own. This model development resulted in our contribution to the paper: *"Nonlinear Grey-box Identification of Sewer Networks with the Backwater Effect: An Experimental Study"* [15]. Where we compared the performance of a nonlinear version of the model used in this thesis (the diffusion wave model) to another nonlinear model, called the kinematic wave model.

The remaining of the thesis is divided into 8 Chapters:

- **Chapter 2 - Modeling -** Presents the derivation of the diffusion wave model for the "Two Tank Topology," later used as a component of both MPC problems and the Kalman filter.
- **Chapter 3 - Simulation and Emulation Setup -** Outlines the two systems on which the model is identified. It also introduces the AAU Smart Water Lab, which is the system used to produce the results for the comparisons of the controllers.
- **Chapter 4 - Model Parameter Identification -** Describes the process which is used to identify the model parameters of the simulation and emulation setup. The identification process is a grey-box, data-driven identification process.
- **Chapter 5 - Kalman Filter** Outlines the theory of the standard Kalman filter. It also presents a noise identification process used to identify the filter parameters. The Kalman filter is later used to estimate initial states for a MPC during the comparison.
- **Chapter 6 - Controller Design** First presents the theory of the standard MPC before expanding it to include chance constraints. The chapter also presents a simple application example of the CC-MPC used on a single tank.
- **Chapter 7 - Controlling the Two Tank Topology -** Is the culmination of the previous chapters. The chapter summarizes the theory presented previously as it pertains to the control of the Two Tank Topology. Further, it presents a comparison study of the standard MPC and the CC-MPC based on results obtained from the emulation setup. The chapter ends with a short conclusion on the comparison.
- **Chapter 8 - Discussion of the Work -** Presents a discussion of the methods and theory used throughout the thesis.
- **Chapter 9 - Conclusion -** Concludes on the work done throughout this thesis.

# Modeling 2

Combined sanitation network is an umbrella term covering underground pipe or tunnel systems used to transport sewage and rainwater from residential and commercial areas to treatment facilities or disposal. Modeling a complete sanitation network is outside the scope of this thesis, as sanitation networks generally include a vast array of network components with several sub-networks.

Instead, we focus on modeling a single topology commonly found in sanitation networks used in urban areas. We have dubbed this network topology the *Two Tank Topology*. The Two Tank Topology is shown in *Figure 2.1* and consists of two tanks, two pumps, a rising main, and a gravity pipe.

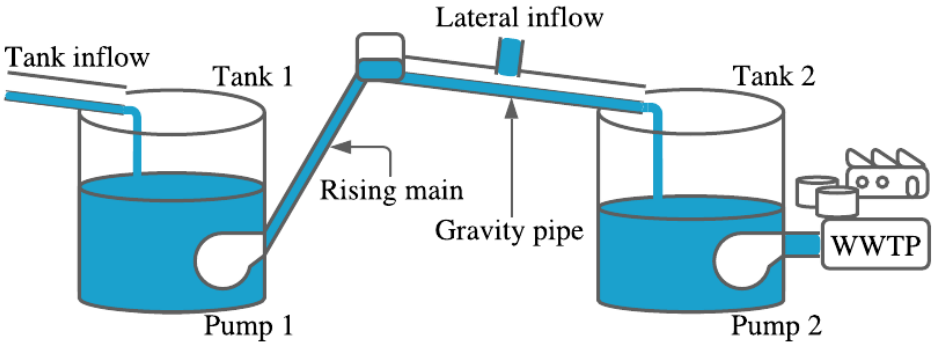



**Figure 2.1: Illustration of the Two Tank Topology with each component named.**

This chapter aims to develop a linear state space model for the Two Tank Topology, which can be used in a MPC. Choosing this topology limits us from having to model the effects of gates, weirs, and other network elements that lend themselves poorly to linear models.

The chapter is organized into two sections: In *section 2.1 Model Elements* a linear description of each of the previously mentioned network element is presented. And in *section 2.2 General Model* a general state space description of the Two Tank Topology is made using the already developed models of the system elements.

## 2.1 Model Elements

In this section, a linear, time and space discretized mathematical description of each element in the Two Tank Topology is created. We chose to develop time discretized mathematical models rather than continuous time models as they lend themselves to ease of implementation in digital control and simulation. The goal of developing individual element descriptions, is to use them in a linear state space model.

We first present a short discussion of the relevancy of including each element's dynamics in the model of the Two Tank Topology. In the Two Tank Topology, we consider the following elements:

- Two tanks,
- a rising main,
- two pumps
- and a gravity pipe.

Rising mains are generally always completely filled with wastewater. This leads to fast dynamics, as whenever the pump supplying the rising main is turned on, sewage will start flowing out of the main without significant delay. The fast dynamics lead to the rising main being of little interest in this project.

The dynamics of the Gravity pipe are significantly slower than the dynamics of the rising main. The most critical dynamic exhibited by a gravity pipe is, in fact, a delay, and some simple models only consider this delay [16]. However, in a control environment, it can be useful to have a more complex model of a gravity pipe, as such a model will include other relevant dynamics.

Having discussed gravity pipe dynamics, it is also important to touch on tank dynamics. The dynamics of a tank are described as an integration of flow. This means that in and out-flow is accumulated. Therefore we can assume that if a pump actuates the tank, the dynamics of the tank are described by the dynamics of the pump in addition to the integration. However, we assume that the pump dynamics are very fast compared to the integration dynamics; therefore, we choose to neglect them.

When a flow phenomenon governs a tank's in or out-flow, it is important to include this in the description of the tank dynamics. This is the case for the inflow of tank two in the Two Tank Topology; therefore, we explore two outflow phenomenons in this section.

### 2.1.1 Gravity pipe

Care has been put into developing a novel model for the gravity pipe. This is in spite of a lot of different models already having been developed by other authors. We chose to develop our own model for the gravity pipe as we (Both authors) were contributing to a paper on modeling gravity driven sewers at the time of writing the thesis [15]. Since we did some model development during a previous semester, some results may be identical to the ones shown in the previous semester's report [17].

It is common for models of the gravity pipe to utilize the Saint Venant equations as a starting point. We elected to take these equations as our starting point as well. Some

authors focus on developing linear state space modes [18] as we have, other focus on developing transfer function models [19], [20].

The Saint-Venant equations consist of two specific equations, which are derived using Reynolds transport theorem. This theorem describes the dynamic behavior of fluid systems in general [17,page 19-25]. The two Saint-Venant equations are shown in equation (2.1) and (2.2).

$$\frac{\partial A(x,t)}{\partial t} = -\frac{\partial Q(x,t)}{\partial x}, \tag{2.1}$$

$$S_f - S_b + \frac{\partial h(x,t)}{\partial x} + \frac{1}{gA(x,t)} \left( \frac{\partial}{\partial x} \left( \frac{Q(x,t)^2}{A(x,t)} \right) + \frac{\partial Q(x,t)}{\partial t} \right) = 0, \tag{2.2}$$

where $A(x,t)$ is the wetted area, $Q(x,t)$ is the flow inside the channel and $h(x,t)$ is the water level in the channel. These are functions of time $t$ and space $x$. In control applications with sanitation networks as the focus, models utilizing the full Saint-Venant equations are seldom used. This is because the precision a model gains from having additional terms, often does not outweigh the increase in model complexity. We elect to use a reduced version of the Saint-Venant equations, called the diffusion wave equations, which ignores the terms $\frac{\partial}{\partial x} \left( \frac{Q(x,t)^2}{A(x,t)} \right) + \frac{\partial Q(x,t)}{\partial t}$ of (2.2) yielding the equation:

$$S_f - S_b + \frac{\partial h(x,t)}{\partial x} = 0. \tag{2.3}$$

Here $S_f$ is the friction slope of the gravity pipe and $S_b$ the pipe beds slope. We can find an expression for the friction slope using the Darcy–Weisbach equation [21]. This expression is, however, highly nonlinear. To include the effects of the bed's slope in a linear state space model, we elect to linearize the nonlinear Darcy–Weisbach equation around an operating point using Taylor approximation. This yields the equation:

$$(\gamma_1 + \gamma_2 Q(x,t) + \gamma_3 h(x,t)) - S_b + \frac{\partial h(x,t)}{\partial x} = 0, \tag{2.4}$$

where $\gamma_1$, $\gamma_2$ and $\gamma_3$ are parameters caused by linearization, expressed in (2.5), (2.6) and (2.7) [17,pages 40-43].

$$\gamma_1 = \frac{f}{2 \cdot g} \left( \frac{\left(4 \cdot w + 6 \cdot h_{op}\right)}{4 \cdot w^3 {h_{op}}^3} \cdot \left|Q_{op}\right| Q_{op} - \frac{\left(2 \cdot w + 4 \cdot h_{op}\right)}{4 \cdot w^3 {h_{op}}^3} \cdot \frac{{Q_{op}}^3}{\left|Q_{op}\right|} \right), \tag{2.5}$$

$$\gamma_2 = \frac{f}{2 \cdot g} \frac{\left(w + 2 \cdot h_{op}\right)}{4 \cdot w^3 {h_{op}}^3} \cdot \frac{{Q_{op}}^3}{\left|Q_{op}\right|}, \tag{2.6}$$

$$\gamma_3 = -\frac{f}{2 \cdot g} \frac{\left(4 \cdot h_{op} + 3 \cdot w\right)}{4 \cdot w^3 \cdot {h_{op}}^4} \cdot \left|Q_{op}\right| Q_{op}. \tag{2.7}$$

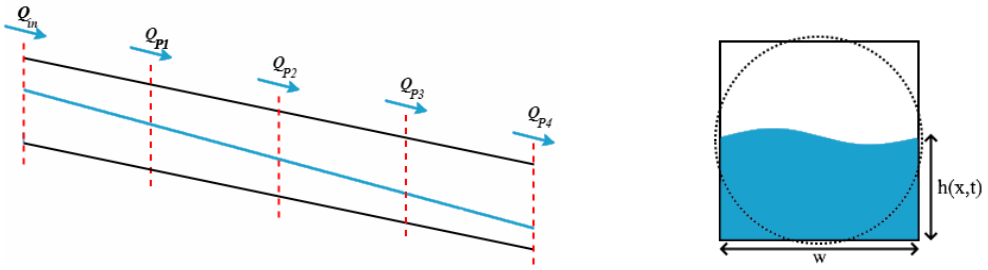
Solving (2.4) for $-Q(x,t)$ allows us to insert it into the (2.1). This yields the partial differential equation of two variables shown as (2.9).

$$-Q(x,t) = \frac{1}{\gamma_2}(\gamma_1 + \gamma_3 h(x,t) - S_b + \frac{\partial h(x,t)}{\partial x}), \tag{2.8}$$

$$\Downarrow$$

$$\frac{\partial A(x,t)}{\partial t} = \frac{1}{\gamma_2}(\frac{\partial^2 h(x,t)}{\partial x^2} + \gamma_3 \frac{\partial h(x,t)}{\partial x}). \tag{2.9}$$

To get a partial differential equation of one variable, we want to express the relation between the wetted area $A(x,t)$ and water level $h(x,t)$. For a round pipe, this relation is nonlinear, so to have the model remain linear, we assume the pipe is square, yielding the relation $A(x,t) = w \cdot h(x,t)$, where $w$ is the width of the square pipe, see *Figure 2.2b*.




**(a) Dividing the pipe into sections.** **(b) Square pipe approximation.**

**Figure 2.2: Partitioning and approximation of a pipe [17].**

This yields the partial differential equation on the form (2.10):

$$\frac{\partial h(x,t)}{\partial t} = \frac{1}{\gamma_2 \cdot w}\left(\frac{\partial^2 h(x,t)}{\partial x^2} + \gamma_3 \frac{\partial h(x,t)}{\partial x}\right). \tag{2.10}$$

Second order partial differential equations can be expressed on the standard form:

$$R\,\frac{\partial^2 h(x,t)}{\partial x^2} + S\,\frac{\partial^2 h(x,t)}{\partial x \partial t} + T\,\frac{\partial^2 h(x,t)}{\partial t^2} = W(x,t,h(x,t),\frac{\partial h(x,t)}{\partial x},\frac{\partial h(x,t)}{\partial t}),$$

they are characterized by the value of their roots: $S^2 - 4RT$ where the partial differential equation is said to be *hyperbolic* if $S^2 - 4RT > 0$, *parabolic* if $S^2 - 4RT = 0$ and *elliptic* if $S^2 - 4RT < 0$ [22,pages 24-25]. Rearranging (2.10) to standard form:

$$\frac{\partial h(x,t)}{\partial t} - \frac{\gamma_3}{\gamma_2 \cdot w}\frac{\partial h(x,t)}{\partial x} = \frac{1}{\gamma_2 \cdot w}\frac{\partial^2 h(x,t)}{\partial x^2},$$

we see that $T = 0$, $S = 0$ and $R = \frac{1}{\gamma_2 \cdot w}$, so therefore (2.10), is a parabolic partial differential equation.

The fact that (2.10) is parabolic allows us to apply numerical analysis as a way of time and space discretizing the equation. This discretization is done using finite difference approximations for the derivatives in (2.10) [22,pages 29-33]. We specifically use the forward difference approximation, backward difference approximation[1] and central difference approximation. This yields (2.11):

$$\underbrace{\frac{\partial h(x,t)}{\partial t}}_{\text{Forward difference}} = \frac{1}{\gamma_2 \cdot w}\Big(\underbrace{\frac{\partial^2 h(x,t)}{\partial x^2}}_{\text{Central difference}} + \gamma_3 \underbrace{\frac{\partial h(x,t)}{\partial x}}_{\text{Backward difference}}\Big),$$

$$\Downarrow$$

$$\frac{h_j^{n+1} - h_j^n}{\Delta t} = \frac{1}{\gamma_2 \cdot w}\left(\frac{h_{j-1}^n - 2 \cdot h_j^n + h_{j+1}^n}{\Delta x^2} + \gamma_3 \frac{h_j^n - h_{j-1}^n}{\Delta x}\right). \tag{2.11}$$

[1] Often referred to as forward and backward Euler approximation [17]

Generally, discretizing in both time and space can be seen as "gridding" the space of solutions $(t, x)$ [22,page. 29]. We will be using the now discretized partial differential equation for control applications. We will rewrite it on state space form. Note that the choice of using the forward difference approximation in time is what allows us to rewrite (2.11) on a state space form since the forward approximation introduces $h_j^{n+1}$.

$$h_j^{n+1} = \frac{\Delta t}{w \cdot \Delta x}\left(h_{j-1}^n\left(\frac{1-\gamma_3 \cdot \Delta x}{\gamma_2 \cdot \Delta x}\right) - h_j^n\left(\frac{2-\gamma_3 \cdot \Delta x}{\gamma_2 \cdot \Delta x}\right) + h_{j+1}^n \frac{1}{\gamma_2 \cdot \Delta x}\right) + h_j^n. \quad (2.12)$$

Having (2.11) rewritten as (2.12), a certain observation can be made. It arises from the relation $\frac{1-\gamma_3 \cdot \Delta x}{\gamma_2 \cdot \Delta x} > \frac{1}{\gamma_2 \cdot \Delta x}$, between the effects of the up- and down-stream levels[2]. The relation shows that the level downstream of a pipe section has a lesser effect on the level in the pipe section at the next time step than the level of the upstream pipe section. It occurs as a result of using backward difference approximation on the $\frac{\partial h(x,t)}{\partial x}$-term in (2.10).

In order to get a high resolution model of a pipe, it is advantages to divide a pipe into a number of sections as shown in *Figure 2.2a*. This will yield a set of linear equations, as seen in *section 2.2 General Model*.

Looking at *Figure 2.2a* it can be observed that we need to define the boundary conditions for the pipe. These boundary conditions are the descriptions of the first pipe section $j = 1$ and the last pipe section $j = N_x$, where $N_x$ denotes the total number of sections. If equation (2.12) is to be used to describe boundary sections, then it is necessary to define $h_0$ and $h_{N_x+1}$.

However, describing the sections at either end of the pipe can be more complicated than using the flows in and out of the pipe. This is because the flows can often be directly measured or estimated. If the spacial coordinate of the flow in the mass balance equation (2.1) is discretized using the backward difference approximation, equation (2.13) is obtained. Note that additionally $\frac{\partial h}{\partial t}$ is discretized in the same way as in (2.11) to achieve the state space formulation.

$$h_j^{n+1} = \frac{\Delta t}{w \cdot \Delta x}\left(Q_{j-1}^n - Q_j^n\right) + h_j^n. \quad (2.13)$$

Now the inlet section, $Q_{j-1}$ from (2.13) is defined by the pipe inflow, while the $Q_j$ term is described by (2.8). Likewise, in the last section of the pipe, $Q_j$ is defined by the gravity pipe outflow and $Q_{j-1}$ is described by (2.8). The complete derivation can be found in [17,page 43-45]. The equation (2.13) will become useful later when pipes bordering other network elements have to be described.

### 2.1.2 Retention tank

The retention tank equation is derived from the mass conservation theorem. A tank acts as a flow integrator in a system. The dynamics of the tank is described by the linear differential equation shown in (2.14) [17,page 46]:

$$\frac{d(M_{tank}(t))}{dt} = m_{in}(t) - m_{out}(t). \quad (2.14)$$

[2]This is assuming that $\gamma_2$ is positive and $\gamma_3$ is negative, which is the case for positive $\Delta x$ and $\Delta t$, see (2.6) to (2.7) [17]

To use (2.14) in a linear, time discrete model, it is time discretized using the forward finite difference method and shown in (2.15):

$$h_j^{n+1} = \frac{\Delta t}{A}(Q_{in}^n - Q_{out}^n) + h_j^n = \phi \Delta t(Q_{in}^n - Q_{out}^n) + h_j^n, \tag{2.15}$$

where $\phi = \frac{1}{A}$ is the tank model parameter and $Q_{in}$ and $Q_{out}$ are respectively the inflows and the outflows. In the Two Tank Topology, the outflow from both tanks consists of a flow caused by actuation ($Q_{FP}$, $FP$ - flow provider) as well as a flow caused by the tank overflowing($Q_{of}$). If the tank is a part of a network, the inflow and outflow of the tank can be a boundary condition for bordering network elements. Different boundary conditions will be described in the following subsection.

### 2.1.3 Pipe boundary conditions

Boundary conditions are used to describe the relationship between bordering network elements, such as the flow relation between tanks and pipes. This section presents three boundary equations, one using a pump and two boundary equations relating the outflow from a gravity pipe to the inflow to a tank.

Pumps are treated as "flow providers" or free variables in a network model. This is because the pump flow is the variable manipulated in a control setting. Therefore the pump flow is often written as:

$$Q^n = u^n, \tag{2.16}$$

where $Q$ is the flow of the pump, $u$ is the control input.

The flow from a rectangular pipe into a tank can be described using two different equations depending on whether the pipe is free-flowing or flows into a tank below the tank's waterline. A free-flowing pipe can be assumed to exhibit the same dynamics as the flow over the sharp-crested weir [23]. In "Modeling and Control of Hydrosystems" [24,pages 160-162] a model for weir flow is presented, assuming a weir crest $W_s = 0$. Free flow into a tank as seen in *Figure 2.3a* is expressed as:

$$Q_j^n = C_d\sqrt{2g} \cdot (h_j^n)^{3/2}, \tag{2.17}$$

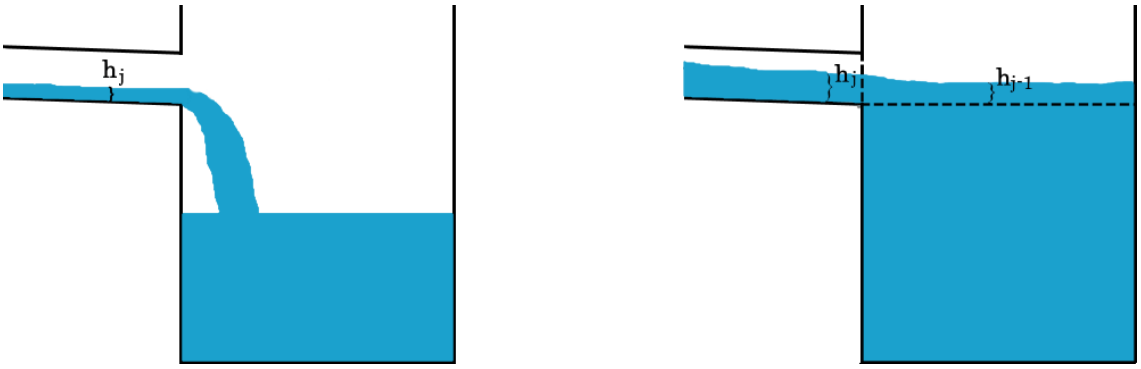
where $C_d$ is a discharge coefficient.

Equation (2.17) is a nonlinear function of water level. To make (2.17) applicable in a linear model we choose to neglect the power term in (2.17), which yields:

$$Q_j^n \approx C_d\sqrt{2g} \cdot h_j^n. \tag{2.18}$$

If the pipe outlet is below the water level in the tank, see *Figure 2.3b*, a different equation should be used to describe the tank inflow. This phenomenon is called submerged flow and is described by the submerged flow equation. The submerged flow equation (2.19) is similar to the free flow equation (2.17) except it includes the water level in the tank as well as the water level in the pipe. The equation for submerged flow is shown in (2.19).

$$Q_j^n = C_d\sqrt{2g} \cdot (h_j^n - h_j^{n+1})^{3/2}. \tag{2.19}$$




**(a) Free-flow condition.** **(b) Submerged flow condition.**

**Figure 2.3: Illustrations of the two considered tank inflows.**

Equation (2.19) is linearized like (2.17) yielding:

$$Q_j^n \approx C_d\sqrt{2g} \cdot (h_j^n - h_j^{n+1}). \tag{2.20}$$

The reason for simply removing the power rather than expanding the nonlinear equations into a second order Taylor series lies in the fact that a $Q(h) = 0$ relation is wanted at $Q(0)$. This is in order to ensure that the free flow condition is linear and not afine-linear.

## 2.2 General Model

With a description of the components that make up the Two Tank Topology and expressions for component boundaries, we begin to describe the Two Tank Topology as a set of state equations. The state equations are used to represent the Two Tank Topology on a discrete state space form. The general form of a state space system is shown in (2.21).

$$\begin{aligned} \boldsymbol{x}^{n+1} &= \boldsymbol{A}\boldsymbol{x}^n + \boldsymbol{B}\boldsymbol{u}^n, \\ \boldsymbol{y}^n &= \boldsymbol{C}\boldsymbol{x}^n + \boldsymbol{D}\boldsymbol{u}^n. \end{aligned} \tag{2.21}$$

Where **Bold** indicates that a variable or constant is a matrix or vector and $\boldsymbol{x}$ is the state vector, $\boldsymbol{y}$ is the output vector, $\boldsymbol{u}$ is the input vector, $\boldsymbol{A}$ is the system matrix, $\boldsymbol{B}$ is the input matrix $\boldsymbol{C}$ is the output matrix and $\boldsymbol{D}$ is the feedthrough matrix.

The first state equation was presented in *subsection 2.1.1 Gravity pipe* as equation (2.12). It describes how the water level in a pipe relates to the water level of neighbouring pipe sections. The equation is again stated below, and expressed for generalized parameters: $\theta_1 \triangleq \frac{1}{w \cdot \Delta x}$, $\theta_2 \triangleq \frac{1-\gamma_3 \cdot \Delta x}{\gamma_2 \cdot \Delta x} \cdot \theta_1$ and $\theta_3 \triangleq \frac{1}{\gamma_2 \cdot \Delta x} \cdot \theta_1$.

$$\begin{aligned} h_j^{n+1} &= \frac{\Delta t}{w \cdot \Delta x_b}\left(h_{j-1}^n\left(\frac{1-\gamma_3 \cdot \Delta x}{\gamma_2 \cdot \Delta x}\right) - h_j^n\left(\frac{1}{\gamma_2 \cdot \Delta x} + \frac{1-\gamma_3 \cdot \Delta x}{\gamma_2 \cdot \Delta x}\right) + h_{j+1}^n\frac{1}{\gamma_2 \cdot \Delta x}\right) + h_j^n \\ &= \Delta t(h_{j-1}^n\theta_2 - h_j^n\,(\theta_3 + \theta_2) + h_{j+1}^n\theta_3) + h_j^n. \end{aligned} \tag{2.22}$$

To describe the full topology we have to describe this same relation for pipe sections boarding pumps and tanks, see (2.13).

For the first pipe section $h_j^n = h_1^n$, which borders a pump, the outflow $Q_{out}$ can be found by applying the forward difference method to (2.4) such that we get:

$$Q_{out}^n = \left( h_j^n \left( \frac{1 - \gamma_3 \cdot \Delta x}{\gamma_2 \cdot \Delta x} \right) - h_{j+1}^n \frac{1}{\gamma_2 \cdot \Delta x} + \frac{S_b - \gamma_1}{\cdot \gamma_2} \right), \tag{2.23}$$

Then if we assume that a pump provides the inflow $Q_{in} = Q_{FP1}$ into the pipe section, the state equation for the first pipe section is:

$$\begin{aligned} h_1^{n+1} &= \frac{\Delta t}{w \cdot \Delta x_b} \left( Q_{FP1} - h_1^n \left( \frac{1 - \gamma_3 \cdot \Delta x}{\gamma_2 \cdot \Delta x} \right) + h_2^n \frac{1}{\gamma_2 \cdot \Delta x} - \frac{S_b - \gamma_1}{\gamma_2} \right) + h_1^n \\ &= \Delta t (Q_{FP1}\theta_1 - h_1\theta_2 + h_2\theta_3 - \theta_4) + h_1^n. \end{aligned} \tag{2.24}$$

Where $\theta_4 \triangleq \frac{S_b - \gamma_l}{\gamma_2}\theta_1$ is another generalized system parameter.

The inflow into the last pipe section $h_j^n = h_L^n$ is described by applying the backward difference method to (2.4) such that we get:

$$Q_{in}^n = \left( h_{j-1}^n \left( \frac{1 - \gamma_3 \cdot \Delta x}{\gamma_2 \cdot \Delta x} \right) - h_j^n \frac{1}{\gamma_2 \cdot \Delta x} + \frac{S_b - \gamma_1}{\cdot \gamma_2} \right). \tag{2.25}$$

Then using the the free flow boundary conditions described in *subsection 2.1.3 Pipe boundary conditions* as outflow we get the following equation:

$$\begin{aligned} h_L^{n+1} &= \frac{\Delta t}{w \cdot \Delta x_b} \left( h_{L-1}^n \left( \frac{1 - \gamma_3 \cdot \Delta x}{\gamma_2 \cdot \Delta x} \right) - h_L^n \frac{1}{\gamma_2 \cdot \Delta x} + \frac{S_b - \gamma_1}{\gamma_2} - C_d \sqrt{2g} h_L^n \right) + h_L^n \\ &= \Delta t \ (h_{L-1}^n \theta_2 - h_L^n \theta_3 + \theta_4 - \theta_5 h_L^n) + h_L^n. \end{aligned} \tag{2.26}$$

Note that the model shown here only consider the free flow boundary case. With state equations for all pipe sections in the Two Tank Topology, as well as a description of the tank from *subsection 2.1.2 Retention tank*, we can present a full state space representation of the Two Tank Topology:

$$\begin{aligned} \begin{bmatrix} h_{T1}^{n+1} \\ h_{P1}^{n+1} \\ h_{P2}^{n+1} \\ \vdots \\ h_{PL}^{n+1} \\ h_{T2}^{n+1} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 - \theta_2 \Delta t & \theta_3 \Delta t & 0 & 0 & \cdots & 0 \\ 0 & \theta_2 \Delta t & 1 - (\theta_2 + \theta_3)\Delta t & \theta_3 \Delta t & 0 & \cdots & 0 \\ \vdots & & & \ddots & & & \\ 0 & & \cdots & 0 & \theta_2 \Delta t & 1 - (\theta_5 + \theta_3)\Delta t & 0 \\ 0 & & \cdots & 0 & 0 & \phi_2 \frac{\theta_5}{\theta_1} \Delta t & 1 \end{bmatrix} \cdot \begin{bmatrix} h_{T1} \\ h_{P1}^n \\ h_{P2}^n \\ \vdots \\ h_{PL}^n \\ h_{T2}^n \end{bmatrix} \\ &+ \begin{bmatrix} -\phi_1 \Delta t & 0 & -\phi_1 \Delta t & 0 \\ \theta_1 \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \vdots & & \ddots & \\ 0 & 0 & 0 & 0 \\ 0 & -\phi_2 \Delta t & 0 & -\phi_2 \Delta t \end{bmatrix} \cdot \begin{bmatrix} Q_{FP1} \\ Q_{FP2} \\ Q_{ofTI} \\ Q_{ofT2} \end{bmatrix} + \begin{bmatrix} 0 \\ -\theta_4 \Delta t \\ 0 \\ \vdots \\ 0 \\ \theta_4 \Delta t \\ 0 \end{bmatrix} \end{aligned} \tag{2.27}$$

Equation (2.27) deviates from the form presented in (2.21) at the start of this section by an affine-term. The affine-term is unavoidable, unless the operating points used while linearizing the gammas in the $\gamma$'s are set to zero, see (2.5) to (2.7). For now we will assume that $\boldsymbol{C} = \boldsymbol{I}$ and $\boldsymbol{D} = 0$.

### 2.2.1 Stability of the General Model

In this section we want to explore the conditions for stability of the general system shown in (2.27).

When a partial differential equation is discretized, the solutions of the now discretized equation may be unstable, even if the original equations described the behavior of a stable system. In this section, we look at the stability of the general model presented in (2.27). We start by defining the stability of a discrete LTI system:

Stability is a condition on the numerical solution. For a stable discrete system, all outputs must remain bounded for bounded inputs as time progress $n \to \infty$. This implies that all errors must remain bounded for bounded inputs as well. More precisely for a discrete linear state space system, on the form:

$$\begin{aligned} \boldsymbol{x}^{n+1} &= \boldsymbol{A}\boldsymbol{x}^n + \boldsymbol{B}\boldsymbol{u}^n, \\ \boldsymbol{y}^n &= \boldsymbol{C}\boldsymbol{x}^n + \boldsymbol{D}\boldsymbol{u}^n, \end{aligned} \tag{2.28}$$

stability is defined by the roots of the characteristic polynomial, $(z\boldsymbol{I} - \boldsymbol{A})^{-1}$ . The characteristic polynomial arises from the z-transform of a continuous system [25]. Specifically, for a system to be causal and stable, the unit circle must be included in the region of convergence of the z-transform, and the system has to be right-sided. This is equivalent to stating that the absolute value of the roots of $(z\boldsymbol{I} - \boldsymbol{A})^{-1}$ has to be less than 1 [26,pages. 105-121]. Or again is equivalent to stating $|eig(\boldsymbol{A})| < 1$.

To ease the analysis of the general system shown in (2.27) we elect to decompose it into three subsystems: One for the first tank $A_1 \in \mathcal{R}$, one for the gravity pipe sections $\boldsymbol{A_2} \in \mathcal{R}^{N_x \times N_x}$, where $N_x$ denotes the number of pipe sections, and one for the last tank $A_3 \in \mathcal{R}$. For systems connected in series, decomposition can be done as shown in (2.29).

$$\boldsymbol{A} = \left[\begin{array}{c|c|c} \boldsymbol{A_1} & 0 & 0 \\ \hline \boldsymbol{B_2} \cdot \boldsymbol{C_1} & \boldsymbol{A_2} & 0 \\ \hline 0 & \boldsymbol{B_3} \cdot \boldsymbol{C_2} & \boldsymbol{A_3} \end{array}\right]. \tag{2.29}$$

It can be shown that for systems connected in series, it holds that the decomposition has the following property:

$$eig(\boldsymbol{A}) = eig(\boldsymbol{A_1}) \cup eig(\boldsymbol{A_2}) \cup eig(\boldsymbol{A_3}).$$

That is, the set of eigenvalues of $\boldsymbol{A}$ is the union of the sets eigenvalues for the decomposed systems. We can therefore choose to analyze the decomposed systems individually. A decomposition of the system from (2.27) is shown in (2.30), assuming four pipe sections, $N_x = 4$.

$$\boldsymbol{A} = \left[\begin{array}{c|cccc|c} 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1-\theta_2\Delta t & \theta_3\Delta t & 0 & 0 & 0 \\ 0 & \theta_2\Delta t & 1-(\theta_2+\theta_3)\,\Delta t & \theta_3\Delta t & 0 & 0 \\ 0 & 0 & \theta_2\Delta t & 1-(\theta_2+\theta_3)\,\Delta t & \theta_3\Delta t & 0 \\ 0 & 0 & 0 & \theta_2\Delta t & 1-(\theta_5+\theta_3)\,\Delta t & 0 \\ \hline 0 & 0 & 0 & 0 & \phi_2\frac{\theta_5}{\theta_1}\Delta t & 1 \end{array}\right]. \tag{2.30}$$

Observing the systems $A_1$ and $A_3$, it is easy to see that their eigenvalues are equal to 1. This means that the tanks are marginally stable. This fits well with the intuitive understanding of a tank as an integrator. We are, however, able to actuate both tanks directly using pumps. Therefore it is possible to stabilize the tank levels as a part of closed loop control.

Opposed to $A_1$ and $A_3$ we expect $\boldsymbol{A_2}$ to be asymptotically stable for the model to behave in accordance with our physical intuition of the system. We expect any amount of water added to a pipe section to propagate down the pipe and into tank 2. In other words we expect the that a system with initial states $\boldsymbol{x_0}$ of the form $\boldsymbol{x}^{n+1} = \boldsymbol{A}\boldsymbol{x}^n$ to propagate to $\mathbf{0}$ as $n \to \infty$.

Finding exact eigenvalues for tridiagonal matrices expressed generally is an active research field. Said Kouachi from Qassim University has published several papers on eigenvalues of tridiagonal matrices [27], [28]. Although $\boldsymbol{A_2}$ has some nice properties, an expression for its exact eigenvalues has not been developed. Rather than undertaking the task of finding exact eigenvalues, we will try and define an upper limit for $\boldsymbol{A_2}$'s eigenvalues. This can be done using spectral density. The spectral density of a matrix is defined as shown in (2.31).

$$\rho(\boldsymbol{A}) = \max\{|\lambda| : \lambda \in \sigma(\boldsymbol{A})\}. \tag{2.31}$$

Where $\sigma(\boldsymbol{A})$ is the spectrum[3] of $\boldsymbol{A}$. We know that if a matrix is square, then the upper bound for the spectral radius is given by (2.32) [29,page. 390].

$$\rho(\boldsymbol{A}) \leq \min\{\max_i \sum_{j=1}^{N_x} |a_{ij}|, \max_j \sum_{i=1}^{N_x} |a_{ij}|\}, \tag{2.32}$$

which after substituting into the spectral density in (2.31), becomes:

$$\max\{|\lambda| : \lambda \in \sigma(\boldsymbol{A})\} \leq \min\{\max_i \sum_{j=1}^{N_x} |a_{ij}|, \max_j \sum_{i=1}^{N_x} |a_{ij}|\}. \tag{2.33}$$

Restating $\boldsymbol{A_2}$ we see it includes the terms $\theta_1$, $\theta_2$, $\theta_3$ and $\theta_5$.

$$\boldsymbol{A_2} = \begin{bmatrix} 1-\theta_2\Delta t & \theta_3\Delta t & 0 & 0 \\ \theta_2\Delta t & 1-(\theta_2+\theta_3)\Delta t & \theta_3\Delta t & 0 \\ 0 & \theta_2\Delta t & 1-(\theta_2+\theta_3)\Delta t & \theta_3\Delta t \\ 0 & 0 & \theta_2\Delta t & 1-(\theta_5+\theta_3)\Delta t \end{bmatrix}$$

Examining the expressions for $\theta$'s, we note that they are positive, for positive $\Delta x$ and $w$ as we know that $C_d$, $g$ and $\gamma_2$ are positive and $\gamma_3$ is negative. The $\gamma$s are shown in (2.5) to (2.7) and the expressions for the $\theta$'s are shown in (2.34) to (2.37).

$$\theta_1 = \frac{1}{w \cdot \Delta x} \tag{2.34}$$

$$\theta_2 = \frac{1-\gamma_3 \cdot \Delta x}{\gamma_2 \cdot \Delta x}\theta_1 \tag{2.35}$$

$$\theta_3 = \frac{1}{\gamma_2 \cdot \Delta x}\theta_1 \tag{2.36}$$

$$\theta_5 = C_d\sqrt{2g}\theta_1 \tag{2.37}$$

[3]The spectrum of a matrix is the set of its eigenvalues

Then examining $\boldsymbol{A_2}$ we see that $\boldsymbol{A_2}$ will have all positive entries as long as $(\theta_2+\theta_3)\Delta t < 1$ and $(\theta_5+\theta_3)\Delta t < 1$. We therefore assume $(\theta_2+\theta_3)\Delta t < 1$ and $(\theta_5+\theta_3)\Delta t < 1$. This assumption has turned out to hold during the identification processes, which yielded stable systems, see *chapter 4 Model Parameter Identification*. Furthermore the assumptions are an indirect result of a physical property of the system, called: the conversion of mass.

Applying (2.32) to find the upper bound for eigenvalues of $\mathbf{A_2}$ yields the row and column sums:

$$\sum_{j=1}^{4} |a_{ij}| = r_1 = |1-\theta_2\Delta t| + \theta_3\Delta t$$
$$r_2 = r_3 = \theta_2\Delta t + |1-(\theta_2+\theta_3)\Delta t| + \theta_3\Delta t = 1$$
$$r_4 = \theta_2\Delta t + |1-(\theta_5+\theta_3)\Delta t|$$

$$\sum_{i=1}^{4} |a_{ij}| = c_1 = |1-\theta_2\Delta t| + \theta_2\Delta t = 1$$
$$c_2 = c_3 = \theta_3\Delta t + |1-(\theta_2+\theta_3)\Delta t| + \theta_2\Delta t = 1$$
$$c_4 = \theta_3\Delta t + |1-(\theta_5+\theta_3)\Delta t| \leq |1-\theta_5\Delta t|$$

Keep in mind that we assumed $(\theta_5+\theta_3)\Delta t < 1$ together with $\theta_1, \theta_2, \theta_3, \theta_5 > 0$. This allows $c_4$ to be reduced. With the row and column sums found we restate the expression for spectral density of $\mathbf{A_2}$:

$$\max\{|\lambda| : \lambda \in \sigma(\boldsymbol{A})\} \leq \min[\max(1, r_1, r_4), \quad \max(1, \quad c_4)] \tag{2.38}$$
$$\text{Assuming: } (\theta_2+\theta_3)\Delta t < 1$$
$$(\theta_5+\theta_3)\Delta t < 1$$
$$\theta_1, \theta_2, \theta_3, \theta_5 > 0$$

We can see from (2.38) that we can only ensure the marginal stability by placing the requirements on either row sums $r_1$ and $r_4$ or the column sum $c_4$. That results in the following sets of sufficient conditions for ensuring marginal stability of $\mathbf{A_2}$:

$$\text{Sufficient Condition 1:} \tag{2.39}$$
$$\theta_5 + \theta_3 \geq \theta_2,$$
$$\theta_2 \geq \theta_3,$$
$$\text{And Sufficient Condition 2:} \tag{2.40}$$
$$\theta_5\Delta t \leq 1,$$
$$\text{Assuming:} \tag{2.41}$$
$$(\theta_2+\theta_3)\Delta t < 1,$$
$$(\theta_5+\theta_3)\Delta t < 1,$$
$$\theta_1, \theta_2, \theta_3, \theta_5 > 0.$$

Ideally the sufficient conditions would have been expressed as a relation $\frac{\Delta t}{\Delta x}$. This is because the $\frac{\Delta t}{\Delta x}$ relation is what typically results in instability when discretizing partial differential

equations. We could therefore for the sake of argument decompose the $\theta$ parameters into their components. The general components are however not what we choose to identify in chapter 4, and a $\frac{\Delta t}{\Delta x}$ relation is therefore of no use to us.

For the sake of completeness we include a short appendix on the Consistency and Convergence of the pipe equations which make up $\mathbf{A_2}$. See Appendix B.

# Simulation- and Emulation-Setup 3

In this chapter, we will present an emulation and a simulation setup. Both setups are structured to similarly the Two Tank Topology presented and modeled in the last chapter. This will allow the parameters of the general model to be identified on both setups. The identification is presented in *chapter 4 Model Parameter Identification.*

Though a conclusion on the validity of a control scheme can be reached using a simulation, the behavior observed in emulation is considered to be closer to the behavior of a real network. Generally, the emulation of a system comes with extra challenges compared to the simulation of a system. These challenges can, for example, be handling: sensor imprecision, none ideal actuators, and minor changes in the emulation environment.

The emulation setup is created using the AAU Smart Water Lab. The purpose of emulating the Two Tank Topology is: To show that parameter identification and control can be made, on systems subject to real-world imperfections.

The simulation setup is created according to a real-life sanitation network located in Fredericia - Danmark. We created the simulation setup with the purpose of verifying the validity of the parameter identification on a system on real-world scale. Note that we elected not to implement any of the MPCs on the simulation setup.

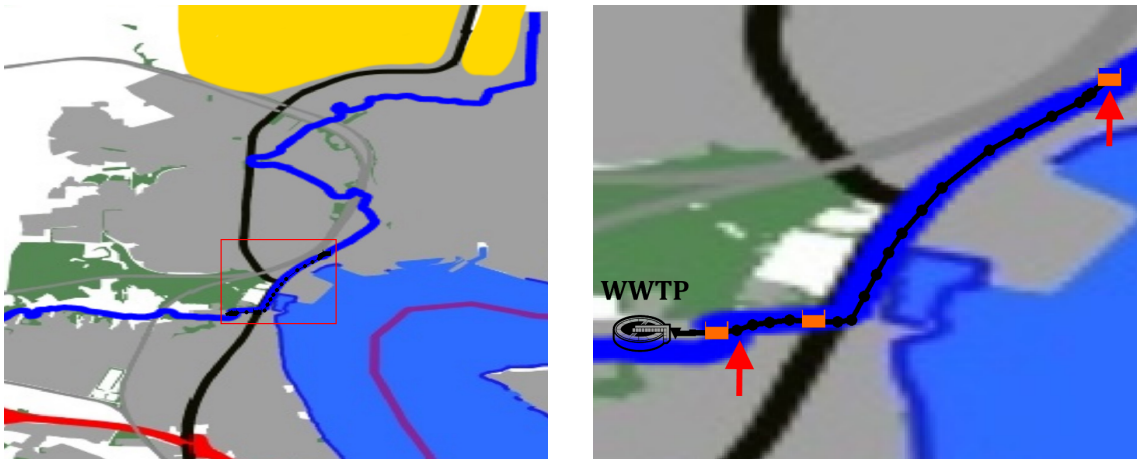
## 3.1 Simulation Setup - Fredericia Network

Federecia is a city located in the region of southern Jutland in Denmark. Information about the cities drainage network has been provided by the local utility company "Frederecia Spildevand og Energi A/S."

A subset of Frederecias main sewer line is considered in this thesis, as it closely resembles the Two Tank Topology. A depiction of the main sewer line (in blue), is shown in *Figure 3.1a* and a zoomed version depicting the section of interest is shown in *Figure 3.1b*

The sewer line subsection consists of three tanks, 19 manholes, 19 pipe sections, and two pumps. The two pumps are used to empty the tanks located at either end of the network subsection. The middle tank acts as a buffer.

The max flow which the pump located at the rightmost tank can provide is $1\frac{m^3}{s}$ and the max inflow the wastewater treatment plant (WWTP) can handle is $1.8\frac{m^3}{s}$. The inflow to the treatment plant dictates the max flow of the leftmost pump. The pipe sections all have a diameter of $1.6m$ except one, which has a diameter of $1.5m$. The slope profile of the




**(a) City and surrounding area, the red box indicates the location of the area shown in *Figure 3.1b***

**(b) Section of interest. The read arrows denote inflow locations, the orange marks denote the tanks and the blue line denotes the main sewer line.**

**Figure 3.1: Fredereica Danmark - Yellow marks industrial area, gray area marks residential area.**

network is depicted in *Figure 3.2*, and except for two pipe sections, all of the pipe sections have a bed slope of around $0.0024\frac{m}{m}$.

In a real scenario, a rain event would cause inflows into the network at many locations along the pipe. We, however, elect to ignore these lateral inflows in the identification part of the project. We elect only to consider inflows a the location indicated by the rightmost red arrow in *Figure 3.1b*. The inflow locations (red arrows) in *Figure 3.1b* correspond to the locations at which the network subsection, in *Figure 3.1b*, is connected to the remaining part of Fredericias main sewer line.

The network subsection is simulated using EPA SWMM and pySWMM. EPA SWMM is a program used for simulating single rain events or long-term hydrologic processes, such as sewage routing. The simulation tool is configured to use a high fidelity model based on the full Saint Venant equations.

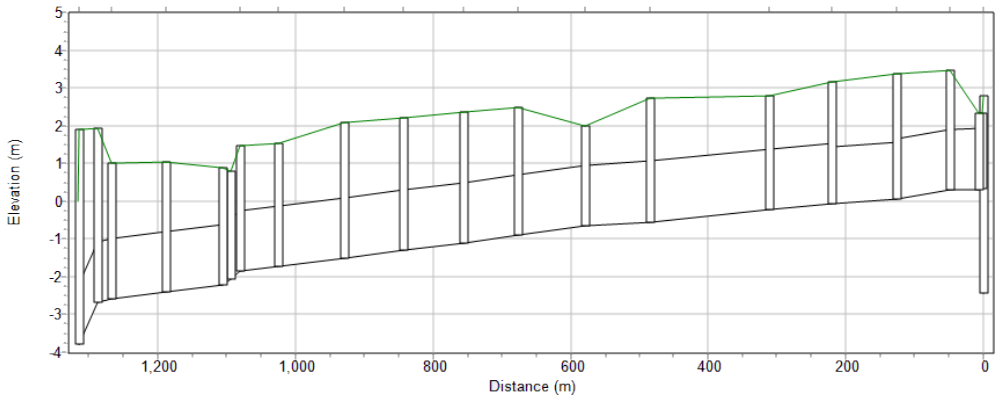



**Figure 3.2: Slope profile of the Fredericia network section.**

Though the high fidelity model is sound, a few key points have to be kept in mind when the model is being used as a substitute for a real system:

- The water level measurements in each pipe section are averages of the water level. This is opposed to a real system, where the water level is measured at a single location in a pipe section.
- The result of the simulations is a numerical approximation of the solution to a partial differential equation. This means that the simulation results are not exact solutions.
- The system simulated is only as good as the information provided. A physical system has small irregularities which can add additional dynamics. These cannot always be captured by a simulation.

Therefore it can be advantageous to emulate a setup as well as simulating it. In the next section, the emulation setup of the Two Tank Topology is presented.

## 3.2 Emulation Setup - AAU Water Lab

The AAU Smart Water Lab is a modular lab located at the Aalborg campus of Aalborg University. The lab is used to emulate different water utility networks, such as sanitation networks and drinking water distribution networks.

The purpose of emulating the Two Tank Topology is to show that parameter identification can be done, not only in simulation but also under real-world conditions. Furthermore, emulating a control setup lends extra validity to results obtained and theoretical analysis completed as part of this thesis.

To emulate the Two Tank Topology, see Figure 2.1, which consists of two tanks, two pumps, and a gravity pipe, we have to use supporting equipment. To provide water, as inflow into tank 1, and to provide lateral inflow, we need a buffer tank. Furthermore, to catch water at the outlet simulating the Waste Water Treatment Plant, we need a buffer tank. The entire network, with buffer tanks and additional pumps, is shown in *Figure 3.3*.

The AAU Smart Water Lab consists of a Central Control Unit (CCU) and a number of Local Units (LU). Each LU serves to emulate a specific element in a real-world water distribution network. When combined, multiple LUs can be used to emulate a network section.

To create the setup shown in *Figure 3.3* two types of LUs have to be employed. These are the Pumping Station and the Consumer Unit. Two Pumping Units are used to create the Two Tank Topology, and two Consumer Units are used as buffer/auxiliary tanks. In appendix Appendix C an overview of the LUs full physical structure can be found.

### 3.2.1 Water lab control

The AAU Smart Water Lab is constructed as a centralized control scheme. A Central Control Unit (CCU) transmits pump and valve references to the individual LUs and the LUs return flow and level measurements. The LUs are controlled locally by a Raspberry PI running a "virtual" codesys PLC on top of Ubuntu. While virtual PLC is set up the same for all LUs, each type of LU has a different number of ADCs and DACs. The number

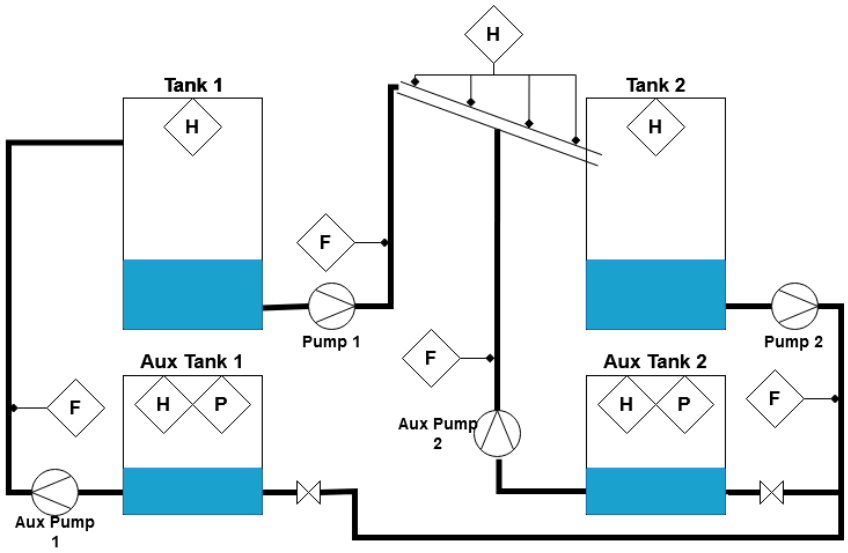



**Figure 3.3: Schematic of the two tank lab setup with sensors indicated. Pressure sensor (P), Level sensor (H) and Flow sensor (F).**

of ADCs and DACs is determined by the function of the LU. A description of the AAU Smart Water Lab Software/Hardware interconnections are shown in *Figure 3.4*.

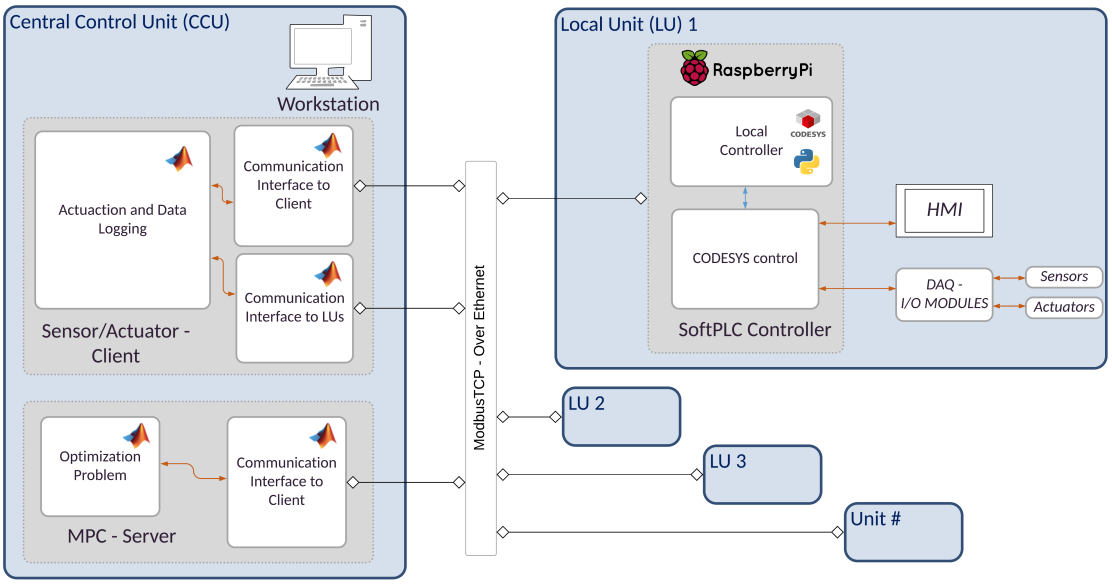



**Figure 3.4: AAU Smart Water Lab Software/Hardware interconnections**

Figure 3.4 shows the CCU being "split" into two different entities, a server, and a client entity. The client is responsible for updating the pump and control values on the LUs as well as collecting data. The server's responsibility is to calculate the control action. Having the CCU split into two entities allows the control action and data sampling to happen asynchronously. This will prove an advantage later. More on the specific sampling and control timing is presented in *subsection 7.2.6 The Control System Software Setup*.

### 3.2.2 Pump control

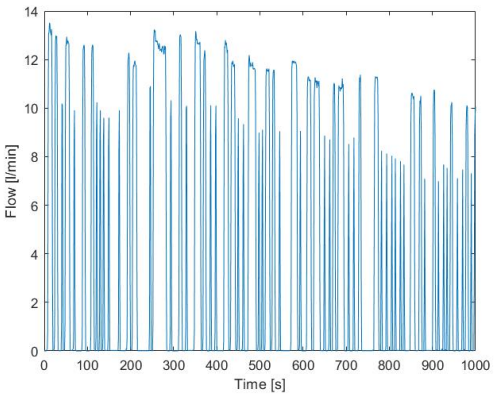
The CCU controls the pumps on a specific LU through communication with the LUs PLC. A value between 0 and 100 can be transmitted to a LU corresponding to a percentage of a pump's max output.

However, the flow at the max output depends on a lot of factors, such as the resistance of the pipes and the pressure on the feeding side of the pump. Therefore there is no static relationship between the "output max percentage" and flow. An example of how flow changes independent of pump output percentage setting is shown in Figure 3.5a. Figure 3.5a specifically shows the flow produced by a pump being actuated using an on/off controller, where the pump is either running at 100% or 0%. In spite of the pump running at 100% it can be seen that the flow produced decreases with time.

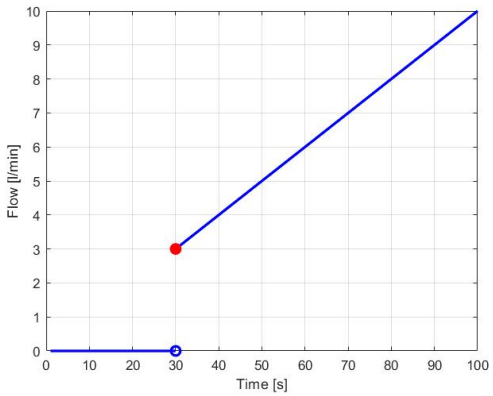
Not only is the flow/percentage relation changing, but there is also a discontinuity in the relation. This discontinuity is shown in Figure 3.5b. It can be seen that at low pump settings the pump does not produce any flow. The exact point at which the relation is discontinues is however unknown, as this point like the flow/percentage relation is dependant on a number of factors.

To try and compensate for the changing dynamics, a PI controller is implemented for each pump in the system. The PI controller is implemented at the top of the control hierarchy on the CCU. Though implementing the controllers on the CCU adds extra dead time to the control loop, compared to a similar implementation directly on each LU, it allows for faster implementation and easier reconfiguration. The PI controllers are tuned manually since their design is not the main focus of this project.

Baseline parameters for the PI controllers were established by tuning the controllers to a first-order model with dead time, fitted to the systems step response. The baseline P and I parameters were then used in the real system's controllers, where they were tweaked based on the observed behavior when operating around the discontinuity.




**(a) A pump is actuated at using on off control. The two settings are 100% and 0%.**




**(b) Graph illustrating the discontinuity in the flow/percentage relation of the pumps used to emulate the Two Tank Topology.**

**Figure 3.5: Dynamics of pumps in the AAU Smart Water Lab**

The P and I parameters were found, along with approximate values at which the discontinuity is located. The resulting values are shown in table 3.1 and table 3.2.

| Pump Name | Starting Point(%) | Flow at Starting Point (L/min) | Flow at 100% (L/min) |
|---|---|---|---|
| Pump 1 | 45 | 3.4 | 8.3 |
| Pump 2 | 40 | 5.4 | 19.5 |
| Aux pump 1 | 45 | 4 | 9.5 |
| Aux pump 2 | 55 | 3.5 | 12.5 |

**Table 3.1: Minimum and maximum flow values**

| Pump Name | Proportional Coefficient | Intergral Coefficient |
|---|---|---|
| Pump 1 | 0.1 | 0.3 |
| Pump 2 | 0.145 | 0.61 |
| Aux pump 1 | 0.15 | 0.45 |
| Aux pump 2 | 0.1 | 0.4 |

**Table 3.2: P and I coefficient used in pump controllers**

# Model Parameter Identification 4

In *section 2.2 General Model* we developed a parametrized model describing the Two Tank Topology. The final state space description is shown in (2.27). For the particular implementation of the Two Tank Topology, parameters for the system have to be identified. Note that the tank parameters $\phi_1$ and $\phi_2$ are the only parameters in the system that have a direct physical meaning. This is because they are defined as the inverse of a tank's area $\phi_i \triangleq \frac{1}{A_i}$. We assume that this parameter is known and need not be estimated.

The gravity pipe parameters $\theta$ are highly correlated, and calculating them reliably from system observations is hard. Therefore the parameters of the gravity pipe will be identified using level and flow data. This approach will be tested and validated with a model in the high-fidelity simulation software *EPA-SWMM* as well as using the experimental lab setup described in the previous chapter. This chapter aims to formulate parameters estimation for the gravity pipe such that our approximated state space model matches the behavior captured in an actual system. In the following section, we describe the gravity pipe parameters identification, and later we present the results along with the validation of the models.

## 4.1 Gravity Pipe Model Identification

We consider a subsystem of the Two Tank Topology shown in (2.27) consisting of the gravity pipe and the second tank for the purpose of identifying the pipe parameters. This part of the network is independent of the first tank since the boundary condition on the pipe inflow is defined by pump 1, see *subsection 2.1.3 Pipe boundary conditions*. On the other hand, the pipe outflow is dependant on the tank state in the submerged flow case. In the free-flow case, it is beneficial to know the pipe outflow, and it can be estimated with the tank equation (2.15) if a flow measurement is not directly available.

The parameter identification is defined as a constrained linear least square program. Let $\boldsymbol{\theta} \triangleq \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5\}$ denote the parameters of our linear pipe model. The pipe inflow $Q_{in}$ and the tank outflow $Q_{out}$, in (2.27) represented as $Q_{FP1}$ and $Q_{FP2}$ respectively, are given for each time step. Let $N_s$ denote the number of level sensors along the gravity pipe. We define $N_x$ ($N_x \geq N_s$) to be the number of pipe sections after discretization. The number of pipe sections after discretization is equal to the number of pipe states in the general model. We assume that the ideal response to $Q_{in}$ can be obtained which yields the water level measurement values $h_j$, $j \in \{1, 2, ..., N_s\}$ as well as the tank water level $h_{T2}$ value at

times $t_i$, $i \in \{0, 1, ..., N_t\}$. $N_t$ defines the time frame.

It is important to note that we consider two boundary conditions, which yield two different model descriptions for the system. It is required that a single measurement set describes only one of the two scenarios since a switch model is out of the scope of this thesis. Consequently, this means that we have to develop two separate parameter identification procedures, one for each case. Then when we are faced with one of the two boundary cases, we use the appropriate identification method. First the generalized parameter identification problem is formulated in (4.1), (4.2) and (4.3). Then in the following text, the two different dynamics representations for (4.2) are elaborated on.

The estimated parameters $\boldsymbol{\theta}$ and the initial states $\boldsymbol{h}^{t_0}$, $h_{T2}^{t_0}$ are found by solving the optimization problem:

$$\begin{pmatrix} \boldsymbol{\theta}^* \\ \boldsymbol{h}^{t_0} \\ h_{T2}^{t_0} \end{pmatrix} = \arg \min_{\boldsymbol{\theta}, h^{t_0}, h_{T2}^{t_0}} \underbrace{\sum_{i=0}^{N_t} \left[ \left( h_{\mathrm{T2}}^{t_i} - \hat{h}_{\mathrm{T2}}^{t_i} \right)^2 + \sum_{j=1}^{N_s} \left( h_j^{t_i} - \hat{h}_j^{t_i} \right)^2 \right]}_{\mathcal{J}} \tag{4.1}$$

s.t.

$$\begin{bmatrix} \hat{\boldsymbol{h}}^{t_i+1} \\ \hat{h}_{T2}^{t_i+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}(\boldsymbol{\theta}) \\ \boldsymbol{H}(\boldsymbol{\theta}) \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{h}}^{t_i} \\ \hat{h}_{T2}^{t_i} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}(\boldsymbol{\theta}) & \mathbf{0}_{N_x \times 1} \\ 0 & -\phi_2 \Delta t \end{bmatrix} \begin{bmatrix} Q_{in}^{t_i} \\ Q_{out}^{t_i} \end{bmatrix} + \boldsymbol{\Delta}(\boldsymbol{\theta}) \tag{4.2}$$

$$\theta_1, \quad \theta_2, \quad \theta_3, \quad \theta_5 > 0, \tag{4.3}$$

where $\hat{\boldsymbol{h}} \in \mathbb{R}^{N_x}$ is the vector of states, $Q_{in}$ and $Q_{out}$ are the input signals. The system dynamics are represented by the state space equation (4.2) where $\boldsymbol{A}(\boldsymbol{\theta}) \in \{\boldsymbol{A_1}(\boldsymbol{\theta}), \boldsymbol{A_2}(\boldsymbol{\theta})\}$, $\boldsymbol{A_1}, \boldsymbol{A_2} \in \mathbb{R}^{N_x \times N_x+1}$ and $\boldsymbol{B}(\boldsymbol{\theta}) \in \mathbb{R}^{N_x}$ are on the form shown in (4.4). In the above problem, the number of pipe sections $N_x$ and time frame $N_t$ are fixed and given. The system matrix $\boldsymbol{A}(\boldsymbol{\theta})$ can have two different forms depending on the boundary condition for the last pipe section, see subsection 2.1.3. The matrix with index 1 describes the free-flow case while the matrix with index 2 describes the submerged flow case.

$$\boldsymbol{A_1} = \begin{bmatrix} 1-\theta_2\Delta t & \theta_3\Delta t & 0 & 0 & & \cdots & 0 \\ \theta_2\Delta t & 1-(\theta_2+\theta_3)\Delta t & \theta_3\Delta t & 0 & & \cdots & 0 \\ \vdots & & & \ddots & & & \\ 0 & & \cdots & 0 & \theta_2\Delta t & 1-(\theta_5+\theta_3)\Delta t & 0 \end{bmatrix}, \tag{4.4a}$$

$$\boldsymbol{A_2} = \begin{bmatrix} 1-\theta_2\Delta t & \theta_3\Delta t & 0 & 0 & & \cdots & 0 \\ \theta_2\Delta t & 1-(\theta_2+\theta_3)\Delta t & \theta_3\Delta t & 0 & & \cdots & 0 \\ \vdots & & & \ddots & & & \\ 0 & & \cdots & 0 & \theta_2\Delta t & 1-(\theta_5+\theta_3)\Delta t & \theta_5\Delta t \end{bmatrix}, \tag{4.4b}$$

$$\boldsymbol{B} = \begin{bmatrix} \theta_1\Delta t \\ \mathbf{0}_{N_x-1}^T \end{bmatrix}. \tag{4.4c}$$

The vector $\boldsymbol{H}(\boldsymbol{\theta}) \in \{\boldsymbol{H_1}(\boldsymbol{\theta}), \boldsymbol{H_2}(\boldsymbol{\theta})\}$, $\boldsymbol{H_1}, \boldsymbol{H_1} \in \mathbb{R}^{N_x+1}$ in equation (4.2) is the mapping of the system states to the tank water level $\hat{h}_{T2}$. Like the $\boldsymbol{A}(\boldsymbol{\theta})$ matrix it can have

two expressions from (4.5), with 1 representing the free-flow case and 2 representing the submerged flow case.

$$\boldsymbol{H_1} = \begin{bmatrix} \mathbf{0}_{(1\times Nx-1)} & \phi_2 \frac{\theta_5}{\theta_1}\Delta t & 1 \end{bmatrix}, \tag{4.5a}$$

$$\boldsymbol{H_2} = \begin{bmatrix} \mathbf{0}_{(1\times Nx-1)} & \phi_2 \frac{\theta_5}{\theta_1}\Delta t & 1 - \phi_2 \frac{\theta_5}{\theta_1}\Delta t \end{bmatrix}. \tag{4.5b}$$

The matrix $\boldsymbol{\Delta}(\boldsymbol{\theta}) \in \{\boldsymbol{\Delta_1}(\boldsymbol{\theta}), \boldsymbol{\Delta_2}(\boldsymbol{\theta})\}$, where $\boldsymbol{\Delta_1}, \boldsymbol{\Delta_2} \in \mathbb{R}^{N_x}$, represents a constant offset in transition of states. Similar to the $\boldsymbol{A}(\boldsymbol{\theta})$ and $\boldsymbol{H}(\boldsymbol{\theta})$ matrices it has a different expression depending on the boundary condition, shown in (4.6). The indices are assigned the same way as in (4.4) and (4.5).

$$\boldsymbol{\Delta_1} = \begin{bmatrix} -\theta_4\Delta t & \mathbf{0}_{N_x-2} & \theta_4\Delta t & 0 \end{bmatrix}^T, \tag{4.6a}$$

$$\boldsymbol{\Delta_2} = \begin{bmatrix} -\theta_4\Delta t & \mathbf{0}_{N_x-2} & (\theta_4 - \theta_5 T_o)\Delta t & \phi_2 \frac{\theta_5}{\theta_1} T_o \Delta t \end{bmatrix}^T, \tag{4.6b}$$

where $T_o$ represents the level at which the pipe enters the tank. That level has to be subtracted from total tank water level if we want to describe the backflow from the tank, see subsection 2.1.3 and specifically *Figure 2.3*. Notice also that when the pipe enters at the bottom of the tank, $T_o = 0$, the two matrices $\boldsymbol{\Delta_1}, \boldsymbol{\Delta_2}$ are equivalent. The constrains in (4.3) restrict the parameters $\theta_1, \theta_2, \theta_3, \theta_5$ to be positive. This is imposed based on the physical insight of the parameters and also in order for our system stability considerations to hold.

The stated estimation problem can be computed using the *Gradient-based search algorithm* [30]. To estimate the parameters and the initial states we use the *nlgreyest* toolbox in MATLAB. It implements a set of different numerical, iterative search methods for parameter estimation. The *Adaptive subspace Gauss-Newton search* method was used in the following estimations.

## 4.2 System Identification Results

This section presents the results of applying our method of parameter identification on three different cases. In order to have a quantitative measure of the goodness of the identification, we first perform a benchmark experiment. For this purpose, we created a simple, ideal case of the Two Tank Topology using the simulation software EPA-SWMM. It solves the full Saint-Venant equations to simulate the flow in a gravity pipe. The goal is to identify our simplified linear model with the high fidelity (HI-FI) simulation. The test model is depict in *Figure 4.1*.

In this test we consider the inflow and the outflow pumps (*FP1*, *FP2*), four pipe sections (*P1-P4*) and the downstream tank(*T2*). It is important to note that at the pipe outlet, the free-flow condition is created; see *subsection 2.1.3 Pipe boundary conditions*. Meaning the outflow borders above the *T2* maximum level. That way, in normal operation, as well as in the case of the tank overflow, the water is never flowing back into the pipe. The parameters of the specific elements are shown in the table 4.1. The rest of the network is used to make sure that there is enough water in the system to run a simulation and can be chosen arbitrarily. The inlet flow $Q_1$ enters the sewer at *N1* and we observe the outflow

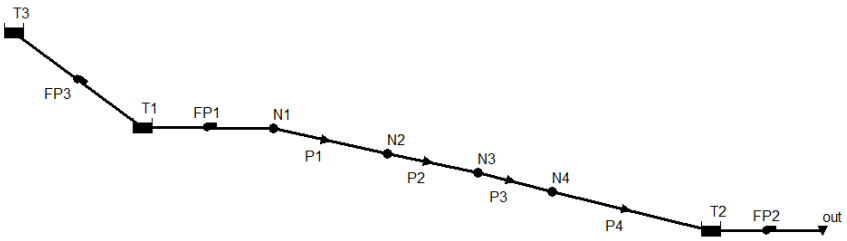



**Figure 4.1: Test network in EPA-SWMM. The *T1-3* are tanks, *FP1-3* are the pumps. *P1-4* are the discrete pipe sections. *N1-4* are junctions where the water level is measured.**

$Q_2$ from the downstream tank *Figure 4.2b*. The outflow $Q_2$ consists of a constant flow provided by *FP2* that removes water from *T2* and the additional flow which captures the tank overflow (in particular around times $t = 110$ min, $t = 200$ min and 260 min). It can be seen that in cases of tank overflow, the $Q_2$ flow in *Figure 4.2b* increases instantaneous. Additionally we capture the water levels at each node (*N1* to *N4*) and the water level in *T2*.

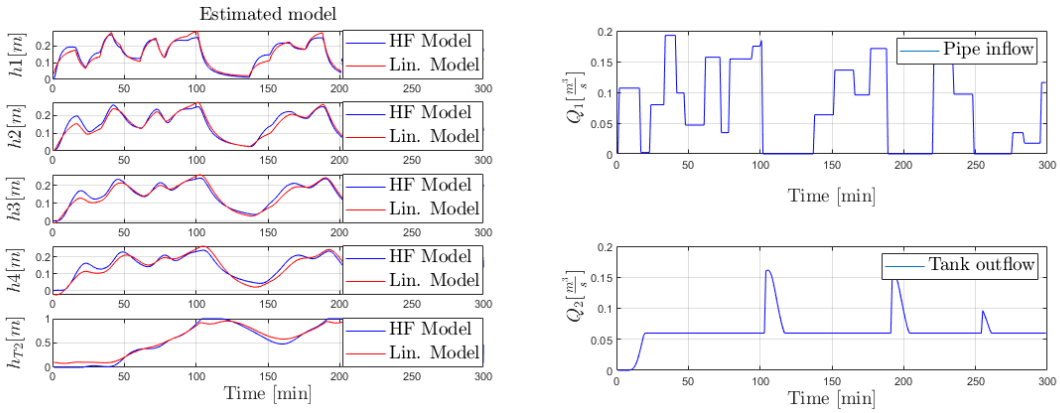
Since the goal is to use one model in all scenarios for inlet flow, the system is excited with a randomized sequence of flows shown as $Q1$ in *Figure 4.2b*. The intention is to capture as many relevant dynamics in the data as possible.

The validation of the identified model is shown in *Figure 4.2a*. The five hours prediction with identified parameters is compared with the captured measurements. It can be seen that we obtained a sufficiently good approximation.

*Figure 4.3* shows the residual analysis of the identified model with respect to the identification data set. *Figure 4.3a* show the residuals for each pipe section and the last tank while *Figure 4.3a* shows the distribution of the residuals. The residuals are scaled to the maximum depth value and calculated as $r_i = \frac{h_i - \hat{h}_i}{\max(h_i)} 100\,[\%]$. This normalization is convenient if we want to compare it with the results of further experiments. Once we successfully solved the parameter estimation for the simple simulation we proceed to apply the same procedure to the Fredericia network subsection described in *section 3.1 Simulation Setup - Fredericia Network*.

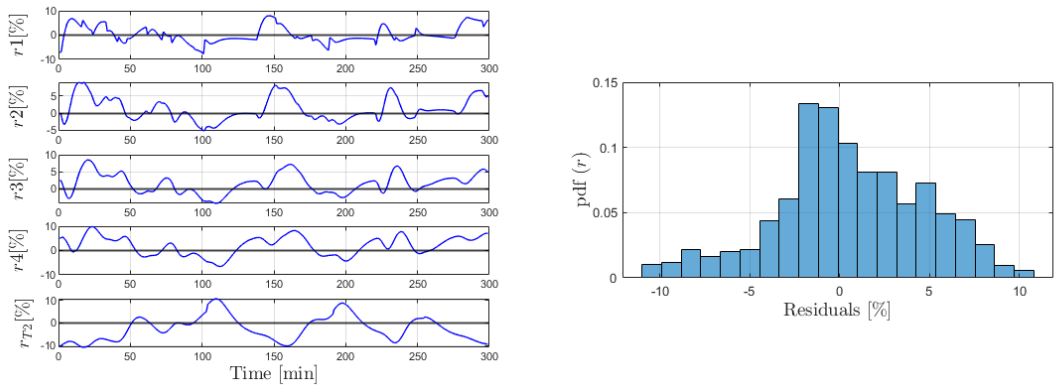
| Element/Property | Inital depth | Max depth | Invert. Elev. | Coefficient |
|---|---|---|---|---|
| Tank 1 (T1) | 2.5 | 3 | 0 | 150 |
| Tank 2 (T2) | 0 | 1 | 0 | 200 |
| | Length | Max depth | Inlet offset | Slope |
| Pipe 1-4 (P1-P4) | 400 | 0.5 | 0 | $\frac{1}{200}$ |

**Table 4.1: Element settings in EPA SWMM.**




**(a) Estimated model prediction.** **(b) Input flows.**

**Figure 4.2: Model validation.**




**(a) Residuals.** **(b) Residuals distribution.**

**Figure 4.3: Residual analysis.**

### 4.2.1 Results of the Fredericia network simulation identification

As in the previous experiment, we have a similar configuration of the network with a downstream tank and four level measurements that are approximately equally spaced along the main sewer line. The placement of the sensors is shown in *Figure 4.4*. Although we in this part of the network have nineteen measure points available (19 manholes, *section 3.1 Simulation Setup - Fredericia Network*), we choose to go with four measurements in order to match the conditions we are able to emulate in the AAU Smart Waterlab. We will consider only four pipe sections (states in our model) for the initial validation, which we measure. The number of states affects the quality of the identification. Therefore we will perform the identification multiple times on the same data but with a different number of states, shown in *section 4.2.1 Augmented model*. Since we have the data for the dimensions of the network, we can fix the tank parameter $\phi_2 = \frac{1}{A}$, where $A = 10.011$ [m$^2$] and only estimate the parameters of the pipe $\boldsymbol{\theta}$.

Unlike in the previous experiment, we observe a different boundary condition at the end of the pipe. Namely, the pipe enters the tank at its bottom (with an offset of 0 [m]), as shown in *Figure 4.5a*.

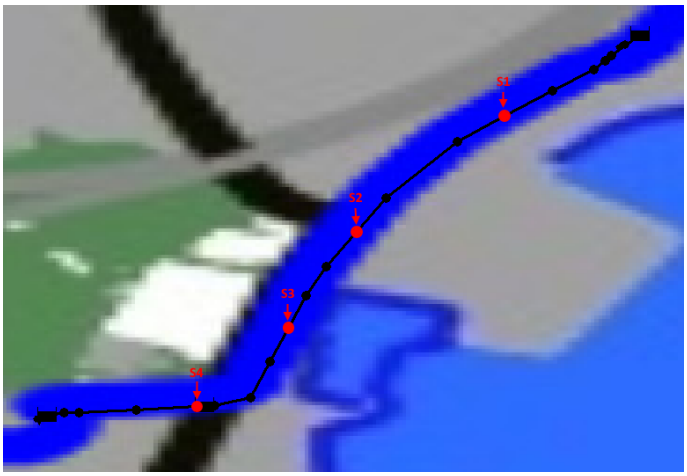
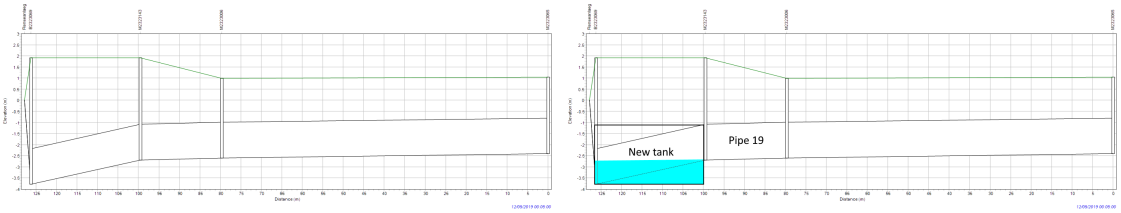



**Figure 4.4: Sensor placement on the Fredericia network.**

The last pipe also enters the tank at a steeper angle compared to the rest of the network, see *Figure 3.2*, and we made the assumption when deriving the model that the slope of the pipe was approximately the same along its entire length. Since the outflow tank is not a retention tank, it has a small volume compared to the volume of the pipe sections in the network. The volume of the tank in the operating range 0 - 2.7 $[m]$ is $\sim$ 27 $m^3$ and the volume of the last pipe section before the tank is $\sim$ 54 $m^3$). This volume difference makes the tank dynamics a lot faster than the dynamics of gravity pipe, which amplifies a highly nonlinear behavior at the boundary. All these observations account for a large discrepancy between our estimated model and the HI-FI simulation.

We propose a solution: We consider the last pipe in the network as extended storage together with the tank, making a new tank. This consideration makes our newly created tank a nonlinear system since the area is now a nonlinear function of the water level. However, we elect not to model a nonlinear tank, but we instead approximate the tank as having a linear description with the same capacity. Therefore we added the last pipe's volume to the tank and scaled the area of the tank from 10.011 [$m^2$] to 29.94 [$m^2$]. That way, we have the same capacity in the operating range ( 0 - 2.7 $[m]$), but it is approximated with a linear function. The dynamics of the new tank are also slower now since the area is increased almost three times. However, since we want the optimal match of the dynamics, we do not fix the parameter $\phi_2$ in the estimation.




**(a) Before approximating the last pipe.** **(b) After approximating the last pipe.**

**Figure 4.5: Approximating the end-part of the network structure.**

Now *Pipe 18* from *Figure 4.5b* is the last pipe section in the network. It is joined with the new tank with an inflow offset of 1.1 m. When the water level in the tank is under this level, the free flow dynamics apply. But the new tank still has a much smaller volume than the volume of the sewer line. For a comparison, the new tank volume is 80 $m^3$ and the volume of *Pipe 18* is 40 $m^3$. This means that it is likely that the tank will fill above 1.1 m.

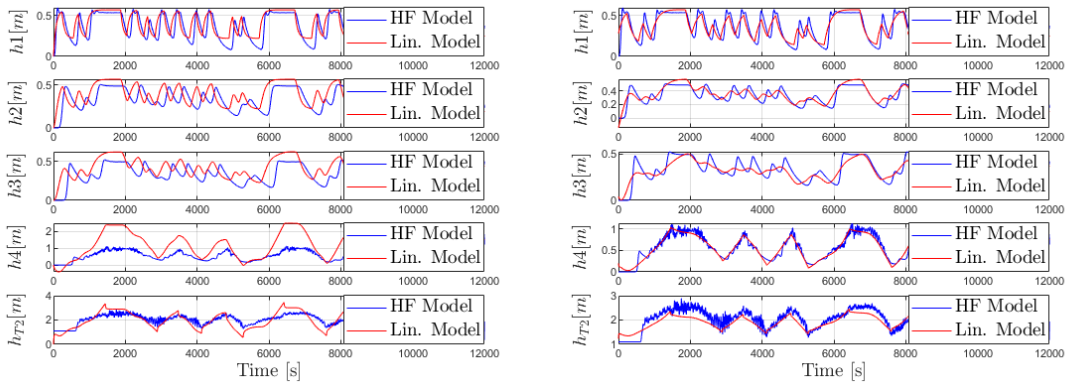
Consequently, the pipe is constantly filling backward as the tank is filling. However, our model with the backflow boundary condition should be able to capture these dynamics. The limitation to the model is that it works as long as the pipe is not entirely filled nor completely empty. We constrain ourselves to only consider the backflow condition at the boundary. Therefore we define an operating range for the tank to be 1.1 to 2.7 [m] during the simulation.

In summary, the goal of the identification in this section is to try to identify the backflow effect from the tank into the pipe together with the rest of the pipe dynamics described in *subsection 2.1.1 Gravity pipe*. The model validation is shown in *Figure 4.6b*. The five-hour prediction with estimated parameters is plotted against the validation data for comparison. Some unestimated nonlinearities in the last section and the tank are obvious, but the model captures the general trend. The tank linear approximation with constant area reflects on the tank dynamics, which can be seen in *Figure 4.6b*, bottom plot. We also calculate the residuals for a comparison. Residuals show the unmodeled dynamics in the system.

If we closely observe the model response in *Figure 4.6b*, we can conclude that the delays between sections are not well captured. Our model propagates the flow faster through the network, which results in smoothening of later states. In particular, states number two and three are smoothened. The reason for that is that our model, by having only four states, inherently does not capture the delay of more than one time step between each measurement. It can be seen in *Figure 4.6a* that with initial parameters, which are calculated based on the physical properties of the system, the dynamics are well predicted but without delay. Therefore in the following subsection, we will augment our model with unmeasured states to account for the delay. It is important to note that in the simulation, the pipe level is averaged along each section, which results in less delay between each sensor. In a real-world system, this delay to the sensor is much more pronounced.

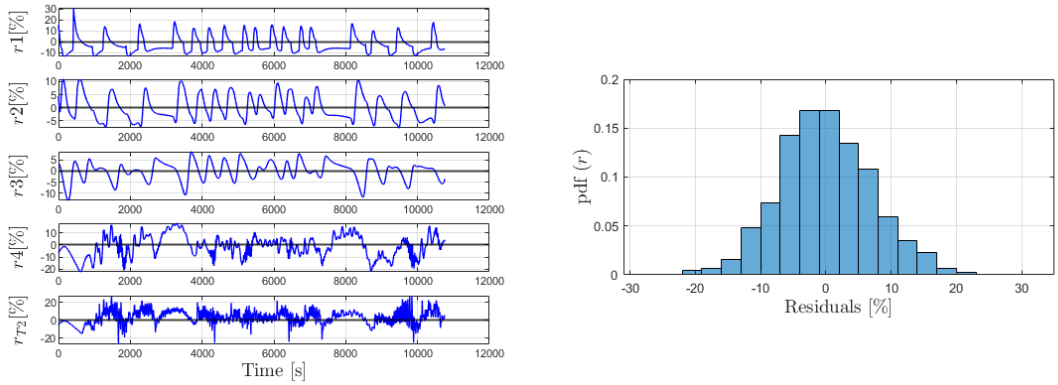
### Augmented model

In this section, we analyze improvement in the linear model's fit by adding more states. Note that the number of sensors and their placement in the network will still stay unchanged. The accuracy of the model on the same data, with a different number of states, is shown in *Figure 4.8a*. By adding more states, the model prediction accuracy increases since those augmented states now account for the delay between the sensors. However, after a certain threshold, for example, $N_x = 16$, the model improvement is not significant compared to its computational demand. The apparent reason for that is our choice of




**(a) Initial model.** **(b) After parameter estimation.**

**Figure 4.6: Fredericia model estimation.**




**(a) Residuals.** **(b) After removing the last pipe.**

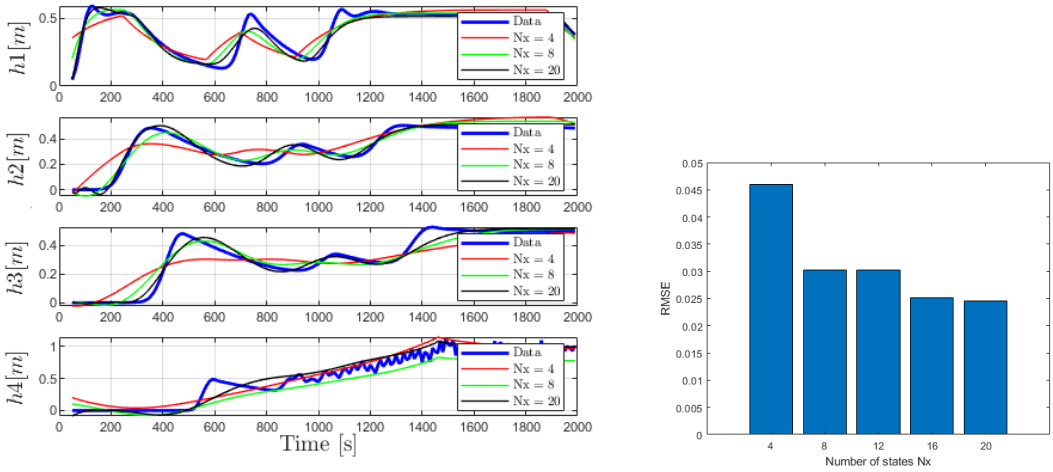
**Figure 4.7: Changing the end-part network structure**

the sampling time, $\Delta t = 10$ [s]. This sampling time is chosen in order to capture the fast dynamics of the tank and the last pipe, which are coupled. Comparing the sampling time to the gravity pipe routing delay, one augmented state is not significant enough to capture the actual delay. We calculated the full pipes routing delay to be approximately 15 minutes, which is equal to 3 minutes between each sensor.

The improvement in model accuracy is measured using Root Mean Squared Errors (RMSEs) and shown in *Figure 4.8b*. The RMSEs are calculated as:

$$\text{RMSE} = \frac{1}{h_{\max}} \sqrt{\frac{\sum_{n=1}^{N_s} \sum_{t=0}^{N_t} (\hat{h}_n^t - h_n^t)^2}{N_s \cdot N_t}}, \tag{4.7}$$

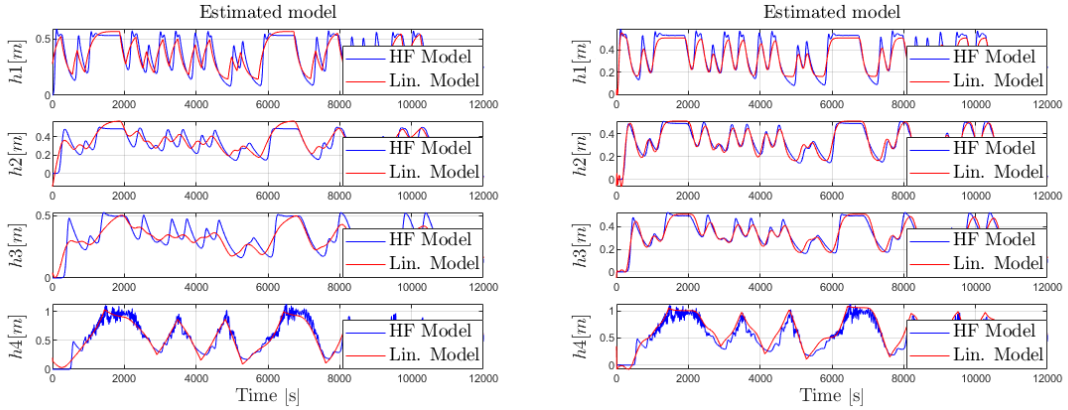
where $N_s$ is the number of sensors, $N_t$ is time frame. The RMSEs are scaled with the maximum possible water level in the pipe. In the calculation shown in *Figure 4.8b* the fourth state is not included. The numerical instability caused by the simulation can make RMSE particularly large for a certain set of measurements. The other reason for not including the tank state is that since it is the boundary state, it is not affected by the delay because it has a different expression from former states. In other words, it has an




**(a) Model accuracy for different $N_x$.** **(b) RMSE for different $N_x$.**

**Figure 4.8: Model accuracy improvement by additional states for the Fredericia experiment.**

extra free parameter (parameter $\theta_5$) which defines its dynamics. The comparison of the model predictions with four and 16 states are shown against each other in *Figure 4.9*. The improvement of the prediction is obvious, especially in states 2 and 3, but it comes at the price of having a high number of states.




**(a) 3 [h] prediction with $N_x = 4$.** **(b) 3 [h] prediction with $N_x = 16$.**

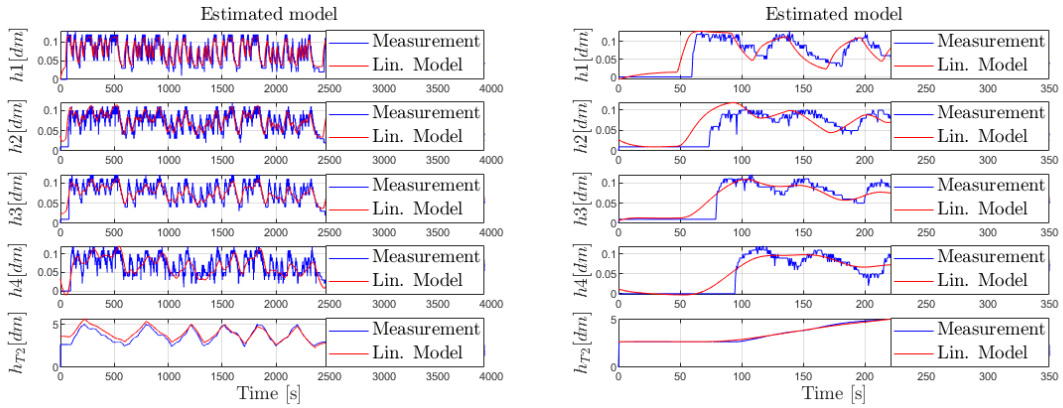
**Figure 4.9: Model prediction with different number of states.**

### 4.2.2 Results of the AAU Smart Water Lab emulation identification

In this subsection, we present the gravity pipe parameter identification results in the AAU Smart Water Lab. An overview of the lab setup is shown in *section 3.2 Emulation Setup - AAU Water Lab*, *Figure 3.3*. For the parameters identification, we use the two tanks and a gravity pipe. Each tank is equipped with inflow and outflow pumps. We collect the flow measurements from the pumps and the level measurements from the tanks. Along the gravity pipe, we have four level sensors, which are used to collect level data.

The mentioned measurements which are used for identification are collected in the

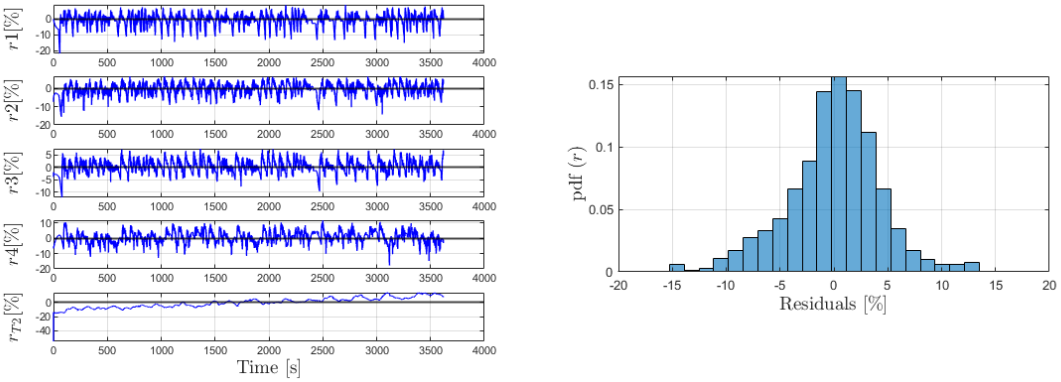
experiment with sampling time $\Delta t = 0.5$ [s]. The simple control for the experiment is built in MATLAB Simulink and executed in real-time. The results of the parameter identification on 1 hour data is shown in *Figure 4.10a* and the first 300$s$ segment is depicted in *Figure 4.10b*. The model with four states is used in the estimation ($N_x = 4$).




**(a) 1 [h] prediction.** **(b) 300 [s] prediction.**

**Figure 4.10: Model validation for the waterlab experiment.**

On the *Figure 4.11a* the residuals of the model are plotted, and in *Figure 4.11b* the distribution of the residuals is shown. Unlike the previous experiments in simulation, the residuals are composed both from a model prediction uncertainty and measurement uncertainty. In the lab experiment, we have the same challenge with the delay prediction

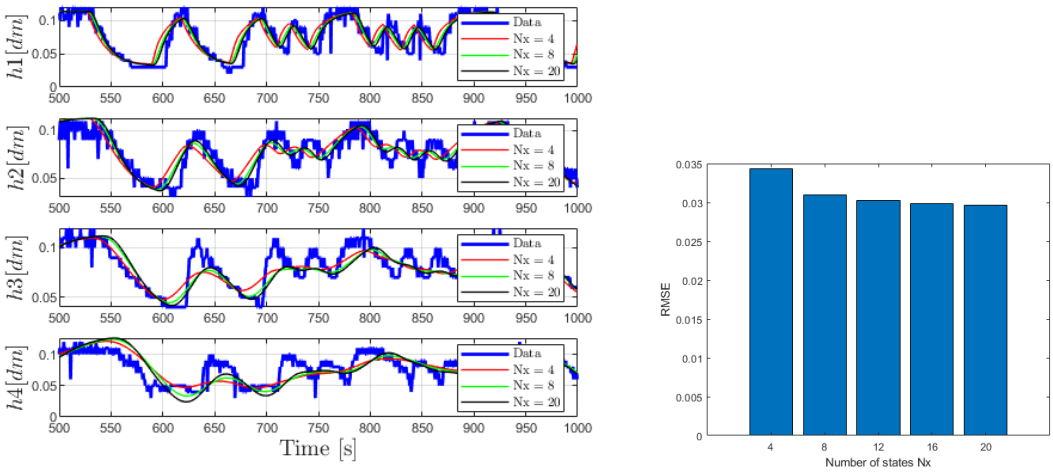



**(a) Model residuals.** **(b) Distribution of residuals.**

**Figure 4.11: Waterlab residual analysis.**

as in the Fredericia network described in the previous section. The model, having only four states, can not predict the delay between sensors. Therefore we have to augment the model with additional states between each sensor. The improvement of the model relative to the number of states is shown in *Figure 4.12a*. The improvement is not that significant, especially in the later states. The RMSE comparison is shown in *Figure 4.12b*. The improvements are even less significant when compared to the Fredericia model. Note that adding augmented states cannot completely compensate for the delay, even with a high number of augmented states. We find that the delay is inversely proportionate to

the average flow intensity in the pipe. The higher the flow, the smaller the delay and vise versa. That would mean that with different flow conditions, the number of needed states would be changing, making the model nonlinear and hard to use. The other possible solution would be to model the delay separately into $N_x$ additional states and keep the number of level states minimal. However, this approach is not implemented in this work, but it is considered a possible solution if better predictions are required.




**(a) Model accuracy for different $N_x$.** **(b) RMSE for different $N_x$.**

**Figure 4.12: Model accuracy improvement by additional states for the lab experiment.**

## 4.3 Summary

In this chapter, we formulated a data-driven parameter identification method for the linear gravity pipe model. Although it is a data-driven method, the estimation was based on an actual physical model. The big advantage of using a physical model instead of a heuristic model is that the intuition in the model behavior is preserved. The models used are relatively simple and linear, making them easy to use in a control application. The identification was tested on data, from both a simple simulation setup and a more complex real network simulation setup, with more complex dynamics. We also tested it on an emulated network in the AAU Smart Water Lab. Both implementations of the identification method, in simulation and emulation, showed that presented parameter identification is feasible. Though only simple linear models were used, the identification process produced good results. The models could be improved in regard to predictions of transport delay. Capturing the transport delay has been identified as the most significant limitation of these linear models. Using residual analysis, the model uncertainties were provided, which will be useful in a Kalman Filter implementation shown in *chapter 5 State estimator*.

The biggest advantage of this parameter identification method is that it can easily be implemented for larger networks since it is based on the data gathered from the network. The model is also based on the water level measurement in the gravity pipe, which is more readily available and more accessible than the equivalent flow measurements. We elected

to use long free-running model predictions since we plan to use the linear model of the AAU Smart Water Lab in Model Predictive Control.

During the remainder of the thesis, we extended the presented parameter identification framework. A discussion of the changes can be found in *section 8.1 System Identification*.

# State estimator 5

As this thesis aims to explore the viability of Model Predictive control in real-world sanitation networks, it is worth considering the possibility of having states that are not directly measurable. Though a sanitation system would ideally have many high-quality sensors supporting control efforts, this is rarely the case in the real-world, as money becomes a factor. Therefore, a state estimator becomes a requirement, and we elect to present a possible approach to creating one.

The two most common estimators used in linear system theory are the Luenberger Observer and the Kalman filter. These two estimators are similar in the way they estimate unknown states. There is, however, one key difference in the assumptions used to create them: The Kalman filter assumes measurements and models to be imperfect and, therefore, compensates for this. The Luenberger Observer, on the other hand, assumes an ideal model and ideal measurements.

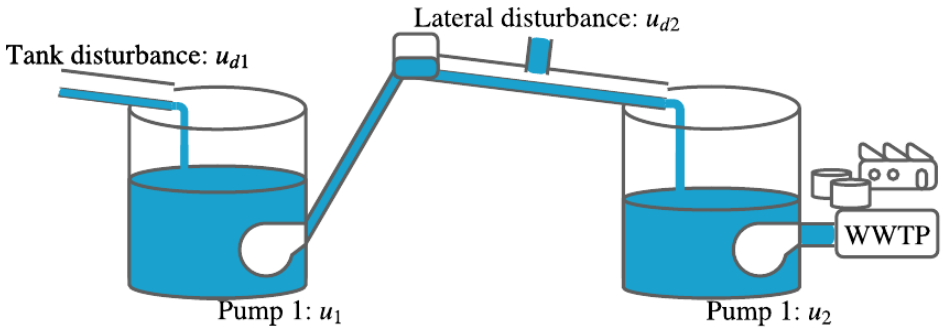



**Figure 5.1: Illustration of the Two Tank Topology, with the disturbance inflows.**

In practical applications, i.e., in non-simulation cases, the measurements obtained from a system will contain measurement noise. Even in the AAU Smart Water Lab, which is a controlled environment, sensor noise is a significant fact that should be taken into account. It is also worth noting that most systems cannot be described perfectly by linear system theory. Therefore it is safe to assume that any linear model will introduce model noise.

In real-world sanitation systems, tank levels would almost always be measured. Furthermore, the measurement noise present on these measurements is likely to be insignificant. Therefore we choose to focus on creating a Kalman filter for the pipe sections exclusively.

However, before we begin, we make an addition to the description of the four pipe state, state space model presented in *section 2.2 General Model*. We add a disturbance inflow $\mathbf{u_d}^n$. The disturbance inflow enters the Two Tank Topology, in tank 1 and in the middle of the gravity pipe. The disturbance inflows points of entry are described by $\mathbf{B_d}$ in (5.1) and shown in *Figure 5.1*.

$$\begin{aligned}
&\mathbf{x}^n = \mathbf{A}\mathbf{x}^{n-1} + \mathbf{B}\mathbf{u}^n + \mathbf{B_d}\mathbf{u_d}^n + \boldsymbol{\Delta} \\
&\quad \Downarrow \\
&\mathbf{x}^n = \mathbf{A}\mathbf{x}^{n-1} + \mathbf{B}\mathbf{u}^n + \begin{bmatrix} \phi_1\Delta t & 0 \\ 0 & 0 \\ 0 & \theta_1\Delta t \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_{d1}^n \\ u_{d2}^n \end{bmatrix} + \boldsymbol{\Delta}
\end{aligned} \tag{5.1}$$

In order for us to evaluate the Kalman filter in this chapter, we will assume that we have full knowledge of the disturbance inflow $\mathbf{u_d}^n$. This is, of course, not the case in real life since the inflows into a sanitation network would be determined by rainfall and wastewater production. The exact mechanics of how knowledge of $\mathbf{u_d}^n$ is obtained for use in a control situation will be presented in *chapter 7 Controlling the Two Tank Topology*.

This chapter will present three main things: an introduction to the Kalman filter, an analysis of the model noise and measurement noise affecting the emulation and simulation, and finally, a qualitative evaluation of the constructed Kalman filter.

## 5.1 The Kalman Filter Algorithm

In this section, a short intro to the Kalman filers structure is given. The Kalman filter algorithm can be said to have three steps. To continuously estimate the system states, these three steps are run repeatedly, using old estimates of state and error covariance. The Kalman filter can be thought of as an algorithm and has the mathematical description [31]:

1. Prediction Step:
$$\hat{\mathbf{x}}_-^n = \mathbf{A}\hat{\mathbf{x}}^{n-1} + \mathbf{B}\mathbf{u}^n + \mathbf{B_d}\mathbf{u_d}^n + \boldsymbol{\Delta}, \tag{5.2}$$
$$\mathbf{P}_-^n = \mathbf{A}\mathbf{P}^{n-1}\mathbf{A}^T + \mathbf{Q}, \tag{5.3}$$
2. Kalman Gain Step:
$$\mathbf{K}^n = \mathbf{P}_-^n\mathbf{C}^T(\mathbf{C}\mathbf{P}_-{}^n\mathbf{C}^T + \mathbf{R})^{-1}, \tag{5.4}$$
3. Estimation Step:
$$\hat{\mathbf{x}}^n = \hat{\mathbf{x}}_- + \mathbf{K}^n(\mathbf{z}^n - \mathbf{C}\hat{\mathbf{x}}_-^n), \tag{5.5}$$
$$\mathbf{P}^n = \mathbf{P}_-^n - \mathbf{K}^n\mathbf{C}\mathbf{P}_-^n. \tag{5.6}$$

Where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\boldsymbol{\Delta}$ are system matrices, $\mathbf{Q}$ and $\mathbf{R}$ are noise covariance matrices, $\hat{\mathbf{x}}_-^n$, $\mathbf{P^n}_-$, $\mathbf{P}^n$ and $\mathbf{K}^n$ are variables used internally in the filter, $\mathbf{z}^n$ and $\mathbf{u}^n$ are inputs into the

filter and $\hat{\mathbf{x}}^n$ is the output from the filter. Note that $n$ is the time index and $(\cdot)_-$ refers to a predicted value and that $\hat{\cdot}$ refers to an estimated value.

Description of the Kalman filter's steps:

- In the **Prediction step** a prediction of the state $\hat{\mathbf{x}}_-^n$ and error covariance $\mathbf{P}_-^n$ is made. This is done base on the model matrices from the Two Tank Topology $\mathbf{A}$, $\mathbf{B}$, $\mathbf{B_d}$ and $\mathbf{\Delta}$ as well as the expected inputs $\mathbf{u}^n$ and $\mathbf{u_d}^n$. Furthermore, the predicted error covariance $\mathbf{P}^n$ is dependant on the variance of the model noise $\mathbf{Q}$. Where model noise is the mismatch between the model and the real system.
- In the **Kalman Gain Step** the Kalman gain is computed. This is done using the previously calculated predicted error covariance $\mathbf{P}_-^n$, the system matrix $\mathbf{C}$, and the variance of the measurement noise $\mathbf{R}$.
- Lastly the **Estimation Step** updates the previously made predictions $\hat{\mathbf{x}}_-^n$ and $\mathbf{P}_-^n$ to estimates $\hat{\mathbf{x}}^n$ and $\mathbf{P}^n$. This is done using the predictions and the Kalman Gain $\mathbf{K}^n$ as well as the measurement from the real system $\mathbf{z}^n$.

There is an important difference between the predicted state and the estimated state. The predicted state $\hat{\mathbf{x}}_-^n$ is a "guess" based only on a model, made before any measurement is made. The Estimated state $\hat{\mathbf{x}}^n$ is the "guess" adjusted based on measurements. This adjustment is made using the weighting error between the expected measured value $\mathbf{C}\hat{\mathbf{x}}_-^n$ and the actual measurement $\mathbf{z}^n$ by the Kalman gain $\mathbf{K}^n$. A similar process is done for the prediction of error covariance $\mathbf{P}_-^n$ and the estimate of error covariance $\mathbf{P}^n$.

## 5.2 Noise Model

For us to use the Kalman filter algorithm, the two noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ need to be determined. The derivation of the Kalman filter is based on the assumption that the measurement noise $\mathbf{V} \sim \mathcal{N}(0, \mathbf{R})$ and the model noise with covariance $\mathbf{W} \sim \mathcal{N}(0, \mathbf{Q})$. In practical applications, this assumption does not change that $\mathbf{Q}$ and $\mathbf{R}$ are often used as tuning parameters and adjusted manually until the desired fit is reached. This adjustment is made knowing that increasing the entries of $\mathbf{R}$ will place more weight on the measurements while increasing $\mathbf{Q}$ will place more weight on the model. While this method of determining $\mathbf{Q}$ and $\mathbf{R}$ is valid, we will use a more analytical approach in this thesis.

Knowing that the Two Tank Topology with disturbance inflow and noise can be express as:

$$\mathbf{x}^n = \mathbf{A}\mathbf{x}^{n-1} + \mathbf{B}\mathbf{u}^n + \mathbf{B_d}\mathbf{u_d}^n + \mathbf{\Delta} + \mathbf{w}^n \tag{5.7a}$$

$$\mathbf{z}^n = \mathbf{C}\mathbf{x}^n + \mathbf{v}^n, \tag{5.7b}$$

where $\mathbf{z}^n$ is a measurement of the systems output, and assuming that the $\mathbf{C}$ from general model developed in *section 2.2 General Model* is exact. Where $\mathbf{w}^n = \mathbf{W}(j)$ [1] can be said to be the additive model noise and $\mathbf{v}^n = \mathbf{V}(j)$ can be said to be the additive measurement noise.

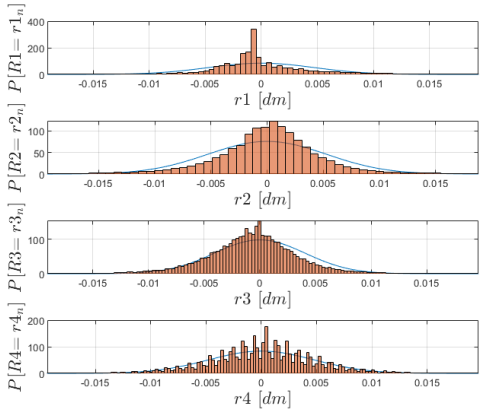
In *section 4.2 System Identification Results* a measure for the fit of the model is found, using residuals. If these residuals are the difference between one step predictions and

[1] Meaning that $\mathbf{w}^n$ is a single realization of the random variable $W$ at an arbitrary point $j$

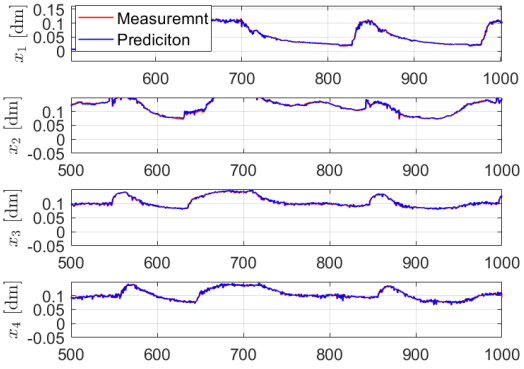
measured values then the distributions of these would be exactly the distributions of the sum of model and measurement noise $\mathbf{W} + \mathbf{V}$. The rest of this chapter will be dedicated to the analysis of the noise observed in the AAU Smart Water Lab.

### 5.2.1 Noise model of the AAU Smart Water Lab

Observing the combined noise distributions shown in *Figure 5.2a* it can be noted that they seem to be normally distributed. These distributions are obtained by comparing lab measurements to one step predictions found using our four pipe model $r_i = |x_i - \hat{x}_i|$. The measurements and one step predictions are shown in *Figure 5.2b*.




**(a) Distribution combined model and measurement noise.**




**(b) One-step predictions.**

**Figure 5.2: Noise identification data. With high sensor precision.**

Though the combined distribution of the model and sensor noise $\mathbf{U} = \mathbf{W} + \mathbf{V}$ is approximately normal, separating the distribution $\mathbf{U}$ into its two terms does not seem straightforward. We do, however suggest one approach, knowing that both $\mathbf{W}$ and $\mathbf{V}$ have zero mean and that they are then completely defined by their covariance matrices.

We start with the assumption that the noise observed on a sensor is independent of the noise observed on the other sensors. Meaning that the off-diagonal entries of the measurement noise covariance matrix $\mathbf{R}$ will be zeros, as shown in (5.8).

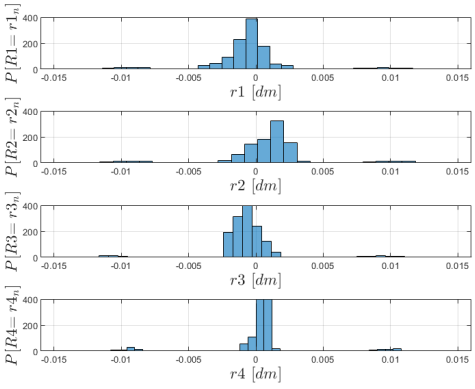
$$\mathbf{R} = \begin{bmatrix} \sigma_{v1}^2 & 0 & 0 & 0 \\ 0 & \sigma_{v2}^2 & 0 & 0 \\ 0 & 0 & \sigma_{v3}^2 & 0 \\ 0 & 0 & 0 & \sigma_{v3}^2 \end{bmatrix} \tag{5.8}$$

This means that any of the entries on the off-diagonal of $\mathbf{U}$'s covariance matrix must be the result of model uncertainties. Therefore we elect to find the off-diagonal entries of the model covariance matrix $\mathbf{Q}$ by looking at $cov(\mathbf{U})$. Note that $cov(\mathbf{U})$ is the sample model covariance of the residuals $r_i = |x_i - \hat{x}_i|$. This yields equation (5.9).

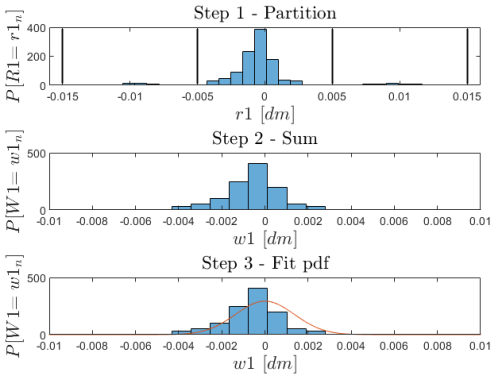
$$\mathbf{Q} = \begin{bmatrix} \sigma^2_{W1} & -0.0425 & 0.0483 & -0.0121 \\ -0.0425 & \sigma^2_{W2} & -0.0141 & 0.0124 \\ 0.0483 & -0.0141 & \sigma^2_{W3} & 0.0314 \\ -0.0121 & 0.0124 & 0.0314 & \sigma^2_{W4} \end{bmatrix} \cdot 10^{-5} \tag{5.9}$$

We still have to determine the variances of both $\mathbf{W}$ and $\mathbf{V}$. We cannot find the variances based on the data presented in *Figure 5.2*. However, we notice that the residuals of the gravity pipe states are almost exclusively between $-0.01$ $dm$ and $0.01$ $dm$.

This leads us to run an experiment where we in software truncated measurements to have a precision of $0.01$ $dm$ or $1$ $mm$. This yields the truncated data results with the new residual distribution shown in *Figure 5.3a*.




**(a) Distribution combined model and measurement noise, with truncated measurements.**




**(b) Three step process applied to create the measurement distribution.**

**Figure 5.3: Noise identification data. With high sensor precision.**

As can be seen from the distributions of the truncated data set, the noise is distributed around points with modulus $1mm$. Knowing the level sensors used in the pipes have a precision of 1 mm, leads us to make an **assumption on the lab noise**: The means of the points around modulus $1mm$ is determined by the measurement noise $\mathbf{V}$, while the distributions around these means are determined by the model noise $\mathbf{W}$.

We, therefore, elect to manipulate the truncated distributions shown in *Figure 5.3a* to create a distribution for the model noise $\mathbf{W}$, which aligns with the assumption on the Lab Noise. For each pipe state a three step process is applied to the residual data:

1. The residual data is partitioned into sections of size 0.01 dm.
2. Then, the sum of each partition is placed in its middle. The reason for placing the sum of each partition in its middle lies in the middle representing a possible measurement value.
3. Lastly, a continuous normal distribution is fitted to the data.

The process described in the above three points is depicted in *Figure 5.3b*. The three step process yields the distributions of the model noise $\mathbf{W}$ shown in *Figure 5.4*.

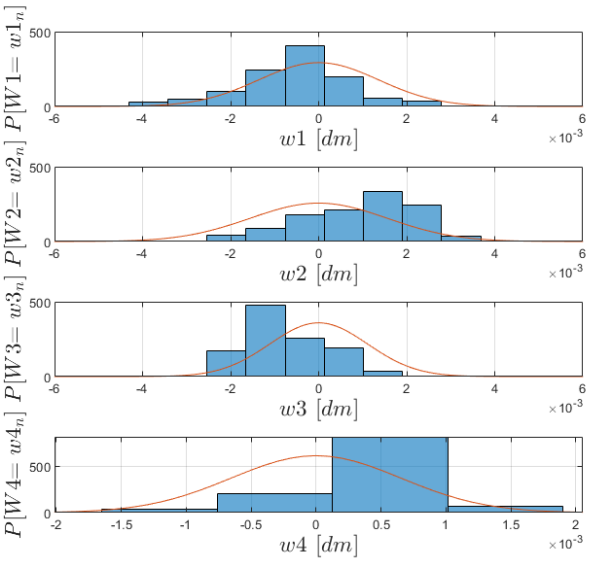



**Figure 5.4: Resulting distributions from applying the three process.**

The variances of the model noise $\sigma^2_{Wj}$ for $j \in \{1, 2, 3, 4\}$ can be inserted into the model noise covariance matrix $\mathbf{Q}$, yielding (5.10). Note that $\mathbf{Q}$ already contains the off-diagonal sample covariances.

$$\mathbf{Q} = \begin{bmatrix} 0.1855 & -0.0425 & 0.0483 & -0.0121 \\ -0.0425 & 0.2415 & -0.0141 & 0.0124 \\ 0.0483 & -0.0141 & 0.1231 & 0.0314 \\ -0.0121 & 0.0124 & 0.0314 & 0.0427 \end{bmatrix} \cdot 10^{-5} \tag{5.10}$$

Having found the covariance matrix $\mathbf{Q}$ and knowing $\mathbf{U} = \mathbf{W} + \mathbf{V}$ we can use the relation $cov(\mathbf{U}) = cov(\mathbf{W}) + cov(\mathbf{V}) = \mathbf{Q} + \mathbf{R}$ for independent variables such that we get $cov(\mathbf{U}) - \mathbf{Q} = \mathbf{R}$. The result is shown in (5.11).

$$\mathbf{R} = \begin{bmatrix} 0.1137 & 0 & 0 & 0 \\ 0 & 0.1388 & 0 & 0 \\ 0 & 0 & 0.0882 & 0 \\ 0 & 0 & 0 & 0.1698 \end{bmatrix} \cdot 10^{-4} \tag{5.11}$$

With this, we have characterized the model noise in (5.10) and measurement noise (5.11), on the AAU Smart Water Lab, using a four pipe state version of the general model. In the next section, a qualitative analysis of the Kalman filter's performance is evaluated.

## 5.3 Resulting Filter

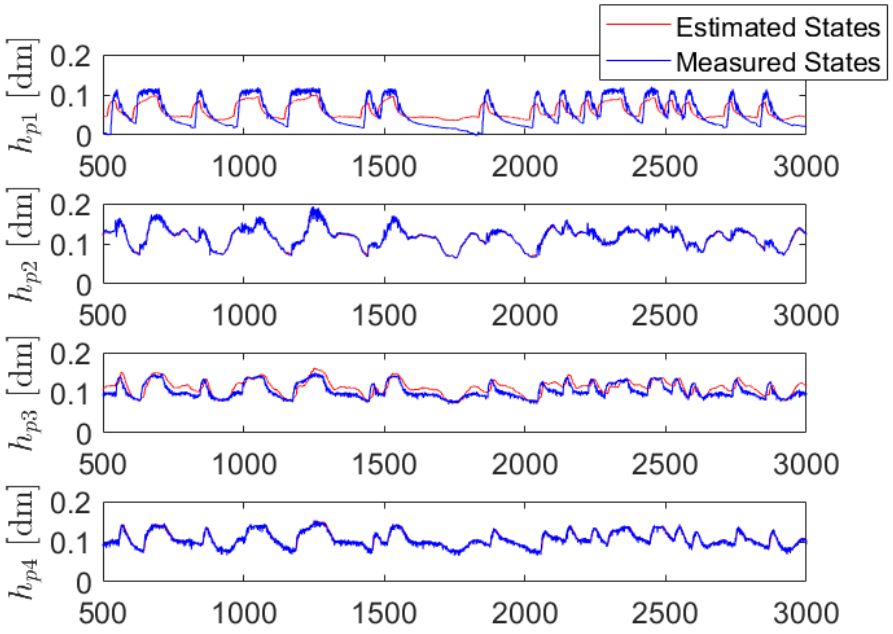
With the covariances for the noise in the lab experiments found, a Kalman filter can be created and in this section a qualitative analysis of one will be presented.

The Kalman filter is used as an estimator on the same data sets used to identify the noise models. Two state measurements were however removed from the measurment set to check the quality of the Kalman filter as an estimator. The removed states are pipe states 1 and 3.

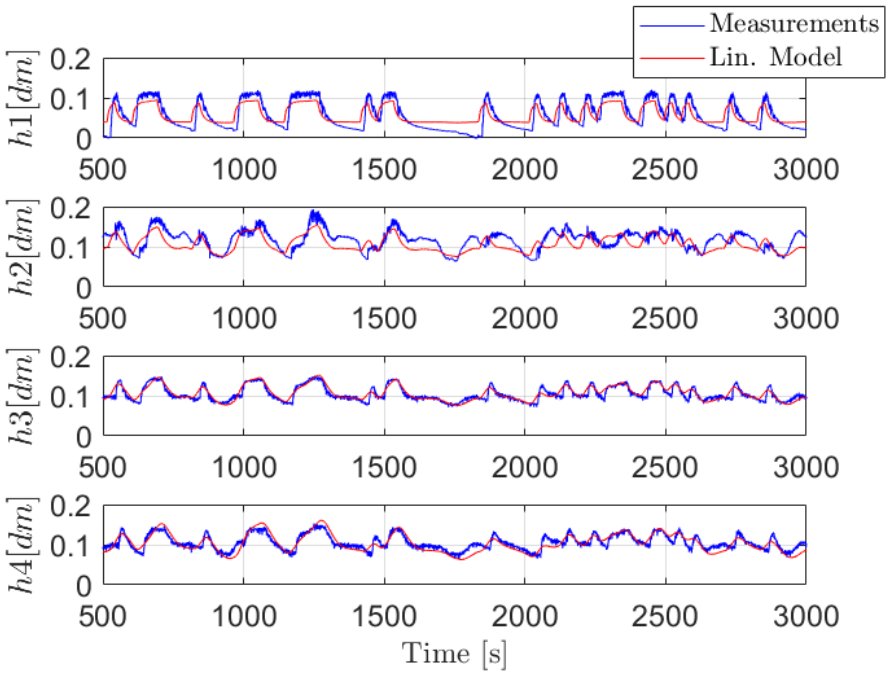
The resulting Kalman gain and error covariance for the estimator was found after having applied the Kalman filter:

$$K = \begin{bmatrix} 0.0602 & 0.0145 \\ 0.3215 & 0.0121 \\ 0.0284 & 0.1534 \\ 0.0148 & 0.1566 \end{bmatrix}, P = 10^{-4} \begin{bmatrix} 0.2105 & 0.0084 & 0.0498 & 0.0025 \\ 0.0084 & 0.0446 & 0.0039 & 0.0021 \\ 0.0498 & 0.0039 & 0.0875 & 0.0261 \\ 0.0025 & 0.0021 & 0.0261 & 0.0266 \end{bmatrix}$$

*Figure 5.5a* shows the estimate of the lab states. The advantage of implementing a Kalman filter, over a Luenberger Observer as an estimator can be seen in the estimation of the lab states. As not only does the Kalman filter estimate the unmeasured state, but it also removes the high frequency sensor noise from the measured states. Note that we added, the free running result, to give the reader a frame of reference. The free running model is shown in *Figure 5.5b*.




**(a) Kalman Filter used for state estimation of pipe state 1 and 3.**




**(b) Free running predictions with lateral inflow.**

**Figure 5.5: Figures for qualitative analysis of Kalman filter performance.**

# Controller Design 6

Having presented the topics which enable us to do Model Predictive control, this chapter starts by presenting the setup of an optimization program for both a "Standard" Model Predictive Controller (MPC) and a Stochastic Model Predictive Controller (SMPC). The SMPC utilizes chance constraints; therefore, we call it the Chance Constrained MPC (CC-MPC).

Since control of the tank level, with overflow prevention in mind, will be an important control objective, we elect to illustrate the CC-MPC framework on a single tank example, see *section 6.3 Single tank SMPC*. Besides illustrating the CC-MPC performance, the goal of the single tank example, in particular, is to emphasize the important challenges in the control of the sanitation networks. We will start with a presentation of MPC structure and formulation, and thereafter we gradually expand the standard structure as we identify new challenges.

## 6.1 Model Predictive Control

The "standard" MPC is a deterministic controller designed under the assumption that there is no uncertainty in the system model, measurement, or input. Particularly for sanitation networks, it would mean that the rain inflows into the network are perfectly known at each step of the optimization horizon. Generally, for any system described by a linear set of system equations, it would mean that the matrices **A B** and **C** would always be exact. Both these assumptions are quite optimistic, but often a standard MPC provides good results despite these assumptions. This is due to its receding horizon implementation.

General for all types of MPCs is that they are represented as an optimization problem, consisting of a cost function and a constraint set. The constraint set making up a part of an MPC includes the dynamics of the system the controller is created for. In our case, these dynamics are the ones derived in *chapter 2 Modeling*.

The fact that dynamics are described by a linear model allows us to formulate the MPC problem as a quadratic program. This is convenient, as the solution to quadratic program can be guaranteed to have a global minimum [32]. The classical formulation of a quadratic cost MPC includes the terms shown in (6.1). Though depending on the application of the MPC, the cost function might include other terms or neglect to include some of the terms shown in (6.1).

$$\mathcal{J} = \min_{\boldsymbol{u}} \sum_{k=1}^{H_p} \left\| \mathbf{z}_k - \mathbf{z}_k^{ref} \right\|_{\mathbf{Q}_k}^2 + \sum_{k=0}^{Hu-1} \left\| \boldsymbol{\Delta} \mathbf{u}_k \right\|_{\mathbf{R}_k}^2 . \tag{6.1}$$

This formulation of the MPC is based on the one presented in [32,p. 55-56 & 74-75]. The cost function from (6.1) consist of the quadratic penalty $\mathbf{R}_k$ on the change of the controller action $\mathbf{\Delta u}_k$, at each time step $k$. The $\left\|\mathbf{z}_k - \mathbf{z}_k^{ref}\right\|^2_{\mathbf{Q}_k}$ term defines the penalty for the deviation of the output $\mathbf{z}_k$ from a desired reference $\mathbf{z}_k^{ref}$.

**Remark:** the notation $\mathbf{z}_k$ is an implicit shorthand for the notation $\hat{z}(n+k|n)$ which means the prediction of $z$ at time $n+k$ made at time $n$. We are aware that this notation differs from the notation used for the previous chapters of this thesis. We, however elect to use the notation as we believe it is suitable for an MPC description. The time between the time indices $k$ is defined as $T_{mpc}$, $Hp \in \mathbb{R}_{\geq 1}$ is the prediction horizon and $Hu \in \mathbb{R}_{\geq 0}$ is the control horizon. When an MPC is used with the general model of the Two Tank Topology $T_{mpc} = \Delta t$.

In the classical MPC formulation, the constraint set is obtained by describing the dynamics of the controlled system. In the case of a system described by linear dynamics, the constraints are linear. A linear system described by a standard state space formulation with known disturbance can be formulated as shown in (6.2), (6.3) and (6.4).

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{B_d}\mathbf{u}_{d_k}, \tag{6.2}$$

$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k, \tag{6.3}$$

$$\mathbf{\Delta u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}. \tag{6.4}$$

The formulation of the bounds presented in (6.2), (6.3) and (6.4), is very close to the form of the linear description of the Two Tank Topology, which was presented in *section 2.2 General Model*, where $\mathbf{u_{d_k}}$ is a forecast of the water inflow into the first tank and middle pipe section. The bounds on system states and inputs form the linear inequality constraints, called box constraints. For a deterministic setup, the box constraints have the following form:

$$\underline{\mathbf{x}}_k \leq \mathbf{x}_k \leq \overline{\mathbf{x}}_k, \tag{6.5}$$

$$\underline{\mathbf{u}}_k \leq \mathbf{u}_k \leq \overline{\mathbf{u}}_k, \tag{6.6}$$

$$\underline{\mathbf{\Delta u}}_k \leq \mathbf{\Delta u}_k \leq \overline{\mathbf{\Delta u}}_k. \tag{6.7}$$

For the system presented in section 2.2, these bounds would be determined by the tank sizes, pipe sizes, and pump capacities. In general, other system specific constraints might also be included.

The above equations serve as an outline for a standard MPC formulation. In the following section, we are expanding the base framework to a Stochastic MPC. To that end, we introduce the uncertainties found in a sanitation system and explore one way of handling them in the SMPC.

## 6.2 Stochastic Model Predictive Control

So far, we only presented the deterministic MPC problem. Due to the strict assumptions on knowledge of the system, it makes sense to extend the standard MPC program by

taking a system's model uncertainties into account, such as prediction uncertainty and model uncertainty.

We now assume that our model, measurements and disturbance predictions are uncertain. We elect to handle uncertainty as stochastic additive uncertainty. The predicted model uncertainty, denoted as $\mathbf{w}_k$ and the predicted measurement uncertainty $\mathbf{v}_k$ are random vectors that follow a zero mean Gaussian distribution, $\mathbf{W} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma_w})$ and $\mathbf{V} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma_v})$. Additionally, we formulate the disturbance prediction as consisting of a mean disturbance $\mathbf{d}_{k,mean}$ which is deterministic and a additive stochastic part $\mathbf{d}_{k,uncertain} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma_d}_k)$, resulting in the disturbance vector $\mathbf{u}_{d_k} = \mathbf{d}_{k,mean} + \mathbf{d}_{k,uncertain}$.

The presence of stochastic uncertainty requires a reformulation of the deterministic MPC program from the previous section. In the deterministic case, the optimum is obtained by minimizing the cost of a known system. Since the actual system is unknown in the stochastic case, the optimal cost has to be obtained by minimizing the expected cost. The reformulation is shown in (6.8) and (6.9).

$$\mathcal{J} = \min_{\boldsymbol{u}} E\left[\sum_{k=0}^{H_p-1} \left\|\mathbf{z}_k - \mathbf{z}_k^{ref}\right\|_{\mathbf{Q}}^2 + \|\mathbf{\Delta u}_k\|_{\mathbf{R}}^2\right] \tag{6.8}$$

$$= \min_{\boldsymbol{u}} \sum_{k=0}^{H_p-1} \left\|\mu_{\mathbf{z}k} - \mathbf{z}_k^{ref}\right\|_{\mathbf{Q}}^2 + tr(\mathbf{Q}\mathbf{\Sigma_{z_k}}) + \|\mathbf{\Delta u}_k\|_{\mathbf{R}}^2 \tag{6.9}$$

where $tr()$ is the trace operator. The derivation of (6.9) from (6.8) can be found in Appendix D. Notice that the obtained cost function is deterministic in terms of the mean $\mu_{\mathbf{z_k}}$ and variance $\mathbf{\Sigma_z}$ of the output, rather than dependant on the actual output $\mathbf{z_k}$.

The equality constraints also have to be reformulated. This is done in the same way as the reformulation of the cost function; by describing the expected behavior of the state $E[\mathbf{x}_{k+1}]$ and output $E[\mathbf{z}_k]$. Taking all the uncertainties described at the beginning of the section, into account the mean system dynamics can be presented as shown in (6.10) and (6.11).

$$E[\mathbf{x}_{k+1}] = \mu_{\mathbf{x}k+1} = \mathbf{A}\mu_{\mathbf{x}k} + \mathbf{B}\mathbf{u}_k + \mathbf{B_d}\mathbf{d}_{k,mean}, \tag{6.10}$$

$$E[\mathbf{z}_k] = \mu_{\mathbf{z}k} = \mathbf{C}\mu_{\mathbf{x}k}, \tag{6.11}$$

As the variance $\mathbf{\Sigma_{z_k}}$ at each time step is a part of the cost function (6.9) the state variance dynamics has to be described as well. This is done in (6.12), (6.13) and (6.14).

$$\mathbf{\Sigma}_{\mathbf{x}k+1} = \mathbf{A}\mathbf{\Sigma}_{\mathbf{x}k}\mathbf{A}^T + \mathbf{B_d}\mathbf{\Sigma}_{\mathbf{d}k}\mathbf{B_d}^T + \mathbf{\Sigma_w}, \tag{6.12}$$

$$\mathbf{\Sigma}_{\mathbf{z}k} = \mathbf{C}\mathbf{\Sigma}_{\mathbf{x}k}\mathbf{C}^T + \mathbf{\Sigma}_{\mathbf{v}k}. \tag{6.13}$$

$$\mathbf{\Sigma}_{\mathbf{d}k+1} = f(\mathbf{\Sigma}_{\mathbf{d}k}, k), \tag{6.14}$$

The dynamics of the disturbance variance $\mathbf{\Sigma}_{\mathbf{d}k+1}$ can, in principle, take any form, as long as they are not a function of the optimization parameter $\mathbf{u}_k$ or the states $\mathbf{x}_k$. If the disturbance variance for example take the form shown in (6.14) the variance can be pre-calculated and the term $tr(\mathbf{Q}\mathbf{\Sigma_{z_k}})$ can be assumed constant. Note that the variance of the

state in this MPC problem is described by a linear open loop state space equation (6.12). That means that the variance of $\mathbf{z_k}$ grows and settles for stable systems as $k$ increases. This is unavoidable, and we quickly grow unsure of our prediction.

One solution for lowering this growth in uncertainty is to introduce a so-called: Closed Loop Tube Control Policy [33]. The idea is to utilized a controller $\upsilon_k = -\mathbf{K}(\mathbf{x}_k - \mu_{\mathbf{x}k})$ in a closed loop setting to drive the variance towards a steady state. This makes the control action on the system $\boldsymbol{\Upsilon}_k = \mathbf{u}_k + \upsilon_k$. For the tube control policy $\upsilon$ to drive the variances towards a steady state, it is required that all additive uncertainties have constant variances at each time step $k$ in the prediction horizon $H_p$. The $\mu_{\mathbf{x}k}$ is the prediction of the state, while the $\mathbf{x}_k$ is the actual state value at time $k$. This means that the tube controller drives the states towards the mean dynamics. Introducing the controller yields the following closed loop variance dynamics:

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{x}k+1} &= (\mathbf{A}-\mathbf{BK})\boldsymbol{\Sigma}_{\mathbf{x}k}(\mathbf{A}-\mathbf{BK})^T + \mathbf{B_d}\boldsymbol{\Sigma}_{\mathbf{d}k}\mathbf{B_d}^T + \boldsymbol{\Sigma}_{\mathbf{w}}, \\ \boldsymbol{\Sigma}_{\mathbf{d}k+1} &= f(\boldsymbol{\Sigma}_{\mathbf{d}k}, k), \\ \boldsymbol{\Sigma}_{\mathbf{z}k} &= \mathbf{C}\boldsymbol{\Sigma}_{\mathbf{x}k}\mathbf{C}^T + \boldsymbol{\Sigma}_{\mathbf{v}k}.\end{aligned} \tag{6.15}$$

A tube controller also makes the implemented input $\boldsymbol{\Upsilon}_k$ a random variable, since it is dependant on $\mathbf{x}_k$, which is a random variable. Therefore if we penalize the implemented input we have to expand the cost function from (6.9) in order to account for uncertainty of the $\boldsymbol{\Upsilon}$ policy. The redefined cost is shown in (6.16).

$$\mathcal{J} = \min_{\mathbf{u}} \sum_{k=0}^{H_p-1} \left\| \mu_{\mathbf{z}|k} - \mathbf{z}_k^{ref} \right\|_{\mathbf{Q}}^2 + tr(\mathbf{Q}\boldsymbol{\Sigma}_{\mathbf{z}|k}) + \sum_{k=0}^{Hu-1} \left\| \mu_{\boldsymbol{\Delta}\mathbf{u}_k} \right\|_{\mathbf{R}}^2 + tr(\mathbf{R}\boldsymbol{\Sigma}_{\boldsymbol{\Delta}\boldsymbol{\Upsilon}|\mathbf{k}}), \tag{6.16}$$

where the $tr(\mathbf{R}\boldsymbol{\Sigma}_{\boldsymbol{\Delta}\boldsymbol{\Upsilon}|\mathbf{k}})$ term, like the $tr(\mathbf{Q}\boldsymbol{\Sigma}_{\mathbf{z}|k})$ term, is independent of chosen optimal control policy, determined by the MPC. Both trace terms can be assumed constant and therefore they can be neglected from the cost function.

The feedback matrix $\mathbf{K}$ can be found using standard LQR control synthesis. In the MPC framework the tube controller action is implemented on top of the MPC's optimal control action as $\upsilon_0 = -\mathbf{K}(\mathbf{x}_0 - \mu_{\mathbf{x}0})$. The $\mathbf{x}_0$ is the initial state condition for the MPC optimization problem, and the $\mu_{\mathbf{x}0}$ is the prediction of the initial state condition from the previous MPC iteration.

We choose not to expand $tr(\mathbf{R}\boldsymbol{\Sigma}_{\boldsymbol{\Delta}\boldsymbol{\Upsilon}|\mathbf{k}})$ term in this chapter. The tube policy with LQR tuning is implemented later the CC-MPC for the lab experiment, see *subsection 7.2.4 The Chance Constrained Model Predictive controller used in a second solution.*

With the cost function and equality constraints converted to fit in a stochastic framework, we are only missing a description of the inequality constraints. In principle, the box constraints presented in section 6.1 could simply be described in terms of their mean values. But as the focus of the thesis is control of sanitation networks, such an approach could result in large wastewater overflows if the disturbance forecast has high uncertainty. Instead, we choose to utilize chance constraints, which are presented in the following section.

### 6.2.1 Chance constraints

Authors presenting the Stochastic MPC usually refer to chance constraints as the method for constraining a system. The chance constraints define the constraint on the probability of deterministic constraints being true.

$$\mathcal{P}(\mathbf{x}_k \in \mathcal{X}) \geq \alpha, \tag{6.17}$$

where the set $\mathcal{X}$ represents a set of possible values for $\mathbf{x}_k$, which define the deterministic boundaries [11]. The result of the probabilistic constraint is a tightening of the deterministic set that will guarantee for any states realizations that the states are inside the set with $\alpha$ probability. For sanitation networks, this for example, allows us to put a probabilistic bound on wastewater overflows, even under large uncertainties in a forecast.

Consider now that the set $\mathcal{X}$ represents all possible realizations of states $\mathbf{x}$ that are inside the box constraint defined in (6.5). In other words, the set $\mathcal{X}$ is an intersection of $n = \dim(\bar{\mathbf{x}}) + \dim(\underline{\mathbf{x}})$ halfspaces as:

$$\mathcal{X} = \left\{ \mathbf{x}_k | \mathbf{h}_j^T \mathbf{x}_k \leq b_j, \forall j = 1, 2, ..., n \right\}, \tag{6.18}$$

where $\mathbf{h}_j \in \mathbb{R}^{\dim(\mathbf{x})}$ is a vector that selects a single state in $\mathbf{x}$ and $b_j \in \mathbb{R}_+$ is a corresponding deterministic bound [11]. According to [11] and using Boole's inequality the chance constraint (6.17) can be defined as a sum of the individual half space chance constraints as:

$$\mathcal{P}(\mathbf{x}_k \in \mathcal{X}) \leq \sum_{j=1}^{n} \mathcal{P}(\mathbf{h}_j^T \mathbf{x}_k \leq b_j). \tag{6.19}$$

Now we proceed with deriving a single chance constraint for the upper bound deterministic constraint. The rest are derived analogously. The selected state $\mathbf{h}_j^T \mathbf{x}_k$ has to remain under the deterministic bound $\mathbf{h}_j^T \bar{\mathbf{x}}_k$ with $\alpha_j$ probability:

$$\mathcal{P}(\mathbf{h}_j^T \mathbf{x}_k \leq \mathbf{h}_j^T \bar{\mathbf{x}}_k) \geq \alpha_j, \tag{6.20}$$

where the left side of the inequality defines a cumulative distribution function (CDF) of $\mathbf{h}_j^T \mathbf{x}_k$. For normal distributions it is dependant only on the mean $\mathbf{h}_j^T \mu_{\mathbf{x}k}$ and variance $\mathbf{h}_j^T \mathbf{\Sigma}_{\mathbf{x}k} \mathbf{h}_j$ of the random variable, as in (6.21) [11].

$$\mathcal{P}(\mathbf{h}_j^T \mathbf{x}_k \leq \mathbf{h}_j^T \bar{\mathbf{x}}_k) = \phi \left( \frac{\mathbf{h}_j^T \bar{\mathbf{x}}_k - \mathbf{h}_j^T \mu_{\mathbf{x}k}}{\sqrt{\mathbf{h}_j^T \mathbf{\Sigma}_{\mathbf{x}k} \mathbf{h}_j}} \right), \tag{6.21}$$

where $\phi(\bar{x}) \triangleq \mathcal{P}(x \leq \bar{x})$ is the CDF of a standard normal distribution with $x \sim \mathcal{N}(0, 1)$. Now substituting (6.21) into (6.20) we obtain the expression (6.23) for chance constraint

tightening as:

$$\phi\left(\frac{\mathbf{h}_j^T\bar{\mathbf{x}}_k - \mathbf{h}_j^T\mu_{\mathbf{x}k}}{\sqrt{\mathbf{h}_j^T\boldsymbol{\Sigma}_{\mathbf{x}k}\mathbf{h}_j}}\right) \geq \alpha_j, \tag{6.22}$$

$$\Updownarrow$$

$$\frac{\mathbf{h}_j^T\bar{\mathbf{x}}_k - \mathbf{h}_j^T\mu_{\mathbf{x}k}}{\sqrt{\mathbf{h}_j^T\boldsymbol{\Sigma}_{\mathbf{x}k}\mathbf{h}_j}} \geq \phi^{-1}(\alpha_j),$$

$$\Updownarrow$$

$$-\mathbf{h}_j^T\mu_{\mathbf{x}k} \geq -\mathbf{h}_j^T\bar{\mathbf{x}}_k + \sqrt{\mathbf{h}_j^T\boldsymbol{\Sigma}_{\mathbf{x}k}\mathbf{h}_j}\phi^{-1}(\alpha_j),$$

$$\Updownarrow$$

$$\mathbf{h}_j^T\mu_{\mathbf{x}k} \leq \mathbf{h}_j^T\bar{\mathbf{x}}_k \underbrace{-\sqrt{\mathbf{h}_j^T\boldsymbol{\Sigma}_{\mathbf{x}k}\mathbf{h}_j}\phi^{-1}(\alpha_j)}_{\text{tightening term}}, \tag{6.23}$$

where $\phi^{-1}(\alpha_J)$ is the inverse cumulative distribution function of a standard normal distribution [11]. Notice that we obtained a deterministic constraint on the expected value of the state. The initial boundary on the states $\bar{\mathbf{x}}_k$ is tightened each step by the term $-\sqrt{\mathbf{h}_j^T\boldsymbol{\Sigma}_{\mathbf{x}k}\mathbf{h}_j}\phi^{-1}(\alpha_j)$. If the uncertainty in our prediction grows, the result will be the increase in tightening of the states. This tightening will make sure that the state predictions stay in deterministic bounds (6.5) for any realization of the disturbance with $\alpha \cdot 100\%$ certainty. On the other hand this tightening can cause the infeasibility of the optimization problem.

For example, in the application where the chance constraints is used for tightening the upper bound of a retention tanks level, heavy inflows may cause infeasibility. This is because during some heavy inflow scenarios, even with a maximum outflow, the tank level can not be kept under the tightened constraint generally. If the tightening is larger, then the range of tightened state $(\overline{\mathbf{x}} - \underline{\mathbf{x}})$, the state constraints will become ill-posed.

In the rest of the chapter, we aim to illustrate the concepts of SMPC through an example application to a single tank.

## 6.3 Single tank SMPC

In the modeling chapter, *subsection 2.1.2 Retention tank*, we have described the single tank dynamics as a flow integration. To validate and prove the concepts from previous sections, we will restate the model with specific parameters. The configuration that we consider in this section is depicted in *Figure 6.1.* From that, we can derive a simple discrete state space equation describing the tank level $h$, shown in (6.24).

$$h_{k+1} = h_k + \frac{\Delta T}{A}(d_k - u_k - q_{of|k}), \tag{6.24}$$

where $d$ is the inflow disturbance, $u$ is the controlled outflow and $q_{of}$ is the flow that is a result of the tank overflow. Model parameters $\Delta T$ and $A$ denote respectively the sampling

time and the tank cross-sectional area. The parameters of the state space model are shown in table 6.1.

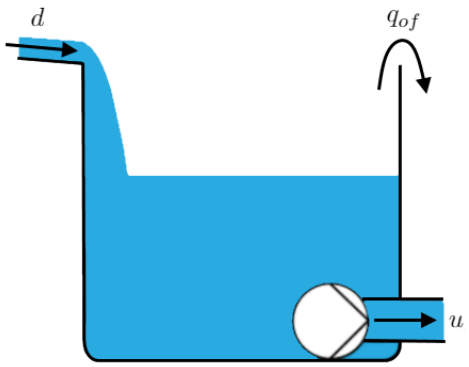



**Figure 6.1: Single tank system used for concept validation.**

| Parameter | Value |
|---|---|
| $\mathbf{A}$ | 1 |
| $\mathbf{B} \triangleq \frac{\Delta T}{A}$ | 1 |
| $\mathbf{B_d} \triangleq \frac{\Delta T}{A}$ | 1 |
| $\mathbf{C}$ | 1 |
| $\mathbf{\Delta}$ | 0 |
| $\mathbf{\Sigma_w}$ | 0 |
| $\mathbf{\Sigma_v}$ | 0 |
| $\mathbf{A_d}$ | 1 |
| $\mathbf{\Delta_d}$ | 0 |
| $\mathbf{\Sigma_{d0}}$ | 0.0045 |
| $\mathbf{\Sigma_{x0}}$ | 0 |

**Table 6.1: Single tank model parameters.**

It is important to specify what inflow disturbance we consider in this setup. For the sake of providing an application example, we declare that the inflow disturbance represents the collected rain runoff that enters the tank combined with the wastewater flow from a connected adjoining network. At this point, we do not form the pattern of the disturbance inflows to resemble flows observed in a real network, as it is not a focus of this section.´An attempt at creating realistic data patterns for use in emulation will be presented in *subsection 7.1.1 Disturbance design.* It is assumed that we can obtain an uncertain inflow forecast at any time $k$ for $N$ steps in the future, i.e., the $N$ steps disturbance prediction is known at any time $k$. The uncertainty of a prediction is assumed to be normally distributed with constant variance $\Sigma_d = \Sigma_{d0} = 0.0045$, see (6.14) and table 6.1.

We can now formulate an MPC optimization problem. The MPCs objective is to minimize the overflow of the tank. The cost function punishes deviation from a reference in tank level as well as the control action, the pumped outflow, simulating a cost to running the pumps. The uncertainty, which is propagated from disturbance into the state, results in the stochastic cost function. We therefore use the expected cost, which is defined as shown in (6.9) with $\text{tr}(Q\Sigma_{h_k}) = Q\Sigma_{h_k}$. The final formulation of the expected cost is shown in (6.25).

$$\begin{aligned} \mathcal{J} &= \min_u E\left[\sum_{k=0}^{H_p-1} Q(h_k - h_{ref|k})^2 + Ru_k^2\right] \\ &= \min_u \sum_{k=0}^{H_p-1} Q(\mu_{h_k} - h_{ref|k})^2 + Q\Sigma_{h_k} + Ru_k^2, \end{aligned} \tag{6.25}$$

where $\mu_{h_k}$ represents the mean tank dynamics and $\Sigma_{h_k}$ represents the state prediction variance. The mean dynamics and variance dynamics are stated in (6.26) and (6.27).

$$\mu_{h_{k+1}} = \mu_{h_k} + \phi(d_{k,mean} - u_k - q_{of|k}), \tag{6.26}$$
$$\Sigma_{h_{k+1}} = \Sigma_{h_k} + \Sigma_d, \tag{6.27}$$

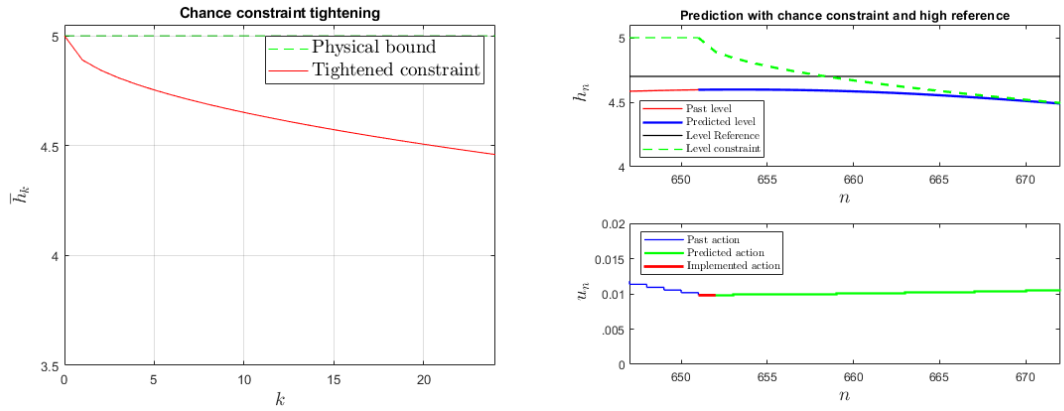
where $\phi = \frac{\Delta T}{A}$ is the tank parameter. The $d_{k,mean}$ represents a rain forecast and $q_{of|k}$ is the overflow at time $k$. We also have to specify physical constraints on the system. The tank level $\mu_{h_k}$ and the pump flow $u_k$ is constrained as:

$$2 \leq \mu_{h_k} \leq 5, \tag{6.28}$$
$$0 \leq u_k \leq 0.03, \tag{6.29}$$
$$-0.03 \leq \Delta u_k \leq 0.03. \tag{6.30}$$

When the tank level is operated close to the upper bound, for example in the heavy rain scenarios, the controller can potentially underperform due to stochasticity in the input and therefore cause unnecessary overflow. In order to prevent that, we define the chance constraint on the upper bound according to (6.23). The acceptable constraint violation is chosen to be 5%. The volume constraint is reformulated with a chance constraint tightening as:

$$2 \leq \mu_{h_k} \leq 5 - \sqrt{\Sigma_{h_k}}\phi^{-1}(0.95). \tag{6.31}$$

The $-\sqrt{\Sigma_{h_k}}\phi^{-1}(0.95)$ represents a tightening factor of the physical constraint defined in (6.28). The upper level bound is now dependent on the uncertainty of the prediction at each step throughout the horizon length. If we now calculate the uncertainty at each time step according to (6.27) with the assumed uncertainties defined in the beginning, we obtain the more conservative or "tighter" upper constraint as showed in *Figure 6.2a*. The longer the horizon is, the more uncertain the prediction becomes, since no local controller is used to control the variance dynamics in (6.27).




**(a) Upper level constraint tightening for $H_p = 24$ step prediction horizon.**

**(b) Level prediction with chance constraint tightening.**

**Figure 6.2: Constraint tightening.**

In *Figure 6.2a* the chance constrained upper level bound is showed in red, with the prediction horizon length of 24 steps. This tightening will result in a more conservative controller action near the upper bound. To demonstrate this, we perform an experiment where the reference is set to be close to the upper bound. The result is shown in *Figure 6.3*. The level reference that the controller is trying to follow is $h_{ref} = 4.7$. Notice that in the *Figure 6.3* bottom plot, the level is always below the reference and does not follow it. The controller is more cautious around the physical upper bound and does not allow the level to grow.

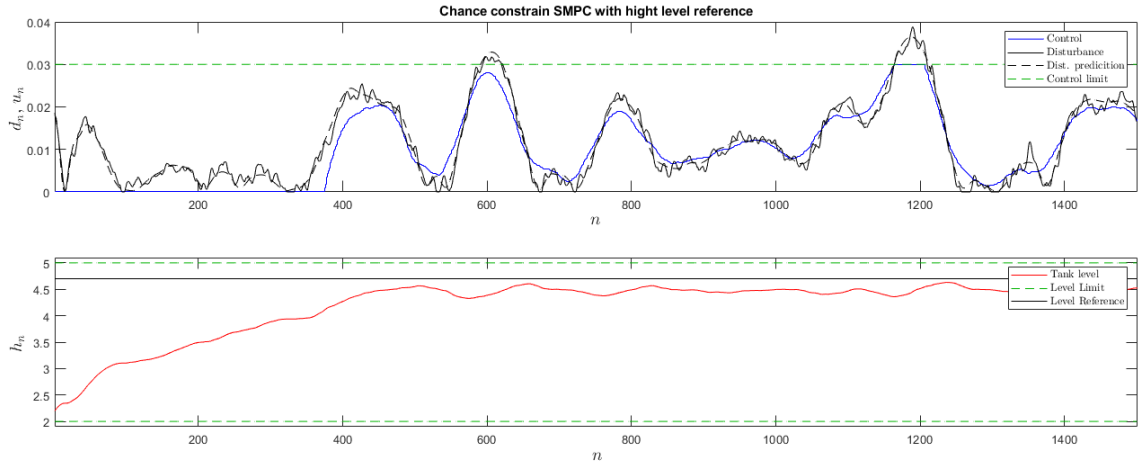



**Figure 6.3: Single tank controller conservatives due to chance constraint tightening.**

In *Figure 6.2b* a single MPC prediction step is depict, specifically the prediction step at $k = 652$. We know from before that the prediction has to stay under the tightened constraint thorough the entire prediction horizon. In fact, the MPC performs such that the tightened constraint is never violated. At this point, one can argue that this "underperformance" in reference tracking can be reduced if we allow more aggressive control action $\Delta u$. While this is a valid argument, which is also verified in Appendix E, keep in mind that the presented experiment in *Figure 6.3* is an artificial setting with the main purpose to outline the conservatives of the CC-SMPC.

### 6.3.1 Feasibility concerns

So far, we only considered a tank system under normal conditions. This setup means that the disturbance's volume is generally smaller than the maximum volume that can be removed from the system. That way it is unlikely for the system to break the defined bounds (6.31) and thus result in an unfeasible optimization. However, in a realistic setup, the tank would occasionally operate under harder conditions. During certain heavy rain scenarios, overflow of the tank is inevitable. Therefore, we have to expand the current SMPC setup to handle these scenarios and ensure feasibility.
The standard way of ensuring feasibility is with a slack variable. The use of slack variables is common in research dealing with tank overflows, and we suggest the reader refer to the following paper for more details [8].

With slack variables implemented for the level constraint in (6.31), the upper bound can relaxed for a high cost to ensure feasibility. This implies that the slack variable becomes the new optimization variable that has to be included in the objective function, with a high cost, compared to other objectives. A redefined cost function and level chance constraint with slack variable are stated in (6.32) and (6.33) and (6.34) respectively.

$$\mathcal{J} = \min_{u, S_{\text{ub}}} \sum_{k=0}^{H_p-1} Q(\mu_{h_k} - h_{ref|k})^2 + Q\Sigma_{h_k} + Ru_k^2 + C_{\text{ub}} S_{\text{ub}|k}, \tag{6.32}$$

$$2 + S_{\text{ub}|k} \leq \mu_{h_k} \leq 5 - \sqrt{\Sigma_{h_k}}\phi^{-1}(0.95) + S_{\text{ub}|k}, \tag{6.33}$$

$$0 \leq S_{\text{ub}|k}, \tag{6.34}$$

where $C_{\text{ub}}$ is the costs of using the slack $S_{\text{ub}}$. For cost $C_{\text{ub}}$ applies $C_{\text{ub}} \gg Q$ in order to achieve the desired behaviour. In case of an overflow the slack is added from both sides of the constraint, thereby conserving the physical tank volume [8]. Notice that the slacks enter the cost function linearly. The reason is that for the quadratic function, a square of very small value is even smaller. In an MPC notion, it is cheap to use very little amounts of slack all the time. The slack is also constrained to only take positive values. Otherwise, because of the linear cost the cost function will be reduced indefinitely. In *Figure 6.4* an example with two overflow scenarios captured with a slack variable is shown, in pink.

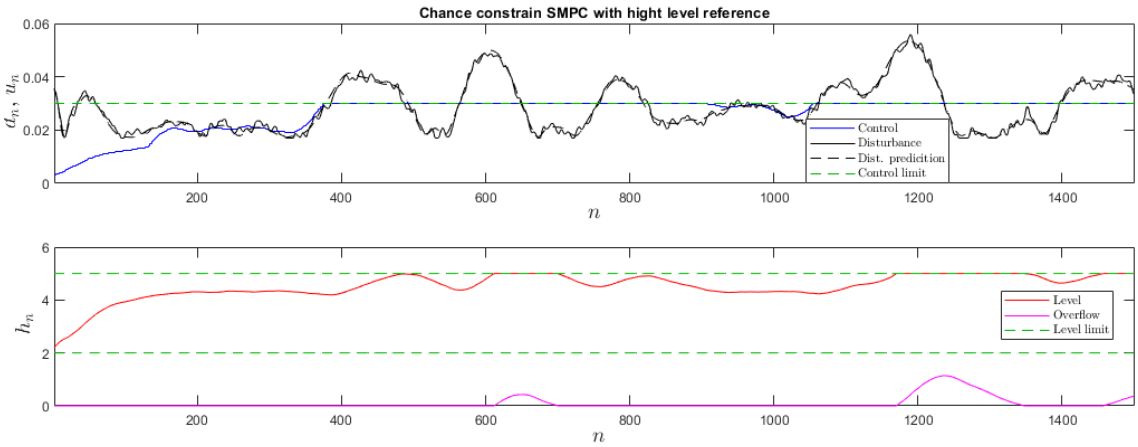



**Figure 6.4: Slack varaible capturing overflow.**

Although this method of using slack variables has proven to be useful in most MPC applications for overflow prevention, it has some flaws when applied in a CC-SMPC framework. In the following subsection, we present why this method can be insufficient. We also further improve the method to compensate for the insufficiency.

### 6.3.2 Problems with using the conventional slack variable approach

The biggest reason why the previous method for overflow handling is insufficient with chance constraint problem is that the slack $S_{\text{ub}}$ captures both chance constraint relaxation

$\sqrt{\Sigma_{h_k}}\phi^{-1}(0.95)$ and the actual overflow. This is not well illustrated in the example shown in *Figure 6.4*. This is because the tightening $\sqrt{\Sigma_{h_k}}\phi^{-1}(0.95)$ at $k = 0$ is always 0, see *Figure 6.2a*.

In general, when the measurement uncertainty is present in the system this is not the case, see variance dynamics at (6.12) and (6.13).
In summary, the problem with combining overflow with CC relaxation is that the slack captures both overflow and volume tightening. Additionally, the CC relaxation and the overflow have the same cost. It would be beneficial if we can specify a different cost for CC relaxation and overflow. Intuitively a constraint relaxation, although it increases the risk of an overflow, is not as harmful as an actual overflow.

Another consideration that is especially valid for MPC of sanitation network research is that by adding the slack variable on both sides of the constraint (6.33), the tank level is raised beyond actual physical values. This can cause problems in cases when the tank is a part of a bigger model, in particular at the end of the gravity pipe with a submerged flow, check *subsection 2.1.3 Pipe boundary conditions*. Since the pipe model depends on the tank level, the pipe model would break or behave strangely. A workaround for this problem can be built, but it would ruin the integrity of the presented models.

### 6.3.3 Solution to conventional slack variable problem

In this section, we proceed with an alternative construction of a CC-SMPC with overflow capturing.

First we keep the $S_{\text{ub}}$ slack variable, but in a slightly altered configuration. We only intend to use this slack variable to relax the chance constraint. We therefore place an upper constraint on the slack variable that is the maximum of the chance constraint tightening factor $\sqrt{\Sigma_{h_k}}\phi^{-1}(0.95)$, shown in (6.35).

$$0 \leq S_{\text{ub}|k} \leq \sqrt{\Sigma_{h_k}}\phi^{-1}(0.95). \tag{6.35}$$

Second, the volume that is contained in the slack variable has to remain in the tank and should not be removed as an overflow. Therefore in (6.33) we include the slack only in the upper bound, as:

$$2 \leq \mu_{h_k} \leq 5 - \sqrt{\Sigma_{h_k}}\phi^{-1}(0.95) + S_{\text{ub}|k}. \tag{6.36}$$

We have lost the ability to handle the overflow, and the optimization becomes unfeasible if overflow happens. In order to solve the infeasibility problem we utilize the flow $q_{of|k}$ presented in the original model, see (6.24). At the same time, we remove the level that corresponds to the overflow. That way, the model will not end up in an unfeasible state value. We elect to consider $q_{of|k}$ as another controller variable. It thus has to be included as an optimization variable in the CC-MPC problem. The cost function is therefore expanded with these new variables, as shown in (6.37).

$$\mathcal{J} = \min_{u, S_{\text{ub}}, q_{of}} \sum_{k=0}^{H_p-1} Q(\mu_{h_k} - h_{ref|k})^2 + Q\Sigma_{h_k} + Ru_k^2 + C_{\text{ub}} S_{\text{ub}|k} + P_k q_{of|k}, \tag{6.37}$$

where $P_k$ is the cost on an overflow variable $q_{of|k}$. In this configuration, the controller itself will "define" the overflow amount. Therefore we have to be careful to set the proper cost on the overflow variable. Generally, it should apply that $P_k \gg C_{\text{ub}} \gg Q$. The only problem with this solution is that in some combination of $P_k$ and $Q$ and the current state $h_k$, an optimal solution can result in an early overflow $q_{of}$. This is unwanted behavior, as we only want overflow if it is inevitable. We can observe such unwanted behavior in the example, shown in *Figure 6.5*.

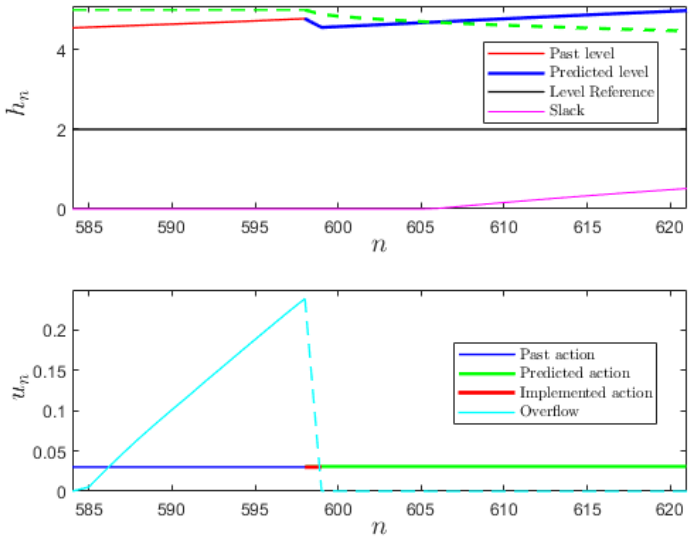



**Figure 6.5: Controller wants to use overflow prematurely in order to reduce the level.**

Despite a very large cost on overflow, the controller uses overflow prematurely to remove some water from the tank. We can see in *Figure 6.5* that the overflow (in light blue) increased before the actual tank level (in red) reached a physical maximum. Since this does not resemble a real-world behavior, the solution is "meaningless". Notice that the controller predicts a decrease in level, shown in blue in *Figure 6.5*, due to the premature overflow.

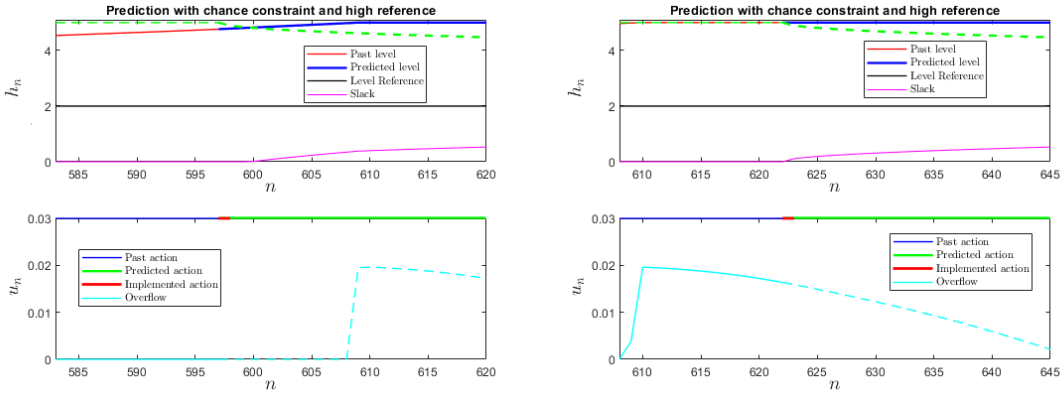
We conclude that only requiring that overflow has a relatively high cost compared to every other objective in (6.37) is insufficient. Therefore we elect to choose $P_k$ as a time dependant linearly decreasing function. That way, using the overflow variable is the cheapest at the end of the prediction horizon. It is still the most expensive variable to use at each time step. The requirements on the overflow variables cost $P_k$ can be written as:

$$P_k \gg C_{\text{ub}} \gg Q, \quad \forall k = 0, ..., H_p - 1, \tag{6.38}$$

$$\Delta P_k = -\mathcal{C}, \quad \forall k = 0, ..., H_p - 2, \quad \mathcal{C} \in \mathbb{R}^+, \tag{6.39}$$

$$\Delta P_k \triangleq k \mapsto P_{k+1} - P_k. \tag{6.40}$$

With these requirements on $P_k$, we expect overflow at $k = 0$ to happen only if necessary. To validate our expectation, we use the same experiment setup for the single tank as we used in the example shown in *Figure 6.5*. The prediction at the same time $n = 598$ is depicted in *Figure 6.6a*. Unlike in the previous example, the overflow is actually predicted when the level reaches maximum. At $n = 607$, overflow happens and is captured in $q_{of}$, see *Figure 6.6b*.




**(a) Overflow prediction in the future.** **(b) Actual overflow captured in $q_{of}$.**

**Figure 6.6: Decreasing cost on overflow variable**

This method of considering the overflow as an optimization variable together with time dependant cost function provides the desired behavior for our tank CC-MPC. The full experiment from which the examples on fig*Figure 6.6a* and *Figure 6.6b* has been taken is shown in *Figure 6.7*.

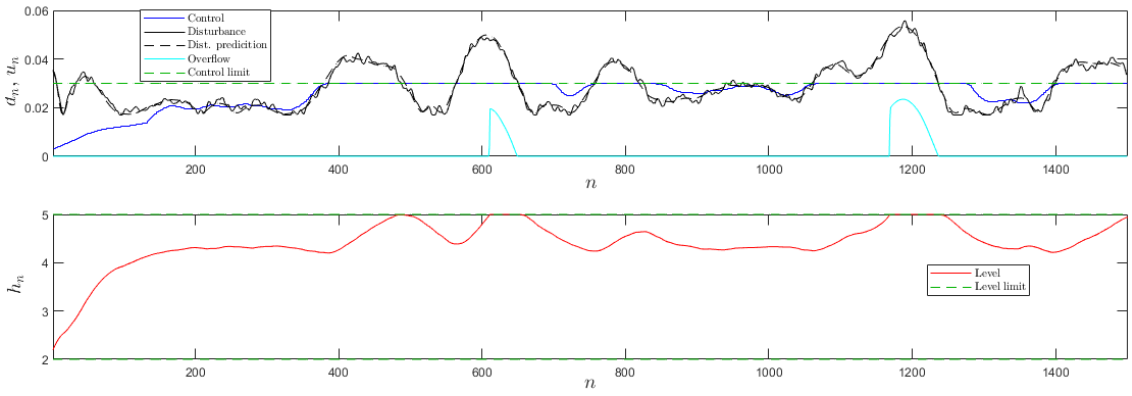



**Figure 6.7: Overflow captured in an optimization variable $q_{of}$**

### 6.3.4 Final formulation of Single Tank CC-SMPC

Finally, in this section, we summarize all the takeaways from previous sections to create a final CC-SMPC formulation for the single tank setup. The CC-MPC optimization program is formulated as a quadratic optimization program:

$$\mathcal{J} = \min_{u, S_{\text{ub}}, q_{of|k}} \sum_{k=0}^{H_p-1} Q(\mu_{h_k} - h_{ref|k})^2 + Q\Sigma_{h_k} + Ru_k^2 + R'\Delta u_k^2 + C_{\text{ub}} S_{\text{ub}|k} + P_k q_{of|k}, \tag{6.41}$$

$$s.t.$$

$$\mu_{h_{k+1}} = \mu_{h_k} + \phi(d_{k,mean} - u_k - q_{of|k}), \tag{6.42}$$

$$\Sigma_{h_{k+1}} = \Sigma_{h_k} + \phi^2\Sigma_d + \Sigma_w + \Sigma_v, \tag{6.43}$$

$$\underline{\mu_{h_k}} \leq \mu_{h_k} \leq \overline{\mu_{h_k}} + S_{\text{ub}|k} - \sqrt{\Sigma_{h_k}}\Phi^{-1}(\alpha), \tag{6.44}$$

$$\underline{u_k} \leq u_k \leq \overline{u_k}, \tag{6.45}$$

$$\underline{\Delta u_k} \leq \Delta u_k \leq \overline{\Delta u_k}, \tag{6.46}$$

$$\Delta u_k \triangleq k \mapsto u_{k+1} - u_k. \tag{6.47}$$

Equation (6.42) represents the mean system dynamics, in this case the mean tank dynamics. It is used for predicting the future states' values under expected (mean) disturbance. The uncertainty of the mean model prediction is described with the variance dynamics (6.43). In equations (6.44), (6.45) and (6.46) the system constraints are defined on the states $\mu_h$, the controller action $u$ and the change of the controller action $\Delta u$. In particular, the equation (6.44) defines a chance constraint on the upper level bound $\overline{mu_{h_k}}$. It accounts for an uncertainty of predictions by tightening the constraint on the level, see *subsection 6.2.1 Chance constraints* and *section 6.3 Single tank SMPC*. The objective function $\mathcal{J}$ is optimized for the controller action $u$, the chance constraint relaxation slack variable $S_{\text{ub}}$ and the overflow capturing slack variable $q_{of}$. Considering overflow prevention as a main requirement for the CC-MPC, the overall tuning recommendation is specified in (6.48) and (6.49).

$$P_k \gg C_{\text{ub}} \gg Q, \quad \forall k = 0, ..., H_p - 1, \tag{6.48}$$

$$\Delta P_k = -\mathcal{C}, \quad \forall k = 0, ..., H_p - 2, \quad \mathcal{C} \in \mathbb{R}^+, \tag{6.49}$$

$$\Delta P_k \triangleq k \mapsto P_{k+1} - P_k. \tag{6.50}$$

The cost $P_k$ on an overflow variable must be higher than the cost on other objectives. It is also necessary to make the cost $P$ a time-dependant cost for the overflow capturing to work properly. The cost $P_k$ is a non-constant function that decreases with $k$. According to (6.49) it is also a linear function. Note as well that the two slack variables enter the objective function linearly.

# Controlling the Two Tank Topology

7

This chapter presents a comparison of the CC-MPC and standard MPC control strategies implemented for the Two Tank Topology and tested in the AAU Smart Water Lab. The main focus of the comparison study is the minimization and prevention of overflow, in addition to outflow smoothing and equal use of tank storage. The control strategies represent the integration of all the methods that are presented in previous chapters.

The comparison is based on a set of lab experiments. We aim to design these lab experiments to emulate realistic scenarios in a sanitation network as close as possible. That includes the careful design and scaling of disturbances. The Water Lab configuration and scale are described in *subsection 7.1.2 Lab scaling*.

The chapter starts with a description of the control problem, as well as a description of the disturbance design. Thereafter a description is given of how the model, Kalman filter, MPC, and CC-MPC have been set up to handle the control problem. Lastly, a discussion of the experimental results is given.

## 7.1 The Control Problem

In the previous chapter 2 and chapter 4, methods for modeling and identifying sanitation networks have been presented. Both chapters have been concerned with the network topology we have dubbed the Two Tank Topology again shown in *Figure 7.1*.

The Two Tank Topology is comprised of two tanks that are collecting wastewater and the rain. The two tanks are connected with a gravity pipe used for the transport of wastewater. The gravity pipe is considered a passive element because the water volume pumped into it is only routed by gravity.

The system is subject to wastewater inflows which are introduced into the first tank and into the middle of the gravity pipe. The inflows can be described as a combination of the household discharge and the rain runoff.

When a sewer system has to rout both household discharge and rain water in a combined system, the likelihood of overflow is increased compared to when a sewer system uses separate pipes for the two. This is because wastewater treatment in a treatment plant is a biological process in which a bacterial culture has to be slowly grown. For the bacterial culture to grow, the treatment plant needs a slowly chancing inflow of wastewater [2]. When the two inflow types are combined into a single system, heavy rain events can cause the treatment plant to have a lack of capacity. This increases the likelihood of overflow as

untreated wastewater "backs up" the sewer system. Wastewater overflow is unwanted, as it can lead to major problems such as contamination of drinking water or damaging public and private property.

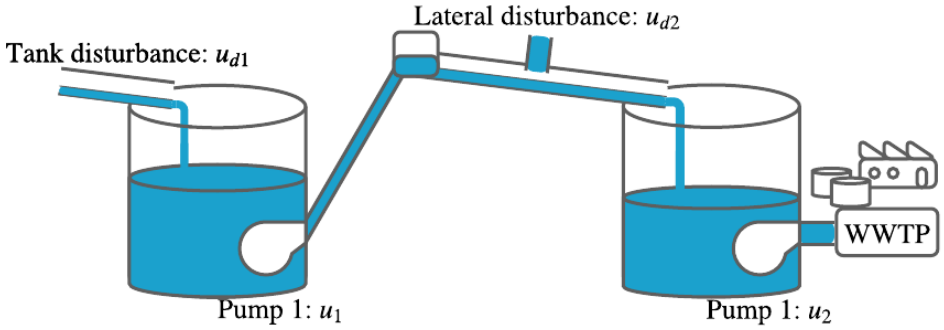



**Figure 7.1: An illustration of the Two Tank Topology**

The inflows of wastewater into a combined system can, from a control perspective, be considered as a disturbance. Thankfully weather forecasts are provided for real-world systems, and these can be used to "predict" the effect heavy rain will have on a system.

Based on just presented observations, we define two objectives for control of combined sewers, stated in prioritized order as:

1. Minimize the overflow in the network and
2. Provide constant flow towards a Waste Water Treatment Plant.

Overflow avoidance and minimization can be identified as the main objective for a controller.

We elect to consider the disturbances observed in a real-world combined sewer as the ones we want to emulate using the Two Tank Topology in the AAU Smart Water Lab. In order to state the control problem in full, we need a description of the inflow from household discharge and rain runoff for the lab. We have designed these descriptions for the control problem in the next subsection.

### 7.1.1 Disturbance design

In real-world sanitation networks, the household discharge has a relatively consistent, periodic shape with a period of 24 hours. Therefore a sufficiently good prediction of wastewater discharge can be made based on the historical data. One such set of data is shown in *Figure 7.2*, with a single day measurement outlined in red.

Rain inflow, on the other hand, is more complex to predict. Rain inflow depends on a lot of parameters, such as the shape and size of the catchment area and the rain intensity in that area. Rain intensity prediction can be provided by meteorological institutes [34]. However, identifying the dynamics that translate rain intensity to inflow in a sanitation network, called catchment dynamics, is a field of research in itself. We elect not to dive

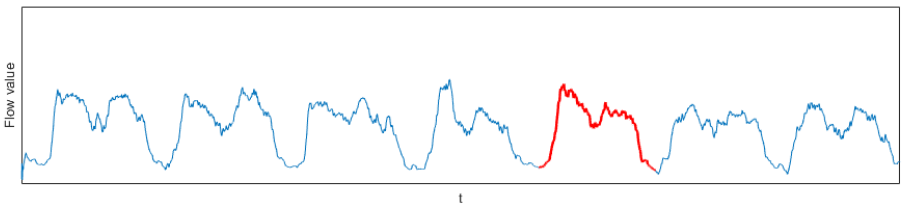



**Figure 7.2: Wastewater flow from residential areas for a period of 7 days.**

into the topic of catchment dynamics in this thesis. Rather we choose to design the inflows based on simple reasoning and assumptions about the rain disturbances.

Our approach to designing the inflow disturbances for the lab can be summarized in two steps:

- First, we design a time series forecast for each type of disturbance separately.
- Then, based on the original time series, we created a number of realizations for each of the two disturbance inflows. These realizations are also called ensembles.

If we were to get actual rain intensity forecasts provided by a meteorological institute, they would usually be provided as ensembles. The forecast ensembles would likely be provided along with a probability for each realization to occur.

### Disturbance due to household discharge

Household discharge ensembles are created using the data set that we already presented earlier as a starting point. From this starting point we elected to create nine more ensembles by altering the data sets basic shape. The ensembles are created by adding nine different realizations of the Gaussian random process $\mathbf{X} = [X_1, X_2, ..., X_N]^T$ of uncorrelated samples with zero mean $\mu_{\mathbf{X}} = 0$ and constant sample variance $\Sigma_{X_1} = \Sigma_{X_2} = ... = \Sigma_{X_N} = \Sigma_{\mathbf{X}}$, where N is the length of the ensemble data set. The variance of the random process $\mathbf{X}$ is calculated based on the average "daily" variance over 30 days of data extracted from data set in *Figure 7.2*. From there, to ensure a conservative prediction, the variance $\Sigma_{\mathbf{X}}$ is increased by a factor of two. Two arbitrary days of household discharge ensembles are shown in *Figure 7.3*.

*Figure 7.4* shows a seven day mean prediction with the uncertainty window, shown in grey, around it. It can be observed that the mean conserved the two peek daily shape and that the highest uncertainty is present around these peeks.

Notice that until now we did not specify the scale of the axis in **??** and *Figure 7.4*, but rather presented the shapes in general. In *Figure 7.4* we specify the scale on both axes as it is used in the lab experiment. The scale in *Figure 7.4* is introduced to give a general idea of the values of the uncertainty and the range of the signal. The scaling of the lab emulation is expanded upon in *subsection 7.1.2 Lab scaling*.

### Disturbance due to rain inflow

At the beginning of *subsection 7.1.1 Disturbance design* we stated that creating a rain inflow prediction is a challenging task. Unfortunately, unlike the wastewater disturbance

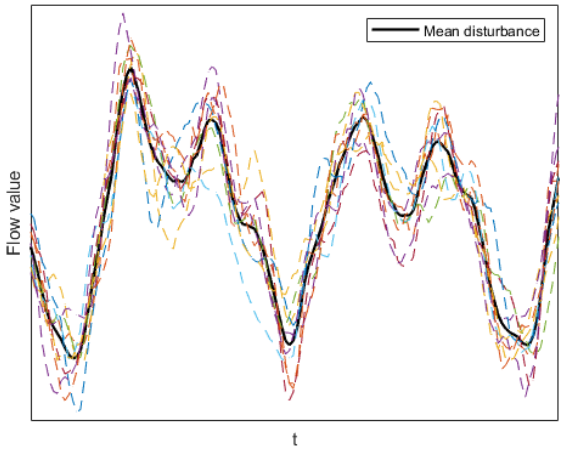



**Figure 7.3: Ensemble of 10 predictions for wastewater disturbance with the mean.**

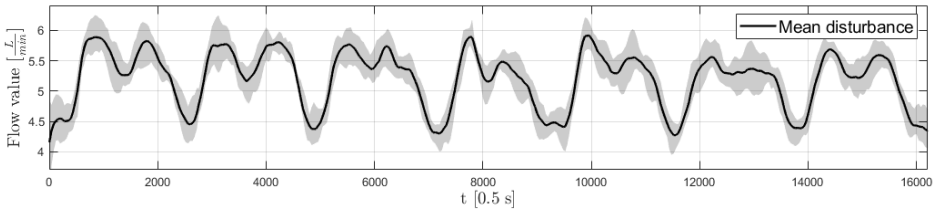



**Figure 7.4: The 7 days mean wastewater disturbance prediction with an uncertainty window.**

design, we do not have a historical measurement of the rain inflows. In case when these measurements are available, [35] proposed a method for forecasting a multi-step prediction of rain runoff.

For our application, we elect to design the rain disturbance inflow directly. The rain inflow is designed manually with prior assumptions on the shape of the storm event. For example, the storm event equates to the "sudden" increase of the inflow. In *Figure 7.5* ensemble realizations are presented for the rain inflow, together with a mean prediction of the rain inflow.

The *Figure 7.5* shows two storm events. Notice by observing the mean prediction together with two outlined realizations that a rain event does not assume any particular shape. Also, the length and intensity vary from one event to another, and so do the dry times in between. The rain predictions have more variation compared to the wastewater disturbance. The start and end time is fixed for all of the realizations. However, since the forecasts have different shapes, we declare that these shapes will have to mimic the uncertainty in a storm start time. Note that some forecasts peek immediately at the start, while for some others, it takes some time until they reach the peak intensity. Also, there are no completely dry weather predictions within the rain events. The rain and wastewater disturbances are combined and scaled for the lab setup. The complete, scaled version of the disturbance to the first tank is showed in *Figure 7.6*. The mean disturbance given to

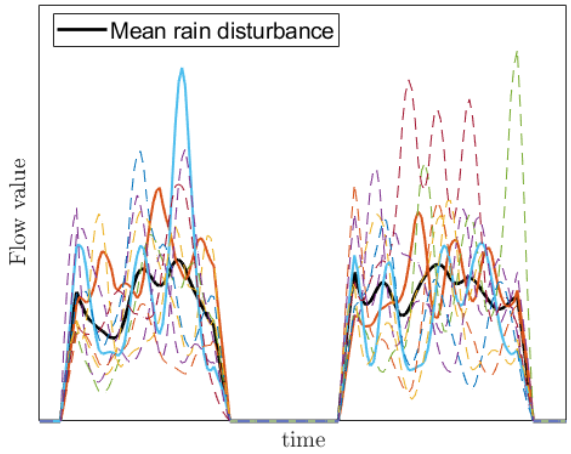



**Figure 7.5: Ensemble of 10 predictions for rain inflow disturbance with the mean prediction.**

the MPC is plotted in black, while the actual disturbance implemented in the experiments is plotted in red. The uncertainty window is also shown around the mean.

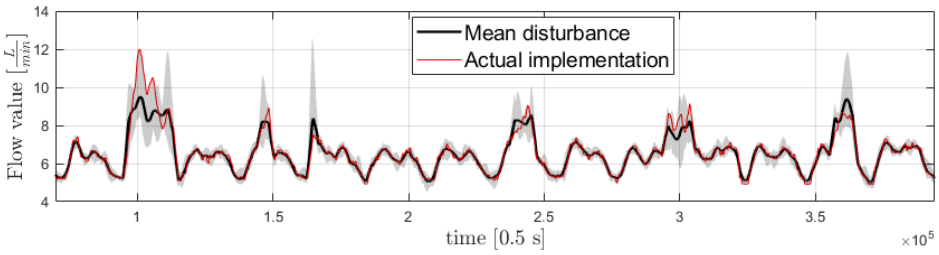



**Figure 7.6: The 14 days combined disturbance to the tank 1. Black is the prediction given to MPC, red is actual disturbance and grey overlay show the uncertainty of the prediction.**

In *Figure 7.7* a single dry day with only wastewater disturbance present is presented alongside the wet day where a rain event occurs. It can be seen that for those two particular days, the mean is very precise on the dry day, which is expected because the uncertainty is very small. On the wet day, the volume of the actual rain is smaller than the predicted volume. Based on that, we can expect the performance of the MPC and CC-MPC is affected by this difference in volume on the wet days. We do not expect to see a huge difference in performance between MPC and CC-MPC on dry days.

In summary, we have designed two disturbance sets comprised of wastewater from households and rain runoff. One disturbance enters the first tank, and the second disturbance enters in the middle of the pipe. The disturbances that are designed in this section are used in both of the experiments that follow. One out of 10 realizations is implemented on the system, and both controllers, MPC and CC-MPC, receive the same mean disturbance prediction. The CC-MPC additionally requires the variance of the disturbance prediction for the chance constraint implementation.

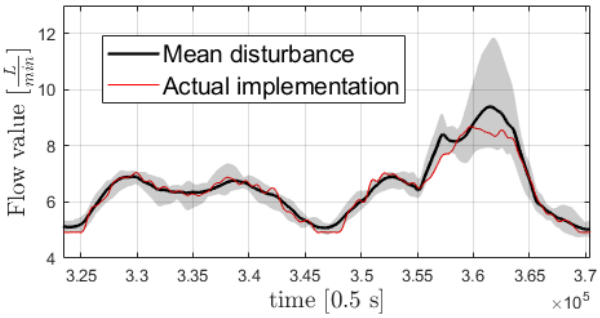



**Figure 7.7: Dry and wet day side by side comparison of prediction uncertainty. Dry day is from 3.25 to 3.455 and wet day is from 3.455 to 3.7.**

### 7.1.2 Lab scaling

In this subsection, we describe how scaling of the disturbances and pumps in the lab is conducted. Both flow values and time values are scaled. We specify the scale of the lab so that they compare to real-world time and dimensions. It is important to have some idea about the scale of the emulated system since in the disturbance design, we have a notion of a single day.

The sampling time in the lab is set to 0.5 s and the average flow that can be moved through the system in steady state is $8\frac{L}{min}$. Because the pumps in the lab cannot operate with low flows, which is described in *section 3.2 Emulation Setup - AAU Water Lab*, the lower bound on pumps is subtracted from the calculation of the effective volume moved in time.

With this in mind, knowing the lower bound is set to $4\frac{L}{min}$ we elect the lowest upper bound for a pump which is $8\frac{L}{min}$. This is done to have a conservative estimation and produces an effective flow range of $4\frac{L}{min}$. The $4\frac{L}{min}$ effective flow range means that our retention tank in the lab with $25L$ volume can be emptied in $\sim 6$ minutes, at the fastest.

We assume that a reasonable time for emptying a real-world retention tank is approximately one-third of the day. Based on that assumption, we chose the scale of 18 min in the lab to represent a single day, 24 hours. That means that in one simulated day the simulation advances 2160 steps. We are running the experiments for 12 real-world hours. With scaling, this amounts to $\sim 40$ days in lab time.

Another scaling consideration has to do with the pump bounds. We already mentioned the problem with lower bound on the possible flow operation, limiting the range of flows for possible disturbance implementations. For that reason, all the signals, including the disturbances presented in this section, have a constant offset. The disturbance intensities are scaled such that the wastewater component of the disturbance takes up 30% to 40% of the pumps operating range. The last 60% to 70% is reserved for the rain events. Based on that, pump 1 and pump 2 are bounded such that on dry days, where only wastewater disturbances are present, the tanks can be kept at a level reference. This way, the overflow can occur only due to rain events.

## 7.2 Solutions to the Control problem

Stating the control problem in general terms leads to the following objectives for a control system.

1. Minimize the overflow in the network and
2. Provide constant flow towards a Waste Water Treatment Plant.

We would, however like to extend these two objectives with a third:

3. Balance the wastewater along with the available storage.

This is to help further counteract the effect of the disturbance inflows resulting from heavy rain events. It is critical to develop a balancing strategy, such that the wastewater is distributed equally in the network if possible, even if there is no anticipated rain event in the near future. Otherwise, we expect the poor wastewater distribution can result in an undesirable spillage of the waste.

With the Two Tank Topology set up as shown in *Figure 7.1* these objectives directly translate to objectives for the standard MPC and the CC-MPC. The numerical values resulting from the weighted importance of each objective will be presented in the MPC and CC-MPC framework. This can be found in subsection 7.2.3 and subsection 7.2.4 of this chapter.

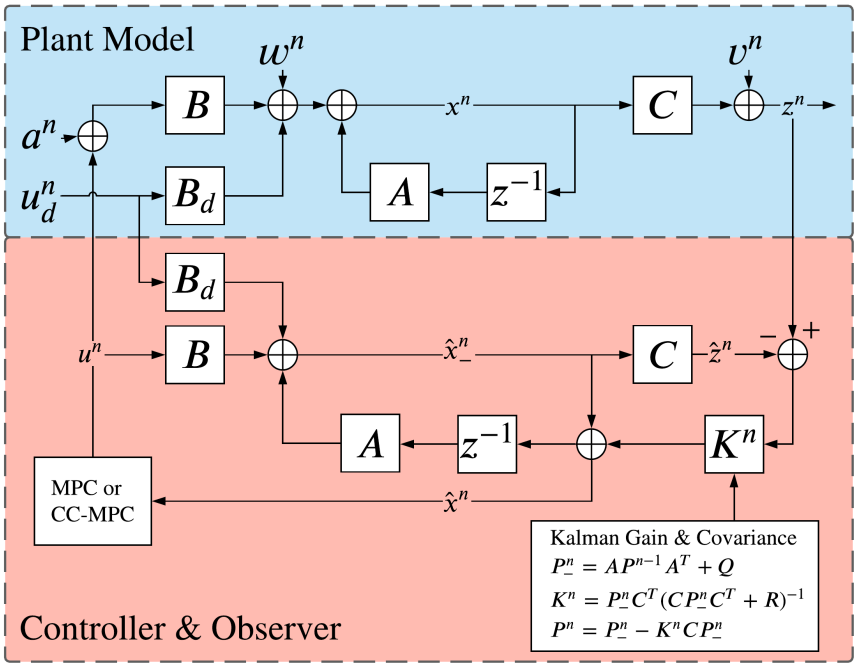



**Figure 7.8: The Control system stated as a block diagram.**

The MPCs do, however not solve the control problem by themselves, which is why we present the control setup for the Two Tank Topology, shown in *Figure 7.8*. The control system contains a plant model, an observer, and a controller. The plant is modeled using a linear stochastic model, the observer is implemented using a Kalman filter, and the controller used is either the MPC or the CC-MPC.

The theory and examples for each of these control system components have been presented in previous chapters. However, as they pertain to the coming tests, the specifics will be presented later in this section.

All but one of the variables shown in *Figure 7.8*, should be known to the reader, but they will quickly be recapped: $n$ is the time index. $\mathbf{A}$, $\mathbf{B}$, $\mathbf{B_d}$, and $\mathbf{C}$ are deterministic system matrices describing the Two Tank Topology, where $\mathbf{u}^n$ is the control variable. $\mathbf{w}^n$ and $\mathbf{v}^n$ are stochastic model and measurement uncertainty, $\mathbf{K}^n$ is Kalman gain, and $\mathbf{P}^n$ is error covariance. Lastly $\mathbf{u_d}^n$ is the disturbance inflow, or as seen later the predicted disturbance inflow, see *subsection 7.1.1 Disturbance design*. The one variable which has not been presented yet is the input uncertainty $\mathbf{a}^n$ which is an additive noise term added to the control signal.

The rest of this section is dedicated to describing how the system model and Kalman filter is set up to help solve the control problem. This is followed by a short description of how the control system is implemented in software on The AAU Smart Water Lab. The section ends with a description of the CC-MPC, and MPC.

### 7.2.1 The Model used in the solution

In *chapter 4 Model Parameter Identification*, a model of the gravity pipe in the Two Tank Topology with four discrete pipe sections was identified. Observing the results of that chapter, it can be seen that increasing the number of discrete pipe sections decreases the RMSE of the residuals found by comparing the model to measurements.

This would initially lead to the assumption that one should aim for a model with an arbitrarily high number of pipe sections. This is, however not the case. That is because whenever the model is used in an MPC framework, the computational load increases with an increased number of states. In both the MPC and Kalman filter, we choose to use a system model where the pipe is considered to have eight sections. This choice is made as we believe that eight pipe sections is a good compromise between complexity and precision.

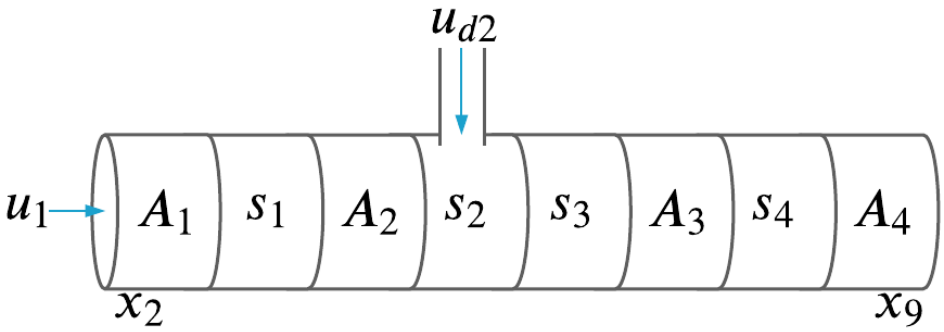



**Figure 7.9: Distribution of sensors and lateral inflow $u_{d2}$ along discrete sections of the gravity pipe. Where $S_\bullet$ denotes the sensed states and $A_\bullet$ denotes the augmented states.**

As mentioned in the introduction to the Kalman filter chapter, it is possible to extend the Two Tank Topology model to include lateral inflow in the gravity pipe. This is convenient as we stated the control problem as having exactly such an inflow. This leads to a system description which is an extension of (2.27), shown in (7.1).

$$\mathbf{x}^{n+1} = \mathbf{A}\mathbf{x}^n + \mathbf{B}\mathbf{u}^n + \mathbf{B_{of}}\mathbf{u_{of}}^n + \mathbf{B_d}\mathbf{u_d}^n + \mathbf{\Delta}. \tag{7.1}$$

Here the state vector $\mathbf{x}^n \in \mathbb{R}^{10\times 1}$, the system matrix $\mathbf{A} \in \mathbb{R}^{10\times 10}$, the input matrices $\mathbf{B},\ \mathbf{B}_{of},\ \mathbf{B}_d \in \mathbb{R}^{10\times 2}$, while the control inputs $\mathbf{u},\ \mathbf{u}_{of} \in \mathbb{R}^{2\times 1}$, and disturbance $\mathbf{u}_d \in \mathbb{R}^{2\times 1}$. Note that (7.1) deviates from (2.27). By having the input matrix $\mathbf{B}$ split into two matrices $\mathbf{B}$ and $\mathbf{B}_{of}$. The overflow control input and overflow input matrix is shown in (7.2).

$$\mathbf{B_{of}} \cdot \mathbf{u_{of}}_{|k} = \begin{bmatrix} \phi_1\Delta t & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \theta_1\Delta t \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_{of1|k} \\ u_{of2|k} \end{bmatrix}. \tag{7.2}$$

As the lab only contains four level sensors in the gravity pipe, see *chapter 3 Simulation- and Emulation-Setup*, and as the pipe model now includes eight states, we have to choose the arrangement of the sensors. We chose the arrangement shown in *Figure 7.9*. In *section 8.1 System Identification* a discussion of the choices sensor locations and inflow location is presented.

The increase in pipe states from four to eight, requires a new description of the output matrix $C \in \mathbb{R}^{6\times 10}$. Knowing that we can measure the tank levels as well as four pipe levels yields the output equation shown in (7.3).

$$\mathbf{z}^n = \mathbf{C}\mathbf{x}^n = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h_{T1}^n \\ h_{P1}^n \\ \vdots \\ h_{P8}^n \\ h_{T2}^n \end{bmatrix}. \tag{7.3}$$

With all these new considerations in mind, an updated version of the identification process was performed. This yielded the parameters presented in table 7.1 and a visual comparison of measurements and a free-running model, using the parameters is shown in *Figure 7.10*.

| $\theta_1$ | $\theta_2$ | $\theta_3$ | $\theta_4$ | $\theta_5$ | $\phi_1$ | $\phi_2$ |
|---|---|---|---|---|---|---|
| 0.0578 | 0.1378 | $1.000 \cdot 10^{-4}$ | -0.0051 | 0.1000 | 0.2037 | 0.2037 |

**Table 7.1: The resulting parameters from the system identification of the eight state pipe model.**

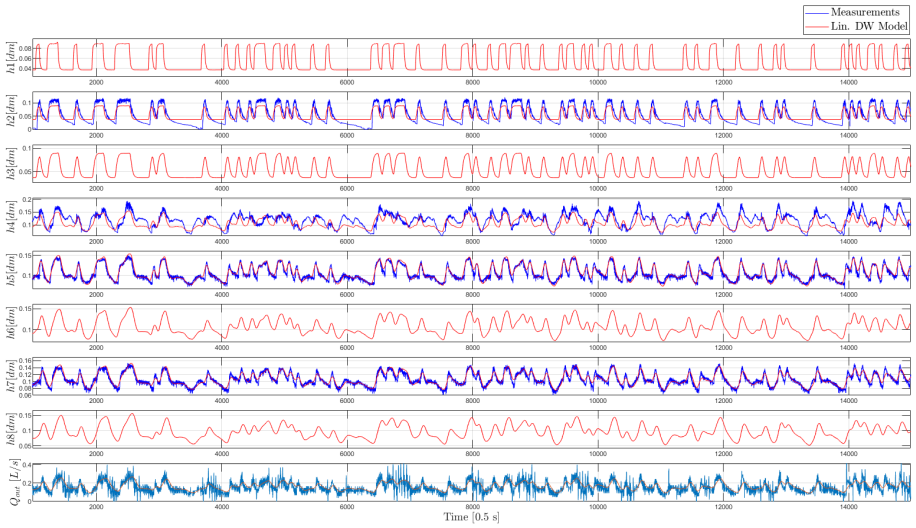



**Figure 7.10: Results of parameter identification with augmented model.**

In *subsection 2.2.1 Stability of the General Model* two sufficient conditions for stability were found. The conditions are again stated here (7.4) and (7.5) along with the assumptions on the $\theta$ parameters in (7.6). It can be seen that sufficient condition 1 is not upheld.

$$\text{Sufficient Condition 1:} \tag{7.4}$$
$$\theta_5 + \theta_3 \geq \theta_2,$$
$$\theta_2 \geq \theta_3,$$
$$\text{And Sufficient Condition 2:} \tag{7.5}$$
$$\theta_5 \Delta t \leq 1,$$
$$\text{Assuming:} \tag{7.6}$$
$$(\theta_2 + \theta_3)\Delta t < 1,$$
$$(\theta_5 + \theta_3)\Delta t < 1,$$
$$\theta_1, \theta_2, \theta_3, \theta_5 > 0.$$

Therefore we have to make sure that the second sufficient condition is. This is done by selecting $\Delta t$ such that it adheres to the inequalities specified in (7.5) and (7.6).

$$\Delta t < \frac{1}{\theta_5} = 10s, \tag{7.7}$$
$$\Delta t < \frac{1}{(\theta_2 + \theta_3)} = 7.2499s, \tag{7.8}$$
$$\Delta t < \frac{1}{(\theta_5 + \theta_3)} = 9.99s. \tag{7.9}$$

As $7.2499s$ ends up being the upper bound on $\Delta t$, we have to choose one below. We elect $\Delta t = 5$ as this makes future calculations for the MPCs easier. In the next section, we

present a method for extending the results found in chapter 5 to the new system model, which now includes eight states.

### 7.2.2 The state estimator used in the solution

In *chapter 5 State estimator* we obtained a description of the measurement noise and model noise present in the AAU Smart Water Lab when a four pipe state model is used. We furthermore chose to implement a Kalman filter for the pipe states rather than for every state.

In this section we continue with this approach. The general model including stochastic noise is shown in equations (7.10a) and (7.10b).

$$\mathbf{x}^n = \mathbf{A}\mathbf{x}^{n-1} + \mathbf{B}\mathbf{u}^n + \mathbf{B_d}\mathbf{u_d}^n + \mathbf{\Delta} + \mathbf{w}^n \tag{7.10a}$$

$$\mathbf{z}^n = \mathbf{C}\mathbf{x}^n + \mathbf{v}^n, \tag{7.10b}$$

Where for the four pipe state model $\mathbf{C} = \mathbf{I} \in \mathbb{R}^{6\times 6}$. Extending the model from four to eight pipe states, the measurement noise should remain unchanged while the model noise should not. This section will therefore be used to develop a covariance $\mathbf{Q}$ describing the model noise for the eight pipe state model. Developing this covariance $\mathbf{Q}$ can, however not be done using the same approach as in chapter 5. This is simply because the before-mentioned approach relies on having measurements available for all states.

Therefore, a creative solution is needed, and we choose to explore two empirical approaches that take a starting point in the model noises from the four state model. From chapter 5 we have the covariance matrices shown in (7.11) describing the model noise relation between pipe states, in the four state model.

$$\mathbf{Q} = \begin{bmatrix} 0.1855 & -0.0425 & 0.0483 & -0.0121 \\ -0.0425 & 0.2415 & -0.0141 & 0.0124 \\ 0.0483 & -0.0141 & 0.1231 & 0.0314 \\ -0.0121 & 0.0124 & 0.0314 & 0.0427 \end{bmatrix} \cdot 10^{-5}, \tag{7.11}$$

The two approaches for developing a covariance matrix $\mathbf{Q}$ for the eight state model are based on two different assumptions:

- The first approach simply acknowledges that we have a sparsity of information in regards to the model noise. Therefore leaving the entries in the covariance matrix where no data is available as equal to zero. The sparse covariance matrix is shown as $\mathbf{Q_s}$ in (7.12).
- The second approach relies on the physical intuition of the pipe. Each state in the pipe can roughly be described as a time-delayed version of the previous pipe state. Therefore, it is not a far stretch to assume that the model noise between two adjacent pipe sections is similar. If we, therefore, populate the entries in the covariance matrix where no information is yet obtained, with the noise information from an adjacent pipe section, then we obtain $\mathbf{Q_f}$ which is shown in (7.13).

As the two matrices have been created empirically, it is prudent to check if they satisfy the requirements for covariance matrices. We know that covariance matrices should be symmetric and positive semi-definite.

$$\mathbf{Q_s} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1855 & 0 & -0.0425 & 0.0483 & 0 & -0.0121 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.0425 & 0 & 0.2415 & -0.0141 & 0 & 0.0124 & 0 \\ 0 & 0.0483 & 0 & -0.0141 & 0.1231 & 0 & 0.0314 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.0121 & 0 & 0.0124 & 0.0314 & 0 & 0.0427 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot 10^{-5}. \tag{7.12}$$

$$\mathbf{Q_f} = \begin{bmatrix} 0.1855 & 0.1855 & -0.0425 & -0.0425 & 0.0483 & 0.0483 & -0.0121 & -0.0121 \\ 0.1855 & 0.1855 & -0.0425 & -0.0425 & 0.0483 & 0.0483 & -0.0121 & -0.0121 \\ -0.0425 & -0.0425 & 0.2415 & 0.2415 & -0.0141 & -0.0141 & 0.0124 & 0.0124 \\ -0.0425 & -0.0425 & 0.2415 & 0.2415 & -0.0141 & -0.0141 & 0.0124 & 0.0124 \\ 0.0483 & 0.0483 & -0.0141 & -0.0141 & 0.1231 & 0.1231 & 0.0314 & 0.0314 \\ 0.0483 & 0.0483 & -0.0141 & -0.0141 & 0.1231 & 0.1231 & 0.0314 & 0.0314 \\ -0.0121 & -0.0121 & 0.0124 & 0.0124 & 0.0314 & 0.0314 & 0.0427 & 0.0427 \\ -0.0121 & -0.0121 & 0.0124 & 0.0124 & 0.0314 & 0.0314 & 0.0427 & 0.0427 \end{bmatrix} \cdot 10^{-5}. \tag{7.13}$$

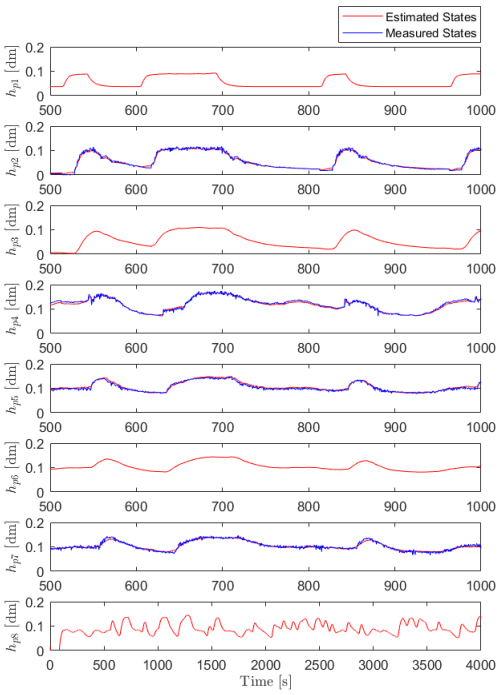
Both $\mathbf{Q_f}$ and $\mathbf{Q_s}$ can be confirmed to be symmetric by simple visual inspection. However, checking for positive semi-definiteness can be done in multiple ways, such as Cholesky-factorization or inspection of the Eigenvalues. If we observe the eigenvalues:

$$\sigma(\mathbf{Q_s}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 9.4 \cdot 10^{-24} \\ 2.6 \cdot 10^{-07} \\ 1.1 \cdot 10^{-06} \\ 1.8 \cdot 10^{-06} \\ 2.7 \cdot 10^{-06} \end{bmatrix}, \quad \sigma(\mathbf{Q_f}) = \begin{bmatrix} -7.3 \cdot 10^{-22} \\ 3.8 \cdot 10^{-38} \\ 1.0 \cdot 10^{-22} \\ 5.3 \cdot 10^{-22} \\ 5.3 \cdot 10^{-07} \\ 2.2 \cdot 10^{-06} \\ 3.6 \cdot 10^{-06} \\ 5.5 \cdot 10^{-06} \end{bmatrix}.$$

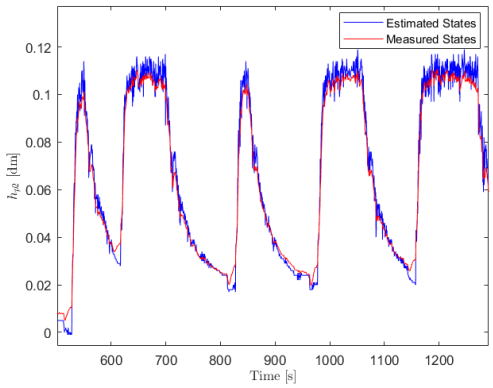
It can be observed that one of the eigenvalues of $\mathbf{Q_f}$ is negative; therefore, it cannot be a covariance matrix. We, therefore, elect to use $\mathbf{Q_s}$ as the covariance matrix for the model noise in the eight pipe state Kalman filter.

The Kalman filters ability to estimate the state can be seen in *Figure 7.11a* and *Figure 7.11b*. The measured states seem to be low pass filtered, while non-measured states seem to behave as we would expect by acting as delayed versions of the measured pipe states.

Having found an adequate covariance matrix describing the model noise $\mathbf{W}$, we proceed with a presentation of the of the MPC followed by a presentation of the CC-MPC. These are the controllers which will be used to solve the control problem.




(a) Kalman state estimates for the eight pipe sections.




(b) The filtering effect exhibited by the Kalman filter on pipe state two $h_{p2}$.

**Figure 7.11: The effect of the Kalman filter on the eight pipe state model**

### 7.2.3 The Model Predictive controller used in a solution

In this section, our realization of a standard MPC is presented. This controller is used in conjunction with the model and Kalman filter presented in this section to provide part of the results presented in *section 7.3 Results*. The structure of the MPC is identical to the structure presented in *section 6.1 Model Predictive Control*, except that it features the ability to handle overflow using the control variable $\mathbf{u_{of}}$.

The MPC formulation is presented as a cost function $\mathcal{J}$ and as a set of constraints:

$$\mathcal{J} = \min_{\mathbf{u},\mathbf{u_{of}}} \sum_{k=1}^{H_p} \left\| \mathbf{z}_k - \mathbf{z}_k^{ref} \right\|_{\mathbf{Q}}^2 + \left\| \mathbf{u_{of}}_{|k} \right\|_{\mathbf{P}_k} + \sum_{k=0}^{Hu-1} \|\mathbf{\Delta u}_k\|_{\mathbf{R}}^2 \tag{7.14}$$

s.t

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{B_d}\mathbf{u_d}_{|k} + \mathbf{B_{of}}\mathbf{u_{of}}_{|k}, \tag{7.15}$$

$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k, \tag{7.16}$$

$$\mathbf{\Delta u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}, \tag{7.17}$$

$$\mathbf{u_{of}}_{|k} \geq 0. \tag{7.18}$$

$$Hu = Hp \tag{7.19}$$

$$\underline{\mathbf{x}} \leq \mathbf{x}_k \leq \overline{\mathbf{x}}, \tag{7.20}$$

$$\underline{\mathbf{u}} \leq \mathbf{u}_k \leq \overline{\mathbf{u}}, \tag{7.21}$$

$$\underline{\mathbf{\Delta u}} \leq \mathbf{\Delta u}_k \leq \overline{\mathbf{\Delta u}}. \tag{7.22}$$

Where the system model matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{B_d}$, and $\mathbf{C}$ are presented as the eight pipe state model previously in this section. The bounds for $\mathbf{x}$ and $\mathbf{u}$ come from the physical limitations of the lab, with the exception of $\underline{\mathbf{x}}$. The lower bound for the pipe states are

decreased from 0 to $-0.5$ in order to avoid problems with feasibility. The bounds are listed below:

$$\underline{\mathbf{x}} = \begin{bmatrix} 1.5 \\ -0.5 \\ \vdots \\ -0.5 \\ 1.5 \end{bmatrix} [dm], \quad \overline{\mathbf{x}} = \begin{bmatrix} 6.99 \\ 0.5 \\ \vdots \\ 0.5 \\ 6.50 \end{bmatrix} [dm], \quad \underline{\mathbf{u}} = \begin{bmatrix} 0.057 \\ 0.1 \end{bmatrix} \begin{bmatrix} L \\ \overline{s} \end{bmatrix}, \quad \overline{\mathbf{u}} = \begin{bmatrix} 0.13 \\ 0.22 \end{bmatrix} \begin{bmatrix} L \\ \overline{s} \end{bmatrix},$$

$$\underline{\mathbf{\Delta u}} = \begin{bmatrix} 0.0083 \\ 0.0011 \end{bmatrix} \begin{bmatrix} L \\ \overline{s} \end{bmatrix}, \quad \overline{\mathbf{\Delta u}} = \begin{bmatrix} -0.0083 \\ -0.0011 \end{bmatrix} \begin{bmatrix} L \\ \overline{s} \end{bmatrix}.$$

The values for the bounds on $\mathbf{\Delta u}$ as opposed to the bounds on $\mathbf{u}$ and $\mathbf{x}$ are chosen as part of the MPC design. The upper and lower bound on the slew rate of pump 2 $\mathbf{\Delta u_2}$ is chosen to make large flow variations into the treatment plant impossible. The number 0.0011 is calculated to allow the pump to turn fully on from the lower bound $\underline{\mathbf{u_2}}$ to the upper bound $\overline{\mathbf{u_2}}$ in 9 minutes in lab or 12 hours in real-life. The upper and lower bound for the slew rate of pump 1 $\mathbf{\Delta u_1}$ was selected empirically.

The three objectives presented in the *section 7.2 Solutions to the Control problem*, have been included in the MPC using the three terms in the cost function (7.14). The output minus reference term $\mathbf{z}_k - \mathbf{z^{ref}}_k$ is meant to ensure balanced storage, and the $\mathbf{\Delta u}_k$ is meant to penalize changing outflows from tank 2, in order to provide smooth outflows to a treatment plant. Lastly, overflow handling is implemented similarly to the method presented in subsection 6.3.3, where expected overflow is removed from the model predictions using the $\mathbf{B_{of}u_{of}}_{|k}$-term. The use of $\mathbf{u_{of}}_{|k}$ is incentivised using the decreasing cost matrix $\mathbf{P}_k$, as shown in (7.23).

The cost matrices are selected such that $\mathbf{Q}$ and $\mathbf{R}$ remain significantly lower than $\mathbf{P}_k$ as to ensure the correct function of overflow prediction and control $\mathbf{u_{of}}_{|k}$. Secondly, focus is placed on smoothing the outflow to the treatment plant. This smoothing is weighted such that there is a relation between $\mathbf{Q}$ and $\mathbf{R}$ entries is 1/5. This results in the cost matrices shown in (7.23) and (7.24).

$$\mathbf{P}_k = \begin{bmatrix} 1000 \cdot (Hp - k) & 0 \\ 0 & 1000 \cdot (Hp - k) \end{bmatrix}, \tag{7.23}$$

$$\mathbf{Q} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 0 & 0 \\ 0 & 50 \end{bmatrix}. \tag{7.24}$$

In *subsection 7.1.2 Lab scaling* we mentioned that 18 minutes of real time is scaled as to correspond to 1 day simulation time. With this in mind a suitable prediction horizon $Hp$ should be picked along with a suitable model time step $\Delta t$ - Note from *subsection 7.2.1 The Model used in the solution* that $\Delta t$ is the parameter in $\mathbf{A}$ and $\mathbf{B}$.

Knowing that the value $\Delta t = 5s$ leaves us to pick the prediction horizon $Hp$. Ideally, we would have as long of a prediction horizon as possible; there is, however a piratical upper limit, stemming from the maximum computation time, see *subsection 7.2.6 The Control System Software Setup.*

When selecting $Hp$ we choose to rely on two values from the review paper: [1], which we verified empirically during the commissioning of the lab and during Ad-Hoc testing. From the paper and testing, we determined that predictions of no less than half a day should be used when trying to smooth the outflow from tank 2. While predictions of 3 hours are sufficient for overflow prevention. Note this is for "real-world systems," so the values have to be scaled as explained in subsection 7.1.2.

As one of the objectives is to smoothen the outflow as well as prevent the overflow, we elect to go with the 12 hour prediction. This leads us to the following calculation for the number of prediction steps in the horizon $Hp$.

$$\begin{aligned} Hp &= \frac{(\text{second per simulation day}) \cdot days}{\Delta t} \\ &\Downarrow \\ Hp &= \frac{(18 \cdot 60) \cdot \frac{1}{2}}{5} = 108 \end{aligned} \tag{7.25}$$

In the next section a framework for the CC-MPC is presented. CC-MPC will in conjunction with the model and Kalman filter be used to further develop the results in section 7.3.

### 7.2.4 The Chance Constrained Model Predictive controller used in a second solution

The CC-MPC shares a large part of its description with the MPC presented in the previous section. The nominal controller structure of CC-MPC, which does not consider uncertainties, has a description identical to the MPC. This includes the control objectives, main dynamics, and the design parameters, such as the prediction horizon $Hp$ and the cost matrices $\mathbf{Q}$,$\mathbf{R}$,$\mathbf{P}$. We do, however, extend the nominal formulation of the CC-MPC, such that we take three kinds of uncertainty into account: the model uncertainty $\mathbf{V}_k$, the measurement uncertainty $\mathbf{W}$ and the input uncertainty $\mathbf{A}$.

With this in mind our description of the CC-MPC optimization program used to solve the MPC problem is summarized below. The cost function of the CC-MPC is shown in (7.26).

$$\begin{aligned} \mathcal{J} = \min_{\mathbf{u},\mathbf{u_{of}},\mathbf{S_{ub}}} & \sum_{k=0}^{Hp-1} \left\| \mu_{\mathbf{z}|k} - \mathbf{z}_k^{ref} \right\|_{\mathbf{Q}}^2 + tr(\mathbf{Q}\mathbf{\Sigma}_{\mathbf{z}|k}) + \left\| \mathbf{u_{of}}_{|k} \right\|_{\mathbf{P}_k} + \left\| \mathbf{S_{ub}}_{|k} \right\|_{\mathbf{O}} \\ & + \sum_{k=0}^{Hu-1} \left\| \mu_{\mathbf{\Delta u}_k} \right\|_{\mathbf{R}}^2 + tr(\mathbf{R}\mathbf{\Sigma}_{\mathbf{\Delta\Upsilon|k}}) \end{aligned} \tag{7.26}$$

The cost function now include three additional terms: $tr(\mathbf{R}\mathbf{\Sigma}_{\mathbf{z|k}})$, $tr(\mathbf{R}\mathbf{\Sigma}_{\mathbf{\Delta\Upsilon|k}})$ and $\left\| \mathbf{S_{ub|O}} \right\|$. The two *trace* terms are a consequence of having to express the the cost function as expected cost. This way of expressing the cost function is introduced in subsection 6.3.4.

The optimization variable $\mathbf{S_{ub|k}}$ is introduced for the CC-MPC setup, in *subsection 6.3.1 Feasibility concerns.* $\mathbf{S_{ub}}$ in an optimization variable that is used to cancel the effects of the chance constraints in the case when chance constraints result in an unfeasible optimization problem. The cost matrix $\mathbf{O}$ in $\left\|\mathbf{S_{ub}}_{|k}\right\|_{\mathbf{O}}$ is selected such that it is bigger than $\mathbf{R}$ and $\mathbf{Q}$ but still much smaller than the last entry of $\mathbf{P}_k$.

The equality constraints and variance dynamics of the CC-MPC are summarized below.

$$\mu_{\mathbf{x}k+1} = \mathbf{A}\mu_{\mathbf{x}|k} + \mathbf{B}\mu_{\mathbf{u}k} + \mathbf{B_d}\mu_{\mathbf{ud}|k}, \tag{7.27}$$

$$\mu_{\mathbf{z}k} = \mathbf{C}\mu_{\mathbf{x}}, \tag{7.28}$$

$$\mu_{\mathbf{\Delta u}|k} = \mu_{\mathbf{u}|k} - \mu_{\mathbf{u}|k-1}, \tag{7.29}$$

$$\mathbf{\Sigma_{x}}_{|k+1} = (\mathbf{A}-\mathbf{BK})\mathbf{\Sigma_{x}}_{|k}(\mathbf{A}-\mathbf{BK})^T + \mathbf{B\Sigma_a B}^T + \mathbf{B_d\Sigma_d B_d}^T + \mathbf{\Sigma_w}, \tag{7.30}$$

$$\mathbf{\Sigma_{z}}_{|k} = \mathbf{C\Sigma_{x}}_{|k}\mathbf{C}^T + \mathbf{\Sigma_{v}}_{|k}, \tag{7.31}$$

$$\mathbf{\Sigma_\Upsilon} = K\mathbf{\Sigma_{x}}_k K^T + \mathbf{\Sigma_a}, \tag{7.32}$$

$$\begin{aligned}\mathbf{\Sigma_{\Delta\Upsilon}} = \mathbf{K}[&(\mathbf{A}+\mathbf{BK}-\mathbf{I})\mathbf{\Sigma_x}(\mathbf{A}+\mathbf{BK}-\mathbf{I})^T \\ &+ \mathbf{\Sigma_w} + \mathbf{B\Sigma_a B}^T + \mathbf{B}_d\mathbf{\Sigma_d B}_d^T]\mathbf{K}^T + 4\mathbf{\Sigma_a},\end{aligned} \tag{7.33}$$

$$\mathbf{\Upsilon}_k = \mathbf{u}_k + \upsilon_k. \tag{7.34}$$

The system model matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{B_d}$, and $\mathbf{C}$ are the same as the ones presented for the MPC. The variance dynamics are used to define the chance constraints for the tank states, as in 6.3.4.

The chance constraints for the tank states and for the control action are shown below. The deterministic bounds on the expected state $\mu_{\mathbf{x}}$, input $\mu_{\mathbf{u}}$ and slew rate $\mu_{\mathbf{\Delta u}}$ are the same as in the MPC formulation.

$$\mathbf{h}_j^T\underline{\mu_{\mathbf{x}}} \le \mathbf{h}_j^T\mu_{\mathbf{x}|k} \le \mathbf{h}_j^T\bar{\mathbf{x}} + \mathbf{h}_j^T\mathbf{S_{ub}} - \sqrt{\mathbf{h}_j^T\mathbf{\Sigma_{x}}_{|k}\mathbf{h}_j}\phi^{-1}(0.95) \quad \text{for } j \in \{1, 10\}, \tag{7.35}$$

$$\mathbf{h}_j^T\mathbf{S_{ub}} \le \sqrt{\mathbf{h}_j^T\mathbf{\Sigma_{x}}_{|k}\mathbf{h}_j}\phi^{-1}(0.95) \tag{7.36}$$

$$\underline{\mu_{\mathbf{x}}} \le \mu_{\mathbf{x}|k} \le \overline{\mu_{\mathbf{x}}}, \tag{7.37}$$

$$\underline{\mu_{\mathbf{u}}} \le \mu_{\mathbf{u}k} \le \overline{\mu_{\mathbf{u}}} - \sqrt{\mathbf{q}_j^T\mathbf{\Sigma_\Upsilon}_{|k}\mathbf{q}_j}\phi^{-1}(0.95) \quad \text{for } j \in \{1, 2\}, \tag{7.38}$$

$$\underline{\mu_{\mathbf{\Delta u}}} \le \mu_{\mathbf{\Delta u}|k} \le \overline{\mu_{\mathbf{\Delta u}}}, \tag{7.39}$$

where the $\mathbf{h}_j$ represent the selector matrix that select the tank states in (7.35). Likewise, the $\mathbf{q}_j$ matrix is the selector matrix for control variables in (7.38).

Notice that in the chance constraint formulation, the square root function is used to define the tightening of the deterministic constraints. An important implementation detail is that the variance dynamics can be precalculated before every CC-MPC iteration. That is because the variance dynamics cannot be affected by the optimal policy $\mathbf{u}$. Consequently, the terms $\sqrt{\mathbf{h}_j^T\mathbf{\Sigma_\Upsilon}_{|k}\mathbf{h}_j}\phi^{-1}(0.95)$ and $\sqrt{\mathbf{h}_j^T\mathbf{\Sigma_{x}}_{|k}\mathbf{h}_j}\phi^{-1}(0.95)$ get predetermined values before the CC-MPC optimization problem is computed. The input disturbance variance $\mathbf{\Sigma_d}$ is different for every iteration of the CC-MPC problem.

Notice that we employ the tube controller, introduced in 6.2, in order to settle the state variance dynamics propagation in (7.30). The tube controller is a state feedback controller, which makes its contribution a random variable. We have to make sure that the tube

controller policy can be implemented on top of the CC-MPC's policy. Therefore, we have to expand the CC-MPC's control action constraint to form a chance constraint.

The tube controller is implemented as an LQR controller and tuned in accordance with Bryson's rule [25]. Bryson's rule is shown in equations (7.40) and (7.41).

$$Q_{LQR|ii} = \frac{1}{(\text{maximum acceptable value of } x_i - \mu_{x|i})^2}, \quad \text{for } i \in \{1, 2, ..., 6\} \tag{7.40}$$

$$R_{LQR|jj} = \frac{1}{(\text{maximum acceptable value of } \upsilon_i)^2}, \quad \text{for } j \in \{1, 2\} \tag{7.41}$$

We selected the maximum allowed flow for the LQR to be 0.5 $\frac{L}{min}$, while the maximum deviation of the state from the mean is set to 0.1 $dm$. The computed $\mathbf{K}$ matrix is show in (7.42).

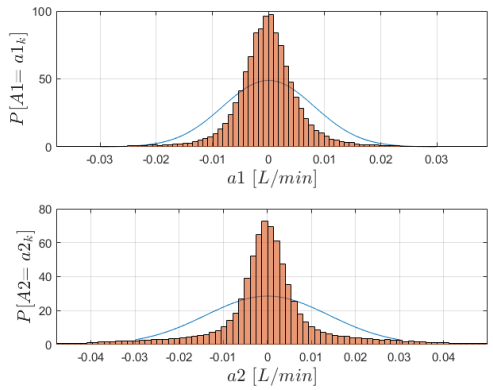
$$\mathbf{K} = \begin{bmatrix} -7.8 & 5.2 & 5.5 & 5.5 & 5.4 & 5.1 & 4.9 & 4.7 & 4.6 & 1.3 \\ -1.4 & -10.5 & -12.0 & -13.7 & -15.5 & -17.6 & -20.0 & -22.6 & -25.6 & -07.9 \end{bmatrix} \cdot 10^{-2}. \tag{7.42}$$

To express the variance dynamics, we need a description of the variances for the input noise $\mathbf{\Sigma_a}$, the model noise $\mathbf{\Sigma_w}$, the sensor noise $\mathbf{\Sigma_v}$ and the variance of the forecast $\mathbf{\Sigma_d}$. We have already identified the variance of the two noise terms $\mathbf{\Sigma_w} = \mathbf{Q}$ and $\mathbf{\Sigma_v} = \mathbf{R}$ as part of the Kalman filter design. In the next subsection we find expressions for the two remaining terms.

### 7.2.5 Calculation of input and forecast uncertainty

We know that the distribution of the input noise $A$ can be found by observing the difference between the actual inputs and the input reference for the pumps. This difference results in the residual distribution shown in *Figure 7.12a*.

Fitting a zero mean normal distribution to it we get the value for the covariance matrix $\mathbf{\Sigma_a}$ shown in *Figure 7.12a*.




**(a) Distributions of input uncertainty $a_n$ for each pump**

$$\mathbf{\Sigma_a} = \begin{bmatrix} 0.0666 & 0 \\ 0 & 0.1955 \end{bmatrix} \cdot 10^{-3}. \tag{7.43}$$

Let's look at the definition of chance constraints in our SMPC constraint formulation (7.35). We notice that the deterministic upper bound is tightened with a term that is dependant on the variance of states $\mathbf{\Sigma}_{\mathbf{x}k}$ at each time step $k$. We already presented how the tightening increases due to variance propagation dynamics. In (7.30), even for stable $(\mathbf{A} - \mathbf{BK})$ the variance increases to a certain point because of the additive uncertainties from other components in the system, like $\mathbf{\Sigma}_a$, $\mathbf{\Sigma}_w$ and $\mathbf{\Sigma}_d$. If we assume that the disturbance uncertainty $\mathbf{\Sigma}_d$ is constant for the entire prediction horizon, then the variance on tank states will eventually converge.

If we look back to the disturbance prediction in *Figure 7.6*, we can see that the uncertainty of the prediction changes with a resolution of 0.5 $s$. However, we elected to run the controller with the constant uncertainty of disturbance forecast for the entire prediction horizon. Therefore, on each MPC iteration, the $\mathbf{\Sigma}_d$ is recalculated by averaging the uncertainty of the disturbance forecast, $H_p$ steps in the future.

### 7.2.6 The Control System Software Setup

In *chapter 3 Simulation- and Emulation-Setup* we outlined the hardware and software setup used in the AAU Smart Water Lab. Now having explained the setup of the MPCs, we outline the computational constraints.

In *Figure 7.13* the timing of the software running on the Central Control Unit (CCU) is shown. From chapter 3 we see that the CCU is divided into two entities: the "The sensor/actuator - Client" setup in Simulink and the "MPC - Server" setup in Matlab. The

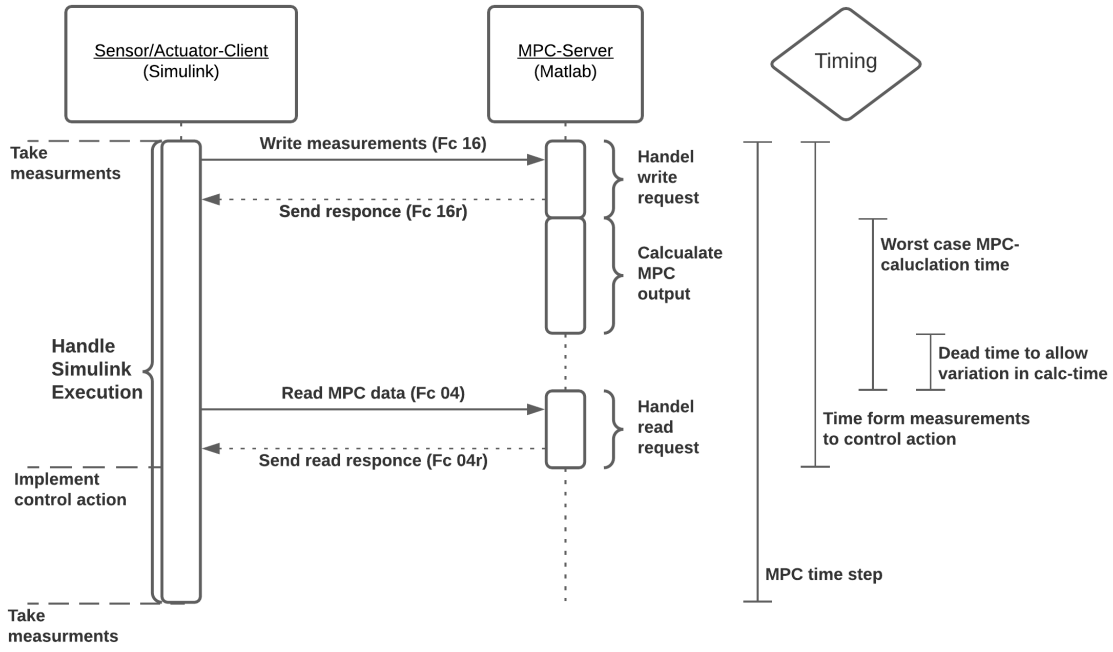



**Figure 7.13: Timing of the server and client execution.**

reason for doing this is that the lab is set up so that the "The sensor/actuator - Client" has to run at a high frequency of $2Hz$ in order to allow PI controllers to function. If the MPCs were to run at the same frequency it would require the control outputs to be

computed in less than $0.5s$. This is simply not possible for any reasonably complex model with a reasonable prediction horizon $Hp$.

The MPCs are instead computed asynchronously. The timing is determined by the model step size $\Delta t = 5s$, decided upon in the model subsection of this section. The MPC runs with a period of $5s$, meaning every $5s$ the client pushes new measurements to the MPC, using Modbus, where after $1s$ later the client requests the solution to the MPC problem. This allows the MPC to use up to close to $1s$ for computing the control inputs.

The upper bound on the computation time could be increased. This does potentially lower the performance. The decrease in performance would be caused by the states deviating from the measurements in the time it takes the MPC to compute the new control input. Compensating for this deviation in the states is possible, but we believe that having the $1s$ upper limit on computation is reasonable for our system and $Hp$ [32].

## 7.3 Results

In this sections we start by introducing the experiment setup, that we created in order to test the MPC and the CC-MPC. Further we evaluate the performance based on three criteria, specified in *section 7.2 Solutions to the Control problem*. The intention is to use the results of the two individual experiments for a comparison of the controllers in *section 7.4 Comparison of Results*.

The experiment is designed to simulate $\sim 40$ real days where there is approximately 15 rainy days with different rain intensities. This equates to a $\sim 13$ hour long experiment with the scale specified in *subsection 7.1.2 Lab scaling* in mind. The disturbance inflow (both rain and wastewater disturbance) is same for all the experiments. The initial conditions are also the same. The gravity pipe is initialized empty and both of the retention tanks star at a reference. The two tanks operate in the range from $150mm$ to $700mm$. The $150mm$ lower bound is due to the way the lab is constructed. The lower bound is a requirement in order for lab to work properly, since it is only an offset in the measurement, it might as well be ignored. The reference $\mathbf{z^{ref}}_k$ for both tanks is set to $300mm$. The reason for setting the reference higher then the lower tank bound is that we intended to leave the room such that we can observe the cautious behaviour of the controllers in the anticipation of a coming rain event. This is the expected behaviour according to authors' intuition of the controllers.

The disturbances enter into the first tank and into the middle of the gravity pipe. The outflow pump of the first tank, pumps the water into the pipe and the outflow pump of tank two pumps the water into the buffer tank, used to simulate a Waste Water treatment Plant (see *Figure 7.14*). The buffer tanks provide water for disturbance pumps. This way the water is circulating through the system. The overview of a lab setup is presented before in *section 3.2 Emulation Setup - AAU Water Lab*, *Figure 3.3*. In the event of overflow, the overflow volume flows into external tanks and is removed from the system for the rest of the experiment. We can therefore use the level measurements from the external tanks to compare with the overflow prediction that is done in the MPC. Since the measure of the overflow is not available in a real-world sanitation network, the external

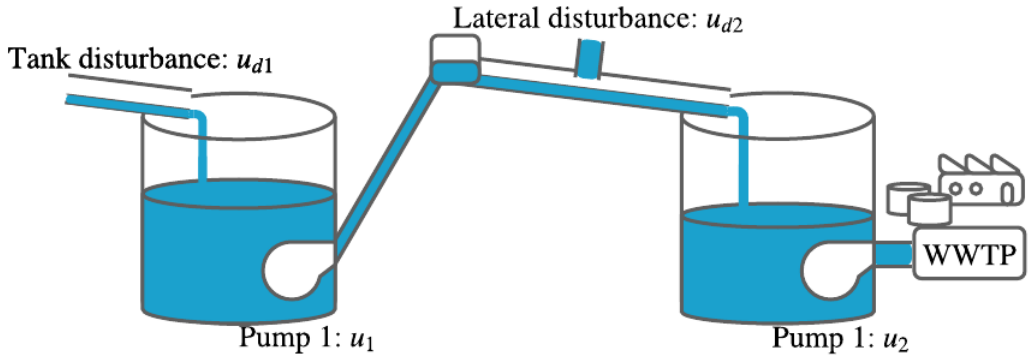



**Figure 7.14: Illustration of the Two Tank Topology with each component named**

tanks measurements are only used for validation of the MPCs predictions.

The exact operating range for all the pumps are specified in *Table 3.1*. In the following figures, used for presenting the final results of the experiments, (*Figure 7.15*, *Figure 7.16*, *Figure 7.19* and *Figure 7.20*) the reference signals are plotted in blue, while the actual measurements are plotted in red. The $d_1$ *reference*, $d_2$ *reference*, $q_1$ *reference* and $q_2$ *reference* are the reference signals for a local PI pump flow controllers. The $d_1$ *reference*, $d_2$ *reference* are the disturbance flow references $\mathbf{u_d}$ that where presented in *subsection 7.1.1 Disturbance design*, while $q_1$ *reference* and $q_2$ *reference* are the control signals computed by either MPC controller $\mathbf{u}$. The $T_1$ and $T_2$ references are the tank references mentioned above ($300mm$).

In the following sections, we present the results of each of the two individual experiments. First the standard MPC and then the CC-SMPC. The general overview of experiment results are presented in the same way for both the experiments. The general results consist of 2 graphs each, where the first graph represents the tank 1 inflow, level and outflow respectively and the second graph represents the tank 2 inflow, level and outflow. In addition results from particularly interesting events are presented as well.

### MPC experimental results

The overview of the entire experiment is showed in *Figure 7.15* and *Figure 7.16*. In this section we will comment on the standard MPCs ability to minimize overflow, its ability to keep the outflow from tank 2 slowly varying and its ability to balance the wastewater along the available storage.

By observing the tank levels in *Figure 7.15* and *Figure 7.16* we can see that no overflow happened in either of the tanks during the experiment. The physical bounds of both tanks are plotted as the green dashed lines.

Since no overflow happened during this experiment, we have to evaluate the performance of the MPC on its behaviour up to and during rain events that could potentially cause

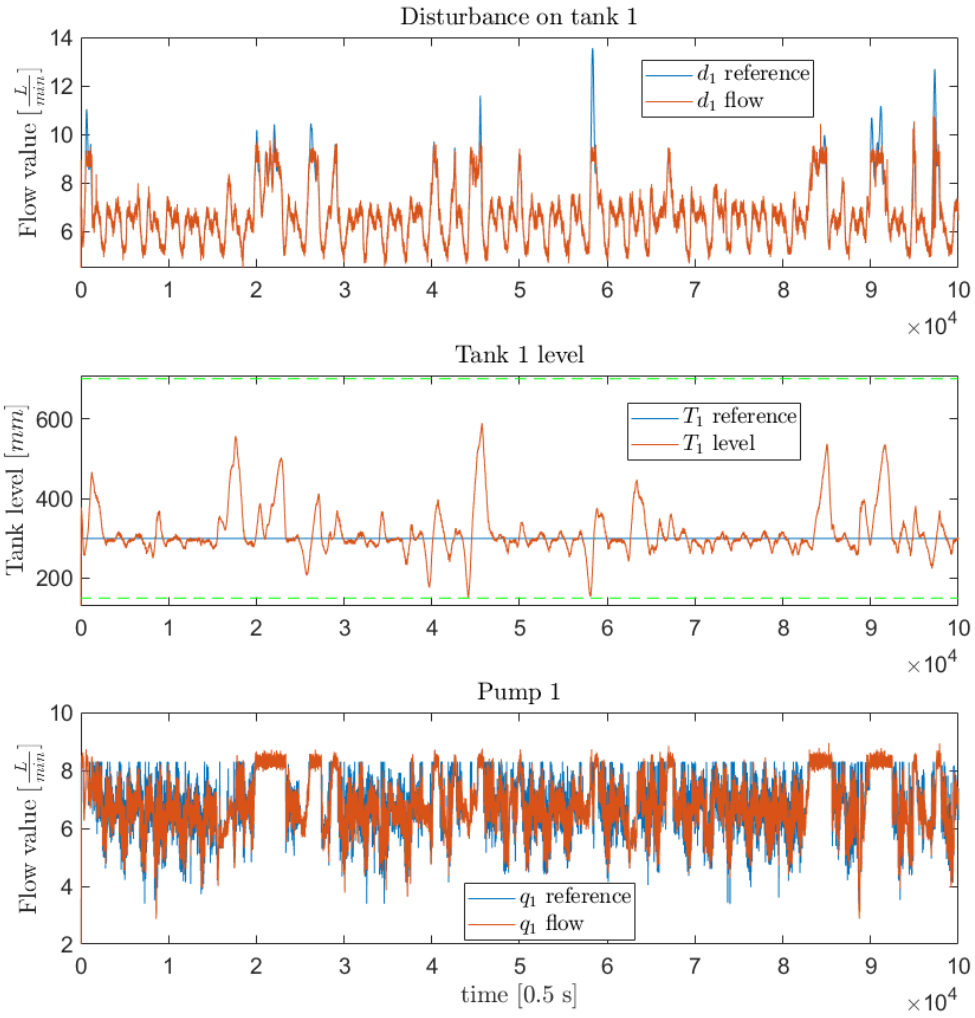



**Figure 7.15: MPC results overview, tank 1 inflow, level and outflow plots.**

overflow. This is done by looking at the tank levels. We choose to inspect the results during the two heaviest rain events. The fist rain event happened from $1.5 \cdot 10^4 0.5s$ to $1.8 \cdot 10^4 0.5s$ and the second rain event happened from $4.3 \cdot 10^4$ $0.5s$ to $4.8 \cdot 10^4$ $0.5s$. It is important to take into account that the prediction horizon is only half a day, while both of the rain events last for approximately $1,2$ days.

The first rain event is depict in *Figure 7.17*. The *Figure 7.17a* shows the MPC's prediction right before the rain event starts, and *Figure 7.17b* shows the predictions in the middle of the rain event. The disturbance predictions "$d_1$ and $d_2$ *prediction*" show the forecast prediction for the entire horizon while "$d_1$ and $d_2$ *flow*" show the actual disturbance which is implemented in the lab. Based on the forecast, the MPC solves for the optimal control action $\mathbf{u_k}$, for "$q_1$ and $q_2$ *prediction*" which represent the action of the pumps. The predicted optimal tank levels trajectory is depicted in "$T_1$ and $T_2$ *prediction*".

If we observe *Figure 7.17a*; plot $(row, col) = (1, 2)$; we can see that the MPC under predicts the actual disturbance significantly. This under prediction causes the controller

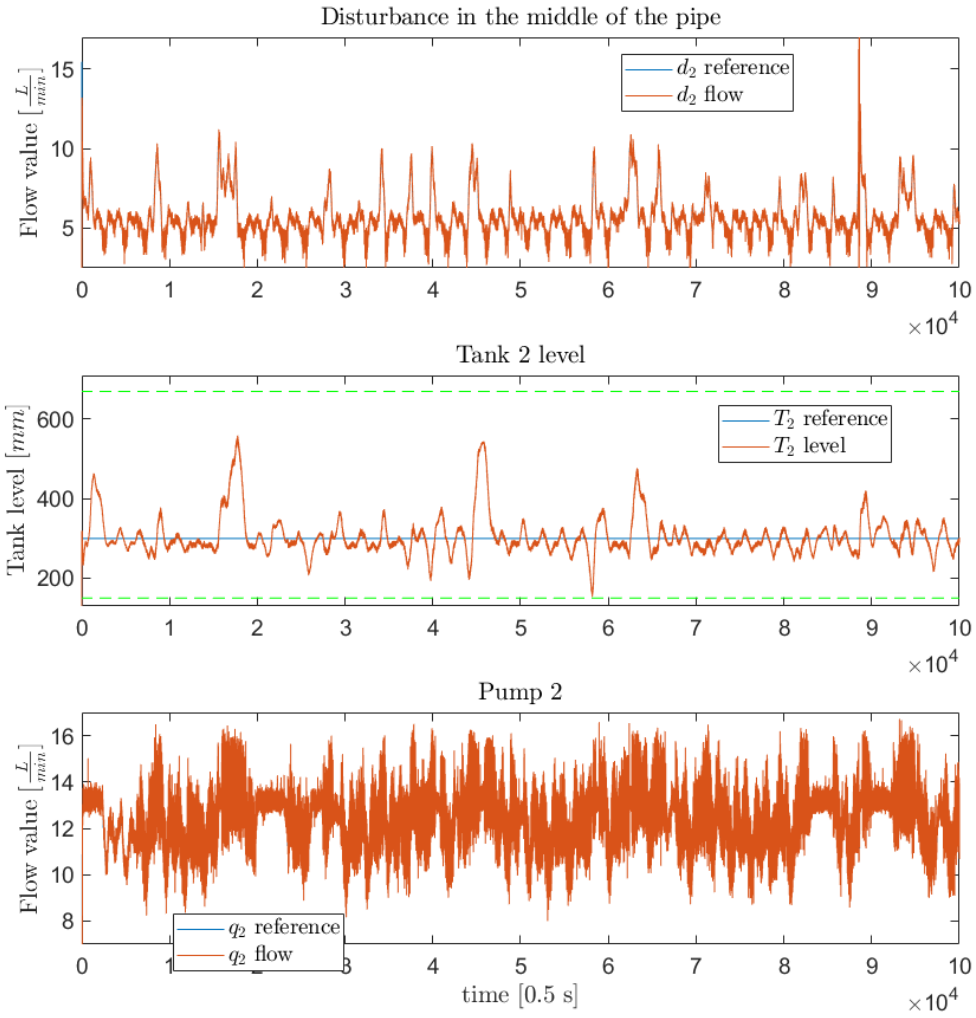
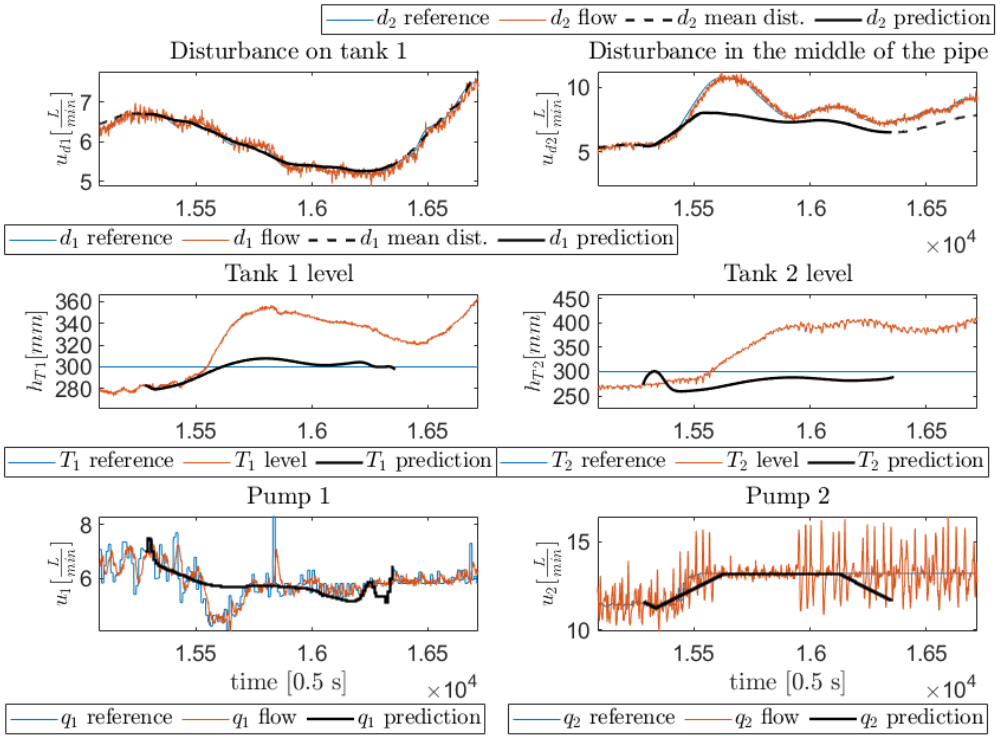



**Figure 7.16: MPC results overview, tank 2 inflow, level and outflow plots.**

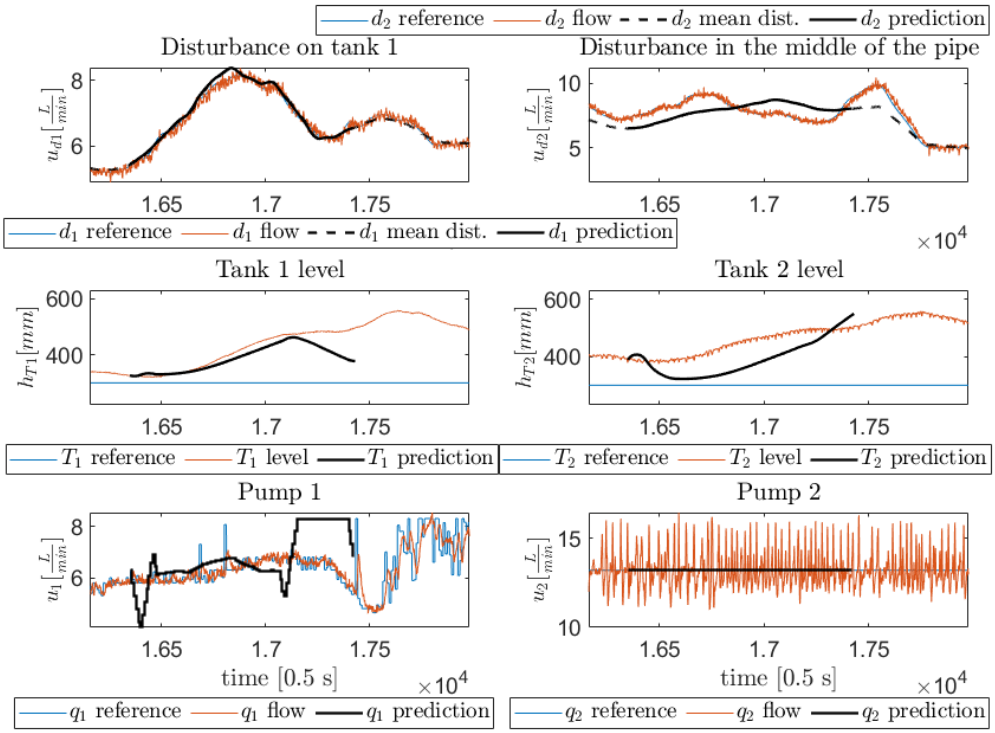
to expect, that it can keep the tank level at the reference. It therefore it does not act aggressive preemptively to minimize the level of the tank in the future. The none-aggressive behaviour can be seen on the slope of pump 2s prediction when compared to the slope of the implemented action (look at *Figure 7.17a*; plot $(3, 2)$).

In *Figure 7.17b* the performance of the MPC during the same rain event, only a few prediction steps in the future is shown. The MPC now predicts the maximum action on pump 2. In both cases, pump 1 is the pump that varies its action in order to optimize the storage of the water in the system. The variations in pump 1 allows pump 2 to avoid large changes in the inflow to the treatment plant.

The second rain event observed is depict on the *Figure 7.18*. At this event, the predictions forecast is much better, which results in better state trajectory predictions. It can also be seen that the MPC empties the tanks below the reference before the rain event starts. In both tanks the level is brought to the lower bound, before the rain starts. The level is




**(a) MPC predictions before the rain start.**




**(b) MPC performance in the middle of the rain event #1**

**Figure 7.17: Rain event number 1. The predictions of the MPC and predicted action.**

depicted as red in *Figure 7.18*; plot $(2,1)$ and $(2,2)$. This is expected behaviour in cases where high inflow is anticipated.

We can see on both pump 2 plots that the reference tracking in not working satisfyingly. The chattering behaviour observed is due to the buffer tank balancing algorithm. Unfortunately, the balancing interferes with the dynamics of pump 2, which makes the local PI controller perform poorly. We chose not to address this problem. Therefore, we can not conclude on the performance of the MPC on the criterion of smooth outflow.

However, we can evaluate the MPC on the reference provided to pump 2, plotted in blue. Where the MPC performance is good. For most of the experiment pump 2's flow is kept relatively constant. We expect we would be able to further improve the outflow smoothing performance with a longer prediction horizons, i.e. one and a half to two days. We were not able to test this hypothesis because of the computational limitations. This is further discussed in *section 7.4 Comparison of Results*.

### CC-MPC experimental results

In the previous section, we evaluated the performance of the standard MPC. We now present the results of the CC-MPC from the same experimental setup. This means that the initial conditions are the same and the implemented disturbances and disturbance forecasts are the same. First, the general overview of the experiment is presented in *Figure 7.19* and *Figure 7.20*. Later, we focus on a specific segments of the experiment in order to analyze the performance of the controller. The performance is evaluated on: its ability to minimize overflow, its ability to keep the outflow from tank 2 slowly varying and its ability to balance the wastewater along the available storage.

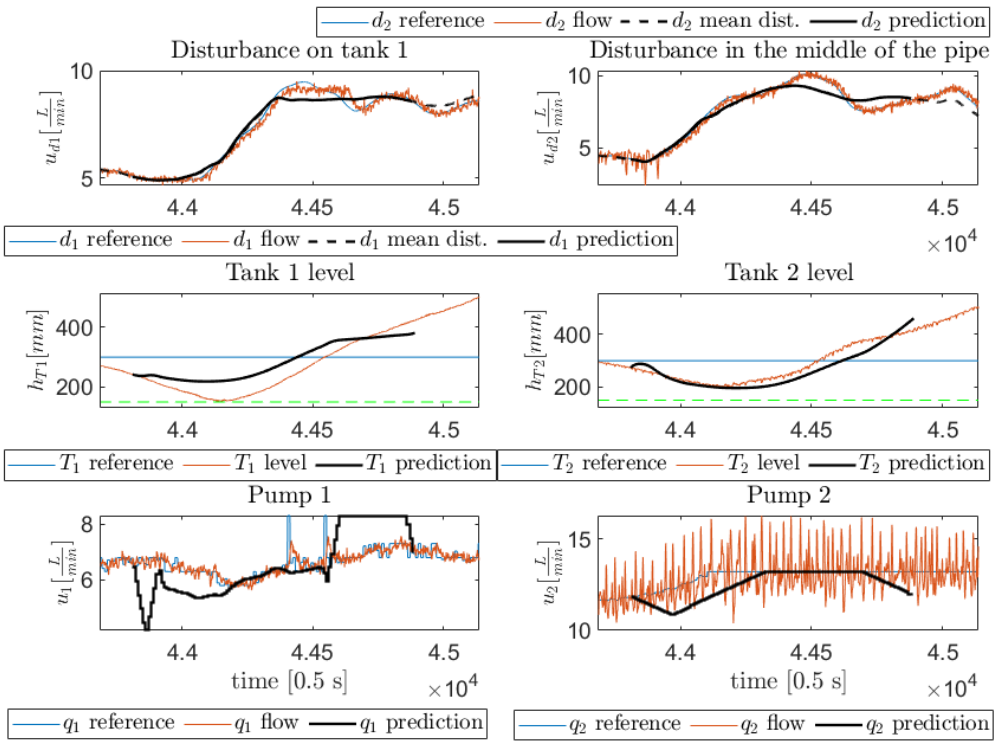



**Figure 7.18: Rain event number 2. The predictions of the MPC and predicted action. The cautious action.**

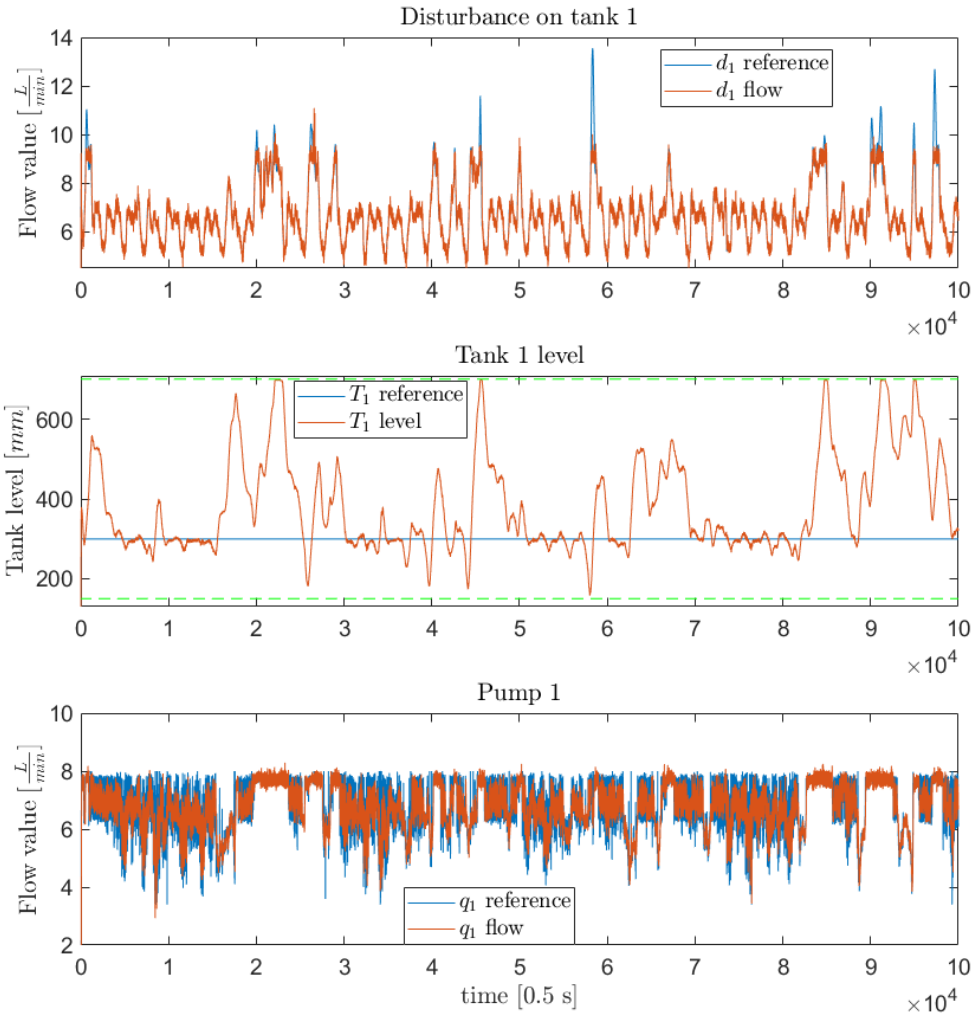



**Figure 7.19: CC-MPC results overview, tank 1 inflow, level and outflow plots.**

On the middle plots in *Figure 7.19* and *Figure 7.20*, the tank levels are depict. We can observe that for certain rain events both tanks have their entire capacity utilized. This inevitably leads to overflow. A very obvious overflow happened around $2.35 \cdot 10^4$ $[0.5s]$ and $9.25 \cdot 10^4$ $[0.5s]$ in tank 1, see *Figure 7.19*. Therefore we elect to analyze the performance of the CC-MPC by looking at the results around these particular rain events. Additionally, we compare the overflow predictions made by the MPC to the actual overflow which is measured using the external tanks.

First, we analyze the rain event that starts around $1.5 \cdot 10^4$ $[0.5s]$. This rain event is a combination of two successive heavy rainy days. System behaviour for the entire two day span is showed in *Figure 7.24*. The figure also shows the CC-MPCs predictions at four instances. In 3 out of 4 predictions the controller received a forecast which has a rain inflow, lower than the implemented inflow.

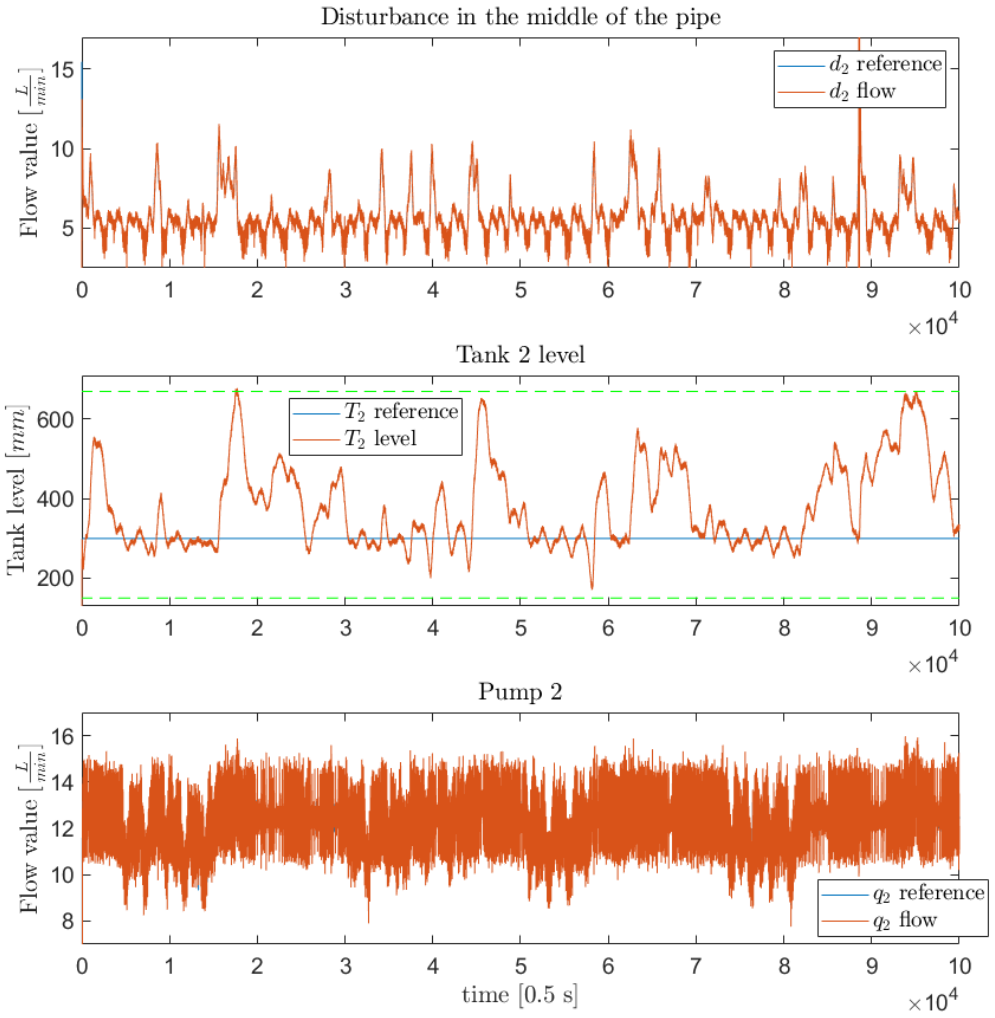



**Figure 7.20: SMPC results overview, tank 2 inflow, level and outflow plots.**

We have previously noted that the knowledge of uncertainty reduces the tank level constraints, making the CC-MPC more conservative. This means that the CC-MPC becomes more conservative the more the uncertain predictions are. The tightening on the constraints around $1.5 \cdot 10^4$ $[0.5s]$ is depict in *Figure 7.21*. The tightening on the state is not significant, only 2% to 3% of the level range. This is due to the use the LQR controller, along with the MPC. The result of using the LQR controller is a reduction in the tightening of tank level constraints, while the LQR controller tries to drive the states towards the MPC's predicted state trajectories. Because the LQR controller is controlling a portion of both pumps range, this range has to be reserved from the MPC, see (7.38).

This reservation results in a decrease in the maximum control signal which the MPC is allowed to supply, showed in *Figure 7.21*. However, since the level tightening is insignificant, when the system is operating at the tank reference, the CC-MPC does not act cautiously. By cautiously we mean that the CC-MPC would drive the state below the reference in order to prepare for the coming heavy inflow.

Some cautious action can be seen before and in the first prediction that is plotted in *Figure 7.24*. But this is only due to the fact that the CC-MPC predicts that it will be able to hold the reference in the future, by performing this action.

After the first prediction, the second pump is already saturated to the maximum value that the MPC can provide (which is around $12.3\frac{L}{min}$). This means that the conservativeness which the chance constraints provides can not be utilized in order to further help in the prevention of the coming overflow. Because by the time the CC-MPC predicts the potential overflow it is already applying maximum outflow using pump 2. So it is left to pump 1 to act optimally in order to balance the sewer. Pump 1 can however only try to prevent overflow until it is also providing its maximum outflow similarly to the pump 2. Taking all this into account, the conclusion is that the overflow at $2.35 \cdot 10^4$ $[0.5s]$ happens due to the fact that with maximum outflow at all times, the pumps are not able to remove enough volume from the tank and the overflow is inevitable.

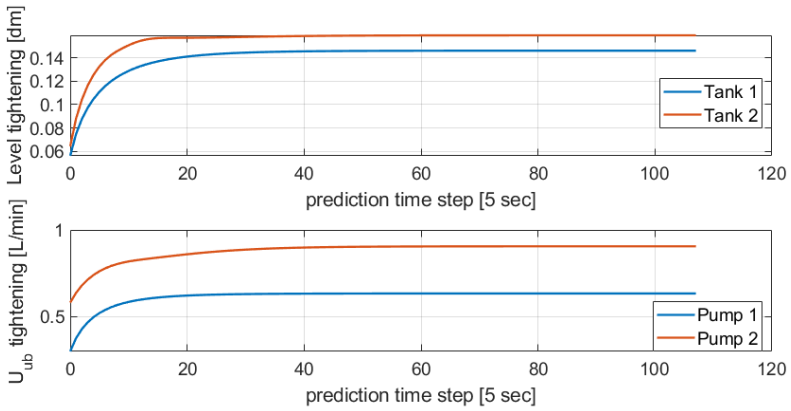



**Figure 7.21: Tightening of tank constraints and control pump flows due to uncertainties of the model and predictions.**

In regards to outflow smoothing using pump 2, we can conclude that there is not much variation for periods of time longer then a single day, which is wanted behaviour. Pump 1 on the other hand is allowed to vary more and thereby perform finer control on both tanks. The same problem of poor local PI control, experienced for by the standard MPC is present in this experiment. The cause is also the same.

We analyze how accurate the CC-MPC is at estimating and predicting the overflow. In *Figure 7.22* the overflow event described before is depicted. This time the figure is extended to include overflow estimation and prediction on the bottom most plot. Each plot shown in *Figure 7.22* respectively shows: the rain forecast that caused the overflow, a prediction of tank 1's level reaching the upper bound, the optimal action for pump 1 and the predicted overflow.

On the overflow plot in *Figure 7.22* , the red signal represents the actual overflow volume measured using the external tank. The blue signal represents a one step solutions for the CC-MPC overflow variable. In other words it represents what the MPC estimates as the overflow during the experiment. The black signal represents a single prediction of the overflow from around $2.22 \cdot 10^4$ $[0.5s]$. It is important to note here that the estimated

overflow (blue) and the predicted overflow (black) are both the solution of the MPCs optimization problem.

We can see from *Figure 7.22* that even though the rain inflow is underpredicted for most of the horizon, the MPC tends to overestimate the overflow volume (blue) significantly. The prediction of the overflow (black) is even worse, but that is due to the tightening of the upper bound of the control inputs **u**, caused by the LQR controller. The tightening can be seen the second bottom most plot in *Figure 7.22*. The length and dynamics of an actual overflow event should not be considered, because we do not take into account the dynamics between internal and external tank when we calculate the actual overflow volume (showed in red). Even though the overflow estimated from MPC can not be used for a precise measure, it can still serve as a indication that the overflow event happened and as a rough estimate of the overflow volumes.

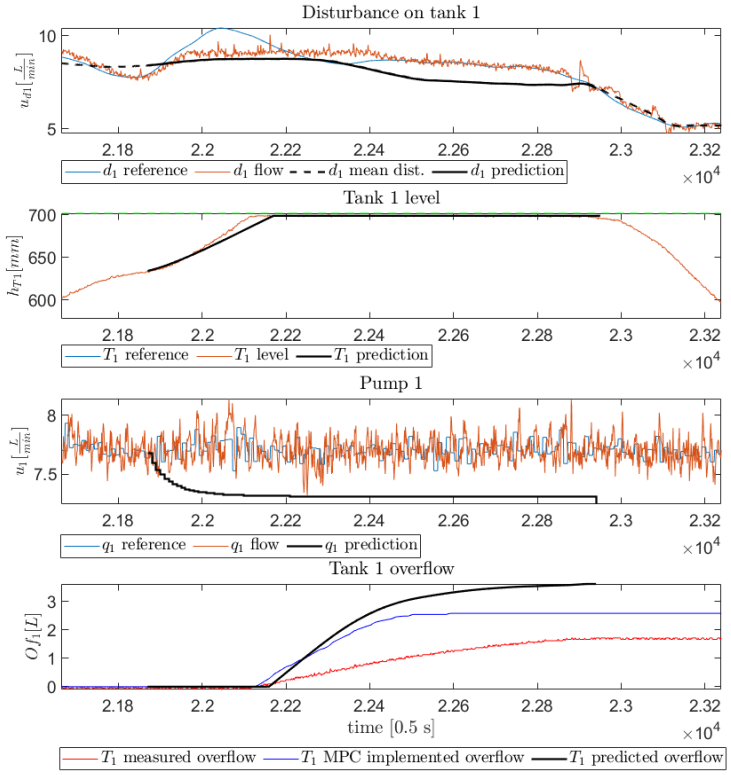



**Figure 7.22: The overflow prediction during the rain #1 scenario.**

## 7.4 Comparison of Results

In this section a comparison of the MPC and CC-MPC will be conducted. This comparison will take offset in the results presented in the previous section. The reason we can compare the two MPCs based on the results, lies in the fact that the experiments used to produce the results were set up with the same initial conditions and using the same inflow disturbances. However to compare the two controllers properly, a extra experiments have been conducted, as this will help to showcase the specific features of each controller.

The control objectives which were presented in *section 7.2 Solutions to the Control problem*, will be the comparison criteria. As a good measure the objectives are listed again:

1. Minimize overflow in a network.
2. Avoid fast varying flow to a Waste Water Treatment Plant.
3. Balance wastewater along available storage.

As the results from the experiment utilizing the standard MPC did not exhibit any overflow, one could be tempted to immediately conclude that the standard MPC is superior with regards to overflow prevention. We however suspect that the superiority of the MPC shown in the results has to do with our experiment and design choices. Elaborating on this we expect that a combination of three factors lead to the MPC outperforming the CC-MPC:

- Low uncertainty of the disturbance predictions.
- "Short" prediction horizon $Hp$.
- The use of the LQR controller in the CC-MPC.

Optimally we would re-do the experiments taking these three facts into account. Re-running experiments to test for longer $Hp$ and higher uncertainty, would however require a major rework of the current control and software setup, and would require much consideration on experiment design. Instead we choose to argue base on the currently available information.

The tightening of the tank levels upper bound $\overline{x}$, can be seen in *Figure 7.21* and is due to the chance constraints presented in (7.35). In order for the tightening to have an effect on the control $\mathbf{u}$ in anticipation of a rain event, assuming the system is operating the the reference, the tightening has to decrease the upper bound significantly. This phenomenon is showcased in 6.3, and again shown in figure.

Observing the first of the four predictions in *Figure 7.24*, we can see that the MPC does not predict the tank to approach the upper bound. This is due to a to small $Hp$. If the $Hp$ had been larger both the CC-MPC and the standard MPC would have been able to preemptively start emptying tank one. In the case where the CC-MPC had a significant tightening of the upper bound the CC-MPC would start to preemptively empty the tank before the standard MPC. Thereby outperforming the standard MPC. But as we have neither a "significant" decrease in the upper-bound nor a long enough $Hp$ for our system, no performance benefits can be observed for the CC-MPC.

The tightening of the upper bound using the CC-MPC, is as little as $2-3\%$. This means by

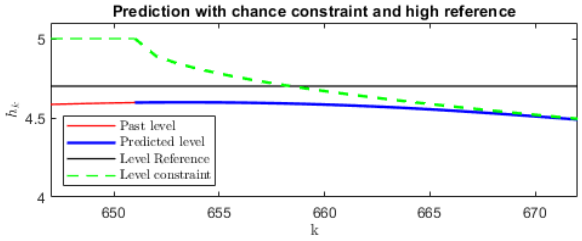



**Figure 7.23: The significant tightening of the upper bound that affects the control action.**

the time the tightening is relevant to the CC-MPC, it is already pumping at full capacity. Observing the last of the four predictions in *Figure 7.24*, where the MPC predicts the overflow, the tightening is activated. However when the tightening becomes active the MPC cannot use this in order to calculate the better action, since it is already providing the maximum outflow.

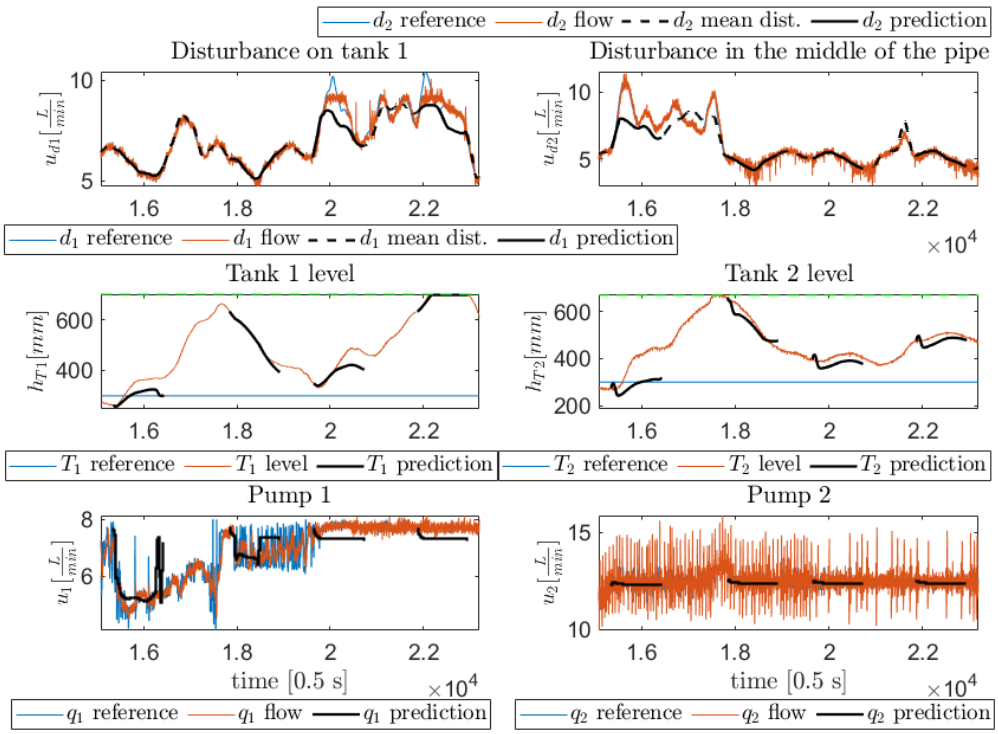



**Figure 7.24: Overflow scenario #1 with SMPC predictions and optimal control actions.**

The tightening is a result of the variance dynamics $\mathbf{\Sigma}_{\mathbf{x}|k+1}$, show in (7.30). The variance dynamics are a function of the system dynamics and the uncertainties in the system: $\mathbf{\Sigma_a}$, $\mathbf{\Sigma_d}$, $\mathbf{\Sigma_w}$ and $\mathbf{\Sigma}_{\mathbf{x}|k}$. In order to have additional tightening of the tanks upper bound the inputs would have to be more uncertain, compared to the dimensions of the system. It is however not prudent to increase uncertainty in a system simply to make a specific controller perform better.

We have now touched upon what we could do in order to make the CC-MPC perform better in the comparison, specifically by looking a $Hp$ and the effect the uncertainties have on the tightening of the tanks upper bound. We are however also aware that the implemented LQR controller helps to decrease the variance of the system, at the cost of some of the CC-MPCs pump control range. By observing and comparing the two results, we conclude that the biggest contributing factor to the fact that the standard MPC outperforms the CC-MPC, comes from this loss of control range (see *Figure 7.24* and *Figure 7.21*).

As testing the hypothesis that: "the use of the LQR controller is responsible for the overflow" does not require a major reconfiguration of the control structure or experimental setup, we have elected to test it. In order to allow the CC-MPC to control the entire

range of **u**, like the standard MPC, we simply remove the LQR controller. Note, that an additional benefit of removing the LQR controller is that the tightening of the upper bound is increased. As shown in *Figure 7.25*.

In *Figure 7.26* and *Figure 7.27* the results of running the experiment using the CC-MPC with out the LQR controller is shown. From the results we can see that the CC-MPC was able to avoid overflow, but that the performance was still worse than the standard MPC. Even without the LQR the tightening of the upper bound still did not increase to a point where they provided a performance benefit. Inspecting the control signals closely the CC-MPC without LQR and the standard MPC, output very similar signals. We therefore assume the difference in performance comes down to variations in the laboratory setup.

## 7.5 Conclusion

Based on the experiments presented the use of the CC-MPC for overflow prevention in sanitation networks, cannot be concluded to have any benefits over using the standard MPC. It can be speculated that the CC-MPC might yield performance benefits if used on a very uncertain sanitation system with a long prediction horizon.

When using the CC-MPC for control of a sewer system the addition of an LQR control policy, seems to degrade performance. The LQR control policy have been shown to be useful in other applications, such as in the control of a race car, see [11]. For a race car, being conservative with regard to its control inputs, such as speed, prevents the car from slipping out in the corners, which are considered very uncertain scenarios. Because the LQR controller results in an decrease in the control input it means that the CC-MPC has a decreased ability to counteract overflow.

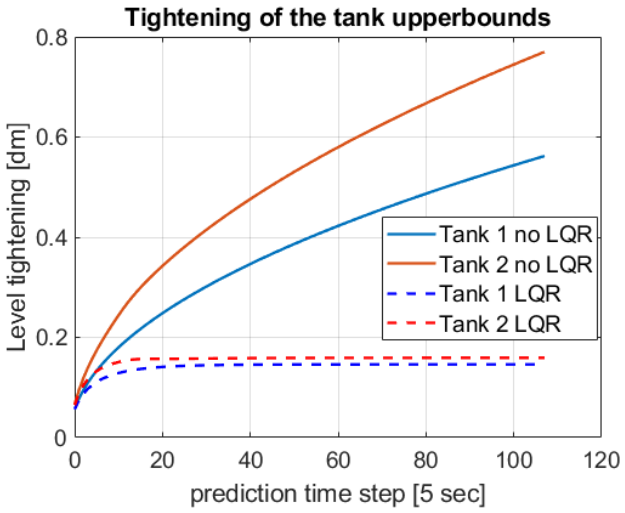



**Figure 7.25: The tightening of the tank upper bounds with no LQR contribution.**

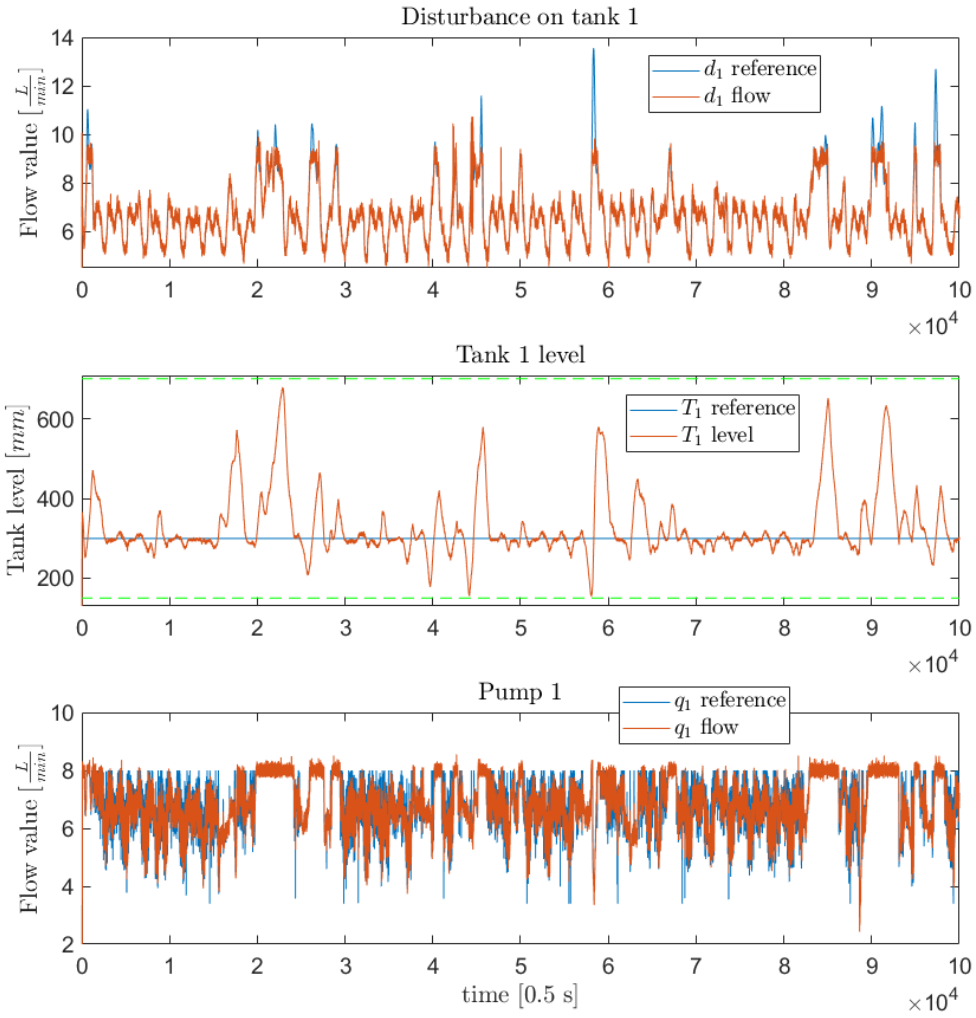



**Figure 7.26: CC-MPC results overview without the LQR contribution. Tank 1 inflow, level and outflow plots.**

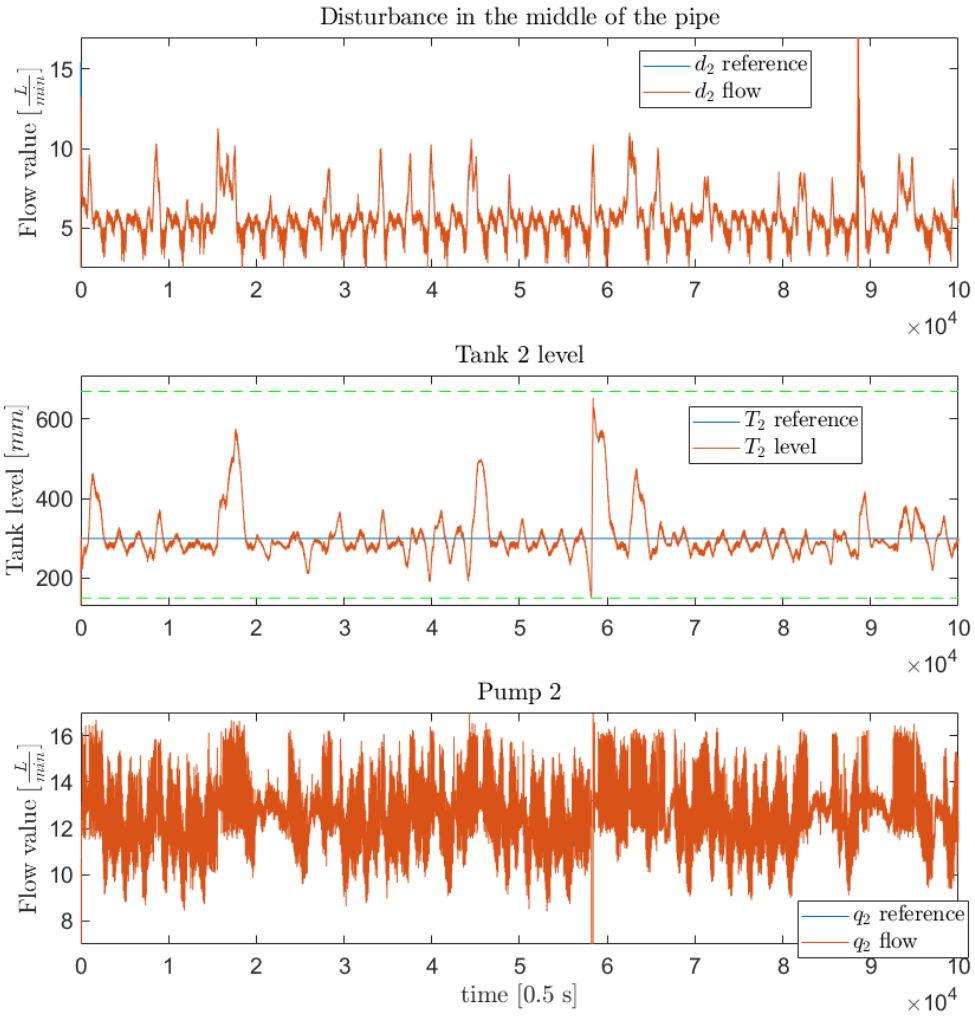



**Figure 7.27: CC-MPC results overview without the LQR contribution. Tank 2 inflow, level and outflow plots.**

# Discussion of the work 8

This chapter presents a discussion on several topics from the thesis, which we believe required a further comment. These topics include:

- noise identification,
- changes to the identification process,
- model capabilities,
- model stability,
- overflow estimation and
- disturbance design.

## 8.1 System Identification

In this section, we discuss the quality of our system identification framework, developed in *chapter 4 Model Parameter Identification.* We further discuss what changed to accommodate the altered system that we used in the control section of the thesis. We also summarize the limitations and challenges of our identification approach. As we only used the AAU Smart Water Lab model in the control section, this model will be the focus of the discussion.

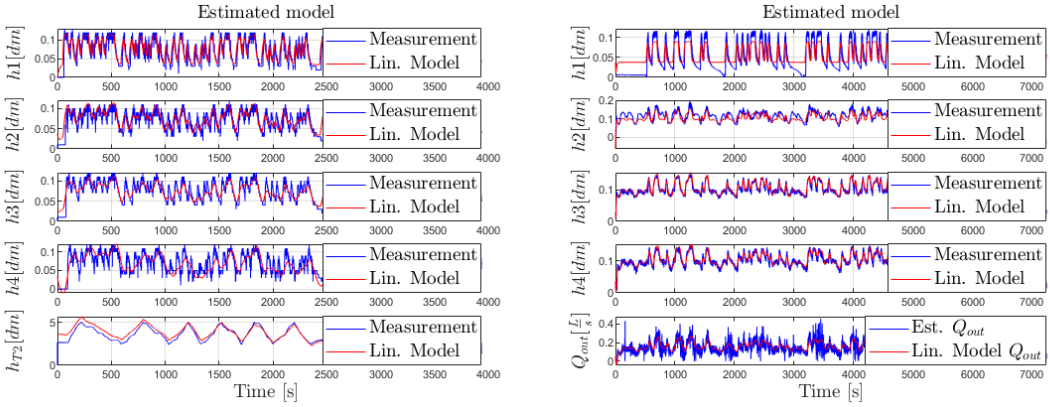
First, let us recap the lab parameter identification from *subsection 4.2.2 Results of the AAU Smart Water Lab emulation identification.* In the subsection, we estimate the parameters of a gravity pipe. The parameter estimation method is a data-driven optimization problem that iteratively finds the best fit of the model parameters to the measurements gathered from the system. The tank parameters are considered known since they directly represent a physical value, the area of the tank, which is easily obtainable from the lab schematics in Appendix C.

In *Figure 8.1a* and *Figure 8.1b* a results of the parameter identification are presented. It shows a free-running prediction in red and actual measurements in blue. *Figure 8.1a* depicts the original parameter identification, where it is important to notice that we use the level of tank 2 as a reference measurement. The tank level is the bottom-most plot. Using the tank level as a measurement in system identification is not a good approach for two reasons:

- The effect of integration caused by the tank model is present in the free running model which makes the model fit worse.
- The framework assumes there is a tank at the pipe outlet. This makes the identification approach less general.

Because of these two reasons, we elected to change the framework slightly. The change switched the identification process to use the pipe's outflow as the last reference data set rather than the tank level. The outflow of the pipe is depicted in the bottom-most plot in *Figure 8.1b*.

In the lab setup, we had to estimate the outflow based on the tank measurements and the tank outflow. This does not lead to a better estimation of the pipe outflow, as the outflow estimate is only good as the data provided. Thus we can argue that the new implementation did not achieve better precision. The new framework is more general as it is independent of boundary element at the output of the gravity pipe.




**(a) The original identification method** **(b) New parameter estimation.**

**Figure 8.1: Comparison of updated parameter estimation framework.**

If we closely observe *Figure 8.1b*, we can see that the measurements of the pipe states $h2$, $h3$ and $h4$ look different from the corresponding states on the *Figure 8.1a*. It appears that there is a constant offset in all three state measurements. Furthermore, the dynamics of the first pipe section no longer clearly translates through the rest of the data set, as it does in *Figure 8.1a*. The reason for this is that the new experiment, which we elected to use for the model synthesis in the *section 7.2 Solutions to the Control problem*, is subject to a lateral disturbance between the sensors measuring $h2$ and $h3$.

The lateral disturbance defines the majority of the dynamic behavior of the states $h2$, $h3$, and $h4$. This disturbance affects the level and the flow in the pipe section that comes before the inflow point. This is an interesting observation. We elect to call this phenomenon "the backflow effect".

One of the original reasons for choosing the Diffusion Wave model lies in the fact that we expected it to be able to model this backflow effect. Unfortunately, using our linear Diffusion Wave model, we were not able to capture this effect in state $h2$ to a satisfying degree. At this time, we do not have a solid argument to prove that it is impossible to capture this effect with a linear version of the Diffusion Wave model. Therefore, we did not introduce the backflow effect in any of the previous chapters. Instead, we elected to proceed with the linear Diffusion Wave model through the thesis. The obvious alternative to using the Diffusion Wave model would have been using the linear Kinematic Wave model

from [8].

We co-authored a paper where the same problem of capturing the backflow phenomenon in the same setup has been researched [15]. The paper presents a comparison of the nonlinear Kinematic Wave model and the nonlinear Diffusion Wave model. The comparison focuses explicitly on capturing the backflow phenomenon. The paper showed that it is possible to capture the backflow effect using a nonlinear Diffusion Wave model.

Since the backflow effect is always measured at the second sensor in the AAU Smart Water lab, we had to find a way to include this behavior in our linear Diffusion Wave model. We had to make a workaround using our linear model to capture the effect. We did this by modeling the disturbance inflow as if it entered into state $h2$. That yields the model prediction, showed in *Figure 8.1b*, which we consider satisfactory.

We already mentioned that the tank parameters are considered known and that they are not a part of the system identification. We discovered that the flow sensors which are used in the lab are not very precise and have a constant negative deviation from the actual flow value. We, therefore, chose to trust the level sensors in the tank instead. The incorrect flow measurements did not cause a problem in the identification of the pipe. This is because the data-driven method compensates for the flow discrepancy in the parameters. For more information on the flow discrepancy, see *section 8.4 Overflow Estimation*

The flow discrepancy does, however affect the tank models. This is because the tank parameters are not considered as part of the identification problem. The flow discrepancy can be seen when correlating the tank's inflow and outflow with the tank's volume change. If we want to make the entire system robust to this flow error, we would have to include both tanks' parameters in the parameter estimation.

## 8.2 Model Considerations

In *subsection 2.2.1 Stability of the General Model* we found two sets of sufficient conditions for marginal stability of the discretized gravity pipe with any number of discrete pipe sections. These two sets of sufficient conditions are listed in (8.1) and (8.2). Both sets are subject to the same assumptions on parameter relations specified below in (8.3).

$$\text{Sufficient Condition 1:} \quad (8.1)$$
$$\theta_5 + \theta_3 \geq \theta_2,$$
$$\theta_2 \geq \theta_3$$

$$\text{And Sufficient Condition 2:} \quad (8.2)$$
$$\theta_5 \Delta t \leq 1$$

$$\text{Assuming:} \quad (8.3)$$
$$(\theta_2 + \theta_3)\Delta t < 1,$$
$$(\theta_5 + \theta_3)\Delta t < 1,$$
$$\theta_1, \theta_2, \theta_3, \theta_5 > 0.$$

$$\text{Sufficient Condition 1:} \quad (8.4)$$
$$\theta_5 + \theta_3 \geq \theta_2,$$
$$\theta_2 \geq \theta_3$$

$$\text{And Sufficient Condition 2:} \quad (8.5)$$
$$\Delta t \leq a_3 \cdot \Delta x$$

$$\text{Assuming:} \quad (8.6)$$
$$\Delta t < \frac{a_1 \Delta x^2}{b_1 \Delta x + 2},$$
$$\Delta t < \frac{a_2 \Delta x^2}{b_2 \Delta x + 1},$$
$$\theta_1, \theta_2, \theta_3, \theta_5 > 0.$$

As mentioned, in *subsection 2.2.1 Stability of the General Model* we typically consider the relation $\frac{\Delta t}{\Delta x}$, when considering the stability of a discretized partial differential equation. The sufficient conditions are expanded from (8.2) and (8.3) and reduced to include the terms $\Delta t$ and $\Delta x$. This yields a restatement of the sufficient conditions shown in (8.5) and (8.6).

Notice that we did not expand the first sufficient condition (8.4). This is simply because $\Delta t$ is not a part of the two equations making up the condition. Additionally we have throughout the identification process found, that the first sufficient condition does not hold. Because the relation $\theta_5 + \theta_3 \geq \theta_2$ does not hold. Note that we have not proven that (8.4) does not hold, rather we have found the result empirically. To illustrate the result we have depicted three identification results in *Table (8.1)*.

| - | 4 State Mode | 8 State Model | 12 State Model |
|---|---|---|---|
| $\theta_1$ | 0.034 | 0.057 | 0.080 |
| $\theta_2$ | 0.086 | 0.137 | 0.194 |
| $\theta_3$ | 0.0065 | 0.0001 | 0.0001 |
| $\theta_4$ | -0.002 | -0.005 | -0.007 |
| $\theta_5$ | 0.055 | 0.100 | 0.100 |

**Table 8.1: Estimated parameters for different number of model states.**

Sufficient Condition 2 in (8.5) and the general assumptions in (8.6), have been expanded. The expansion is done using the general variables $a$ and $b$. Here we observe that we get

three increasing functions, as shown in *Figure 8.2*. This fits well with our intuition. We would expect that longer pipe sections, would result in slower dynamics, and therefore allow for a larger $\Delta t$.

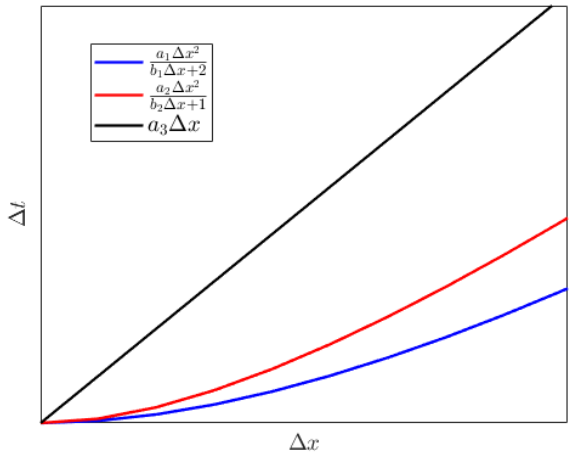



**Figure 8.2: $\frac{\Delta t}{\Delta x}$ relation functions.**

In *chapter 2 Modeling* we created the generalized parameters $\theta$ based on physical parameters. One could imagine that the most straightforward approach to system identification is identifying the physical parameters themselves. This is, however, not possible because we have more physical parameters than we have equations. It may, however, still be possible to find a physical interpretation for two of the generalized parameters.

- $\theta_1 = \frac{1}{w \cdot \Delta x}$ can be interpreted as a conversion factor between level change pr second and flow.
- and $\theta_5 = C_d\sqrt{2g}\theta_1$ can be interpreted as free-flow factor which determine the relation between the level in the last pipe section and the pipe outflow.

We are aware of the width of the actual gravity pipe in the AAU Smart Waterlab. Therefore we can roughly estimate the value of $\Delta x$, assuming that the system identification process identifies the system parameters, such that they represent a combination of the physical values. There is, however no guarantee. We can try to calculate $\Delta x$ based on three models with 4, 8 and 12 pipe section states. We would expect $\Delta x$ to be halved when going from 4 pipe states to 8 pipe states, and we would further expect a reduction of $\Delta x$ of around one third from 8 til 12 pipe states. We investigate this using (8.7).

$$\Delta x = \frac{1}{w \cdot \theta_1}, \tag{8.7}$$

$$\text{For 4 states: } \Delta x = 50.64\ dm,$$

$$\text{For 8 states: } \Delta x = 34.60\ dm,$$

$$\text{For 12 states: } \Delta x = 24.42\ dm.$$

The results are roughly as we expect, taking into account that we are taking the diameter of a round pipe, which in the model is being approximated by a square pipe. We therefore

conclude that the above results indicate that the parameter identification is estimating the physical parameters indirectly.

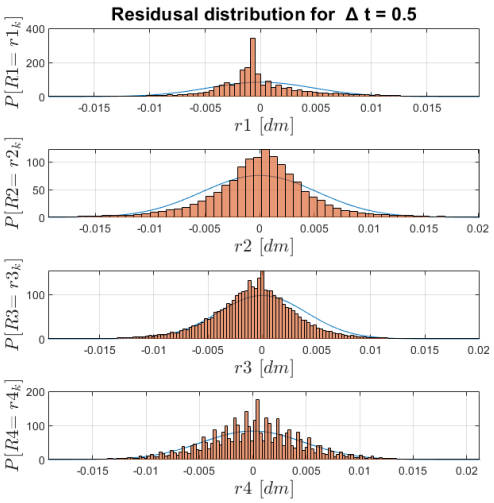
## 8.3 Noise Identification

Noise identification for the AAU Smart Water Lab was done in two different chapters of this thesis. These chapters are: *chapter 5 State estimator*, *section 7.2 Solutions to the Control problem*. The noise identification presented in both chapters was based on the assumption that the model used had a $\Delta t = 0.5$. It follows that the model noise found is unique to the model with that specific $\Delta t$.

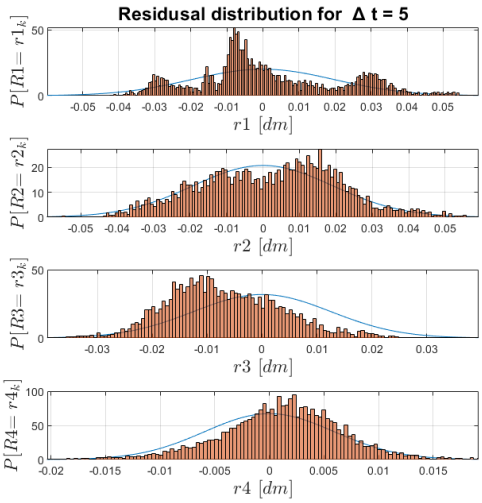
The implementation details of the control system were outlined in *subsection 7.2.6 The Control System Software Setup*. In that section, we highlighted that the "CCU client" and "CCU server" operate at different frequencies, meaning the MPCs and the Kalman filter operate at different frequencies. The execution period for the Kalman filter is $0.5s$ while the execution periods of the MPCs are $5s$.

Therefore, we have made a mistake in assuming that the noise models identified for the Kalman filter could be used directly in the CC-MPC. *Figure 8.3* shows the combined model and measurement noise in the case of $\Delta t = 0.5$ and $\Delta t = 5$, for a pipe model with four states.

It can be seen that the pipe residuals in *Figure 8.3b* have a much higher variance than the pipe residuals shown in *Figure 8.3a*. If we were to use the higher variance models when calculating the variance dynamics for the CC-MPC, we expect that it would increase the constraint tightening. For a detailed explanation of the impact additional constraint tightening can have on control performance, refer to *section 7.4 Comparison of Results*.




**(a) Original noise identification of combined sensor and measurement noise.**




**(b) New noise identification of combined sensor and measurement noise.**

**Figure 8.3: Comparison of residual distributions, between original results, and new results using the correct $\Delta t$.**

## 8.4 Overflow Estimation

*Figure 7.22* in the result section depicts the CC-MPCs overflow estimation and the actual overflow measures found using the external tanks in the lab setup. There is a significant difference between the estimated overflow and the actual overflow. We aim to explain the discrepancy in this section.

We know the tanks discrete state space equations are described by (8.8) and (8.9).

$$\begin{aligned} h_j^{n+1} &= \phi \Delta t (Q_{in}^n - Q_{out}^n) + h_j^n, \\ &\Downarrow \\ h_{T1}^{n+1} &= \phi_1 \Delta t (u_{d1}^n - (u_1^n + u_{of1}^n)) + h_{T1}^n, \qquad (8.8) \\ h_{T2}^{n+1} &= \phi_2 \Delta t \left( \frac{\theta_5}{\theta_1} \cdot h_{P8} - (u_2^n + u_{of2}^n) \right) + h_{T2}^n, \qquad (8.9) \end{aligned}$$

where $n$ is the time index, $Q_{in}^n$ and $Q_{in}^n$ are the in and outflows, $h_j^{n+1}$ is the water level in tank $j$, $u_{of}^n$ is the overflow control input and $u^n$ is the control input to the pumps. In the event of an overflow, we assume that the change in the state is zero, meaning that we can solve for the overflow variable in both state space equations by equating $h_j^{n+1} = h_j^n$. This yields (8.10) and (8.11) as the equations for overflow.

$$u_{of1}^n = (u_{d1}^n - u_1^n) \qquad (8.10)$$

$$u_{of2}^n = \left( \frac{\theta_5}{\theta_1} \cdot h_{P8} - u_2^n \right), \qquad (8.11)$$

This makes sense intuitively. However, when we consider that the terms in both overflow equations are subject to uncertainty, the discrepancy in the overflow estimation becomes apparent. With uncertainty, the overflow equations become (8.12) and (8.13).

$$u_{of1}^n = \left( d_{mean1|0} + d_{uncertainty1|0} - (u_1^n + a_1^n) \right) \qquad (8.12)$$

$$u_{of2}^n = \left( \frac{\theta_5}{\theta_1} \cdot x_{P8} + v_4^n - (u_2^n + a_2^n) \right), \qquad (8.13)$$

where uncertainties come in the form of actuation noise $\mathbf{a}^n$, uncertainty in the inflow forecast $\mathbf{u_{d|k}} = \mathbf{d_{mean|k}} + \mathbf{d_{uncertainty|k}}$, described in *section 6.2 Stochastic Model Predictive Control*, and measurement noise $\mathbf{v}^n$. Even if these uncertainties are small compared to the in and outflows, they end up causing a significant difference in the estimated overflow volume. This is because, the overflow volume is calculated as an integration of the two overflow variable $\mathbf{u_{of}}$.

An example of overflow estimation for tank 1 was created in lab, using the standard MPC. We elected to give the MPC a forecast without uncertainty, in hopes to eliminate $d_{uncertainty}$'s effect on the overflow estimation. The overflow volume is found based on three different methods. The result of each method is shown as a function of time in *Figure 8.4*. The first overflow calculation is done based on the level of the external tank (yellow) using the equation $V_{overflow} = \Delta h_{T1} \cdot Area_{T1}$. We assume that this is accurate. The overflow estimated by the standard MPC (red), seems to be under estimating. We expected this and it can be because of two reasons:

- it is either because of measurement uncertainty on the flow sensors,

- or because of the input uncertainty $a_n$.

As the standard MPC does not take the input uncertainty into account, this could be a possible factor. However if we calculate the overflow volume, based on the actual flow measurement of the disturbance pump 1 and pump 1, we get even worse results (blue). The calculation using the pump measurements is a integration of the difference between the pump inflow and pump outflow at at every time step. We can therefore conclude that the poor overflow estimation is likely due to uncertain flow measurements. It is not

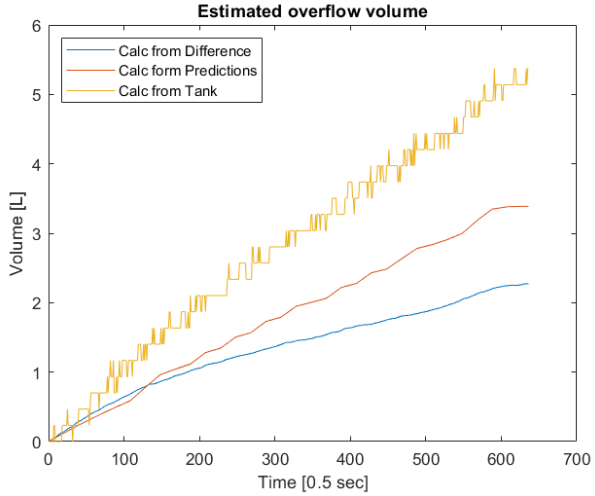



**Figure 8.4: Three different methods used for calculating the overflow.**

rare that flow sensors provide "poor" readings, even in a laboratory environment. For this reason, the question arises: How can overflow estimation be done without using flow sensors? The short answer is that in the AAU Smart Water Lab it cannot. However, in the real-world, a level sensor gate combination is often used. The idea is that a tank can be drained by the use of a gate, and the level sensor can be used to precisely estimate the volume before the gate is used. However, the gate setup does present a problem if it is to be used in conjunction with the MPC descriptions presented in this thesis. This is because the binary states of the gate, either open or closed, would lead to a mixed integer problem.

## 8.5 Disturbance design

In this section, we discuss our design of the disturbances for the lab experiments and how the uncertainties of the created disturbances are taken into account in the CC-MPC. We also specify the downsides of the approach taken to disturbance design in the thesis and present potential improvements. Additionally, we restate the considerations about the prediction horizon $Hp$.

In *subsection 7.1.1 Disturbance design* we described our approach to designing the disturbances for the lab experiment. The approach is again summarized here. The disturbance inflows consist of both domestic wastewater and rain water. We designed each of the disturbances separately before combining them into a single disturbance used in the lab. We end up with 10 ensembles for each disturbance in the system. The mean

signal found from the ensembles represents the disturbance forecast, while the variance between the ensembles defines the uncertainty of the forecast.

The rain inflow disturbances designed were not based on data. The rain inflow ensembles are designed manually with certain assumptions on how the storm event should be observed in the sanitation network. The variations between the ensembles are also created manually. An unrealistic assumption was made during the ensemble's creation. This assumption was that for every ensemble for a single rain event started at the same time and that there were no dry predictions for that particular rain event.

The limiting factor for disturbance design in the lab is the pumps' operating range. The rain inflow should be implemented on top of the wastewater discharge. For this reason, we reserved the 40 % of the range for wastewater inflow and the rest of the range for the rain inflows. In reality, the rain inflows are much higher than the wastewater inflows. In conclusion, the rain design should preferably be based on the actual data. We have been very restricted on obtaining real-world ensemble forecasts. The is because the forecasts were is simply not available in the precision we would need for use in a MPC framework.

In *subsection 7.2.5 Calculation of input and forecast uncertainty* we describe that we use a constant value for the disturbance variance $\Sigma_d$ at every step in the prediction horizon $Hp$. This value is calculated such that for the entire prediction $Hp$ ahead, the variance is averaged. Then the average variance is used in the state variance dynamics. Instead of averaging, we could have taken the actual variance at each step $k$ for the calculation of state variance dynamics. This would yield a nonlinear state variance propagation, which would result in different chance constraint tightening, than the one showed in the experimental results. The comparison of the tightening of the chance constraints with and without uncertainty averaging is shown in *Figure 8.5*. The tightening is compared at $1.5 \cdot 10^4$ $[0.5s]$ before the first overflow event which is shown in *Figure 7.24*.

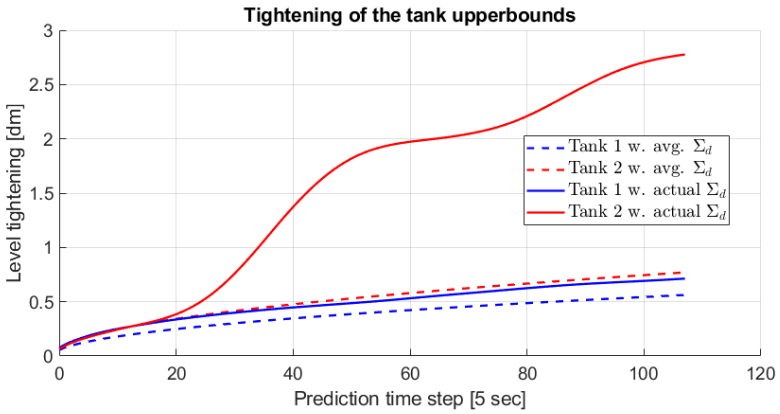



**Figure 8.5: Comparison of the tank level tightening with averaging $\Sigma_d$ and with actual $\Sigma_d$.**

We can see that during the uncertain prediction, the tightening on the second tank grows significantly, compared to the tightening with average variance. We specifically chose to compare the two tightenings at $1.5 \cdot 10^4$ $[0.5s]$, because of the very uncertain rain prediction in the middle of the pipe, see *Figure 7.24*. Furthermore, we expected the CC-MPC to

perform better than the standard MPC at this specific point in time.

The tightening reaches the 50% of the tank range towards the end of the prediction horizon, while previously, it was only at 2 to 3% of the tank range. For a detailed explanation of the impact additional constraint tightening can have on control performance, refer to *section 7.4 Comparison of Results*.

The tightening in *Figure 8.5* is calculated for $Hp = 12$ hours. This is equal to 108 steps in the lab experiment, when the step length is equal to $5s$. In *section 7.4 Comparison of Results* we stated that for better performance in regards to outflow smoothing, as well as for the better overflow handling, we would require a longer prediction horizon $Hp$. We also speculated that the longer $Hp$ would improve the performance of the CC-MPC. Now, with the new tightening calculation, shown in *Figure 8.5* and a longer $Hp$, it is even more apparent that the CC-MPC would be able to make use of the tightened constraint within the prediction horizon.

The satisfying length of the prediction horizon would be $Hp = 24$ hours. In the experimental setup, the single-day prediction horizon is equal to a 216 step prediction. Currently, we cannot achieve this $Hp$ in the experiments due to computational time constraints. Namely, the computational time for solving the MPC problem is limited to 1 second. With 108 steps prediction horizon, the computational time is between 0.5 and 0.8 seconds. We empirically validated that a longer than half-day prediction horizon is not feasible with a current experimental setup. The solution to the computational constraints would be the increase in the MPC's period from $5s$ to $10s$. That would enable us to increase the allowed computational time. However, this updated implementation would require a significant rework of the experimental setup and is left for future development.

# Conclusion 9

The purpose of this thesis was to compare the performance of a stochastic MPC scheme to the standard deterministic case, when used for overflow prevention in sanitation networks. In this thesis, we chose to use chance constrained MPC for comparing against the standard MPC scheme. These MPCs were implemented and tested in the AAU Smart Water Lab emulation environment. The purpose of using an emulation environment rather than a simulation setup was verifying that both MPC could work in a system subject to real-world inaccuracies.

Both MPCs had the same three operational objectives which they were compared upon. These three objectives are shown below in prioritized order:

1. Minimize overflow in a network.
2. Avoid fast varying flow to a Waste Water Treatment Plant.
3. Balance wastewater along available storage.

Along with designing the MPC control scheme, the following has been developed in the project as a supporting framework:

- A data-driven modelling framework, relying on the Diffusion Wave approximation of the nonlinear Saint Venant partial differential equations describing gravity flow inside the pipes,
- a grey-box system identification method, utilizing the level measurements taken inside the network, moreover incorporating the simplified, physically-based model structure of the pipes,
- furthermore a noise identification method to describe model- and measurement noise later utilized in state estimation with a Kalman filter.

We found that the diffusion wave model worked well for describing the flow through a gravity pipe. With the data driven parameter identification approach we were able to identify the parameters in the diffusion wave model. The parameter identification approach worked well for the Two Tank Topology, when all states in the model were measured. The identification approach does however require care when used to identify the diffusion wave model if the model contains unmeasured states. The Kalman filter was implemented and functioned using noise models derived based on an empirical approach. The Kalman filter was used in order to enable the use of MPCs which included models with unmeasured states.

The standard MPC has already been proven to be useful in control of sewer systems, we therefore elected to extend it to take the stochastic nature of sewer systems into account e.g rain events. The comparison of the standard MPC and the CC-MPC, did not yield a clear

conclusion on which is overall better in overflow prevention in sanitation networks.Based on our observations, the CC-MPC has a use in control sanitation networks if the available model is subject to large uncertainties and if computational resources allow for a long prediction horizon. However if computational resources are limited, or if a good predictions of disturbance inflow is available, we have not seen a reason to use the CC-MPC over the standard MPC. This is because the added complexity of the CC-MPC might actually decrease performance, due to being too conservative in critical situations, which can lead to overflow risk events.

# Bibliography


[1] 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.

[2] Camilla Molander. Influence of excessive water on wastewater treatment performance. Master's thesis, Chalmers University of Technology, 2015.

[3] Grundfos A/S. Optimised design of stormwater. `https://www.grundfos.com/content/dam/Global%20Site/Market%20areas/Water%20utility/Downloads/Brochures/design%20of%20Stormwater%20Tanks.pdf`. Accessed: 2020-16-09.

[4] Why control sanitary sewer overflows? `https://www3.epa.gov/npdes/pubs/sso_casestudy_control.pdf`. Accessed: 2021-17-02.

[5] Aalborg Kommune. Separatkloakering. `https://www.aalborg.dk/miljoe-energi-og-natur/spildevand-og-kloakering/separatkloakering`. Accessed: 2021-27-05.

[6] G Dirckx, M Schütze, S Kroll, Ch Thoeye, G De Gueldre, and B Van De Steene. Rtc versus static solutions to mitigate cso's impact. In *12nd International Conference on Urban Drainage, 2011b. Porto Alegre, Brazil*, 2011.

[7] Carlos Ocampo-Martinez. *Model predictive control of wastewater systems*. Springer Science & Business Media, 2010.

[8] Krisztian Mark Balla, Christian Schou, Jan Dimon Bendtsen, and Carsten Kallesøe. Multi-scenario model predictive control of combined sewer overflows in urban drainage networks. In *2020 IEEE Conference on Control Technology and Applications (CCTA)*, pages 611–618, United States, September 2020. IEEE. 2020 IEEE Conference on Control Technology and Applications (CCTA) ; Conference date: 24-08-2020 Through 26-08-2020.

[9] Jan Lorenz Svensen, Hans Henrik Niemann, Anne Katrine Vinther Falk, and Niels Kjølstad Poulsen. Chance-constrained model predictive control a reformulated approach suitable for sewer networks. *arXiv e-prints*, pages arXiv–2008, 2020.

[10] Rafal Wisniewski & Carsten Kallesøe Jorge Val Ledesma. Smart water infrastructures laboratory (swil). `https://vbn.aau.dk/da/equipments/smart-water-infrastructures-laboratory-swil#:~:text=The%20AAU%20Smart%20Water%20Infrastructures,Collection%20or%20District%20Heating%20Systems.` Accessed: 2021-27-05.

[11] Lukas Hewing, Juraj Kabzan, and Melanie N. Zeilinger. Cautious model predictive control using gaussian process regression. *IEEE Transactions on Control Systems Technology*, 28(6):2736–2743, Nov 2020.

[12] Su Ki Ooi and E. Weyer. Closed loop identification of an irrigation channel. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*, volume 5, pages 4338–4343 vol.5, 2001.

[13] Erik Weyer. System identification of an open water channel. *Control Engineering Practice*, 9(12):1289 – 1299, 2001.

[14] Krisztian Mark Balla, Carsten Kallesøe, Christian Schou, and Jan Dimon Bendtsen. Nonlinear grey-box identification with inflow decoupling in gravity sewers. In *Nonlinear Grey-Box Identification with Inflow Decoupling in Gravity Sewers*. IFAC Publisher, February 2020.

[15] Krisztian Mark Balla, Casper Houtved Knudsen, Adis Hodzic, Jan Dimon Bendtsen, and Carsten Kallesøe. Nonlinear grey-box identification of sewer networks with the backwater effect: An experimental study. 2021.

[16] Luca Vezzaro and Morten Grum. A generalised dynamic overflow risk assessment (dora) for real time control of urban drainage systems. *Journal of Hydrology*, 515:292–303, 2014.

[17] Adis Hodžić, Casper Knudsen, David Michalik, Frederik Rasmussen, and Jakob Poulsen. Model predictive control of combined sewer overflows in urban drainage networks. Masters project, Aalborg University, Institut for Elektroniske Systemer, 2020.

[18] Z. Yanggui, C. Lihui, L. Dewei, and H. Xingyao. Simplified state-space model and validation of irrigation canal systems. In *2015 34th Chinese Control Conference (CCC)*, pages 2002–2007, 2015.

[19] Violaine Dalmas, Gérard Robert, Gildas Besançon, and Didier Georges. Simplified non-uniform models for various flow configurations in open channels. *IFAC-PapersOnLine*, 50:12320–12325, 07 2017.

[20] X. Litrico and V. Fromion. Boundary control of linearized saint-venant equations oscillating modes. In *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, volume 2, pages 2131–2136 Vol.2, 2004.

[21] Carsten Skovmose Kallesøe. Canal flow (saint venant equations). 2020.

[22] G. Evans, J. Blackledge, and P. Yardley. *Numerical Methods for Partial Differential Equations*. Springer, 2000.

[23] Vito Ferro. Theoretical end-depth-discharge relationship for free overfall. *Journal of Irrigation and Drainage Engineering-asce - J IRRIG DRAIN ENG-ASCE*, 125, 01 1999.

[24] Xavier Litrico and Vincent Fromion. *Modeling and Control of Hydrosystems*. Springer, 2009.

[25] Abbas Emami-Naeini Gene F. Franklin, J. David Powell. *Feedback Control of Dynamic Systems 8th edition*. 8 ed. Pearson, 0.

[26] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. 3 ed. Pearson, 2014.

[27] Said Kouachi. Eigenvalues and eigenvectors of some tridiagonal matrices with non-constant diagonal entries. *Applicationes Mathematicae*, 35:107–120, 01 2008.

[28] Said Kouachi. Eigenvalues and eigenvectors of tridiagonal matrices. *ELA. The Electronic Journal of Linear Algebra [electronic only]*, 15, 03 2006.

[29] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2013.

[30] A. Wills and B. Ninness. On gradient-based search for multivariable system estimates. *IEEE Transactions on Automatic Control*, 53(1):298–306, 2008.

[31] Mohinder S. Grewal and Angus P. Amdrews. *Kalman Filtering Theory and Practice Using MATLAB*. WILEY, 2008.

[32] Jan Marian Maciejowski. *Predictive control: with constraints*. Prentice Hall, 2002.

[33] William S Levine, Lars Grüne, Rafal Goebel, Saša V Rakovic, Ali Mesbah, Ilya Kolmanovsky, Stefano Di Cairano, Douglas A Allan, James B Rawlings, Martin A Sehr, et al. Handbook of model predictive control. 2018.

[34] Danish Meteorological Institute. Numerical weather prediction. `http://research.dmi.dk/research/research-topics/numerical-weather-prediction/?fbclid=IwAR1JxnXq4uK3ALSCuTGJ6R36rpqtTtmEDPLJ6knc3gbgAENb0HrnwTg338A`. Accessed: 2021-02-06.

[35] Roland Löwe, Peter Steen Mikkelsen, and Henrik Madsen. Stochastic rainfall-runoff forecasting: parameter estimation, multi-step prediction, and evaluation of overflow risk. *Stochastic environmental research and risk assessment*, 28(3):505–516, 2014.

Project Proposal for 9th and 10th semester CA students at Aalborg University

# Stochastic Model Predictive Control for Combined Sewer Overflows in Urban Drainage Networks

Contacts:
Carsten S. Kallesøe (Supervisor, e-mail: csk@es.aau.dk)
Krisztian M. Balla (Co-supervisor: kmbal@es.aau.dk)
Rahul Misra (Co-supervisor: rmi@es.aau.dk )

## 1. Problem statement

Open-channel hydraulic systems, e.g. Urban Drainage Networks (UDNs), are among the most vital infrastructures within the urban water cycle. More specifically, combined UDNs carry domestic wastewater and rain run-off towards treatment plants, where the sewage is treated before it is discharged to the environment. Knowing the flow in such systems is essential to optimize storage and thus avoid intermittent overflows [1]. However, an issue with UDNs is the lack of flow measurements, although optimization-based control typically relies on flow balances within the network. Hence, a possible way to build control models and reconstruct system states is to utilize cost-efficient level sensors and to support the control with online estimation algorithms.

Another issue with the control of UDNs is the presence of hydrological and meteorological phenomena, implying that the system processes are influenced by disturbances, stochastic in their nature. The main goal of the regulation of UDNs is to minimize the effect of these disturbances (e.g. rain infiltration, domestic sewage and groundwater infiltration), over a system which under normal operation would not require any action. Hence, control needs to be robust towards weather forecasts and towards the uncertainty of the rain-runoff dynamics [2].

A third issue with the control of UDNs is the fact that they are increasingly being pushed to their capacity limits. This is partly due to changing weather conditions, resulting in increased amount and more frequent high-intensity rain events, for which standard control strategies (e.g. On/Off level control) and the old infrastructure are not prepared.

## 2. Project tasks

The project plan can include the following tasks, depending on whether it is a semester project or long thesis:

1. System identification:
   System identification (Grey-box) of gravity pipes, relying on the physical laws described by the Saint-Venant hyperbolic Partial Differential Equations. The test and performance assessment of different system identification and sensor setup configurations can be carried out. It is anticipated that the fusion of level sensor data and pump discharge flow data is used to obtain model parameters. Some results on this subject are available in [3].
2. State estimation:
   Full state observability in sewer applications is typically not available, hence only some subsets of states are measured. In this practical setup, the unmeasured states need to be computed consistent with the measured states and outputs. This mainly regards the issue of limited number of level sensors along the gravity channels. If time allows, the State Estimation Problem (SEP) can be extended to Moving Horizon Estimation (MEH), which suits well the proposed MPC framework. Studies on SEP and MEH problems within sewer applications can be found in [4].
3. Stochastic Model Predictive Control:
   Although standard MPC methods offer a certain degree of robustness to system uncertainties due to their receding-horizon implementation (i.e. rainfall precipitation, rain-runoff dynamics), stochastic MPC is a natural extension to deal with the uncertainties systematically [5].

In this project, uncertainty is introduced on inflows to pumping stations due to both rainfall runoff and to some periodic domestic waste water activities. In [6] and [7], three form of stochastic MPC approaches have been implemented on drinking water distribution and hydrogen-based microgrid applications. It is a suggestion in this project to try to implement one or more approaches (i.e. Multi-Scenario(MS) MPC, Tree-based(TB) MPC and Chance-Constrained(CC) MPC) and compare their performances on the proposed UDN application. An MS-MPC study on a drainage network case study can be found in [8].

4. Multi-objective optimization:
   Control in UDNs is typically proposed as a multi-criteria optimization with conflicting objectives. It is a possibility to investigate and analyze the operational objectives and the choice of weighing parameters regarding the optimization. The operational objectives consist of: overflow prevention, smoothing of treatment plant inlet flow and equal use of available storage. For more detailed study in the systematical design of optimization in sewer networks, consult [9].

An idea of the implemented control on the AAU Smart Water lab is shown in Figure 1.

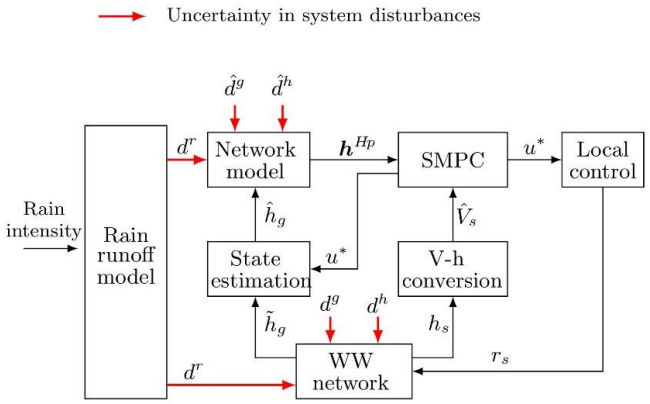



Figure 1- An idea of a closed-loop control loop where SMPC abbreviates Stochastic Model Predictive Control, moreover the red arrows denote the points where uncertain system disturbances enter the system.

Prerequisites:

1. Basic understanding of the modeling of hydraulic systems, such as drinking water distribution and open-channel hydraulic networks. (The knowledge gained by the 8th semester course: “Modeling of Mechanical and Thermal Systems”.)
2. Basic knowledge in Model Predictive and Optimal Control. (The knowledge gained by the 8th semester course: “Optimality and Robustness”.)

## 3. Experimental 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.

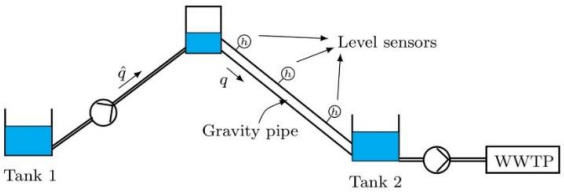



Figure 2 – Sketch of the proposed network topology.

In this project, a network with two retention tanks are considered, connected by a gravity pipe, where the stored sewage volumes are controlled by pumps. Furthermore, this topology can be modified along with the points where the disturbances enter the system. An illustration of the laboratory setup is shown in Figure 3.

Figure 3- Modular Water Laboratory at Aalborg University.

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. Implementation and real-time test of basic on-off controllers for later comparison with MPC.
4. Implementation and real-time test of the developed SMPC/SEP/MEH algorithms.

## 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] 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.

[3] Balla, K. M., Schou, C., Bendtsen, J. D., & Kallesøe, C. (2020). Nonlinear Grey-Box Identification with In-flow Decoupling in Gravity Sewers. Accepted at the 21st IFAC World Congress (IFAC).

[4] Joseph-Duran, B., C. Ocampo-Martinez, and G. Cembrano (2015), Output-feedback control of combined sewer networks through receding horizon control with moving horizon estimation, Water Resour. Res., 51,8129–8145, doi:10.1002/2014WR016696

[5] A. Mesbah, "Stochastic Model Predictive Control: An Overview and Perspectives for Future Research," in *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30-44, Dec. 2016, doi: 10.1109/MCS.2016.2602087.

[6] Grosso, J., Velarde Rueda, P., Ocampo-Martinez, C., Maestre, J.M. & Puig, V. (2016). Stochastic model predictive control approaches applied to drinking water networks. Optimal Control Applications and Methods. 38. 10.1002/oca.2269.

[7] Velarde, P., Valverde, L., Maestre, J., Ocampo-Martinez, C.A., Bordons, C. On the comparison of stochastic model predictive control strategies applied to a hydrogen-based microgrid. "Journal of power sources", 1 Març 2017, vol. 343, p. 161-173.

[8] Balla, K. M., Schou, C., Bendtsen, J. D., & Kallesøe, C. (2020). Multi-scenario Model Predictive Control of Combined Sewer Overflows in Drainage Networks. Accepted at 2020 IEEE Conference on Control Technology and Applications (CCTA).

[9] A. L. Mollerup, P. S. Mikkelsen, and G. Sin, “A methodological approach to the design of optimising control strategies for sewer systems,” Environmental Modelling and Software, vol. 83, pp. 103–115, 2016.

# Consistency and Convergence of The General Pipe Model

B

A finite difference scheme used for approximating partial differential equations is said to be consistent if the resulting partial difference equation approaches the differential equation as $\Delta x,\ \Delta t \to 0$. This implies that consistency is the condition on the numerical scheme [22,pages. 43-45].

Convergence refers to the approach of the solution of an approximating set of linear difference equations to the solution of a linear partial differential equation as $\Delta x,\ \Delta t \to 0$. This implies that convergence is the condition on the numerical solution [22,pages. 46-48].

The analysis of consistency is straight forward as backward and central finite differential approximations were used for creating the difference equations describing the pipes. These three differential approximations are simply the limit definitions of the derivatives, excluding the limit term. We know that the limit definition of a partial differential is given as:

$$\frac{\partial f(x,t)}{\partial x} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x,t) - f(x,t)}{\Delta x}. \tag{B.1}$$

Leading to the definition of the second partial derivative as a derivative of derivatives:

$$\begin{aligned} \frac{\partial^2 f(x,t)}{\partial x^2} &= \lim_{\Delta x \to 0} \frac{\frac{f(x+\Delta x,t)-f(x,t)}{\Delta x} - \frac{f(x,t)-f(x-\Delta x,t)}{\Delta x}}{\Delta x}, \\ &\Updownarrow \\ \frac{\partial^2 f(x,t)}{\partial x^2} &= \lim_{\Delta x \to 0} \frac{f(x+\Delta x,t) - 2f(x,t) + f(x+\Delta x,t)}{\Delta x^2}. \end{aligned}$$

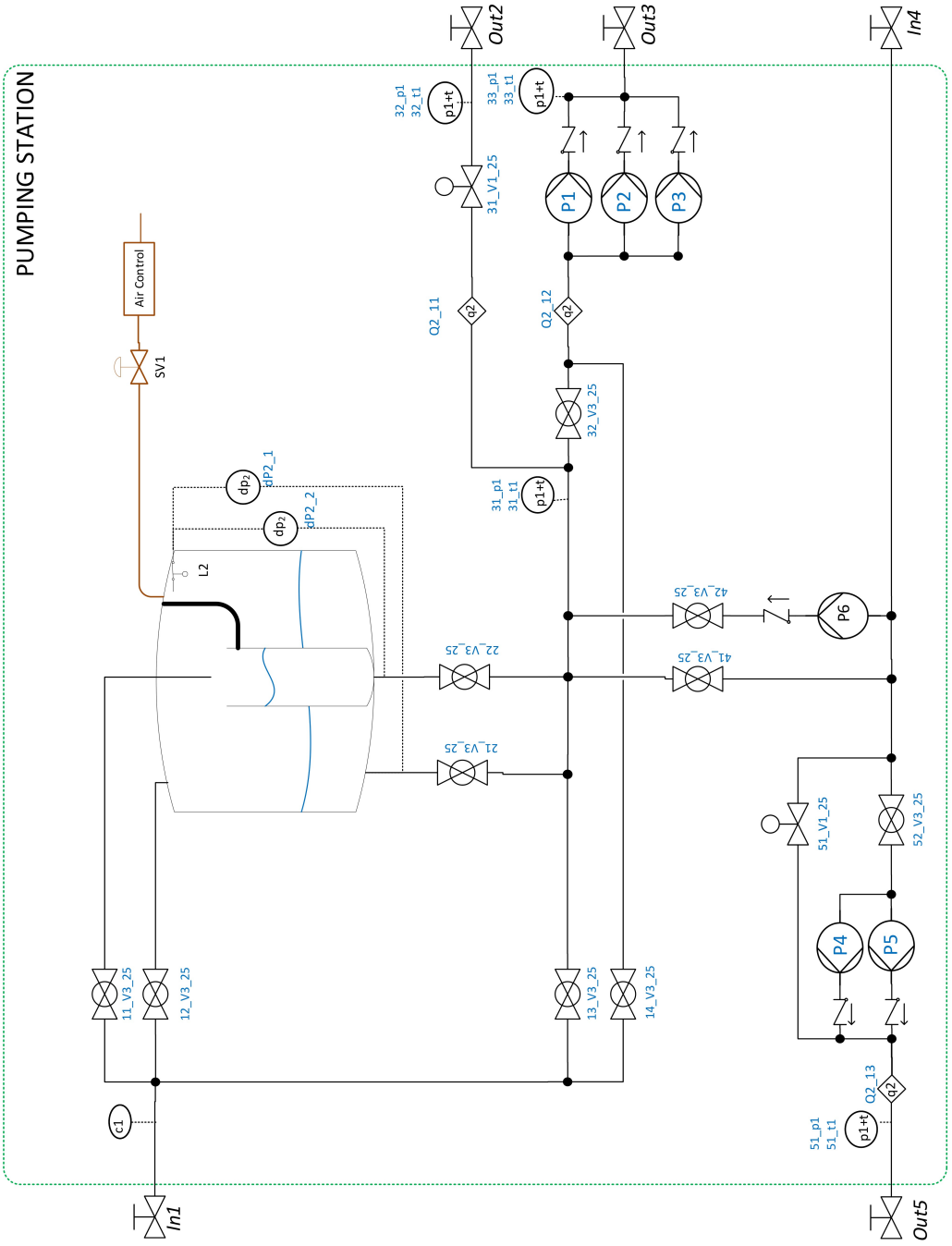
Comparing these with the difference equation describing the pipe, see (B.2) and see *subsection 2.1.1 Gravity pipe*, it can be seen that introducing the limits yields the differential equation.

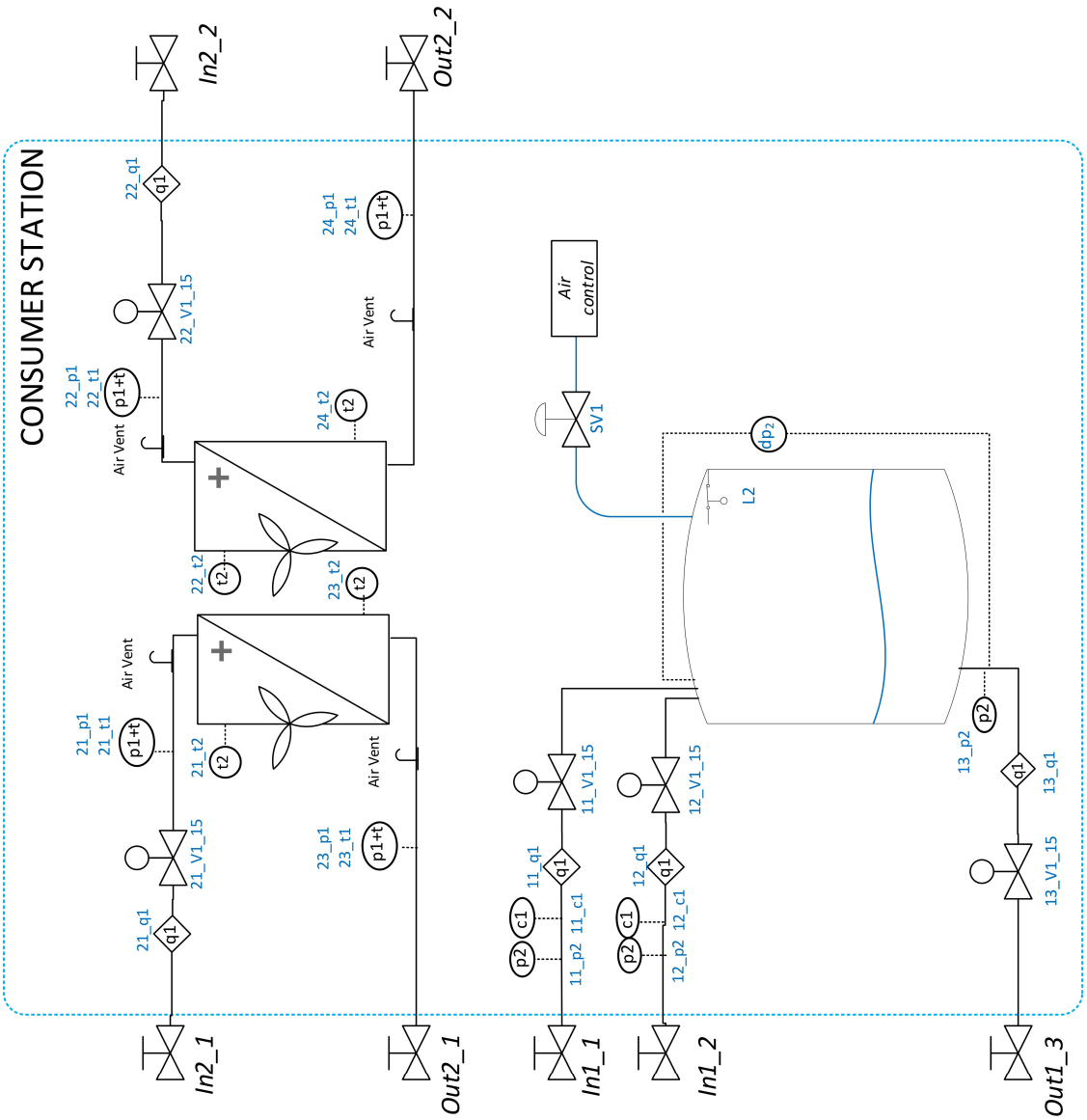
$$\begin{aligned} \frac{h_j^{n+1} - h_j^n}{\Delta t} &= \frac{1}{\gamma_3 \cdot w}\left(\frac{h_{j-1}^n - 2 \cdot h_j^n + h_{j+1}^n}{\Delta x^2} + \gamma_3 \frac{h_j^n - h_{j-1}^n}{\Delta x}\right) \qquad \text{(B.2)} \\ &\Downarrow \\ \lim_{\Delta x, \Delta t \to 0} \frac{h_j^{n+1} - h_j^n}{\Delta t} &= \lim_{\Delta x, \Delta t \to 0} \frac{1}{\gamma_3 \cdot w}\left(\frac{h_{j-1}^n - 2 \cdot h_j^n + h_{j+1}^n}{\Delta x^2} + \gamma_3 \frac{h_j^n - h_{j-1}^n}{\Delta x}\right) \\ &\Updownarrow \\ \frac{\partial h(x,t)}{\partial t} &= \frac{1}{\gamma_2 \cdot w}\left(\frac{\partial^2 h(x,t)}{\partial x^2} + \gamma_3 \frac{\partial h(x,t)}{\partial x}\right) \end{aligned}$$

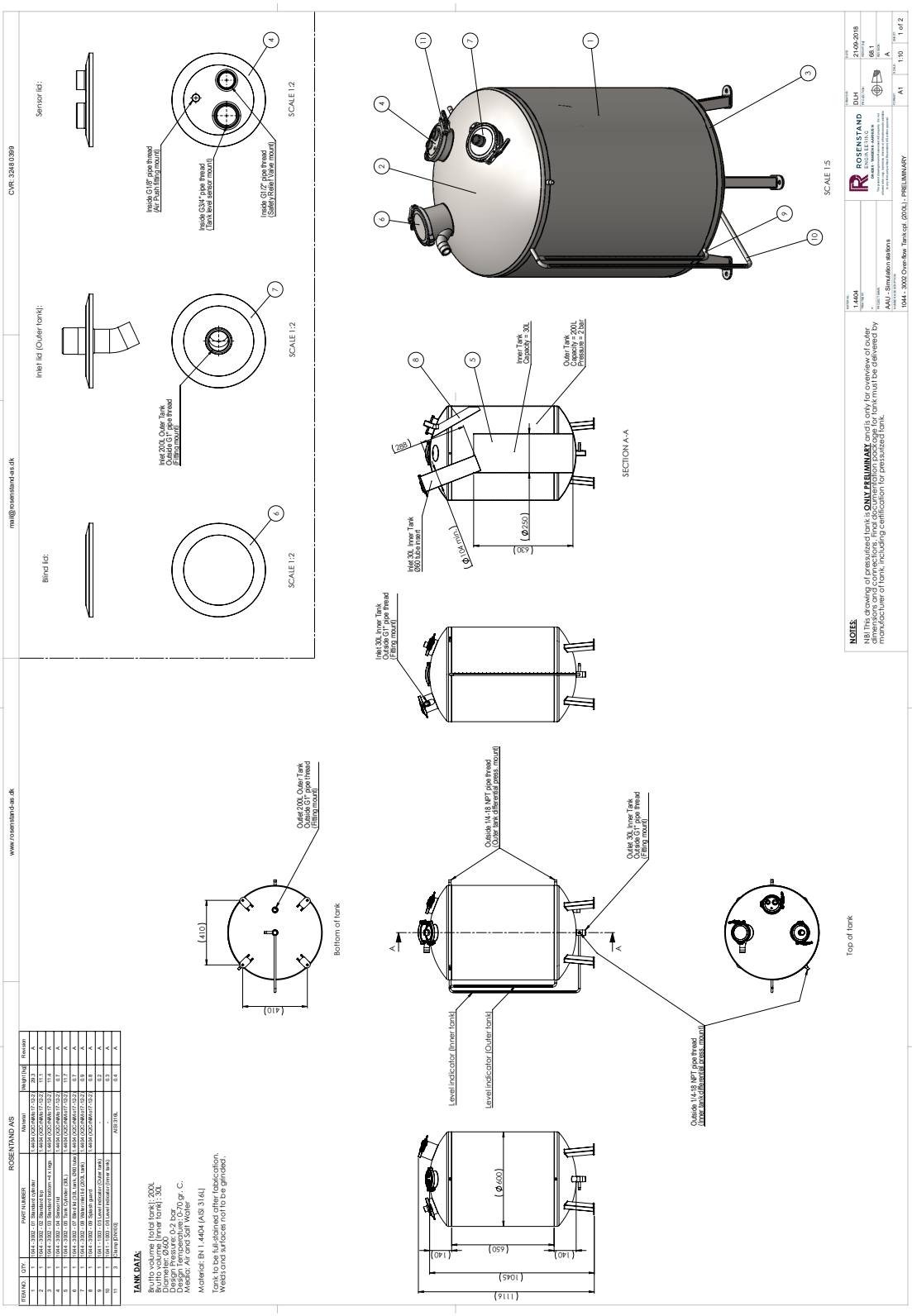
This means that the finite difference scheme used is consistent. This does however not ensure that the numerical solutions of the difference equation are correct. For that we need to prove convergence. This can be done using Lax's stability theorem which states: "If a linear finite difference equation is consistent with a properly posed initial-value problem then stability is necessary and sufficient for convergence of $\boldsymbol{\phi}$ to $\boldsymbol{\Phi}$ as mesh lengths tend to zero" [22]. Where $\boldsymbol{\phi}$ is the numerical solution of the approximated difference equation, and $\boldsymbol{\Phi}$ is the solution of the partial differential equation.

In the case of the gravity pipe portion of our system, in *subsection 2.2.1 Stability of the General Model* named $\boldsymbol{A_2}$ , Lax's stability theorem cannot be directly applied. This is because while consistency has been proven, but we do not have a condition on stability.

# Lab Documentation C




CONSUMER STATION
In2_2
Out2_2
In2_1
Out2_1
In1_1
In1_2
Out1_3
Air control
SV1
L2
Air Vent
22_q1
22_V1_15
22_p1
22_t1
24_p1
24_t1
24_t2
22_t2
23_t2
21_p1
21_t1
21_t2
21_V1_15
21_q1
23_p1
23_t1
11_V1_15
11_q1
11_p2
11_c1
12_V1_15
12_q1
12_p2
12_c1
13_p2
13_q1
13_V1_15




| ITEM NO. | QTY. | PART NUMBER | Name | Weight [kg] | Revision |
|---|---|---|---|---|---|
| 1 | 1 | 1044 - 3002 - 01 - Standard cylinder | 1.4404 (OCr18Ni9) (1.7-12.2 | 29.1 | A |
| 2 | 1 | 1044 - 3002 - 02 - Standard lid | 1.4404 (OCr18Ni9) (1.7-12.2 | 11.1 | A |
| 3 | 1 | 1044 - 3002 - 03 - Standard bottom +4 x legs | 1.4404 (OCr18Ni9) (1.7-12.2 | 11.4 | A |
| 4 | 4 | 1044 - 3002 - 04 - Leg holder | 1.4404 (OCr18Ni9) (1.7-12.2 | 0.4 | A |
| 5 | 1 | 1044 - 3002 - 05 - Tank (Outer) (30L) | 1.4404 (OCr18Ni9) (1.7-12.2 | 11.7 | A |
| 6 | 1 | 1044 - 3002 - 06 - Blind lid (Outer tank) | 1.4404 (OCr18Ni9) (1.7-12.2 | 3.1 | A |
| 7 | 1 | 1044 - 3002 - 07 - Blind lid (30L tank, Ø60 hole) | 1.4404 (OCr18Ni9) (1.7-12.2 | 0.7 | A |
| 8 | 1 | 1044 - 3002 - 08 - Inlet lid (Outer tank) | 1.4404 (OCr18Ni9) (1.7-12.2 | 3.1 | A |
| 9 | 1 | 1044 - 3002 - 09 - Sensor lid (30L tank) | 1.4404 (OCr18Ni9) (1.7-12.2 | 0.8 | A |
| 10 | 1 | 1041 - 1003 - 01 Level indicator (Outer tank) | - | 0.2 | A |
| 11 | 1 | 1041 - 1003 - 02 Level indicator (Inner tank) | AISI 316L | 0.3 | A |
| 12 | 3 | Hexagon bolts | | 0.4 | A |

# Derivation of Cost Function for a Stochastic MPC

D

From the classical formulation of an MPC we take the expectation of the cost function yielding:

$$\mathcal{J} = \min_{\boldsymbol{u}} \mathbf{E}\left[\sum_{k=1}^{H_p}\left\|\mathbf{z}_k - \mathbf{z}_k^{ref}\right\|_{\mathbf{Q}_k}^2 + \sum_{k=0}^{Hu-1}\|\boldsymbol{\Delta}\mathbf{u}_k\|_{\mathbf{R}_k}^2\right] \tag{D.1}$$

$$= \min_{\boldsymbol{u}} \sum_{k=1}^{H_p}\mathbf{E}\left[\left\|\mathbf{z}_k - \mathbf{z}_k^{ref}\right\|_{\mathbf{Q}_k}^2\right] + \sum_{k=0}^{Hu-1}\mathbf{E}\left[\|\boldsymbol{\Delta}\mathbf{u}_k\|_{\mathbf{R}_k}^2\right] \tag{D.2}$$

It is possible to show that the expectation of a quadratic form random variable, can be rewritten in terms of its mean and variance, see the hand written note in this appendix

$$\mathbf{E}\left[\|\mathbf{X}\|_{\mathbf{Q}}^2\right] = \|\mu_{\mathbf{X}}\|_{\mathbf{Q}}^2 + tr(\mathbf{Q}\boldsymbol{\Sigma}_{\mathbf{X}}), \tag{D.3}$$

see the hand written note in this appendix. Then if we assume that $z_k$ is a random variable while the control input $u_k$ and reference $\mathbf{z}_\mathbf{k}^\mathbf{ref}$ is deterministic:

$$\mathcal{J} = \min_{\boldsymbol{u}} \sum_{k=1}^{H_p}\left\|\mu_{\mathbf{z}_k} - \mathbf{z}_k^{ref}\right\|_{\mathbf{Q_k}}^2 + tr(\mathbf{Q}\cdot var(\mathbf{z}_k - \mathbf{z}_k^{ref})) + \sum_{k=0}^{Hu-1}\mathbf{E}\left[\|\boldsymbol{\Delta}\mathbf{u}_k\|_{\mathbf{R}_k}^2\right] \tag{D.4}$$

$$= \min_{\boldsymbol{u}} \sum_{k=1}^{H_p}\left\|\mu_{\mathbf{z}_k} - \mathbf{z}_k^{ref}\right\|_{\mathbf{Q_k}}^2 + tr(\mathbf{Q}\cdot\boldsymbol{\Sigma}_{\mathbf{z}_k}) + \sum_{k=0}^{Hu-1}\|\boldsymbol{\Delta}\mathbf{u}_k\|_{\mathbf{R}_k}^2 \tag{D.5}$$

If we on the other hand assume that there is uncertainty in the way we apply the control input:

$$\mathcal{J} = \min_{\boldsymbol{u}} \sum_{k=1}^{H_p}\left\|\mu_{\mathbf{z}_k} - \mathbf{z}_k^{ref}\right\|_{\mathbf{Q_k}}^2 + tr(\mathbf{Q}\cdot\boldsymbol{\Sigma}_{\mathbf{z}_k}) + \sum_{k=0}^{Hu-1}\left\|\mu_{\boldsymbol{\Delta}\mathbf{u}_k}\right\|_{\mathbf{R_k}}^2 + tr(\mathbf{R}\cdot\boldsymbol{\Sigma}_{\boldsymbol{\Delta}\mathbf{u}_k})$$

We want to re-express the expectation of a quadratic random variable, in terms of a trace.

$$E\left[\|X\|_Q^2\right] = E[\underbrace{X^\top Q X}_{\text{scalar}}] \tag{D.6}$$

The matrix vector product written on general form:

$$QX = \sum_{j=1}^{n} x_j q_j \tag{D.7}$$

Where $q_j$ is a column vector. This means that the vector vector product can be expressed as:

$$X^\top Q X = \sum_{i=1}^{n} x_i \sum_{j=1}^{n} x_j q_{j,i} = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j q_{j,i} \tag{D.8}$$

This means:

$$\begin{aligned} E\left[X^\top Q X\right] &= \sum_{i=1}^{n}\sum_{j=1}^{n} E\left[x_i x_j\right] q_{j,i} \\ &= \sum\sum \left(\sigma_{q_{i,j}} + \mu_i \mu_j\right) q_{j,i} \\ &= \sum\sum \sigma_{j,j} \cdot q_{i,j} + \sum\sum \mu_j q_{j_1 i} \\ &= \underbrace{\sum\sum \sigma_{i,j} q_{i,j}}_{\text{Definition of trace}} + \mu^\top Q \mu \\ &= \operatorname{tr}(Q\Sigma) + \mu^\top Q \mu \end{aligned} \tag{D.9}$$

# Supplementary experiment E

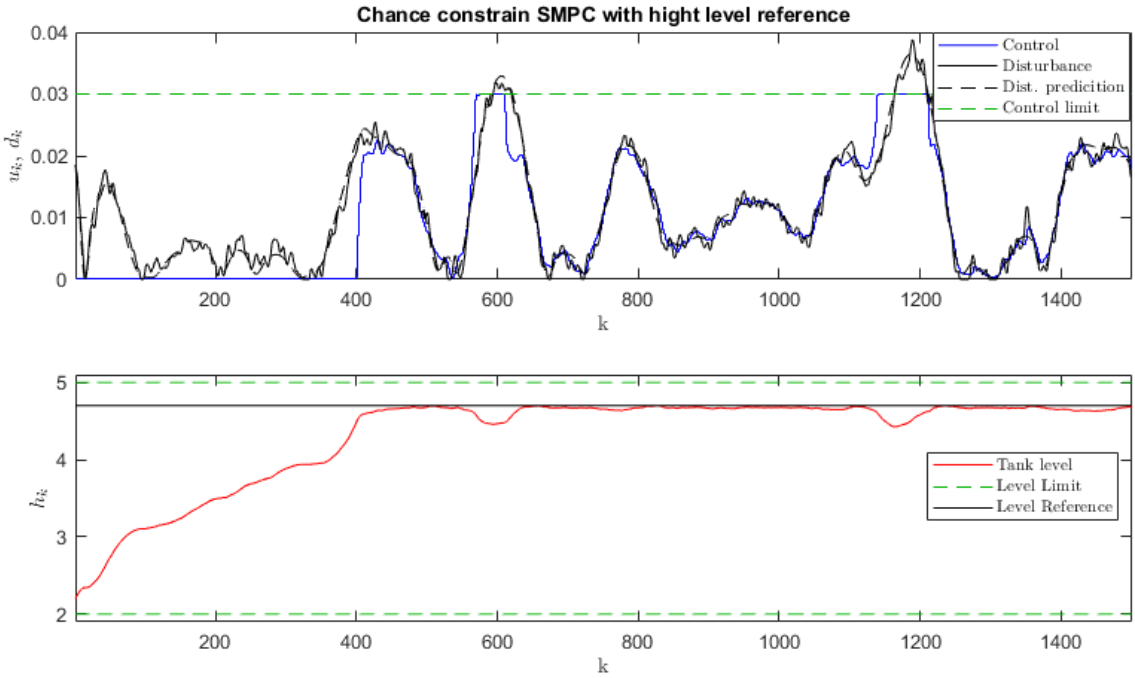



**Figure E.1: Single tank controller conservatives due to chance constraint tightening, supplementary example**

This example supports the *section 6.3 Single tank SMPC*, where we present the conservatives of SMPC when the system is operated near the physical upper bound. There we conclude with a high reference tracking setup that the controller underperforms and is not able to follow the high reference. In this example it is obvious that a controller reasonably follows the reference most of the time. The difference is in the objective function. Two objective functions can be seen on (E.1) and (E.2).

$$\mathcal{J}_1 = \sum_{k=0}^{H_p-1} 0.01(\mu_{h_k} - h_{ref|k})^2 + 0.01\Sigma_{h_k} + 40u_k^2, \tag{E.1}$$

$$\mathcal{J}_2 = \sum_{k=0}^{H_p-1} 0.01(\mu_{h_k} - h_{ref|k})^2 + 0.01\Sigma_{h_k} + 0.4u_k^2, \tag{E.2}$$

where $\mathcal{J}_1$ is the objective function for the example presented in *section 6.3 Single tank SMPC* and $\mathcal{J}_2$ is the objective function for *Figure E.1*. The control action in the later is

more aggressive. We can see that if we compare *Figure E.2* with *Figure 6.2b*.

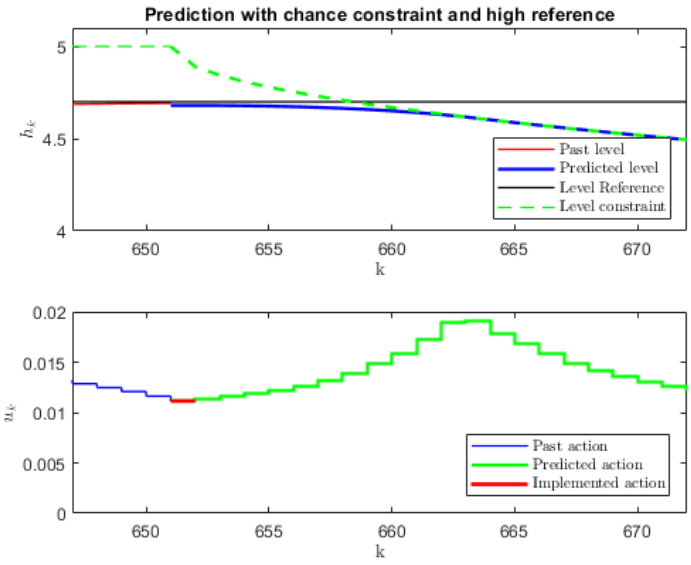



**Figure E.2: Level prediction with more aggressive controller**

The purpose of this supplementary example is to support the idea that the conservatives of a SMPC presented in the main part can be reduced with more aggressive action. However, we can still observe cautious behaviour around 600 and 1200 steps where controller preemptively lowers the level before the high disturbance arrives. But, we consider this example not as representative of the phenomena as the one that is presented in the *section 6.3 Single tank SMPC*.