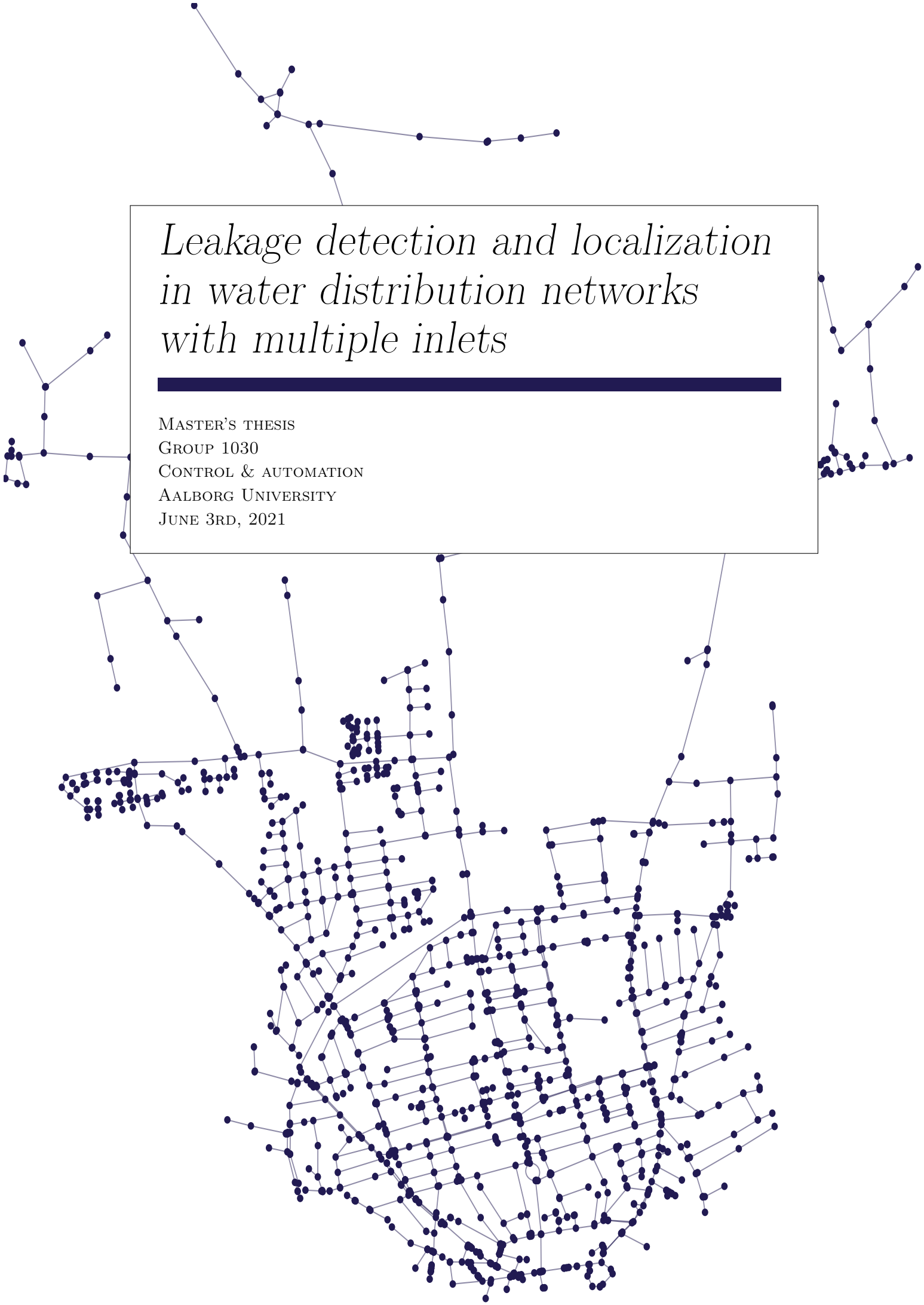
Leakage detection and localization in water distribution networks with multiple inlets

Master's thesis
Group 1030
Control & automation
Aalborg University
June 3rd, 2021



AALBORG
UNIVERSITY

STUDENT REPORT

Titel:

Leakage detection and localization in water distribution networks with multiple inlets

Project:

Master's thesis

Projectperiod:

01/02-2021 - 03/06-2021

Projectgroup:

1030

Participants:

Esben H. Nielsen
Mark Christensen
Simon N. Gnap

Supervisors:

Carsten S. Kallesøe
Saruch S. Rathore

Page number: 95
Page number with appendix: 122
Finished: 03-06-2021

Department of Electronic Systems
Control & Automation
Fredrik Bajers Vej 7B
9220 Aalborg
http://www.es.aau.dk/

Synopsis:

Freshwater is increasingly becoming a limited resource. That makes efficient Water Distribution Networks (WDNs) key. However, in some countries, up to 60% of water is lost during distribution, mainly caused by leaks. Leaks can be difficult to detect and localize. This report proposes a novel method for detecting and localizing leaks in a WDN with multiple inlets.
The method uses a residual that changes in mean in case of a leak. The residual is generated from the difference between a sensor measurement and an estimated value, coming from a novel steady-state WDN model. The residual contains noise, making it difficult to detect a change in mean. This is overcome using the method Generalized Likelihood Ratio. For leakage localization, a sensitivity matrix is used to generate hypothetical residuals for a leak at each node. Comparing the direction of these to the direction of the real residual nodes likely to be leaky is identified.
The proposed method is verified to work in simulation. However, results on a small scale physical setup indicate some practical issues properly caused by poor parameter estimation. This lead to the conclusion that great care must be taken when obtaining these. Further, the robustness of the method towards model inaccuracies might need to be increased.

Preface

This report is composed by group 1030, 4th semester of the Control and Automation master at Aalborg University in the spring of 2020.

The authors would like to thank the supervisors, Carsten S. Kallesøe and Saruch S. Rathore, for their excellent supervision. Especially for always having a positive and open-minded attitude toward suggestions and materials made by the group and only coming with constructive comments. Further, the authors would like to thank Jorge V. Ledesma for giving the authors an introduction to the Smart Water Infrastructure Lab (SWIL).

Readers guide

The report is intended to be read in chronological order. Figures (Fig), tables (Tab), assumptions (Asm) and equations (eq) is referred to by their number and their identifier in bold, like for instance: **Fig. 2.1**. Similarly, a chapter is referred to as **Ch. x** and a section is referred to as **Sec. x.y** or **Sec. x.y.z**. For citations an author-year style named APA style is used. This report uses in general SI-units unless else is stated. Many plots will, however, use alternative units as these are easier to understand. This typically means changing pa, sec and m^3/s to bar, h and m^3/h

Esben H. Nielsen

Mark Christensen

Simon N. Gnap

List of constants and variables

Here is a list containing the most important constant and variables used in the modelling of the system.

Variables		
Variable name	Description	Unit
a	Flow mapping vector	$\cdot$
A	Area or State matrix	m^2 or $\cdot$
B	Fundamental loop matrix or input matrix	$\cdot$ or $\cdot$
c	Number of inlets in the WDN	$\cdot$
C	Output matrix	$\cdot$
d	Demand	m^3/s
d_f	Demand at open vertices	m^3/s
d_l	Demand of a leakage	m^3/s
d_{lm}	Magnitude of a leakage	m^3/s
D	Diameter or feedforward matrix	m or $\cdot$
e	Single edge	$\cdot$
$\mathcal{E}$	Set of edges	$\cdot$
f	Pipe friction factor	$\cdot$
F	Free flow demand picking matrix	$\cdot$
$\mathcal{G}$	Graph consisting of vertices and edges	$\cdot$
h	Head	m
H	Incidence matrix	$\cdot$
J	Inertance matrix	kg/m^4
k_f	Form loss coefficient	$\cdot$
$k_{f,m}$	Form loss coefficient per meter	$1/\mathrm{m}$
k_0	Sample number of leakage	$\cdot$
K	Feedback gain	$\cdot$
K_i	Integral gain	$\cdot$
l	Number of chords/loops	$\cdot$
$\hat{l}$	Estimated leaking node	$\cdot$
L	Length	m
m	Number of edges in the graph	$\cdot$
M	Demand multiplier	$\cdot$
M_w	Window length	$\cdot$
n	Number of nodes in the graph	$\cdot$
OD	Valve opening degree	%
p	Pressure	Pa
q	Component flow	m^3/s

r	Residual	Pa
ref	Reference for flows or pressure	m^3/s or Pa
Re	Reynolds number	·
S	Sensitivity matrix	·
t	Time	s
t_0	Time of leakage	s
th	Threshold	·
u	Input vector	·
v	Demand distribution vector	·
v	Single vertex/node	·
$\mathcal{V}$	Set of vertices/nodes	·
x	State vector	·
z	Geodesic level or sample	m or ·
α	Estimated coefficients	kg/m^7
β	Estimated coefficients	Pa
γ	Reference node demand multiplier	·
Γ	Combined pressure across pipes and valves	Pa
κ	Simplified pipe friction coefficient	kg/m^7
λ	Pressure drop over pipes caused by friction	Pa
μ	Pressure drop over valves	Pa
ζ	Pressure difference over pumps	Pa
σ	Total consumer consumption	m^3/s
ϕ	Decision function	·
ω	Angular velocity	Hz
ψ	Decision signal	·
$\bar{\psi}$	Truncated decision signal	·
$\mathcal{H}_0$	Hypothesis of no change in mean	·
$\mathcal{H}_1$	Hypothesis of a change in mean	·

Table 1: List of variables

Constants			
Variable name	Description	Value	Unit
a_{h0}	Pump coefficient	$8.8011 \cdot 10^{-4}$	·
a_{h1}	Pump coefficient	11.3179	s/m^2
a_{h2}	Pump coefficient	$3.9205 \cdot 10^6$	s^2/m^5
g	Gravitational acceleration	9.82	m/s^2
k_{vs}	Valve resistance at max. opening	1	m^3/h
ε	Pipe roughness	$0.05 \cdot 10^{-3}$	m
ρ	Density of water at 10°C	998	kg/m^3
ϑ	Kinematic viscosity of water at 10°C	$1.3 \cdot 10^{-6}$	m^2/s

Table 2: List of constants

Variable subscripts	
Subscript name	Description
i	Component counter
n	The reference node
d	Delayed state space model
c	Concatenated state space model
r	Reduced state space model
ci	Concatenated state space model with integral states
$\mathcal{C}$	The chords of a graph
$\mathcal{T}$	Spanning tree of a graph

Table 3: List of subscripts

Variable accents	
accent name	Description
$\hat{x}$	Estimated value of x
$\bar{x}$	Vector/matrix with reference row removed
$\dot{x}$	Time derivative of x
$\breve{x}$	Nominal value of x
δx	Deviation of x caused by leakage
$\tilde{x}$	Small signal deviation of x
$x^\star$	Operating point of x
$\acute{x}$	Non-inlet nodes contained in x
$\grave{x}$	Inlet nodes contained in x
Δx	Difference in x over an edge

Table 4: List of accents

Contents

Introduction 1

Freshwater is necessary for humans to survive. In most areas, large distribution networks for supplying water from a freshwater source to the tap in your house has been made. This network is called a water supply system (WSS).

One of the reoccurring problems in WSSs around the world is that of water lost to leakages in the pipes. To repair a leakage in the network the location of the leakage must first be known. This is often very difficult and time consuming by the fact that most of the pipes are buried under ground. Because of this, a leakage can be present in the network for a long time before it is remedied.

As a consequence of this, large amounts of clean drinking water is lost every year around the world and in some countries 30 to 60 percent of the produced water is lost in the distribution network [Miljøstyrelsen, n.d.]. While the average loss percentage across Denmark is 7.29% [Danva, 2020] and, with that, among the lowest in the world, this still amounts to a large amount of wasted water every year. This not only means that clean drinking water is wasted but energy as well, since it requires energy to pump the water to the surface and to treat it to be drinkable. This problem is addressed by one of the global Sustainable Development Goals made by the UN, namely goal 6.4 which addresses the need for increased efficiency of water use and ensure sustainable supply of drinkable water. Here, the common goal is to achieve reduced water scarcity and hereby reduce the number of people suffering from this [UN, 2015].

The aim of this report is to investigate methods for detecting and localizing leakages in a water distribution network. These methods should be a supplement or improvement to currently existing methods. The goal is to reduce the area under investigation as to decrease the time it takes to locate and repair pipeline leakages. This ultimately amounts to increased water efficiency and sustainability to keep water waste as low as possible.

1.1 Chapter overview

The report is split into several chapters. Here follows a brief introduction to each chapter which highlights the main subjects in each chapter.

Chapter 2: Problem analysis investigates some attributes regarding how consumers use fresh water. The chapter looks into the main problem in WDN's where leakages are one of the main problems. This conclusion leads to an investigation of current ways of detecting. From this, a problem formulation is stated. Lastly, in **Sec. 2.5** the limitations set and assumptions made in this project are listed.

Chapter 3: System model uses graph theory to describe a model for a WDN. In this chapter, both a dynamic and a reduced (steady-state) model is derived for the WDN. The dynamic model is for control design and the reduced model is for leakage detection. The reduced model derived is novel since it extends a single inlet WDN model to a multi inlet WDN model in a new way. The new multi inlet WDN model sets special requirements for the control of the WDN.

Chapter 4: Laboratory Setup introduces the setup used for testing the derived methods in real life. The Laboratory setup is named Smart Water Infrastructure Laboratory (SWIL) and can modular built different water network scenarios. It is found that the SWIL has significant communication delays. These must be taken into account for the control design. Lastly, the chapter contains descriptions of the modular built scenario, the parameterization of the scenario and the parameters needed for emulating a WDN.

Chapter 5: Control design introduces a MIMO output state feedback controller that can uphold the special requirements set by the system model and can handle the communication delay in the SWIL. States' feedback gains are derived on a reduced order version of the dynamic model with a Padé approximation appended. These gains are converted to output feedback gains by a method not seen by the authors before. This method used some very special properties of state-space matrices that arise after the model reduction. Lastly, the controller is thoroughly tested both in simulation and the SWIL.

Chapter 6: Leakage detection first describes how residuals are generated from the difference between measured pressure and the steady-state model estimated pressure. These residuals are somewhat noisy and almost impossible to detect leakage on directly. However, the residuals do have a change in mean when a leak is present. It is described how Gaussian Likelihood Ratio (GLR) is used to detect this change in mean and thereby detect the leak. Lastly, the approach suggested is tested in both simulations and the SWIL.

Chapter 7: Leakage localization starts by describing how the parameters of this multiple inlet steady-state model are expected to change in case of a leakage at a node. This includes a novel derivation since it is applicable for multiple inlet WDN´s, as it previously only had been derived for the single inlet case. The derived change in parameters is used to generate an expected residual for each possible leak. The direction of each expected residual is then compared to the direction of the real residual. The expected residual, which is the closest to having the direction of the real residual, is the one most likely to describe what node is leaky. Lastly, the approach suggested is tested in both simulation and the SWIL.

Chapter 8: Assessment introduces two further tests. First, it is tested how well the leakage detection and the leakage localization work when coupled together where they previously are tested individually. Secondly, the leakage detection is tested on a large scale simulation model of a WDN. Lastly, the findings of this report are evaluated in a discussion and a conclusion.

Problem analysis 2

This chapter investigates how consumers use water, what challenges/problems there are in distributing the water to the consumer, and some methods for dealing with these. Lastly from all this information, a problem statement and a set of limitations is made.

2.1 Consumer behaviour

In Denmark, an average inhabitant utilizes 101 litres of water per day [Danva, 2020]. The usage of water per person per day has been steadily declining for many years. The usage of water is not split evenly over the day. This could make it difficult to predict or simulate. Fortunately, a typical household follows a pattern in water usage, using a lot of water in the morning and evening and very little during the night. Then looking at an entire residential area this pattern will almost always be quite close to being correct. Companies have also been lowering their drinking water usage for many years [Danva, 2020]. A group of companies doing similar things (like businesses located in offices or restaurants) also have similar water usage patterns. These patterns in water usage are called diurnal curves. Some examples of diurnal curves are sketched in **Fig. 2.1** [Methods et al., 2003]. In this plot demand multiplier is a number which multiplies the average hourly demand to fit the current usage at a time. Diurnal curves are useful for analysing a Water Distribution Network (WDN) since they predict and simulate the water usage of an area.

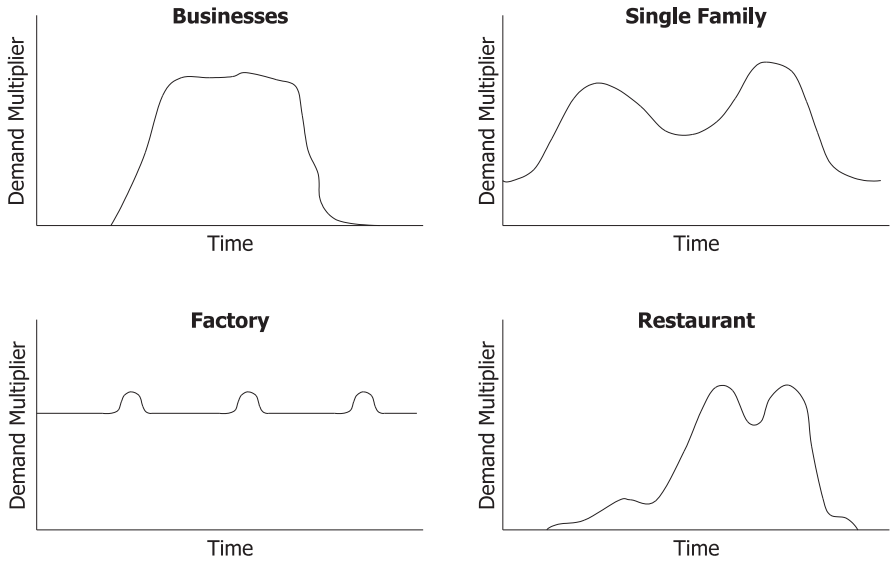


Figure 2.1: Diurnal curve for different user categories [Methods et al., 2003]

2.2 Water supply system

Consumers demand a stable and steady supply of water. To accommodate this large water supply systems (WSSs) are needed. They are usually split into two major parts namely the trunk-main network and the water distribution networks [Jensen and Kallesøe, 2016]. The trunk-main network is typically in charge of transporting large amounts of water from treatment facilities to supply a number of water distribution networks. The WDN then has the purpose of taking care of supplying the end consumer. The WDN can, depending on the city, be rather large. An example of a WDN is shown in **Fig. 2.2**. Therefore such networks are further divided into sub-networks, which are called District Metered Areas (DMAs). A DMA is also called a Pressure Management Area (PMA) if it is pressure controlled. In these networks the objective is to control the pressure according to a reference, such that the consumers are ensured an appropriate level of comfort [Jensen and Kallesøe, 2016]. The construction of WDNs consists of interconnected components

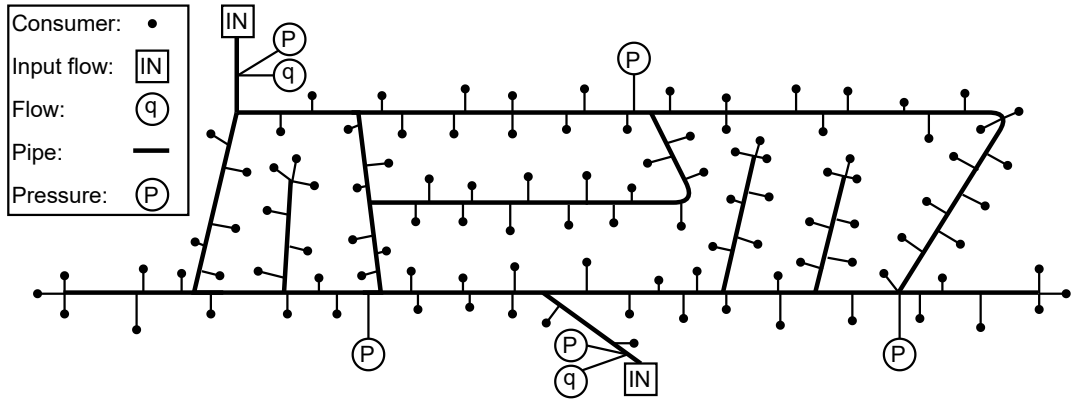


Figure 2.2: An example of how a water distribution network can look. Here there are multiple inlets, consumers and pressure sensors. This figure is a principle drawing inspired by figures in [Christodoulou et al., 2017] and [Jensen et al., 2018].

such as pipes, valves, pumps, tanks and reservoirs. In **Fig. 2.2** only pipes and sensors (flow and pressure) are shown. The pressure sensors not located at the inlets are more typical for modern water distribution setups. Generally, the trend is going towards having more sensors throughout the WDN. The other elements may be implicitly there eg. there could be a pump at the input flow or a valve at each consumer.

As described in the introduction, one of the main problems in WDNs is leakages and the detection of these. In Denmark 7.29% of water was lost to leakages in 2019 [Danva, 2020]. The reduced water usage in Denmark also means that it gets more difficult to keep the percentage lost to leakages down. Since 2015 the percentage lost has been varying between 7.22% and 8.05% instead of going down as previously [Danva, 2020]. This can indicate that evermore sophisticated ways of detecting and localizing water leakages fast are needed to secure that the percentage of water lost keeps on being steady or decreasing. A lot of other countries has, as previously mentioned, much higher loss percentages. Here, good and cheap techniques for detecting and localizing water leakages would be smart so that it is financially beneficial for the water company to remove leaks compared to just pumping extra water up.

2.3 Leakage detection and localization

Today, the techniques used to detect and localize leakage are split into two categories: leakage detection and leakage localization. The first category concerns the ability to detect when changes in a water distribution network (WDN) occurs. The second category concerns with localization of a leakage. As of today, several commercially available methods for leakage detection is by measuring the water flow from the distributor during night time, as the consumers at this time of the day have minimal water consumption, and thus the flow is more predictable. A leakage will then become apparent since the water consumption will increase. Combining this with a flow sensor for the individual consumer, where data is transmitted to the water distributor, can then, in the same manner, indicate potential leakages by looking at the flow at the inlet of the consumer versus the flow at the outlet of the distributor [Naturstyrelsen, 2015]. Furthermore, once a leakage has been detected, the task of localizing leakages can be realized with e.g. acoustic noise loggers with mobile usage or permanent placement. However several problems arise by using such equipment. There exist different types of acoustic noise loggers depending on the material of the pipe under investigation [Naturstyrelsen, 2015], which has the potential to become expensive. Moreover, the amount of time and area that is to be covered in order to locate a leakage can be too extensive [Jensen and Kallesøe, 2016] and may result in a considerable loss of drinkable water.

Another approach which is becoming readily available as the pricing of IoT sensors are decreasing is a combination between a model of the WDN and sensor measurements. In year 2018 the average cost of IoT sensors have decreased by nearly 200% since 2004 [Microsoft Dynamics 365, 2019]. The lower cost allows for using higher amounts of e.g. pressure and flow sensors in a WDN with lower expense [Danva, 2020]. In turn, this results in large amounts of measurement data, which can be used to compare against a model of the WDN. This is used to indicate when the WDN deviates from the nominal performance defined by the model [Naturstyrelsen, 2015]. Additionally, by exploiting a leakage localization scheme, by using the model, a leakage can be narrowed down to a specific area in which the acoustic noise loggers can be used to locate the source and thereby reducing the search area [Jensen and Kallesøe, 2016]. When a leak appears it

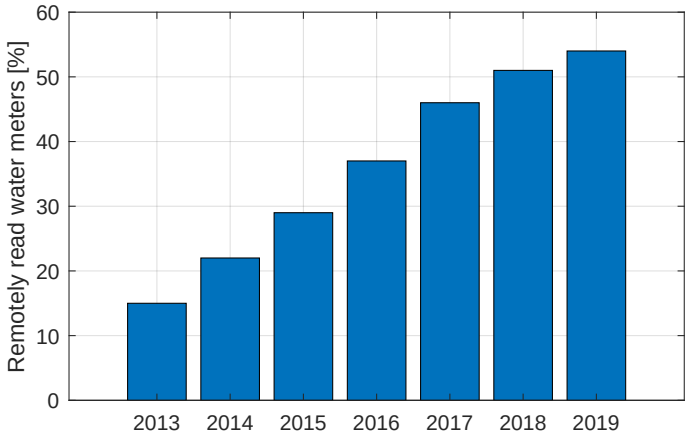


Figure 2.3: Share of water meters in Denmark which is remotely read making them useful for leakage detection [Danva, 2020]

must change some sensor outputs (often flow and pressure sensors) to be detectable. If this is the case, it should be possible to automatically detect it (leakage detection). In some cases, it is even possible to give an estimate of where the leak is (leakage localization). Since pressure measurements are more widely available and spread out through the system it is easier to use these when doing leakage localization. Flow measurements are mainly available at inlets (pumping stations) and outlets (consumers). The reason for this is the high price of large flow sensors for main water pipelines. Pressure sensors are fairly cheap compared to this.

When a leak appears it has been observed from [Jensen and Kallesøe, 2016] that the resulting change in measured pressure is relatively small and not sufficient for direct threshold detection. An idea for leakage detection is to generate a residual. A residual is a signal which changes magnitude considerably in case of a fault (in this case a leak). The residual proposed in [Jensen and Kallesøe, 2016] is defined as the predicted pressure ($\hat{p}(t)$) obtained by a model subtracted from the observed pressured ($p(t)$), that is: $r(t) = p(t) - \hat{p}(t)$. Similar ideas of using residuals for leakage detection and localization is proposed in [Perez et al., 2014]. This work is further expanded upon in [Rathore et al., 2021] and implemented in simulation for a single inlet WDN. The model used for residual generation needs to be very precise due to the small change of pressure at the leakage point. Often, a data-driven model is needed to get a sufficiently precise model [Jensen and Kallesøe, 2016].

2.4 Problem statement

The loss of drinkable water is a globally reoccurring problem. The magnitude of the problem is dependent on the country. Currently, the cost of equipment, the amount of time and size of the area being investigated to search and repair pipeline leakages in a WDN is extensive. This report aims to generalize the residual based leakage localization method from [Rathore et al., 2021] to work with multiple inlet WDNs and combining this with a detection scheme based on the same residuals. This leads to the following problem statement:

"How can a leakage detection and localization method be developed for a multiple inlet water distribution network which utilizes knowledge about the system and the actuators and sensors available?"

2.5 Project limitations and assumptions

As the size of water distribution networks covers whole cities in general, the number of consumers are determined by the number of inhabitants, businesses, factories and restaurants. Depending on the city under investigation the number of consumers and the size of the network can be cumbersome. In this project, physical testing will be limited to a simple WDN constructed using the AAU Smart Water Infrastructure Lab (SWIL) for practical reasons. In simulations, both a model of the simple WDN and a model of the WDN in a section of Randers is used.

As described in **Sec. 2.1** consumer behaviour varies widely based on the type of consumer. In this report, it is assumed that all customers have the diurnal curve of the type single-family. For a District Metering Area (DMA), this is a fair assumption since they will mostly hold users of the same type due to the way DMA's are intentionally built this way. Further, it is assumed that each consumer emulated uses a fraction of the total water demand. This fraction is for simplification kept fixed where in real life it would slowly change over time.

For simplification, it is chosen in the project to disregard Water towers and only look at WDNs without these. This is also a fair simplification since there exist DMAs and even entire WDNs without water towers. Throughout the project, multiple assumptions will be used regarding the modelling of WDNs. Assumptions only used in a single chapter are listed in that chapter. Below assumptions used throughout the report is listed.

Assumption 2.1 *The water contained in a WDN is incompressible*

Assumption 2.2 *The temperature of the water is* 10 °C. *This amounts to a constant density* $\rho = 998\ \frac{\mathrm{m}^3}{\mathrm{kg}}$ *and kinematic viscosity* $\vartheta = 1.3 \cdot 10^{-6}\ \frac{\mathrm{m}^2}{\mathrm{s}}$

Assumption 2.3 *Pipes in a WDN are completely filled with water*

Assumption 2.4 *Leakages in the WDN only happen at non-inlet vertices and only one leakage is active at a time*

2.6 Solution concept and overview

From the problem statement presented in **Sec. 2.4** a leakage detection and localization scheme is proposed which consist of three main parts: Residual generation, leakage detection and leakage localization, connected as seen in the block diagram in **Fig. 2.4**.

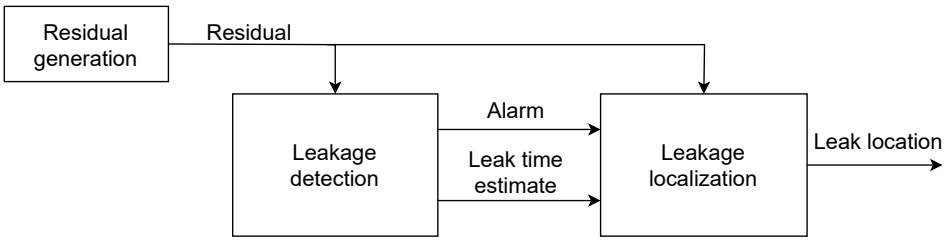


Figure 2.4: Proposed block diagram for leakage detection and localization

A residual is generated in the leftmost block and fed into both the detection and the localization block. If a leak is present this will cause an offset in the residual which will, if the leak is big enough, cause the detection block to activate the alarm signal which will trigger the leakage localization block. Because of the fact that sensor data is used in the generation of the residual, some level of noise is expected to be present on the signal. The effect of this noise can be attenuated by obtaining multiple samples of the residual and calculating the mean value. Since the results of the leak localization is based on this mean value it is important that it is calculated from samples of the residual from when a leak is present.

A way of doing this is to obtain these samples after the alarm signal has occurred but this will an introduce a delay before the leak can be localized. If, however, a delay is already present from the actual leak time to the time the alarm signal is triggered, previous samples of the residual from this period can be used since it will contain information about the leak. This requires an estimate of the leak time to be made from the detection block, which is then used to determine how many samples back in time can be used.

In the next chapter graph models of a multi-inlet water distribution network is presented. Presented is both a dynamic model and a steady-state model. The first is to be used for control design and stability analysis and the second is to be used for the residual generation

System model 3

In this chapter, the mathematical model of a water distribution network (WDN) is derived and described using graph theory and simple models for the components of a WDN. In **Sec. 3.1** the basics of graph theory are introduced and in **Sec. 3.2** related to a WDN. Here, the definitions for WDN graphs are made. In **Sec. 3.3** the models for the components constituting a WDN are derived. These are used in **Sec. 3.4** for setting up a dynamic model for a WDN using graph theory. In **Sec. 3.5** a reduced-order model is derived using the previous definitions.

3.1 Fundamental graph theory

The mathematical model of a WDN is based on the fundamental graph theory, consisting of basic definitions of a graph, trees, fundamental circuits, and matrix representations. The following graph theory is based on [Deo, 1974].

A graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ consists of a set of vertices $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ and a set of edges $\mathcal{E} = \{e_1, e_2, ..., e_m\}$, where n is the number of vertices and m is the number of edges in a graph, respectively. Each edge e_i is associated with a pair of vertices $\{v_i, v_j\}$, where $i, j \in \{1, 2, ..., n\}$. An example of a digraph, also known as a directed graph, is shown in **Fig. 3.1**. Here, the difference from an ordinary graph being the direction assigned to each egde. Since this project only will consider digraphs, such a graph will be denoted as just a graph. The graph shown in **Fig. 3.1** consists of $n = 4$ vertices and $m = 5$ edges.

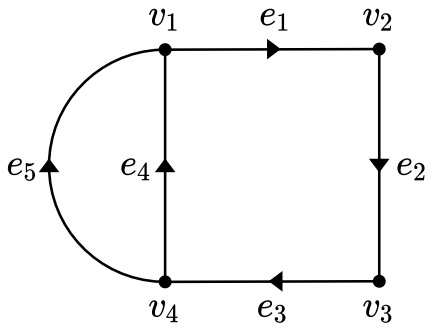


Figure 3.1: Graph example

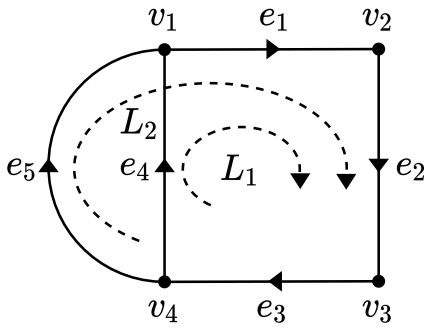


Figure 3.2: Graph example with loops

Furthermore, in graph theory the notion of a walk is defined as a sequence of vertices and edges. A walk begins and ends at vertices and the edges in such a walk may only appear once. Vertices may however appear more once. By this, a walk is either open or closed. Here the set of vertices $\mathcal{V}$ and edges $\mathcal{E}$ constituting a walk in $\mathcal{G}$ is defined as a sub-graph of

$\mathcal{G}$. An open walk begins at some vertex and ends in a terminal vertex but the same vertex does not appear twice. This is called an elementary path. By **Fig. 3.1** such a path could be $\{v_1, e_1, v_2, e_2, v_3, e_3, v_4\}$. In a closed walk the same vertex is allowed to appear twice, where the walk begins and ends up in the same vertex. A closed walk is also referred to as a loop or circuit.

In graph theory the notion of a tree $\mathcal{T}$ is defined as a connected graph without any loops. A graph is connected if there is at least one path between every pair of vertices in $\mathcal{G}$. Furthermore, it follows that in a tree there is only a single path between every pair of vertices. A tree $\mathcal{T}$ can also be a spanning tree if the graph $\mathcal{G}$ is connected and $\mathcal{T}$ is a subgraph of $\mathcal{G}$ and $\mathcal{T}$ contains all vertices of $\mathcal{G}$. In **Fig. 3.1** the spanning tree could for instance be defined as $\mathcal{T} = \{\mathcal{V}, \mathcal{E}_\mathcal{T}\} = \{v_1, v_2, v_3, v_4, e_1, e_2, e_3\}$. The remaining edges, which is not a part of the spanning tree is called chords and are denoted as $\mathcal{C}$. For this specific spanning tree the chords are $\mathcal{E}_\mathcal{C} = \{e_4, e_5\}$. A chord is defined as if adding one edge to a tree, exactly one loop is formed. A loop formed by adding an edge is called a fundamental loop. The number of fundamental loops in a graph is equal to the number of chords. This is depicted in **Fig. 3.2**, where two loops are shown.

3.1.1 Matrix representations of a graph

A convenient way of representing a graph is through matrix representations. For instance, the incidence matrix, H, is constructed based on a directed graph as seen in **Eq. 3.1** with the dimensions $n \times m$ where n is the number of vertices and m is the number of edges.

$$h_{ij} = \begin{cases} -1, & \text{if the } j^{th} \text{ edge is entering the } i^{th} \text{ vertex} \\ 0, & \text{if the } j^{th} \text{ edge is not connected to the } i^{th} \text{ vertex} \\ 1, & \text{if the } j^{th} \text{ edge is leaving the } i^{th} \text{ vertex} \end{cases} \tag{3.1}$$

When constructing H it can be advantageous to sort it by means of the chords and tree as follows: $H = [H_\mathcal{C} \quad H_\mathcal{T}]$. The sorted incidence matrix based on the graph seen in **Fig. 3.1** is given by **Eq. 3.2**. Furthermore, the reduced incidence matrix $\bar{H} = [\bar{H}_\mathcal{C} \quad \bar{H}_\mathcal{T}]$ is obtained by removing an arbitrary row in H. The removed row is then referred to as the reference vertex.

$$H = \begin{bmatrix} H_\mathcal{C} & H_\mathcal{T} \end{bmatrix} = \begin{array}{c} \begin{matrix} e_4 & e_5 & e_1 & e_2 & e_3 \end{matrix} \\ \left[\begin{array}{cc|ccc} -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 1 & 1 & 0 & 0 & -1 \end{array}\right] \end{array} \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} \tag{3.2}$$

Moreover, the fundamental loop matrix is constructed as seen in **Eq. 3.3**. Here, the matrix dimension are $l \times m$, where $l = m - n + 1$ is the number of fundamental loops and also correspond to the number of chords. The loop matrix for the particular graph in **Fig.**

3.1 is seen in **Eq. 3.4**.

$$b_{ij} = \begin{cases} -1, & \text{if the } j^{th} \text{ edge is in the } i^{th} \text{ loop and it's direction} \\ & \text{does not follow the loop direction} \\ 0, & \text{if the } j^{th} \text{ edge is not in the } i^{th} \text{ loop} \\ 1, & \text{if the } j^{th} \text{ edge is in the } i^{th} \text{ loop and it's direction} \\ & \text{follow the loop direction} \end{cases} \tag{3.3}$$

$$B = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.4}$$

B can also be constructed mathematically from the expression shown in **Eq. 3.5** [Deo, 1974].

$$B = \begin{bmatrix} I_l & -\bar{H}_{\mathcal{C}}^T \bar{H}_{\mathcal{T}}^{-T} \end{bmatrix} \tag{3.5}$$

3.2 Graph theory for WDNs

In **Sec. 3.1**, graph theory is introduced. In this section, extra definitions are shown to make graph theory a useful modeling tool for WDNs. Specifically it introduces definitions for open hydraulic networks.

When modeling a water distribution network using graph theory each edge represents either a pipe, a valve, or a pump. The nodes represent the connection points between these components. Further, a node can have an external flow, flowing either into it or away from it, coming from a source not seen as a part of the network. These flows are named demands and are given as $d_i(t)$ for the demand at the i^{th} node. An inflow could come from a reservoir. An outflow could come from a consumer using water. Demands are represented as arrows going into a node (**Fig. 3.3**). If the demand at node i is an inflow $d_i(t)$ is positive, that is for an inflow $d_i(t) \geq 0$.

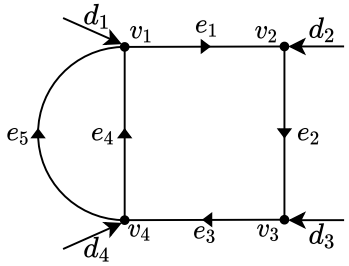


Figure 3.3: A graph with external demands

Using these definitions for flows in the network, Kirchhoff's current law (KCL) can be used to set up the relation between the demands and the edge flows. KCL states that the sum of all electrical currents entering or leaving a node is equal to zero which is equivalent to mass conservation for hydraulic networks. By applying this to vertex v_1 in the example shown

in **Fig. 3.3** the following flow relation is obtained: $q_1 - q_5 - q_4 - d_1 = 0 \Rightarrow q_1 - q_5 - q_4 = d_1$ where q_i is the flow in e_i. Using matrix notation all the vertex demands can be represented in a single equation by making use of the incidence matrix, H, as seen in **Eq. 3.6** below.

$$Hq(t) = \begin{bmatrix} H_{\mathcal{C}} & H_{\mathcal{T}} \end{bmatrix} \begin{bmatrix} q_{\mathcal{C}}(t) \\ q_{\mathcal{T}}(t) \end{bmatrix} = d(t) \tag{3.6}$$

where $q(t) \in \mathbb{R}^m$ is a vector containing the flows for all the edges and $d(t) \in \mathbb{R}^n$ is a vector containing all the vertex demands.

Because of mass conservation in the network the amount water flowing out of the network must be equal to the amount of water flowing in, which leads to the number of independent demands being equal to $n-1$ as shown in **Eq. 3.7** below:

$$\sum_{i}^{n} d_i(t) = 0 \Rightarrow d_n(t) = -\sum_{i}^{n-1} d_i(t) \tag{3.7}$$

By making use of a proof presented by [Kallesøe et al., 2015] this equation can also be represented with matrix notation yielding **Eq. 3.8**

$$d(t) = \begin{bmatrix} I_{n-1} \\ -\mathbb{1}^T \end{bmatrix} \bar{d}(t) = H_{\mathcal{T}} \bar{H}_{\mathcal{T}}^{-1} \bar{d}(t) \tag{3.8}$$

where $\bar{d}(t) \in \mathbb{R}^{n-1}$ is the demands at the non-reference vertices.

Besides the demands each node also has a pressure associated with it given as $p_i(t)$ for the absolute pressure at the i^{th} node. The vector $p(t) \in \mathbb{R}^n$ is defined to contain all the node pressures and is partitioned into $p(t) = [\bar{p}(t)^T \quad p_n(t)]^T$ where $p_n(t)$ is the pressure at the reference node and $\bar{p}(t)$ is the pressure at the remaining nodes. The pressure difference across each edge is obtained as

$$\Delta p(t) = \begin{bmatrix} \Delta p_{\mathcal{C}}(t) \\ \Delta p_{\mathcal{T}}(t) \end{bmatrix} = H^T p(t) = \begin{bmatrix} H_{\mathcal{T}}^T \\ H_{\mathcal{C}}^T \end{bmatrix} p(t) \tag{3.9}$$

where $\Delta p(t) \in \mathbb{R}^m$ is a vector containing the pressure differences across each edge with $\Delta p_i(t)$ being the pressure difference across the i^{th} edge. Using the tree part ($\mathcal{T}$) of the incidence matrix H and the pressure difference vector $\Delta p(t)$ and removing the reference node (giving $\bar{H}_{\mathcal{T}}$ and $\bar{p}(t)$) one can get an expression for the relative pressure in the nodes $\bar{p}(t)$ from the pressure difference in the tree edges $\Delta p_{\mathcal{T}}(t)$ as shown in **Eq. 3.10**.

$$\begin{aligned} \bar{H}_{\mathcal{T}}^T \bar{p}(t) &= \Delta p_{\mathcal{T}}(t) \\ \bar{p}(t) &= \bar{H}_{\mathcal{T}}^{-T} \Delta p_{\mathcal{T}}(t) \end{aligned} \tag{3.10}$$

As previously stated each edge correspond to a component (either a valve, a pump or a pipe). The next section aims to model these components in order to obtain a function which maps the flow $q_i(t)$ and for a pump and a valve also a control signal ($\omega_i(t)$ for a pump and $OD_i(t)$ for a valve) into the pressure difference across the edge $\Delta p_i(t)$.

3.3 Component models

In this section, the components which correspond to the edges of the graph are modelled. As stated earlier, an edge either corresponds to a pump, a valve, or a pipe. For the pipes, both a dynamic model and a steady-state model are presented. For the pumps and the valves, steady-state models are presented. This is followed by a description of what dynamics and delay one might see from the pump and the valve on a physical plant. The dynamic model is used for stability analysis of the system and the steady-state model is used for leakage detection and localization.

3.3.1 Pipe model

This section aims to develop the model for the pipes used in the WDN. Note that the vector notation introduced in **Sec. 3.2** is discarded while deriving the pipe model since only a single pipe is considered, e.g. $q(t)$ is a scalar value. The vector notation is reintroduced in the final equation for the pipe model with the subscript i for the i^{th} component.

The model is developed by applying Newton's second law, which states that the rate of change of the linear momentum of a system is equal to the net force acting on that system, to the water inside the pipe. This relation is shown in **Eq. 3.11**.

$$\frac{d\mathcal{P}(t)}{dt} = \frac{d(Mv(t))}{dt} = F_{net}(t) \tag{3.11}$$

where $\mathcal{P}(t)$ is the linear momentum of the water inside the pipe, M is the mass of the water, $v(t)$ is its velocity and $F_{net}(t)$ is the net force acting on the water. By assuming that the pipe is always filled with water and using the fact that water is incompressible the mass becomes a constant and can be moved outside the differential. The net force is written as a sum of the different forces acting on the water namely the input-, output- and the resistance force. An illustration of a pipe with the forces are shown in **Fig. 3.4** and the resulting equation in **Eq. 3.11**.

$$M\frac{dv(t)}{dt} = F_{in}(t) - F_{out}(t) - F_{res}(t) \tag{3.12}$$

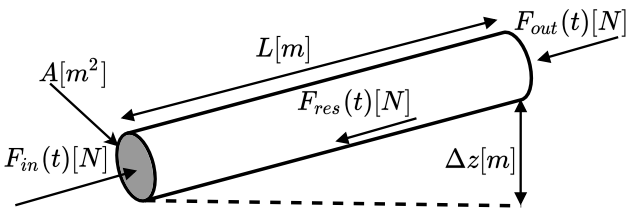


Figure 3.4: Single pipe

By inserting expressions for the mass, input force and output force, **Eq. 3.12** is expanded

and simplified yielding **Eq. 3.13**:

$$\begin{aligned} AL\rho\frac{dv(t)}{dt} &= (Ap_{in}(t) + A\rho g z_{in}) - (Ap_{out}(t) + A\rho g z_{out}) - F_{res}(t) \\ AL\rho\frac{d\left(\frac{q(t)}{A}\right)}{dt} &= Ap_{in}(t) - Ap_{out}(t) + A\rho g z_{in} - A\rho g z_{out} - F_{res}(t) \\ L\rho\frac{d\left(\frac{q(t)}{A}\right)}{dt} &= p_{in}(t) - p_{out}(t) + \rho g z_{in} - \rho g z_{out} - \frac{F_{res}(t)}{A} \\ \frac{L\rho}{A}\frac{dq(t)}{dt} &= \Delta p(t) + \rho g \Delta z - \frac{F_{res}(t)}{A} = \Delta p(t) + \rho g \Delta z - \lambda(q(t)) \end{aligned} \tag{3.13}$$

In **Eq. 3.13** A is the cross sectional area of the pipe, L is pipe length, ρ is the water density, $q(t)$ is the water flow, g is the gravitational acceleration, $p_{in}(t)$ and $p_{out}(t)$ is the pressure at the in- and outlet of the pipe respectively and z_{in} and z_{out} is the elevation of the in- and outlet of the pipe respectively. The pressure and elevation differences, $\Delta p(t)$ and Δz, is defined as $(p_{in}(t) - p_{out}(t))$ and $(z_{in} - z_{out})$ respectively. For ease of notation the function $\lambda(q(t))$ is introduced and represents the pressure drop caused by the resistance force, $F_{res}(t)/A$. This function will be explained in the following subsection.

Pipe resistance

The pressure drop caused by the pipe friction, $\lambda(q(t))$, is divided into two terms, one for the surface resistance, h_f, and one for the form resistance, h_m. Both h_f and h_m is given as a head loss which is related to pressure as $p = h\rho g$ as is done in **Eq. 3.14**.

$$\lambda(q(t)) = \frac{F_{res}(t)}{A} = h_f \rho g + h_m \rho g \tag{3.14}$$

The form resistance h_m, also commonly referred to as minor losses, represents the resistance caused by pipe fittings, bends, curves etc. in the pipe. As can be seen in **Eq. 3.15a** [Swamee and Sharma, 2008] the form resistance is dependent on the form-loss coefficient, k_f. This coefficient varies according to the type of resistance that is present along the pipe e.g. angle of the bend, type of fitting, etc. In the case where multiple form resistances is present along the same pipe, the total resistance can be expressed as a sum as seen in **Eq. 3.15b**.

$$h_m = k_f \frac{8q(t)^2}{\pi^2 g D^4} \tag{3.15a}$$

$$h_m = \sum_j K_{f,j} \frac{8q(t)^2}{\pi^2 g D^4} \tag{3.15b}$$

The surface resistance, h_f, aims to model the resistance caused by the roughness of the inside of the pipe. It is chosen to model this by the Darcy–Weisbach equation [Swamee and Sharma, 2008] which is shown in **Eq. 3.16**.

$$h_f = f \frac{8Lq(t)^2}{\pi^2 g D^5} \tag{3.16}$$

where the function, f, is the friction factor given by **Eq. 3.17**. The function changes depending on the type of flow present in the pipe which is determined by Reynolds

number, Re. For laminar flows ($Re \leq 2000$) the friction factor is calculated using the Hagen–Poiseuille equation [Swamee and Sharma, 2008] while the expression used for turbulent flows ($Re \geq 4000$) is the one developed by Colebrook, 1939. For flows in the transitional range ($2000 < Re < 4000$) it is chosen to calculate the friction factor with the same expression used by the water network simulation tool EPANET [Rossman et al., 2020]. Since the expression for the friction factor at transitional flows is based on a cubic interpolation formula, it is rather extensive and is therefore not written out in this section for the sake of readability. It can instead be seen in **Eq. A.2** in **App. A.1**.

$$f = \begin{cases} \dfrac{64}{Re} & , Re \leq 2000 \\ \textbf{Eq. A.2} & , 2000 < Re < 4000 \\ 1.325 \left[ln \left(\dfrac{\varepsilon}{3.7D} + \dfrac{5.74}{Re^{0.9}} \right) \right]^{-2} & , Re \geq 4000 \end{cases} \tag{3.17}$$

The Reynolds number, Re, is given in **Eq. 3.18** below

$$Re = \frac{4q(t)}{\pi \vartheta D} \tag{3.18}$$

In **Eq. 3.18** ϑ is the kinematic viscosity of the water. Inserting **Eq. 3.15b** and **3.16** into **Eq. 3.14** gives the combined expression for $\lambda(q(t))$ seen in **Eq. 3.19**. To preserve the sign/direction of the flow, $q(t)^2$ is written as $|q(t)|q(t)$ thus allowing for flows in both directions.

$$\lambda(q(t)) = \left(f \frac{8L\rho}{\pi^2 D^5} + \sum_j K_{f,j} \frac{8\rho}{\pi^2 D^4} \right) |q(t)|q(t) \tag{3.19}$$

By substituting $\lambda(q)$ into **Eq. 3.13** the pipe model in **Eq. 3.20** is obtained. Furthermore, the index i is added as a subscript to indicate the i^{th} pipe in the network.

$$\frac{L_i \rho}{A_i} \frac{dq_i(t)}{dt} = \Delta p_i(t) + \rho g \Delta z_i - \left(f_i \frac{8L_i\rho}{\pi^2 D_i^5} + \sum_j K_{fi,j} \frac{8\rho}{\pi^2 D_i^4} \right) |q_i(t)|q_i(t) \tag{3.20}$$

The model can be simplified by using the fact that the flows in water networks is mostly turbulent with high Reynolds numbers [Kallesøe et al., 2015]. In this case the friction factor, f, changes very little with changing flow and is therefore assumed to be independent of $q(t)$. This means that the whole pipe resistance term can be written as in **Eq. 3.21**.

$$\lambda(q(t)) = \kappa |q(t)|q(t), \tag{3.21}$$

Here κ is a constant only depending on the pipe parameters. This simplifies the dynamic pipe model to the one in **Eq. 3.22**. This model is used to develop the dynamic model in **Sec. 3.4**.

$$\frac{L_i \rho}{A_i} \frac{dq_i(t)}{dt} = \Delta p_i(t) + \rho g \Delta z_i - \kappa |q_i(t)|q_i(t) \tag{3.22}$$

As stated in the beginning of this section, leakage detection mostly takes place when the water network is in steady state and the fast pipe dynamics can therefore be neglected by setting $\frac{dq(t)}{dt} = 0$. With these simplifications the pipe model can further be reduced to **Eq. 3.23** which will be used in **Sec. 3.5** to develop a reduced model.

$$\Delta p_i(t) = \kappa_i |q_i(t)|q_i(t) - \rho g \Delta z_i \tag{3.23}$$

3.3.2 Valve model

To simulate the end-users in the distribution network the valve component is introduced. This component can vary the flow through it by adjusting its opening degree thus creating a variable pressure drop. Valves are usually physically small components so the pressure drop caused by the elevation difference is therefore assumed to be very small and thus neglected. Furthermore, because of the small size, the internal volume of the valve will be very small compared to the pipes in the network, which means that the dynamics caused by the volume flow through it will be very small. This effect is therefore neglected as well. With these assumptions, the valve is modeled by the algebraic equation shown in **Eq. 3.24**. This model is based on the work done by [Rathore, 2020].

$$\Delta p_i(t) = \mu_i(q_i(t), OD_i(t)) = \frac{1}{K_{v,i}(OD_i(t))^2}|q_i(t)|q_i(t) \tag{3.24}$$

where $\mu_i(q_i(t), OD_i(t))$ is the function describing the pressure drop over the i^{th} valve, $OD_i(t) \in [0, 1]$ is the opening degree of the valve ranging from 0 to 100% and $K_{v,i}(OD_i(t))$ is the valve specific conductive function. By assuming a valve with a linear conductive function $K_{v,i}(OD_i(t))$ is then given as $K_{v,i}(OD(t)) = K_{vs,i} \cdot OD_i(t)$, where $K_{vs,i}$ is defined as the volume flow though the valve at maximum opening degree ($OD_i(t) = 1$) with a pressure difference over the valve at one bar. Substituting this into **Eq. 3.24** yields **Eq. 3.25** which is the model for the pressure drop over the i^{th} valve used in this project.

$$\Delta p_i = \mu_i(q_i(t), OD_i(t)) = \frac{1}{(K_{vs,i} \cdot OD_i(t))^2}|q_i(t)|q_i(t) \tag{3.25}$$

3.3.3 Pump model

The pump pressure difference model presented in this section is derived in [Kallesøe, 2005]. In **Fig. 3.5** a simplified cross-section of a centrifugal pump is shown with the parameters used in the model displayed.

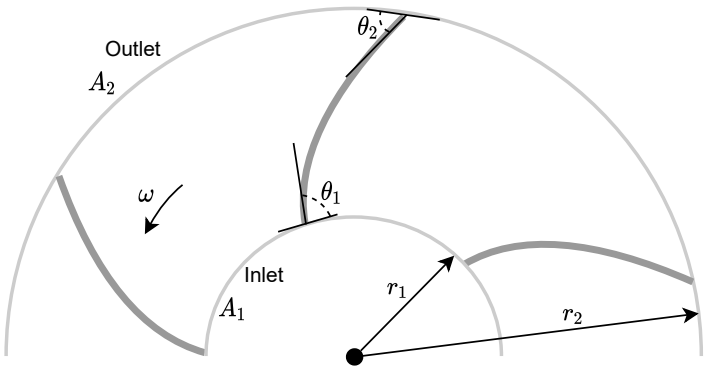


Figure 3.5: Impeller drawing with parameters. Inspired by figures in [Grundfos, 2006] and [Kallesøe, 2005]

In **Eq. 3.26** the relation between pressure difference $\Delta p_i(t)$, flow through the pump $q_i(t)$

and angular velocity of the pump $\omega_i(t)$ is shown.

$$\Delta p_i(t) = -\zeta_i(q_i(t), \omega_i(t)) = -\rho \cdot g \cdot \left(a_{h0,i} \cdot \omega_i(t)^2 + a_{h1,i} \cdot \omega_i(t) \cdot q_i(t) - a_{h2,i} \cdot q_i(t)^2\right) \tag{3.26}$$

In **Eq. 3.26** ρ is the water density and g is the gravitational constant. The coefficients in this model, $a_{h0,i}$, $a_{h1,i}$ and $a_{h2,i}$, are presented in **Eq. 3.27**.

$$a_{h0,i}(t) = \sigma_{s,i}(t) \left(\frac{r_{2,i}^2}{g} - \frac{r_{1,i}^2}{g}\right) - K_{s,i} \cdot K_{d,i}^2 \tag{3.27a}$$

$$a_{h1,i}(t) = 2 \cdot K_{s,i} \cdot K_{d,i} - \sigma_{s,i}(t) \left(\frac{r_{2,i}}{gA_{2,i}} \cot(\theta_{2,i}) - \frac{r_{1,i}}{gA_{1,i}} \cot(\theta_{1,i})\right) \tag{3.27b}$$

$$a_{h2,i} = K_{s,i} + K_{f,i} \tag{3.27c}$$

In **Eq. 3.27a** $K_{s,i}$ is a constant which maps the difference between the actual flow $q_i(t)$ and the design flow $q_{d,i}(t)$ into shock losses in head: $h_{s,i}(t) = K_{s,i} \cdot (q_i(t) - q_{d,i}(t))^2$. The design flow is described by the constant $K_{d,i}$, $q_{d,i}(t) = K_{d,i}\omega_i(t)$. $\sigma_{s,i}(t)$ is a slip factor which in general is non-linearly dependent on $q_i(t)$ and $\omega_i(t)$. $r_{1,i}$ and $r_{2,i}$ in **Eq. 3.27a** correspond to the radii shown in **Fig. 3.5**. In **Eq. 3.27b** $A_{1,i}$ is the inlet area of the impeller (ring area) and $A_{2,i}$ is the outlet area of the impeller. $\theta_{1,i}$ is the inlet blade angle and $\theta_{2,i}$ is the outlet blade angle as shown in **Fig. 3.5**. In **Eq. 3.27c** $K_{f,i}$ is a constant mapping flow into friction losses in head: $h_{f,i}(t) = K_{f,i}q_i(t)^2$.

The parameters $a_{h0,i}$, $a_{h1,i}$ and $a_{h2,i}$ can be identified from the datasheet. This is described in **App. A.3**.

3.3.4 Actuator dynamics and control delay

When using actuators such as pumps and valves in a real-world application there might be several factors that can make distinctions from theoretical models. Firstly, the dynamics might have an influence in terms of the motor used to adjust the OD of the valve. That is, the change in OD is not instantaneous. For a pump, the same reasoning holds for the motor driving the impeller. Secondly, delays resulting from communication links between actuators and a microprocessor could prove difficulties when designing controllers as shown in [Rathore, 2020]. As seen in **Fig. 3.6** the block diagram illustrates how a reference signal is affected before an actuator begins to act and following how the sensors also has some delay and dynamics.

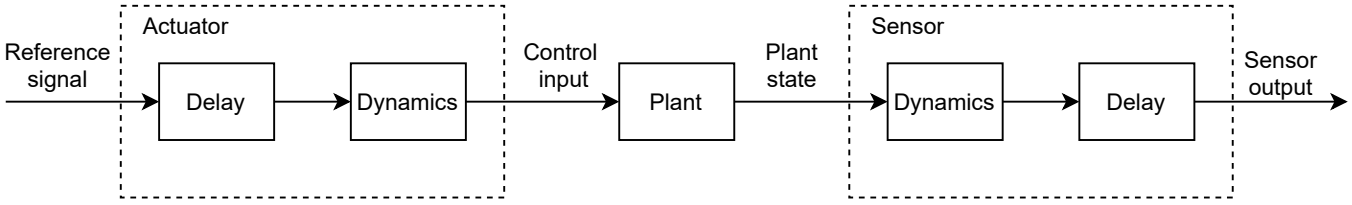


Figure 3.6: Block diagram showing the path from reference signal to the system input through the communication delays and actuator dynamics

3.4 Dynamic Model

In **Sec. 3.3** the different components constituting a WDN is described with respect to the governing equations. Extending upon this, a complete dynamical system model of the WDN, which includes the dynamics of the individual component, can be derived. Such a model is not only suited for simulation purposes but also for control design. The model allows for simulating the dynamics regarding the flows $q(t)$ through each edge and the pressure differences $\Delta p(t)$ across each edge. Taking a starting point by combining the pipe model **Eq. 3.20**, the pump model **Eq. 3.26**, and the valve model **Eq. 3.25** the general component model describing the pressure differences across each edge is constructed as seen in **Eq. 3.28a**. This is also referred to as the stacked model as $\Delta p(t)$ is a vector of all pressure differences across all edges in the WDN [Kallesøe, 2020].

$$\Delta p(t) = J\dot{q}(t) + \lambda(q(t)) - \rho g \Delta z + \mu(q(t), OD(t)) - \zeta(q(t), \omega(t)) \tag{3.28a}$$

$$\Delta p(t) = J\dot{q}(t) + \Gamma(t) - \rho g \Delta z - \zeta(t) \tag{3.28b}$$

where $\Gamma(t) = \lambda(q(t)) + \mu(q(t), OD(t))$ and the dependency upon $q(t)$ and $OD(t)$ is implicit. In the same manner, $\zeta(t) = \zeta(q(t), \omega(t))$ depends implicitly on $q(t)$ and $\omega(t)$. Then, by multiplying the loop matrix B from the left and using the loop law (KVL) on the stacked model in **Eq. 3.28b**, that is $B\Delta p = 0$ and $B\Delta z = 0$.

$$B\Delta p(t) = BJ\dot{q}(t) + B\Gamma(t) - \rho g B \Delta z - B\zeta(t) \tag{3.29a}$$

$$0 = BJ\dot{q}(t) + B\Gamma(t) - B\zeta(t) \tag{3.29b}$$

The flows $q(t)$ through each edge in the WDN is given **Eq. 3.30** [Kallesøe et al., 2015]

$$q(t) = B^T q_{\mathcal{C}}(t) + B_d^T \bar{d}(t) = B^T q_{\mathcal{C}}(t) + B_d^T \bar{F} \bar{d}_f(t) \tag{3.30}$$

where $q_{\mathcal{C}}(t)$ is the chord flows, $B_d = [0 \quad \bar{H}_{\mathcal{T}}^{-T}]$, $\bar{d}(t)$ is the demands at non-reference vertices, $\bar{d}_f(t)$ is the demands at open vertices (open to atmospheric pressure) except the reference vertex, and $\bar{F} \in \mathbb{R}^{n-1 \times o-1}$ is a matrix that maps $\bar{d}_f(t)$ to the remaining demands $\bar{d}(t)$. Here, n is the number of vertices and o is the number of demands open to atmospheric pressure. Furthermore, **Eq. 3.30** can be partitioned into a chord and tree part as seen in **Eq. 3.31**.

$$q(t) = \begin{bmatrix} q_{\mathcal{C}}(t) \\ q_{\mathcal{T}}(t) \end{bmatrix} = \begin{bmatrix} I \\ -\bar{H}_{\mathcal{T}}^{-1} \bar{H}_{\mathcal{C}} \end{bmatrix} q_{\mathcal{C}}(t) + \begin{bmatrix} 0 \\ \bar{H}_{\mathcal{T}}^{-1} \end{bmatrix} \bar{F} \bar{d}_f(t) \tag{3.31}$$

Inserting **Eq. 3.30** into **Eq. 3.29b** results in **Eq. 3.32a**. Then, rearranging with respect to $\dot{q}_{\mathcal{C}}(t)$ and $\dot{\bar{d}}_f(t)$ gives **Eq. 3.32b**. This equation is the first equation, which is used to construct the standard form.

$$0 = BJ(B^T \dot{q}_{\mathcal{C}}(t) + B_d^T \bar{F} \dot{\bar{d}}_f(t)) + B\Gamma(t) - B\zeta(t) \tag{3.32a}$$

$$BJB^T \dot{q}_{\mathcal{C}}(t) + BJB_d^T \bar{F} \dot{\bar{d}}_f(t) = -B\Gamma(t) + B\zeta(t) \tag{3.32b}$$

To derive an expression for the second equation used to construct the standard form, the goal is now to find some relation of $\dot{\bar{d}}_f$. Looking at the part of **Eq. 3.28a** only constituting the spanning tree, that is

$$\Delta p_{\mathcal{T}}(t) = J_{\mathcal{T}} \dot{q}_{\mathcal{T}}(t) + \Gamma_{\mathcal{T}}(t) - \rho g \Delta z_{\mathcal{T}} - \zeta_{\mathcal{T}}(t) \tag{3.33}$$

Then, taking **Eq. 3.33** and using the relations $\Delta p_{\mathcal{T}}(t) = H_{\mathcal{T}}^T p(t)$ and $\Delta z_{\mathcal{T}} = H_{\mathcal{T}}^T z$ to arrive at **Eq. 3.34**. Then, multiplying this equation from the left by $\bar{H}_{\mathcal{T}}^{-T}$ and using the relation $\bar{H}_{\mathcal{T}}^{-T} H_{\mathcal{T}}^T = [I \quad -\mathbb{1}]$.

$$H_{\mathcal{T}}^T p(t) = J_{\mathcal{T}} \dot{q}_{\mathcal{T}}(t) + \Gamma_{\mathcal{T}}(t) - \rho g H_{\mathcal{T}}^T z - \zeta_{\mathcal{T}}(t) \tag{3.34}$$

$$\bar{H}_{\mathcal{T}}^{-T} H_{\mathcal{T}}^T p(t) = \bar{H}_{\mathcal{T}}^{-T} J_{\mathcal{T}} \dot{q}_{\mathcal{T}}(t) + \bar{H}_{\mathcal{T}}^{-T} \Gamma_{\mathcal{T}}(t) - \rho g \bar{H}_{\mathcal{T}}^{-T} H_{\mathcal{T}}^T z - \bar{H}_{\mathcal{T}}^{-T} \zeta_{\mathcal{T}}(t) \tag{3.35}$$

where $p(t) = [\bar{p}^T(t) \quad p_n]^T$ and $\bar{p}(t) = [p_1(t) \quad p_2(t) \quad ... \quad p_{n-1}(t)]^T$, and n is the number of vertices as defined in **Sec. 3.2**. To arrive at **Eq. 3.36a** the following is used: $[I \quad -\mathbb{1}]p(t) = \bar{p}(t) - \mathbb{1}p_n$ and $[I \quad -\mathbb{1}]z = \bar{z} - \mathbb{1}z_n$. Furthermore, some indices of $\bar{p}(t)$ contains vertices that are open in the sense of atmospheric pressure. The collection of open vertices are denoted $\bar{p}_f(t)$. Furthermore, the reference node p_n is chosen to be an open node. In turn, this makes the relative pressure between the remaining open vertices and $p_n(t)$ equal to zero. This property is used in **Eq. 3.36b** together with the relation $\bar{p}_f(t) = \bar{F}^T \bar{p}(t)$

$$\bar{p}(t) - \mathbb{1}p_n = \bar{H}_{\mathcal{T}}^{-T} J_{\mathcal{T}} \dot{q}_{\mathcal{T}}(t) + \bar{H}_{\mathcal{T}}^{-T} \Gamma_{\mathcal{T}}(t) - \rho g(\bar{z} - \mathbb{1}z_n) - \bar{H}_{\mathcal{T}}^{-T} \zeta_{\mathcal{T}}(t) \tag{3.36a}$$

$$\begin{aligned} 0 = \bar{F}^T \bar{p}(t) - \bar{F}^T \mathbb{1}p_n = \bar{F}^T \bar{H}_{\mathcal{T}}^{-T} J_{\mathcal{T}} \dot{q}_{\mathcal{T}}(t) + \bar{F}^T \bar{H}_{\mathcal{T}}^{-T} \Gamma_{\mathcal{T}}(t) \\ - \rho g \bar{F}^T (\bar{z} - \mathbb{1}z_n) - \bar{F}^T \bar{H}_{\mathcal{T}}^{-T} \zeta_{\mathcal{T}}(t) \end{aligned} \tag{3.36b}$$

Then, substituting the tree part $q_{\mathcal{T}}(t)$ of **Eq. 3.31** into **Eq. 3.36b**, expanding the expression and rearranging by isolating terms with $\dot{q}_{\mathcal{C}}(t)$ and $\dot{\bar{d}}_f(t)$ to the left side of the equality

$$\begin{aligned} &- \bar{F}^T \bar{H}_{\mathcal{T}}^{-T} J_{\mathcal{T}} \bar{H}_{\mathcal{T}}^{-1} \bar{H}_{\mathcal{C}} \dot{q}_{\mathcal{C}}(t) + \bar{F}^T \bar{H}_{\mathcal{T}}^{-T} J_{\mathcal{T}} \bar{H}_{\mathcal{T}}^{-1} \bar{F} \dot{\bar{d}}_f(t) \\ &\quad = -\bar{F}^T \bar{H}_{\mathcal{T}}^{-T} \Gamma_{\mathcal{T}}(t) + \rho g \bar{F}^T (\bar{z} - \mathbb{1}z_n) + \bar{F}^T \bar{H}_{\mathcal{T}}^{-T} \zeta_{\mathcal{T}}(t) \end{aligned} \tag{3.37}$$

The task is now to expand **Eq. 3.37** and use the definitions for the loop matrix B and B_d as to simplify the expression as seen in **Eq. 3.38a** and **Eq. 3.38b**.

$$\begin{aligned} &\bar{F}^T \begin{bmatrix} 0 & \bar{H}_{\mathcal{T}}^{-T} \end{bmatrix} \begin{bmatrix} J_{\mathcal{C}} & 0 \\ 0 & J_{\mathcal{T}} \end{bmatrix} \begin{bmatrix} I \\ -\bar{H}_{\mathcal{T}}^{-1} \bar{H}_{\mathcal{C}} \end{bmatrix} \dot{q}_{\mathcal{C}}(t) + \bar{F}^T \begin{bmatrix} 0 & \bar{H}_{\mathcal{T}}^{-T} \end{bmatrix} \begin{bmatrix} J_{\mathcal{C}} & 0 \\ 0 & J_{\mathcal{T}} \end{bmatrix} \begin{bmatrix} 0 \\ \bar{H}_{\mathcal{T}}^{-T} \end{bmatrix} \bar{F} \dot{\bar{d}}_f(t) \\ &\quad = -\bar{F}^T \bar{H}_{\mathcal{T}}^{-T} \Gamma_{\mathcal{T}}(t) + \rho g \bar{F}^T (\bar{z} - \mathbb{1}z_n) + \bar{F}^T \bar{H}_{\mathcal{T}}^{-T} \zeta_{\mathcal{T}}(t) \end{aligned} \tag{3.38a}$$

$$\begin{aligned} &\bar{F}^T B_d J B^T \dot{q}_{\mathcal{C}}(t) + \bar{F}^T B_d J B_d^T \bar{F} \dot{\bar{d}}_f(t) \\ &\quad = -\bar{F}^T \bar{H}_{\mathcal{T}}^{-T} \Gamma_{\mathcal{T}}(t) + \rho g \bar{F}^T (\bar{z} - \mathbb{1}z_n) + \bar{F}^T \bar{H}_{\mathcal{T}}^{-T} \zeta_{\mathcal{T}}(t) \end{aligned} \tag{3.38b}$$

Using **Eq. 3.32b** and **Eq. 3.38b** together constitutes the non-linear system seen in **Eq. 3.39** [Kallesøe, 2020].

$$J_{ex} \begin{bmatrix} \dot{q}_{\mathcal{C}}(t) \\ \dot{\bar{d}}_f(t) \end{bmatrix} = - \begin{bmatrix} I & -\bar{H}_{\mathcal{C}}^T \bar{H}_{\mathcal{T}}^{-T} \\ 0 & \bar{F}^T \bar{H}_{\mathcal{T}}^{-T} \end{bmatrix} \left(\begin{bmatrix} \Gamma_{\mathcal{C}}(t) \\ \Gamma_{\mathcal{T}}(t) \end{bmatrix} - \begin{bmatrix} \zeta_{\mathcal{C}}(t) \\ \zeta_{\mathcal{T}}(t) \end{bmatrix} \right) + \rho g \begin{bmatrix} 0 \\ \bar{F}^T \end{bmatrix} (\bar{z} - \mathbb{1}z_n) \tag{3.39}$$

where J_{ex} is given in **Eq. 3.40**

$$J_{ex} = \begin{bmatrix} BJB^T & BJB_d^T \bar{F} \\ \bar{F}^T B_d J B^T & \bar{F}^T B_d J B_d^T \bar{F} \end{bmatrix} \tag{3.40}$$

Assuming that the inverse of J_{ex} exist **Eq. 3.39** can be rewriting to the standard non-linear state space form:

$$\begin{bmatrix} \dot{q}_{\mathcal{C}}(t) \\ \dot{\bar{d}}_f(t) \end{bmatrix} = \dot{x} = f(x,u) \tag{3.41}$$

where x is the state vector and $u = [\omega^T \quad OD^T]^T$ is the control inputs to the pumps. However, OD, and thereby the outlet demand, is in this project seen as a disturbance and not a control variable.

The network investigated in this project will be setup such that all the nodes open to atmospheric pressure are connected to the rest of the network through either a pump or a valve representing either a pumping station or a consumer, respectively. It is chosen to implement a control strategy that keeps one of the nodes connected to a pump at a constant pressure while the other pumps are flow controlled. The outputs of the system are therefore chosen to be $y = [d_{f,p} \quad p_p]^T$ where $d_{f,p}$ is a vector of all the flows though the flow controlled pumps and p_p is the pressure on the network side of the pressure controlled pump. While the flows $d_{f,p}$ are all part of the state vector a separate equation is needed to calculate the pressure p_p. To acquire the equation for the pressure at non reference nodes, $\bar{p}$, is used since it will contain p_p. The expression for $\bar{p}$ is presented in **Eq. 3.42** below which is only dependent on the states and inputs. Because of this the output function can be represented as seen in **Eq. 3.43**

$$\bar{p}(t) = \bar{H}_{\mathcal{T}}^{-T} J_{\mathcal{T}} \dot{q}_{\mathcal{T}}(t) + \bar{H}_{\mathcal{T}}^{-T} \Gamma_{\mathcal{T}}(t) - \rho g(\bar{z} - \mathbb{1} z_n) - \bar{H}_{\mathcal{T}}^{-T} \zeta_{\mathcal{T}}(t) + \mathbb{1} p_n \tag{3.42}$$

$$y = \begin{bmatrix} d_{f,p}(t) \\ p_p(t) \end{bmatrix} = h(x,u) \tag{3.43}$$

3.4.1 Introducing a leakage into the dynamic model

As the aim of this project is to detect and localize leakages, one must be able to introduce leakages in the dynamic model. This is however only for simulation purposes when simulating the dynamic model.

In this section, the dynamic model (standard form) is derived with leakages. This derivation utilizes the assumption shown in **Asm. 2.4**. The idea of introducing a leakage is based on the fact that if a leakage occurs, it is seen as a demand for the leaking vertex.

Taking a starting point in **Sec. 3.4**, the flows $q(t)$ through each edge in the WDN are given by **Eq. 3.30**. Now, the vector of demands $\bar{d}$ except the reference vertex are rewritten in terms of the leakage as follows: $\bar{d} = \bar{F}\bar{d}_f + \bar{d}_l$, where $\bar{d}_l = [d_{l,1} \quad d_{l,2} \quad ... \quad d_{l,n-1}]^T$ is the reduced leakage demand vector. Substituting the expression for $\bar{d}$ into **Eq. 3.30** yields **Eq. 3.44a**. Taking the derivative of **Eq. 3.44a** results in **Eq. 3.44b**

$$q(t) = B^T q_{\mathcal{C}}(t) + B_d^T \bar{F} \bar{d}_f(t) + B_d^T \bar{d}_l(t) \tag{3.44a}$$

$$\dot{q}(t) = B^T \dot{q}_{\mathcal{C}}(t) + B_d^T \bar{F} \dot{\bar{d}}_f(t) + B_d^T \dot{\bar{d}}_l(t) \tag{3.44b}$$

$$\dot{q}(t) = \begin{bmatrix} \dot{q}_{\mathcal{C}}(t) \\ \dot{q}_{\mathcal{T}}(t) \end{bmatrix} = \begin{bmatrix} I \\ -\bar{H}_{\mathcal{T}}^{-1} \bar{H}_{\mathcal{C}} \end{bmatrix} \dot{q}_{\mathcal{C}}(t) + \begin{bmatrix} 0 \\ \bar{H}_{\mathcal{T}}^{-1} \end{bmatrix} \bar{F} \dot{\bar{d}}_f(t) + \begin{bmatrix} 0 \\ \bar{H}_{\mathcal{T}}^{-1} \end{bmatrix} \dot{\bar{d}}_l(t) \tag{3.44c}$$

Then, substituting **Eq. 3.44b** into **Eq. 3.29b** gives **Eq. 3.45**.

$$0 = BJ\dot{q}(t) + B\Gamma(t) - B\zeta(t) \tag{3.45a}$$

$$0 = BJ(B^T\dot{q}_{\mathcal{C}}(t) + B_d^T\bar{F}\dot{\bar{d}}_f(t) + B_d^T\dot{\bar{d}}_l(t)) + B\Gamma(t) - B\zeta(t) \tag{3.45b}$$

$$BJB^T\dot{q}_{\mathcal{C}}(t) + BJB_d^T\bar{F}\dot{\bar{d}}_f(t) = -B\Gamma(t) + B\zeta(t) - BJB_d^T\dot{\bar{d}}_l(t) \tag{3.45c}$$

Here, **Eq. 3.45c** is the first differential equation, which constitutes the standard form. The second differential equation is derived similarly as in **Sec. 3.4**. Restating **Eq 3.36b** in **Eq 3.46a** and substituting the tree part $\dot{q}_{\mathcal{C}}(t)$ from **Eq. 3.44c** into the equation.

$$0 = \bar{F}^T\bar{H}_{\mathcal{T}}^{-T}J_{\mathcal{T}}\dot{q}_{\mathcal{T}}(t) + \bar{F}^T\bar{H}_{\mathcal{T}}^{-T}\Gamma_{\mathcal{T}}(t) - \rho g\bar{F}^T(\bar{z} - \mathbb{1}z_n) - \bar{F}^T\bar{H}_{\mathcal{T}}^{-T}\zeta_{\mathcal{T}}(t) \tag{3.46a}$$

$$\begin{aligned} 0 = \bar{F}^T\bar{H}_{\mathcal{T}}^{-T}J_{\mathcal{T}}(&-\bar{H}_{\mathcal{T}}^{-1}\bar{H}_{\mathcal{C}}\dot{q}_{\mathcal{C}}(t) + \bar{H}_{\mathcal{T}}^{-1}\bar{F}\dot{\bar{d}}_f(t) + \bar{H}_{\mathcal{T}}^{-1}\dot{\bar{d}}_l(t)) \\ &+ \bar{F}^T\bar{H}_{\mathcal{T}}^{-T}\Gamma_{\mathcal{T}}(t) - \rho g\bar{F}^T(\bar{z} - \mathbb{1}z_n) - \bar{F}^T\bar{H}_{\mathcal{T}}^{-T}\zeta_{\mathcal{T}}(t) \end{aligned} \tag{3.46b}$$

Next, expanding the terms in parenthesises, isolating terms including $\dot{q}_{\mathcal{C}}(t)$ and $\dot{\bar{d}}_f(t)$ at left hand side, and identify the relations for the loop matrix B and B_d, the expression simplifies as seen in **Eq. 3.47a**. The procedure follows exactly as in **Sec. 3.4** from **Eq. 3.37** to **Eq. 3.38b**.

$$\begin{aligned} &\bar{F}^T B_d JB^T\dot{q}_{\mathcal{C}}(t) + \bar{F}^T B_d JB_d^T\bar{F}\dot{\bar{d}}_f(t) = \\ &- \bar{F}^T\bar{H}_{\mathcal{T}}^{-T}\Gamma_{\mathcal{T}}(t) + \rho g\bar{F}^T(\bar{z} - \mathbb{1}z_n) + \bar{F}^T\bar{H}_{\mathcal{T}}^{-T}\zeta_{\mathcal{T}}(t) - \bar{F}^T B_d JB_d^T\dot{\bar{d}}_l(t) \end{aligned} \tag{3.47a}$$

Then, combining **Eq. 3.45c** and **Eq. 3.47a** results in the non-linear dynamic model seen in

$$\begin{aligned} J_{ex}\begin{bmatrix}\dot{q}_{\mathcal{C}}(t)\\ \dot{\bar{d}}_f(t)\end{bmatrix} = -\begin{bmatrix} I & -\bar{H}_{\mathcal{C}}^T\bar{H}_{\mathcal{T}}^{-T}\\ 0 & \bar{F}^T\bar{H}_{\mathcal{T}}^{-T}\end{bmatrix}\left(\begin{bmatrix}\Gamma_{\mathcal{C}}(t)\\ \Gamma_{\mathcal{T}}(t)\end{bmatrix} - \begin{bmatrix}\zeta_{\mathcal{C}}(t)\\ \zeta_{\mathcal{T}}(t)\end{bmatrix}\right) + \rho g\begin{bmatrix}0\\ \bar{F}^T\end{bmatrix}(\bar{z} - \mathbb{1}z_n) \\ - \begin{bmatrix}BJB_d^T\\ \bar{F}^T B_d JB_d^T\end{bmatrix}\dot{\bar{d}}_l(t) \end{aligned} \tag{3.48}$$

where J_{ex} is given in **Eq. 3.49**

$$J_{ex} = \begin{bmatrix} BJB^T & BJB_d^T\bar{F}\\ \bar{F}^T B_d JB^T & \bar{F}^T B_d JB_d^T\bar{F}\end{bmatrix} \tag{3.49}$$

From the standard form in **Eq. 3.49**, it is evident that the introduction of a leakage into the model adds a new term, the derivative of the leakage, namely $\dot{\bar{d}}_l(t)$. In simulation it is not necessary to include $\bar{d}_l(t)$ as a state in the model. Instead, $\dot{\bar{d}}_l(t)$ is seen as a control input to the model. This requires $\dot{\bar{d}}_l(t)$ to be known. In simulation $\dot{\bar{d}}_l(t)$ is constructed based on the leakage $\bar{d}_l(t)$. In **Fig. 3.7** and **Fig. 3.8** a leakage with a size of -0.1 m^3/h is shown. The leak is however rate limited in contrast to an ideal step, which would be infinitely fast. $\dot{\bar{d}}_l(t)$ is then made such that it matches the slope of $\bar{d}_l(t)$.

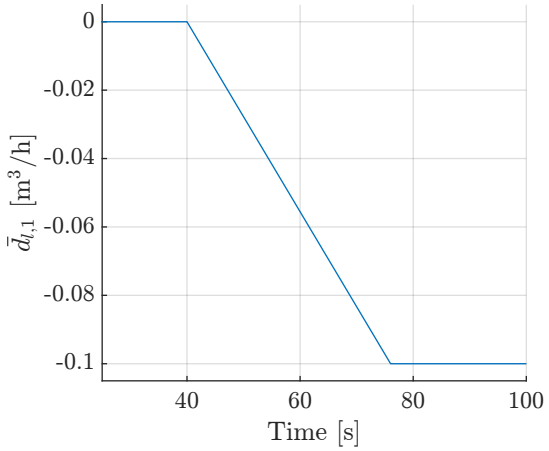


Figure 3.7: Leakage step with the slope $\dot{\bar{d}}_l$

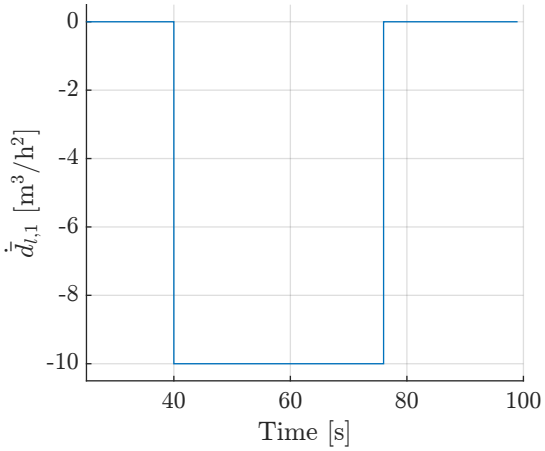


Figure 3.8: The rate at which the leakage changes

Furthermore, the pressure at each vertex in the WDN except the reference vertex is given as in **Eq. 3.42** but with $\dot{q}_{\mathcal{T}}(t)$ given as the tree part of **Eq. 3.44c**, which then considers the leakage.

3.5 Reduced-order model

The idea behind the reduced-order model shown in this section is that there can, using some assumptions, be found a set of constants, α_i and β_i, which maps the reference node pressure ($p_n(t)$) and the total consumption of water ($\sigma(t) \geq 0$) into the i^{th} node pressure $p_i(t)$, where i can be any integer from 1 to $n-1$. The reference node, the n^{th} node, must be chosen so that it is an inflow to the WDN (a pump). The reduced-order model is reduced in the sense that it sees the pumps and valves as demands. It further is a steady-state model which neglects the dynamics of the pipes. The two main assumptions used for the reduced model is displayed in **Asm. 3.1** and **Asm. 3.2**.

Assumption 3.1 *It is assumed that the distribution between the $n-1$ free nodal demands is fixed, that is there exists a vector $\mathrm{v} \in \mathbb{R}^{n-1}$ such that:*

$$\bar{d}(t) = -\mathrm{v} \cdot \sigma(t) \tag{3.50}$$

Assumption 3.2 *The hydraulic resistance in the pipes is assumed to take the simple form presented in **Eq. 3.21** or repeated here in **Eq. 3.51**.*

$$\lambda_i(q_i(t)) = \kappa_i |q_i(t)| q_i(t) \tag{3.51}$$

Asm. 3.1 is valid as long as the WDN consists of users of the same type [Jensen and Kallesøe, 2016]. This is often the case in a District Metered Areas (DMAs). The benefit of having users of the same type is large.These users will have the same surge in water usage during half time of a soccer game. Since only distribution between users is assumed and not what specific curve each user has, this should only lead to the expected pressure

drops as every user will have this surge. Therefore, having a model which relies on the same user type and not on predefined diurnal curves should be beneficial. **Asm. 3.2** is clearly just a simplification of the models presented in **Sec. 3.3.1**

The relation between the pressure of the i^{th} node, the consumption of water, and the reference node pressure is shown in **Eq. 3.52** and was proposed in [Kallesøe et al., 2015] and expanded on in [Jensen and Kallesøe, 2016]. However, this model has only been proven to be valid for single inlet systems where the total consumption of water is equal to the demand from the reference node: $\sigma(t) = d_n(t)$.

$$p_i(t) = -\alpha_i \cdot \sigma(t)^2 - \beta_i + p_n(t) \tag{3.52}$$

In the next section properties which are only known to be valid for the single inlet model is summarized. This is done to make it apparent what assumptions need to be proven for the multiple inlet model as well.

3.5.1 Single inlet model

The single inlet model presented in [Kallesøe et al., 2015] and [Jensen and Kallesøe, 2016] relies on the graph theory presented in **Sec. 3.1** and **Eq. 3.2**. Further, it uses the two assumptions **Asm. 3.1** and **Asm. 3.1**. For the single inlet model **Eq. 3.50** from **Asm. 3.1** is equal to **Eq. 3.53** since the total consumption of water, $\sigma(t)$, is equal to the demand supplied at the inlet node, $d_n(t)$, due to mass conservation.

$$\bar{d}(t) = -\mathrm{v} \cdot d_n(t) \tag{3.53}$$

For single inlet systems v has the property shown in **Eq. 3.54**.

$$\sum_{i=1}^{n-1} \mathrm{v}_i = 1 \tag{3.54}$$

Using these properties it is proven in [Kallesøe et al., 2015] that a vector $a \in \mathbb{R}^m$, which is partitioned into a chord and a tree part $a = [a_{\mathcal{C}}^T \quad a_{\mathcal{T}}^T]^T$, can be found mapping the inflow, $d_n(t)$, to the flows in the edges ($q(t) \in \mathbb{R}^m$). This relation is shown in **Eq. 3.55**.

$$q(t) = \begin{bmatrix} q_{\mathcal{C}}(t) \\ q_{\mathcal{T}}(t) \end{bmatrix} = a d_n(t) \tag{3.55}$$

Using the graph theory defined in **Sec. 3.2** a flow can be mapped into a pressure. Further, using that: $d_n(t) \cdot |d_n(t)| = d_n(t)^2$ since $d_n(t) \geq 0$ and gathering constant terms, eg. height differences between nodes in a constant term β_i, one arrives at the expression for the pressure in the i^{th} node shown in **Eq. 3.56** or equivalently in vector from in **Eq. 3.57**. α_i and β_i are parameters which is to be identified from time series data using linear regression [Jensen and Kallesøe, 2016]. It is important to note that in these equations p_n is not an open node as in the previous section. Instead it is the reference pressure node inside the network. The reduced model does not have any open nodes.

$$p_i(t) = -\alpha_i \cdot d_n(t)^2 - \beta_i + p_n(t) \tag{3.56}$$

$$\bar{p}(t) = -\alpha d_n(t)^2 - \beta + \mathbb{1} p_n(t) \tag{3.57}$$

Furthermore, it can be derived what α and β represents. First in **Eq. 3.58** a steady state version of the pressure expression from **Eq. 3.42** is shown. Further this expression excludes the terms for valve and pumps since these are seen directly as demands in the reduced model. Substituting $q_{\mathcal{T}}(t) = a_{\mathcal{T}} d_n(t)$ into **Eq. 3.58** and using that $\lambda_{\mathcal{T}}(q_{\mathcal{T}}(t))$ is a homogeneous function of degree two, the expression is given as seen in **Eq. 3.59**.

$$\bar{p}(t) = \bar{H}_{\mathcal{T}}^{-T} \lambda_{\mathcal{T}}(q_{\mathcal{T}}(t)) - \rho g(\bar{z} - \mathbb{1} z_n) + \mathbb{1} p_n(t) \tag{3.58}$$

$$\bar{p}(t) = \bar{H}_{\mathcal{T}}^{-T} \lambda_{\mathcal{T}}(a_{\mathcal{T}}) d_n^2 - \rho g(\bar{z} - \mathbb{1} z_n) + \mathbb{1} p_n(t) \tag{3.59}$$

From **Eq. 3.59** it is clearly seen that α and β is equal to the expressions in **Eq. 3.60**.

$$\alpha = -\bar{H}_{\mathcal{T}}^{-T} \lambda_{\mathcal{T}}(a_{\mathcal{T}}) \qquad \beta = \rho g(\bar{z} - \mathbb{1} z_n) \tag{3.60}$$

3.5.2 Multiple inlet model

The reduced-order model considered is based on the work presented in [Kallesøe et al., 2015]. Since this model is developed for WDNs with a single inlet, this subsection aims to extend the model to WDNs with multiple inlets. The model developed in [Kallesøe et al., 2015] works with a single pumping inlet, which is controlled according to a pressure reference. For a WDN with multiple inlets, the chosen approach is to follow the proposed method with only one inlet being pressure controlled and the rest are to be flow controlled to make the remaining pumps act as a consumer but with the opposite sign of demand. Here, it is assumed that the flow-controlled pumps can deliver a specific demand and that the pressure-controlled pump can accommodate a pressure reference. Furthermore, as illustrated in **Sec. 3.2** the positive direction of demands is defined as going into the WDN. In turn, this means that consumer demands are negative (going out of the system) and the inlet demands are positive (going into the system).

As described in **Sec. 3.2**, Kirchhoffs node law must be uphold. Moreover, due to mass conservation **Eq. 3.61** is valid

$$\sum_{i=1}^{n-c} d_i(t) + \sum_{i=n-c+1}^{n} d_i(t) = 0 \tag{3.61}$$

where n is the number of vertices, c is the number of inlets, and $d(t) \in \mathbb{R}^n$ is a vector of demands partitioned as $d(t) = [\acute{d}(t)^T \quad \grave{d}(t)^T]^T$ with $\acute{d}(t) \in \mathbb{R}^{n-c}$ and $\grave{d}(t) \in \mathbb{R}^c$. $\acute{d}(t)$ is defined to contain the flows going out of the system and the nodes where there is zero demand, that is $\acute{d}(t) \leq 0$. $\grave{d}(t)$ is defined to contain the inflows to the system, that is $\grave{d}(t) \geq 0$. Here the pressure-controlled pump is denoted $d_n(t)$ and is chosen to be the reference node.

The reduced order model in [Kallesøe et al., 2015] is derived based on two assumptions. Firstly, it is assumed that **Eq. 3.50** is true but the second assumption given by **Eq. 3.54** does not hold in the multiple inlet case. In the multiple inlet case $\mathrm{v} \in \mathbb{R}^{n-1}$ is a demand distribution vector partitioned as $\mathrm{v} = [\acute{\mathrm{v}}^T \quad \grave{\mathrm{v}}^T]^T$, where $\acute{\mathrm{v}} \in \mathbb{R}^{n-c}$ and $\grave{\mathrm{v}} \in \mathbb{R}^{c-1}$. v is in charge of distributing the total consumer demand to the individual demands $\bar{d}(t)$, except for the demand of the pressure controlled pump $d_n(t)$. For the multiple inlet case the property of v seen in **Eq. 3.54** does not hold. Therefore, the next step is to derive

an expression for v such that **Eq. 3.50** is upheld in the case of multiple inlets with the control configuration as described above. More specifically, it should allow for v having negative components and thus allowing flow-controlled inlets acting as consumers but with the opposite flow direction.

For ease of derivation the summation in **Eq. 3.54** is split in two parts - consumers and inlets, respectively. This is seen by **Eq. 3.62** and **Eq. 3.63**.

$$\sum_{i=1}^{n-c} d_i(t) = -\sigma(t) \tag{3.62}$$

$$\sum_{i=n-c+1}^{n} d_i(t) = \sigma(t) \tag{3.63}$$

Using **Eq. 3.62** and substituting the part of $\bar{d}(t)$ only concerning non inlets from **Eq. 3.50** results in **Eq. 3.64**. Next, pulling out $\sigma(t)$ and dividing on both sides of the equation results **Eq. 3.65**. Notice that this property of v is similar to **Eq. 3.54**.

$$\sum_{i=1}^{n-c} -\mathrm{v}_i\sigma(t) = -\sigma(t) \tag{3.64}$$

$$(-\sigma(t))\sum_{i=1}^{n-c} \mathrm{v}_i = -\sigma(t) \Rightarrow \sum_{i=1}^{n-c} \mathrm{v}_i = 1 \tag{3.65}$$

Next, the task is to find an expression for the remaining part of $d(t)$ which is the demands of the inlets, namely $\grave{d}(t)$. Using **Eq. 3.63** and pulling out the last element of $d(t)$ ($d_n(t)$) and substituting **Eq. 3.50** is then expressed in **Eq. 3.66**.

$$\sum_{i=n-c+1}^{n} d_i(t) = d_n(t) + \sum_{i=n-c+1}^{n-1} -\mathrm{v}_i \cdot \sigma(t) = \sigma(t) \tag{3.66}$$

Pulling $\sigma(t)$ out of the summation in **Eq. 3.66** and rearranging by subtracting $d_n(t)$ and dividing by $\sigma(t)$ on both sides of the equation results in **Eq. 3.67**. This describes how the nodal demands for the inlets are distributed.

$$d_n(t) - \sigma(t)\sum_{i=n-c+1}^{n-1} \mathrm{v}_i = \sigma(t) \Rightarrow \sum_{i=n-c+1}^{n-1} \mathrm{v}_i = -1 + \frac{d_n(t)}{\sigma(t)} \tag{3.67}$$

To this end, the resulting property of v is shown in **Eq. 3.68** by using the derived expressions from **Eq. 3.65** and **Eq. 3.67**

$$\sum_{i=1}^{n-1} \mathrm{v}_i = \sum_{i=1}^{n-c} \mathrm{v}_i + \sum_{i=n-c+1}^{n-1} \mathrm{v}_i = \frac{d_n(t)}{\sigma(t)} \tag{3.68}$$

This describes the distribution of the $n-1$ free vertex demands, now allowing for a WDN with multiple inlets.

Furthermore, the component flows $q(t)$ for the entire WDN are given by **Eq. 3.69** and are dependent on the chord flows $q_{\mathcal{C}}(t) \in \mathbb{R}^l$. This relation is then a mapping of chord

flows $q_{\mathcal{C}}(t)$ and demands $\bar{d}(t)$ to the remaining flows in the WDN, namely $q_{\mathcal{T}}(t) \in \mathbb{R}^{n-1}$ [Kallesøe et al., 2015].

$$q(t) = \begin{bmatrix} q_{\mathcal{C}}(t) \\ q_{\mathcal{T}}(t) \end{bmatrix} = B^T q_{\mathcal{C}}(t) + \begin{bmatrix} 0_{l \times n-1} \\ \bar{H}_{\mathcal{T}}^{-1} \end{bmatrix} \bar{d}(t) \tag{3.69}$$

where $l = m - n + 1$. Substituting the expression of B given by **Eq. 3.5** into **Eq. 3.69** results in **Eq. 3.70**.

$$q(t) = \begin{bmatrix} I_l \\ -\bar{H}_{\mathcal{T}}^{-1}\bar{H}_{\mathcal{C}} \end{bmatrix} q_{\mathcal{C}}(t) + \begin{bmatrix} 0_{l \times n-1} \\ \bar{H}_{\mathcal{T}}^{-1} \end{bmatrix} \bar{d}(t) \tag{3.70}$$

In [Kallesøe et al., 2015] a proof has been made under **Asm. 3.1** and **Asm. 3.2** that allows for computing the flows $q(t)$ by means of a unique vector a and the total consumer consumption $\sigma(t)$, as seen in **Eq. 3.71**. This proof is however only shown for a single inlet WDN. Therefore, the following will prove that the vector a exists and is valid for a multiple inlet WDN using some assumptions.

$$q(t) = \begin{bmatrix} q_{\mathcal{C}}(t) \\ q_{\mathcal{T}}(t) \end{bmatrix} = a\sigma(t) = \begin{bmatrix} a_{\mathcal{C}} \\ a_{\mathcal{T}} \end{bmatrix} \sigma(t) \tag{3.71}$$

Taking a starting point using **Eq. 3.70** and substituting the relation of $q_{\mathcal{C}}$ given by **Eq. 3.71** to arrive at an expression for the tree flows $q_{\mathcal{T}}$ as seen in **Eq. 3.72**. Here the assumption in **Asm. 3.1**

$$q_{\mathcal{T}}(t) = -\bar{H}_{\mathcal{T}}^{-1}\bar{H}_{\mathcal{C}} a_{\mathcal{C}} \sigma(t) + \bar{H}_{\mathcal{T}}^{-1}\bar{d}(t) \tag{3.72a}$$

$$q_{\mathcal{T}}(t) = -\bar{H}_{\mathcal{T}}^{-1}\bar{H}_{\mathcal{C}} a_{\mathcal{C}} \sigma(t) - \bar{H}_{\mathcal{T}}^{-1}\mathrm{v}\sigma(t) \tag{3.72b}$$

In [Kallesøe et al., 2015] $d_n(t) = \sigma(t)$, which is not the case for multiple inlets. Instead, relations of $d_n(t)$ and $\sigma(t)$ are to be derived. Starting from **Eq. 3.63**, the expression in **Eq. 3.73b** is derived

$$\sigma(t) = \sum_{i=n-c+1}^{n} d_i(t) \tag{3.73a}$$

$$= d_n(t) + \sum_{i=n-c+1}^{n-1} d_i(t) \tag{3.73b}$$

Applying the relation of $d_i(t) = -\mathrm{v}_i \cdot \sigma(t)$ as seen in **Eq. 3.50** and rearranging results in **Eq. 3.74** and **Eq. 3.76**.

$$d_n(t) = \sigma(t) - \sum_{i=n-c+1}^{n-1} -\mathrm{v}_i \sigma(t) \tag{3.74a}$$

$$= \sigma(t) + \sigma(t) \sum_{i=n-c+1}^{n-1} \mathrm{v}_i \tag{3.74b}$$

$$= \left(1 + \sum_{i=n-c+1}^{n-1} \mathrm{v}_i\right) \sigma(t) \tag{3.74c}$$

$$\sigma(t) = \left(1 + \sum_{i=n-c+1}^{n-1} \mathrm{v}_i\right)^{-1} \cdot d_n(t) \tag{3.75}$$

$$\sigma(t) = \gamma \cdot d_n(t) \quad , \quad \gamma = \left(1 + \sum_{i=n-c+1}^{n-1} \mathrm{v}_i\right)^{-1} \tag{3.76}$$

Lastly, substituting **Eq. 3.76** into the expression of $q_{\mathcal{T}}$ from **Eq. 3.72** results in the final derivation in **Eq. 3.77b**.

$$q_{\mathcal{T}}(t) = -\bar{H}_{\mathcal{T}}^{-1}\bar{H}_{\mathcal{C}}a_{\mathcal{C}} \cdot \gamma \cdot d_n(t) - \bar{H}_{\mathcal{T}}^{-1}\mathrm{v} \cdot \gamma \cdot d_n(t) \tag{3.77a}$$

$$q_{\mathcal{T}}(t) = \gamma(-\bar{H}_{\mathcal{T}}^{-1}\bar{H}_{\mathcal{C}}a_{\mathcal{C}} \cdot d_n(t) - \bar{H}_{\mathcal{T}}^{-1}\mathrm{v}) \cdot d_n(t) \tag{3.77b}$$

With this, it can be seen that $q_{\mathcal{T}}(t)$ in **Eq. 3.77a** is a linear function of $d_n(t)$, which proves that the reduced-order model can be applied to a WDN with multiple inlets by using the derivations of the relation between $\sigma(t)$ and $d_n(t)$, and the property of v in **Eq. 3.68**.

As a further note it can clearly be seen that $a_{\mathcal{T}}$ is equal to the expression in **Eq. 3.78** from **Eq. 3.77b**

$$a_{\mathcal{T}} = -\bar{H}_{\mathcal{T}}^{-1}\bar{H}_{\mathcal{C}}a_{\mathcal{C}} - \bar{H}_{\mathcal{T}}^{-1}\mathrm{v} \tag{3.78}$$

3.5.3 Introduction of a leakage in the reduced-order model

As a part of the derivation of the leakage localization scheme presented in **Sec. 7.1** it is necessary to introduce the notion of a leakage in the WDN. This is done by distinguishing between the WDN while operating under nominal conditions (no leakage) and while a leakage is present, that is a deviation from the nominal operation. Firstly, the demand vector $d(t)$ is split into two parts, $\breve{d}(t)$, the demand when no leak is present, denoted as nominal demand, and $\delta d(t)$, the change from nominal demand due to a leak. This definition is shown in **Eq. 3.79** [Rathore et al., 2021]

$$d(t) = \breve{d}(t) + \delta d(t) \tag{3.79}$$

It is clearly seen that if no leak is present $d(t) = \breve{d}(t)$. In a similar manner $a_{\mathcal{C}}$ and v are defined in **Eq. 3.80** and **Eq. 3.81**. Here, $\delta a_{\mathcal{C}}$ and δv are the deviations caused by a leakage.

$$a_{\mathcal{C}} = \breve{a}_{\mathcal{C}} + \delta a_{\mathcal{C}} \tag{3.80}$$

$$\mathrm{v} = \breve{\mathrm{v}} + \delta \mathrm{v} \tag{3.81}$$

Lastly, $\bar{p}(t)$ given by **Eq. 7.1** is also defined to be split into two parts in **Eq. 3.82**.

$$\bar{p}(t) = \breve{\bar{p}}(t) + \delta\bar{p}(t) \tag{3.82}$$

As a part of the derivation of the leakage localization algorithm in **Sec. 7.1**, an expression is to be derived for the pressure drops across the edges in the WDN under the influence of a leakage. Taking a starting point in **Eq. 3.9** and relating it to flow dependent pressure drops across the edges due to $\lambda(q(t))$ (pipes) and pressure drops due to differences in geodesic level Δz

$$\Delta p = H^T p = \lambda(q(t)) - \Delta z \tag{3.83}$$

where p are the absolute pressures in the WDN. Then, multiplying from the left by the loop matrix B and collecting terms

$$B(\Delta p + \Delta z) = B\lambda(q(t)) = 0 \tag{3.84}$$

In **Eq. 3.84** the relation given by Kirchhoff's mesh law ($B\Delta p = 0$ and $B\Delta z = 0$) is used. Next, using the fact that $\lambda(q(t))$ can be partitioned into a chord and a tree part $\lambda(q(t)) = [\lambda_{\mathcal{C}}(q_{\mathcal{C}}(t))^T \quad \lambda_{\mathcal{T}}(q_{\mathcal{T}}(t))^T]^T$. Using **Eq. 3.84** and substituting the definition of the loop matrix B and the partitioning of $\lambda(q(t)$ is seen in **Eq.**

$$B\lambda(q(t)) = \begin{bmatrix} I & -\bar{H}_{\mathcal{C}}^T \bar{H}_{\mathcal{T}}^{-T} \end{bmatrix} \lambda(q(t)) = 0 \tag{3.85a}$$

$$\lambda_{\mathcal{C}}(q_{\mathcal{C}}(t)) - \bar{H}_{\mathcal{C}}^T \bar{H}_{\mathcal{T}}^{-T} \lambda_{\mathcal{T}}(q_{\mathcal{T}}(t)) = 0 \tag{3.85b}$$

Then, substituting the expression for $q_{\mathcal{C}}(t) = \gamma a_{\mathcal{C}} d_n(t)$ and the expression for $q_{\mathcal{T}}(t)$ given by **Eq. 3.77b** into **Eq. 3.85b** resulting in **Eq. 3.86**

$$0 = \lambda_{\mathcal{C}}\left(\gamma a_{\mathcal{C}} d_n(t)\right) - \bar{H}_{\mathcal{C}}^T \bar{H}_{\mathcal{T}}^{-T} \lambda_{\mathcal{T}}\left(\gamma \cdot (-\bar{H}_{\mathcal{T}}^{-1} \bar{H}_{\mathcal{C}} a_{\mathcal{C}} - \bar{H}_{\mathcal{T}}^{-1} \mathrm{v}) d_n(t)\right) \tag{3.86}$$

Since $\lambda(q(t))$ is given as the form described in **Sec. 3.3.1**, namely $\lambda(q(t)) = \kappa|q(t)|q(t)$, the equation can be written as seen in **Eq. 3.87**. Here, it is used that $\lambda(q(t))$ is a homogeneous function of degree 2, which allows for using the relation $\lambda(\gamma a_{\mathcal{C}} d_n(t)) = \lambda(\gamma a_{\mathcal{C}}) d_n(t)^2$ [Rathore et al., 2021].

$$\left(\lambda_{\mathcal{C}}\left(\gamma a_{\mathcal{C}}\right) - \bar{H}_{\mathcal{C}}^T \bar{H}_{\mathcal{T}}^{-T} \lambda_{\mathcal{T}}\left(\gamma \cdot (-\bar{H}_{\mathcal{T}}^{-1} \bar{H}_{\mathcal{C}} a_{\mathcal{C}} - \bar{H}_{\mathcal{T}}^{-1} \mathrm{v})\right)\right) d_n(t)^2 = g(a_{\mathcal{C}}, v) d_n(t)^2 = 0 \tag{3.87}$$

As can be seen from **Eq. 3.87** it consists of the variable $a_{\mathcal{C}}$, which is unknown. The remaining variables are known. Because of this it is possible to find a solution for $a_{\mathcal{C}}$ numerically by substituting values of γ and v.

Summary of multiple inlet model

In summary it has been found that vector of flows in all edges $q(t)$ can be found from the reference node inlet flow $d_n(t)$ only using a vector of constants a and a scalar γ under some assumptions. This relation is shown in **Eq. 3.88** for both a single flow q_i in edge i and for the vector of flows q in all edges.

$$q_i(t) = a_i \cdot \gamma \cdot d_n(t) \tag{3.88a}$$

$$q(t) = a \cdot \gamma \cdot d_n(t) \tag{3.88b}$$

Further, under the same assumptions the demand at each node is described by **Eq.**

$$\bar{d}(t) = -\mathrm{v}\sigma(t) = -\mathrm{v}\gamma d_n(t) \tag{3.89}$$

In **Eq. 3.90** the expression for the pressure $p_i(t)$ at the i^{th} node and the vector of pressures $p(t)$ is shown again.

$$p_i(t) = -\alpha_i \cdot d_n(t)^2 - \beta_i + p_n(t) \tag{3.90a}$$

$$\bar{p}(t) = -\alpha d_n(t)^2 - \beta + \mathbb{1} p_n(t) \tag{3.90b}$$

Here, α, and β in **Eq. 3.90** are coefficients which are to be estimated using linear regression on time series data $\bar{p}(t)$, $d_n(t)^2$, and $p_n(t)$. Further, the component flows $q(t)$ is calculated by using the coefficients in a, γ, and measurements $d_n(t)$. From **Eq. 3.89** the demand vector $\bar{d}(t)$ is given by the distribution vector v and the total consumption. Here, a relation of $\sigma(t)$ is found for a multiple inlet WDN. By similar arguments to those in **Sec. 3.5.1** α and β can be found to be equal to the expression in **Eq. 3.91** by using the expression for $a_{\mathcal{T}}$ from **Eq. 3.78**.

$$\alpha = -\bar{H}_{\mathcal{T}}^{-T}\lambda_{\mathcal{T}}(\gamma a_{\mathcal{T}}) = -\bar{H}_{\mathcal{T}}^{-T}\lambda_{\mathcal{T}}(\gamma \cdot (-\bar{H}_{\mathcal{T}}^{-1}\bar{H}_{\mathcal{C}}a_{\mathcal{C}} - \bar{H}_{\mathcal{T}}^{-1}\mathrm{v})) \qquad \beta = \rho g(\bar{z} - \mathbb{1}z_n) \tag{3.91}$$

Lastly, the pressure drops across the edges in the WDN while a leakage is present is given in **Eq. 3.92**

$$\left(\lambda_{\mathcal{C}}\left(\gamma a_{\mathcal{C}}\right) - \bar{H}_{\mathcal{C}}^{T}\bar{H}_{\mathcal{T}}^{-T}\lambda_{\mathcal{T}}\left(\gamma \cdot (-\bar{H}_{\mathcal{T}}^{-1}\bar{H}_{\mathcal{C}}a_{\mathcal{C}} - \bar{H}_{\mathcal{T}}^{-1}\mathrm{v})\right)\right) d_n(t)^2 = g(a_{\mathcal{C}}, v)d_n(t)^2 = 0 \tag{3.92}$$

In the next chapter, the physical WDN called the Smart Water Infrastructure Laboratory (SWIL) is introduced. Here, the different modules, which can be used to construct a real-life WDN is described. Furthermore, a structure of the WDN scenario is chosen such that it is applicable to the SWIL. Lastly, considerations of different parameters related to the WDN are looked upon and approximated.

Laboratory Setup 4

This chapter presents the Smart Water Infrastructure Laboratory, which is used to realize specific WDN scenarios in a real setting. Here, the different modules constituting the Smart Water Infrastructure Laboratory of importance are described. Furthermore, an explanation will follow of the connections between each module which are used to emulate the scenario shown in **Sec. 4.2**. Lastly, a description follows of how the different valve types and pumps are operated and how the sensor measurements are acquired.

4.1 Smart Water Infrastructure Laboratory

The Smart Water Infrastructure Laboratory (SWIL) is used to emulate real-world WDNs by utilizing the different modules such as pipe stations, consumer stations, and pumping stations [AAU, n.d.]. Each of these stations/modules is built with valves that are operated manually by hand. This allows for connecting the different modules with external pipe elements such that a specific WDN can be made to fit a specific scenario. The different modules are described in **Sec. 4.1.2**, **4.1.3**, and **4.1.4**.

4.1.1 Symbol guide

In **Tab. 4.1** the symbols appearing in the drawings in the coming sections are displayed. The symbols used does not follow a standard, but instead relies on a mix of symbols utilized by previous reports and by the documentation of the SWIL.

Symbol	Description
—	Connection
•	Connection node
⌜	Air vent
	Pipe segment
	Pump
	Electronic on/off valve
	Manual external valve
	Electronic control valve

Symbol	Description
	Electronic control valve air
	Check valve
	Flow sensor
	Temperature sensor
	Level sensor/switch
	Pressure sensor
	Differential pressure sensor

Table 4.1: Symbols guide for components used in the drawings off the laboratory setup

4.1.2 Pipe module

The purpose of the pipe module is to emulate pipes in a WDN. The pipe module in the SWIL is shown in **Fig. 4.1** and the schematic of the pipe module is shown in **Fig. 4.2**. The module consists of pipes with a fixed length and a diameter, which varies depending on the pipe channel. This is indicated underneath a pipe element as (length $[m]$/diameter $[mm]$). The two top pipe channels have varying diameter dependent on which pipe module is used (indicated with xx in the diagram). In this project, the first pipe module has a pipe with a diameter of 13 mm. The second has a pipe diameter of 20 mm.

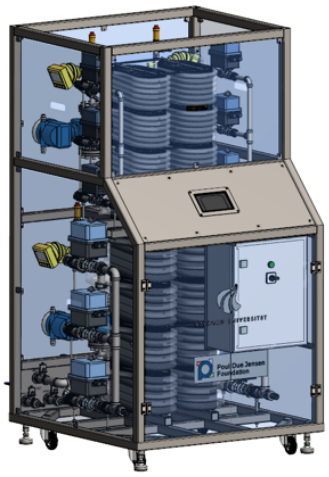


Figure 4.1: 3D illustration of the pipe module [AAU, n.d.]

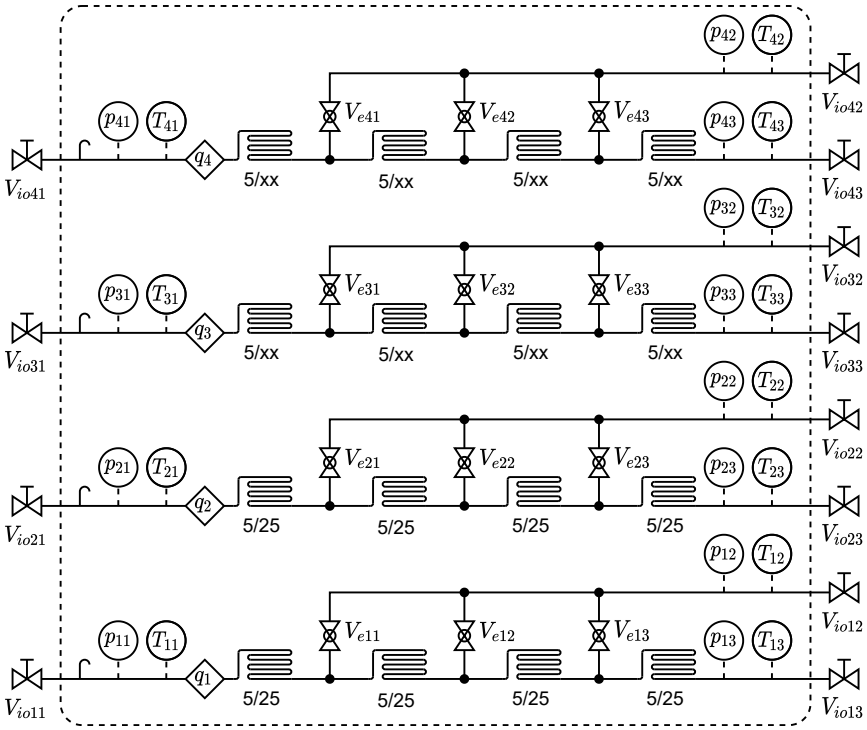


Figure 4.2: Diagram of the pipe module. The symbols used are described in **Tab. 4.1**

As seen in **Fig. 4.2**, the pipe module is build with four pipe channels. Each of these channels consists of different elements of which a description is given in **Tab. 4.1**. The module is equipped with multiple sensors such as pressure (p), temperature (T), and flow (q) as well as electronically controlled on/off valves denoted as e.g. V_{e11}. The pressure and temperature are measured by a combined sensor. **Tab. 4.2** contains a list of the used sensors and valves in this module. Opening or closing these valves allows for choosing a desired total length of a pipe channel. The external input/output valves denoted e.g. V_{io11} are used to make an external connection between other modules in the SWIL.

Type	Diagram name	Part number
Flow	q_i	Endress&Hauser Proline Promag 10-DN15
Pressure Temperature	p_{ii} T_{ii}	Grundfos Direct Sensor RPI+T 0-1.6bar
Electronic Valve	V_{eii}	Bürkert. 2/2 Ball Valve DNxx 8804 (On/Off)

Table 4.2: Parts used in the pipe module. In diagram name i is to be replaced with a number and xx is to be replaced with the pipe diameter at that valve

4.1.3 Consumer station

The consumer station is used to emulate consumers in a water network. It consists of two control valves (V_{11} and V_{12}) that can let water into a tank which is at some pressure (from 0 atm to 0.3 atm). To secure that the tank does not overflow it has an outlet in the bottom with a control valve (V_{13}). This outlet is part of the auxiliary network. The auxiliary network has the purpose of securing that water can be transferred back from the consumer stations to the pump stations to be reused. The auxiliary network is only needed in the laboratory since the fresh water in real life would be pumped from the ground, be used at the consumer and end in the sewers. Furthermore, the module has three external valves which are operated manually. This can be used to connect external pipe elements.

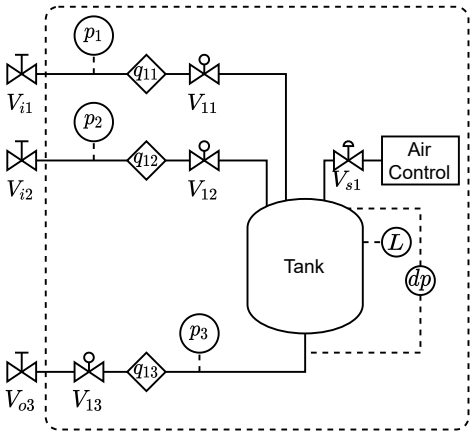


Figure 4.3: Consumer station

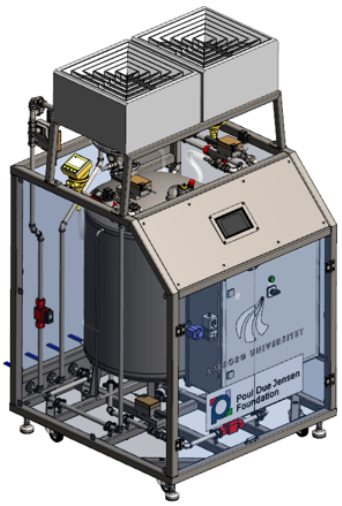


Figure 4.4: 3D illustration of the consumer module [AAU, n.d.]

It has to be mentioned that the consumer module also includes a heating consumer part (seen at the top of the module in **Fig. 4.4**). Since this part is not used in this project it is not drawn in **Fig. 4.3** and will not be described further. Is also has to be mention that the consumer station can be configured to act as a water tank. This will also not be used in this report and is therefore not described. The consumer module is equipped with a set of sensors, a level switch/sensor and as previously mention 3 control valves in total. These are listed in **Tab. 4.3**.

Type	Diagram name	Part number
Flow	q_{ii}	Festo SFAW-32-TG12-TG12-PNLK-PNVBA
Pressure	p_i	Jumo. 404327 0-500mbar
Differential pressure	dp	Jumo. 404382
Level switch	L	Sick LFV230-XXSGBTPM0100
Valve air	V_{si}	Danfoss EV210B (#032U3655) + BE024DS
Control valve	V_{ii}	Belimo LQR24A-SR DN15 + R2015-1-S1
Air control		Festo. VPPE

Table 4.3: Parts used in the consumer station. In diagram name i is to be replaced with a number and xx is to be replaced with the pipe diameter at that valve

The performance of the valves is tested by applying steps of the opening degree (OD) in **App. A.2.1**. Here a significant input-output delay of around 10 seconds from an input is applied to the output shows on the flow and the pressure sensor. This delay is believed mainly to come from the opening of the valve (input delay) as these adjust fairly slow. The delay from input to output in the valves is not accounted for in this report as the valves only are seen as a disturbance. Further, the controller made for the valves is hand-tuned to be stable. This hand-tuned controller is described in **App. A.7**.

4.1.4 Pumping station

The pumping station has two purposes. Firstly, being a pumping station which supplies water to the consumers in the network and secondly being a part of the auxiliary network where the pumps are used to return the water from the consumers. A 3D sketch of the physical module is shown in **Fig. 4.5** and a diagram showing the components in the pumping station and how they are connected is presented in **Fig. 4.6**.

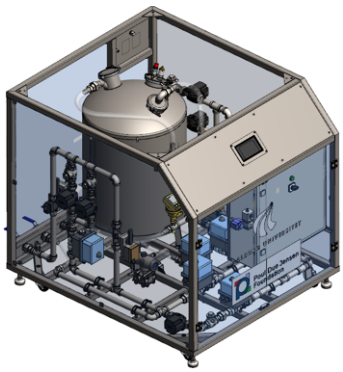


Figure 4.5: 3D illustration of the Pumping station [AAU, n.d.]

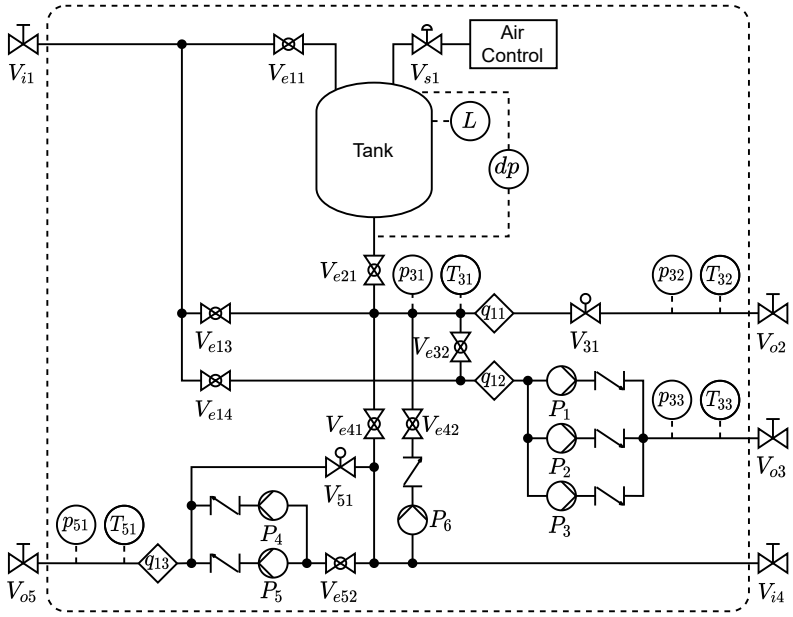


Figure 4.6: Pump station

As with the other modules the pumping station can be configured in a number of different ways by opening and closing the electronic ball valves. By opening the valves V_{e21}, V_{e32} and closing V_{e13}, V_{e14}, V_{e41}, V_{e42}, and V_{31} a path is made from the bottom of the tank to the input of the three parallel coupled pumps P_1, P_2 and P_3, which can then be used to supply water to the network with the tank acting as a reservoir.

The pumps P_4 and P_5 are used as a part of the auxiliary network to return the water from the consumer tanks to the tank in the pumping station. This is done by making an external connection with a pipe from V_{o5} to V_{i1} and opening the two electronic ball valves V_{e11} and V_{e52}.

A list containing the part numbers of the different components in the pumping station are shown in **Tab. 4.4**.

Type	Diagram name	Part number
Flow	q_{ii}	Festo SFAW-100-G1-G1-PNLK-PNVBAM12
Pressure Temperature	p_{ii} T_{ii}	Grundfos Direct Sensor RPI+T 0-1.6bar
Differential pressure	dp	Jumo. 404382
Level switch	L	Sick LFV230-XXSGBTPM0100
Valve air	V_{si}	Danfoss EV210B (#032U3655) + BE024DS
Control valve	V_{ii}	Belimo LQR24A-SR DN15 + R2015-1-S1
Electronic Valve	V_{eii}	Bürkert. 2/2 Ball Valve DN25 8804 (On/Off)
Pump	P_i	Grundfos UPM3 25-75 130
Air control		Festo. VPPE

Table 4.4: Parts used in the consumer station. In diagram name i is to be replaced with a number and xx is to be replaced with the pipe diameter at that valve

The performance of the pumps are tested by applying a step and observing the response in both flow and pressure. The full results and the plots of this are found in **App. A.2.2**. To summarize these, it is found that the pump, like the valve, has significant delays from the input is applied till it shows on sensor measurements. The setup has around 4 seconds of delay, which is probably split somewhat evenly between input and output delay. Further, it is found that internal filters in both the pressure and the flow sensor results in small overshoots in the outputs. The delay found here must be taken into account when the controller is designed in **Ch. 5**.

4.1.5 Interfacing

The SWIL interfacing is split into several parts which is seen in **Fig. 4.7**. At the core is a PC which acts as the Central Control Unit (CCU). The PC uses Matlab/Simulink as the implementation language for the controller and the communication between each individual module. Each of the modules has its own local computer in the form of a Raspberry Pi, which has two ethernet connections and is thus acting as a brigde between two networks. On one side the Raspberry Pi is connected to a LAN with the CCU and the other modules using the Modbus protocol. On the other ethernet connection, the Raspberry Pi is communicating with the DAQ (data acquisition) Input/Output module via the EtherCAT protocol. The DAQ Input/Output module handles the commands to the actuators and the data from the sensors. With this setup it is possible to access the sensors and actuators on the different modules from Matlab/Simulink on the CCU. Furthermore, a HMI is connected to the Raspberry Pi making it possible to access the sensors and actuators locally.

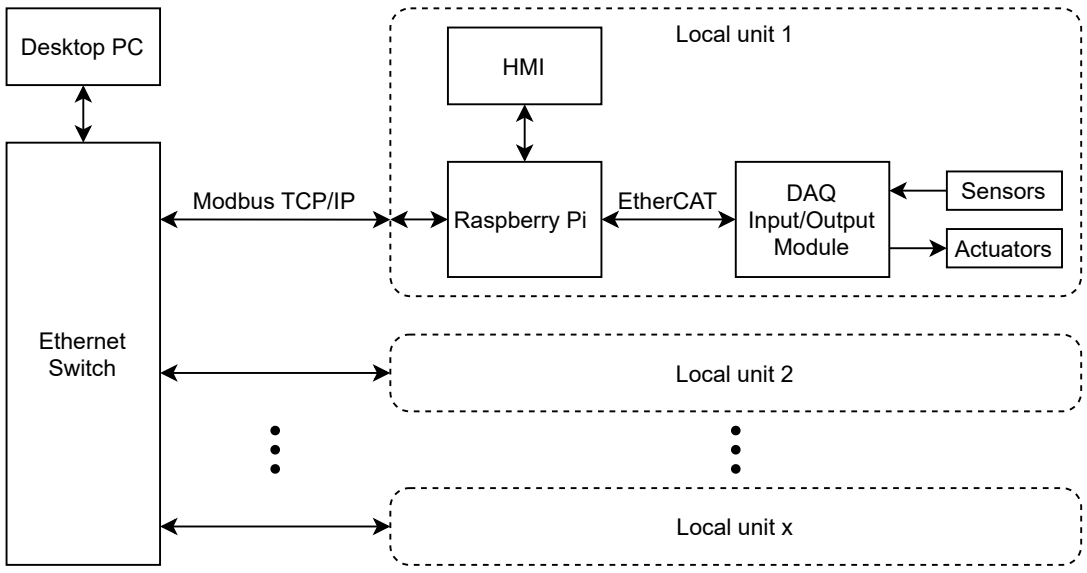


Figure 4.7: Diagram of the different modules and the communication channels constituting the SWIL

4.2 Test scenario

In this section the test scenario is described. This scenario is build in the SWIL with pipe interconnections between each module. The scenario is illustrated in **Fig. 4.8**. The construction of this is made so that it utilizes two pumping stations, which corresponds to using two different pumping inlets. These are denoted e_{14} and e_{15}. In this network two consumer stations are used and the valves e_{11}, e_{12}, and e_{13} are used to emulate consumers of water. As the consumer modules are equipped with two possible inlets to the tank, there is the possibility of adding a single leakage to the network. Furthermore, two pipe modules are utilized to introduce pipes of different lengths and diameter between vertices. These are denoted $e_4, e_5, ..., e_{10}$. The vertices $v_{10}, v_{11}, v_{12}, v_{13}$, and v_{14} are seen as infinite reservoirs, which are open to atmospheric pressure. In every module, the reservoirs are tanks with a level controller, keeping the water level at a certain level. This controller utilizes an auxiliary network consisting of pipes, valves and pumps, which is in charge of transporting water from the consumer stations and into the tanks of the pumping stations. To construct the scenario, each of the modules has to be connected in a specific manner. The overall diagram showing all the modules interconnected, as to construct the specific scenario in the SWIL, is seen in **App. A.12**. In this overall diagram in the SWIL the pipes, pumps and valves are labeled with the corresponding edges and vertex numbers. To interconnect each module, pipes are manually attached to the external valves of each module. However, these interconnections are not taken into account in **Fig. 4.8** as the dynamics of these pipes are neglected based on the assumption seen in **Asm. 4.1**.

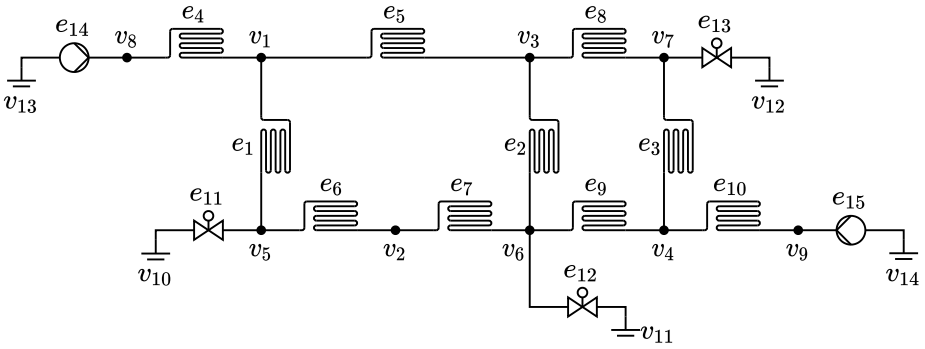


Figure 4.8: Lab scenario for simulation

Assumption 4.1 *The pipes interconnecting the modules are of infinitely small length and infinitely large diameter.*

For this setup, the pipe parameters are seen in **Tab. 4.5**. Here, the lengths, diameters and roughness heights of the pipes connecting the network are shown. The roughness height is chosen to be of plastic pipes while the minor loss coefficient, K_f, is estimated later in **Sec. 4.3**. The height of the nodes are measured manually and presented in **Tab. 4.6**. Furthermore, the valve parameters and pump coefficients are seen in **Tab. 4.7** and **Tab. 4.8**. The pump coefficients are estimated in **App. A.3**.

Pipe	**Length [m]**	**Diameter [mm]**	**Roughness height [mm]**
e_1	30	25	0.05
e_2	10	13	0.05
e_3	10	25	0.05
e_4	5	25	0.05
e_5	5	13	0.05
e_6	5	25	0.05
e_7	5	20	0.05
e_8	5	25	0.05
e_9	15	20	0.05
e_{10}	15	25	0.05

Table 4.5: Pipe parameters

Node	v_1	v_2	v_3	v_4	v_5	v_6	v_7
Elevation [m]	0.0	0.0	0.0	0.0	1.0	1.0	1.0

Node	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}
Elevation [m]	0.1	0.1	1.0	1.0	1.0	0.1	0.1

Table 4.6: Table showing the elevation of the nodes in the network.

Valve	$\boldsymbol{K_{vs}}$
e_{11}	1
e_{12}	1
e_{13}	1

Table 4.7: Valve parameters [Belimo, 2020]

Pump	$\boldsymbol{a_{h2}}$ [s^2/m^5]	$\boldsymbol{a_{h1}}$ [s/m^2]	$\boldsymbol{a_{h0}}$ [•]
e_{14}	$3.9205 \cdot 10^6$	11.3179	$8.8011 \cdot 10^{-4}$
e_{15}	$3.9205 \cdot 10^6$	11.3179	$8.8011 \cdot 10^{-4}$

Table 4.8: Estimated pump coefficients

4.2.1 Graph model

To simulate the scenario in **Fig. 4.8** it has to be redrawn in terms of the graph framework described in **Sec. 3.3**. Here, the components corresponds to edges and the water flowing from the reservoirs are drawn as demands going into the network. The graph with flow directions is seen in **Fig. 4.9**.

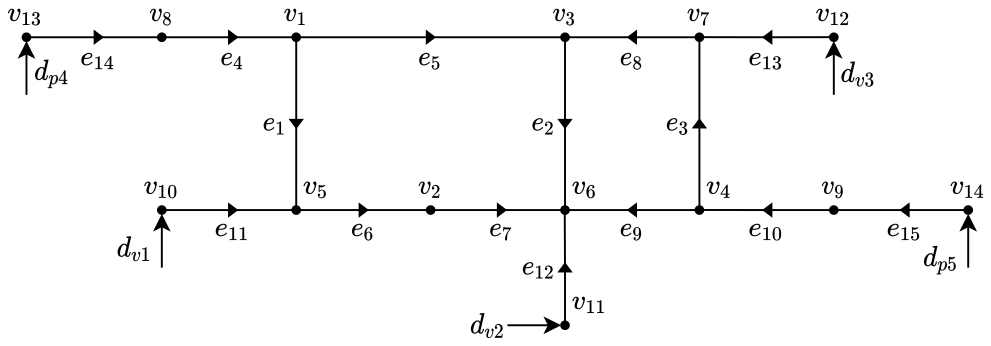


Figure 4.9: Graph of the lab scenario for simulation

Using the graph theory from **Sec. 3.1**, the incidence matrix H, loop matrix B, and the mapping from vertices to demands F are constructed, as seen in **App. A.4**.

4.2.2 Reduced-order model

The scenario from **Fig. 4.8** is redrawn in **Fig. 4.10** in terms of the reduced-order model framework. Here, the edges corresponding to valves and pumps are simply replaced by demands by **Asm. 4.2**.

Assumption 4.2 *The valves and pumps are able to deliver a specific demand*

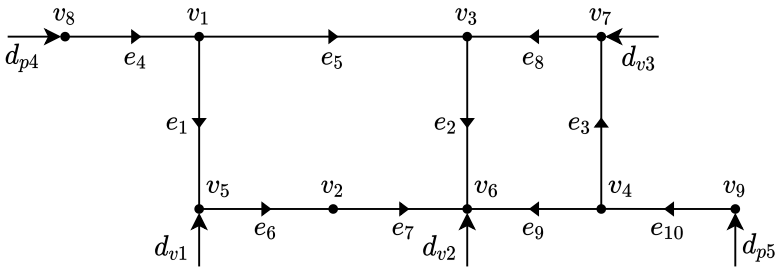


Figure 4.10: Reduced-order graph of the lab scenario for simulation

4.2.3 Sensor & leakage nodes

In a real-life water distribution network, every node will not have a pressure sensor attached to them. Further, not every edge will have a flow sensor attached to them. Lastly, the demand flow might not either have a flow sensor attached that is capable of transmitting back to the distributor live.

Therefore the constructed leakage detection/localization must not require a system that is full of sensors. Therefore, to keep this project relevant in real life, it is chosen to limit the number of sensors available for both the control and the leakage detection/localization.

For a real WDN, it is expected that there will be sensors around the inlet, meaning that there are flow sensors to measure the inlet demands and pressure sensors to measure the pressure at the inlet node. In the case of the SWIL this corresponds to having flow sensors for d_{p4}/q_{14} and d_{p5}/q_{15}. It further corresponds to having pressure sensors at v_8 and v_9, namely that p_8 and p_9 are measured. However, to reduce the number of sensors per node, to get closer to a real scenario, it is chosen not to use p_8 as this sensor isn't strictly required.

It is chosen to see two additional pressure sensors in the SWIL as available. This number comes from that it would be difficult to say anything with only a single sensor. On the other hand, seeing three or more pressures as available is getting close to having a sensor in each node. Even two is a bit high considering that there only are 7 nodes that are not inlet nodes. By this argumentation, two pressure sensors are used since it is seen as the best compromise.

When choosing the locations of the pressure sensors several factors should be considered. In a real WDN, a thing that might be smart to consider is placing the sensors at nodes that

are easily accessible (cheap to get to). Most importantly is placing the pressure sensors at nodes where the pressure changes as much as possible if there is a leak present, this is very important for the leakage detection. Further, the pressure sensors must be placed so that the changes in the measured pressures coming from a leak at any node is as different as possible from the changes at any other node. The last part is important for leakage localization.

There exist several papers describing methods for optimizing the placement but it is still a large research field. [Ribeiro et al., 2015] describes a method that uses the TrustRank Algorithm to identify the nodes where the pressures estimated from the model is most certain. This approach can be smart if the measured pressures are subtracted from the estimated to generate residuals. [Ribeiro et al., 2015] further describes some alternatives to this. The other methods range from defining the best location as a trade-off between reliability and deployment costs, optimizing for pressure sensitivity and using Monte Carlo simulations for finding sensor placements that make leakage localization easy.

The above indicates that the topic of sensor placement is quite complex. Further, currently, it seems that there does not exist an easy plug and play solutions. Therefore it is chosen not to focus on this topic and instead just choose two sensor placements somewhat arbitrarily with some distance between them. It is chosen to place the pressure sensors at v_3 and v_5 corresponding to p_3 and p_5.

In a water distribution network, leakages can appear everywhere. However, for simplification and as an assumption a leakage is seen as appearing only at nodes. This assumption will be close to being correct as long as each edge is short, which means that the distance between each node is kept small. When emulating a leak in the SWIL the leak is made so that it appears in the node. For practical reasons, it selected to only emulate leaks in a few nodes on the SWIL. This selection comes from that it takes a long time to reconfigure the SWIL to change the leak node. It is arbitrarily chosen that on the SWIL leakages are emulated to appear at node 1 and node 6 (v_1 and v_6). However, in simulation, a leak at every node is made to test the suggested leakage detection and leakage localization method.

Measured pressures	p_3	p_5	p_9
Measured flows	d_{p4}/q_{14}	d_{p5}/q_{15}	
Leakage nodes	v_1	v_6	

Table 4.9: Reduced set measured states and leak nodes in the SWIL. These are used for control, leakage detection and leakage localization

4.3 Parameterization

In this section, several parameters to be able to run the simulation of the SWIL and run a test in the SWIL is obtained. These include parameters for the demand curve, the demand distribution and the friction coefficients of the pipes.

4.3.1 Consumption curve and demand distribution vector

To emulate consumers in the SWIL a numerical demand curve for each consumer is needed. This numerical demand curve is set as a reference which is to be held by a controller. It is assumed that each consumer demands has a fixed fraction of the total nominal demand, which is denoted $\breve{\sigma}(t)$. For this to be possible it is also noted that the consumers need to be of the same type. This is assumed to be the case as described in the project limitations (Sec. 2.5). The current nominal total demand at a given time t is given in **Eq. 4.1**. Here $\sigma^\star$ is the average total demand (often called the base demand) and $M(t)$ is a time-dependent multiplier that has an average value of one. This also correspond to that $M(t)$ for the period 0 to 24 h integrates to 24.

$$\breve{\sigma}(t) = \sigma^\star \cdot M(t) \tag{4.1}$$

The references for the demands of all the outlet nodes are calculated from **Eq. 4.2**. Here $\breve{\mathrm{v}}$ is the nominal value of the demand distribution vector v, which relates to the outlets (as defined in **Sec. 3.5.2**).

$$ref_{\breve{d}}(t) = -\breve{\mathrm{v}} \cdot \breve{\sigma}(t) = -\breve{\mathrm{v}} \cdot \sigma^\star \cdot M(t) \tag{4.2}$$

The reference for the pump is set to be a fixed fraction of actual total demand $\sigma(t)$ which is defined as: $\sigma(t) = \breve{\sigma}(t) + \delta\sigma(t)$, where $\delta\sigma(t)$ is the demand caused by the presence of a leak. The reference for the flow controlled pump is expressed in **Eq. 4.3**. In this equation $\grave{\breve{d}}(t)$ and $\grave{\mathrm{v}}$ is the part of the demand distribution vector related to the flow controlled inlets. Further, it is used that the fraction supplied of the total demand should not change from the nominal case to the case where a leak is present ($\grave{\breve{\mathrm{v}}} = \grave{\mathrm{v}}$). Lastly, the conversion between $\sigma(t)$ and $d_n(t)$ from **Eq. 3.76** is also used.

$$ref_{\grave{\breve{d}}}(t) = -\grave{\breve{\mathrm{v}}}\sigma(t) = -\grave{\mathrm{v}}\sigma(t) = -\gamma\grave{\mathrm{v}}d_n(t) \tag{4.3}$$

The parameters for the base demand ($\sigma(t)$) and the nominal fraction of usage ($\breve{\mathrm{v}}$) is chosen to be: $\sigma^\star = 0.75$ m^3/h and $\breve{\mathrm{v}} = [0_{1\times4} \quad 0.25 \quad 0.42 \quad 0.33 \quad -0.6]^T$. The dimension of the vector $\breve{\mathrm{v}}$ fits with the reduced-order model.

Consumption curve/demand multiplier curve

In **Sec. 2.1** the typical consumer behaviour over a day is described. The users are selected to have the category of a single-family. In [Rathore, 2020] parameters for such a curve is listed. This curve however does not match a 100% with the one shown **Fig. 2.1** as it has the highest peak in the morning and not in the evening. This is deemed to be irrelevant to this report. The curve from [Rathore, 2020] is described by a Fourier series. It has been found necessary to cramp this curve a bit to make it run on the SWIL (OD for the valves saturates if this is not done). The curve for $M(t)$ is defined as a second-order Fourier series described by **Eq. 4.4**.

$$M(t) = a_{m0} + a_{m1}\cos(w_m t) + b_{m1}\sin(w_m t) + a_{m2}\cos(2w_m t) + b_{m2}\sin(2w_m t) \tag{4.4}$$

The Fourier series have the coefficients shown in **Tab. 4.10**. The time unit for these coefficients is hours.

Name	a_{m0}	a_{m1}	a_{m2}	b_{m1}	b_{m2}	w_m
Value	1	-0.155	0.044	-0.217	-0.005	0.261

Table 4.10: Coefficients for the Fourier series 24 hour demand multiplier curve

These coefficients give the demand multiplier curve shown in **Fig. 4.11**.

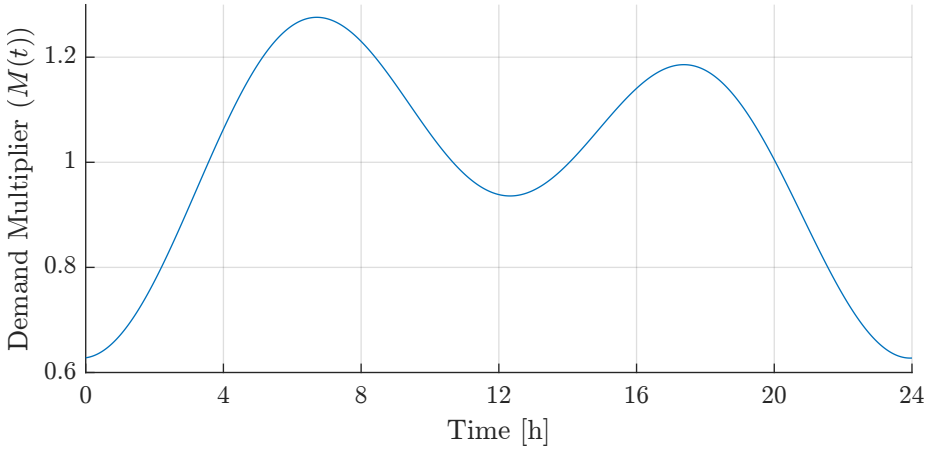


Figure 4.11: The demand multiplier curve used for multiplying the base demand to obtain the total demand

If a 24-hour demand multiplier curve is used on the setup it would take a long time before any results could be found. It is therefore decided to speed up the demand multiplier curve. How much it should be speedup is a trade-off, increasing the speed significantly would make it fast to run through many different scenarios. On the other hand one should not speed it up so much that the system doesn't reach some steady-state like behaviour at each multiplier region. It is chosen to make a day on the setup 2 hours long, meaning a 12 times speedup from real-time. The demand multiplier curve is easily changed to this by multiplying w_m with 12 giving $w_m = 3.132$. The new curve that is used for the rest of this report is shown in **Fig. 4.12**.

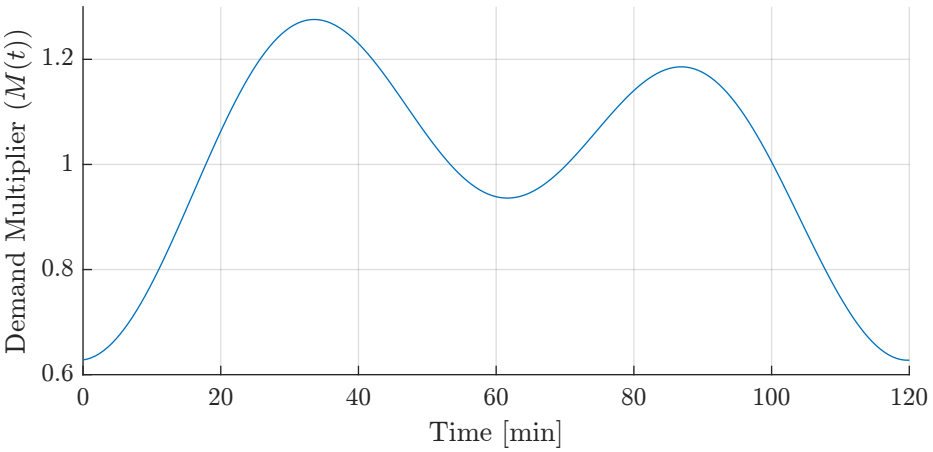


Figure 4.12: The speedup demand multiplier curve used for multiplying the base demand to obtain the total demand

4.3.2 Dynamic model: Friction pipes

The pipe model in **Sec. 3.3** contains a flow-dependent friction term which includes a parameter describing the minor losses in a pipe, namely the minor loss coefficient, K_f. Normally, this coefficient is calculated via adding together coefficient pr bend, narrowing etc. However, this is difficult on the SWIL. Instead, K_f is estimated via steady-state data and the pipe model. The model for the pressure drop over a pipe in steady-state is described by **Eq. 3.23**. By isolating the friction term **Eq. 4.5** is obtained,

$$\kappa = (\rho g \Delta z + \Delta p) \cdot \frac{1}{|q|q} \tag{4.5}$$

which can be evaluated with flow and pressure measurements from the real system to get an estimate of κ. The definition for κ is given in **Eq. 4.6** which is isolated for K_f yielding **Eq. 4.7**.

$$\kappa = f\frac{8L\rho}{\pi^2 D^5} + K_f\frac{8\rho}{\pi^2 D^4} \tag{4.6}$$

$$K_f = \left(\kappa - f\frac{8L\rho}{\pi^2 D^5}\right) \cdot \left(\frac{8\rho}{\pi^2 D^4}\right)^{-1} \tag{4.7}$$

A test was done on the system by setting the control inputs to the pumps and control valves constant and wait for the system to settle. Values for the flow, q, and the pressure drops, Δp was recorded and **Eq. 4.5** and **Eq. 4.7** was used to get the values for K_f for each of the pipes. The results are shown in **Tab. 4.11**. Furthermore, to make comparison between the results for each pipe easier the obtained values for K_f is divided by the length of the corresponding pipes to get a value for K_f per meter denoted $K_{f,m}$.

Edge #	q [m^3/h]	Δp [Bar]	K_f	$K_{f,m}$
1	0.30665	0.1215	104.4	3.4811
2	0.065994	0.1065	44.08	4.4084
3	0.1775	0.1016	48.63	4.8629
4	0.44979	$-9.683\cdot10^{-6}$	22.68	4.5365
5	0.14262	0.02269	33.54	6.709
6	0.12031	-0.09998	-88.07	-17.614
7	0.12031	0.1077	154.9	30.99
8	-0.076628	-0.1014	330.8	66.164
9	0.12754	0.1068	96.05	6.4033
10	0.30504	-0.003835	14.95	0.99691

Table 4.11: Results from the minor loss estimation test

Looking at the results, it is seen that the value for $K_{f,m}$ on a few of the pipes, edge 6, 7 and 8, seems to deviate a lot from the rest. Especially the obtained estimate for edge 6 seems off since it is negative, which is not a sensible value for the minor loss coefficient. The odd value is therefore suspected to have been caused by a measurement error. Since the rest of the values for $K_{f,m}$ are fairly close grouped together it is chosen to take a mean of these to use for all the pipes and ignore the outliers. Doing this results in a mean value of $K_{f,m}$ of 4.49. The minor loss coefficients for each of the pipes is then this value multiplied by the length of the corresponding pipe.

Control design 5

This chapter deals with the construction of a linearized model of the nonlinear dynamical system, which takes pure time delays into account. After this, poles much faster than the rest are removed by model reduction. Further, this accommodates the problem of the original linearized system not being controllable. Next, the linearized and reduced model is used to design and implement a linear Multiple Input Multiple Output (MIMO) controller to accommodate the theory presented in **Sec. 3.5.2**. Here, one pump is to be flow controlled and the latter to be pressure controlled. The controller developed is an output feedback controller with integral action.

The gains for this output feedback controller is obtained by using a special transform from state feedback gains to output feedback gains, which the authors of this report have not found in other literature. This transform is only possible in the special case where C is a square matrix of full rank. Lastly, the designed controller is implemented in simulation and on the real WDN in the SWIL to compare performance.

In this section, only the controller for the pumps is made. In a real-life, this would be sufficient. Since this is an emulation setup, a controller also needs to be made for the valves that emulate the consumer demands. These controllers are chosen to be simple PI controllers, one for each valve. A description of these are found in **App. A.7**. For the remainder of this section and report, the consumer demands and the opening degree of the consumer valves are seen as disturbances and not as control inputs.

5.1 Linearized model

Since the dynamic model described in **Sec. 3.4** is non-linear the need for a linear model is necessary in order to design a linear control scheme. The general form of a non-linear system model is seen in **Eq. 5.1**. Here, $x(t)$ is the state vector, $u(t)$ is the input vector, f is the function which maps the states and inputs to changes in the states. Furthermore, h is a function that maps to the system outputs. These functions are described by the non-linear dynamic model from **Eq. 3.41** and **3.43**

$$\dot{x}(t) = f(x(t), u(t)) \tag{5.1a}$$

$$y(t) = h(x(t), u(t)) \tag{5.1b}$$

For the system in **Eq. 5.1** a linear representation can be determined around an operating point $(x^\star, u^\star)$. The linear state space system at this operating point is defined in **Eq. 5.2**, where $\tilde{x}(t)$ and $\tilde{u}(t)$ are small signal values denoted as $\tilde{x}(t) = x(t) - x^\star$ and $\tilde{u}(t) = u(t) - u^\star$.

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + B\tilde{u}(t) \tag{5.2a}$$
$$\tilde{y}(t) = C\tilde{x}(t) + D\tilde{u}(t) \tag{5.2b}$$

Here, the state space matrices A, B, C, and D are found by applying the Jacobian and evaluating it at the operating point as seen in **Eq. 5.3**. In the same manner C and D are found by applying the Jacobian for $h(x(t), u(t))$ [Franklin et al., 2015].

$$A = \left.\frac{\partial f(x(t), u(t))}{\partial x}\right|_{x^\star, u^\star} \quad , \quad B = \left.\frac{\partial f(x(t), u(t))}{\partial u}\right|_{x^\star, u^\star} \tag{5.3}$$

As for the scenario described in **Sec. 4.2** the states $x(t)$, which originates from the non-linear model are $x(t) = [q_C^T(t) \quad \bar{d}_f{}^T(t)]^T = [q_1(t) \quad q_2(t) \quad d_{v1}(t) \quad d_{v2}(t) \quad d_{v3}(t) \quad d_{p4}(t)]^T$ and the control inputs $u(t) = [\omega_4(t) \quad \omega_5(t)]^T$. Here, $\omega_4(t)$ is the input to the pump located at e_{14}, which is flow controlled and $\omega_5(t)$ being the input to the second pump located at e_{15} (reference node), which is pressure controlled. From these states and control inputs the linear system matrices are formed. Since the objective is to control the flow $d_{p4}(t) = q_{14}(t)$ and pressure $p_9(t)$ at node v_9 the output mapping function $h(x(t), u(t))$ is chosen such that it only consists of equations of $q_{14}(t)$ and $p_9(t)$. Thus the outputs $y(t)$ from the linearized model are selectively chosen to be $y(t) = [q_{14}(t) \quad p_9(t)]^T$. The resulting state space matrices are shown in **App. A.5.1**. Here the operating points are chosen as $x^\star = [0.270 \quad 0.076 \quad -0.186 \quad -0.314 \quad -0.250 \quad 0.450]^T$, which is in units of m^3/h and $u^\star = [69.44 \quad 67.72]^T$. The operating points are found by first applying a constant control to the real system, SWIL, and let it settle to steady-state. The states are then extracted and used as initial values for the WDN simulation. However, discrepancies are observed, as it did not reach the same steady-state values with the same control input. To reach the same steady states an optimization tool (Steady State Manager) in Simulink is used. This tool found a set of steady-state values that correspond to those of SWIL, by adjusting the control input. Then the simulation and SWIL can reach the same steady-state values through a deviation in control inputs.

From this, the state space matrices are formed. These are shown numerically in **App. A.5.1**. As can be seen in **Eq. 5.4** the eigenvalues of the linearized open loop state space model are all placed in the left half plane. It turns out that the controllability matrix $\mathcal{C} = [B \quad AB \quad ... \quad A^{n-1}B]$ of the linearized system does not have full row rank as $rank(\mathcal{C}) = 4$. Further investigation shows that by using MATLAB to determine the rank, an option can be set for tolerance when indices are simply too large. Adjustment of this tolerance shows that the rank alternates from being full rank to having rank deficiencies, i.e. alternating from being controllable to uncontrollable. The system is then deemed to be weakly controllable. This might be the cause of $\mathcal{C}$ being ill-conditioned as $cond(C) = 3.53 \cdot 10^{16}$. This problem is dealt with in **Sec. 5.2** where a procedure called model reduction is used.

$$\lambda = \begin{bmatrix} -89.742 & -49.337 & -13.541 & -1.762 & -0.952 & -0.851 \end{bmatrix} \tag{5.4}$$

5.1.1 Approximation of time delay

As a consequence of the time delay seen in **Fig. A.3** and **Fig. A.4** it must be incorporated as a part of the linearized model as it generally reduces the stability of a system [Franklin et al., 2015]. When designing a controller for such a system the time delay must be taken into account to ensure stability. An exact time delay in the Laplace domain is given by the left side of **Eq. 5.5** with T_d being the time delay. The exact time delay is approximated by a first order Padé approximation seen at the right-hand side of **Eq. 5.5** [Franklin et al., 2015].

$$e^{-T_d s} \approx \frac{1 - \left(\frac{T_d s}{2}\right)}{1 + \left(\frac{T_d s}{2}\right)} \tag{5.5}$$

As described in **Sec. 5.1** the outputs $y(t)$ are selectively chosen as the outputs of the linearized model. The Padé approximation is then used to approximate a time delay T_d seconds for each $y(t)$. This corresponds to a combined transfer function as seen in **Eq. 5.6**.

$$H(s) = \begin{bmatrix} h_{11}(s) & 0 \\ 0 & h_{22}(s) \end{bmatrix} \tag{5.6}$$

The combined transfer function $H(s)$ can then be represented in state space form as seen in **Eq. 5.7**.

$$\dot{\tilde{x}}_d(t) = A_d\tilde{x}_d(t) + B_d\tilde{u}_d(t) \tag{5.7a}$$
$$\tilde{y}_d(t) = C_d\tilde{x}_d(t) + D_d\tilde{u}_d(t) \tag{5.7b}$$

where A_d, B_d, C_d, and D_d are the time delayed system matrices, $\tilde{x}_d(t)$ is the small signal delayed states, and $\tilde{u}_d(t)$ is the small signal inputs.

5.1.2 Concatenation of state space models

As described in **Sec. 3.3.4** there might be several causes of delays in dynamic systems such as delay of actuator dynamics and/or sensor dynamics. In this section, the delays are combined as one delay and then appended to the linearized system model as to incorporated the effect of the time delay. This is illustrated in **Fig. 5.1**. In this way the output $\tilde{y}(t)$ is used as the input $\tilde{u}_d(t)$ to the time delay model. Here as a simplification the delay is seen as a measurement delay only, that is there is no input delay. Later this assumption is checked by simulating the system with both an input delay and a measurement delay. From **Fig. 5.1** the equation for $\dot{\tilde{x}}_d(t)$ is rewritten as seen in **Eq. 5.8** using the fact that $\tilde{u}_d(t) = \tilde{y}(t) = C\tilde{x}(t) + D\tilde{u}(t)$.

$$\dot{\tilde{x}}_d(t) = A_d\tilde{x}_d(t) + B_d\tilde{y}(t) \tag{5.8a}$$
$$= A_d\tilde{x}_d(t) + B_d(C\tilde{x}(t) + D\tilde{u}(t)) \tag{5.8b}$$
$$= A_d\tilde{x}_d(t) + \Psi\tilde{x}(t) + \Phi\tilde{u}(t) \tag{5.8c}$$

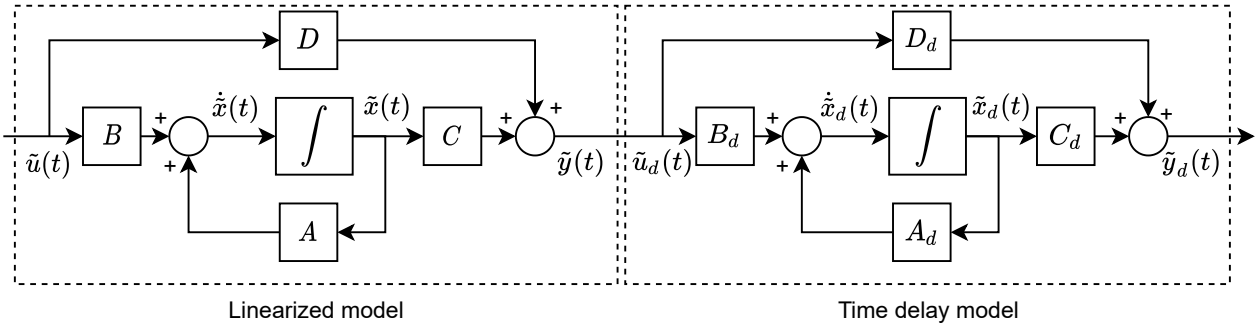


Figure 5.1: Block diagram of the linearized system model and the delay model

where $\Psi = B_d C$ and $\Phi = B_d D$. Using **Eq. 5.2** and **Eq. 5.8** and the concatenated state vector $\tilde{x}_c(t) = [\tilde{x}(t) \quad \tilde{x}_d(t)]^T$, the corresponding state space model is seen in **Eq. 5.9**.

$$\dot{\tilde{x}}_c(t) = \begin{bmatrix} \dot{\tilde{x}}(t) \\ \dot{\tilde{x}}_d(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ \Psi & A_d \end{bmatrix} \begin{bmatrix} \tilde{x}(t) \\ \tilde{x}_d(t) \end{bmatrix} + \begin{bmatrix} B \\ \Phi \end{bmatrix} \tilde{u}(t) \tag{5.9}$$

Next, the output equation $\tilde{y}_c(t)$ for the concatenated model is derived in **Eq. 5.10**.

$$\begin{aligned} \tilde{y}_c(t) &= C_d \tilde{x}_d(t) + D_d \tilde{y}(t) && \text{(5.10a)} \\ &= C_d \tilde{x}_d(t) + D_d (C\tilde{x}(t) + D\tilde{u}(t)) && \text{(5.10b)} \\ &= C_d \tilde{x}_d(t) + D_d C\tilde{x}(t) + D_d D\tilde{u}(t) && \text{(5.10c)} \\ &= \begin{bmatrix} D_d C & Cd \end{bmatrix} \begin{bmatrix} \tilde{x}(t) \\ \tilde{x}_d(t) \end{bmatrix} + D_d D\tilde{u}(t) && \text{(5.10d)} \end{aligned}$$

From **Eq. 5.9** and **5.10** the concatenated state space model is on the form shown in **Eq. 5.11**.

$$\begin{aligned} \dot{\tilde{x}}_c(t) &= A_c \tilde{x}_c(t) + B_c \tilde{u}(t) && \text{(5.11a)} \\ \tilde{y}_c(t) &= C_c \tilde{x}_c(t) + D_c \tilde{u}(t) && \text{(5.11b)} \end{aligned}$$

where $\tilde{u}(t) = \tilde{u}(t)$ and $\tilde{y}_c(t) = [q_{14d}(t) \quad p_{9d}(t)]^T$ are the delayed outputs. For this system, the eigenvalues are located in the left half plane as shown in **Eq. 5.12**. Further investigation of the concatenated system matrices shows that the system is uncontrollable as the system has a state vector of dimension $n = 8$ and $rank(\mathcal{C}_c) = 4$. Here, the plausible cause of this is based on the same reasoning as described in **Sec. 5.2**. In the next section the problem of uncontrollability is dealt with by means of model reduction.

$$\lambda_c = \begin{bmatrix} -89.742 & -49.337 & -13.541 & -1.762 & -0.952 & -0.851 & -0.4 & -0.4 \end{bmatrix} \tag{5.12}$$

5.2 Model reduction

As previously described in **Sec. 5.1**, the concatenated model contains eigenvalues which are located far away compared to the majority as can be seen in **Eq. 5.12**. It is therefore deemed that the two eigenvalues located at -0.4 are fairly dominant. Furthermore, the concatenated model has proven to be uncontrollable with only 3 linearly dependent rows. A way to remove fast modes/eigenvalues and deal with uncontrollability is by applying model reduction.

The method for model reduction used for this project is the residualization method, which is used for systems operating at low frequencies [Skogestad and Postlethwaite, 2005]. As mentioned the dominating eigenvalues are those of the time delayed system with states $\tilde{x}_d(t)$. The remaining eigenvalue with state space equations and states $\tilde{x}_s(t)$ are then removed by means of the residualization method. This is done by firstly partitioning the concatenated system matrices as seen in **Eq. 5.13** and **Eq. 5.14**.

$$\dot{\tilde{x}}_c(t) = \begin{bmatrix} A_{c11} & A_{c12} \\ A_{c21} & A_{c22} \end{bmatrix} \begin{bmatrix} \tilde{x}_s(t) \\ \tilde{x}_d(t) \end{bmatrix} + \begin{bmatrix} B_{c1} \\ B_{c2} \end{bmatrix} \tilde{u}(t) \tag{5.13}$$

$$y_c(t) = \begin{bmatrix} C_{c1} & C_{c2} \end{bmatrix} \begin{bmatrix} \tilde{x}_s(t) \\ \tilde{x}_d(t) \end{bmatrix} + D_c \tilde{u}(t) \tag{5.14}$$

From the partitioned system given by **Eq. 5.13** and **Eq. 5.14** the reduced model are then defined by **Eq. 5.15**, which is a modified version in accordance to [Skogestad and Postlethwaite, 2005] to fit the order of the states in the state vector at which the concatenated matrices are defined. This method is the equivalent of residualizing the states to be removed by setting $\dot{\tilde{x}}_s(t) = 0$, solving for a given state, and substitute into the model [Skogestad and Postlethwaite, 2005].

$$A_r = A_{c22} - A_{c21} A_{c11}^{-1} A_{c12} \tag{5.15a}$$

$$B_r = B_{c2} - A_{c21} A_{c11}^{-1} B_{c1} \tag{5.15b}$$

$$C_r = C_{c2} - C_{c1} A_{c11}^{-1} A_{c12} \tag{5.15c}$$

$$D_r = D_c - C_{c1} A_{c11}^{-1} B_{c1} \tag{5.15d}$$

Here, A_r, B_r, C_r and D_r are the matrices constituting the reduced linearized model with numerical values shown in **App. A.5.3** with states $\tilde{x}_d(t)$. Further investigation of this model shows that $cond(\mathcal{C}_r) = 4.34$ and $rank(\mathcal{C}_r) = 2$, which implies controllability. The eigenvalues of the reduced system is shown **Eq. 5.16**.

$$\lambda_r = \begin{bmatrix} -0.4 & -0.4 \end{bmatrix}^T \tag{5.16}$$

5.3 Controller design

The controller is designed on the reduced model introduced in the previous section. This section will first describe standard state feedback with integral action. Secondly, this section describes how the state feedback designed can be transformed into output feedback utilizing some special properties of this system which arises after model reduction is applied to handle the system being weakly controllable. As mention previously this transform has not been found by the authors in other literature. This transform makes it possible to obtain controller gains by using standard poles placement methods for state feedback and then transforming these gains to work as output feedback gains with the same pole placement. This is smart since $y_r(t)$ is fully measurable whereas the states $x_r(t)$ is not and further doesn't have any physical interpretation since they just describe some approximation to delay. Lastly, this section places the poles at desirable locations.

5.3.1 State feedback with integral action

This section describes state feedback with integral action. This approach is the basis for the design of the output feedback with integral action. The system which the feedback is designed for is displayed in **Fig. 5.2**. The system is described by the state space equations in **Eq. 5.17**.

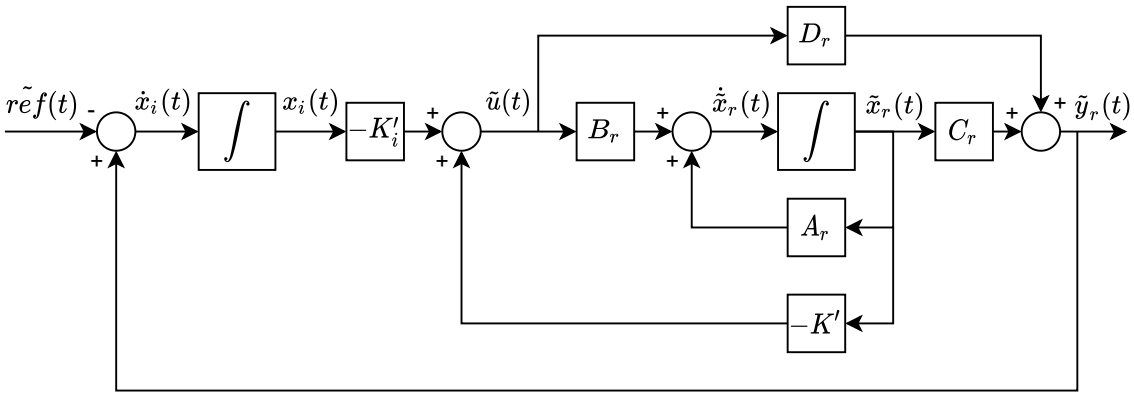


Figure 5.2: System with state feedback and integral action

$$\begin{bmatrix}\dot{\tilde{x}}_r(t)\\ \dot{x}_i(t)\end{bmatrix} = \begin{bmatrix}A_r & 0\\ C_r & 0\end{bmatrix}\begin{bmatrix}\tilde{x}_r(t)\\ x_i(t)\end{bmatrix} + \begin{bmatrix}B_r\\ D_r\end{bmatrix}\tilde{u}(t) + \begin{bmatrix}0\\ -I\end{bmatrix} r\tilde{e}f(t) \tag{5.17a}$$

$$\tilde{y}_r(t) = \begin{bmatrix}C_r & 0\end{bmatrix}\begin{bmatrix}\tilde{x}_r(t)\\ x_i(t)\end{bmatrix} + D_r\tilde{u}(t) \tag{5.17b}$$

The control law is defined in equation **Eq. 5.18**

$$\tilde{u}(t) = -\begin{bmatrix}K' & K_i'\end{bmatrix}\begin{bmatrix}\tilde{x}_r(t)\\ x_i(t)\end{bmatrix} = -K'\tilde{x}_r(t) - K_i'x_i(t) \tag{5.18}$$

When the system is on this form standard tools for placing the eigenvalues are available. For this report Matlabs function *place* is used. The closed loop function of this system is

showed in **Eq. 5.19**.

$$\begin{bmatrix}\dot{\tilde{x}}_r(t)\\ \dot{x}_i(t)\end{bmatrix} = \begin{bmatrix}A_r - B_rK' & -B_rK_i'\\ C_r - D_rK' & -D_rK_i'\end{bmatrix}\begin{bmatrix}\tilde{x}_r(t)\\ x_i(t)\end{bmatrix} + \begin{bmatrix}0\\ -I\end{bmatrix}\tilde{ref}(t) \tag{5.19a}$$

$$\tilde{y}_r(t) = \begin{bmatrix}C_r - D_rK' & -D_rK_i'\end{bmatrix}\begin{bmatrix}\tilde{x}_r(t)\\ x_i(t)\end{bmatrix} \tag{5.19b}$$

This section has given an introduction to state feedback design with integral action. However, the state $x_r(t)$ is unknown and not measurable since it is origins from the Pade approximation and has no physical interpretation. The first signal which is measurable and has a physical interpretation is $y_r(t)$, therefore it is desirable to do feedback on this instead, this is called output feedback. The next section defines the output feedback gains which are derived from the state feedback gains.

5.3.2 Output feedback with integral action

The combined new system with reference, integral action and the output feedback is displayed on **Fig. 5.3**.

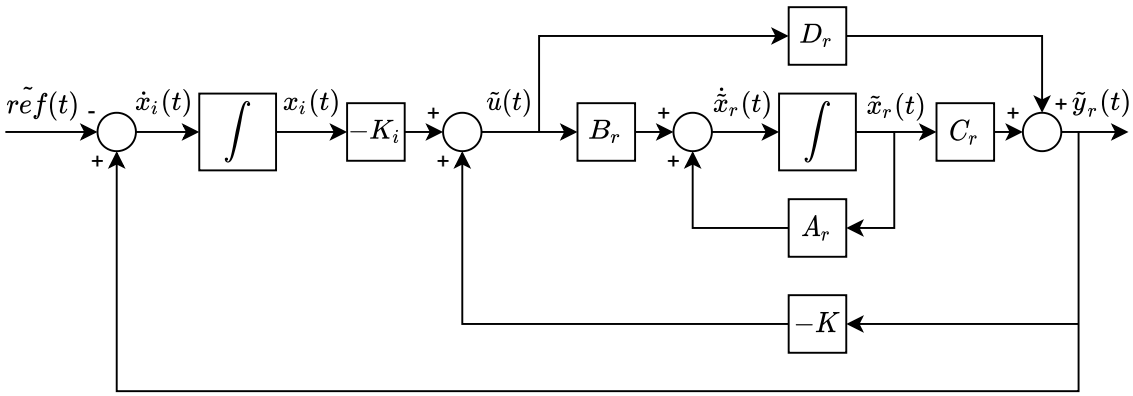


Figure 5.3: Output feedback

The system has the same equations as the ones shown in **Eq. 5.17**. However, it has a different feedback law. This new feedback law is shown in **Eq. 5.20**.

$$\tilde{u}(t) = -\begin{bmatrix}K & K_i\end{bmatrix}\begin{bmatrix}\tilde{y}_r(t)\\ x_i(t)\end{bmatrix} = -K\tilde{y}_r(t) - K_ix_i(t) \tag{5.20}$$

Inserting an expression for $\tilde{y}_r(t)$ in **Eq. 5.20** and rewriting a bit one ends with the expression in **Eq. 5.21d**.

$$\tilde{u}(t) = -K\tilde{y}_r(t) - K_ix_i(t) \tag{5.21a}$$

$$\tilde{u}(t) = -KC_r\tilde{x}_r(t) - KD_r\tilde{u}(t) - K_ix_i(t) \tag{5.21b}$$

$$(I + KD_r)\tilde{u}(t) = -KC_r\tilde{x}_r(t) - K_ix_i(t) \tag{5.21c}$$

$$\tilde{u}(t) = -(I + KD_r)^{-1}KC_r\tilde{x}_r(t) - (I + KD_r)^{-1}K_ix_i(t) \tag{5.21d}$$

The expression in **Eq. 5.21d** is quite similar to the one in **Eq. 5.18**. In **Eq. 5.22** this relation is pointed out.

$$\tilde{u}(t) = -\underbrace{(I + KD_r)^{-1}KC_r}_{K'}\tilde{x}_r(t) - \underbrace{(I + KD_r)^{-1}K_i}_{K_i'}x_i(t) \tag{5.22}$$

Using the newly found relations between K and K' the output feedback gain can be obtained from the state feedback gains by the transform shown in **Eq. 5.23d**.

$$K' = (I + KD_r)^{-1}KC_r \tag{5.23a}$$
$$K' + KD_rK' = KC_r \tag{5.23b}$$
$$K' = K(C_r - D_rK') \tag{5.23c}$$
$$K = K'(C_r - D_rK')^{-1} \tag{5.23d}$$

Similarly using the relation between K_i and K_i' the updated integral gain can be obtained be the transform in **Eq. 5.24b**.

$$K_i' = (I - KD_r)^{-1}K_i \tag{5.24a}$$
$$K_i = (I - KD_r)K_i' \tag{5.24b}$$

Using these gains results in that the closed loop system gets the same poles desired from the state feedback when using output feedback. It has to be noted that this way of doing it is only possible due to the special case of C_r and D_rK' being square matrices.

The closed loop system with output feedback correspond to the system displayed in **Eq. 5.25**. This system can be converted to the one displayed in **Eq. 5.19** by using the relations between K' and K defined in **Eq. 5.23a** and using the relation between K_i' and K_i defined in **Eq. 5.24a**.

$$\begin{bmatrix}\dot{\tilde{x}}_r(t)\\ \dot{x}_i(t)\end{bmatrix} = A_{ri}\begin{bmatrix}\tilde{x}_r(t)\\ x_i(t)\end{bmatrix} + \begin{bmatrix}0\\ -I\end{bmatrix}\tilde{ref}(t) \tag{5.25a}$$

$$A_{ri} = \begin{bmatrix}A_r - B_rK(I + D_rK)^{-1}C_r & -B_rK(I + D_rK)^{-1}D_rK_i - B_rK_i\\ (I + D_rK)^{-1}C_r & -(I + D_rK)^{-1}D_rK_i\end{bmatrix}$$

$$\tilde{y}_r(t) = \begin{bmatrix}(I + D_rK)^{-1}C_r & -(I + D_rK)^{-1}D_rK_i\end{bmatrix}\begin{bmatrix}\tilde{x}_r(t)\\ x_i(t)\end{bmatrix} \tag{5.25b}$$

5.3.3 Selection of the poles

In this section, the placement of the poles is chosen. Since the system is slow and only changes reference slowly, there is no need to fasten the natural dynamics of the system. On the contrary, slowing the system could make sense to increase robustness against fast disturbances. However, slowing the dynamics of the system significantly has been found to increase the k gain while lowering k_i. Increasing k isn't desirable as the state it is multiplied with includes noise. The poles of the initial reduced system with an integrator appended (**Eq. 5.17**) has poles located as seen in **Eq. 5.26**

$$\lambda_{ri} = \begin{bmatrix}-0.4 & -0.4 & 0 & 0\end{bmatrix} \tag{5.26}$$

From the augmentations above it is decided not to move the poles far away from their initial position. By a bit of manual tuning on both the simulation model and on the actual setup, it has been found that gains moving the poles to the locations as seen in **Eq.** yields a satisfying behaviour of the system.

$$\lambda_{ri} = \begin{bmatrix} -0.472 & -0.361 & -0.056 & -0.014 \end{bmatrix} \tag{5.27}$$

This placement of the poles gives the state feedback and integral gain shown in **Eq. 5.28**.

$$K' = \begin{bmatrix} 0.023 & 0.113 \\ 0.000 & 0.135 \end{bmatrix} \qquad K_i' = \begin{bmatrix} 1.113 & 1.161 \\ 0.011 & 1.385 \end{bmatrix} \tag{5.28}$$

Changing this to output feedback by using the relations found in **Eq. 5.23d** and **Eq. 5.24b** gives the result shown in **Eq. 5.29**.

$$K = \begin{bmatrix} 1.711 & 8.043 \\ 0.017 & 9.594 \end{bmatrix} \qquad K_i = \begin{bmatrix} 1.027 & 1.029 \\ 0.010 & 1.228 \end{bmatrix} \tag{5.29}$$

Now it is verified that the original system from **Sec. 5.1.2** is stable with these gains, that is, none of the uncontrolled and controlled states are unstable. Further, it is checked that the closes loop eigenvalues corresponding to the placed eigenvalues of the reduced order system are in somewhat the desired location. To do all this, the integral action has to be appended to the model. This is seen in **Eq. 5.30**, where $\dot{x}_i$ is derived from the block diagram in **Fig. 5.4**.

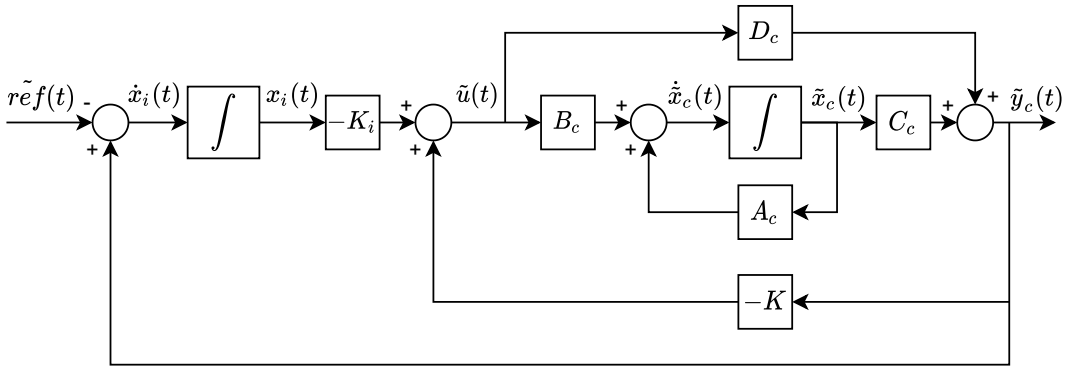


Figure 5.4: Block diagram of the concatenated system model, which is controlled by output feedback with integral action

$$\dot{\tilde{x}}_c(t) = A_c\tilde{x}_c(t) + B_c\tilde{u}(t) \tag{5.30a}$$
$$\dot{x}_i(t) = C_c\tilde{x}_c(t) + D_c\tilde{u}(t) - Ir\tilde{e}f(t) \tag{5.30b}$$

From **Eq. 5.30**, the concatenated system with integral action is formed as seen in **Eq. 5.31**. As the reference does not have influence on the stability of the system it is neglected in further derivation.

$$\begin{bmatrix} \dot{\tilde{x}}_c(t) \\ \dot{x}_i(t) \end{bmatrix} = \begin{bmatrix} A_c & 0 \\ C_c & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_c(t) \\ x_i(t) \end{bmatrix} + \begin{bmatrix} B_c \\ D_c \end{bmatrix} \tilde{u}(t) \tag{5.31}$$

Next, the closed loop system is formed by finding an expression for the control law as seen in **Eq. 5.32**. This is done by utilizing **Fig. 5.4**.

$$\tilde{u}(t) = -K_i x_i(t) - K(C_c \tilde{x}_c(t) + D_c \tilde{u}(t)) \tag{5.32a}$$

$$(I + KDc)\tilde{u}(t) = -K_i x_i(t) - KC_c \tilde{x}_c(t) \tag{5.32b}$$

$$\tilde{u}(t) = \begin{bmatrix} -(I + KDc)^{-1} KC_c & -(I + KDc)^{-1})K_i \end{bmatrix} \begin{bmatrix} \tilde{x}_c(t) \\ x_i(t) \end{bmatrix} \tag{5.32c}$$

Substituting the control law in **Eq. 5.32** into **Eq. 5.31** results in the two expression given in **Eq. 5.33**.

$$\dot{\tilde{x}}_c(t) = A_c - B_c(I + KD_c)^{-1}\tilde{x}_c(t) - B_c(I + KD_c)^{-1}K_i x_i(t) \tag{5.33a}$$

$$\dot{x}_i(t) = C_c - D_c(I + KD_c)^{-1}\tilde{x}_c(t) - D_c(I + KD_c)^{-1}K_i x_i(t) \tag{5.33b}$$

from where the closed loop state space system is given in **Eq. 5.34**.

$$\begin{bmatrix} \dot{\tilde{x}}_c(t) \\ \dot{x}_i(t) \end{bmatrix} = \begin{bmatrix} A_c - B_c(I + KD_c)^{-1}KC_c & -B_c(I + KD_c)^{-1}K_i \\ C_c - D_c(I + KD_c)^{-1}KC_c & -D_c(I + KD_c)^{-1}K_i \end{bmatrix} \begin{bmatrix} \tilde{x}_c(t) \\ x_i(t) \end{bmatrix} \tag{5.34}$$

From the closed loop system in **Eq. 5.34** the eigenvalues are located as seen in **Eq. 5.35**

$$\lambda_{ci} = \begin{bmatrix} -89.732 & -49.329 & -13.43 & -1.724 & -0.938 & -0.828 \\ & & & -0.472 & -0.335 & -0.0593 & -0.0139 \end{bmatrix} \tag{5.35}$$

which shows that the designed output feedback controller with integral action is a stabilizing controller that manages to preserve the eigenvalues λ_s from the linearized system model and the eigenvalues that were placed by choosing the gains K and K' with little deviation.

5.4 Controller test

In this section, the controller designed is tested in simulation and on the lab setup. This will test the robustness of the controller to things that were not taken into account when it was designed. Firstly, it is by simulation tested how different magnitudes of delay affects the stability of the controller. The simulation uses the non-linear model of system described in **Sec. 3.4**. In the same test, the controller is also set to follow the demand curve from **Sec. 4.3.1** to see how it handles getting quite far away from the operating point it is designed to operate around. This also means that the controller is subject to a change in the opening degree of the consumer valves, which gives a disturbance it must be able to handle. The opening degree of the consumer valves, $OD_i(t)$ is controlled by a simple PI-controller as described in **App. A.7**.

When the controller is tested on the lab setup, it is again set to follow the demand curve giving the same disturbances as mentioned for simulation. On the lab setup, the controller is also affected by sensor noise. This again will test the robustness of the controller.

5.4.1 Simulation

The simulation test consists of three subtests. In the first test, the extreme case for smaller than anticipated delay is tested, that being that there isn't any delay at all. In the second test, it is assumed that there is a delay of 5 seconds. This is the delay the controller is designed for. However, this 5-second delay is split into a 2.5-second input delay and a 2.5-second measurement delay. As a reminder, the delay has for the design phase been seen as a combined 5-second measurement delay. The split delay is a more realistic estimate of how the delay behaves in the real setup. Lastly, the controller is tested with a delay double the size it was designed for. That is a 5-second input delay and a 5-second measurement delay.

No delay in the feedback loop

Here the controller is tested without any delay in the system. In **Fig. 5.5** the control inputs for the pump is plotted and in **Fig. 5.6** the output state of the system is shown. Both plots clearly show that the controller doesn't have any trouble with meeting the references set and has zero tendencies toward oscillation/instability when no delay is present. This was to be expected since this is by far the easiest test.

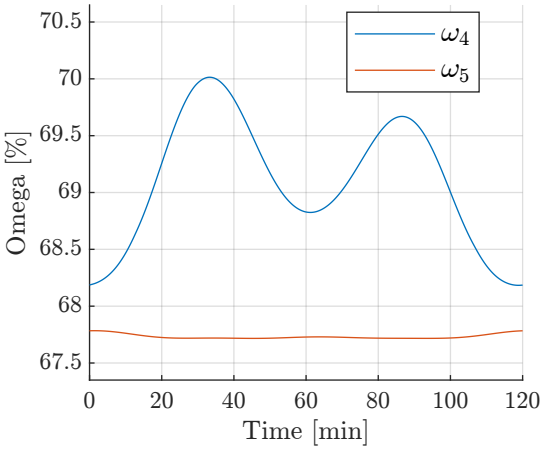


Figure 5.5: Control inputs - No delay in the feedback loop

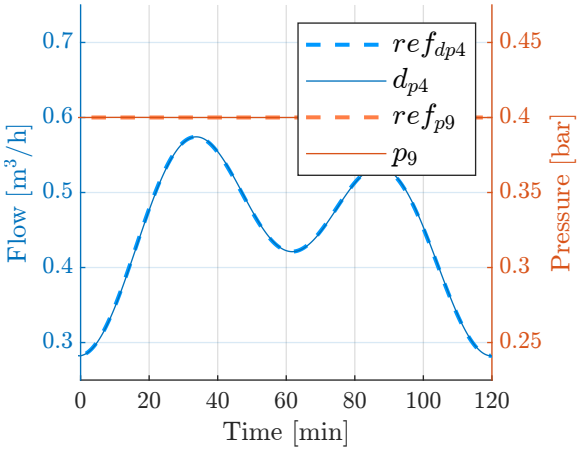


Figure 5.6: Output states - No delay in the feedback loop

5 seconds of delay in the feedback loop

In this test, a 2.5-second input delay and a 2.5-second measurement delay is introduced. The control inputs are shown in **Fig. 5.7** and the output states are plotted in **Fig. 5.8**. Again, these plots show little sign of unstable behaviour. However, they show that the control input has a small tendency to very small oscillations at low flows. These are however deemed so small that they aren't a cause for concern.

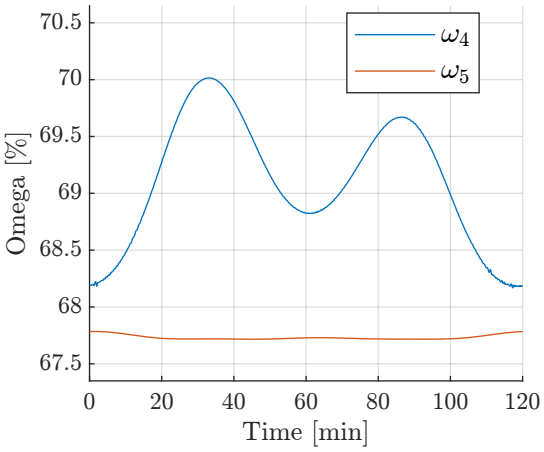


Figure 5.7: Control inputs - 5 seconds of delay in the system

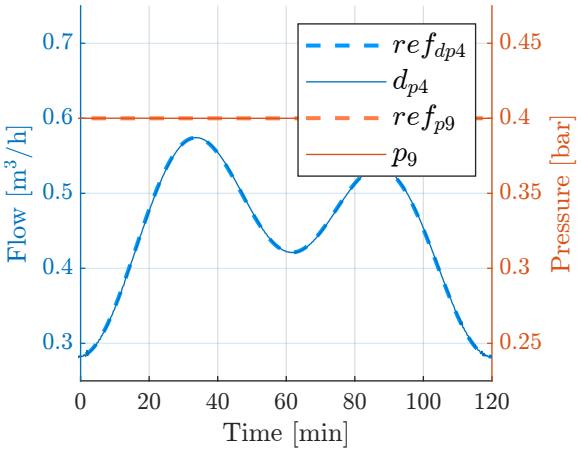


Figure 5.8: Output states - 5 seconds of delay in the feedback loop

10 seconds of delay in the feedback loop

Lastly the controller is testes with an input delay of 5 seconds and an output delay of 5 seconds. The control inputs are shown in **Fig. 5.9** and the output states are shown in **Fig. 5.10**. These clearly show the same tendency as for the 5-second test, having small oscillations at small flows. As an aside, the system behaves marginally stable from around 15 seconds of total delay and becomes completely unstable from around 25 seconds of total delay.

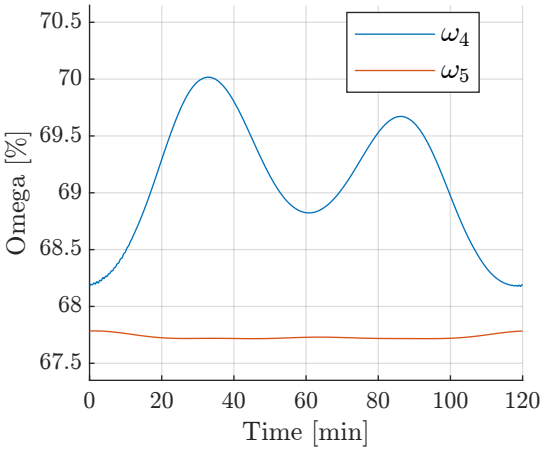


Figure 5.9: Control inputs - 10 seconds of delay in the feedback loop

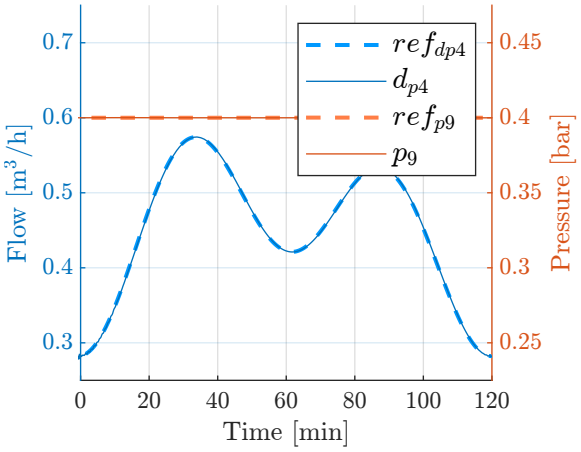


Figure 5.10: Output states - 10 seconds of delay in the feedback loop

5.4.2 Lab test

For the lab test the controller is set to follow the demand curve described in **Sec. 4.3.1**. To verify that the control is able to keep the output references to a fair degree these are plotted in **Fig. 5.11**. Further it is verified that the control outputs doesn't change rapidly and in an unstable way in **Fig. 5.12**. Lastly it is observed that the states of the system are not unstable when using this controller, this is seen in **Fig. 5.13**.

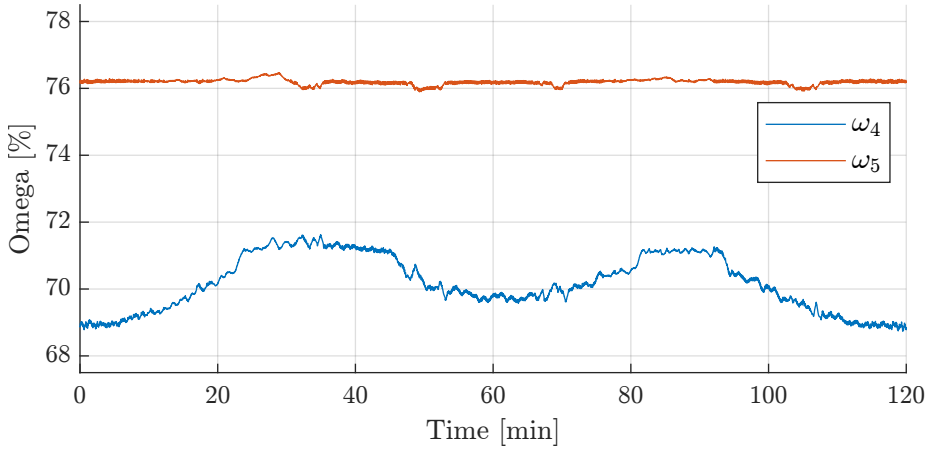


Figure 5.11: Control signals for the pumps

In **Fig. 5.11** the control input to the system is plotted. They are more fluctuating than the ones from simulations. This can be caused by the noise which is added by the sensors. However, some of it also might come from the controller has a slight tendency to oscillations. The oscillations are especially noticeable when the controller follows a small flow reference. As seen in the previous section, this is also the case in simulation. On the lab setup, the oscillations seem to be significantly bigger than in simulation. Lastly, it is also seen that the operating point of the pumps in the lab is somewhat different from the one from simulation. This is probably due to modelling inaccuracies and productions differences in the pumps.

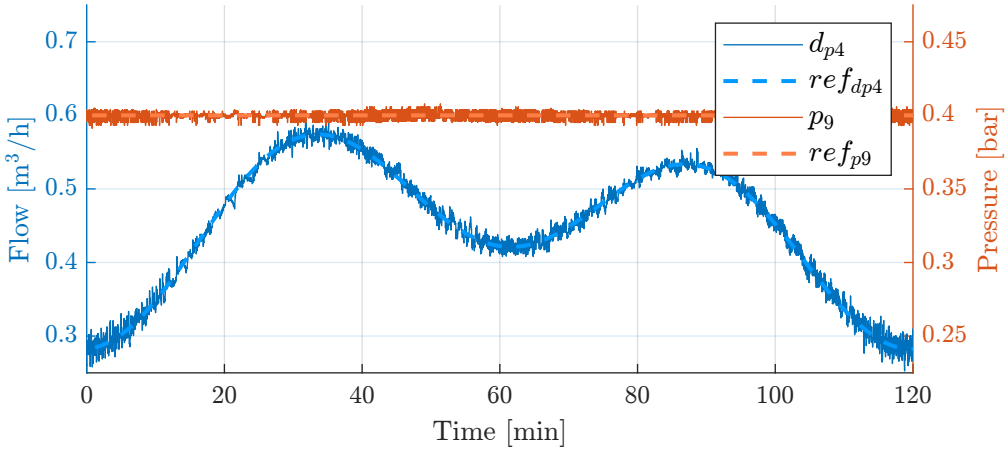


Figure 5.12: The two output states plotted together with their references

In **Fig. 5.12** the output states are plotted. The output states have a lot more variation than the ones from the simulations. Some of it is expected to come from the measurement noise and some of it is expected to be caused by the small oscillations of the controller. The plot also shows that the controller still can follow the demand curve. Clearly, there could be optimized on the controller to try to get the small oscillations away. Further filtering the sensor measurements and/or the controller output might also be beneficial. It is also to be mentioned that the control of the valves introduced significant variations and

oscillations since they never really settle due to a dead-band in them. This can be seen by the valve states ($d_{v1}(t)$, $d_{v2}(t)$ and $d_{v3}(t)$) in **Fig. 5.13**. The controller for the valves is further described in **App. A.7**. It is chosen not to do more to these variations since these can be used to test the robustness of the leakage detection. Further, in a real-life setting significant noise and variations from the ideal demand curve is to be expected.

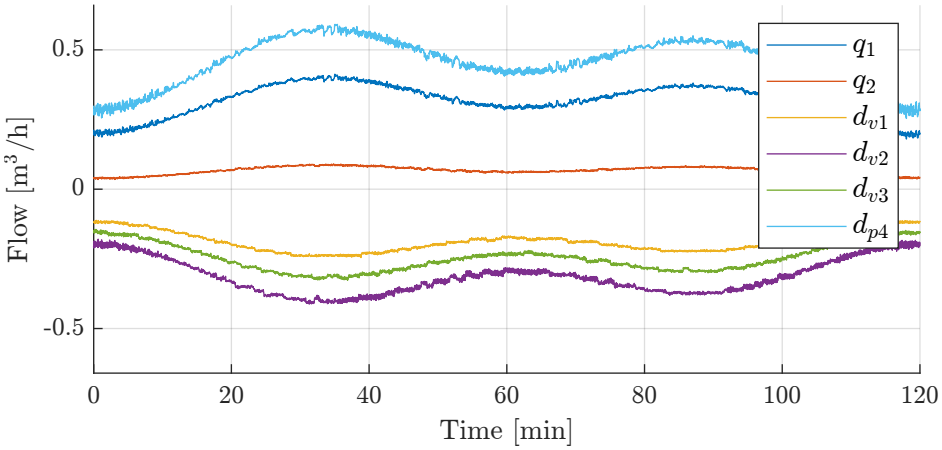


Figure 5.13: All the states except the x_i states

Lastly in **Fig. 5.13** the states are plotted. These are mostly for verifying that none of the states is unstable with the designed controllers. This is not the case but obviously, the small oscillations from the controller combined with sensor noise also show on the states.

Leakage detection 6

This chapter deals with the detection of leakages in a WDN. Taking a starting point the method described in [Jensen and Kallesøe, 2016] and the reduced model described in **Sec. 3.5** the coefficients of the reduced model are estimated to generate a residual. Next, the characteristics of the residual are investigated to see the behaviour when leakage appears in a WDN and to determine which distribution the noise can be approximated as. Furthermore, the residual is passed to an algorithm, which makes a WDN with a leakage stand out from a nominal model of a WDN by looking at the change of parameters. Here, a choice of algorithm is made based on the use case. Furthermore, the algorithm is implemented both in simulation and used against measured data from SWIL.

The general structure of the leakage detection scheme is illustrated in **Fig. 6.1**. The generation of residuals r_i is based on direct pressure measurements at selected nodes in the WDN and the estimated pressure $\hat{p}_i$ at those nodes calculated by the reduced order model. To detect changes in a WDN caused by leakages, the residual is used as it contains information about how the WDN is functioning. Any deviation from the reduced model will show by the residual. However, the deviation can prove difficult to detect for minor deviations. Thus, a detection algorithm is needed to make a clear distinction between both large and small deviations from the model.

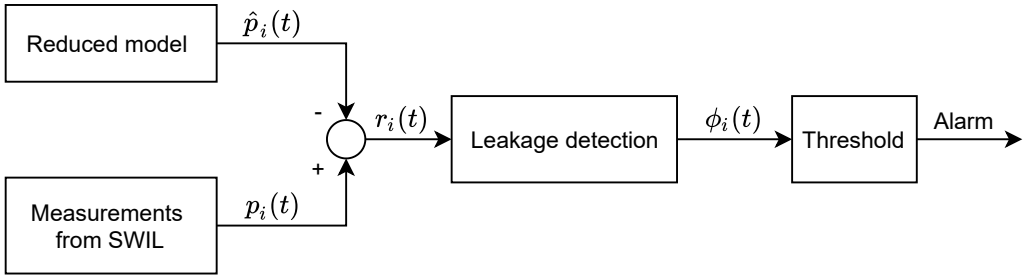


Figure 6.1: Block diagram showing the leakage detection principle using a residual generated from direct measurements and the reduced order model

6.1 Estimation of coefficients for reduced order model

The aim of this section is to estimate the coefficients for the reduced model used to generate the residual as shown in the block diagram in **Fig. 6.1**. Since developing and tuning the leakage detection algorithm might require multiple test, which is time consuming on the real system, it is advantageous to do this in simulation. In this case the measurements of p_i will come from the full simulation model described in **Sec. 3.4** instead of from sensors

on the real system. Because of discrepancies between the simulation model and the real system a second reduced model for the simulation model is also made.

6.1.1 Theory

In **Sec. 3.5** it is shown that the steady state node pressures in the network can be mapped from the pressure and demand at the reference node with the constants α and β as seen in **Eq. 6.1**

$$p_i(t) = -\alpha_i \cdot d_n(t)^2 - \beta_i + p_n(t) \tag{6.1}$$

where p_i is the pressure at the i^{th} node and d_n and p_n is the demand and pressure at the reference node respectively. In order to use this model the coefficient α and β needs to be found. Since **Eq. 6.1** is linear in these two coefficients a suitable method for estimating them is through linear regression [Montgomery, 2012], which aims to find a linear model for a set of observations. The linear model is shown in **Eq. 6.2**

$$Y_i(t) = A_i \cdot X_i(t) + B_i + \epsilon_i(t) \tag{6.2}$$

where A and B are the regression coefficients, X and Y are referred to as the regressor and response respectively and ϵ is an error term representing the difference between Y and the line $A \cdot X + B$. The pressure model in **Eq. 6.1** is transformed to this form by applying a change of variables as shown in **Eq. 6.3**.

$$Y_i(t) = p_i(t) - p_n(t) \tag{6.3a}$$
$$X_i(t) = d_n(t)^2 \tag{6.3b}$$
$$A_i = -\alpha_i, \quad B_i = -\beta_i \tag{6.3c}$$

By obtaining a set of observations of X_i and Y_i the method of least squares can be used to estimate the regression coefficients, A and B [Montgomery, 2012], which can then be transformed back to the coefficients α and β, thus completing the reduced pressure model in **Eq. 6.1**.

The least squares approach used to determine the A and B coefficients is the one described in [Geil, 2008] where they are found as:

$$\begin{bmatrix} B_i \\ A_i \end{bmatrix} = (C_i^T C_i)^{-1} C_i^T \cdot y_i \tag{6.4}$$

where $C_i = [\mathbb{1} \quad x_i]$ with x_i being a vector of X_i observation and y_i is a vector of the corresponding observations of Y_i.

6.1.2 Data collection and results from lab setup

As stated in **Sec. 4.2.3** it is chosen to work with a flow sensor to measure the demand at the reference node, v_9, together with a pressure sensor at the same node. These sensors gives the measurements of d_n and p_n. As for p_i it is chosen to work with the two pressure sensors located at the nodes v_3 and v_5, which gives the pressure measurements p_3 and p_5.

The SWIL was configured to run continuously over a period of 14 hours with the consumer demand curves described in **Sec. 4.3.1** while logging the sensor values. Since the model is to be used for residual generation to indicate when a leakage occurs it is important that the data is collected from the system without any leakages. The data is then split into two sets, one for estimation and one for validation. The estimation data set is then transformed using **Eq. 6.3** and a line is fitted using the least squared method. This results in two A coefficients and two B coefficients which gives the two corresponding sets of α and β coefficients shown in **Tab. 6.1**.

Node	α	β
3	$79.77 \cdot 10^{-3}$	$-9.05 \cdot 10^{-3}$
5	$58.37 \cdot 10^{-3}$	$89.08 \cdot 10^{-3}$

Table 6.1: Coefficients for the reduced model for the lab setup

Inserting the coefficient from the table into **Eq. 6.1** yields the reduced models for the pressures in node 3 and 5 presented in **Eq. 6.5**, where the $\hat{p}_i(t)$ notation is introduced to distinguish between the measured pressure values, p_i, and the estimated pressure values from the reduced model.

$$\hat{p}_3(t) = -79.77 \cdot 10^{-3} \cdot d_n(t)^2 + 9.19 \cdot 10^{-3} + p_n(t) \tag{6.5a}$$

$$\hat{p}_5(t) = -58.37 \cdot 10^{-3} \cdot d_n(t)^2 - 89.27 \cdot 10^{-3} + p_n(t) \tag{6.5b}$$

The validation data is presented in the two figures below where it is compared with the output of the reduced model evaluated on this data. In **Fig. 6.2** the two pressures, p_3 and p_5, is plotted against the squared demand which shows the linear relationship between them which is captured by the two regression lines shown in the same figure. In **Fig. 6.3** a time series of data from the two pressure sensors, p_3 and p_5, together with the predicted pressures from the reduced model, $\hat{p}_3$ and $\hat{p}_5$ is plotted. Here it is seen that the reduced models manages to capture the pressure behaviour of the real system with only the reference pressure and demand as inputs.

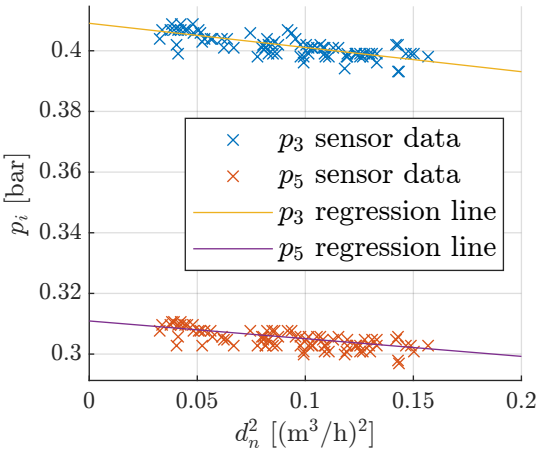


Figure 6.2: The squared reference demand plotted against the pressure. (SWIL)

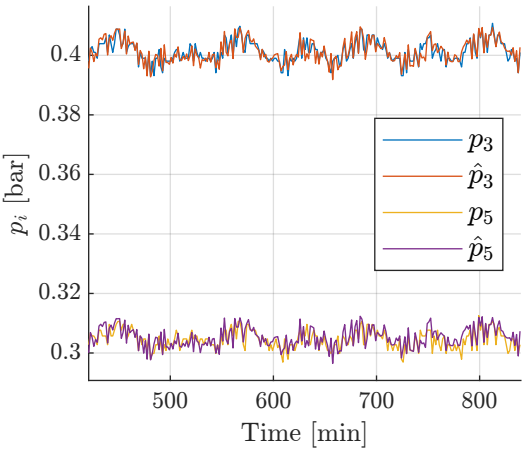


Figure 6.3: Time series of the pressure and the reduced model. (SWIL)

6.1.3 Results from simulation

Since the leakage detection algorithm is to be tested first in simulation, with the SWIL being simulated by the full model described in **Sec 3.4**, a set of α and β coefficients is also needed for this. To do this the simulated model is configured to run with the same controllers as implemented on the real system and set to run for the same amount of time. Since a significant amount of noise has been observed on the measurements from the sensors on the SWIL, noise with the same variance has been added to the simulation outputs to make the full order model fit the real system as closely as possible.

With the simulation data obtained the α and β coefficients for the simulation model is obtained using the same methods used to obtain the coefficients for the real system. The resulting coefficients for the simulated model is presented in **Tab. 6.2**.

Node	α	β
3	$170.14 \cdot 10^{-3}$	$-10.58 \cdot 10^{-3}$
5	$210.19 \cdot 10^{-3}$	$88.41 \cdot 10^{-3}$

Table 6.2: Coefficients for the reduced model for the simulation model

Using the coefficients in the table, the model becomes the one seen in **Eq. 6.6**.

$$\hat{p}_3(t) = -170.14 \cdot 10^{-3} \cdot d_n(t)^2 + 10.58 \cdot 10^{-3} + p_n(t) \tag{6.6a}$$

$$\hat{p}_5(t) = -210.19 \cdot 10^{-3} \cdot d_n(t)^2 - 88.41 \cdot 10^{-3} + p_n(t) \tag{6.6b}$$

As was done with the reduced model for the real system, the reduced model for the simulation model is evaluated on a validation data set which can be seen in **Fig. 6.4** and **Fig. 6.5**. This shows the that the simulation model exhibits the same linear relationship between d_n^2 and p_i and that the reduced model is capturing this behaviour.

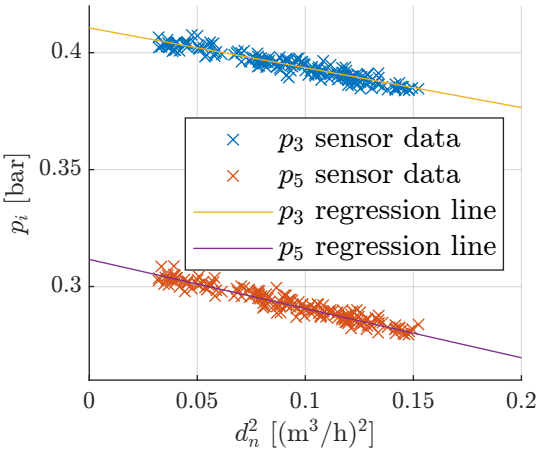


Figure 6.4: The squared reference demand plotted against the pressure. (Simulation)

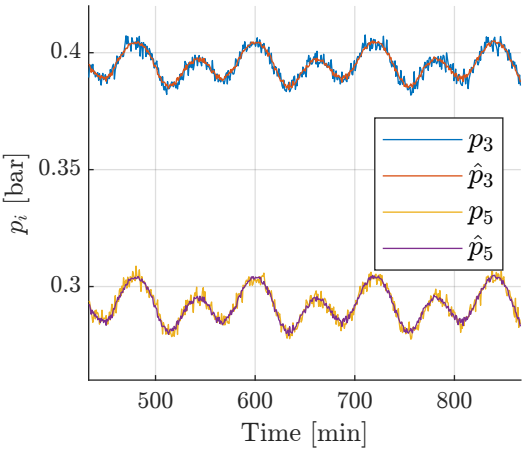


Figure 6.5: Time series of the pressures from the full order model and the reduced model. (Simulation)

6.1.4 Considerations on implementation on real WDN

The purpose of this model is ultimately to be used for residual generation in the process of leakage detection and localization which means that the model must follow the nominal system without leakages. This does, however, pose a problem since a leakage might not be the only change that can happen to a water network. Over time, the distribution between the nodal demands in the network might change when the housing in the area changes, e.g. new houses being added. By changing that distribution **Asm. 3.1** is violated and the model is no longer valid, which will cause it to deviate from the real system. This could then be detected as a leakage by the leakage detection algorithm and cause a false alarm.

A method for avoiding this, is to continuously update the α and β parameters so the reduced model can keep up with the changes in the network not caused by a leakage. To implement this it is important to consider the time frames on which a leak can change the system. If the model is updated too fast it might adapt to the system with a leakage before the leakage can be detected, thus making the residual signal useless.

A problem similar to this has been found to be present in the SWIL. If a new set of α and β coefficients are estimated after a restart of the system, these new coefficients are significantly different to the old set. This means that some parameters of the system are changing when the system undergoes a restart. Since it has been observed that air is getting trapped throughout the pipes this is suspected to be the cause. Letting the system stay turned on for a long period of time before a test is carried out can mitigate the problem but the system is still different enough to cause an offset on the residual when using α and β coefficients from an earlier test. To avoid this becoming a problem for the tests, it is chosen to estimate new coefficients for the SWIL before every leakage test.

6.2 Residual generation and investigation

The residual generation is an essential part of both leakage detection and localization. Therefore, this section will look upon its characteristics. The residual is calculated as seen in **Eq. 6.7**

$$r_i = p_i - \hat{p}_i + w \tag{6.7}$$

where r_i is the residual at vertex i, p_i is the pressure measurement at vertex i in SWIL, $\hat{p}_i$ is the estimated pressure from the reduced model at vertex i, and w is the combined noise from the estimated pressure $\hat{p}_i$ and p_i. As described in **Sec. 4.2.3** the chosen pressure sensors are placed at vertices 3 and 5 from the scenario shown in **Fig. 4.10**. Therefore, the two residuals r_3 and r_5 are expected. A generated residual (r_5) is shown **Fig. 6.6**. Next, the characteristics of the noise w is looked upon as to determine if it belongs to some distribution. In **Fig. 6.7** a normalized histogram of very sample in the time series of r_5 from **Fig. 6.6** is shown. The histogram is normalized by estimating the probability density function (pdf) using **Eq. 6.8** [MathWorks, n.d.],

$$v_i = \frac{c_i}{N \cdot W_i} \tag{6.8}$$

where v_i is the i^{th} bin value of the estimated pdf, c_i is the number of elements in the i^{th} bin, W_i is the width of the i^{th} bin, and N is the total number of samples of r_5. As seen

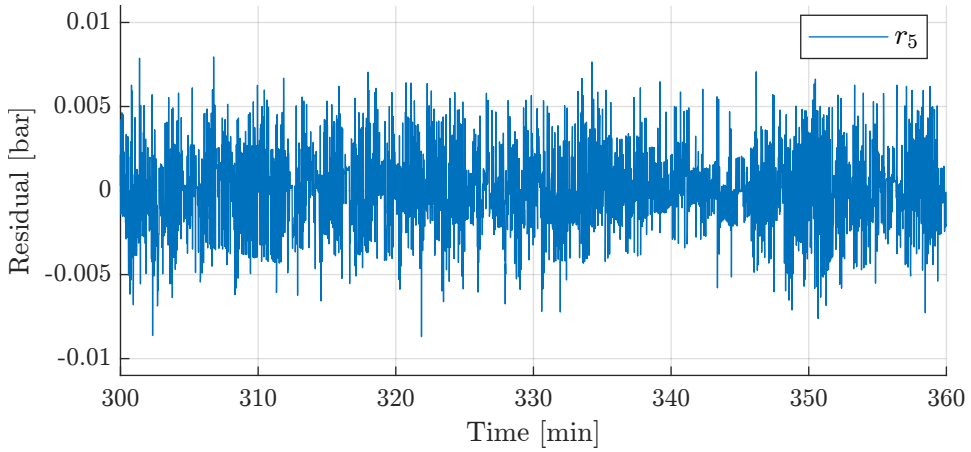


Figure 6.6: The residual generated for vertex 5 from data measured from the SWIL and the reduced model. The residual is a minor part of a large data set, which does not contain a leakage.

by **Fig. 6.7**, a pdf of a normal distribution is realized with the mean and variance of the time series of r_5.

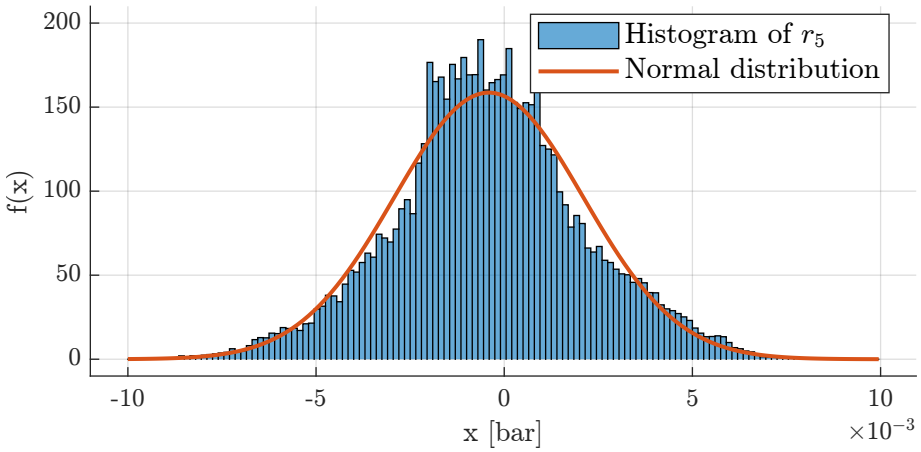


Figure 6.7: Histogram of residual r_5 shown against the normal distribution, which is drawn with a mean and variance calculated from the whole sequence of r_5 without a leakage.

This shows that the noise of the residual could be approximated as being normally distributed. From now on the residual noise w is assumed to be normally distributed, which will be taken advantage of in the leakage detection algorithm described in **Sec. 6.3**.

6.3 Choice of detection method

From the previous sections, the general leakage detection scheme is seen in **Fig. 6.1** and the residual r_5 under nominal operation of the WDN in the SWIL is shown in **Fig. 6.6**. Now it is investigated what can be expected of the residuals in a simulation without signal noise if the system is exposed to leakage. This can be seen by the two residuals r_3 and r_5 in **Fig. 6.8** and **Fig. 6.9**. Here, it is clear that a leakage certainly affects the residuals, since it changes its behavior as can be seen by the change in mean. The aim of this section is then to examine which methods can be used to make a clear distinction between the residual during nominal operation and with a leakage. Being able to do so allows for a simple threshold, which can be used to decide when a leakage has happened.

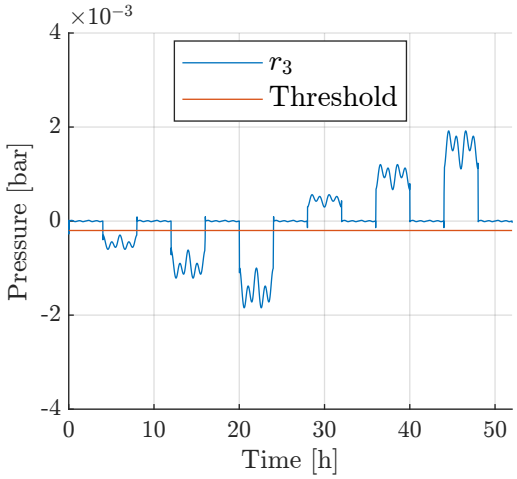


Figure 6.8: The simulated residual r_3 without noise showing clear signs that the WDN is not operating nominally as the WDN is exposed to leakages

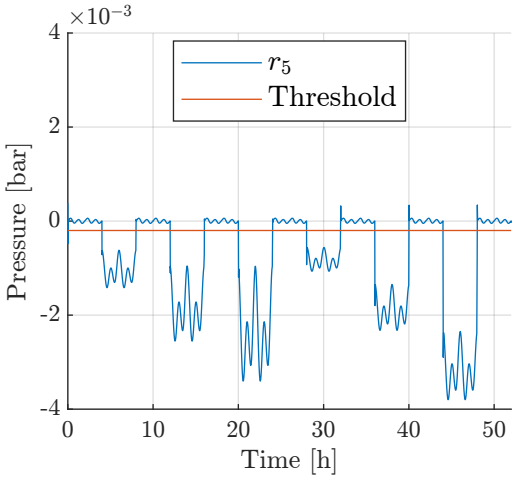


Figure 6.9: The simulated residual r_5 without noise showing clear signs that the WDN is not operating nominally as the WDN is exposed to leakages

The residuals r_3 and r_5 from the simulation are shown in **Fig. 6.8** and **Fig. 6.9**, from the same time series, respectively. At specific time instances, leakages are introduced as seen by **Fig. 6.10**. Here, the first three leakages of different amplitude are located at vertex v_1 as noted by the leakage demand $d_{l,1}$. In the same manner, the latter three leakages $d_{l,6}$ are located at vertex v_6. These leakage locations coincide with the chosen leakage locations as described in **Sec. 4.2.3**. From **Fig. 6.8** and **Fig. 6.9** it is clear that the leakages affect the behaviour of the residuals as an increase in the leakage size has an increased influence. For both residuals, the leakages are detectable by placing a threshold.

The simulated residuals in **Fig. 6.8** and **Fig. 6.9** does not incorporate measurement noise, which becomes apparent in a real world WDN. The effect of measurement noise is clearly seen in **Fig. 6.11** and **Fig. 6.12**. Here, the shaded grey areas are indicating when a leakage is active. These correspond exactly to the time series of leakages shown in **Fig. 6.10** with equivalent magnitudes. By this, it is difficult to distinguish the nominal residual from the event of a leakage. Therefore, a threshold can not be chosen for the residuals themselves, such that a leakage can be detected.

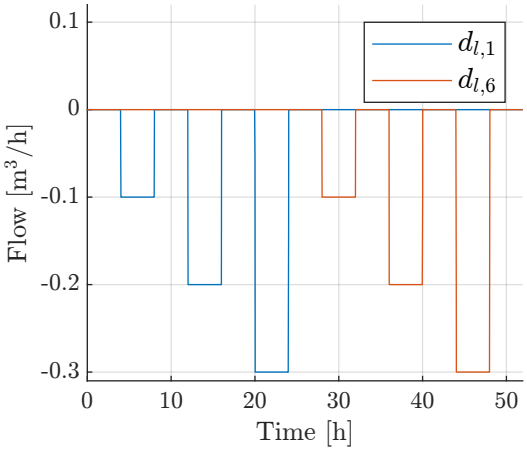


Figure 6.10: Time series of demand leakages $d_{l,1}$ at vertex v_1 and $d_{l,6}$ at vertex v_6, respectively. The leakages change in size and are active for a period of 4 hours in simulation

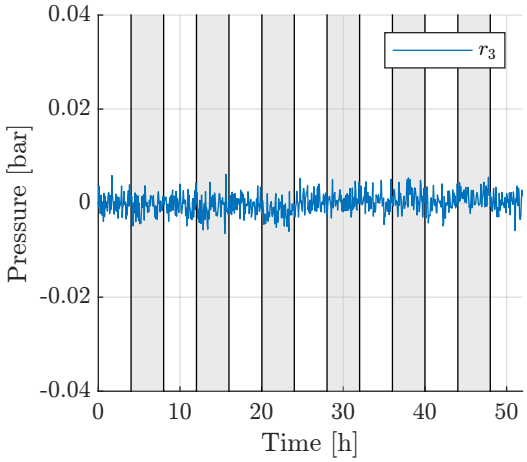


Figure 6.11: Simulated residual r_3 while the WDN is being exposed to the leakage time series from **Fig. 6.10**

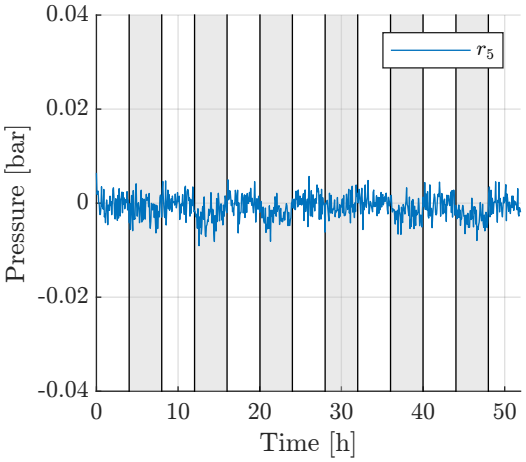


Figure 6.12: Simulated residual r_5 while the WDN is being exposed to the leakage time series from **Fig. 6.10**

Based on this, the residual itself is not appropriate for choosing a threshold when measurement noise is included in the simulation. Therefore, other approaches to detect changes in the mean is looked further into in the next sections.

6.3.1 Cumulative Sum (CUSUM)

CUSUM is an algorithm designed to detect a change in the distribution that inputted data arrives from. This can e.g. mean a change in the mean and/or a change in the variance. As described in **Sec. 6.2** the residuals experience a change in the mean, which is desired to be detected.

Considering a sequence of an independent and identically distributed (IID) random variable $z(1), z(2), \ldots, z(k)$ with probability density function (pdf) $p_\theta(z)$. Here, θ is the scalar parameter that might have changed (a change in the mean in this case), then at time k a

choice is made between the following two hypotheses [Blanke et al., 2015]:

$$\begin{aligned} &\mathcal{H}_0 : \theta = \theta_0 \text{ for } 1 \leq i \leq k \\ &\mathcal{H}_1 : \theta = \theta_0 \text{ for } 1 \leq i \leq k_0 - 1 \text{ and } \theta = \theta_1 \text{ for } k_0 \leq i \leq k \end{aligned} \tag{6.9}$$

where k_0 is the time of the parameter change and θ_1 is denoted as the changed parameter, in this case it changes to a known mean. Further k_0 is unknown [Blanke et al., 2015]. Here, $\mathcal{H}_0$ is the hypothesis that the mean is unchanged and $\mathcal{H}_1$ being the hypothesis of a changed mean. This changed mean is, as described in **Sec. 6.2** caused by a leakage in the WDN.

In order to distinguish between between the nominal and leaky WDN, it is convenient to look at the log-likelihood ratio between $\mathcal{H}_0$ and $\mathcal{H}_1$. This corresponds to taking the ratio between the two known pdf's. This is seen as a part of the decision function $\phi(k)$ in **Eq. 6.10** [Blanke et al., 2015]. Looking at the log-likelihood inside the sum, it is seen that if it is most likely that $z(i)$ belong to the distribution p_{θ_0}, then $s(z(i))$ is smaller than 0. On the other hand if it is most likely that $z(i)$ belong to the distribution p_{θ_0}, then $s(z(i))$ is greater than 0. A thorough derivation of CUSUM is made in **App. A.8**.

$$\phi(k) = \max_{1 \leq j \leq k} S_j^k = \max_{1 \leq j \leq k} \sum_{i=j}^{k} \ln \left(\frac{p_{\theta_1}(z(i))}{p_{\theta_0}(z(i))} \right) \tag{6.10}$$

where j is a counting variable of the hypothetical time of parameter change. It is decided whether the data belongs to $\mathcal{H}_0$ or $\mathcal{H}_1$ by a threshold th, that is:

$$\begin{aligned} &\text{if } \phi(k) \leq th \text{ accept } \mathcal{H}_0 \\ &\text{if } \phi(k) > th \text{ accept } \mathcal{H}_1 \end{aligned} \tag{6.11}$$

6.3.2 Generalized likelihood ratio

As previously described in **Sec. 6.3.1**, CUSUM relies on the fact that the occurrence of a leakage changes the residual to some known mean. This is unfortunately not the case as leakages of different size affects the behaviour of the residual differently. The GLR approaches the problem of the unknown mean of the residual under influence of a leakage, by utilizing an estimate of changed mean μ_1. Furthermore, as described in **Eq. 6.2**, the residual noise is assumed to be Gaussian distributed and that only a change in the mean is happening. In turn, this ends up making the variance σ^2 common for the two distributions $p_{\theta_0} \sim \mathcal{N}(\mu_0, \sigma^2)$ and $p_{\theta_1} \sim \mathcal{N}(\mu_1, \sigma^2)$. Since the mean change μ_1 is unknown it is estimated by taking advantage of the Gaussian assumption, this estimate is denoted $\hat{\mu}_1$. This estimate is found by taking a starting point in the definition of the log-likelihood ratio S_j^k for arbitrary distributions. Next, the definition of the Gaussian distribution given in **Eq. A.25** is substituted into S_j^k for each distribution p_{θ_0} and p_{θ_1}. Hereby the maximum of this expression is found by taking the partial derivative with respect to μ_1, putting it equal to zero and isolating for μ_1. The result of this is the estimate $\hat{\mu}_1$, which is then further substituted into the decision function $\phi(k)$ in **Eq. 6.10**, and thus loosing the dependency upon μ_1 [Blanke et al., 2015]. A detailed derivation is seen in **App. A.8**. The resulting GLR decision function $\phi(k)$ is given in **Eq. 6.12** [Blanke et al., 2015]. By the definition of the GLR, it is designed to run offline, that is, on a finite set of samples

that has been sampled before applying the GLR. In a real scenario, measurements are sampled real time, which requires the GLR to be online. This is done by means of the sliding window M_w. The sliding window is a tuning parameter as it, based on the size, considers a small or large sequence of samples.

$$\phi(k) = \frac{1}{2\sigma^2} \max_{k-M_w+1 \leq j \leq k} \frac{1}{k-j+1} \left(\sum_{i=j}^{k} (z(i) - \mu_0) \right)^2 \tag{6.12}$$

As the algorithm is run, the samples contained in the sliding window are used to calculate the mean square for every j counting from $j = k - M_w + 1$ to $j = k$. As j increases, the number of samples considered in the mean square is reduced. Furthermore, an estimate of the most likely time at which the leakage occurred could be of interest. This estimated time of occurrence is denoted $\hat{k}_0$ and is given in **Eq. 6.13**[Blanke et al., 2015],

$$\hat{k}_0 = \arg \left\{ \frac{1}{2\sigma^2} \max_{k_a-M_w+1 \leq j \leq k_a} \frac{1}{k_a-j+1} \left(\sum_{i=j}^{k_a} (z(i) - \mu_0) \right)^2 \right\} \tag{6.13}$$

where k_a is the alarm time. This is the time where $\phi(k)$ exceeds the threshold th, i.e. a mean change is detected.

6.3.3 Window size and threshold level

As shown above, two parameters are needed to run the GLR algorithm, namely the length of the sliding window, M_w, and a value for the threshold, th.

In [Blanke et al., 2015] two methods are presented to find these parameters. In the first method M_w and th is calculated from requirements on the probability of correct detection and false alarms. This method is, however, based on the special case of the GLR where the estimated fault occurrence time, $\hat{k}_0$, is not needed. If $\hat{k}_0$ is needed the second method can be used. Here [Blanke et al., 2015] proposes an experimental method to tune the parameters using data collected from the real system or from simulation. Since the estimated fault occurrence time is needed for the localization algorithm the latter method is chosen.

To determine the size of M_w, the GLR algorithm is applied to the simulated residuals shown in **Fig 6.11** and **Fig 6.12** multiple times with difference sizes of M_w. Since these residuals comes from a simulation of the WDN with known leakage times the resulting outputs from the GLR can be evaluated on how well it separates the WDN under nominal conditions compared to when a leak is present. In **Fig 6.13a** and **Fig 6.13b** the results are shown for the residuals show in **Fig. 6.11** and **Fig. 6.12** respectively, both tested with a window spanning 1 and 2 hours. The variable M_h is introduced to represent the length of the windows in units of time such that $M_w = M_h/T_s$ where T_s is the sample time. This gives a more intuitive way of describing the window length compared to referring to it as a number of samples. T_s in this reports is 1 sec.

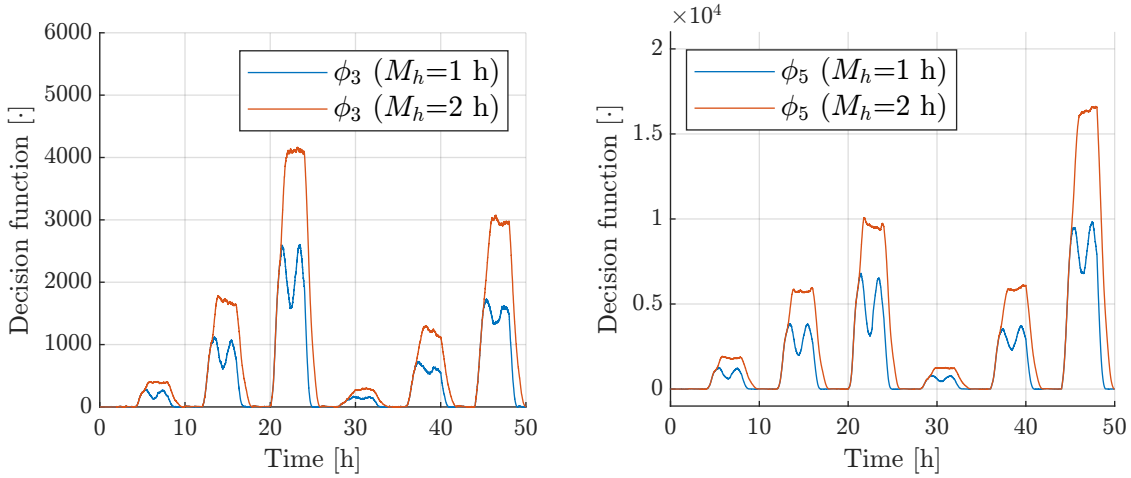


Figure 6.13: Decision function, ϕ, with a sliding window spanning 1 and 2 hours both evaluated on the residual from (a): node 3 seen in **Fig. 6.11** and (b): node 5 seen in **Fig. 6.12**.

The results here show that for both $M_h = 1$ and $M_h = 2$ hours the GLR algorithm gives an indication that a change in the mean value of the residuals has happened for all the leaks shown in **Fig. 6.10**. It is, however, also observed that the decision function, ϕ, for $M_h = 1$ hours varies more during a leak than ϕ for $M_h = 2$ hours. Further, there is a certain logic to using $M_h = 2$ (2 hours on the SWIL correspond to 24 hours in real life) since that exploits the periodicity of the demand. Should there be a small change in mean in the residual during high or low demand periods this hopefully is not detected as a leak due to the utilization of the periodicity. By these arguments, a sliding window spanning 2 hours is selected.

Next a value for the threshold, th, is selected which will be used to decide if the hypothesis that a mean change has occurred is true according to **Eq. 6.11**. Setting the threshold value too low will result in a higher probability of getting false positives. Increasing the threshold will result in a longer detection delay, that is, the time it takes from the leak occurs until the decision function crosses the threshold. A compromise therefore has to be made between these two scenarios. The way this is done for this project is to use measurements from a time period where the system is working under nominal conditions and observing the decision function ϕ. The threshold is then taken to be double the maximum value of ϕ doing this period.

6.4 Simulation results

Applying this threshold to the decision functions from **Fig. 6.13** and zooming in around the leaks happening at $t = 28$ h for ϕ_3 and $t = 44$ h for ϕ_5 produces the two plots shown in figure **Fig. 6.14** and **Fig. 6.15** respectively. When the decision function crosses the threshold the estimated time of the mean change, $\hat{t}$, is calculated using **Eq. 6.13**. Note here that the estimated time of the mean change is given as a sample number by **Eq.**

6.13, namely $\hat{k}_0$, but is converted to time by multiplying with the sample time, T_s, giving $\hat{t}_0$. Two vertical lines are used to indicate both the real leakage time, t_0, and the estimated leakage time, $\hat{t}_0$, on the two figures. Furthermore, some hysteresis is added to the threshold which causes the threshold value to decrease after decision function crosses it. This is done to avoid getting false negatives due to the decision function not being strictly increasing when a leak is present.

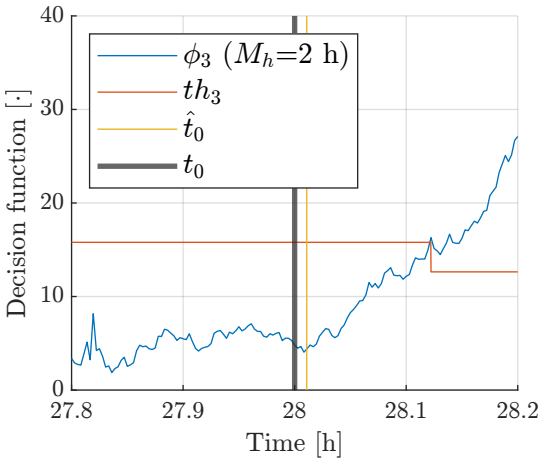


Figure 6.14: The decision function ϕ_3 zoomed in around the leak happening at $t = 28$ h with a leak size of 0.1 m^3/h.

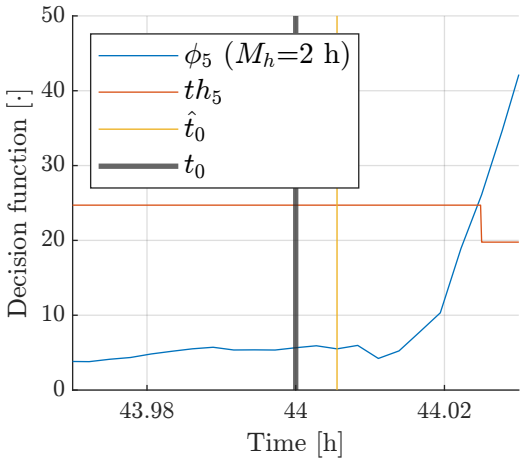


Figure 6.15: The decision function ϕ_5 zoomed in around the leak happening at $t = 44$ h with a leak size of 0.3 m^3/h.

From these two figures it is seen that the size of the leakage has big impact on the time it takes to detect that a leakage has occurred. Since these two leaks represents the best and the worst case scenario with respect to how fast a leak can be detected in the simulation, these results give an idea of what range the detection delays will fall into. For a large leak of 0.3 m^3/h the decision function, ϕ_5, detects the leak after 1.47 min. while the small leak of 0.1 m^3/h is caught by ϕ_3 after 7.27 min. Since the time in this simulation is scaled according to **Sec. 4.3.1**, 1 hours in this simulation corresponds to 12 hours on a real WDN. By scaling the detection delays accordingly, detection delays of 17.6 min and 1.45 hours is obtained for the large and the small leak respectively. These delays are summarized in **Tab. 6.3** together with the estimated leakage times, $\hat{t}_0$.

	Leak size	Detection delay	$\hat{t}_0$ error
ϕ_3	0.1 m^3/h	00:07:16 (01:27:14)	00:00:39 (00:07:47)
ϕ_5	0.3 m^3/h	00:01:28 (00:17:38)	00:00:20 (00:04:02)

Table 6.3: Detection delays and errors of the estimated leak occurrence time for two of the simulated leaks on node 6 seen in **Fig. 6.11** with time format: hh:mm:ss. The time in parentheses corresponds to time on a real life WDN.

6.5 SWIL results

To validate the results from the simulation a test with leaks on the same nodes is carried out on the SWIL. That is, leaks of three different sizes is introduced on node 1 and 6 and

the two residuals, r_3 and r_5 is generated for the two sensor nodes, v_3 and v_5 respectively. The target value for the size of the leaks is the same as used in the simulation, namely 0.1, 0.2 and 0.3 m^3/h. Since the leaks is introduced by controlling the opening degree of a valve in open loop, these leak sizes is not precisely replicated but comes close with 0.08, 0.17 and 0.28 m^3/h for node 1 and 0.06, 0.16 and 0.25 m^3/h for node 6.

This results in 12 different residuals, two for each of the leaks, and 12 corresponding GLR decision functions. The results for the small leaks and the large leaks are presented in **Fig. 6.16** to **Fig. 6.19** for the leaks on node 1 and the results for the leaks on node 6 are presented in **Fig. 6.20** to **Fig. 6.23**. The detection delays and the estimated leak occurrence times for leaks on these two nodes are presented in **Tab. 6.4** and **Tab. 6.5** respectively. The plots from the medium sized leaks, 0.17 m^3/h for node 1 and 0.16 m^3/h for node 6, can be found in **App. A.9**. Here plots showing the full time series of ϕ_3 and ϕ_5 for each of the leaks is presented.

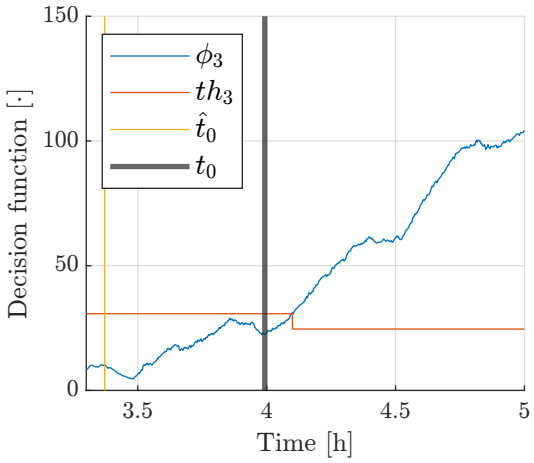


Figure 6.16: GLR algorithm used residual from node 3 with a leak at node 1: $d_{l1}(t) = -0.08$ m^3/h

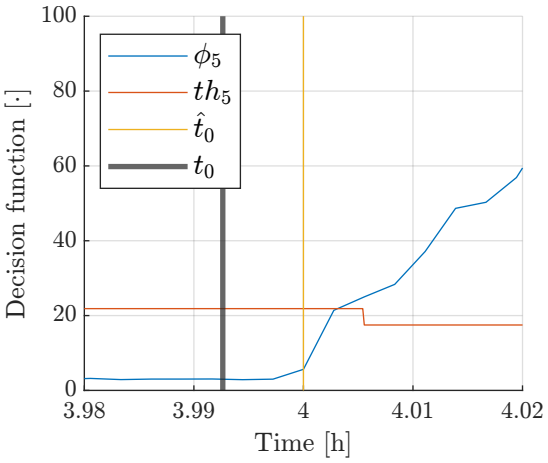


Figure 6.17: GLR algorithm used residual from node 5 with a leak at node 1: $d_{l1}(t) = -0.08$ m^3/h

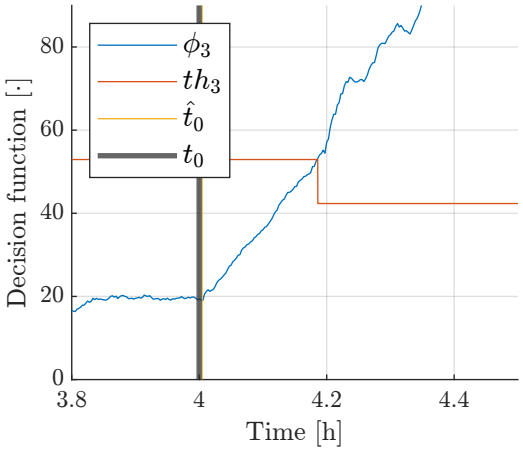


Figure 6.18: GLR algorithm used residual from node 3 with a leak at node 1: $d_{l1}(t) = -0.28$ m^3/h

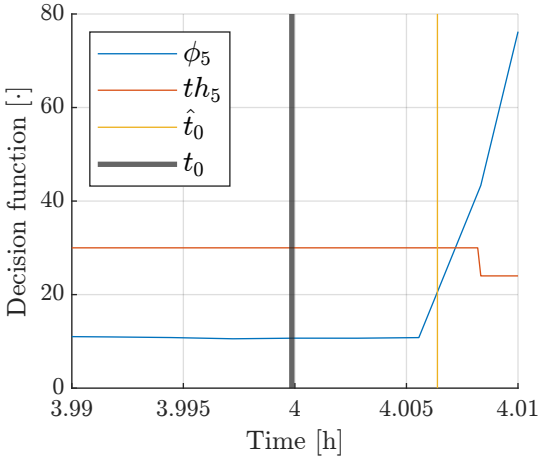


Figure 6.19: GLR algorithm used residual from node 5 with a leak at node 1: $d_{l1}(t) = -0.28$ m^3/h

	Leak size	Detection delay	$\hat{t}_0$ error
ϕ_3	0.08 m^3/h	00:06:26 (01:17:18)	-00:37:16 (-07:27:06)
ϕ_3	0.17 m^3/h	00:04:37 (00:55:18)	-00:38:23 (-07:40:30)
ϕ_3	0.28 m^3/h	00:11:10 (02:14:06)	00:00:17 (00:03:18)
ϕ_5	0.08 m^3/h	00:00:47 (00:09:18)	00:00:26 (00:05:18)
ϕ_5	0.17 m^3/h	00:00:37 (00:07:18)	00:00:23 (00:04:30)
ϕ_5	0.28 m^3/h	00:00:30 (00:06:06)	00:00:24 (00:04:42)

Table 6.4: Detection delays and errors of the estimated leak occurrence time on node 1 on the SWIL with time format: hh:mm:ss. The time in parentheses corresponds to time on a real life WDN.

The results shows that the GLR algorithm is able to detect all the leaks on node 1 and the detection delays seems to follow the same trend as seen in the simulation with smaller leaks giving larger delays. The errors of the estimated leak times, $\hat{t}_0$, also falls within the range on the ones observed in simulation with the exception of the ones based on ϕ_3. Here the leak time for the two smallest leaks, $d_{l1} = -0.08$ and $d_{l1} = -0.17$, is estimated to be before the actual leak occurs, resulting on negative errors. The cause of this negative estimate is made clear by looking at **Fig. 6.16** where it can be seen that the decision function is increasing before the leak occurs. A reason for this increase before the leak could be that the residual naturally has a small change in mean during high flow periods and the leakage is introduced in the middle of one of these periods. However, if the undershoot was caused by this it is expected that the undershoot should also have shown up in simulation, which has been found not to be the case. Another explanation is considered which relates to what is mentioned in **Sec. 6.1.4** about air getting trapped in the pipes. If this is the case then the pipe parameters will slowly change though out the whole test as the air is vented out which will show up as a mean change on the residual.

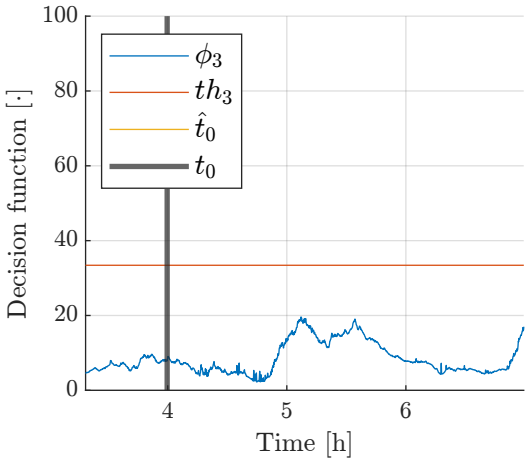


Figure 6.20: GLR algorithm used residual from node 3 with a leak at node 6: $d_{l6}(t) = -0.06 m^3/h$

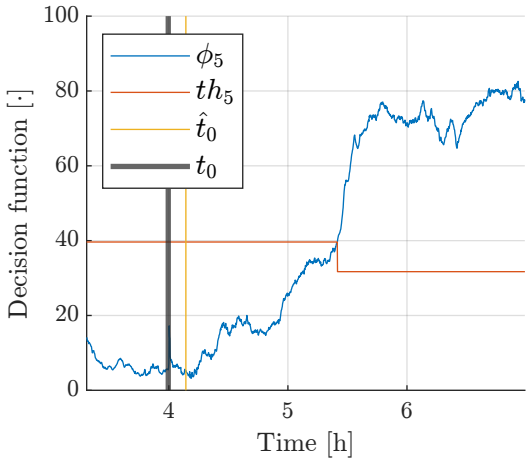


Figure 6.21: GLR algorithm used residual from node 5 with a leak at node 6: $d_{l6}(t) = -0.06 m^3/h$

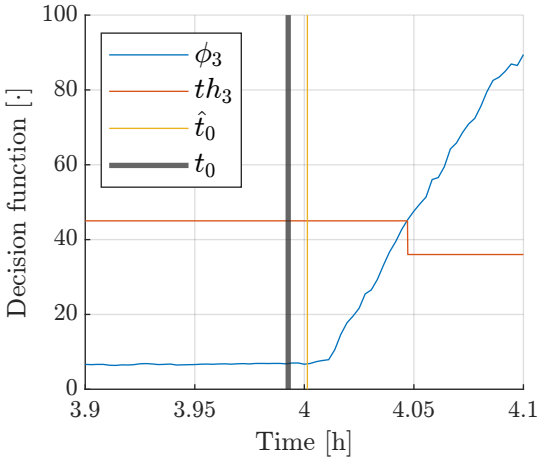


Figure 6.22: GLR algorithm used residual from node 3 with a leak at node 6: $d_{l6}(t) = -0.25\text{m}^3/\text{h}$

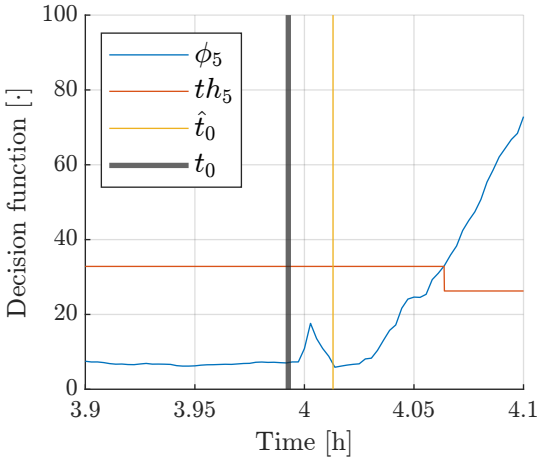


Figure 6.23: GLR algorithm used residual from node 5 with a leak at node 6: $d_{l6}(t) = -0.25\text{m}^3/\text{h}$

	Leak size	Detection delay	$\hat{t}_0$ error
ϕ_3	0.06 m^3/h	Not detected	Not detected
ϕ_3	0.16 m^3/h	00:15:37 (03:07:18)	00:01:18 (00:15:30)
ϕ_3	0.25 m^3/h	00:03:16 (00:39:18)	00:00:32 (00:06:18)
ϕ_5	0.06 m^3/h	01:25:18 (17:03:30)	00:08:52 (01:46:30)
ϕ_5	0.16 m^3/h	00:23:47 (04:45:18)	00:02:52 (00:34:18)
ϕ_5	0.25 m^3/h	00:04:16 (00:51:18)	00:01:13 (00:14:42)

Table 6.5: Detection delays and errors of the estimated leak occurrence time on node 6 on the SWIL with time format: hh:mm:ss. The time in parentheses corresponds to time on a real life WDN.

The results from the leaks on node 6 shows that the GLR algorithm able to detect all of these as well with the exception of the smallest leak, $d_{l6} = -0.06$ which is not detected by ϕ_3. This gives an indication of the minimum size of a leak before the GLR algorithm can detect it. It is also observed that for ϕ_5 the detection delays here is significant larger than the ones seen for the leaks on node 1. It is however to be expected that residuals is affected differently by leaks on different nodes.

6.5.1 Hypothesis selection

The GLR algorithm described above detects a mean change in a scalar. However, this report works with several scalars (r_3 and r_5) for which a mean change must be detected. Because of this, two GLR algorithms are set to run in parallel, one for each of the residuals, resulting in two decision functions, ϕ_3 and ϕ_5 and their corresponding thresholds, th_3 and th_5. Both of the decision functions has the possibility to detect the same leak, but they will not necessarily cross their thresholds at the same time or generate the same estimate of the leak occurrence time, t_0. This however, arises the problem of which one to chose.

Since a small detection delay is important for this application it makes sense to choose the

hypothesis and estimated leak occurrence time from whichever decision function crosses its threshold first. That is, $\mathcal{H}_1$ is accepted if ϕ_3 or ϕ_5 crosses their respective thresholds and $\mathcal{H}_0$ is accepted otherwise. This leads to the reformulation of **Eq. 6.11** seen in **Eq. 6.14**.

$$\begin{aligned} &\text{if } \phi_3(k) \geq th_3 \text{ OR } \phi_5(k) \geq th_5 \text{ accept } \mathcal{H}_1 \\ &\text{else accept } \mathcal{H}_0 \end{aligned} \tag{6.14}$$

Another way of solving this problem of deciding $\mathcal{H}$ based on two residual signals is to exploit the fact that information of the same leak is contained in both of the residuals. This combined information can be used by combining them into a single vector which makes it possible to use the vector case of the GLR as described in [Blanke et al., 2015]. This method is further discussed in **Sec. 8.3.1**.

In the next section, a method for localizing a leakage is introduced. This leakage localization method assumes that the leakage detection has taken place and the correct leak time has been identified. However, a combined test for leakage detection and leakage localization is carried out in **Sec. 8.1**.

Leakage localization 7

In **Sec. 3.5** a reduced-order/steady-state model for a Water Distributions Network (WDN) is derived. The model is proven valid for the case of multiple inlets if a special control strategy is applied. This control strategy involves one pump to be pressure controlled and another to be flow controlled. A controller of this type is developed in **Ch. 5**. In **Ch. 6** the properties of the residuals are investigated and how they are generated. These residuals experience a change in the mean if a leakage is present, which makes it possible to detect the leakage using GLR.

This chapter introduces a method for locating leakages in the WDN using the residuals. The method presented in this chapter is a novel contribution to the previous work by [Rathore et al., 2021], which is further inspired by a method derived by [Perez et al., 2014]. The method presented in [Rathore et al., 2021] are only suited for WDN's with a single inlet. Therefore, **Sec. 7.1** extends the previous work to allow for WDN's with multiple inlets.

This extended localization method is tested in simulation in **Sec. 7.2**. Lastly, the localization method is tested in the Smart Water Infrastructure Laboratory (SWIL) in **Sec. 7.3**

In this section, it is assumed that the correct leak time (k_0) is known. This is done to test the leakage localization isolated from the leakage detection. In Sec. **8.1** the leakage detection and the leakage localization is tested in combination. It is further assumed in this section that the GLR has taken 1000 samples to detect the leak. In turn, this means that leakage localization can use a mean value of the data (the residuals) of 1000 samples. Using a mean of the residual is an advantage since it dampens the disturbance from the noise.

7.1 Theory

The approach presented in this section is based on a method suggested by [Rathore et al., 2021]. However, the method presented in this section differs in several aspects since it is adapted to be applicable for WDN's with multiple inlets.

7.1.1 Definitions and restatement of important equations

The following section build upon the findings in the reduced-order model in the case of multiple inlets presented in **Sec. 3.5.2**. From this, the relations that are derived are to be used in the derivation of the localization method. As a summary, the model shows that

the vector of pressures $\bar{p}(t)$ under certain assumption can be found by a set of constants multiplied with the inlet flow at the reference node ($d_n(t)$) and adding the pressure at the reference node ($p_n(t)$). This is restated in **Eq. 7.1** with the coefficients described by **Eq. 7.2**.

$$\bar{p}(t) = -\alpha(a_{\mathcal{C}}, \mathrm{v})d_n(t)^2 - \beta + \mathbb{1}p_n(t) \tag{7.1}$$

$$\alpha(a_{\mathcal{C}}, \mathrm{v}) = -\bar{H}_{\mathcal{T}}^{-T}\lambda_{\mathcal{T}}(\gamma \cdot (-\bar{H}_{\mathcal{T}}^{-1}\bar{H}_{\mathcal{C}}a_{\mathcal{C}} - \bar{H}_{\mathcal{T}}^{-1}\mathrm{v})) \qquad \beta = \rho g(\bar{z} - \mathbb{1}z_n) \tag{7.2}$$

As an addition compared to **Sec. 3.5.2**, it is here made apparent that α is dependent on $a_{\mathcal{C}}$ and v. This is useful in the coming derivations since it turns out that these parameters change when a leak is present. Here, $a_{\mathcal{C}}$ is a mapping from the inflow $d_n(t)$ to chord flows $q_{\mathcal{C}}(t)$ and v is the demand distribution vector given as $\mathrm{v} = [\acute{\mathrm{v}}^T \quad \grave{\mathrm{v}}^T]^T$, where $\acute{\mathrm{v}} \in \mathbb{R}^{n-c}$, $\grave{\mathrm{v}} \in \mathbb{R}^{c-1}$, n is the number of vertices and c is the number of inlets. Furthermore, as described in the multiple inlet model in **Sec. 3.5.2**, an expression for the pressure drops across the edges in the WDN under the influence of a leakage is restated in **Eq. 7.3**

$$\left(\lambda_{\mathcal{C}}\left(\gamma a_{\mathcal{C}}\right) - \bar{H}_{\mathcal{C}}^{T}\bar{H}_{\mathcal{T}}^{-T}\lambda_{\mathcal{T}}\left(\gamma \cdot (-\bar{H}_{\mathcal{T}}^{-1}\bar{H}_{\mathcal{C}}a_{\mathcal{C}} - \bar{H}_{\mathcal{T}}^{-1}\mathrm{v})\right)\right) d_n(t)^2 = g(a_{\mathcal{C}}, \mathrm{v})d_n(t)^2 = 0 \tag{7.3}$$

Lastly, the multiple inlet model introduces the notion of introducing a leakage into the reduce-order model. This is done by distinguishing between the WDN during nominal operation and and when a leakage is present. This definition is shown for $\bar{p}(t)$ in **Eq. 7.4**.

$$\bar{p}(t) = \breve{\bar{p}}(t) + \delta\bar{p}(t) \tag{7.4}$$

Clearly, the part of interest in **Eq. 7.4** is $\delta\bar{p}(t)$ since this describes the change in pressure due to a leakage. This term is looked further upon in the next section.

7.1.2 Pressure variation caused by a leakage

This section derives an approximation for the change in pressure due to a leakage, namely $\delta\bar{p}(t)$. An approximated expression for $\delta\bar{p}(t)$ is found by making a first order Taylor expansion of **Eq. 7.1** with respect to changes in the parameters $a_{\mathcal{C}}$ and v. The Taylor expansion is shown in **Eq. 7.5** and **Eq. 7.6**.

$$\bar{p}(t) \approx \bar{p}(t)|_{\breve{a}_{\mathcal{C}},\breve{\mathrm{v}}} + \left.\frac{\partial\bar{p}(t)}{\partial \mathrm{v}}\right|_{\breve{a}_{\mathcal{C}},\breve{\mathrm{v}}} (\mathrm{v} - \breve{\mathrm{v}}) + \left.\frac{\partial\bar{p}(t)}{\partial a_{\mathcal{C}}}\right|_{\breve{a}_{\mathcal{C}},\breve{\mathrm{v}}} (a_{\mathcal{C}} - \breve{a}_{\mathcal{C}}) \tag{7.5}$$

$$\bar{p}(t) \approx -\alpha(\breve{a}_{\mathcal{C}}, \breve{\mathrm{v}})d_n(t)^2 - \beta + \mathbb{1}p_n(t) - \left.\frac{\partial\alpha(a_{\mathcal{C}}, \mathrm{v})}{\partial \mathrm{v}}\right|_{\breve{a}_{\mathcal{C}},\breve{\mathrm{v}}} \delta\mathrm{v}d_n(t)^2 - \left.\frac{\partial\alpha(a_{\mathcal{C}}, \mathrm{v})}{\partial a_{\mathcal{C}}}\right|_{\breve{a}_{\mathcal{C}},\breve{\mathrm{v}}} \delta a_{\mathcal{C}}d_n(t)^2 \tag{7.6}$$

The nominal model for pressure is given by $\breve{\bar{p}}(t) = -\alpha(\breve{a}_{\mathcal{C}}, \breve{\mathrm{v}})d_n(t)^2 - \beta + \mathbb{1}p_n(t)$, which is obtained by inserting the nominal values of $a_{\mathcal{C}}$ and v in **Eq. 7.2**. Here, $\breve{a}_{\mathcal{C}}$ is found numerically as described in **Sec. 3.5.2**. This expression clearly correspond to the first part in **Eq. 7.6**. Subtracting $\breve{\bar{p}}(t)$ from the expression in **Eq. 7.6** gives an approximate expression for $\delta\bar{p}(t)$, this expression is shown in **Eq. 7.7**. This equation includes the

change in distribution of the chord flow mapping $\delta a_{\mathcal{C}}$ and the change in the distribution of demands $\delta \mathrm{v}$. Expressions for these are derived in the next section.

$$\delta\bar{p}(t) = \bar{p}(t) - \breve{\bar{p}}(t) \approx -\left(\left.\frac{\partial \alpha(a_{\mathcal{C}}, \mathrm{v})}{\partial \mathrm{v}}\right|_{\breve{a}_{\mathcal{C}}, \breve{\mathrm{v}}} \delta \mathrm{v} + \left.\frac{\partial \alpha(a_{\mathcal{C}}, \mathrm{v})}{\partial a_{\mathcal{C}}}\right|_{\breve{a}_{\mathcal{C}}, \breve{\mathrm{v}}} \delta a_{\mathcal{C}} \right) d_n(t)^2 \tag{7.7}$$

7.1.3 Variation of $a_{\mathcal{C}}$ and v in case of a leakage

Starting by finding an expression for $\delta a_{\mathcal{C}}$ by looking into the property defined in **Eq. 3.87**. Making a taylor expansion of this equation with respect to a change in $a_{\mathcal{C}}$ and v yields the result in **Eq. 7.8** [Rathore et al., 2021].

$$0 \approx g(a_{\mathcal{C}}, \mathrm{v})|_{\breve{a}_{\mathcal{C}}, \breve{\mathrm{v}}}\, d_n(t)^2 + \left.\frac{\partial g(a_{\mathcal{C}}, \mathrm{v})}{\partial a_{\mathcal{C}}}\right|_{\breve{a}_{\mathcal{C}}, \breve{\mathrm{v}}} \delta a_{\mathcal{C}} d_n(t)^2 + \left.\frac{\partial g(a_{\mathcal{C}}, \mathrm{v})}{\partial \mathrm{v}}\right|_{\breve{a}_{\mathcal{C}}, \breve{\mathrm{v}}} \delta \mathrm{v} d_n(t)^2 \tag{7.8}$$

In **Eq. 7.8**, the first part clearly correspond to the original equation ($0 = g(a_{\mathcal{C}}, \mathrm{v}) d_n(t)^2$) in the nominal case by inserting $a_{\mathcal{C}} = \breve{a}_{\mathcal{C}}$ and $\mathrm{v} = \breve{\mathrm{v}}$. Therefore it must be equal to 0, that is $g(a_{\mathcal{C}}, \mathrm{v})|_{\breve{a}_{\mathcal{C}}, \breve{\mathrm{v}}}\, d_n(t)^2 = 0$. Substituting this into the first term in **Eq. 7.8**, dividing by $d_n(t)^2$ and isolating for $\delta a_{\mathcal{C}}$ gives the expression shown in **Eq. 7.9**. Here it is assumed that $\left.\frac{\partial g(a_{\mathcal{C}}, \mathrm{v})}{\partial a_{\mathcal{C}}}\right|_{\breve{a}_{\mathcal{C}}, \breve{\mathrm{v}}}$ is invertible [Rathore et al., 2021].

$$\delta a_{\mathcal{C}} \approx -\left(\left.\frac{\partial g(a_{\mathcal{C}}, \mathrm{v})}{\partial a_{\mathcal{C}}}\right|_{\breve{a}_{\mathcal{C}}, \breve{\mathrm{v}}} \right)^{-1} \left.\frac{\partial g(a_{\mathcal{C}}, \mathrm{v})}{\partial \mathrm{v}}\right|_{\breve{a}_{\mathcal{C}}, \breve{\mathrm{v}}} \delta \mathrm{v} \tag{7.9}$$

Inserting **Eq. 7.9** into **Eq. 7.7** and factorising gives the result shown in **Eq. 7.10**.

$$\delta\bar{p}(t) \approx -\underbrace{\left(\left.\frac{\partial \alpha(a_{\mathcal{C}}, \mathrm{v})}{\partial \mathrm{v}}\right|_{\breve{a}_{\mathcal{C}}, \breve{\mathrm{v}}} - \left.\frac{\partial \alpha(a_{\mathcal{C}}, \mathrm{v})}{\partial a_{\mathcal{C}}}\right|_{\breve{a}_{\mathcal{C}}, \breve{\mathrm{v}}} \left(\left.\frac{\partial g(a_{\mathcal{C}}, \mathrm{v})}{\partial a_{\mathcal{C}}}\right|_{\breve{a}_{\mathcal{C}}, \breve{\mathrm{v}}} \right)^{-1} \left.\frac{\partial g(a_{\mathcal{C}}, \mathrm{v})}{\partial \mathrm{v}}\right|_{\breve{a}_{\mathcal{C}}, \breve{\mathrm{v}}} \right)}_{S} \delta \mathrm{v} d_n(t)^2 \tag{7.10}$$

Next, evaluating all the differentials in S from **Eq. 7.10** gives the result for S shown in **Eq. 7.11**. In this equation $\partial\lambda_{\mathcal{C}} = \left.\frac{\partial \lambda(q_{\mathcal{C}})}{q_{\mathcal{C}}}\right|_{\gamma \cdot \breve{a}_{\mathcal{C}}}$ and $\partial\lambda_{\mathcal{T}} = \left.\frac{\partial \lambda(q_{\mathcal{T}})}{q_{\mathcal{T}}}\right|_{\gamma \cdot (-\bar{H}_{\mathcal{T}}^{-1} \bar{H}_{\mathcal{C}} \breve{a}_{\mathcal{C}} - \bar{H}_{\mathcal{T}}^{-1} \breve{\mathrm{v}})}$.

$$\begin{aligned} S = &\underbrace{\bar{H}_{\mathcal{T}}^{-T} \partial\lambda_{\mathcal{T}} \bar{H}_{\mathcal{T}}^{-1}}_{A^{-1}} - \underbrace{\bar{H}_{\mathcal{T}}^{-T} \partial\lambda_{\mathcal{T}} \bar{H}_{\mathcal{T}}^{-1}}_{A^{-1}} \underbrace{\bar{H}_{\mathcal{C}}}_{U} \\ &\left(\underbrace{\partial\lambda_{\mathcal{C}}}_{C^{-1}} + \underbrace{\bar{H}_{\mathcal{C}}^{T}}_{V} \underbrace{\bar{H}_{\mathcal{T}}^{-T} \partial\lambda_{\mathcal{T}} \bar{H}_{\mathcal{T}}^{-1}}_{A^{-1}} \underbrace{\bar{H}_{\mathcal{C}}}_{U} \right)^{-1} \underbrace{\bar{H}_{\mathcal{C}}^{T}}_{V} \underbrace{\bar{H}_{\mathcal{T}}^{-T} \partial\lambda_{\mathcal{T}} \bar{H}_{\mathcal{T}}^{-1}}_{A^{-1}} \end{aligned} \tag{7.11}$$

Each term in **Eq. 7.11** has a curly bracket and a specific name. This is to make it apparent that the Woodbury matrix identity can be used on this to simplify the expression. The Woodbury matrix identity states that: $(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$. Using the Woodbury matrix identity on **Eq. 7.11** yields the result for S shown in **Eq. 7.12**.

$$S = \left(\bar{H}_{\mathcal{T}} \left(\partial\lambda_{\mathcal{T}}\right)^{-1} \bar{H}_{\mathcal{T}}^{T} + \bar{H}_{\mathcal{C}} \left(\partial\lambda_{\mathcal{C}}\right)^{-1} \bar{H}_{\mathcal{C}}^{T} \right)^{-1} \tag{7.12}$$

Lastly, substituting S into **Eq. 7.10** gives the expression for δp shown in **Eq. 7.13**.

$$\delta p(t) = -S\delta \mathrm{v} d_n(t)^2 = -\left(\bar{H}_{\mathcal{T}} \left(\partial \lambda_{\mathcal{T}}\right)^{-1} \bar{H}_{\mathcal{T}}^T + \bar{H}_{\mathcal{C}} \left(\partial \lambda_{\mathcal{C}}\right)^{-1} \bar{H}_{\mathcal{C}}^T\right)^{-1} \delta \mathrm{v} d_n(t)^2 \tag{7.13}$$

Clearly, the missing part is now finding a relation between the change in the demand distribution $\delta \mathrm{v}$ and the change in demands $\delta d(t)$ in case of a leakage. Starting by inserting the definitions from **Eq. 3.81** and **Eq. 3.79** into the relation defined in **Eq. 3.89**. The result of this is shown in **Eq. 7.14**

$$\breve{\bar{d}}(t) + \delta \bar{d}(t) = -\gamma(\breve{\mathrm{v}} + \delta \mathrm{v}) d_n(t) \tag{7.14}$$

Inserting $\breve{\bar{d}}(t) = -\gamma \breve{\mathrm{v}} \breve{d_n}(t)$ into **Eq. 7.14** and further isolating and factoring a bit gives **Eq. 7.15**.

$$\gamma \breve{\mathrm{v}} \left(-\breve{d_n}(t) + d_n(t)\right) + \delta \bar{d}(t) = -\gamma \delta \mathrm{v} d_n(t) \tag{7.15}$$

It is known from the definition that $-\breve{d_n}(t) + d_n(t) = \delta d_n(t)$. Further, due to Kirchhoff, the relation $\delta d_n(t) = -\mathbb{1}^T \delta \bar{d}(t)$ must be true. Substituting these two relation into **Eq. 7.14** gives **Eq. 7.16**.

$$-\gamma \breve{\mathrm{v}} \mathbb{1}^T \delta \bar{d}(t) + \delta \bar{d}(t) = -\gamma \delta \mathrm{v} d_n(t) \tag{7.16}$$

Isolating this expression for $\delta \mathrm{v}$ gives the final expression in **Eq. 7.17**.

$$\delta \mathrm{v} = -\frac{1}{d_n(t)} (\frac{1}{\gamma} I - \breve{\mathrm{v}} \mathbb{1}^T) \delta \bar{d}(t) \tag{7.17}$$

Inserting this into **Eq. 7.13** gives the results in **Eq. 7.18** which clearly shows that in case of a leak (change in demand $\delta \bar{d}(t)$) it shows as a change in pressure ($\delta p(t)$).

$$\delta p(t) = S(\frac{1}{\gamma} I - \breve{\mathrm{v}} \mathbb{1}^T) \delta \bar{d}(t) d_n(t) \tag{7.18}$$

7.1.4 Estimated pressure residuals in case of leak at node l

In **Sec. 6.2** the concept of residuals where introduced. The idea behind the localization algorithm is to compare the residuals generated with a set of estimated residuals in case of a leakage at node l.

Eq. 7.18 describes how the pressure will change from the nominal pressure in case of a leak, meaning that $\delta p(t)$ is equal to the estimated vector of residuals, denoted as $\hat{r}(t)$. Using the assumption that a leakage only appears at a single node at the time, it is possible to generate the estimated residual if a leakage appears at the $l'th$ inlet node from **Eq. 7.18**. This is shown in **Eq. 7.19**. Here it is used that $\delta \bar{d}(t) = [e_l^T \quad \breve{\mathrm{v}}^T]^T d_{lm}(t)$ where $e_l \in \{0,1\}^{n-(c-1)}$ is a vector selecting the node where a leak is hypothesized to be present. Here, $\breve{\mathrm{v}} \in \mathbb{R}^{c-1}$ is the part of the nominal demand distribution vector $\breve{\mathrm{v}}$ related to the inlets (the reference node excluded), which is described in **Sec. 3.5.2** (here denoted as v) and $d_{lm}(t) < 0$ is the magnitude of the leak. Firstly, it is possible to write it this way since the leak only appears at a single node. Secondly, due to the way the references for

the inlet flows are made it is known that they will deliver a fixed percentage of the leak flow (expressed in $\breve{\mathrm{v}}$).

$$\hat{r}_l(t) = S(\frac{1}{\gamma}I - \breve{\mathrm{v}}\mathbb{1}^T) \begin{bmatrix} e_l \\ \breve{\mathrm{v}} \end{bmatrix} d_{lm}(t)d_n(t) = -S(\frac{1}{\gamma}I - \breve{v}\mathbb{1}^T) \begin{bmatrix} e_l \\ \breve{\mathrm{v}} \end{bmatrix} |d_{lm}(t)|\, d_n(t) \tag{7.19}$$

This vector of estimated residuals contains a residual for all nodes except for the reference node. This is not practical since this would require n pressure sensors in order to generate the real residuals. Therefore, a picking matrix (F_s) is introduced. This matrix picks out the estimated residuals for the nodes which has a pressure sensor connected to them. This leads to the $\hat{r}_l(t)$ instead being defined as in **Eq. 7.20**

$$\hat{r}_l(t) = \underbrace{-F_s S(\frac{1}{\gamma}I - \breve{\mathrm{v}}\mathbb{1}^T)}_{G} \begin{bmatrix} e_l \\ \breve{\mathrm{v}} \end{bmatrix} |d_{lm}(t)|\, d_n(t) \tag{7.20}$$

7.1.5 Comparison of estimated- and real residual

There exist several ways to compare how equal two vectors are ($r(t)$ and $\hat{r}_l(t)$). [Perez et al., 2014] suggest using the correlation function. [Rathore et al., 2021] suggest the "angle" between the two vectors. That is using the vector inner product and dividing by the length of each residual vector. The latter approach comes with the benefit that the leak magnitude ($d_{lm}(t)$) and the inflow at the reference node ($d_n(t)$) becomes irrelevant. The later approach is therefore chosen.

The definition of the decision signal $\psi(t)$ for a potential leak at node l is shown in **Eq. 7.21**

$$\psi_l(t) = \frac{\langle r(t), \hat{r}_l(t) \rangle}{|r(t)||\hat{r}_l(t)|} \tag{7.21}$$

Inserting the definition of $\hat{r}_l(t)$ into **Eq. 7.21** gives **Eq. 7.22**

$$\psi_l(t) = \frac{\left\langle r(t), G \begin{bmatrix} e_l \\ \breve{\mathrm{v}} \end{bmatrix} |d_{lm}(t)|\, d_n(t) \right\rangle}{|r(t)| \left| G \begin{bmatrix} e_l \\ \breve{\mathrm{v}} \end{bmatrix} |d_{lm}(t)|\, d_n(t) \right|} \tag{7.22}$$

Using that $\left| G[e_l^T \quad \breve{\mathrm{v}}^T]^T\, |d_{lm}(t)|\, d_n(t) \right| = \left| G[e_l^T \quad \breve{\mathrm{v}}^T]^T \right| |d_{lm}(t)|\, |d_n(t)|$ and that $\left\langle r(t), G[e_l^T \quad \breve{\mathrm{v}}^T]^T\, |d_{lm}(t)|\, d_n(t) \right\rangle = \left\langle r(t), G[e_l^T \quad \breve{\mathrm{v}}^T]^T \right\rangle |d_{lm}(t)|\, d_n(t)$ it is clearly seen that $|d_{lm}(t)|$ and $d_n(t)$ cancel each other out since they are both in nominator and the denominator in **Eq. 7.21**. This leads to a simplified expression for the decision signal ψ_l shown in **Eq. 7.23** [Rathore et al., 2021].

$$\psi_l(t) = \frac{\left\langle r(t), G \begin{bmatrix} e_l \\ \breve{\mathrm{v}} \end{bmatrix} \right\rangle}{|r(t)| \left| G \begin{bmatrix} e_l \\ \breve{\mathrm{v}} \end{bmatrix} \right|} \tag{7.23}$$

$\psi_l(t)$ is between -1 and 1, where a value close to 1 indicates that it is likely that the leak is at this node. The leak is most likely to be at the node that maximises **Eq. 7.23**. This is denoted as $\hat{l}(t)$, and is defined in **Eq. 7.24**.

$$\hat{l}(t) = \underset{l=1,\dots,n-c}{\operatorname{argmax}} \psi_l(t) \tag{7.24}$$

This approach is however a bit naive and will often not identify the correct node. Often a set of nodes will have a $\psi_l(t)$ close to each other. Since simplifications were used in the derivation of this leakage localization method there is a potential for that the leak is another node with high $\psi_l(t)$ when $\hat{l}(t)$. The set of nodes likely to hold the leak will often define a neighbourhood of nodes that should be checked for a leak. For convenience, $\psi_l(t)$ is restricted to take values between 0 and 1. If a node has value 0 there is little chance that the leak is at this node since it comes from a $r(t)$ which has the opposite direction to $\hat{r}(t)$. Thereby nodes with the value 0 can quickly be discarded. The restricted $\psi_l(t)$ is shown in **Eq. 7.25**.

$$\bar{\psi}_l(t) = \max\{\psi_l(t), 0\} \tag{7.25}$$

7.2 Simulation results

In the previous section, a method for localising a leakage in a multiple inlet Water Distribution Network (WDN) is derived. This work resulted in the vector $\bar{\psi}_l$ with values between 0 and 1 for each entry. Here the $i'th$ entry corresponds to the $i'th$ node. In $\bar{\psi}_l$, the higher the value at an entry, the more likely it is that the leak is at this node.

This section tests whether the proposed methods work by applying the network from **Sec. 4.2.1** a series of leaks in different magnitude in simulation. The data used for generating $\bar{\psi}_l$ are described in **Sec. 6.3** with the resulting residual being shown in **Fig. 6.11** and **Fig. 6.12**. The data for the plots/table are produced using the real leak time k_0 and taking a mean of the residuals from k_0 to $k_0 + 1000$ as mentioned in the introduction for this chapter.

The plots with the results for $\bar{\psi}$ are shown in **Fig. 7.1** to **Fig. 7.6**. The numerical values for $\bar{\psi}$ for the six tests are listed in **Tab 7.1**.

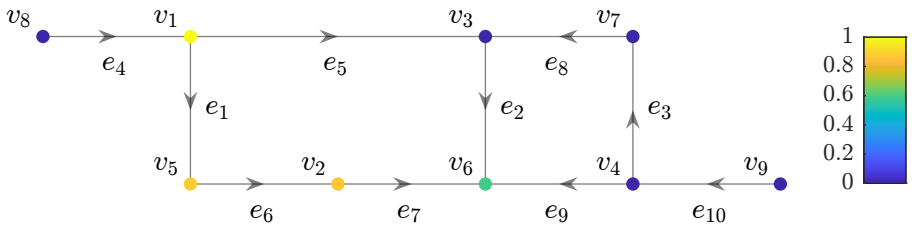


Figure 7.1: Simulated values for $\bar{\psi}(t)$ for a leak at node 1: $d_{l,1}(t) = -0.1\text{m}^3/\text{h}$

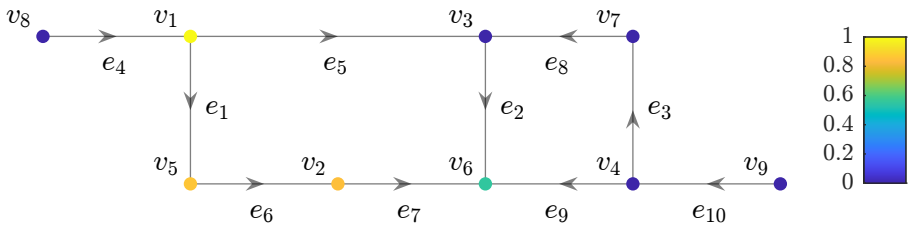


Figure 7.2: Simulated values for $\bar{\psi}(t)$ for a leak at node 1: $d_{l,1}(t) = -0.2\text{m}^3/\text{h}$

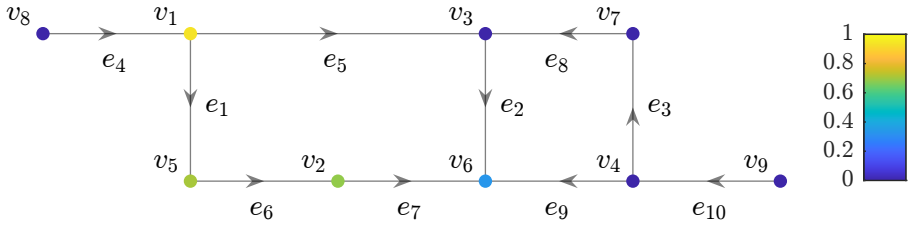


Figure 7.3: Simulated values for $\bar{\psi}(t)$ for a leak at node 1: $d_{l,1}(t) = -0.3\text{m}^3/\text{h}$

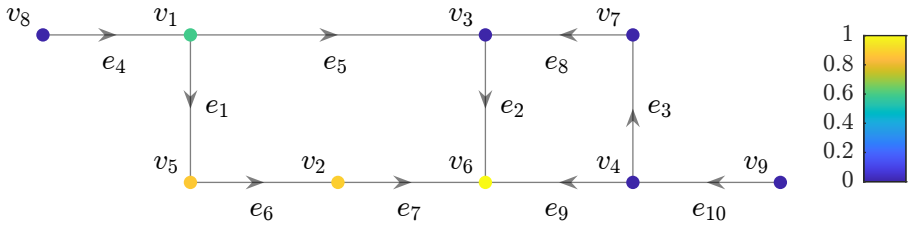


Figure 7.4: Simulated values for $\bar{\psi}(t)$ for a leak at node 6: $d_{l,6}(t) = -0.1\text{m}^3/\text{h}$

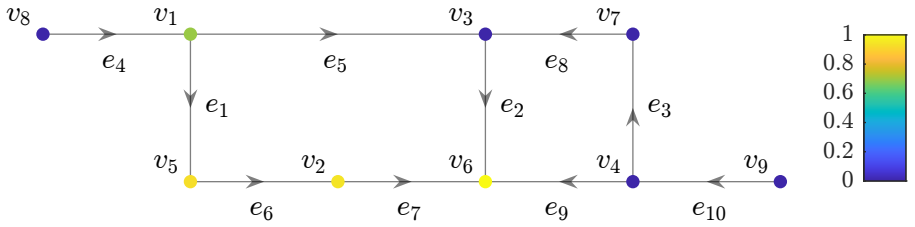


Figure 7.5: Simulated values for $\bar{\psi}(t)$ for a leak at node6: $d_{l,6}(t) = -0.2\text{m}^3/\text{h}$

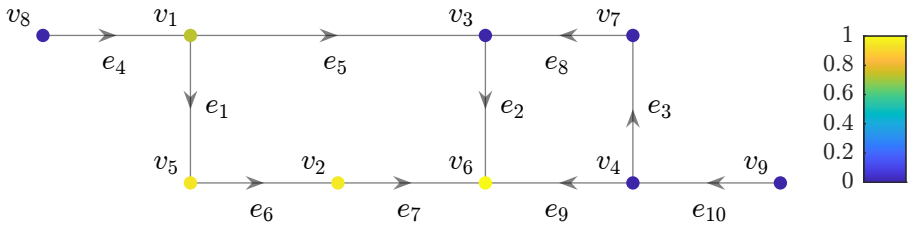


Figure 7.6: Simulated values for $\bar{\psi}(t)$ for a leak at node 6: $d_{l,6}(t) = -0.3\text{m}^3/\text{h}$

Leak node & size	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6	ψ_7
$d_{l,1} = -0.1$ m^3/h	**0.998**	0.867	-0.671	-0.995	0.883	0.591	-0.826
$d_{l,1} = -0.2$ m^3/h	**0.994**	0.846	-0.640	-0.990	0.863	0.558	-0.802
$d_{l,1} = -0.3$ m^3/h	**0.934**	0.685	-0.427	-0.922	0.709	0.332	-0.627
$d_{l,6} = -0.1$ m^3/h	0.590	0.883	-0.985	-0.616	0.868	**0.997**	-0.917
$d_{l,6} = -0.2$ m^3/h	0.691	0.938	-0.999	-0.715	0.926	**0.998**	-0.962
$d_{l,6} = -0.3$ m^3/h	0.731	0.955	-1.000	-0.752	0.945	**0.993**	-0.975
No leakage	-0.821	-0.489	0.196	**0.802**	-0.517	-0.094	0.419

Table 7.1: Simulated values of the ψ vector for the leak with size -0.1, -0.2 and -0.3 m^3/h at node 1 and 6. ψ_l for the estimated leak node ($\hat{l}$) is marked with bold letters. Further, the cells containing the three highest ψ_l's are marked with three shades of grey

As an addition to the tests in this section, in **App. A.10** a leak of the same magnitude is simulated for every node, one node at the time. This is for verifying that the algorithm to some degree identifies the correct leak node and doesn't identify the leaky node as being non-leaky ($\bar{\psi}_l = 0$). It is in this appendix shown that the algorithm detects 6/7 correctly giving the leaky node the highest ψ_l. In the last case the leaky node gets the second highest ψ_l.

From the simulations of the proposed leakage localization method, satisfying results are achieved. It identifies the correct node in 6 out of 7 simulated scenarios (**App. A.10**) and it is further able to handle some variation in magnitude of a leakage. An interesting thing observed during the simulation is that a set of nodes typically has a ψ_l fairly close to each other. In the simulations two main sets have appeared: v_2, v_5 and v_6 (and to some degree also v_1) have values that are close to each other in case of any leak. Further, the set v_3 and v_7 (and to some degree also v_4) does the same. From plots of the estimated residuals with leaks at node l from 1 to 7 ($\hat{r}_l$) it is identified that the direction of the leak is fairly similar in a set of nodes and opposite to each other from set to set.

The results from the simulations confirm that the proposed methods say something about the location of a leak. The next step is testing the proposed method on real-life data from the SWIL. This is described in the next section.

7.3 SWIL results

In this section, a similar test to the one described in the previous section is described. However, this time the data comes from the SWIL and not from a simulation. Further descriptions of the data used in this section are found in **Sec. 6.5** as the data used in this section is the same data that is used for the GLR tests.

As in the last section, it is assumed that the leak time k_0 is known and that the data used for the detection is a mean of the residual from k_0 to $k_0 + 1000$ samples. The results are shown in **Fig 7.7** to **Fig 7.12**. The numerical values for ψ_l for all the different leaks are shown in **Tab 7.2**.

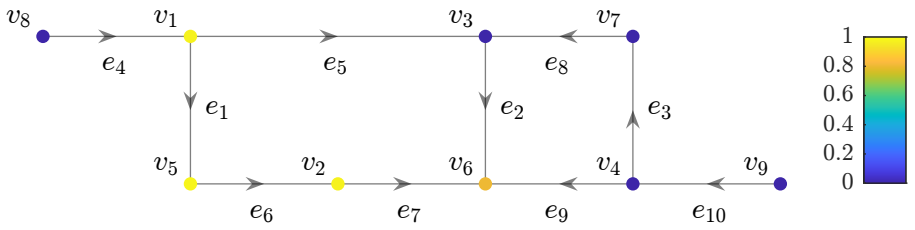


Figure 7.7: Plot of the values of the ψ vector obtained in SWIL for a leak at node 1 of size 0.08 m^3/h: $d_{l,1}(t) = -0.08$ m^3/h

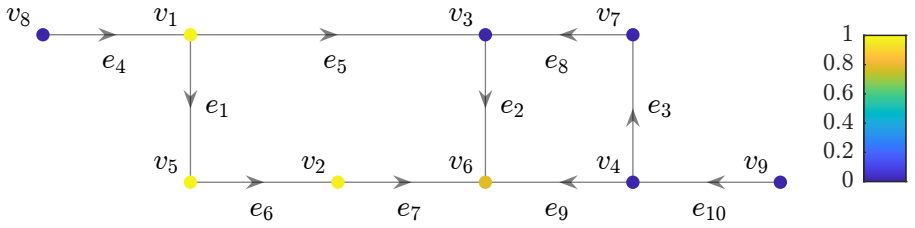


Figure 7.8: Plot of the values of the ψ vector obtained in SWIL for a leak at node 1 of size 0.17 m^3/h: $d_{l,1}(t) = -0.17$ m^3/h

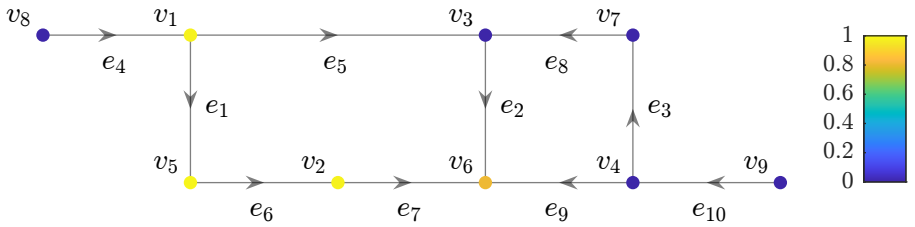


Figure 7.9: Plot of the values of the ψ vector obtained in SWIL for a leak at node 1 of size 0.28 m^3/h: $d_{l,1}(t) = -0.28$ m^3/h

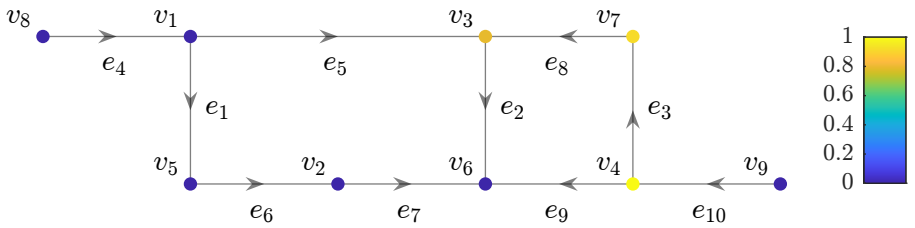


Figure 7.10: Plot of the values of the ψ vector obtained in SWIL for a leak at node 6 of size 0.06 m^3/h: $d_{l,6}(t) = -0.06$ m^3/h

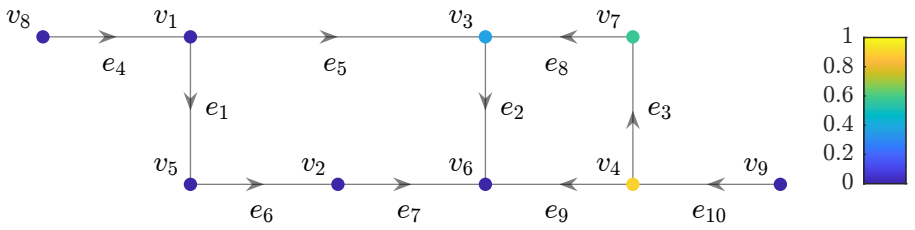


Figure 7.11: Plot of the values of the ψ vector obtained in SWIL for a leak at node 6 of size 0.16 m^3/h: $d_{l,6}(t) = -0.16$ m^3/h

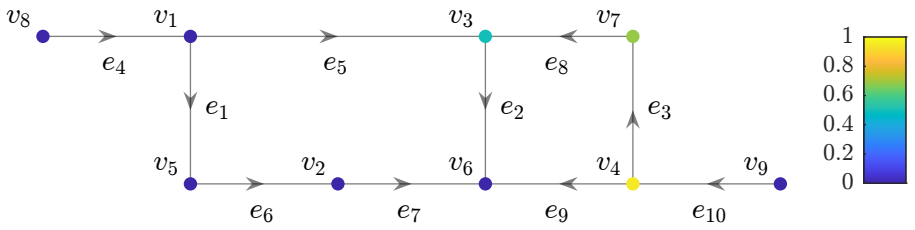


Figure 7.12: Plot of the values of the ψ vector obtained in SWIL for a leak at node 6 of size 0.25 m^3/h: $d_{l,6}(t) = -0.25$ m^3/h

Leak node & size	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6	ψ_7
$d_{l,1} = -0.08$ m^3/h	0.973	0.976	-0.862	-0.980	**0.983**	0.805	-0.956
$d_{l,1} = -0.17$ m^3/h	**0.980**	0.968	-0.844	-0.986	0.976	0.784	-0.946
$d_{l,1} = -0.28$ m^3/h	0.973	0.976	-0.862	-0.980	**0.983**	0.806	-0.957
$d_{l,6} = -0.06$ m^3/h	-0.994	-0.943	0.794	**0.997**	-0.953	-0.728	0.914
$d_{l,6} = -0.16$ m^3/h	-0.910	-0.638	0.369	**0.896**	-0.663	-0.272	0.576
$d_{l,6} = -0.25$ m^3/h	-0.961	-0.743	0.500	**0.951**	-0.764	-0.409	0.689
No leakage	-0.909	-1.000	0.944	0.922	-1.000	-0.905	**0.995**

Table 7.2: ψ_l obtained via test on the SWIL for leaks with three different sizes on nodes 1 and 6. ψ_l for the estimated leak node ($\hat{l}$) is marked with bold letters. Further, the cells containing the three highest ψ_l's are marked with three shades of grey

The results obtained on the SWIL has some of the tendencies as the result obtained in simulation. The same tendency of the nodes mainly being separated in sets are still present. However, for the leaks at node 6, there is a huge difference. The leaks at node 6 are not close to being correctly identified. The value for ψ at node 6 is only the 4th highest and is far lower than the ψ values for the three most likely nodes.

It is expected that the algorithms somewhat poor performance in real life comes from the poor estimates of the pipe resistance κ. Further, it is believed that the cause for this mainly affecting the leakage detection of node 6 is that this node is furthest from any inlet. That means that the flows have to go through a lot of pipes with badly estimated pipe resistance.

A second estimate for why node 6 gives poor results is also made. From simulation data, it is found that one of the residuals barely changes mean in case of a leak at this node. If there in the real residual is a small change in the mean, caused by noise or modelling error, the direction of the residual vector would significantly change. That means that when the real residual vector is compared with the hypothetical residual vector for a leak at node 6 it is not similar resulting in a low ψ.

Assessment 8

In this chapter, the project is assessed. That involves an acceptance test in **Sec. 8.1** where the leakage detection and the leakage localization is combined. Further, for verifying if the suggested leakage detection method is valid, it is tested on a simulation model of a large Water Distribution Network (WDN) in **Sec. 8.2**. Lastly, the methods and results of this report are evaluated in a discussion (**Sec. 8.3**) and a conclusion (**Sec. 8.4**).

8.1 Validation test: Combining detection and localisation

The two separate algorithms developed in **Ch. 6** for leakage detection and in **Ch. 7** for leakage localization is combined in this section with the aim of testing their performance as a single leakage detection and localization scheme. This is done as seen in **Fig. 2.4** where the alarm signal together with the estimated leak time, $\hat{t}_0$, from the detection block is taken as inputs to the localization block. Here a mean value of the residuals is calculated once the alarm is activated and is used to estimate which node has the leak. This mean value is estimated based on all the samples from the time span from $\hat{t}_0$ to the alarm time, t_a, since it will only contain samples from a time period where the leak is active.

The test is carried out on both the simulated data used in **Sec 6.4** and the data collected from the SWIL used in **Sec 6.5**. Using the same data means that the resulting detection delays and leak time estimates will be the same as when the GLR is tested on its own. The only difference here is that the hypothesis selection described in **Sec 6.5.1** is implemented so the results will be based on which decision function crosses its threshold first. Estimating the location of the leak node, $\hat{l}_0$, however, is now significantly different since the number of samples it is based on will be dependent on the detection delay and the leak time estimate from the GLR. If the leak is detected fast and/or the leak time is estimated poorly the number of residual samples available for localization is small. Another scenario that could affect the localization is if the leak time is estimated to lie before the real leak occurs. In that case, the localization result would be based on both residual samples with and without a leak. How using the leakage detection together with the leakage localization is tested in **Sec. 8.1.1** for the simulated data and in **Sec. 8.1.2** for the data collected from the SWIL.

8.1.1 Simulation

The results from the combined leakage detection and localization scheme tested on simulated data is presented in two tables below. The simulated system is subjected to different leakages in the same combination as in the localization test from **Sec. 7.2** with

three different leakage sizes on node one and six, while the rest is subjected to the same leakage size. The resulting detection delays and $\hat{t}_0$ errors for each of the leaks is listed in **Tab. 8.1** together with which decision function detected, the estimated leak node, $\hat{l}$ and the number of samples used to calculate $\hat{l}$. The values of ψ leading to the $\hat{l}$ is listed in **Tab. 8.2** for each of the corresponding leaks.

Leak node and size	Found by	Detection delay	$\hat{t}_0$ error	Nr. of samples	$\hat{l}$
$d_{l,1} = -0.1$ m^3/h	ϕ_5	00:04:20 (00:52:00)	00:00:58 (00:11:36)	202	1
$d_{l,1} = -0.2$ m^3/h	ϕ_5	00:03:46 (00:45:06)	00:00:34 (00:06:42)	192	1
$d_{l,1} = -0.3$ m^3/h	ϕ_5	00:03:13 (00:38:30)	-00:01:10 (-00:13:54)	262	1
$d_{l,2} = -0.1$ m^3/h	ϕ_5	00:01:00 (00:12:00)	00:00:05 (00:01:00)	55	2
$d_{l,3} = -0.1$ m^3/h	ϕ_5	00:00:56 (00:11:06)	00:00:26 (00:05:18)	29	3
$d_{l,4} = -0.1$ m^3/h	ϕ_5	00:00:53 (00:10:30)	00:00:16 (00:03:06)	37	4
$d_{l,5} = -0.1$ m^3/h	ϕ_5	00:00:59 (00:11:42)	00:00:14 (00:02:42)	45	2
$d_{l,6} = -0.1$ m^3/h	ϕ_3	00:08:28 (01:41:36)	00:03:18 (00:39:36)	310	6
$d_{l,6} = -0.2$ m^3/h	ϕ_5	00:02:25 (00:29:06)	-00:00:09 (-00:01:42)	154	6
$d_{l,6} = -0.3$ m^3/h	ϕ_5	00:01:32 (00:18:18)	00:00:25 (00:04:54)	67	5
$d_{l,7} = -0.1$ m^3/h	ϕ_5	00:00:56 (00:11:06)	00:00:17 (00:03:30)	38	3

Table 8.1: Obtained results for the leakage detection method from simulation. Listed is the detection time, the estimated leak time and which detection signal (ϕ) triggers the alarm. Times are listed in the format hh:mm:ss with the emulation time coming first and followed by real-life WDN time in parentheses. Lastly, the hypothesized leakage localization is shown ($\hat{l}$) with the number of samples in mean for leakage localization shown

Leak node and size	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6	ψ_7
$d_{l,1} = -0.1$ m^3/h	**1.000**	0.889	-0.705	-0.998	0.904	0.628	-0.851
$d_{l,1} = -0.2$ m^3/h	**0.990**	0.952	-0.811	-0.994	0.961	0.747	-0.925
$d_{l,1} = -0.3$ m^3/h	**0.982**	0.802	-0.579	-0.976	0.821	0.492	-0.753
$d_{l,2} = -0.1$ m^3/h	0.792	**0.979**	-0.994	-0.812	0.972	0.978	-0.992
$d_{l,3} = -0.1$ m^3/h	-0.715	-0.948	**1.000**	0.737	-0.938	-0.996	0.970
$d_{l,4} = -0.1$ m^3/h	-0.996	-0.937	0.783	**0.998**	-0.948	-0.715	0.907
$d_{l,5} = -0.1$ m^3/h	0.832	**0.991**	-0.984	-0.850	0.986	0.961	-0.998
$d_{l,6} = -0.1$ m^3/h	0.415	0.771	-0.930	-0.444	0.750	**0.963**	-0.818
$d_{l,6} = -0.2$ m^3/h	0.755	0.966	-0.999	-0.776	0.957	**0.988**	-0.983
$d_{l,6} = -0.3$ m^3/h	0.959	0.986	-0.887	-0.968	**0.991**	0.835	-0.971
$d_{l,7} = -0.1$ m^3/h	-0.695	-0.939	**0.999**	0.718	-0.928	-0.998	0.963

Table 8.2: Values of the ψ vector resulting from using a mean of the residual taken from $\hat{t}_0$ to t_a generated by the leakage detection. The bold values shown in this table are the ones which leads to the $\hat{l}$ shown in **Tab. 8.1**

The results in **Tab. 8.1** shows that the correct node is estimated correctly by $\hat{l}$ in 8/11 of the simulated leaks which is similar to the results from **Sec. 7.2** where 10/11 is located correctly. In the cases where the nodes are incorrectly identified, $\hat{l}$ still estimates that leaks are on a node close to the nodes with the real leaks. This shows that the localization

method is capable of estimating the correct leakage nodes or nodes close to the leaks, even for relatively small numbers of samples used for the mean estimation of the residual.

8.1.2 SWIL

As stated in the introduction to this section the SWIL combined test is run on the same data as used in **Sec. 6.5** and **Sec. 6.5**. This leads to the same detection times for the GLR. These times are repeated in **Tab. 8.3**, however, here only the time for the detection signal which exceeds the threshold the fastest is listed. In this table, the number of samples used in the mean of the residual is also listed. Lastly, the estimated leaky node from the leakage localization is shown.

Leak node and size	Found by	Detection delay	$\hat{t}_0$ error	Nr. of samples	$\hat{l}$
$d_{l,1} = -0.08$ m^3/h	ϕ_5	00:00:47 (00:09:18)	00:00:26 (00:05:18)	20	1
$d_{l,1} = -0.17$ m^3/h	ϕ_5	00:00:37 (00:07:18)	00:00:23 (00:04:30)	14	1
$d_{l,1} = -0.28$ m^3/h	ϕ_5	00:00:30 (00:06:06)	00:00:23 (00:04:42)	7	1
$d_{l,6} = -0.06$ m^3/h	ϕ_5	01:25:18 (17:03:30)	00:08:52 (01:46:30)	4585	4
$d_{l,6} = -0.16$ m^3/h	ϕ_3	00:15:37 (03:07:18)	00:01:18 (00:15:30)	859	4
$d_{l,6} = -0.25$ m^3/h	ϕ_3	00:03:17 (00:39:18)	00:00:32 (00:06:18)	165	4

Table 8.3: Obtained results for the leakage detection method on the SWIL. Listed is the detection time, the estimated leak time and which detection signal (ϕ) that triggers the alarm. Times are listed in the format hh:mm:ss with the SWIL emulation time coming first and followed by real-life WDN time in parentheses. Lastly, the hypothesized leakage localization is shown ($\hat{l}$) with the number of samples in mean for leakage localization shown

From **Tab. 8.3**, it is shown that $\hat{l}$ estimates 3/6 of the leakages correctly. However, $\hat{l}$ is not expected to get the correct node, just a node close to the leaky node. As described in **Sec. 7.1** it is better looking at $\psi/\bar{\psi}$ and from this identify a set of nodes which are potentially leaky. Shown in **Tab. 8.4** is the values for ψ.

Leak node and size	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6	ψ_7
$d_{l,1} = -0.08$ m^3/h	**0.986**	0.960	-0.826	-0.991	0.968	0.764	-0.935
$d_{l,1} = -0.17$ m^3/h	**1.000**	0.888	-0.704	-0.998	0.903	0.627	-0.850
$d_{l,1} = -0.28$ m^3/h	**1.000**	0.891	-0.707	-0.999	0.905	0.631	-0.853
$d_{l,6} = -0.06$ m^3/h	-0.993	-0.843	0.636	**0.989**	-0.860	-0.553	0.799
$d_{l,6} = -0.16$ m^3/h	-0.796	-0.452	0.155	**0.776**	-0.481	-0.052	0.381
$d_{l,6} = -0.25$ m^3/h	-0.821	-0.489	0.197	**0.802**	-0.518	-0.095	0.420

Table 8.4: Values of the ψ vector resulting from using a mean of the residual taken from $\hat{t}_0$ to t_a generated by the leakage detection. The values shown in this table are the ones which leads to the $\hat{l}$ shown in **Tab. 8.3**

Looking at ψ in **Tab. 8.4** does not improve the results. Leaks at node 6 are not identified as being very likely even though the leakage is located at this node. This result is expected as this shows the same tendency as the results in **Sec. 7.3** where the ideal leak time and

a long mean of the residual is used. The reasons why the leakage localization is not close to identifying the leak at node 6 are believed to be the same as in **Sec. 7.3**.

The results of combining the leakage localizations with the leakage detection are deemed to be close to the performance of using the leakage localization alone with the ideal leak time and a long mean of the residual. After the two methods are combined the number of leaky nodes correctly identified on the SWIL even increases a bit. However, this is believed to be a coincidence caused by the mean of the residual used for leakage detection is very short and therefore somewhat random.

8.2 Large Scale test: Randers district

In **Sec. 7.1** the leakage localization scheme is described and implemented both in simulation and in the SWIL. In the simulation, the correct leaky nodes are identified, while in the real world, in the SWIL, one of the leaky nodes (node 1) is identified correctly, while node 6 is not identified. This section explores how the leakage localization scheme is applied to a large scale WDN. The large scale WDN which is being used is a simulation model of Randers in EPANET, provided by Verdo A/S [Verdo, n.d.]. Here it is assumed that the exact time of leakage is known, thus no leakage detection is needed. This is because EPANET does not incorporate measurement noise, which results in no noise being present in the residual, making the GLR detection algorithm irrelevant. EPANET is a simulation tool specifically for analyzing hydraulic behaviour in WDN's [U.S. Environmental Protection Agency, 2020]. This tool contains all the necessary elements constituting a WDN, such as junctions (vertices), links (pipes), pumps, reservoirs, valves and tanks, all of which have a unique ID and the option of choosing parameters to the specific element.

8.2.1 Modification of the EPANET model to multiple inlet district

The EPANET Randers model constitute the whole city area, which contains several pressure zones. For this report only a single pressure management area (PMA) with two inlets is needed. The Randers model is modified to only be the part of Randers fitting this specific case. However, other modifications have to be made. Firstly, the flow controlled pump is replaced by a node with the residential demand pattern, such that it is able to deliver the demand at a certain time. Secondly, the pressure controlled pump is replaced with an elevated reservoir as the elevation sets the pressure. These modifications work as if the pumps have ideal controllers. Moreover, when running the hydraulic analysis in EPANET many pipes shows no flow, which is troublesome when calculating the sensitivity matrix S shown in **Eq. 7.12**. This problem is dealt with by simply deleting these pipes and any unconnected graphs caused by this.

8.2.2 Leakage localization in the Randers district

As described, it is assumed that the exact leakage time is known. Because of this, leakage detection is not necessary. To implement the leakage localization scheme, a MATLAB toolbox, "EPANET-MATLAB-Toolkit" [KIOS CoE, n.d.], is used to run the hydraulic

analysis from MATLAB. Here a set of commands can be used to set the total simulation time and extract the pipe flows $q(t)$, nodal demands $d(t)$, nodal pressure $p(t)$ and elevations z. Here, the nodal demand reference $d_n(t)$ is extracted and the remaining vector is denoted as $\bar{d}(t)$. Likewise, the reference nodal pressure $p_n(t)$ is extracted from $p(t)$. It should be noted that EPANET defines a demand flowing into the WDN as negative and a demand going out of the WDN as positive. This definition is however opposite to that of this report and should be accounted for. This WDN consists of $n = 1525$ nodes and $m = 1689$ edges. As a choice it has been decided to have a sensor per 100 nodes, resulting in 16 pressure sensors. These sensor placements are picked such that sensors are spanned throughout the whole network. Next, a reduced-order model is constructed as described in **Sec. 6.1** giving a pressure estimate $\hat{p}(t)$ for each pressure sensor. Hereby the residual $r(t)$ is calculated from $r(t) = p(t) - \hat{p}(t)$.

As a part of the leakage localization scheme, the sensitivity matrix S given by **Eq. 7.12**, is needed. Firstly, the incident matrix for the whole network is required to obtain the partitions of the reduced incident matrix with respect to the chords $\bar{H}_\mathcal{C}$ and tree $\bar{H}_\mathcal{T}$. Lastly, the partial derivatives of the pipe resistance λ are determined as $\partial\lambda_\mathcal{C} = \frac{\partial\lambda(q_\mathcal{C})}{q_\mathcal{C}}\Big|_{\gamma\cdot\breve{a}_\mathcal{C}}$ and $\partial\lambda_\mathcal{T} = \frac{\partial\lambda(q_\mathcal{T})}{q_\mathcal{T}}\Big|_{\gamma\cdot\breve{a}_\mathcal{T}}$, where $\lambda(q) = \kappa|q|q$. It is clear that κ, $\breve{a}_\mathcal{C}$, $\breve{a}_\mathcal{T}$, and γ are unknown. As the MATLAB tool allows for extracting EPANET pipe parameters such as length L, diameter D and roughness coefficient ε for each pipe, the pipe friction factor f can be determined from **Eq. 3.17** using the assumption of a Reynolds number corresponding to that of turbulent flows, making this term negligible. This leads to an estimate of κ using **Eq. 3.19**. However no minor loss coefficients are available for the pipes, which makes this term have no influence. Furthermore, the values of $\breve{a}_\mathcal{C}$ and $\breve{a}_\mathcal{T}$ are found from the expression in **Eq. 3.88** and is restated and isolated in **Eq. 8.1**

$$\begin{bmatrix}\breve{a}_\mathcal{C}\\ \breve{a}_\mathcal{T}\end{bmatrix} = \begin{bmatrix}q_{\mathcal{C},m}\\ q_{\mathcal{T},m}\end{bmatrix} \cdot \frac{1}{\gamma d_{n,m}} \tag{8.1}$$

where $\sigma_m = \gamma d_{n,m}$. Since the network consists of consumers with both residential and business/industry demand curves, **Asm. 3.1** stating a fixed distribution $\breve{v}$ between each $n-1$ nodal demands, is no longer valid. To accommodate for this, a mean of measurements $q_\mathcal{C}(t)$, $q_\mathcal{T}(t)$, $\bar{d}(t)$, $d_n(t)$, and $\sigma(t)$ during a 24 hour period in normal operating condition (without a leakage) is used. These are denoted q_m, $\bar{d}_m$, $d_{n,m}$, and σ_m. From this, the mean nominal demand distribution vector is found as $\breve{\mathrm{v}} = -\bar{d}_m/\sigma_m$ and $\gamma = \sigma_m/d_{n,m}$. From this the sensitivity matrix S is calculated and the decision signal is formed as described in **Eq. 7.22**. To introduce a leakage at specific nodes in the network, a leakage demand curve, which is constant through the whole period, is added to the picked leaky nodes. After 24 hours of normal operation the leakage is then introduced by specifying a base demand for that specific node. In this implementation a leakage of -10 L/s is used. This leakage is then active for a 24 hour period.

Furthermore, the picking matrix $F_s \in \{0,1\}^{16\times n-1}$ and $e_l \in \{0,1\}^{n-(c-1)}$ is constructed and $\breve{\breve{v}} \in \mathbb{R}^{c-1}$ is the nominal demand distribution of the flow controlled inlet. Lastly, ψ_l is restricted to values between zero and one as done in **Eq. 7.25**. It should be noted that the residual used here is a mean of all residuals during the 24 hour period of the leakage.

The result of the leakage localization scheme for the Randers district is seen in **Fig. 8.1** Here, the nodes in the network have been assigned a color, which indicates how probable it is that the leakage is at the specific node. Here 1 (yellow) indicates that a leakage is most likely to be at that node. In the same manner, 0 (dark blue) is an indication of not being probable. From this, it is seen that the localization scheme can locate an area within the network where the leaking node is located. In **App. A.11** the remaining results are shown in **Fig. A.37** and **Fig. A.38**. These show similar results.

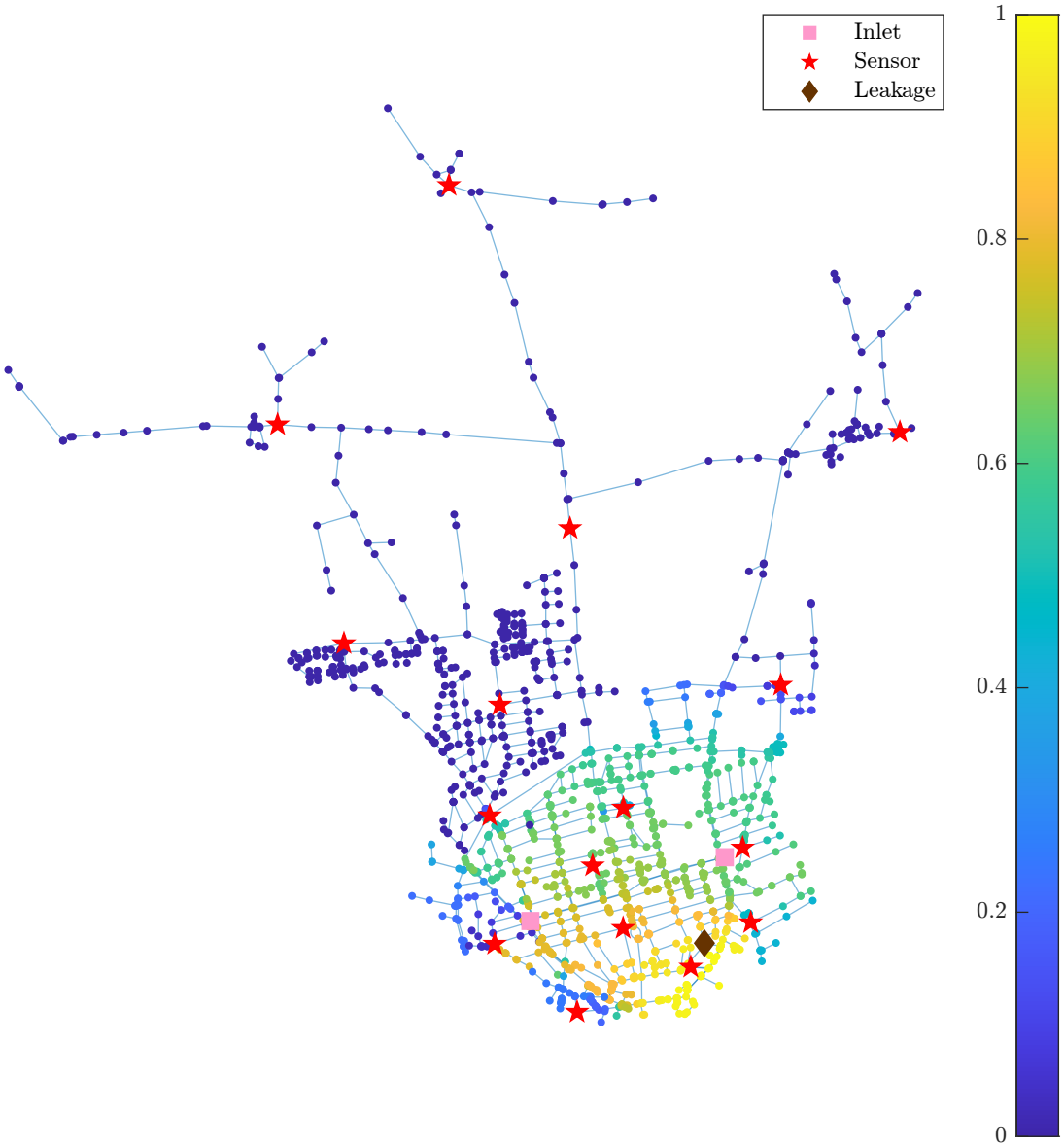


Figure 8.1: Leakage localization applied to Randers district

8.3 Discussion

This section discusses the methods and results used/found in this report. This section also discusses ideas that might improve the performance. A series of further tests that should be made before the algorithm is commercially viable is also discussed in this section.

8.3.1 Detection methods

For leakage detection, the algorithm GLR is used. The GLR has many subversions where the simplest version is used in this report. As mentioned in **Sec. 6.5.1**, a set of these subversions deals with converting a vector signal (a vector of residuals in this report) into a single decision signal. It could be interesting to investigate if this conversion reduces leakage detection times. Further, one of the vector versions of GLR even looks for a change in a specific direction of the vector to investigate if a fault is present [Blanke et al., 2015]. This would make a combination of leakage detection and localization possible. On the other hand, this would also mean having a GLR running for each node since each fault at each node means a different change in vector direction. That makes this approach computationally heavy. However, it is would still be interesting to look further into this approach.

8.3.2 Results

The first main result is the verification of the controller designed in **Sec. 5.4**. The controller can meet the references set for it. However, it is, when implemented on the SWIL, quite prone to oscillations. It is especially at low flows (flow far away from the operating point) prone to oscillations. The controller has only been checked for stability at the operating point. The increase in oscillations at flows far away from the operating point can be caused by the controller not being sufficiently robust. It could be considered designing a controller where stability is checked over a broader range of states. This is, however, not done in this report since the controller fulfilled the requirements set here. Further, the oscillations could even be beneficial for testing the robustness of the leakage detection/localization.

The residual generated in **Ch. 6** has small periodic variations depending on if the total consumption of the Water Distribution Network (WDN) is small or large. These variations lead to periodic change of mean in the residual. This small change in the mean is maybe partly to blame for $\hat{t}_0$ (the expected leak time) being far off the real leak time t_0 in **Sec. 6.5**. It might be possible to reduce these variations if more care is made when finding α and β. Using linear regression, as is done in this report, assumes X (input data) is noise-free for an unbiased estimate to be obtained. This is not 100% true since the flow sensor has some measurement noise. The measurement noise is even squared in the expression as well. The poor results for $\hat{t}_0$ generated by $\phi_3(t)$ could also partly be caused by that the variation in r_3 in case of a leak at node 1 are very small. Lastly, it is to be noted that $\hat{t}_0$ generated by $\phi_3(t)$ in case of a leak at node 1 is not used since $\phi_5(t)$ detects the leak a lot earlier leading to the $\hat{t}_0$ generated from $\phi_5(t)$ to be the one used.

Continuing on the topic of the GLR results, these show that it is possible to detect leakages

down to a certain size and with what seems to be a reasonable speed. The last part is, however, somewhat difficult to say since it is difficult to define what is fast in this aspect. For the results to be verified further a series of test should be carried out using different methods than GLR on the residual. Further, methods that do not use residuals should also be tested for comparison.

The results of leakage localization are very good in simulations where the correct node is almost always identified. This is true both when using the real leak time t_0 and the estimated leak $\hat{t}_0$. However, when applying the leakage localization scheme on data from the SWIL the results are less encouraging. Here the leaks made at node 1 are successfully identified but the ones generated at node 6 are not. For node 6 there is a tendency that as the leak gets bigger the detection gets worse. The values of ψ when no leak is present reveals that ψ as standard expects node 6 to be the leaky node. Therefore, it is hypothesized that the reason that the small leaks at node 6 can be localized is that these small leaks do not move ψ away from their initial bias. These poor results for the leakage localization on the SWIL are expected to be caused by the poor estimate for the friction of the pipes (κ). Further, the expectation is that the reason for leaks successfully being identified at node 1 is that this node is closer to an inlet meaning fewer pipes that can have a wrong friction model between leak and inlet.

Several things might have affected the obtained leakage detection and leakage localization results. Firstly, the detection time might change depending on what time of the day a leak is introduced. In this report, the time of the leak is not considered, meaning that leaks are introduced somewhat randomly. For further verifying the results, it should be tested what difference it makes, by introducing a leak when the demands are at their highest, lowest and base demand value.

Another thing that affects the performance of the leakage detection and the leakage localization is the placements of the pressure sensors. In this report, the chosen placements of the sensors are somewhat arbitrarily as described in **Sec. 4.2.3**. Therefore it is expected that the performance of the suggested algorithm will improve if the optimal set of sensor placement is used. As described in **Sec. 4.2.3** there exits several techniques for finding this optimal placement.

8.3.3 Future investigations/improvements

In real life, it is expected that users/neighbourhoods will have a slightly different demand curve. To accommodate this in testing the demand curve shown in **Sec. 4.3.1** could be skewed for some of the users. Skewing the curve could mean moving the two main usage periods (the two peaks in 4.11) to be slightly earlier or later in the day. Doing this would test the robustness of the detection/localization algorithm since the distribution of v and a would change slightly over time. Some test on this has already been done in **Sec. 8.2**. Here it is found that the method has poor robustness towards different demand curves. However, taking an average of data over a day (assuming day to day fixed distribution) can fix this at the cost of the leakage detection/localization becoming somewhat slow.

Staying on the subject of v and a it could also be interesting to see if the leakage detection

and the leakage localization designed can be adapted to handle if v and a has some known change in distribution during the day. That could be the case if a Pressure Management Area (PMA) both holds businesses and single-family homes (from **Fig. 2.1**), like in **Sec. 8.2**. This could potentially reduce the long leakage detection/localization described before.

After a leakage has happened and it is detected and fixed again, a strategy for reinitialization is needed. Should the detection system restart using the same α and β in the reduced model or will they have changed after the leakage is repaired. Further, the GLR buffer needs to be reinitialized in some manner. It would here be smart to use the samples before $\hat{t}_0$ (the expected leak time) to not start from an empty buffer to speed up the time before GLR is up and running again (a full buffer is needed for GLR to run). All these things need to be investigated further before implementing a real WDN.

In a real WDN, α and β should be updated slowly over time as the WDN slowly change due to the addition of new users or change in consumer behaviour. This is also described in **Sec. 6.1.4**. How these parameters need to be updated and how fast is still an open question. It is expected that the update should be much slower than the expected detection time and slower than the daily variations of the demand curve. In this report, α and β is estimated right before a test is run. Estimating before each test is necessary since it turned out that Smart Water Infrastructure Lab (SWIL) changes behaviour significantly every time it has been stopped and started again. The changing behaviour of the SWIL is believed to be caused by air accumulating quickly in the system when the SWIL is at a standstill. To some degree, the system behaves similarly to previous tests if, before each new test where the system has been at a standstill, it is set to circulate water for 2-3 days. That, however, was not possible in this project. Therefore the coefficients are updated once for each test run. This was found sufficient but updating the coefficient live is preferable as this is also needed in a real WDN.

8.4 Conclusion

This section evaluates the methods and the results of these described throughout the report. Firstly, recall the problem statement.

"How can a leakage detection and localization method be developed for a multiple inlet water distribution network which utilizes knowledge about the system and the actuators and sensors available?"

Firstly, it can be concluded that this report has presented a novel suggestion for a solution to this problem. The method introduced in this report indeed is a valid tactic, which is confirmed in the simulation tests for the leakage detection (**Sec. 6.4**) and the leakage localization scheme (**Sec. 7.2**). Further, the combined test for simulations in **Sec. 8.1.1** shows that individual parts work together. Lastly, **Sec. 8.2** show that the leakage localization method works on large Water Distribution Networks (WDN's) in simulation. The results show that practical implementation, which is done on the Smart Water Infrastructure Lab (SWIL), gives less encouraging results. Here, leakage detection works as intended (**Sec. 6.5**) but the leakage localization is only in some cases able to

correctly point out the leaky node (**Sec. 7.3**). It is from these results expected that the leakage localization might need some corrections to increase its robustness towards modelling errors. One of the problems found, that might need to be corrected to increase robustness, is that if a small set of users follows a different demand pattern, which leads to v changing throughout the day, the leakage localization becomes poor. This is of cause to be expected since it violates the assumption made in this report. In **Sec. 8.2** the problem of different demand patterns is dealt with by taking a 24-hour average. However, it radically slows down leakage localization since 24h of leakage data is needed before localization is possible. Further, it is not tested how this approach affects leakage detection. Lastly, a further problem is that it seems the leakage detection requires precise estimates of the model parameters, especially κ, to successfully localize leaks.

Furthermore, the multiple inlet model developed in **Sec. 3.5.2** is quite restrictive in regards to making several amount of assumptions and requirements for control. This leads to the leakage detection/localization also being restrictive. It has many assumptions that must be upheld for it to work. One of these is that it requires a special form for control to work, that is that only one pump is a pressure controller and the rest are flow controlled and must deliver a fixed percentage of the total demand at all time. This is very restrictive since in the real world there might be times where it is better if one of the pumps is halted to save energy. However, if this requirement is no problem, a special way of finding an output feedback controller is shown in **Sec. 5**.

Even if all the assumptions are upheld the residual found from subtracting the estimated pressure from the measured get some periodic error. This tendency is seen in **Fig. 6.8** and **Fig. 6.9**. These small variations, which corresponds to a change in the mean of the residuals over a short period, is expected to play a role in the large errors of the estimated leak time, $\hat{t}_0$, found by the GLR algorithm for node 3 in **Sec. 7.3**.

Summing up it is concluded that this report has introduced a somewhat working but quite restrictive method for leakage detection and leakage localization, which is applicable for WDN's with multiple inlets. The method works well as long as all the assumptions are held. However, it would be desirable if the robustness of the methods concerning assumptions not being held were bigger.

Bibliography

AAU. (n.d.). *Smart water infrastructures laboratory (swil)*. Retrieved May 27, 2021, from https://vbn.aau.dk/da/equipments/smart-water-infrastructures-laboratory-swil

Belimo. (2020). *Technical data sheet*. Retrieved March 22, 2021, from https://www.belimo.com/mam/Datasheets/en-gb/belimo_R2..xx-S.._datasheet_en-gb.pdf

Blanke, M., Kinnaert, M., Lunze, J., & Staroswiecki, M. (2015). *Diagnosis and fault-tolerant control*. Springer.

Christodoulou, S., Fragiadakis, M., Agathokleous, A., & Xanthos, S. (2017). *Urban water distribution networks: Assessing systems vulnerabilities, failures, and risks*. Elsevier Science & Technology.

Colebrook, C. F. (1939). Turbulent flow in pipes, with particular reference to the transition region between the smooth and rough pipe laws. *Journal of the Institution of Civil Engineers*, *11*(4), 133–156. https://doi.org/10.1680/ijoti.1939.13150

Danva. (2020). *Water in figures 2020 - benchmarking & statistics*. Retrieved February 5, 2021, from https://www.danva.dk/media/7251/2020_water-in-figures_web.pdf

Deo, N. (1974). *Graph theory with applications to engineering and computer science.* Prentice-Hall.

Franklin, G. F., Powel, J. D., & Emami-Naeini, A. (2015). *Feedback control of dynamic systems*. Pearson Education Limited.

Geil, O. (2008). *Elementary linear algebra 2015.* Pearson.

Grundfos. (2006). *The centrifugal pump*. GRUNDFOS Management A/S - Afdeling 3610 Fluid Mekanik.

Grundfos. (2019). *Technical data sheet*. Retrieved March 22, 2021, from https://api.grundfos.com/literature/Grundfosliterature-5439390.pdf

Grundfos. (n.d.). *Upm3k 25-75 130 aza*. Retrieved March 28, 2021, from https://product-selection.grundfos.com/dk/products/up-oem-north-america/upm3-oem/upm3k-59C90602?tab=variant-curves&pumpsystemid=1283546146

Jensen, T. N., & Kallesøe, C. S. (2016). Application of a novel leakage detection framework for municipal water supply on aau water supply lab. *2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, 428–433. https://doi.org/10.1109/SYSTOL.2016.7739787

Jensen, T. N., Kallesøe, C. S., Bendtsen, J. D., & Wisniewski, R. (2018). Plug-and-play commissionable models for water networks with multiple inlets. *2018 European Control Conference (ECC)*, 1–6. https://doi.org/10.23919/ECC.2018.8550092

Kallesøe, C. S. (2005). *Fault detection and isolation in centrifugal pumps* (Doctoral dissertation). Department of Control Engineering, Aalborg University.

Kallesøe, C. S. (2020). Lecture slides: Modelling of mechanical and thermal systems: Open hydraulic networks.

Kallesøe, C. S., Jensen, T. N., & Wisniewski, R. (2015). Adaptive reference control for pressure management in water networks. *2015 European Control Conference (ECC)*, 3268–3273. https://doi.org/10.1109/ECC.2015.7331038

KIOS CoE. (n.d.). *Epanet-matlab-toolkit*. Retrieved May 28, 2021, from https://epanet-matlab-toolkit.readthedocs.io/en/latest/

MathWorks. (n.d.). *Histogram*. Retrieved May 8, 2021, from https://se.mathworks.com/help/matlab/ref/matlab.graphics.chart.primitive.histogram.html

Methods, H., Walski, T. M., Chase, D. V., Savic, D. A., Grayman, W. M., Beckwith, S., & Koelle, E. (2003). *Advanced water distribution modeling and management*. Haestad Press.

Microsoft Dynamics 365. (2019). *2019 manufacturing trends report*. Retrieved February 5, 2021, from https://info.microsoft.com/rs/157-GQE-382/images/EN-US-CNTNT-Report-2019-Manufacturing-Trends.pdf

Miljøstyrelsen. (n.d.). *Find frem til vandtabet*. Retrieved February 5, 2021, from https://mst.dk/natur-vand/vand-i-hverdagen/vandtab

Montgomery, D. C. (2012). *Introduction to linear regression analysis* (Fifth edition.). John Wiley & Sons Ltd.

Naturstyrelsen. (2015). *Ledelsesmæssige og teknologiske metoder til reduktion af vandtab*. Retrieved February 5, 2021, from https://naturstyrelsen.dk/media/174521/ledelsesmaessige-og-teknologiske-metoder-til-at-reducere-vandtabet-i-dans-ny-docx-2.pdf

Perez, R., Sanz, G., Puig, V., Quevedo, J., Escofet, M. A. C., Nejjari, F., Meseguer, J., Cembrano, G., Tur, J. M. M., & Sarrate, R. (2014). Leak localization in water networks: A model-based methodology using pressure sensors applied to a real network in barcelona [applications of control]. *IEEE Control Systems Magazine*, *34*(4), 24–36. https://doi.org/10.1109/MCS.2014.2320336

Rathore, S. S. (2020). *Nonlinear optimal control in water distribution network*. AAU.

Rathore, S. S., Kallesøe, C. S., Wisniewski, R., & Jensen, T. N. (2021). *Leakage localization in municipal water supply using self adaptive reduced network models and sensitivity analysis* [Submitted].

Ribeiro, L., Sousa, J., Marques, A. S., & Simões, N. E. (2015). Locating leaks with trustrank algorithm support. *Water*, *7*(4), 1378–1401. https://doi.org/10.3390/w7041378

Rossman, L. A., Woo, H., Tryby, M., Shang, F., Janke, R., & Haxton, T. (2020). *Epanet 2.2 user manual*. Retrieved February 18, 2021, from https://cfpub.epa.gov/si/si_public_record_Report.cfm?dirEntryId=348882&Lab=CESER

Skogestad, S., & Postlethwaite, I. (2005). *Multivariable feedback control: Analysis and design*. John Wiley & Sons, Inc.

Swamee, P. K., & Sharma, A. K. (2008). *Design of water supply pipe networks*. Wiley. https://books.google.dk/books?id=F5dz28bt5RkC

UN. (2015). *Goal 6: Ensure access to water and sanitation for all*. Retrieved February 5, 2021, from https://www.un.org/sustainabledevelopment/water-and-sanitation/

U.S. Environmental Protection Agency. (2020). *Epanet 2.2 online user's manual*. Retrieved May 27, 2021, from https://epanet22.readthedocs.io/en/latest/

Verdo. (n.d.). *Om verdo*. Retrieved May 27, 2021, from https://www.verdo.com/dk/om-verdo/

Appendix A

A.1 Friction factor for transitional flow

$$
\begin{aligned}
AA &= -1.5634601348517065795 \\
AB &= 0.00328895476345399058690 \\
Y2 &= \frac{\varepsilon}{(3.7D)} + AB \\
Y3 &= -2log_{10}(Y2) \\
FA &= Y3^{-2} \\
FB &= FA \cdot \left(2 - \frac{AA\ AB}{Y2\ Y3}\right) \\
X1 &= 7FA - FB \\
X2 &= 0.128 - 17FA + 2.5FB \\
X3 &= -0.128 + 13FA - 2FB \\
X4 &= 0.032 - 3FA + 0.5FB \\
R &= \frac{Re}{2000}
\end{aligned}
\tag{A.1}
$$

$$
f = X1 + R(X2 + R(X3 + RX4)) \tag{A.2}
$$

A.2 Smart Water Infrastructure Laboratory step responses

To test the performance of the Smart Water Infrastructure Laboratory (SWIL) a series of steps is carried out.

A.2.1 Step response of single valve

This test evaluates the response time from applying an input step to a valve on the consumer station to reading a change in pressure and flow. The resulting response in flow and pressure is shown in **Fig. A.1** and **Fig. A.2**. Here it is clear that there is a significant delay before any output on the sensor is read, around 5 seconds for pressure and 10 seconds for flow. Further, it takes approximately extra 5 seconds for each to settle at a somewhat steady value. As mentioned in **Sec. 3.3.4** these delays are made up of several elements, the slow opening of the valve, the communication delay from the Central Control Unit (CCU) to the actuator and the delay from the sensor transmitting back to the CCU. Further, both sensors have internal filtering of the measured signal which leads to further delays/dynamics.

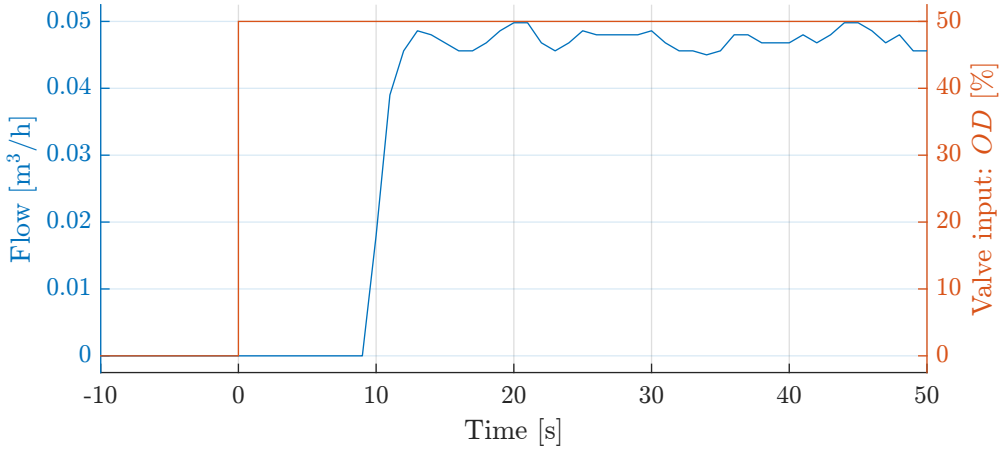


Figure A.1: Flow versus valve opening degree

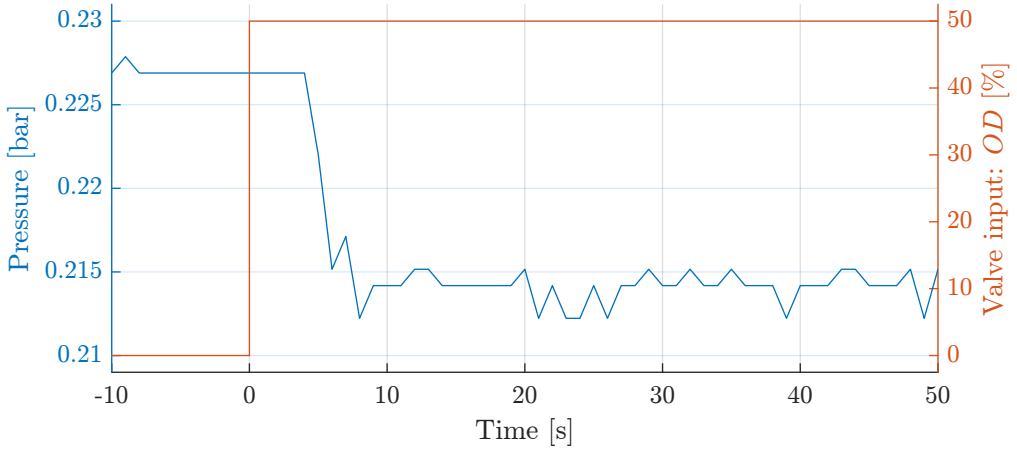


Figure A.2: Pressure versus valve opening degree

A.2.2 Step response of a single pump

The delay and the dynamics of the pump are investigated for the stability analysis and control design made in **Ch. 5**. A single pump is applied a step where the delay and the dynamics from the control input is registered by the flow sensor. The observations for flow and pressure are plotted in **Fig. A.3** and **Fig. A.4**. These plots show a delay from the input to the output of around 4 seconds for both sensors. Further, both show an overshoot in their response. This is deemed to be caused by the internal filtering in the sensors and/or caused by an overshoot in the internal speed controller of the pump since this should not be the natural behaviour of the system.

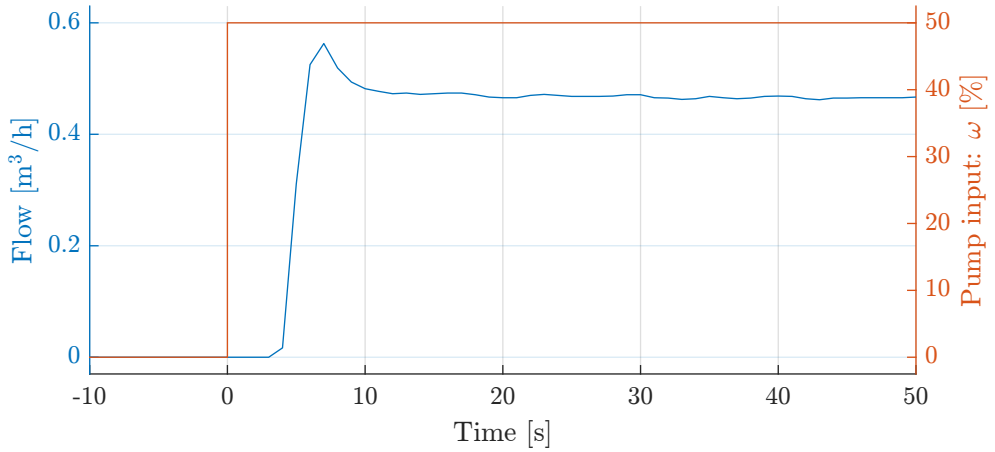


Figure A.3: Flow versus pump input

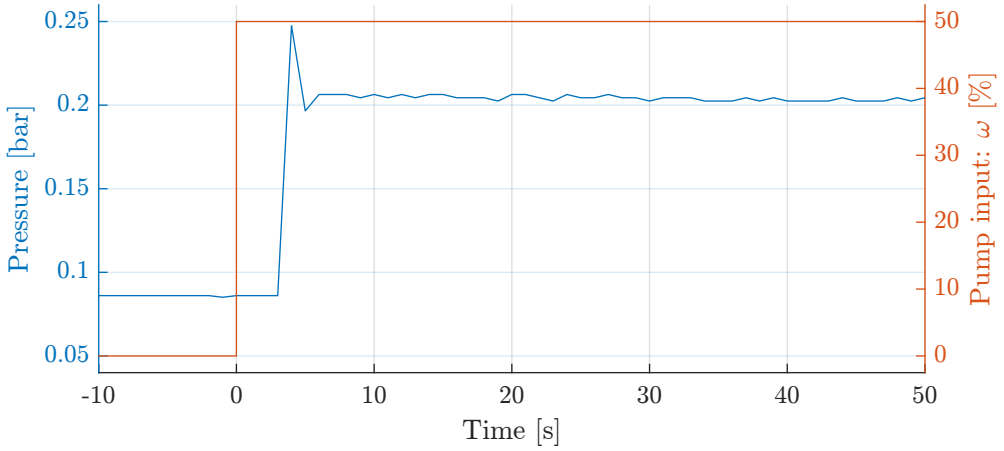


Figure A.4: Pressure versus pump input

A.3 Pump coefficients estimation

As described in **Sec. 3.3.3** the pump model is based on the coefficients ah_0, ah_1, and ah_2. These are to be estimated using the pump curve characteristics shown in [Grundfos, n.d.]. Here, as stated in the datasheet of the pump each curve corresponds to a fixed percentage speed of the pump (5%, 20%, 31%, 41%, 52%, 62%, and 73%). Due to the operating mode of the pump the lower the speed in percent, the faster the pump operates [Grundfos, 2019]. The former is inverted so that a high speed percentage corresponds to the pump operating faster.

To estimate the pump coefficients several data points are picked. The area of interest is however chosen as the area between the pump curves for pump speeds 20% and 52% (inverted speeds: 80% and 48%). For each curve five data points are picked for both flow q and head h.

The coefficients are estimated in MATLAB using the fit function, which fits a surface to the data points by specifying the pump model, $\Delta p = ah_0 \cdot \omega^2 + ah_1 \cdot \omega \cdot q - ah_2 \cdot q^2$, as the function to be estimated. The fit is seen in **Fig. A.5** and the estimated coefficients are seen in

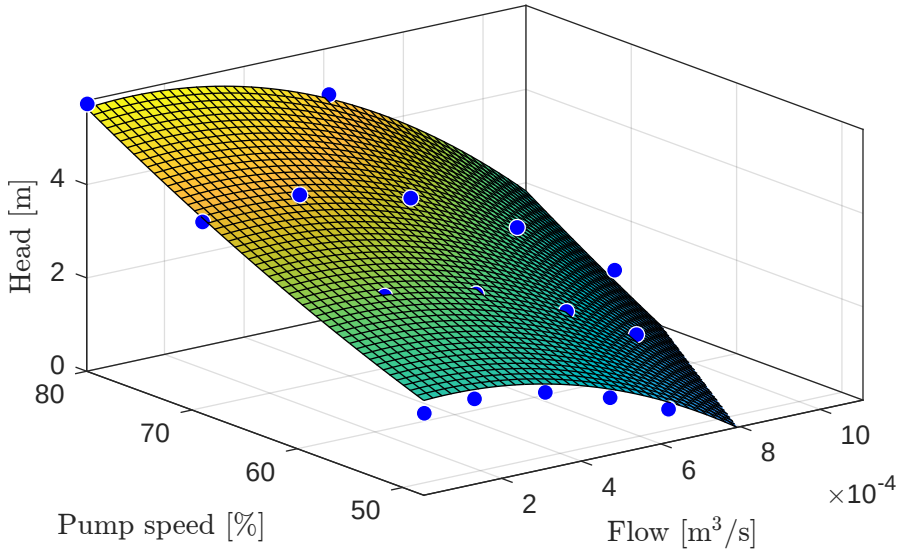


Figure A.5: Pump curve fit from extracted data points

a_{h2} $[s^2/m^5]$	a_{h1} $[s/m^2]$	a_{h0} [•]
$3.9205 \cdot 10^6$	11.3179	$8.8011 \cdot 10^{-4}$

Table A.1: Estimated pump coefficients

A.4 Graph theory matrices

The incidence matrix is constructed as seen in **Eq. A.3** with vertices $n = 14$ and edges $m = 15$.

$$H = \begin{array}{c} \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} & e_{11} & e_{12} & e_{13} & e_{14} & e_{15} \end{matrix} \\ \left[\begin{array}{cc|ccccccccccccc} 1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \\ v_{11} \\ v_{12} \\ v_{13} \\ v_{14} \end{matrix} \end{array} \tag{A.3}$$

The loop matrix B is constructed with loops $l = m - n + 1$ as seen in **Eq. A.4**

$$B = \begin{array}{c} \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} & e_{11} & e_{12} & e_{13} & e_{14} & e_{14} \end{matrix} \\ \left[\begin{array}{cc|ccccccccccccc} 1 & 0 & 1 & 0 & -1 & 1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \begin{matrix} L_1 \\ L_2 \end{matrix} \end{array} \tag{A.4}$$

Next, F is seen in **Eq. A.6** and is constructed based on the demand vector d_f shown in **Eq. A.5**

$$d_f = \begin{bmatrix} d_{v1} & d_{v2} & d_{v3} & d_{p4} & d_{p5} \end{bmatrix}^T \tag{A.5}$$

$$F = \begin{array}{c} \begin{matrix} d_{v1} & d_{v2} & d_{v3} & d_{p4} & d_{p5} \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \\ v_{11} \\ v_{12} \\ v_{13} \\ v_{14} \end{matrix} \end{array} \tag{A.6}$$

A.5 Numerical linear system

Numerical values for the linear state space matrices for the system with and without delay together with the model reduced model. For practical reasons some the units used in these models has been converted from the SI-units to work with flows in cubic meters per hour [m^3/h] and pressures in Bar.

A.5.1 System without delay

$$A = \begin{bmatrix} -4169.0 & 145.1 & 31760.0 & -521.0 & -4839.0 & -174.1 \\ 454.1 & -6195.0 & -678.1 & 20260.0 & -20860.0 & -486.4 \\ 3586.0 & -145.1 & -1.169 \cdot 10^5 & 65480.0 & 4839.0 & 174.1 \\ -6185.0 & -119.2 & 78570.0 & -1.207 \cdot 10^5 & 41240.0 & -436.4 \\ 1847.0 & 579.4 & 3686.0 & 32250.0 & -1.164 \cdot 10^5 & -2938.0 \\ 196.5 & -870.5 & 26070.0 & 3126.0 & 29040.0 & -4811.0 \end{bmatrix} \tag{A.7}$$

$$B = \begin{bmatrix} 187.3 & -4.854 \cdot 10^{-14} \\ 29.23 & 0 \\ -187.3 & -9.708 \cdot 10^{-14} \\ -29.23 & -115.3 \\ -216.2 & -270.3 \\ 432.7 & -9.708 \cdot 10^{-14} \end{bmatrix} \tag{A.8}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & 0.01955 & 0.01955 & 0.01955 & 0.01955 \end{bmatrix} \tag{A.9}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0.01334 \end{bmatrix} \tag{A.10}$$

A.5.2 System concatenated with padé approximated delay

$$A_c = \begin{bmatrix} -4169.0 & 145.1 & 31760.0 & -521.0 & -4839.0 & -174.1 & 0 & 0 \\ 454.1 & -6195.0 & -678.1 & 20260.0 & -20860.0 & -486.4 & 0 & 0 \\ 3586.0 & -145.1 & -1.169{\cdot}10^5 & 65480.0 & 4839.0 & 174.1 & 0 & 0 \\ -6185.0 & -119.2 & 78570.0 & -1.207{\cdot}10^5 & 41240.0 & -436.4 & 0 & 0 \\ 1847.0 & 579.4 & 3686.0 & 32250.0 & -1.164{\cdot}10^5 & -2938.0 & 0 & 0 \\ 196.5 & -870.5 & 26070.0 & 3126.0 & 29040.0 & -4811.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 64.0 & -1440.0 & 0 \\ 0 & 0 & 1.251 & 1.251 & 1.251 & 1.251 & 0 & -1440.0 \end{bmatrix} \tag{A.11}$$

$$B_c = \begin{bmatrix} 187.3 & -4.854 \cdot 10^{-14} \\ 29.23 & 0 \\ -187.3 & -9.708 \cdot 10^{-14} \\ -29.22 & -115.3 \\ -216.2 & -270.3 \\ 432.7 & -9.708 \cdot 10^{-14} \\ 0 & 0 \\ 0 & 0.8539 \end{bmatrix} \tag{A.12}$$

$$C_c = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1.0 & 45.0 & 0 \\ 0 & 0 & -0.01955 & -0.01955 & -0.01955 & -0.01955 & 0 & 45.0 \end{bmatrix} \tag{A.13}$$

$$D_c = \begin{bmatrix} 0 & 0 \\ 0 & -0.01334 \end{bmatrix} \tag{A.14}$$

A.5.3 System after model reduction

$$A_r = \begin{bmatrix} -1440.0 & 0 \\ 0 & -1440.0 \end{bmatrix}, \quad B_r = \begin{bmatrix} 2.472 & -1.726 \\ 0.03094 & 0.8119 \end{bmatrix} \tag{A.15}$$

$$C_r = \begin{bmatrix} 45.0 & 0 \\ 0 & 45.0 \end{bmatrix}, \quad D_r = \begin{bmatrix} -0.03863 & 0.02697 \\ -0.0004835 & -0.01269 \end{bmatrix} \tag{A.16}$$

A.6 Controller step test

The pump controller designed in **Sec. 5** is in that section tested by setting the desired demand curve as a reference for the flow controlled pump and a pressure reference for the pressure regulated pump. As an extra test, there is in this section described what happens if the reference has a step applied. This test is threefold. First a step is applied to ref_{dp4} going from 0.45m^3/h (Operation Point (OP)) to 0.6m^3/h. Secondly, a step is applied to ref_{p9} going from 0.4 bar (OP) to 0.5 bar. Lastly, both ref_{q14} and ref_{p9} is stepped simultaneously with the same increments as previously mentioned.

These three steps are first carried out in simulation on the non-linear model. Further, it is run on the Smart Water Infrastructure Lab (SWIL). In the simulation, the expected 5 s delay (2.5 s input delay and 2.5 s measurement delay) is applied.

For all the steps carried out the opening degree of the valves (OD_1, OD_2 and OD_3) are set to their operating points. Normally these would also follow a reference determined by the demand curve.

A.6.1 Simulation

The results from the first step in simulation is shown in **Fig. A.6** and **Fig. A.7**. The step on ref_{dp4} goes from 0.45m^3/h (OP) to 0.6m^3/h. The reference for p_9 is kept constant at its operating point (0.4 bar). From **Fig. A.6** it is clearly seen that changes in the demand flow (ref_{dp4}) result almost purely in changes in the control input ω_4. From **Fig. A.7** it is clearly seen that the controller meets the set references. It also seen that p_9 is only slightly disturbed (barely visible in this plot) by the change in ref_{dp4}.

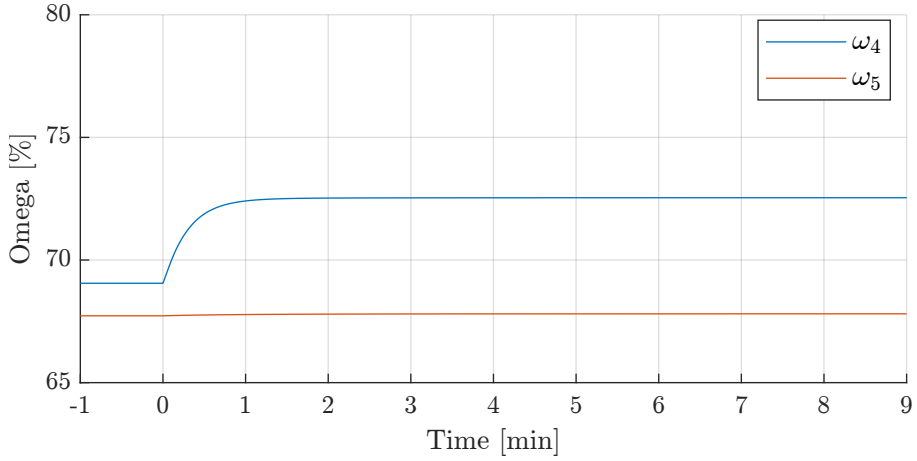


Figure A.6: Simulation: Control inputs - step on d_{p4}

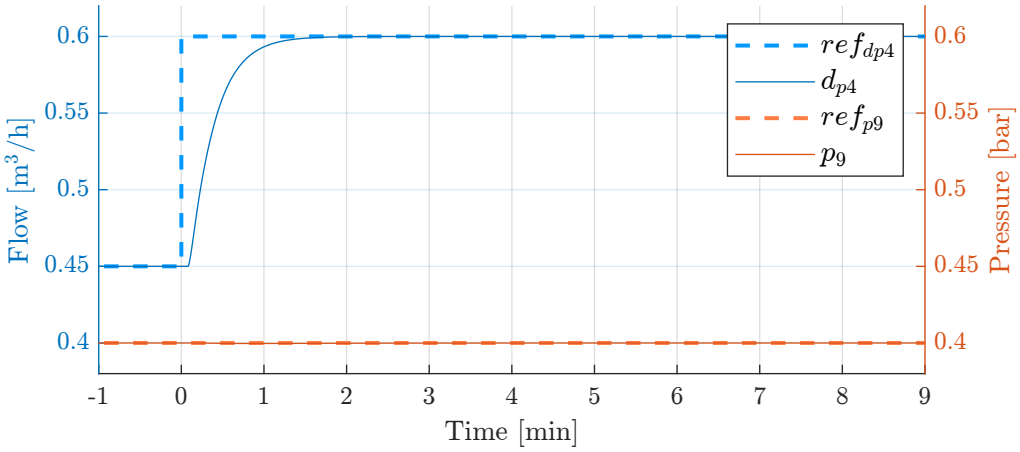


Figure A.7: Simulation: Output states with reference - step on d_{p4}

The results from the second step in simulations is shown in **Fig. A.8** and **Fig. A.9**. The control signal show that both ω_4 and ω_5 is altered to accommodate a change in pressure reference. The step on ref_{p9} goes from 0.4 bar (OP) to 0.5 bar. ref_{dp4} is kept a its OP (0.45m^3/h). It is seen from **Fig. A.9** that the reference on p_9 is met without disturbing d_{p4} significantly.

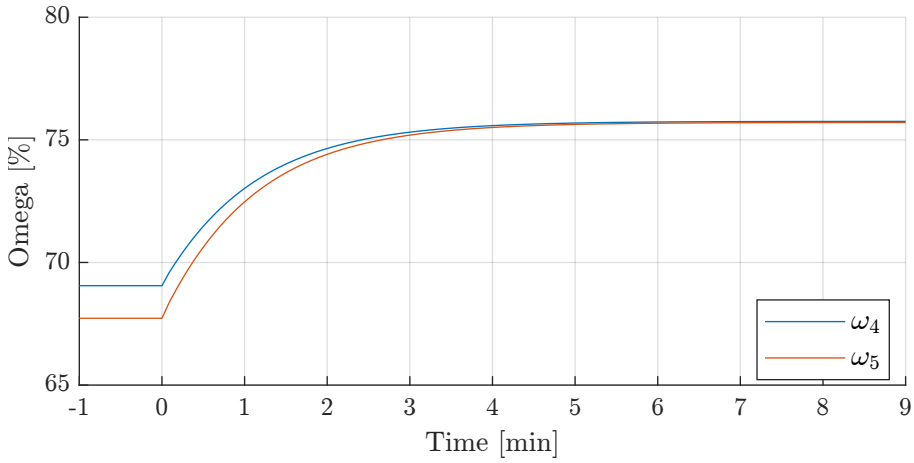


Figure A.8: Simulation: Control inputs - step on p_9

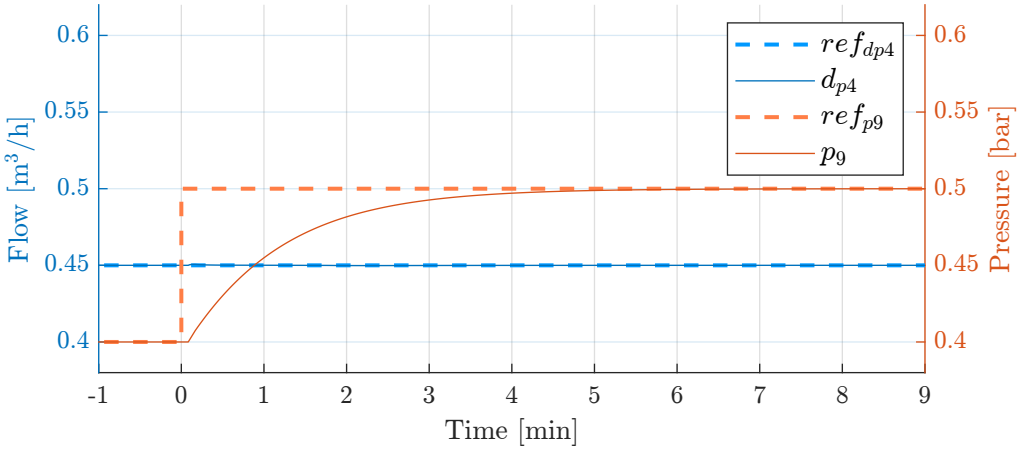


Figure A.9: Simulation: Output states with reference - step on p_9

The last step is a combination of the first two. ref_{dp4} is changed from 0.45m^3/h (OP) to 0.6m^3/h and ref_{p9} is set to go from 0.4 bar (OP) to 0.5 bar. As previously the plots in **Fig. A.10** and **Fig. A.10** the references are met using both pumps.

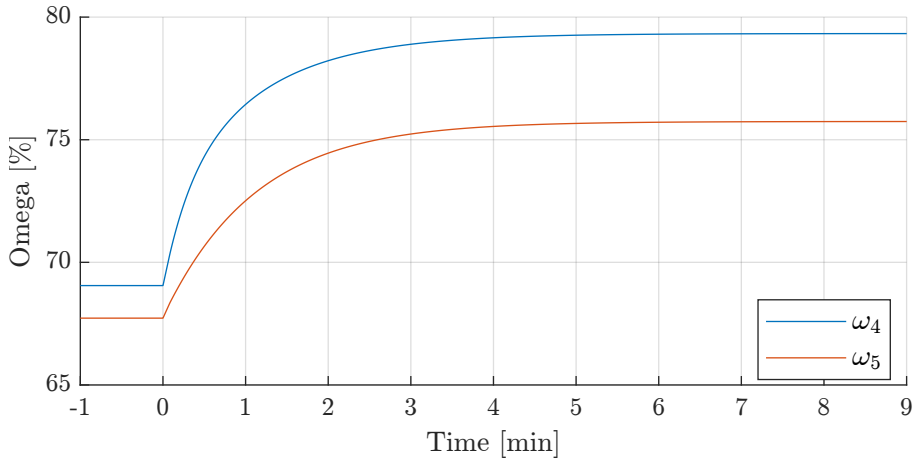


Figure A.10: Simulation: Control inputs - step on both d_{p4} and p_9

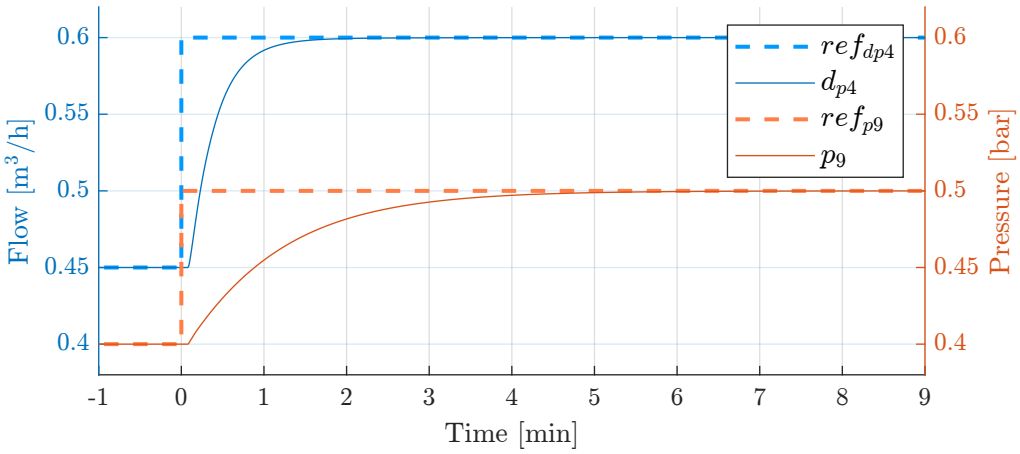


Figure A.11: Simulation: Output states with reference - step on both d_{p4} and p_9

From the simulations, it is concluded that the controller designed is sufficiently fast to handle the change in demands. The only reference which is changing in the simulation scenario is ref_{q14}. The rise time for this state is below 1 minute which is deemed to be sufficiently fast not to cause any larger phase shifts from reference to the state. The rise time for ref_{p9} is around 3 minutes. It should be no problem that the rise time is larger here since there is never applied a change in reference for p_9. Here, on the contrary, it is important that the pressure controller very slowly adapts to counteract the changing disturbances of consumer demands.

The controller is now tested on the SWIL to see if similar performance is observed.

A.6.2 SWIL

The Controller is now tested on the Smart Water Infrastructure Lab (SWIL), here the controller shows the same behaviour as in simulation. The results are plotted in **Fig. A.12** to **Fig. A.17**. Since the results in SWIL are similar to the ones from the simulation the same conclusions are mostly valid. The most significant difference in the SWIL compared to simulations is that the output states are noisy and even show some oscillations even though the control signal seems to be smooth.

The plots show that the control signal mainly changes slowly due to the integral action being quite dominant. This is desirable since the output states contain a lot of noise. The rest of this section is purely plots with a few comments.

In the fist step ref_{q14} is changed from 0.45m^3/h (Operation Point (OP)) to 0.6m^3/h. The resulting control inputs is shown in **Fig. A.12** and **Fig. A.13** and the resulting response in output states are shown in **Fig. A.13**.

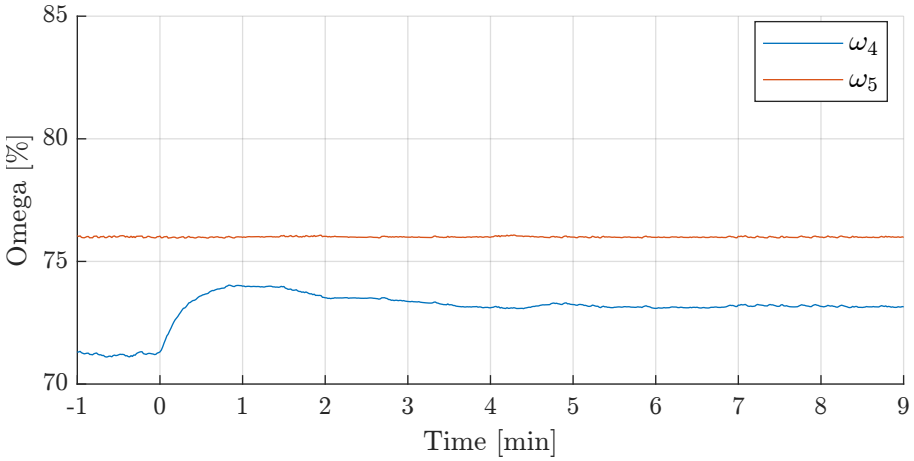


Figure A.12: SWIL: Control inputs - step on d_{p4}

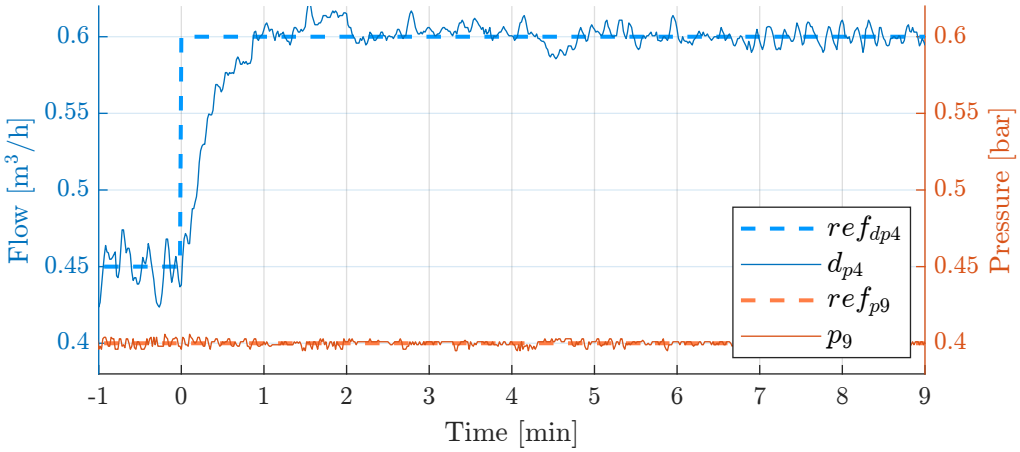


Figure A.13: SWIL: Output states with reference - step on d_{p4}

The second step is on the pressure reference (ref_{p9}) which is set to go from 0.4 bar (OP) to 0.5 bar. The change in control inputs are shown in **Fig. A.14** and the change in output states are shwon in **Fig. A.15**.

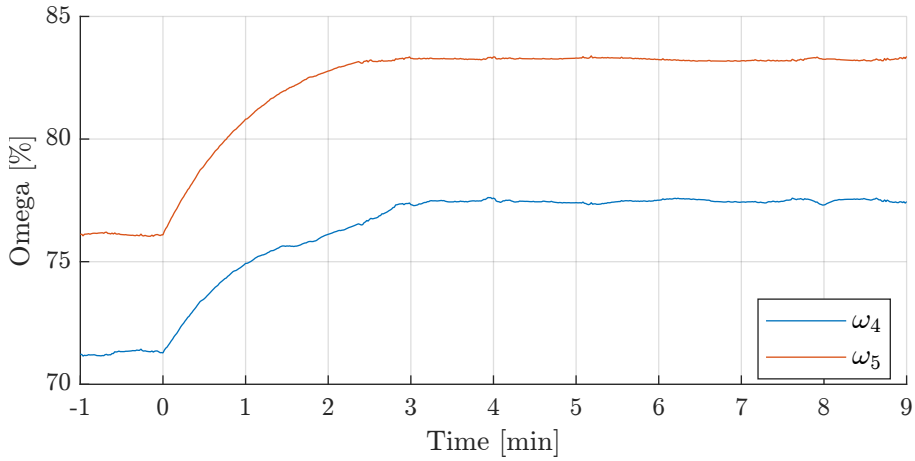


Figure A.14: SWIL: Control inputs - step on p_9

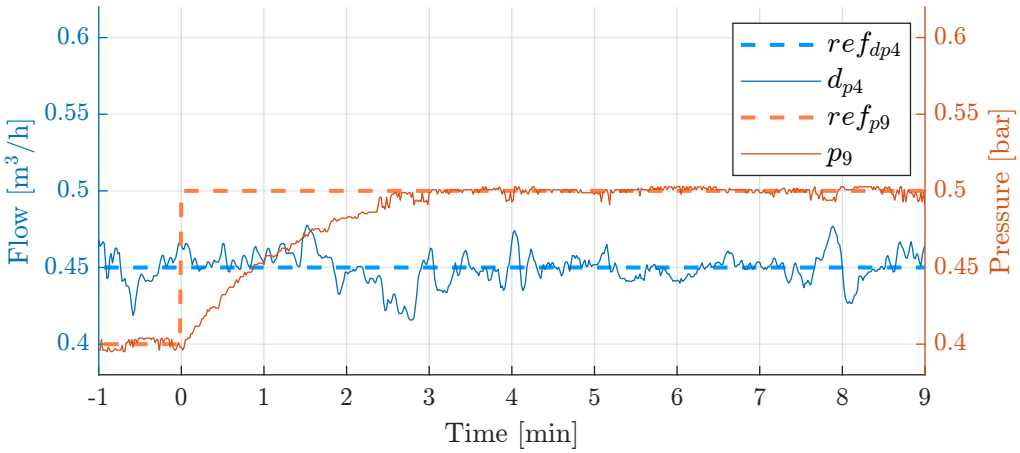


Figure A.15: SWIL: Output states with reference - step on p_9

The last step is a combination of the first two. ref_{dp4} is changed from 0.45m^3/h (OP) to 0.6m^3/h and ref_{p9} is set to go from 0.4 bar (OP) to 0.5 bar. The resulting plots are shown in **Fig. A.16** and **Fig. A.16**.

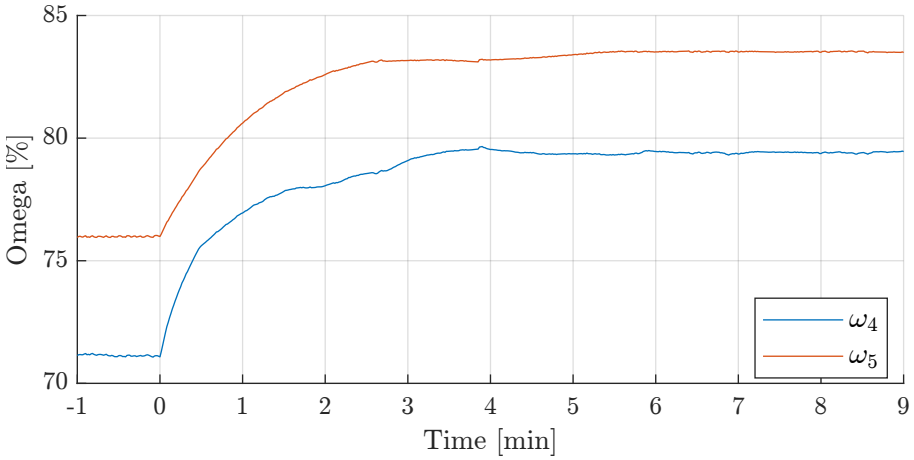


Figure A.16: SWIL: Control inputs - step on both d_{p4} and p_9

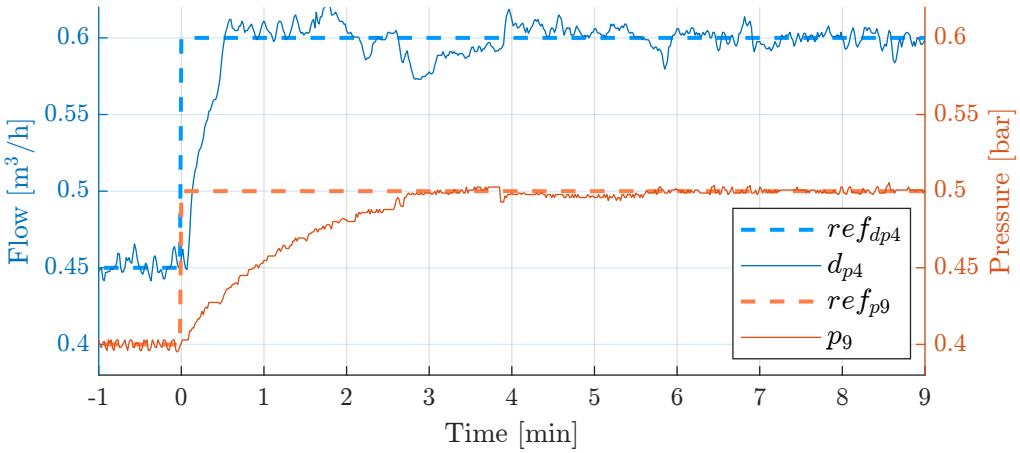


Figure A.17: SWIL: Output states with reference - step on both d_{p4} and p_9

A.6.3 Conclusion

The step responses made in the SWIL shows that the controller does not behave perfectly. The plotted steps clearly show some variations, especially for the flow reference. If this comes from the sensor measurement, the delay or simply the controller being only marginally stable is not investigated. This is not done since the variations are deemed to be so small that the controller can hold the reference to a satisfying degree. The variations seen can in the context of this report even be quite useful for testing the robustness of the leakage localization/detection.

A.7 Consumer demand flow controller

To emulate the consumer demand in simulation and in the Smart Water Infrastructure Laboratory (SWIL) a flow controller for the valves on the consumer modules is needed. Here it is chosen to use a simple hand-tuned PI-controller. This off-cause comes with the disadvantage that it can be hard saying anything about stability, poles and overshoot of the designed controller. However, as long as the controller is not unstable, which will be quite easy to observe, the controller does not need to behave particular nice. If it oscillates some or has a small steady-state error all this will just help to test the robustness of the pump controller and the leakage detection algorithm. Robustness in a real usage scenario is needed since consumers off cause does not match the demand curve perfectly. By this argumentation it is chosen to use a hand-tuned PI-controller.

The equation for calculating the change in control signal $\tilde{OD}_i = OD_i - OD_i^\star$, where $OD_i^\star$ is the i'th valve's operating point, for the PI-controller is shown in **Eq. A.17**. The full control signal OD_i is obtained by adding $\tilde{OD}_i$ with the operating point $OD_i^\star$, that is: $OD_i = \tilde{OD}_i + OD_i^\star$

$$\tilde{OD}_i(t) = K_P \cdot (ref_{dvi}(t) - d_{vi}(t)) + K_I \int_{T=0}^{t} (ref_{dvi}(T) - d_{vi}(T))dT \tag{A.17}$$

After a bit of tuning, it has been found that a proportional gain of $K_P = 5$ and an integral gain $K_I = 10$ gives satisfactory results. This controller can maintain a reference to a satisfying degree with some fluctuations which is fine for this purpose. Some of the fluctuations come from backlash in the valves. this means that if the controller overshoots a bit and has to adjust in the other direction it will take some time before the opening degree actually begins to change. There exist methods to handle backlash, but since the fluctuations that the backlash gives can be useful in testing the robustness of the leakage detection algorithm, these methods are not considered.

A.7.1 Test

The controller is tested on the Smart Water Infrastructure Lab (SWIL) together with the controller for the pumps (**Sec. 5.4.2**). It is tested by setting the demand curve from **Sec. 4.3.1** multiplied with the demand distributions vector (**Eq. 4.2**) as a reference for each flow through a valve. The resulting control signals are shown in **Fig. A.18** and the resulting produced flows are shown in **Fig. A.19**.

It is clearly seen that the flows produced are quite noisy. Some of the noise comes from the sensors, some come from the variations in pump flow and some come from the variations in OD that the controller makes. Further, it is seen that due to that the valves have a significant dead-band when changing direction the control signal for the valves has significant variations in places where the flow references are almost steady.

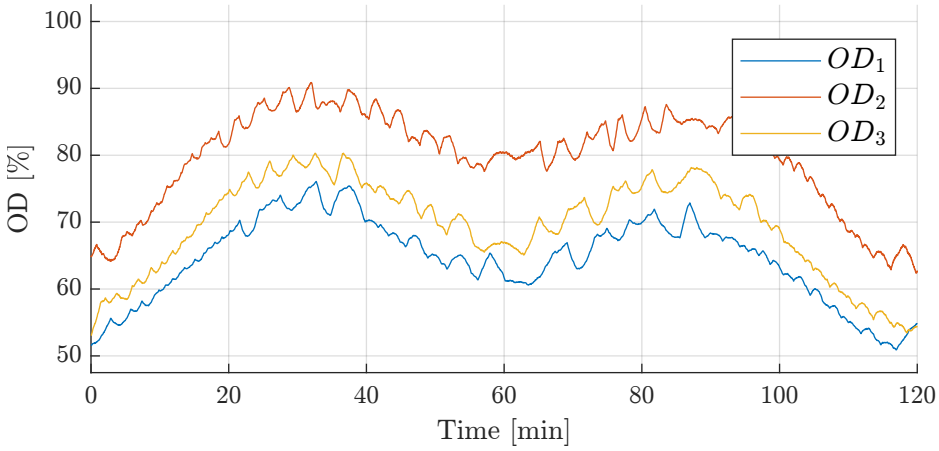


Figure A.18: Control signals for the 3 valves

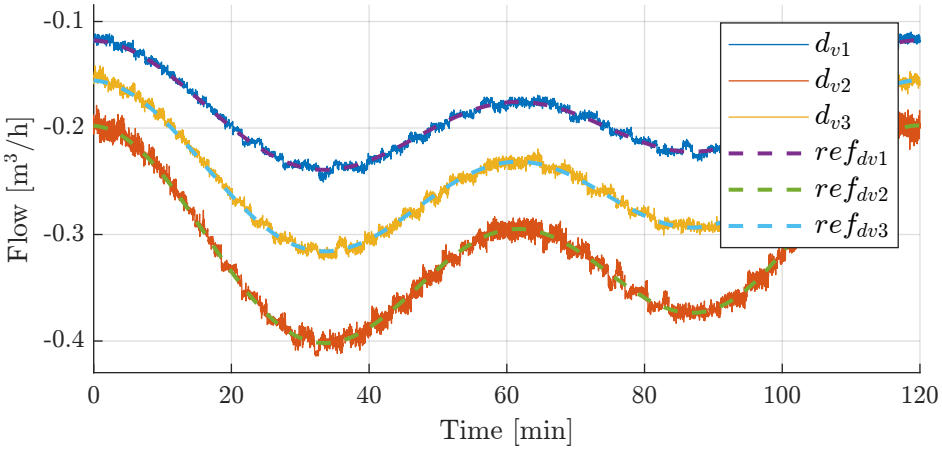


Figure A.19: Flows produced through the valves versus their reference

A.7.2 Conclusion

It is concluded that the controller is to a satisfying degree able to follow a set of references for consumer demand flows. The noisy output is deemed to mainly come from the sensors and from the way the pump is controlled. Even though the noise should come from the consumer demand flow controller it is at an acceptable level. This noise even has the benefit of helping to test the robustness of the leakage detection and the leakage localization.

A.8 Parameter change detection methods

In this section, two methods of detecting a change of parameters are derived, namely CUSUM and GLR. The derivation in the coming sections are based on [Blanke et al., 2015].

A.8.1 Cumulative Sum (CUSUM)

CUSUM is an algorithm designed to detect a change in the distribution that inputted data arrives from. This can e.g. mean a change in the mean and/or a change in the variance. In the case of pressure residuals, the test will be concerning a change in mean.

The equation shown below for the CUSUM is for bath/offline detection. This is simply because the derivation for this is simpler. In the end, these equations are adapted for online detection.

Formal description

In a sequence of an independent and identically distributed (IID) random variable $z(1), z(2), \ldots, z(k)$ with probability density function (pdf) $p_\theta(z)$. Here, θ is the scalar parameter that might have changed (a change in the mean in this case), choose at time k between the following two hypotheses:

$$\begin{aligned} &\mathcal{H}_0 : \theta = \theta_0 \text{ for } 1 \leq i \leq k \\ &\mathcal{H}_1 : \theta = \theta_0 \text{ for } 1 \leq i \leq k_0 - 1 \text{ and } \theta = \theta_1 \text{ for } k_0 \leq i \leq k \end{aligned} \tag{A.18}$$

where k_0 is the time of the parameter change and θ_1 is denoted as the changed parameter, in this case it changes to a known mean. Further k_0 is unknown [Blanke et al., 2015]. Here, $\mathcal{H}_0$ is the hypothesis that the mean is unchanged and $\mathcal{H}_1$ being the hypothesis of a changed mean.

Define the log-likelihood ratio at the i'th sample as:

$$s(z(i)) = \ln\left(\frac{p_{\theta_1}(z(i))}{p_{\theta_0}(z(i))}\right) \tag{A.19}$$

Here it is seen that if it is most likely that $z(i)$ belong to the distribution p_{θ_0}, then $s(z(i))$ is smaller than 0. On the other hand if it is most likely that $z(i)$ belong to the distribution p_{θ_0}, then $s(z(i))$ is greater then 0.

Further, taking the cumulative sum of **Eq. A.19** results in **Eq. A.20**.

$$S_j^k = \sum_{i=j}^{k} s(z(i)) = \sum_{i=j}^{k} \ln\left(\frac{p_{\theta_1}(z(i))}{p_{\theta_0}(z(i))}\right) \tag{A.20}$$

where j is a counting variable of the hypothetical of the parameter change. From this the decision function can be defined as:

$$\phi(k) = \max_{1\leq j\leq k} S_j^k = \max_{1\leq j\leq k} \sum_{i=j}^{k} \ln\left(\frac{p_{\theta_1}(z(i))}{p_{\theta_0}(z(i))}\right) \tag{A.21}$$

It decided whether the data belongs $\mathcal{H}_0$ or $\mathcal{H}_1$ by threshold, that is:

$$\begin{aligned} &\text{if } \phi(k) \leq h \text{ accept } \mathcal{H}_0 \\ &\text{if } \phi(k) > h \text{ accept } \mathcal{H}_1 \end{aligned} \tag{A.22}$$

where h is some threshold value. Naming the first instance where $g(k)$ is above h the alarm time k_a an estimate for time the parameter θ changed, $\hat{k}_0$, is given as:

$$\hat{k}_0 = \underset{1\leq j\leq k_a}{\operatorname{argmax}} S_j^k = \underset{1\leq j\leq k_a}{\operatorname{argmax}} \sum_{i=j}^{k} \ln\left(\frac{p_{\theta_1}(z(i))}{p_{\theta_0}(z(i))}\right) \tag{A.23}$$

Gaussian simplification

As described in **Sec. 6.2**, the noise of the residuals are approximated to be Gaussian distributed. This allows for a simplification in the CUSUM algorithm, which is only valid in the Gaussian case. Given a Gaussian distribution shown by **Eq. A.24**

$$p_\mu(z(i)) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(-\frac{(z(i)-\mu)^2}{2\sigma^2}\right) \tag{A.24}$$

Using the definition of the log likelihood ratio given by **Eq. A.19** and inserting **Eq. A.24** for the pdf p_{μ_0} with mean μ_0 and standard deviation σ, and for p_{μ_1} with mean μ_1 and variance σ. Here it is assumed that only a change in mean is happening, thus making the variance common and constant for both Gaussian distributions. The log likelihood ratio is then reduced as seen in **Eq. A.25**.

$$S_j^k(\mu_1) = \frac{\mu_1 - \mu_0}{\sigma^2} \sum_{i=j}^{k} \left(z(i) - \frac{\mu_0 + \mu_1}{2}\right) \tag{A.25}$$

This is the final expression for CUSUM in the special case of Gaussian distributions. However the mean μ_1 of the residual with leakages is yet unknown. This problem is solved in the next section.

A.8.2 Generalized likelihood ratio (GLR)

As described in **Sec. A.8.1**, CUSUM assumes that the mean of the residual with a leakage mean μ_1 is known. However, in reality $\hat{\mu}_1$ is unknown. To overcome this problem, **Eq. A.25** is maximized with respect to the unknown mean μ_1 as seen in **Eq. A.26**.

$$\frac{\partial S_j^k(\mu_1)}{\partial \mu_1} = \frac{1}{\sigma^2} \sum_{i=j}^{k} \left(z(i) - \frac{\mu_0 + \mu_1}{2}\right) - \frac{k-j+1}{2} \cdot \frac{\mu_1 - \mu_0}{\sigma^2} = 0 \tag{A.26}$$

From this, the estimate of the mean $\hat{\mu}_1$ is given by **Eq. A.27** [Blanke et al., 2015].

$$\hat{\mu}_1(k,j) = \frac{1}{k-j+1} \sum_{i=j}^{k} z(i) \tag{A.27}$$

Inserting **Eq. A.27** into **Eq. A.25** and reducing, results in **Eq. A.28**, which only is given by the parameters for the known Gaussian distribution, namely for the residual without

any leakages μ_0.

$$S_j^k(\hat{\mu}_1(k,j)) = \frac{1}{2\sigma^2}\frac{1}{k-j+1}\left(\sum_{i=j}^{k}(z(i)-\mu_0)\right)^2 \tag{A.28}$$

Substituting **Eq. A.28** into the definition of the decision function $\phi(k)$ given by **Eq. A.21** is then expressed in **Eq. A.29**.

$$\phi(k) = \frac{1}{2\sigma^2}\max_{1\leq j\leq k} S_j^k(\hat{\mu}_1(k,j)) \tag{A.29}$$

Modifying the decision function in **Eq. A.29** to consider a window with M samples is then given in **Eq. A.30** [Blanke et al., 2015]. This window is then run through a whole sequence of samples. This modification makes the algorithm able to run online on the setup.

$$\phi(k) = \frac{1}{2\sigma^2}\max_{k-M+1\leq j\leq k}\frac{1}{k-j+1}\left(\sum_{i=j}^{k}(z(i)-\mu_0)\right)^2 \tag{A.30}$$

A.9 Leakage detection: results from SWIL

This section presents the plots from the GLR test on the SWIL not shown in **Sec. 6.5**. Zoomed in views of the decision functions for the medium sized leaks is shown separately on **Fig. A.20** to **A.23**. On **Fig. A.24** to **Fig. A.29** full time series plots of the two decision functions is presented for each of the leaks.

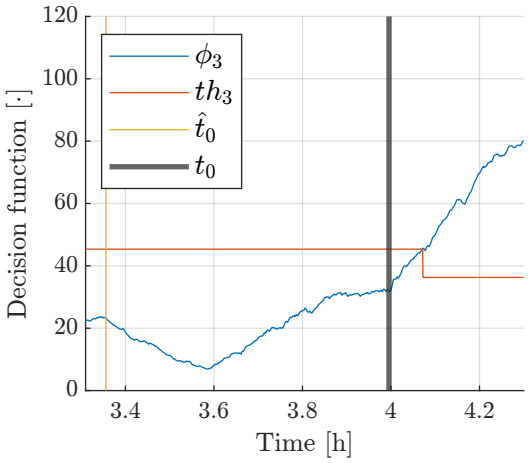


Figure A.20: GLR algorithm using residual from node 3 with a leak at node 1: $d_{l1}(t) = -0.17\text{m}^3/\text{h}$

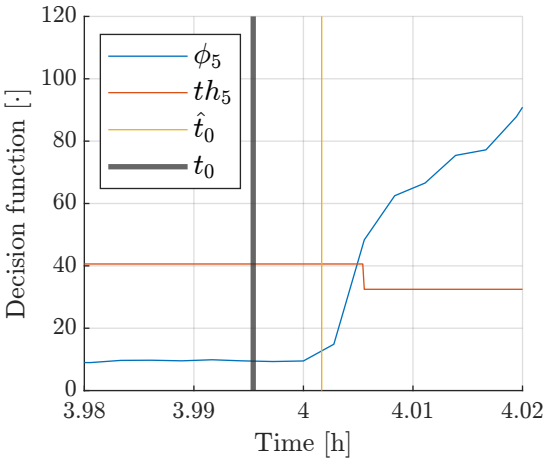


Figure A.21: GLR algorithm useing residual from node 5 with a leak at node 1: $d_{l1}(t) = -0.17\text{m}^3/\text{h}$

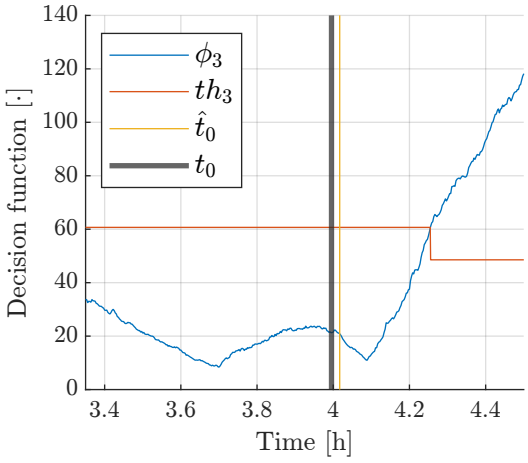


Figure A.22: GLR algorithm using residual from node 3 with a leak at node 6: $d_{l6}(t) = -0.16\text{m}^3/\text{h}$

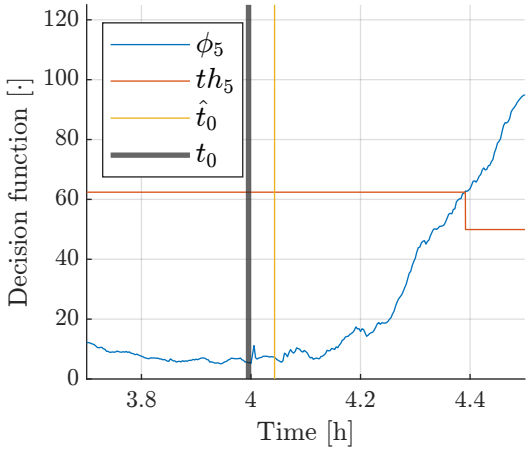


Figure A.23: GLR algorithm using residual from node 5 with a leak at node 6: $d_{l6}(t) = -0.16\text{m}^3/\text{h}$

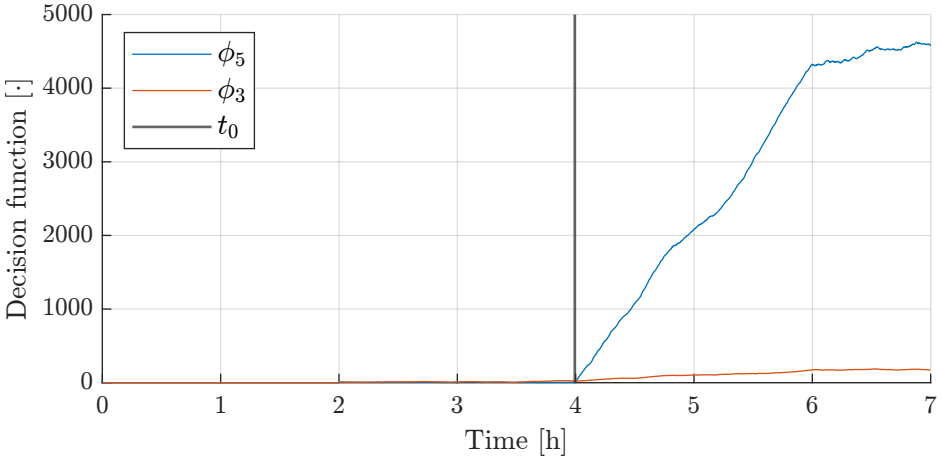


Figure A.24: GLR algorithm useing residuals from node 3 and 5 with a leak at node 1: $d_{l1}(t) = -0.08\text{m}^3/\text{h}$

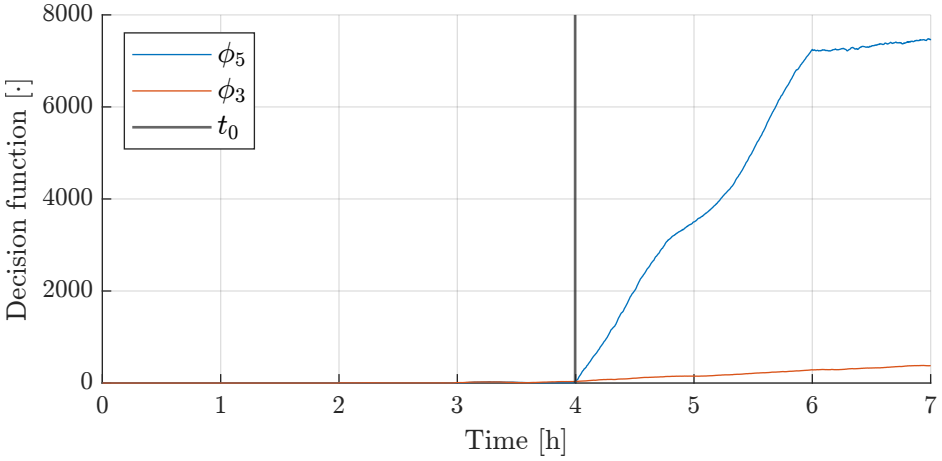


Figure A.25: GLR algorithm using residuals from node 3 and 5 with a leak at node 1: $d_{l1}(t) = -0.17\text{m}^3/\text{h}$

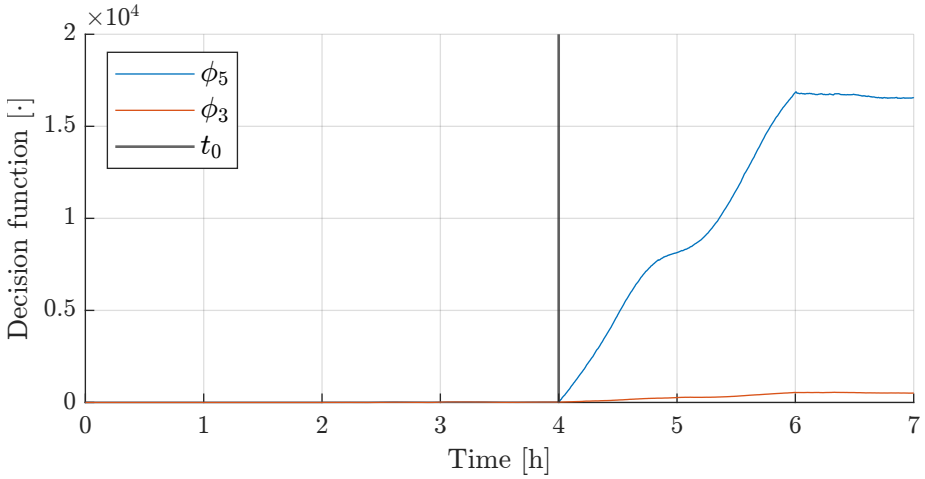


Figure A.26: GLR algorithm using residuals from node 3 and 5 with a leak at node 1: $d_{l1}(t) = -0.28\text{m}^3/\text{h}$

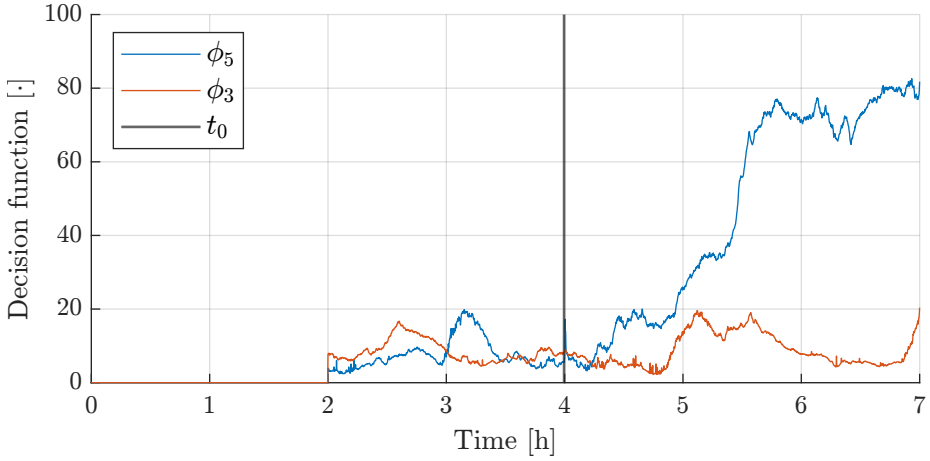


Figure A.27: GLR algorithm using residuals from node 3 and 5 with a leak at node 6: $d_{l6}(t) = -0.06\text{m}^3/\text{h}$

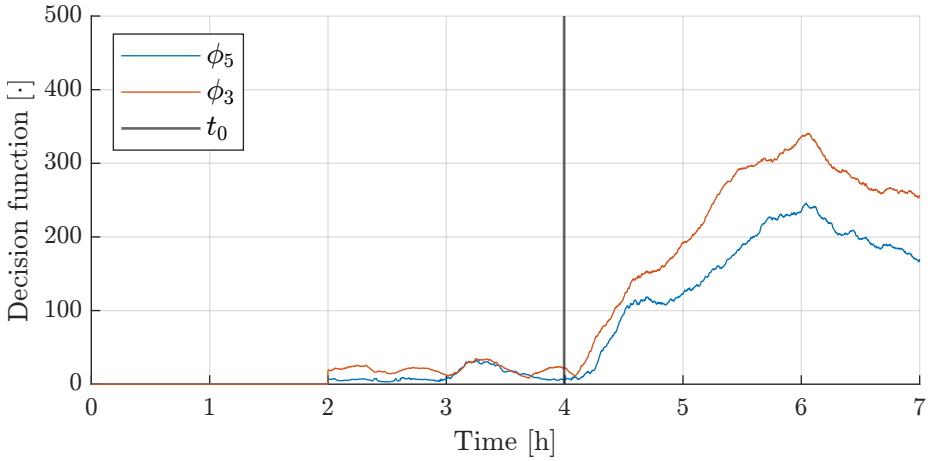


Figure A.28: GLR algorithm using residuals from node 3 and 5 with a leak at node 6: $d_{l6}(t) = -0.16\text{m}^3/\text{h}$

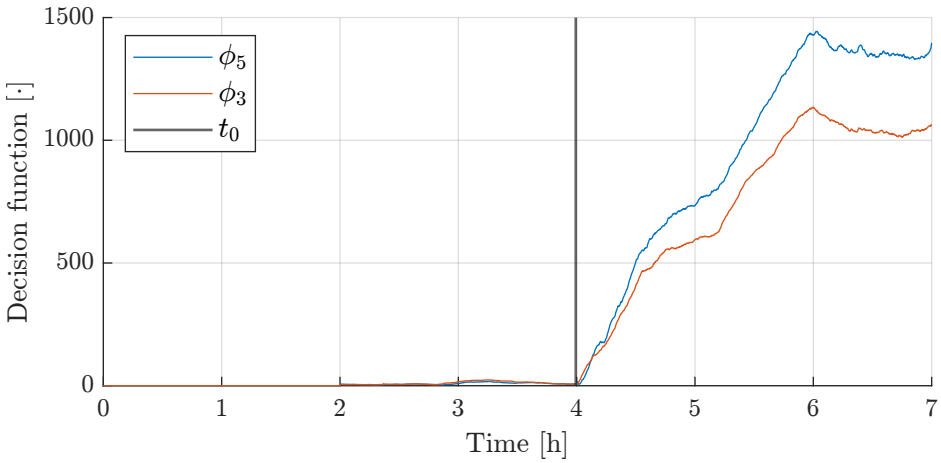


Figure A.29: GLR algorithm using residuals from node 3 and 5 with a leak at node 6: $d_{l6}(t) = -0.25\text{m}^3/\text{h}$

A.10 Leakage localization: Simulation of a leak at every node

In **Sec. 7.2** selected nodes (node 1 and node 6) are applied a leak of two sizes (0.1 m^3/h and 0.3 m^3/h). In this section leaks of the same magnitude are applied on every node one by one and plotted. This is made in addition to the previous plots in **Sec. 7.2**. The plots are produced in simulation. The residual used in $\psi(t)$ is a mean of the residual for the first 1000 samples where a leak is present. Here it is assumed that the true leak time, k_0, is known. The resulting plots are shown in **Fig. A.30** to **Fig. A.36**. The values for ψ at every node for all the leaks are shown in **Fig. A.2**.

The results clearly show that in most instances (6/7) the correct node is identified. Only in a single case, the wrong node is identified ($d_{l,5}$ is identified to be at node 2). However, the second-highest value for this leak is the correct node. Generally, a set of nodes for each leak almost has the same values for ψ_l. The set v_2, v_5 and v_6 (and to some degree also v_1) have values that are close to each other in case of any leak. Further, the set v_3 and v_7 (and to some degree also v_4) also follows each other. All this indicates that the network is split into two main detection regions. This is caused by the residual having almost opposite direction dependent on if the first set has a leaky node or the second set has a leaky node.Further, the residual only changing direction slightly between the nodes in each sets.

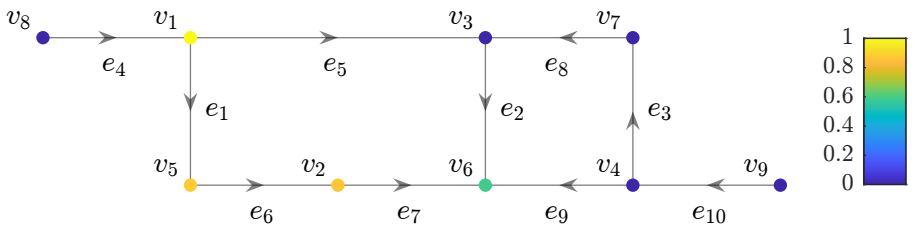


Figure A.30: Simulated values for $\bar{\psi}(t)$ for a leak at node 1: $d_{l,1}(t) = -0.1$ m^3/h

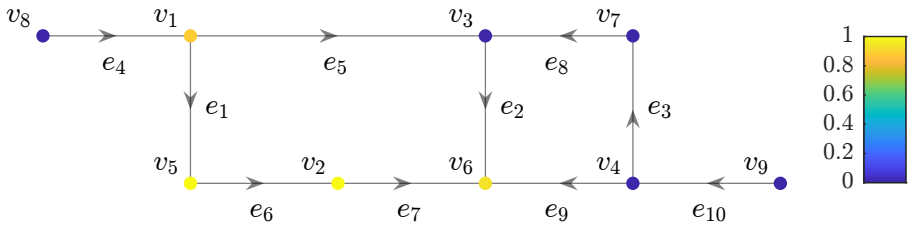


Figure A.31: Simulated values for $\bar{\psi}(t)$ for a leak at node 2: $d_{l,2}(t) = -0.1$ m^3/h

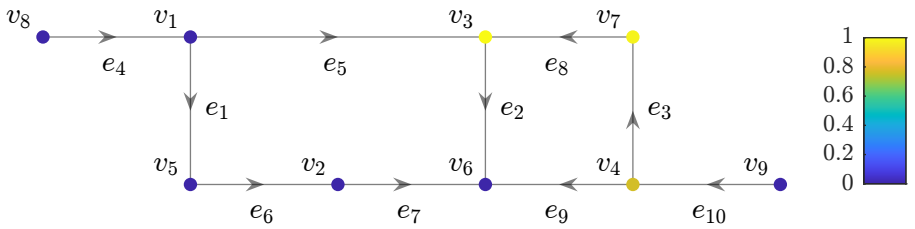


Figure A.32: Simulated values for $\bar{\psi}(t)$ for a leak at node 3: $d_{l,3}(t) = -0.1$ m^3/h

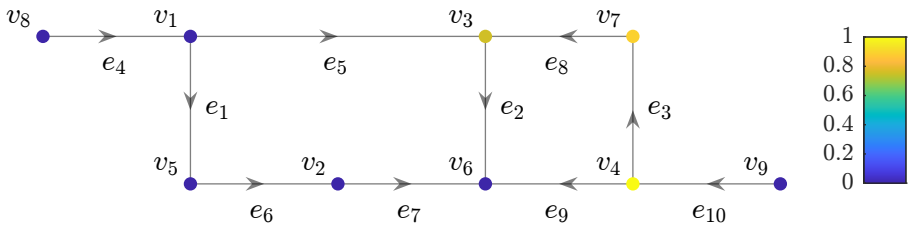


Figure A.33: Simulated values for $\bar{\psi}(t)$ for a leak at node 4: $d_{l,4}(t) = -0.1$ m^3/h

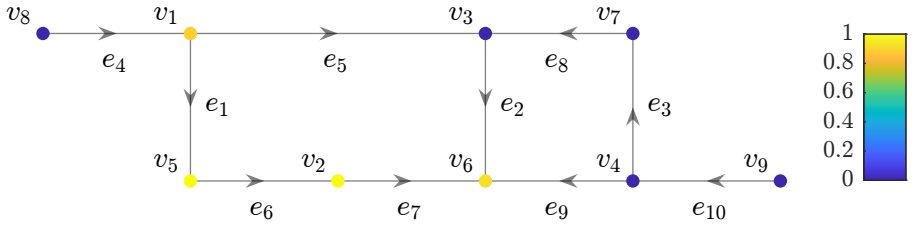


Figure A.34: Simulated values for $\bar{\psi}(t)$ for a leak at node 5: $d_{l,5}(t) = -0.1$ m^3/h

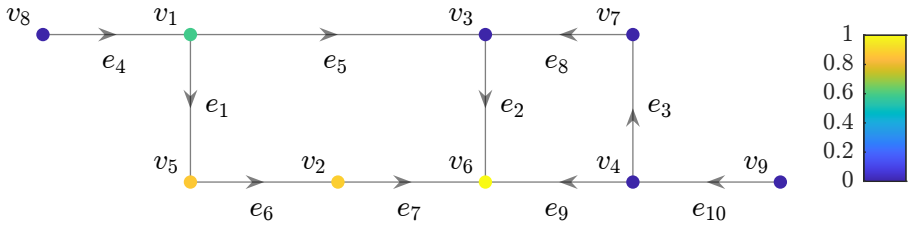


Figure A.35: Simulated values for $\bar{\psi}(t)$ for a leak at node 6: $d_{l,6}(t) = -0.1$ m^3/h

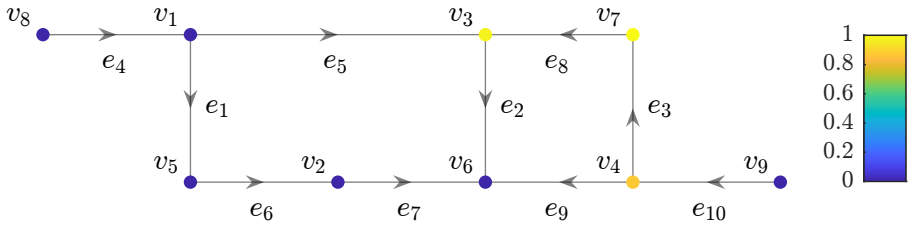


Figure A.36: Simulated values for $\bar{\psi}(t)$ for a leak at node 7: $d_{l,7}(t) = -0.1$ m^3/h

Leak node & size	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6	ψ_7
$d_{l,1} = -0.1$ m^3/h	**0.998**	0.867	-0.671	-0.995	0.883	0.591	-0.826
$d_{l,2} = -0.1$ m^3/h	0.878	**0.999**	-0.965	-0.893	0.997	0.933	-1.000
$d_{l,3} = -0.1$ m^3/h	-0.742	-0.961	**1.000**	0.764	-0.951	-0.991	0.979
$d_{l,4} = -0.1$ m^3/h	-0.998	-0.924	0.761	**1.000**	-0.936	-0.690	0.891
$d_{l,5} = -0.1$ m^3/h	0.890	**1.000**	-0.958	-0.904	0.998	0.923	-0.998
$d_{l,6} = -0.1$ m^3/h	0.590	0.883	-0.985	-0.616	0.868	**0.997**	-0.917
$d_{l,7} = -0.1$ m^3/h	-0.859	-0.996	0.974	0.875	-0.993	-0.946	**1.000**
No leakage	-0.780	-0.428	0.129	**0.759**	-0.458	-0.026	0.357

Table A.2: Simulatad values for the ψ vector for a leak of the size -0.1 m^3/h at node 1 to node 7. ψ_l for the estimated leak node ($\hat{l}$) is marked with bold letters. Further, the cells containing the three highest ψ_l's are marked with three shades of grey

A.11 Leakage localization of Randers district

As described in **Sec. 8.2** the leakage localization scheme is implemented in MATLAB and using a tool to extract data from the Randers EPANET model. The results of this implementation, a part from **Fig. 8.1**, is seen in **Fig. A.37** and **Fig. A.38**. Here, at leakage of 10 L/s is introduced for a period of 24 hours.

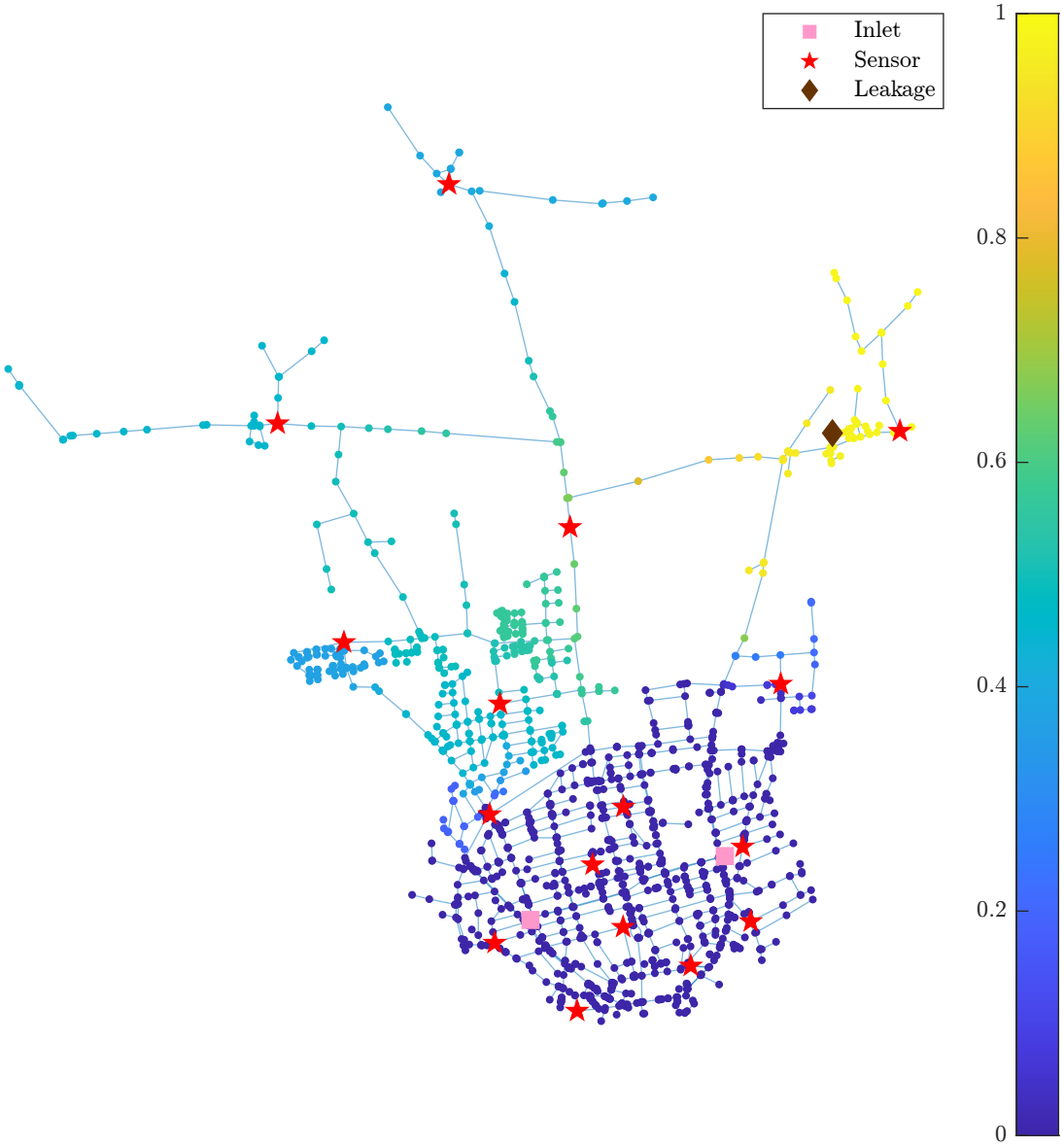


Figure A.37: Leakage localization applied to Randers district

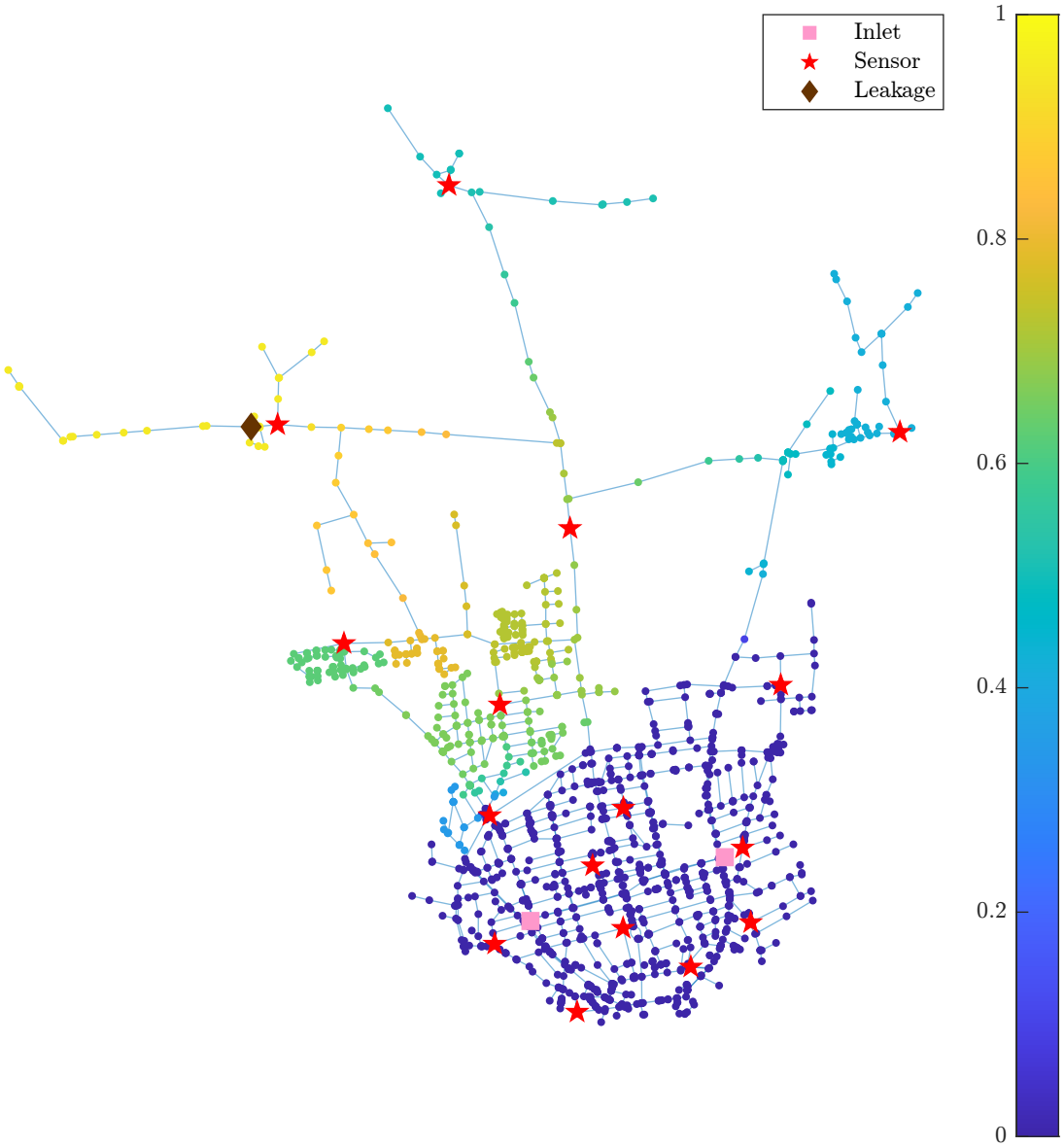


Figure A.38: Leakage localization applied to Randers district

A.12 SWIL interconnections