

Modelling and Networked Control of Water Distribution Networks

Worksheets

CA7 Group 733
Technical Faculty of IT and Design
Department of Control and Automation
Frederik Bajers Vej 7C
9220 Aalborg East
Email: es-21-ca-7-733@student.aau.dk

◆

CONTENTS

1	Introduction	4
2	Graph Theoretical Analysis	5
2.1	Basic Graph Theory	5
2.2	Simplified System	6
2.3	Detailed System	7
2.4	Implementation of Detailed System	9
3	WDN Component Models	10
3.1	Pipe Model	10
3.2	Valve Model	11
3.3	Pump Model	11
3.4	Elevated Water Reservoir Model	12
4	WDN System Model	12
4.1	Assumptions and Lemmas	12
4.2	Modelling an Open Hydraulic Network with an Elevated Reservoir	13
5	Control Structure	15
5.1	Control Structure	15
5.2	System Linearisation	15
5.3	Linearised Model	17
5.3.1	State Space Model - Pump Dynamics	17
5.3.2	State Space Model - Tank Pressure Dynamics	18
5.4	Influence of Pump Delay	18
5.5	The Root Locus Method	19
5.6	Design of Pump Controllers	19
5.6.1	Decentralised Control	19
5.6.2	Model Order Reduction	20
5.6.3	Modelling of Time Delay	21
5.6.4	Requirements for Control Performance	24
5.6.5	Choice of Controller Type	24
5.6.6	Root Locus Design	24
6	Optimal Control and the Linear-Quadratic Regulator	28
6.1	Basic Structure of an Optimal Control Problem	28
6.2	The Linear-Quadratic Control Problem	29
6.3	The Infinite-Horizon Linear-Quadratic Control Problem	29
6.4	Tracking LQR and Integral Action	30
6.5	The Velocity-Form LQR Algorithm	31
6.6	Disturbance-Accommodating Linear Quadratic Regulator for Exogenous Inputs	32
6.7	Implementation and Simulation Study	32
7	Disturbance Estimator	34
7.1	Estimator Objective	34
7.2	Estimator Model	34
7.3	Estimator Type	35
7.4	The Kalman Filter	35
7.4.1	Mathematical Formulation of the Kalman Filter in State Space	35
7.4.2	LTI Kalman Filter	37
7.4.3	Kalman Simulation	38
8	Modelling Network Effects	43
9	Experimental Setup and Results	45
10	Conclusion	50
References		50

11	Appendix	51
11.1	PumpCoefficients	51
12	Model Legend	52

1 INTRODUCTION

The following document details the work done by the CA 733 group at Aalborg University's M.Sc. Control and Automation program during the autumn of 2021. The work addresses modelling and control of a small-scale water distribution network (WDN), and addresses, among other things, cascaded control, velocity-form formulations of the classic Linear-Quadratic Regulator (LQR), root locus design, consumer demand estimation via Kalman filtering, and modelling of packet loss effects in geographically distributed networks.

The paper is structured as follows: In Section 2, we introduce the basic graph theory that is a prerequisite for modelling of WDNs. In Section 3 we then introduce models of the components that describe the behaviour of the edges in a graph-based WDN model. Development of the full model of the WDN then follows in Section 4. After developing the model, we detail our control structure and linearise the non-linear system model in Section 5. Here we also design the PI controllers that govern the inner loop. A review of optimal control then follows in Section 6, where we develop the velocity-form LQR (VF-LQR) used to control the tank level in the outer loop via an equivalent pressure reference. In Section 7, this is followed by a review of Kalman filtering in both its time-varying and LTI form, and a scheme for modelling consumer-induced disturbances by a truncated Fourier series. Finally, Section 8 models the stability impact of packet loss in a hypothesised control structure over a wireless medium, and Section 9 presents the experimental validation of the proposed control structure and the expected impact of packet loss.

2 GRAPH THEORETICAL ANALYSIS

This section describes the interconnected components in a water distributed network (WDN) using Graph Theory. We investigate three different graph networks; a simplified graph model not including pump and valve edges, one which includes pump and valve edges, and one which describes the implemented laboratory network.

2.1 Basic Graph Theory

A **directed graph** is a set of nodes $\{v_1, \dots, v_n\}$ and a set of edges $\{e_1, \dots, e_m\}$, with each edge associated with a node pair $\{v_i, v_j\}$ and an arrow pointing from v_i to v_j .

The **Incidence Matrix** H of a directed graph is defined as:

$$H_{i,j} = \begin{cases} 1 & \text{If the } j\text{th edge } s \text{ leaving the } i\text{th node} \\ -1 & \text{If the } j\text{th edge enters the } i\text{th node} \\ 0 & \text{If the } j\text{th edge is unconnected to the } i\text{th node} \end{cases}$$

$$H \in \mathbb{R}^{n \times m}$$

With n and m being the number of nodes and edges respectively.

The **Reduced Incidence Matrix** \bar{H} is obtained by choosing one of the nodes as reference and removing the corresponding row from H .

$$\bar{H} \in \mathbb{R}^{n-1 \times m}$$

The **spanning tree** is the connected graph but with no loops, i.e you cannot find a route around the graph where you start and end in the same node without entering a node more than once.

A **chord** is an edge which creates exactly one loop if added to the spanning tree.

A **loop** is a unique route along the edges of the graph, where all nodes are unique except the end node. In a loop the end node must also be the start node.

The **loop matrix** B is defined as:

$$B_{i,j} = \begin{cases} 1 & \text{If the direction of the } i\text{th loop and the } j\text{th edge agree} \\ -1 & \text{If the direction of the } i\text{th loop and the } j\text{th edge do not agree} \\ 0 & \text{If the } i\text{th loop does not include the } j\text{th edge} \end{cases}$$

But can also be calculated as shown below:

$$B = [I \quad -\bar{H}_C^T \cdot \bar{H}_T^{-T}] \tag{1}$$

With \bar{H}_C being the matrix containing only chord columns of the reduced incidence matrix. Similarly, \bar{H}_T contains only the non-chord columns.

$$B \in \mathbb{R}^{c \times m}$$

With c and m being the number of chords and edges respectively.

2.2 Simplified System

The graph network shown in Fig. 1 is a simplified graph network of the WDN laboratory implementation and has mainly been used to confirm basic graph theoretical concepts. This model does not include pump and valve edges.

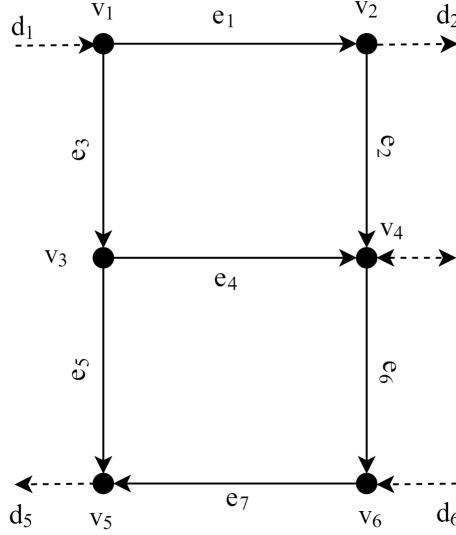


Fig. 1. Graph of simplified WDN network [1]

When applying the rules shown above for the simplified graph model of the WDN the incidence matrix becomes:

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \quad (2)$$

The reduced incidence matrix is formed by taking an arbitrary node as a reference, and removing the corresponding row from Eq. (2). We chose the 4th node, which results in the following reduced incidence matrix:

$$\bar{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 \end{bmatrix} \quad (3)$$

Chords and edges of the spanning tree are:

$$\begin{aligned} E_C &= \{e_1, e_4\} \\ E_T &= \{e_2, e_3, e_5, e_6, e_7\} \end{aligned} \quad (4)$$

$$B = \begin{bmatrix} 1 & 0 & 1 & -1 & -1 & 1 & 1 \\ 0 & 1 & 0 & 0 & -1 & 1 & 1 \end{bmatrix} \quad (5)$$

2.3 Detailed System

The graph network shown in Fig. 2 is a more detailed graphical representation of the WDN. This includes edges for pumps (e_1, e_{11}) and valve (e_3, e_9).

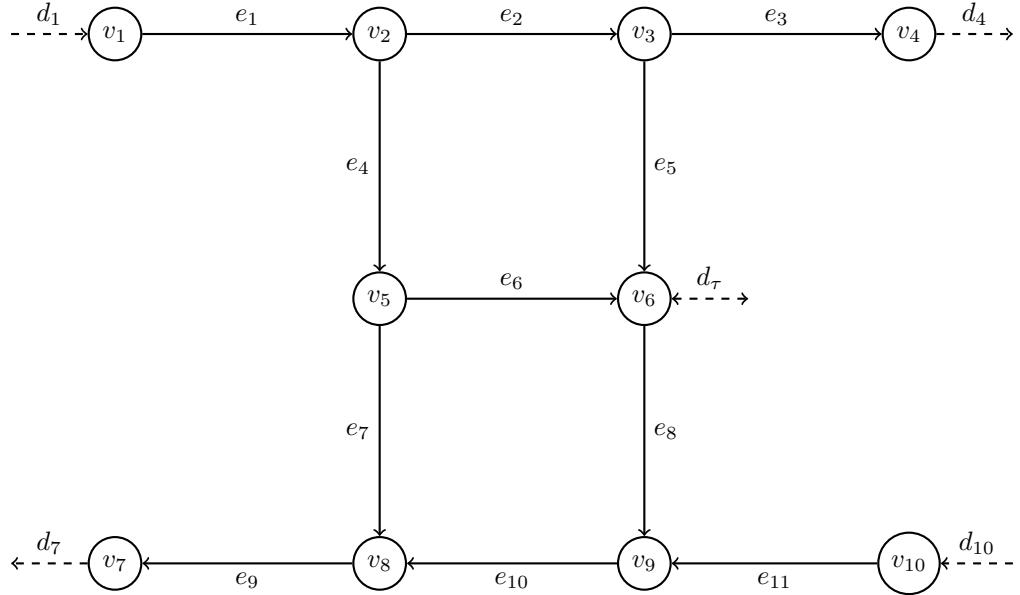


Fig. 2. Graph network of water distribution network

d_1 and d_{10} represent the flow into the system from the pumps. d_4 and d_7 represent the flow out of the system via the valves. d_τ represents the flow in and out of the tank.

The incidence matrix for the graph on Fig. 2 is shown below in Eq. (6).

$$H = \begin{bmatrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} & e_{11} \\ v_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_2 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_3 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_4 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_5 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ v_6 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 & 0 & 0 & 0 \\ v_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ v_8 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 & 0 \\ v_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 \\ v_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

The chords and the spanning tree are chosen from Fig. 2¹

$$\begin{aligned} E_C &= \{e_2, e_6\} \\ E_T &= \{e_1, e_3, e_4, e_5, e_7, e_8, e_9, e_{10}, e_{11}\} \end{aligned}$$

By choosing the 10th node as a reference, and removing it from the incidence matrix, we obtain:

$$\bar{H} = \begin{bmatrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} & e_{11} \\ v_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_2 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_3 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_4 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_5 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ v_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ v_8 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 & 0 \\ v_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 \end{bmatrix} \quad (7)$$

1. Note that you can obtain several spanning trees in this network by choosing different edges as chords

Furthermore we can define the open-node matrix F and tank matrix G :

$$F = \begin{matrix} d_1 & d_{f_1} & d_{f_2} & d_{f_3} & d_{f_4} \\ d_2 & 1 & 0 & 0 & 0 \\ d_3 & 0 & 0 & 0 & 0 \\ d_4 & 0 & 1 & 0 & 0 \\ d_5 & 0 & 0 & 0 & 0 \\ d_6 & 0 & 0 & 0 & 0 \\ d_7 & 0 & 0 & 1 & 0 \\ d_8 & 0 & 0 & 0 & 0 \\ d_9 & 0 & 0 & 0 & 0 \\ d_{10} & 0 & 0 & 0 & 1 \end{matrix}, \quad G = \begin{matrix} d_1 & d_{\tau_1} \\ d_2 & 0 \\ d_3 & 0 \\ d_4 & 0 \\ d_5 & 0 \\ d_6 & 1 \\ d_7 & 0 \\ d_8 & 0 \\ d_9 & 0 \\ d_{10} & 0 \end{matrix} \quad (8)$$

with their reference-respective equivalents given by:

$$\bar{F} = F \setminus F_{10*} \wedge \bar{G} = G \setminus G_{10*} \quad (9)$$

where the notation X_{10*} denotes the entire 10th row² of the matrix X and \setminus is the set relative complement operator. These matrices map demands at their respective nodes into the vector of total demands in the system $d \vee \bar{d}$.

2. Correspondingly, X_{*10} would denote the entire 10th column

2.4 Implementation of Detailed System

It is not possible to implement the WDN model shown in Fig. 2 in the laboratory due to the fact that we don't have access to a 4-way splitter. Therefore, the graph model is altered to fit a model that can be implemented. The graph is shown in Fig. 3.

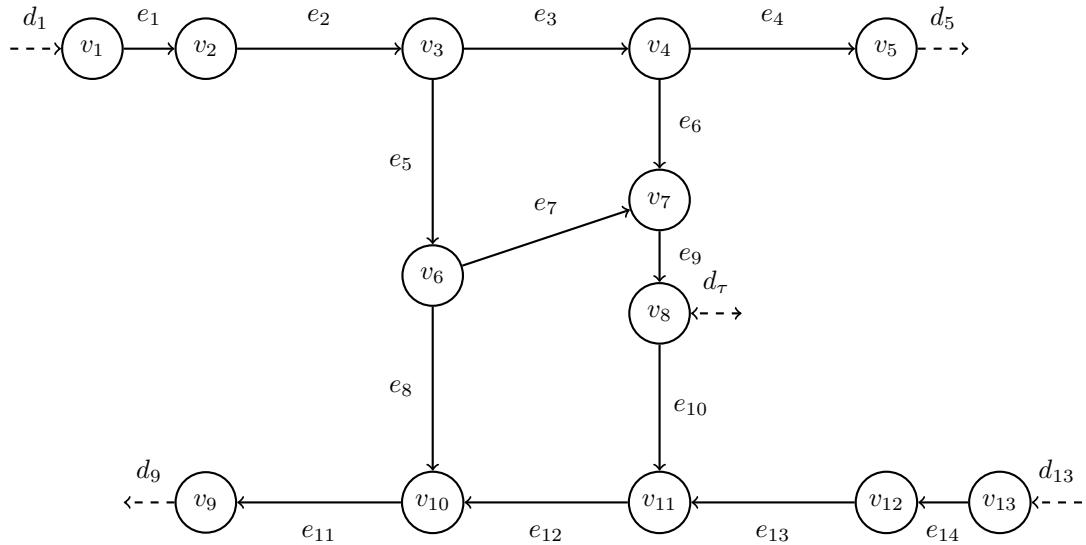


Fig. 3. Graph network of implemented water distribution network.

The incidence matrix becomes:

The reduced incidence matrix \bar{H} for the system given by Eq. (10) is obtained by choosing the node 13 as reference:

The B matrix is obtained by choosing e_3 and e_7 as chords³:

$$B = \begin{matrix} B_1 \\ B_2 \end{matrix} \left[\begin{array}{cccccccccccccccc} e_3 & e_7 & e_1 & e_2 & e_4 & e_5 & e_6 & e_8 & e_9 & e_{10} & e_{11} & e_{12} & e_{13} & e_{14} \\ 1 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 1 & 0 & 0 \end{array} \right] \quad (12)$$

3 WDN COMPONENT MODELS

All components (eg. pipes, valves and pumps) can be described by two variables, namely the flow through the component and the differential pressure across the component:

$$\begin{bmatrix} \Delta p_k \\ q_k \end{bmatrix} = \begin{bmatrix} p_i - p_j \\ q_k \end{bmatrix} \quad (13)$$

The following section will examine these two variables for pipes, valves and pumps. We source the equations from [2].

3.1 Pipe Model

The differential pressure across a pipe can be modelled as follows:

$$\Delta p_k = J_k \cdot \dot{q}_k + \lambda_k(q_k) - \Delta z_k \quad (14)$$

Δp_k	The differential pressure across the k th component	[Pa]
J_k	The mass inertia of the water in the k th pipe	[kg/m ⁴]
q_k	is the flow of water through the k th pipe	[m ³ /s]
$\lambda_k(q_k)$	is the drop in pressure due to friction in the k th pipe	[Pa]
Δz_k	is the drop in pressure due to geodesic level	[Pa]

The mass inertia of water can be described as follows:

$$J = \frac{L \cdot \rho}{A} \quad (15)$$

L	is the length of the pipe	[m]
ρ	is the density of water	[kg/m ³]
A	is the cross sectional area of water	[m ²]

The cross-sectional area of the pipe is assumed to be constant along the pipe.

The causes of flow friction $\lambda_k(q_k)$ are surface resistance h_f and form resistance h_m . The surface resistance can be described with the Darcy-Weisbach equation:

$$h_f = f \cdot \frac{8 \cdot L \cdot q^2}{\pi^2 \cdot g \cdot D^5} \quad (16)$$

h_f	is the head loss from surface resistance	[m]
f	is the pipe friction factor	[·]
D	is the pipe diameter	[m]
g	is the gravitational constant	[m/s ²]

Under the assumption of turbulent flow f is given by:

$$f = 1.325 \cdot \left(\ln \left(\frac{\epsilon}{3.7 \cdot D} + \frac{5.74}{R^{0.9}} \right) \right)^{-2} \quad (17)$$

ϵ	average roughness height in the pipe	[m]
R	is Reynolds number - for turbulent flow $R \geq 4000$	[·]

The form resistance is given by the following:

$$h_m = k_f \cdot \frac{8 \cdot q^2}{\pi^2 \cdot g \cdot D^4} \quad (18)$$

h_m	is the head loss from form resistances	[m]
k_f	is coefficient form loss	[·]

3. Again we note that it is possible to obtain more than one unique spanning tree by choosing different chords.

The drop in pressure due to geodesic level difference is given by:

$$\Delta z_k = \rho \cdot g \cdot \Delta h_k \quad (19)$$

Δh_k is the level difference across the terminals of the k th [m] pipe

Having explained all the components of (14) a complete expression can now be formulated; we take advantage of the fact that resistance losses are expressed in terms of head, and can be expressed in terms of pressure by multiplying by ρ and g .

$$p = \rho \cdot g \cdot h \quad (20)$$

Meaning that the head losses can be expressed in terms of pressure by the following:

$$\lambda_k(q_k) = \left(f \cdot \frac{8 \cdot L \cdot q^2}{\pi^2 \cdot g \cdot D^5} + k_f \cdot \frac{8 \cdot q^2}{\pi^2 \cdot g \cdot D^4} \right) \cdot g \cdot \rho \quad (21)$$

Inserting and reducing into Eq. (14)

$$\Delta p_k = \frac{L \cdot \rho}{A} \cdot \dot{q}_k + \left(f \cdot \frac{8 \cdot L \cdot \rho}{\pi^2 \cdot D^5} + k_f \cdot \frac{8 \cdot \rho}{\pi^2 \cdot D^4} \right) \cdot |q_k| \cdot q_k - \rho \cdot g \cdot \Delta h_k = \mathcal{J}\dot{q} + \lambda(q) + \Delta z \quad (22)$$

The absolute value of one of the flow components in q^2 is taken to preserve the flow direction.

3.2 Valve Model

While the head loss of a valve *can* be explained in terms of its form resistance, as we have done previously in Section 3.1, it is generally impractical to determine the form loss coefficient k_f for valves. Instead, an expression for the head loss can be derived - typically by the manufacturer - via a conductivity function $K_{valve}(OD)$.

Recalling that the pressure drop across a valve is proportional to its squared flow, we may express that:

$$\frac{\Delta p_1}{q_1^2} = \frac{\Delta p_2}{q_2^2} \Leftrightarrow \frac{\Delta p_1}{\Delta p_2} = \frac{q_1^2}{q_2^2} \Leftrightarrow q_1 = q_2 \cdot \sqrt{\frac{\Delta p_1}{\Delta p_2}} \quad (23)$$

We then express the conductivity function $K_{valve}(OD)$, by convention, as corresponding to the flow q_n at a given opening degree that produces exactly a pressure differential of 1 bar, i.e.:

$$q = q_n(OD) \cdot \sqrt{\frac{\Delta p_1}{1}} = K_{valve}(OD) \cdot \sqrt{\Delta p_1} \quad (24)$$

We can then express the pressure differential across the valve for a given flow as:

$$q = K_{valve}(OD) \cdot \sqrt{\Delta p_1} \Leftrightarrow \Delta p_1 = \frac{1}{K_{valve}(OD)^2} \cdot |q| \cdot q = \mu(q, OD) \quad (25)$$

where $q^2 \equiv |q| \cdot q$ is introduced to preserve the directionality of flow. Note that Eq. (25) implies the pressure differential across the pipe varies with both flow rate *and* opening degree.

$K_{valve}(OD)$ may take a variety of forms, but is typically either linear, equal-percentage, or quick-opening.

3.3 Pump Model

The pressure differential across a centrifugal pump, such as the pumps used in this project, is a multivariable function that can generally be approximated by a polynomial expression of the form:

$$\Delta p = a_0 \cdot \omega^2 + a_1 \cdot \omega \cdot q - a_2 \cdot |q| \cdot q \quad (26)$$

where $[a_0, a_1, a_2]$ is a tuple of coefficients that describe the pump's characteristic curve, q is the flow rate through the pump, and ω is the rotational velocity of the pump.

3.4 Elevated Water Reservoir Model

In this section we will develop a hydraulic model of an elevated fluid reservoir, also known as a tank. Under assumption that the cross-sectional area of the tank is constant along its height axis, the pressure at the bottom of the tank is proportional to the fluid level in the tank:

$$p_\tau \propto \zeta \quad (27)$$

The volumetric rate of change is equal to the flow in and out of the tank:

$$\dot{V} = d_\tau \quad (28)$$

Logically, the rate of fluid level change is proportional to the volumetric rate of change:

$$\dot{\zeta} \propto \dot{V} \Leftrightarrow \dot{\zeta} \propto d_\tau \quad (29)$$

From the above it can be concluded that the rate of change of pressure is proportional to flow in and out of the tank:

$$\dot{p}_\tau \propto d_\tau \quad (30)$$

Defining the proportionality constant τ and taking flow out of the tank as positive:

$$\dot{p}_\tau = -\tau \cdot d_\tau \quad (31)$$

p_τ	is the pressure at the node connected to the bottom of the tank	[Pa]
τ	is the tank parameter which depends on the cross sectional area of the tank	[Pa/m ³]
d_τ	is the water flow in and out of the tank. If $q > 0$ water is flowing out of the tank, if $q < 0$ water is flowing into the tank.	[m ³ /s]

The tank parameter is given by:

$$\tau = \rho \cdot g \frac{1}{A} \quad (32)$$

4 WDN SYSTEM MODEL

In this section we will develop the model of the water distribution network, based on the graph theory presented in Section 2.

4.1 Assumptions and Lemmas

In modelling the dynamics of an open hydraulic network, we will make a few basic physical assumptions. Assuming a network with topology described by the incidence matrix H and fundamental loop matrix B , subject to the vector of *nodal* flows $d \in \mathbb{R}^n$, and containing the *edge* flows $q \in \mathbb{R}^m$. We will assume that this system obeys Kirchoff's Nodal Law (KNL), i.e. that:

$$Hq = d \quad (33)$$

We also assume that the system obeys Kirchoff's Mesh Law (KML), i.e.:

$$B\Delta p = BH^T p = 0 \wedge B\Delta h = BH^T h = 0 \quad (34)$$

Furthermore, we assume that the system is conservative with respect to mass, which implies that there can at most be $n - 1$ independent nodal demands, i.e.:

$$d_n = - \sum_{i=1}^{n-1} d_i \quad (35)$$

We will also define a set of graph-theoretical lemmas for convenience, referring to [3] for proof of both:

Lemma 4.1. Let H_T be the incidence matrix partition corresponding to the spanning tree of the graph described by H , and let \bar{H}_T be its equivalent with respect to the reduced incidence matrix \bar{H} . \bar{H}_T is invertible since a tree is a connected graph with $n - 1$ edges [4], and the following holds

$$H_T \bar{H}_T^{-1} = \begin{bmatrix} I_{n-1} \\ \mathbf{1}^T \end{bmatrix} \quad (36)$$

where $\mathbf{1}$ is a vector of ones and $I_{n-1} \in \mathbb{R}^{n-1 \times n-1}$ is an identity matrix.

Lemma 4.2. Let $q \in \mathbb{R}^m$ be the edge flows of a graph with reduced incidence matrix \bar{H} and n nodes. Denote its spanning tree by T , let $q_c \in \mathbb{R}^c$, $c = m - n + 1$ be the graph's chordal flows, and denote the tree partition of the reduced incidence matrix \bar{H}_T . Finally, let B be the graph's fundamental loop matrix, and define $\bar{d} \in \mathbb{R}^{n-1}$ as the vector of non-reference nodal flows. Then the following is true:

$$q = B^T q_c + \begin{bmatrix} 0_{C \times n-1} \\ \bar{H}_T^{-1} \end{bmatrix} \bar{d} \quad (37)$$

We are now ready to start modelling the system.

4.2 Modelling an Open Hydraulic Network with an Elevated Reservoir

We assume a network topology with n nodes and m edges, of which e nodes are open to the atmosphere. By definition, water can only flow in and out of the network at nodes which are open to the atmosphere, implying that demand at all other nodes is zero. The flows at these open nodes d_f can then be mapped into the vector of total flows \bar{d} by:

$$\bar{d} = \bar{F} d_f, \quad \bar{F} \in \mathbb{R}^{n-1 \times e} \quad (38)$$

with the remaining nodal demand at the reference node comprising the dependent flow. The edge flows can then be expressed, using Theorem 4.2 to re-write KNL Eq. (33), as:

$$q = B^T q_C + \begin{bmatrix} 0 \\ \bar{H}_T^{-1} \end{bmatrix} \bar{F} d_f \quad (39)$$

We can modify this to account for tanks by introducing an additional vector of non-zero demand that is *not* open to the atmosphere, denoting it $d_\tau \in \mathbb{R}^\tau$. We can then restate Eq. (38) as:

$$\bar{d} = \bar{F} d_f + \bar{G} d_\tau, \quad \bar{F} \in \mathbb{R}^{e \times n-1} \wedge \bar{G} \in \mathbb{R}^{\tau \times n-1} \quad (40)$$

subsequently allowing Eq. (39) to be restated as:

$$q = B^T q_C + \begin{bmatrix} 0 \\ \bar{H}_T^{-1} \end{bmatrix} (\bar{F} d_f + \bar{G} d_\tau) \quad (41)$$

We can now state the pressure drop across each edge in the network as:

$$\Delta p = \mathcal{J} \dot{q} + \lambda(q) + \mu(q, OD) + \alpha(q, \omega) - \Delta h \quad (42)$$

where $\mathcal{J}, \lambda, \mu, \alpha$ are as defined in Section 3⁴ and q is the vector of edge flows. We now invoke KML as per Eq. (34) on Eq. (42), obtaining:

$$\begin{aligned} 0 &= B \mathcal{J} \dot{q} + B(\lambda(q) + \mu(q) + \alpha(q)) - B \Delta h \\ &= B \mathcal{J} \dot{q} + B(\lambda(q) + \mu(q) + \alpha(q)) \\ &\Rightarrow \\ B \mathcal{J} \dot{q} &= -B(\lambda(q) + \mu(q) + \alpha(q)) \end{aligned} \quad (43)$$

We can then invoke KNL on the left-hand side of Eq. (43), resulting in:

$$B \mathcal{J} B^T \dot{q}_C + B \mathcal{J} \begin{bmatrix} 0 \\ \bar{H}_T^{-1} \end{bmatrix} (\bar{F} d_f + \bar{G} d_\tau) = -B(\lambda(q) + \mu(q) + \alpha(q)) \quad (44)$$

We now develop this expression further via the definition of B in Eq. (1) and via a chord/tree-partition of \mathcal{J} such that:

$$B \mathcal{J} B^T \dot{q}_C + [I_C \quad -\bar{H}_C^T \bar{H}_T^{-T}] \begin{bmatrix} \mathcal{J}_C & 0 \\ 0 & \mathcal{J}_T \end{bmatrix} \begin{bmatrix} 0 \\ \bar{H}_T^{-1} \end{bmatrix} (\bar{F} d_f + \bar{G} d_\tau) = -B(\lambda(q) + \mu(q) + \alpha(q)) \quad (45)$$

This simplifies to:

$$B \mathcal{J} B^T \dot{q}_C - \bar{H}_C^T \bar{H}_T^{-T} \mathcal{J}_T \bar{H}_T^{-1} \bar{F} d_f - \bar{H}_C^T \bar{H}_T^{-T} \mathcal{J}_T \bar{H}_T^{-1} \bar{G} d_\tau = -B(\lambda(q) + \mu(q) + \alpha(q)) \quad (46)$$

We now recall that we have chosen a reference node that is open to atmosphere at $p_{ref} = p_0$, and that Δp can be rewritten as:

$$\Delta p = \begin{bmatrix} \Delta p_C \\ \Delta p_T \end{bmatrix} = \begin{bmatrix} H_C^T \\ H_T^T \end{bmatrix} p \quad (47)$$

4. Note that we have omitted the secondary variables OD, ω for convenience.

It is then, given that an arbitrary node can be reached by pathing solely through the spanning tree, axiomatically true per KML that:

$$H_T^T p = H_T^T \begin{bmatrix} \bar{p} \\ p_0 \end{bmatrix} = \mathcal{J}_T \dot{q}_T + (\lambda_T(q_T) + \mu_T(q_T) + \alpha_T(q_T)) - H_T^T \begin{bmatrix} \bar{h} \\ h_0 \end{bmatrix} \quad (48)$$

We now exploit that:

$$H_T \bar{H}_T^{-1} = \bar{H}_T^{-T} H_T^T \quad (49)$$

which allows us to rewrite Eq. (48) by left-multiplying every term with \bar{H}_T^{-T} and exploiting Theorem 4.1:

$$\bar{p} - \mathbf{1} p_0 = \bar{H}_T^{-T} \mathcal{J}_T \dot{q}_T + \bar{H}_T^{-T} (\lambda_T(q_T) + \mu_T(q_T) + \alpha_T(q_T)) - (\bar{h} - \mathbf{1} h_0) \quad (50)$$

Recalling that \bar{F} maps the open node demands into the total demand, we can then express the following:

$$0 = \bar{F}^T (\bar{p} - \mathbf{1} p_0) = \bar{F}^T \left(\bar{H}_T^{-T} \mathcal{J}_T \dot{q}_T + \bar{H}_T^{-T} (\lambda_T(q_T) + \mu_T(q_T) + \alpha_T(q_T)) \right) - \bar{F}^T (\bar{h} - \mathbf{1} h_0) \quad (51)$$

which implies that:

$$\bar{F}^T \bar{H}_T^{-T} \mathcal{J}_T \dot{q}_T = -\bar{F}^T \left(\bar{H}_T^{-T} (\lambda_T(q_T) + \mu_T(q_T) + \alpha_T(q_T)) \right) + \bar{F}^T (\bar{h} - \mathbf{1} h_0) \quad (52)$$

and finally by restating Eq. (33) as:

$$\begin{aligned} Hq &= d \Rightarrow \\ \bar{H}q &= \bar{d} \Rightarrow \\ \bar{H}_C q_C + \bar{H}_T q_T &= \bar{d} \Rightarrow \\ \bar{H}_T q_T &= -\bar{H}_C q_C + \bar{d} \Rightarrow \\ q_T &= \bar{H}_T^{-1} (-\bar{H}_C q_C + \bar{d}) \end{aligned} \quad (53)$$

and applying Eq. (37) we can rewrite Eq. (52) as:

$$\begin{aligned} \bar{F}^T \bar{H}_T^{-T} \mathcal{J}_T \dot{q}_T &= -\bar{F}^T \bar{H}_T^{-T} \mathcal{J}_T \bar{H}_T^{-1} \bar{H}_C \dot{q}_C + \bar{F}^T \bar{H}_T^{-T} \mathcal{J}_T \bar{H}_T^{-1} \bar{F} \dot{d}_f + \bar{F}^T \bar{H}_T^{-T} \mathcal{J}_T \bar{H}_T^{-1} \bar{G} \dot{d}_\tau \\ &= -\bar{F}^T \bar{H}_T^{-T} (\lambda_T(q_T) + \mu_T(q_T) + \alpha_T(q_T)) + \bar{F}^T (\bar{h} - \mathbf{1} h_0) \end{aligned} \quad (54)$$

By a completely analogous procedure we can extract the tank-connected non-reference nodes as in Eq. (52) via \bar{G} , except the left-hand side is now $p_\tau - \mathbf{1} p_0$:

$$\bar{G}^T \bar{H}_T^{-T} \mathcal{J}_T \dot{q}_T = -\bar{G}^T \bar{H}_T^{-T} (\lambda_T(q_T) + \mu_T(q_T) + \alpha_T(q_T)) + \bar{G}^T (\bar{h} - \mathbf{1} h_0) + (\bar{p}_\tau - \mathbf{1} p_0) \quad (55)$$

which analogously to Eq. (54) becomes:

$$\begin{aligned} \bar{G}^T \bar{H}_T^{-T} \mathcal{J}_T \dot{q}_T &= -\bar{G}^T \bar{H}_T^{-T} \mathcal{J}_T \bar{H}_T^{-1} \bar{H}_C \dot{q}_C + \bar{G}^T \bar{H}_T^{-T} \mathcal{J}_T \bar{H}_T^{-1} \bar{F} \dot{d}_f + \bar{G}^T \bar{H}_T^{-T} \mathcal{J}_T \bar{H}_T^{-1} \bar{G} \dot{d}_\tau \\ &= -\bar{G}^T \bar{H}_T^{-T} (\lambda_T(q_T) + \mu_T(q_T) + \alpha_T(q_T)) + \bar{G}^T (\bar{h} - \mathbf{1} h_0) + (\bar{p}_\tau - \mathbf{1} p_0) \end{aligned} \quad (56)$$

Now, defining the state vector $q_n = [q_C^T \quad d_f^T \quad d_\tau^T]^T$ and re-introducing the actuator variables OD, ω , we can finally collect the full dynamics of the network according to Eqs. (46), (52) and (55) as a nonlinear differential equation:

$$\Phi \mathcal{J} \Phi^T \dot{q} = -\Phi \left(\lambda(q_n) + \mu(q_n, OD) + \alpha(q_n, \omega) \right) + \Psi(\bar{h} - \mathbf{1} h_0) + \mathcal{I}(p_\tau - \mathbf{1} p_0) \quad (57)$$

where p_τ evolves according to:

$$\dot{p}_\tau = -\mathcal{T} \dot{d}_\tau, \quad \mathcal{T} = \text{diag}(\tau_i) \quad (58)$$

and the matrices Φ, Ψ, \mathcal{I} are defined as:

$$\Phi \triangleq \begin{bmatrix} I & -\bar{H}_C^T \bar{H}_T^{-T} \\ 0 & \bar{F}^T \bar{H}_T^{-T} \\ 0 & \bar{G}^T \bar{H}_T^{-T} \end{bmatrix}, \quad \Psi \triangleq \begin{bmatrix} 0 \\ \bar{F}^T \\ \bar{G}^T \end{bmatrix}, \quad \mathcal{I} \triangleq \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \quad (59)$$

5 CONTROL STRUCTURE

This section is concerned with the control structure of the WDN.

5.1 Control Structure

This section documents the structure of the control system for this project as seen in Fig. 4. The intended control structure assumes that the dynamics of the system can be partitioned in two, the fast dynamics of the inertia of the pipes (assuming the pump dynamics are infinitely fast) and the slow dynamics of the tank level.

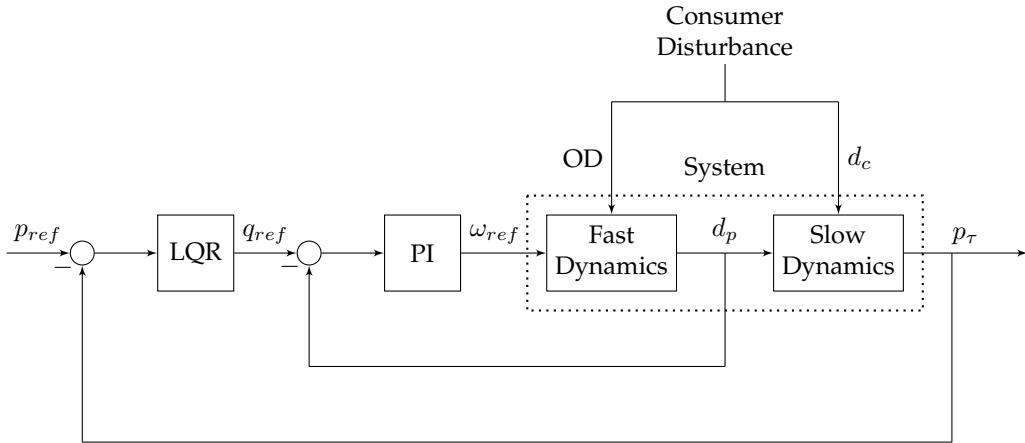


Fig. 4. Control Structure

5.2 System Linearisation

Before the model presented in Eq. (57) is truly useful to us - at least within the scope of the *linear* control strategies considered in this project - we must find a way to turn it into a linear model. The typical approach to this problem is *linearisation*, whereby we exploit the extremely powerful Hartman-Grobman theorem, which we present roughly as outlined in [5]:

Theorem 5.1. (The Hartman-Grobman Theorem) Let E be an open subset of \mathbb{R}^n containing the origin, and let f be a continuously differentiable function on E :

$$f \in C^1(E)$$

Let γ_t be the flow of the nonlinear system $\dot{x} = f(x)$. Assume furthermore that there exists an equilibrium point at the origin:

$$f(0) = 0$$

and that this equilibrium point is hyperbolic:

$$\forall \lambda \in T(A) : \operatorname{Re}(\lambda) \neq 0, \quad A = \nabla f$$

where T is the eigenspace of A . Then there exists a homeomorphism H of some open set U , $0 \in U$ onto the open set V , $0 \in V$, such that $\forall x_0 \in U$, there is an open interval $I_0 \subset \mathbb{R}$, $0 \in I_0$ such that:

$$\forall x_0 \in U \wedge \forall t \in I_0 : H \circ \gamma_t(x_0) = e^{At} H(x_0)$$

At first glance, this theorem looks opaquely mathematical and not immediately applicable. However, in practice, Theorem 5.1 simply tells us that in the immediate vicinity of some hyperbolic equilibrium point of our non-linear system, there exists a *linear* system whose eigenfunctions describe its evolutions exactly. We note that it is not *necessary* to linearise at a hyperbolic equilibrium, but doing so is favourable when possible as it precludes the presence of a center manifold, whose dynamics cannot be captured by linearisation.

Recalling that the first-order Taylor series of a function at a point can be thought of as a generalisation of its tangent line, it is then possible to identify the linearisation of our system via:

$$\dot{x} \approx f(x_0) + \nabla f \Big|_{x_0} (x - x_0) \tag{60}$$

We now revert our attention to the non-linear model of the WDN. We will make the simplifying assumption that $\Phi\mathcal{J}\Phi^T$ is invertible, which is not generally true, but tends to hold for the type of WDN in question. We also introduce the notation $\mathcal{P} : (\Phi\mathcal{J}\Phi^T)^{-1}$, allowing us to rewrite Eq. (57) as:

$$\begin{aligned}\dot{q}_n &= -\mathcal{P}\Phi\left(\lambda(q_n) + \mu(q_n, OD) + \alpha(q_n, \omega)\right) + \mathcal{P}\left(\Psi(\bar{h} - \mathbf{1}h_0) + \mathcal{I}(p_\tau - \mathbf{1}p_0)\right) \\ &= -\mathcal{P}\Phi\left(\lambda(q_n) + \frac{|q_n|q_n}{(K_v \cdot OD)^2} + \alpha(q_n, \omega)\right) + \mathcal{P}\left(\Psi(\bar{h} - \mathbf{1}h_0) + \mathcal{I}(p_\tau - \mathbf{1}p_0)\right) \\ &= -\mathcal{P}\Phi\left(K_\lambda|q_n|q_n + \frac{|q_n|q_n}{(K_v \cdot OD)^2} + a_0\omega^2 + a_1\omega q + a_2|q|q\right) + \mathcal{P}\left(\Psi(\bar{h} - \mathbf{1}h_0) + \mathcal{I}(p_\tau - \mathbf{1}p_0)\right)\end{aligned}\quad (61)$$

We now make the additional observation that $\Psi(\bar{h} - \mathbf{1}h_0)$ does not depend on $\{q_n, OD, \omega, p_\tau\}$. This suggests that, when computing the Taylor expansion, this term disappears under the action of the ∇ operator, i.e. that:

$$\begin{aligned}\nabla \dot{q}_n &= \nabla\left(-\mathcal{P}\Phi\left(\lambda(q_n) + \mu(q_n, OD) + \alpha(q_n, \omega)\right) + \mathcal{P}\left(\Psi(\bar{h} - \mathbf{1}h_0) + \mathcal{I}(p_\tau - \mathbf{1}p_0)\right)\right) \\ &= \nabla\left(-\mathcal{P}\Phi\left(\lambda(q_n) + \mu(q_n, OD) + \alpha(q_n, \omega)\right)\right) + \nabla\mathcal{P}\left(\mathcal{I}(p_\tau - \mathbf{1}p_0)\right) \\ &= \nabla\left(-\mathcal{P}\Phi\left(K_\lambda|q_n|q_n + \frac{|q_n|q_n}{(K_v \cdot OD)^2} + a_0\omega^2 + a_1\omega q + a_2|q|q\right)\right) + \nabla\mathcal{P}\left(\mathcal{I}(p_\tau - \mathbf{1}p_0)\right)\end{aligned}\quad (62)$$

Recognizing furthermore that Φ and \mathcal{J} are simply linear transformations, the linearity of differentiation then allows us to write a general expression for the Taylor expansion of Eq. (61) as:

$$\begin{aligned}\dot{q}_n &\approx f(x_0) + \frac{\partial f}{\partial q_n}\Big|_{x_0} \tilde{q}_n + \frac{\partial f}{\partial OD}\Big|_{x_0} \tilde{OD} + \frac{\partial f}{\partial \omega}\Big|_{x_0} \tilde{\omega} + \frac{\partial f}{\partial p_\tau}\Big|_{x_0} \tilde{p}_\tau \\ &= f(x_0) + \mathcal{P}\left(-\Phi\left(\frac{\partial \Omega}{\partial q_n}\Big|_{x_0} \tilde{q}_n + \frac{\partial \Omega}{\partial OD}\Big|_{x_0} \tilde{OD} + \frac{\partial \Omega}{\partial \omega}\Big|_{x_0} \tilde{\omega}\right) + \mathcal{I}\frac{\partial f}{\partial p_\tau}\Big|_{x_0} \tilde{p}_\tau\right)\end{aligned}\quad (63)$$

where:

$$x_0 = \{q_0, OD_0, \omega_0, p_\tau\} \quad (64)$$

$$\Omega = K_\lambda|q_n|q_n + \frac{|q_n|q_n}{(K_v \cdot OD)^2} + a_0\omega^2 + a_1\omega q_n + a_2|q_n|q_n \quad (65)$$

$$\tilde{q}_n = q_n - q_0 \quad (66)$$

$$\tilde{OD} = OD - OD_0 \quad (67)$$

$$\tilde{\omega} = \omega - \omega_0 \quad (68)$$

$$\tilde{p}_\tau = p_\tau - p_{\tau_0} \quad (69)$$

Writing out each of the partial derivatives in Eq. (63), we get:

$$\frac{\partial \Omega}{\partial q_n}\Big|_{x_0} = a_1\omega_0 + \left(|q_0| + \text{sign}(q_0)q_0\right)\left(K_\lambda + a_2 + \frac{1}{(K_v OD_0)^2}\right) \quad (70)$$

$$\frac{\partial \Omega}{\partial OD}\Big|_{x_0} = -|q_0|q_0 \frac{2}{K_v^2 OD_0^3} \quad (71)$$

$$\frac{\partial \Omega}{\partial \omega}\Big|_{x_0} = a_1q_0 + 2a_0\omega_0 \quad (72)$$

$$\frac{\partial \Omega}{\partial p_\tau}\Big|_{x_0} = 1 \quad (73)$$

and the complete Taylor expansion becomes:

$$\begin{aligned}
\dot{q}_n \approx & f(x_0) - \mathcal{P}\Phi \left(a_1\omega_0 + (|q_0| + \text{sign}(q_0)q_0) \left(K_\lambda + a_2 + \frac{1}{(K_v OD_0)^2} \right) \tilde{q}_n \right) \\
& - \mathcal{P}\Phi \left(\left(-|q_0|q_0 \frac{2}{K_v^2 OD_0^3} \right) \tilde{OD} \right) \\
& - \mathcal{P}\Phi \left((a_1 q_0 + 2a_0 \omega_0) \tilde{\omega} \right) \\
& + \mathcal{P}\mathcal{I}\tilde{p}_\tau
\end{aligned} \tag{74}$$

We will now make three additional abstractions to simplify Eq. (74). We first exploit the trivial condition that, when linearizing at an equilibrium point, $f(x_0) \equiv 0$. We then consider the fact that, in practice, OD and p_τ vary very slowly compared to q_n and ω . It is therefore a reasonable assumption that at steady state for the latter variables, the error introduced by neglecting $\frac{\partial \Omega}{\partial OD}$ is:

$$\mathcal{P}\Phi \left(|\tilde{q}_{ss}| \tilde{q}_{ss} \frac{2}{K_v^2 OD_0^3} \tilde{OD} \right) \tag{75}$$

which, under the assumption that q_n changes much faster than OD , is a constant offset and therefore removable by integral action. Analogously, the error induced by ignoring $\frac{\partial \Omega}{\partial p_\tau}$ is:

$$\mathcal{P}\mathcal{I}\tilde{p}_\tau \tag{76}$$

which is likewise a constant term that may be removed by integral action. Thus, we can restate a greatly simplified form of Eq. (74) as:

$$\dot{q}_n \approx -\mathcal{P}\Phi \left(a_1\omega_0 + (|q_0| + \text{sign}(q_0)q_0) \left(K_\lambda + a_2 + \frac{1}{(K_v OD_0)^2} \right) \tilde{q}_n \right) - \mathcal{P}\Phi \left((a_1 q_0 + 2a_0 \omega_0) \tilde{\omega} \right) \tag{77}$$

5.3 Linearised Model

In the following section state space models are presented both for the fast and the slow system, i.e. the pump flow and the tank pressure dynamics respectively.

5.3.1 State Space Model - Pump Dynamics

Before linearising the system model, presented in Eq. (57), we need to identify the equilibrium points. This is done by evaluating the model at fixed ω and OD , with its dynamics set to 0. We choose pump speeds as an average of those used in Section 11.1, which is 66% for each pump. We set OD to 0.5 for each valve. This yields the following equilibrium point:

$$q_c^* = \begin{bmatrix} q_3 \\ q_7 \end{bmatrix} \begin{bmatrix} 0.1992 \\ 0.0231 \end{bmatrix} \tag{78}$$

$$\bar{d}_f^* = \begin{bmatrix} d_1 \\ d_5 \\ q_9 \end{bmatrix} \begin{bmatrix} 0.3752 \\ -0.2913 \\ -0.2876 \end{bmatrix} \tag{79}$$

$$d_\tau^* = 0 \tag{80}$$

The linearised model of the system can then be expressed on the standard state space form given in Eq. (81) by evaluating Eq. (77) at the equilibrium point:

$$\begin{aligned}
\dot{x} &= Ax + Bu \\
y &= Cx
\end{aligned} \tag{81}$$

$$A = \begin{bmatrix} -0.3236 & -0.0406 & -0.1577 & 10.0671 & -0.7623 & 0.0000 \\ -0.1429 & -0.3189 & 0.2176 & -1.3489 & -6.0984 & 0 \\ -0.0275 & 0.0687 & -0.3968 & 7.1758 & 1.5246 & 0 \\ 0.1089 & 0.0687 & 0.0196 & -20.2792 & 1.5246 & -0.0000 \\ -0.0551 & -0.0443 & -0.0272 & 1.4425 & -15.3466 & -0.0999 \\ 0.1486 & 0.0817 & -0.2877 & 11.1436 & 8.2419 & -0.4174 \end{bmatrix} \tag{82}$$

$$B = \begin{bmatrix} 0.0982 & 0.0000 \\ 0.0078 & 0 \\ 0.1147 & 0 \\ -0.0408 & 0 \\ -0.0087 & -0.0193 \\ -0.0651 & -0.0715 \end{bmatrix} \quad (83)$$

The output matrix C is then constructed to extract the pump flows from the states of the system:

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 & -1 \end{bmatrix} \quad (84)$$

This yields a system with pump speeds $\{\omega_1, \omega_2\}$ as inputs and pump flows $\{d_1, d_{13}\}$ as outputs:

$$\begin{aligned} u &= \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \\ x &= \begin{bmatrix} q_c \\ d_f \\ d_\tau \end{bmatrix} \\ y &= \begin{bmatrix} d_1 \\ d_{13} \end{bmatrix} \end{aligned} \quad (85)$$

5.3.2 State Space Model - Tank Pressure Dynamics

The tank pressure dynamics are significantly simpler than the fast dynamics of the system - in fact, the slow tank pressure dynamics of the system are a linear first-order system that, per Eq. (58), evolve according to:

$$\begin{aligned} \dot{p}_\tau &= -\tau d_\tau \\ &= \tau (d_1 + d_{13} + d_5 + d_9) \end{aligned} \quad (86)$$

We can discretise this with Euler's method, resulting in:

$$\begin{aligned} p_\tau(k+1) &= p_\tau(k) + \tau (d_1(k) + d_{13}(k) + d_5(k) + d_9(k)) t_s = \mathcal{T} (d_1(k) + d_{13}(k) + d_5(k) + d_9(k)) \\ \mathcal{T} &= \tau \cdot t_s \end{aligned} \quad (87)$$

which can be converted into the following state-space system:

$$\begin{aligned} p_\tau(k+1) &= Ap_\tau(k) + B_p d_p(k) + B_c d_c(k) = Ap_\tau(k) + \left(B_p \begin{bmatrix} d_1(k) \\ d_{13}(k) \end{bmatrix} + B_c \begin{bmatrix} d_5(k) \\ d_9(k) \end{bmatrix} \right) \\ A &= 1, \quad B_p = [\mathcal{T} \quad \mathcal{T}], \quad B_c = [\mathcal{T} \quad \mathcal{T}] \end{aligned} \quad (88)$$

We note the presence of $B_c d_c$, an exogenous input over which we have no control. This will be addressed in Sections 6 and 7.

5.4 Influence of Pump Delay

The pumps of the WDN system introduces a delay of approximately 4 seconds. According to [6, pp. 182-183], a time delay θ limits the feasible closed-loop bandwidth of a system to:

$$\omega_c < \frac{1}{\theta} \quad (89)$$

This limitation on the inner loop inherently limits the outer loop as well, so long as standard cascade control practices are observed. Typically, the inner loop is required to be at least 5 times faster than the outer loop.

5.5 The Root Locus Method

For design of the pump controllers the root locus method will be used. The root locus is a powerful visual tool that investigates the location of the closed-loop poles of a system, when introducing feedback with a gain K . The root locus can be drawn by hand by calculating the location of the closed-loop poles of a system for a large number of feedback gains. In practice however, this is done using a software package like Matlab.

While there exist many rules for sketching root locus by hand, only a few of them are truly important to understand how the root locus is used for controller design:

- 1) The closed-loop poles of the system move from the open-loop poles at $K = 0$ to the open-loop zeros as $K \rightarrow \infty$, unless the number of poles and zeros is different, in which case the closed-loop poles will go to or come in from infinity.
- 2) The number of lines going to or from ∞ is equal to the difference between the number of open-loop poles and zeros.
- 3) The angle and place of gravity of the lines going to or from ∞ is dependent of the difference between the number of open-loop poles and zeros and their location in the complex plane.

The first part of rule 1 tells us that the closed-loop poles are equal to the open-loop poles when $K = 0$. This is intuitively understood as $K = 0$ means there is no feedback, thus the closed and open-loop systems are identical. The second part tells us that when $K \rightarrow \infty$ the closed-loop poles will be located at the open-loop zeros. This is understood by analysing the closed-loop transfer function when the open-loop transfer function is given as

$$G_{ol}(s) = \frac{Z(s)}{P(s)} \quad (90)$$

where $Z(s)$ and $P(s)$ are the polynomials describing the zeros and poles respectively. The closed-loop transfer function is thus

$$G_{cl}(s) = \frac{G_{ol}(s)}{1 + KG_{ol}(s)} \quad (91)$$

where the roots are the values of s that make the denominator equal to 0

$$0 = 1 + K \cdot G_{ol}(s) = 1 + K \frac{Z(s)}{P(s)} \quad (92)$$

Restructuring this yields

$$0 = 1 + K \frac{Z(s)}{P(s)} = 1P(s) + K \frac{Z(s)P(s)}{P(s)} = P(s) + KZ(s) \quad (93)$$

This confirms that setting $K = 0$ makes the closed-loop poles equal to $P(s)$ which is exactly the open-loop poles. When $K \rightarrow \infty$ the $KZ(s)$ term is dominating, which is why the closed-loop poles will then equal the open-loop zeros $Z(s)$.

Rules 2 and 3 simply tells us that the possible locations of the closed-loop poles - specified by the loci - changes with the position of the open-loop poles and zeros. This enables the designer to place additional poles and zeros to change the behaviour of the closed-loop system.

5.6 Design of Pump Controllers

This section will cover design of the inner-loop pump controllers, which corresponds to control of the fast dynamics of the system.

5.6.1 Decentralised Control

Based on the obtained linearised model, controllers for the pumps are being introduced. Its desired to obtain decentralised control as this mitigates network faults and decrease delays in the system. Decentralised control can be obtained in a MIMO system if the plant is close to diagonal, meaning that the system can be considered a collection of independent sub-systems [6, p. 91]. In order to validate whether decentralised control can be obtained, the interactions in the off-diagonals of the system are investigated with magnitude plots. For this analysis the tank is assumed to have a constant pressure. This is part of the separation of the inner and outer loop, where the outer loop is assumed to be slow enough to be ignored in analysis of the inner loop. The magnitude plots can be seen in Fig. 5. They show the gain from a input frequency ω on each of the pumps to the output flows of the pumps.

As seen in Fig. 5, the gain from ω_1 to d_{13} is for all frequencies less than -50 dB, compared to -10 dB from ω_1 to d_1 . Likewise, the gain from ω_2 to d_1 is less than -50 dB for all frequencies, compared to -15 dB from ω_2 to d_{13} .

In both cases, the coupling is at least -35 dB corresponding to less than a factor of $\frac{1}{50}$, and the system can reasonably be assumed to have no meaningful cross-coupling, which allows for independent SISO design of the respective controllers solely from the diagonal transfer functions in the transfer matrix.

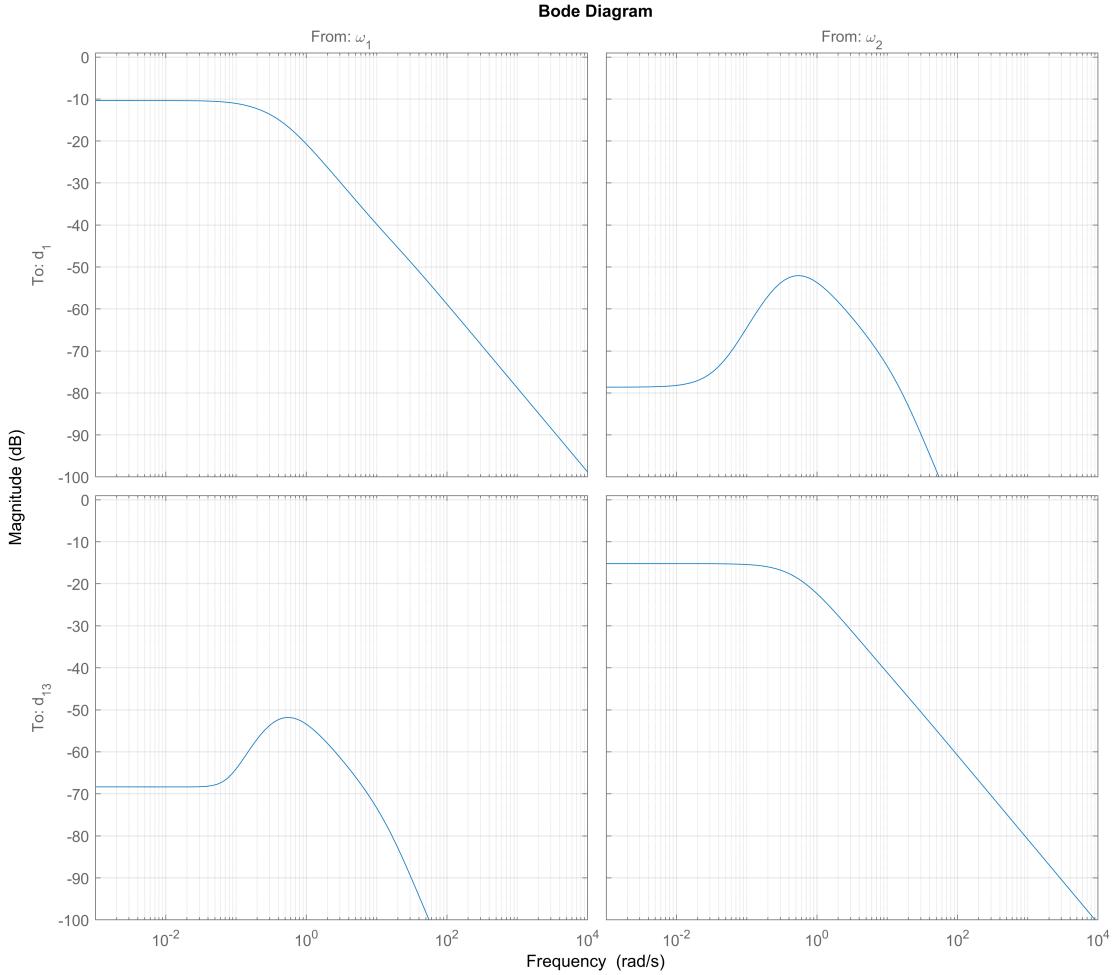


Fig. 5. Magnitude Plot from inputs to outputs

5.6.2 Model Order Reduction

On closer comparison, the diagonal transfer functions can be seen to have approximately identical poles and zeroes in Eq. (94):

$$\begin{aligned} z_{11} &= [-18.9173 \quad -13.9942 \quad -0.5571 \quad -0.3357 \quad -0.1944] \\ z_{22} &= [-20.7216 \quad -14.1600 \quad -0.4079 \quad -0.3639 \quad -0.1597] \\ p_{11} = p_{22} &= [20.7703 \quad -14.8635 \quad -0.5624 \quad -0.3930 \quad -0.3337 \quad -0.1597] \end{aligned} \quad (94)$$

Realising that many of the poles more or less cancel out with the zeros can simplify the model significantly for the root locus design, yielding poles and zeros Eq. (95):

$$\begin{aligned} z &= [-0.1944] \\ p &= [-0.3930 \quad -0.1597] \end{aligned} \quad (95)$$

This reduced-order approximation allows the design of only one controller for both the subsystems.

5.6.3 Modelling of Time Delay

The fast dynamics are known to have a significant delay. However, delays are represented by $e^{-T_d s}$ in the Laplace domain and thus are not rational functions, which means that they cannot be represented as transfer functions. The delay at the pumps can instead modelled with by Padé approximation, which is a rational-function approximation of a delay. The order of the approximation is not unimportant. The higher the order, the closer to a true time delay the result will be, but this comes at the cost of complexity, as a n th-order approximation places n LHP poles and n RHP zeroes to simulate the delay. This is shown in Fig. 6.

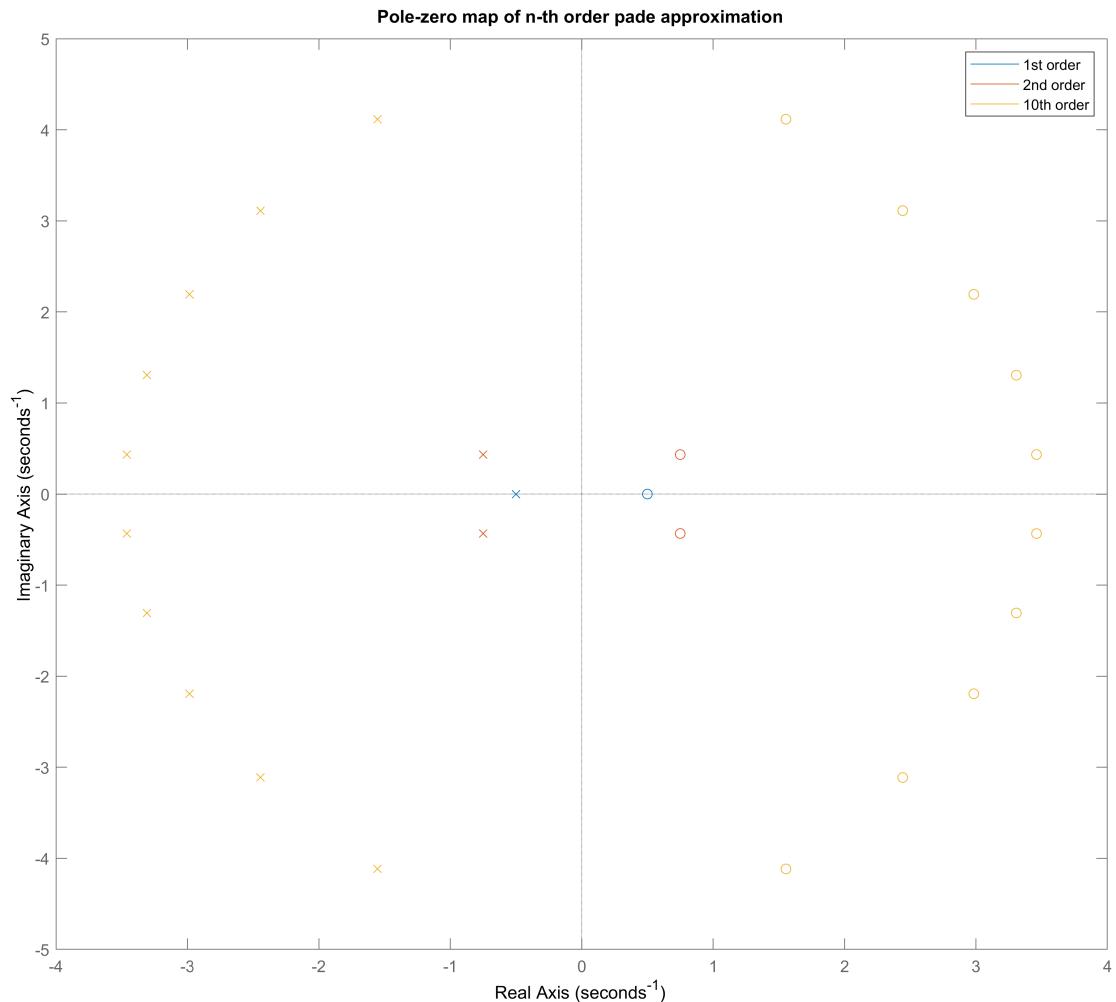


Fig. 6. Pole-zero map of various order Padé approximations of a 4 second delay

An investigation has been performed to show the error in closed-loop pole locations introduced by using a low order approximation. Fig. 7 and Fig. 8 show the closed-loop pole locations when using a 1st, 2nd, and 10th order Padé approximation. The plots show a noticeable difference from 1st to 2nd order, but no significant change beyond that.

When considering both convenience of use and modelling error the optimal approximation order was chosen to be 2.

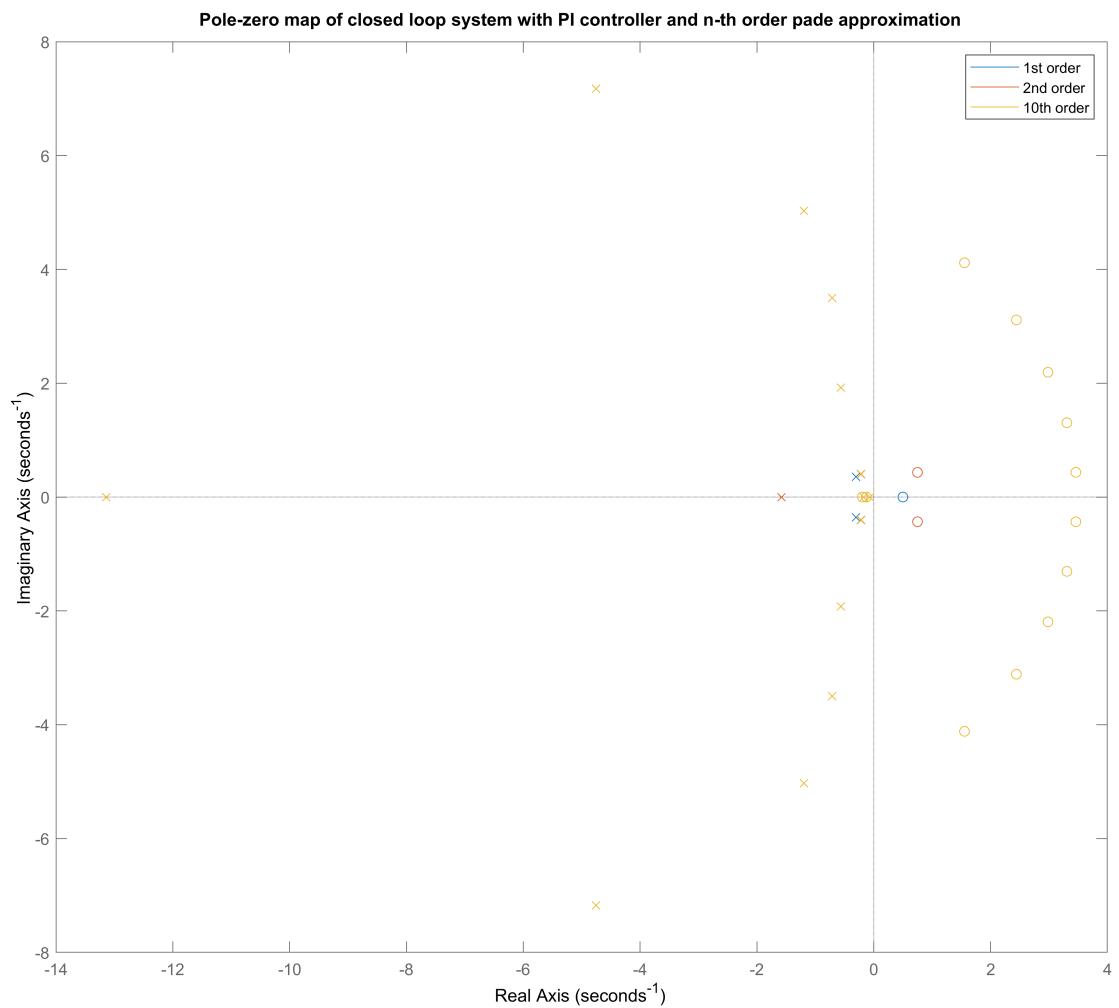


Fig. 7. Pole-zero map of closed-loop pole locations with various order Padé approximations of a 4 second delay

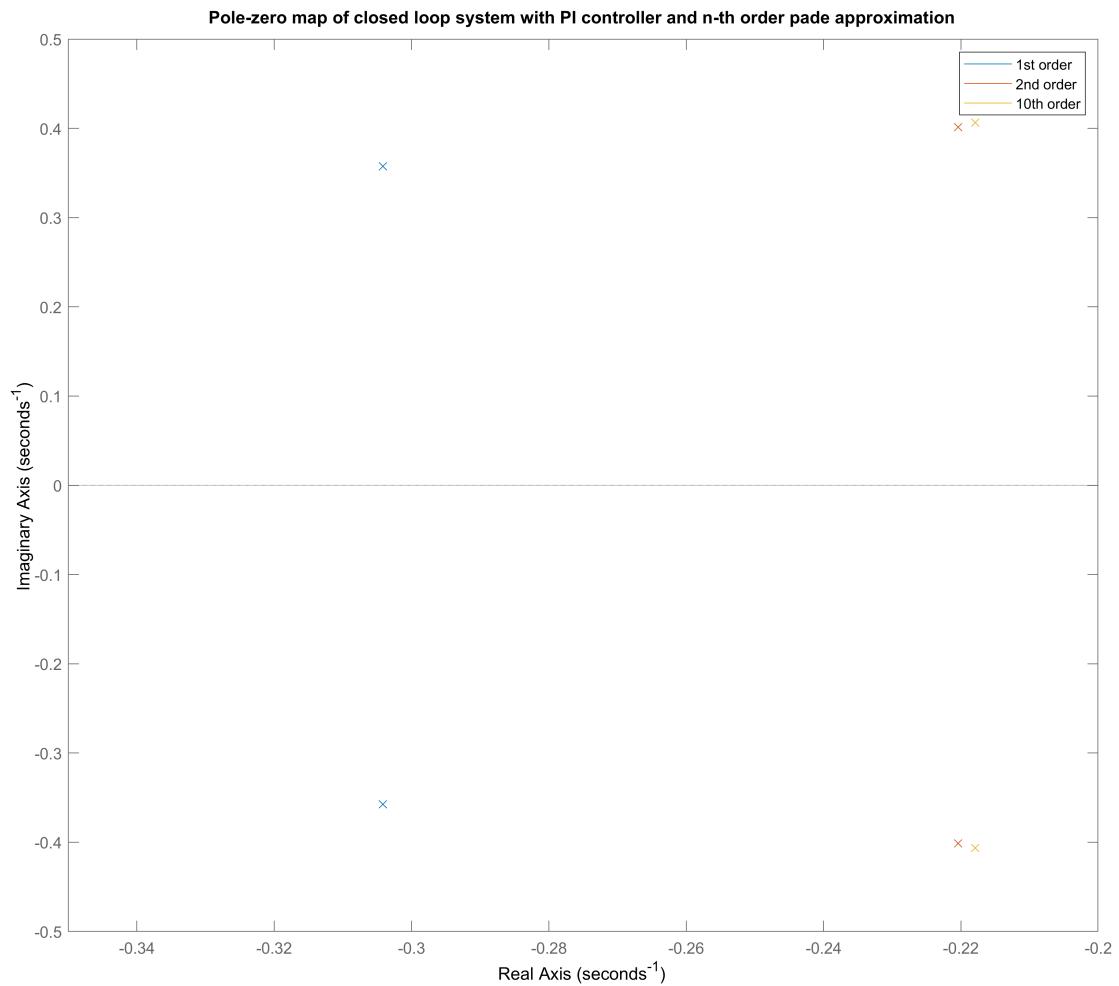


Fig. 8. Pole-zero map of closed-loop pole locations with various order Padé approximations of a 4 second delay. Zoomed in on the dominant complex conjugate pole pair.

5.6.4 Requirements for Control Performance

Before starting the actual root locus design of the controller, the desired performance of the controlled system needs to be defined.

The first priority of the controlled system is for it to be stable. This is achieved by ensuring that the closed-loop poles of the system are placed in the left half plane on the s-plane, in accordance with the Routh-Hurwitz criterion. Simultaneously, the inner loop of the system should be at least 5 times faster than the slow outer system. The desired BW of the inner loop is chosen to be $0.1 \frac{\text{rad}}{\text{s}}$. This in turn is another restriction on the closed-loop pole locations.

Next, the controlled system should have no steady state error for step inputs. This ensures that the outer controller freely can set and achieve the desired pump flow reference. Lastly the system should have no overshoot and low amount of oscillations in its stepresponse, to indicate a sound phase margin to accommodate model uncertainty.

Since the transfer function of the desired subsystem can be simplified as having two real poles and one zero, and the Padé approximation adds two complex poles and zeros, the system can not be described as a second order system. Therefore traditional requirements like damping ratio do not apply in the normal sense. The requirements will thus be simpler, allowing more freedom for the designer to tune the controller based on the actual stepresponse.

The requirements for the controller can be condensed to the following:

- No poles may be in the right half-plane.
- An open-loop pole must be placed in the origin.
- The closed-loop system must have a bandwidth of approximately $0.1 \frac{\text{rad}}{\text{s}}$.
- The closed-loop system must have a time constant of approximately $\frac{1}{BW} = \frac{1}{0.1} = 10\text{s}$.

5.6.5 Choice of Controller Type

A PI controller is considered to be a sufficient solution for the requirements put forth. The internal model principle ensures that integral action will take care of the steady state error, and the gain and zero location can be chosen based on the design criteria.

5.6.6 Root Locus Design

Since the controller is to be used locally for one pump, the root locus is drawn for the transfer function that takes the rotational speed of pump 1 as input and the resulting flow of pump 1 as output. The transfer function is in reduced form, where close pole-zero pairs have been cancelled out. The 2nd order Padé approximation of a 4-second time delay is included.

The controller design is then performed in four steps:

- 1) Place one pole at the origin.
- 2) Place one real zero in the left half plane.
- 3) Select a gain K that yields optimal closed-loop poles.
- 4) Evaluate the step response and readjust the zero location and gain value if needed.

While step 1 requires no intuition, step 2 has to be done rather carefully. Placing the zero too close to the origin will slow down the integration drastically. Placing it too far to the left, however, creates a locus where the possible closed-loop poles are all undesirable. For step 3, the gain was increased until the bandwidth of $0.1 \frac{\text{rad}}{\text{s}}$ was achieved with no oscillations. As step 4 implies, the PI controller was designed after some iterations, resulting in the following controller:

$$G_{PI} = 1.8 \frac{s + 0.125}{s} \quad (96)$$

With this controller the root locus can be seen on Fig. 9. The plot shows zeroes as o's, poles as x's and closed-loop poles as red dots.

The resulting closed-loop step response is seen on Fig. 10. The step response shows an initial undershoot. This is not a result of the physical system, but rather the Padé approximation, which places zeroes in the right half plane. These zeroes cause the step response to undershoot in an attempt to delay the signal. Increasing the order of approximation can reduce this, as shown in Fig. 11. Fig. 11 also shows that no significant error in the step response is caused by using a 2nd order approximation rather than 10th order.

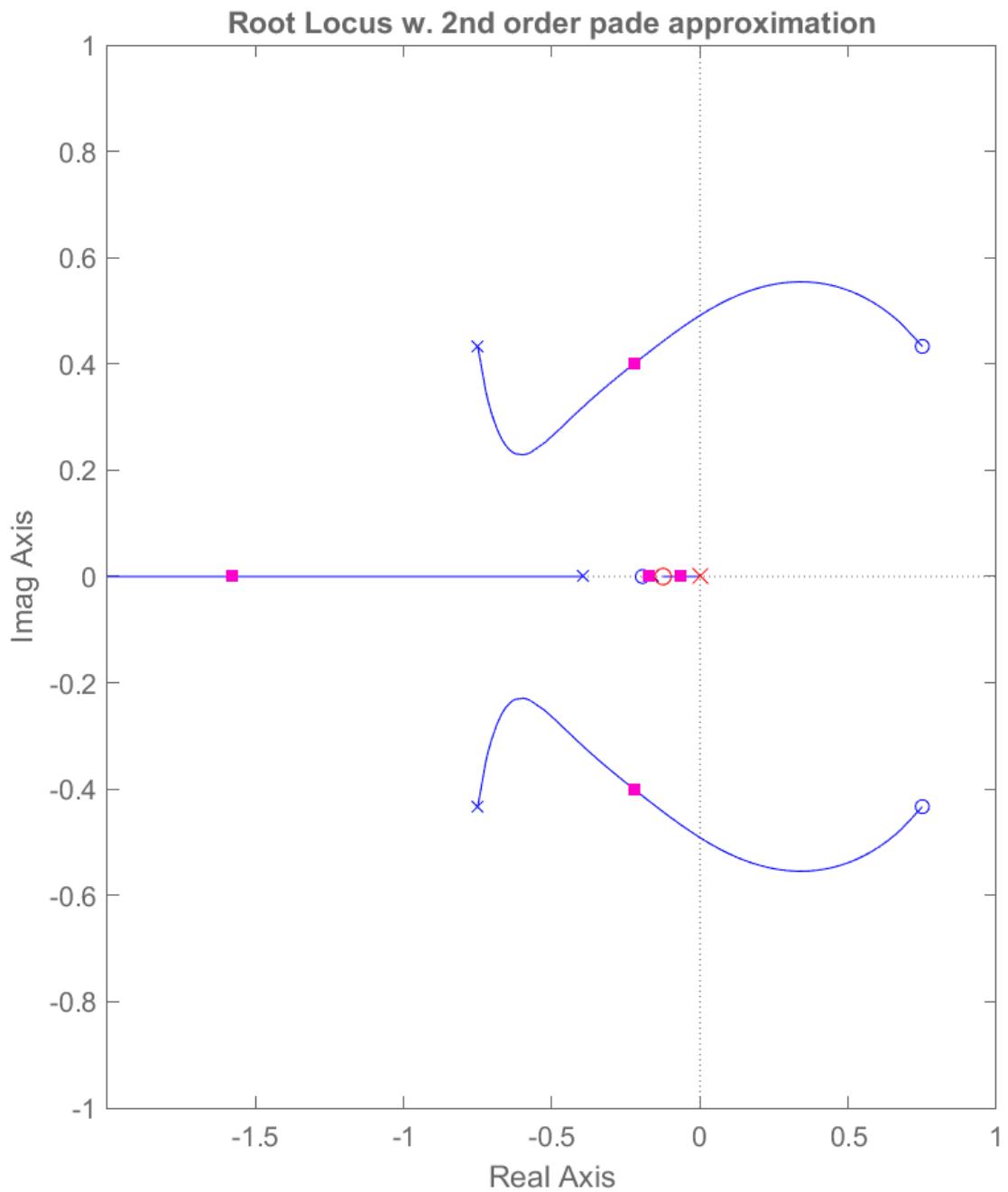


Fig. 9. Root locus of delayed pump transfer function with PI controller

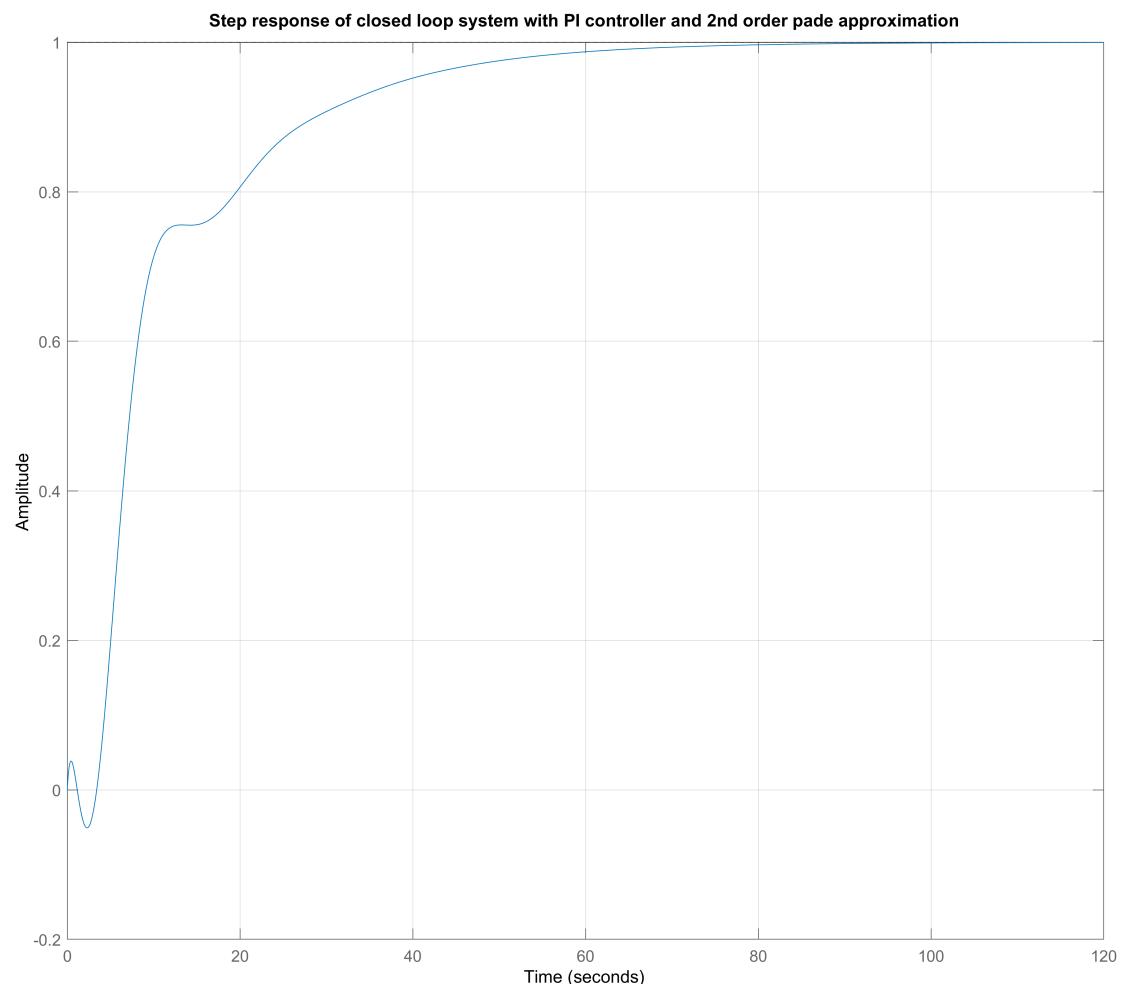


Fig. 10. Step response of the closed-loop system with a 2nd order Padé approximation

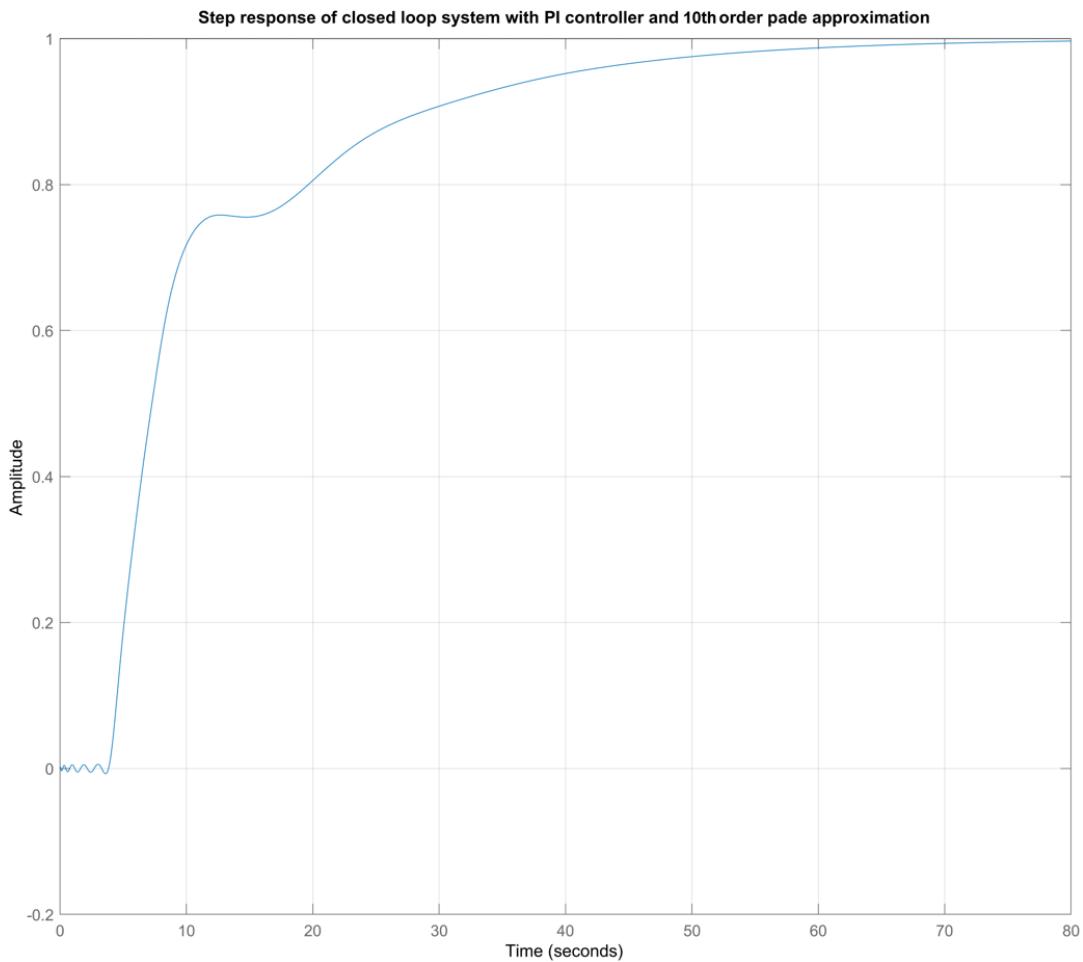


Fig. 11. Step response of the closed-loop system with a 10th order Padé approximation

6 OPTIMAL CONTROL AND THE LINEAR-QUADRATIC REGULATOR

In the following section, we will develop the theory behind the Linear-Quadratic Regulator (LQR) based on the theory of *calculus of variations*. We follow the presentation given in [7] by Liberzon closely, but take the basic results relating to Pontryagin's Minimum Principle and the solution to the Algebraic Riccati Equation for granted.

6.1 Basic Structure of an Optimal Control Problem

We start by sketching the basic structure of an optimal control problem in the language of calculus of variations. Let:

$$\dot{x} = f(x, u, t) \quad (97)$$

$$x(t_0) = x_0, x \in \mathbb{R}^n, u \in U \in \mathbb{R}^m \quad (98)$$

where $t \in \mathbb{R}$ is the time and x, u are functions of t , with U the set of admissible controls. We now seek to minimize a cost functional of the type:

$$J = \mathcal{M}(x(T)) + \int_0^T \mathcal{L}(x(t), u(t)) dt \quad (99)$$

This type of minimization problem is known as a *Bolza problem* and \mathcal{L} is called the *Lagrangian*. To minimize this functional under the constraint of the dynamics in Eq. (98), we can formulate a function known as the *Hamiltonian*:

$$\mathcal{H}(x(t), u(t), \lambda(t), t) = \lambda^T(t) f(x(t), u(t)) + \lambda_0 \mathcal{L}(x(t), u(t)) \quad (100)$$

where λ^T are *Lagrange multipliers* that are commonly referred to as the *costates* of Eq. (97) in the context of control. The optimal state, control, and Lagrange multiplier sequences x^* , u^* , and λ^* are then given by Pontryagin's Minimum Principle, which states that the optimal sequences must minimize the Hamiltonian such that:

$$\forall t \in [0, T] \wedge \forall u \in U : \mathcal{H}(x^*(t), u^*(t), \lambda^*(t), t) \leq \mathcal{H}(x^*(t), u(t), \lambda^*(t), t) \quad (101)$$

while the costates and states must evolve according to each other's respective Hamiltonian gradients:

$$-\dot{\lambda}^T(t) = \frac{\partial}{\partial x} \mathcal{H}(x^*(t), u^*(t), \lambda(t), t) = \frac{\partial}{\partial x} (\lambda^T f(x^*(t), u^*(t)) + \lambda_0 \mathcal{L}(x^*(t), u^*(t))) \quad (102)$$

$$\dot{x} = \frac{\partial}{\partial \lambda} \mathcal{H}(x^*(t), u^*(t), \lambda(t), t) = \frac{\partial}{\partial \lambda} (\lambda^T f(x^*(t), u^*(t)) + \lambda_0 \mathcal{L}(x^*(t), u^*(t))) \quad (103)$$

and the costates must fulfil the terminal condition:

$$\lambda^T(T) = \mathcal{M}_x(x(T)) \quad (104)$$

6.2 The Linear-Quadratic Control Problem

We will now address the well-known LQR problem as a special case of an optimal control problem. Following the same basic problem structure as in Section 6.1, let the dynamics of the system be given by:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (105)$$

$$x(t_0) = x_0 \quad (106)$$

and let a cost functional be given by:

$$J = \int_{t_0}^{t_1} (x^T(t)Q(t)x(t) + u^T(t)R(t)u(t))dt + x^T(t_1)\mathcal{M}x(t_1) \quad (107)$$

with $Q(t) \wedge \mathcal{M}$ symmetric and positive semidefinite, and $R(t)$ symmetric positive definite. Clearly, this is a Bolza problem with the Lagrangian:

$$\mathcal{L}(x(t), u(t)) = x^T(t)Q(t)x(t) + u^T(t)R(t)u(t) \quad (108)$$

and per [7] we may freely choose $\lambda_0 = -1^5$ such that the Hamiltonian becomes:

$$\mathcal{H}(x(t), u(t), \lambda(t), t) = \lambda^T(t)(A(t)x(t) + B(t)u(t)) - x^T(t)Q(t)x(t) - u^T(t)R(t)u(t) \quad (109)$$

Intuitively, the control gradient $\frac{\partial \mathcal{H}}{\partial u}$ of Eq. (109) must vanish along the optimal control trajectory $u^*(t)$ if we are to satisfy the first condition of Pontryagin's Minimum Principle in Eq. (101). Taking the partial differential, we obtain:

$$\frac{\partial \mathcal{H}}{\partial u} = B^T(t)\lambda^*(t) - 2R(t)u^*(t) \quad (110)$$

and thus it seems clear that $u^*(t)$ must satisfy:

$$u^*(t) = \frac{1}{2}R^{-1}(t)B^T(t)\lambda^*(t) \quad (111)$$

It can then be shown - doing so is expressly outside the scope of this report, but relies on the remaining conditions Eqs. (102) to (104) - that $\lambda^*(t)$ must satisfy a relation of the form:

$$\lambda^*(t) = -2P(t)x^*(t) \quad (112)$$

We see that this is quite clearly true at the boundary $t = T$, where per Eq. (104) we must satisfy that:

$$\lambda^T(T) = \mathcal{M}_x(x(T)) = -2\mathcal{M}x^*(T) \quad (113)$$

The relation in Eq. (112) then gives a state feedback control law in the form of:

$$u^*(t) = -R^{-1}(t)B^T(t)P(t)x^*(t) \quad (114)$$

and furthermore it can be shown that $P(t)$ evolves according to dynamics known as the *Riccati differential equation*, which are given by:

$$\dot{P}(t) = -P(t)A(t) - A^T(t)P(t) - Q(t) + P(t)B(t)R^{-1}(t)B^T(t)P(t) \quad (115)$$

6.3 The Infinite-Horizon Linear-Quadratic Control Problem

We will now narrow down the contents of Section 6.2 even further to consider a particularly - and in terms of its solution, rather uniquely beautiful - special case of the LQR problem. Specifically, we will consider the infinite-horizon, time-invariant case with no terminal cost. Thus, let the dynamics be given by:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (116)$$

$$x(t_0) = x_0 \quad (117)$$

and let the cost functional be:

$$J = \int_{t_0}^{\infty} (x^T(t)Qx(t) + u^T(t)Ru(t))dt \quad (118)$$

5. Letting $\lambda_0 = 1$ will simply result in an inverted sign of the optimal gain, and an optimal feedback law of the type $-Kx$ instead of Kx .

In the absence of a terminal cost, this is now known as a *Lagrange problem*. Constructing the Hamiltonian as before, we have:

$$\mathcal{H}(x(t), u(t), \lambda(t), t) = \lambda^T(t)(Ax(t) + Bu(t)) - x^T(t)Qx(t) - u^T(t)Ru(t) \quad (119)$$

As before, the control gradient:

$$\frac{\partial \mathcal{H}}{\partial u} = B^T \lambda^*(t) - 2Ru^*(t) \quad (120)$$

must vanish along the optimal trajectory, necessitating that:

$$u^*(t) = \frac{1}{2}R^{-1}B^T \lambda^*(t) \quad (121)$$

and it can analogously to Section 6.2 be shown that:

$$\lambda^*(t) = -2Px^*(t) \quad (122)$$

such that:

$$u^*(t) = -R^{-1}B^T Px^*(t) \quad (123)$$

with P now time-invariant and fulfilling the *algebraic Riccati equation*:

$$PA + A^T P + Q - PBR^{-1}B^T P = 0 \quad (124)$$

This motivates the - frankly somewhat stunning - conclusion that for the infinite-horizon, time-invariant LQR problem, the optimal state feedback law may be calculated *entirely* offline.

A final important result concerns the global stability of this feedback law. Let an auxiliary equation to Eq. (116) be given by:

$$y = Cx \quad (125)$$

then choosing $Q = C^T C$ such that the cost functional becomes:

$$J(u) = \int_{t_0}^{\infty} (x^T(t)C^T C x(t) + u^T(t)R(t)u(t)) dt \quad (126)$$

guarantees exponential stability of the closed-loop system $\dot{x}^* = (A - BR^{-1}B^T P)x^*$ so long as (A, C) are an observable pair [7].

6.4 Tracking LQR and Integral Action

The aware reader will have noted that the preceding sections in Section 6 strictly address *regulator* problems - i.e., the regulation of a system to the origin. In practice, this is rarely a sufficient problem definition for practical purposes, as most systems have some setpoint or other. Fortunately, casting the Lagrange problem in Eq. (118) as a tracking problem is largely trivial, as we may simply choose a new coordinate system with the target as the origin. We may perform a similar coordinate shift on the control to specify a desired operating point. For convenience of notation, we let $\hat{x} = x - x_r$ and $\hat{u} = u - u_r$ and take the time index t to be implicit. Then we can represent the Lagrange problem in the shifted coordinate system as:

$$J = \int_{t_0}^{\infty} (\hat{x}^T Q \hat{x} + \hat{u}^T R \hat{u}) dt \quad (127)$$

We can by a similar ploy convert Eq. (127) into an *output* tracking problem, which is perhaps the most common practical control problem, as C is simply a linear transformation on the states:

$$J = \int_{t_0}^{\infty} ((C\hat{x})^T Q_y (C\hat{x}) + \hat{u}^T R \hat{u}) dt = \int_{t_0}^{\infty} (\hat{y}^T Q_y \hat{y} + \hat{u}^T R \hat{u}) dt \quad (128)$$

We now note that as with many other state-space control algorithms, an LQR controller does not inherently have integral action. There are multiple ways of adding this to LQR, such as the classical integrator state approach [6]. Here the extended state $\bar{x} = [x \ x_i]^T$ is defined, and the system evolves according to the dynamics:

$$u = -\bar{K}\bar{x} \quad (129)$$

$$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}u + B_r r \quad (130)$$

$$y = \bar{C}\bar{x} \quad (131)$$

$$\bar{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}, \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, B_r = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \bar{C} = [C \ 0], \bar{K} = -[K \ -K_i], \quad (132)$$

This approach, however, introduces the issue of weighting x_i appropriately. This can be an awkward weight to choose, as the state represents the time integral of the tracking error, rather than the tracking error directly. We therefore prefer an alternative integral action design, which we describe in Section 6.5.

6.5 The Velocity-Form LQR Algorithm

We will now present the velocity-form LQR algorithm, as detailed in [8], [9]. We will make the initial assumption that the system dynamics have been discretised by some appropriate method, and we then define the deviation variables:

$$\Delta x_k = x_k - x_{k-1}, \Delta y_k = y_k - r_k, \Delta u_k = u_k - u_{k-1} \quad (133)$$

where k is the sample index. We then define the extended vectors and matrices:

$$\tilde{\zeta}_k = \begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix}, \tilde{u}_k = \Delta u_k, \tilde{A} = \begin{bmatrix} A & 0 \\ CA & I \end{bmatrix}, \tilde{B} = \begin{bmatrix} B \\ CB \end{bmatrix}, \tilde{C} = [0 \ I] \quad (134)$$

and the velocity-form dynamics:

$$\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + \tilde{B}\tilde{u}_k \quad (135)$$

$$\Delta y_k = \tilde{C}\tilde{x}_k \quad (136)$$

$$(137)$$

We now note a number of interesting properties about this velocity-form algorithm. Consider first the origin regulation problem, i.e. $\tilde{x} \rightarrow 0$. Clearly, driving \tilde{x} to 0 must drive the system to settle at exactly r , per the definitions in Eq. (133), as:

$$\begin{aligned} \tilde{x} = 0 &\Rightarrow \Delta x = 0 \\ \tilde{x} = 0 &\Rightarrow \Delta y = 0 \Rightarrow y = r \end{aligned} \quad (138)$$

Consider now the discretized Lagrange problem phrased in terms of these deviation variables:

$$J = \sum_{k_0}^{\infty} (\tilde{x}^T Q \tilde{x} + \tilde{u}^T R \tilde{u}) = \sum_{k_0}^{\infty} \left(\begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix}^T Q \begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix} + \Delta u_k^T R \Delta u_k \right) \quad (139)$$

Clearly, this functional penalizes deviations from the origin of the state space, which is located exactly at $\{0, 0, r\}$, while penalizing deviations from the default control input of 0. Consider now the case of an arbitrary linearisation around some equilibrium point x_e and corresponding operating point u_{op} . A linearization of this type will take the form:

$$\begin{aligned} f(x, u) &\approx \frac{\partial f}{\partial x} \tilde{x} + \frac{\partial f}{\partial u} \tilde{u}, \\ \tilde{x} &= x - x_e, \tilde{u} = u - u_{op} \end{aligned} \quad (140)$$

which lends itself to the interpretation that:

$$\Delta \tilde{x} = x - x_e, \Delta \tilde{u} = u - u_{op} \quad (141)$$

This is a quite attractive quality when contextualised by the Hartman-Grobman theorem, as we expect linearised model dynamics to closely⁶ approximate the real system dynamics in some region around the equilibrium point $\{x_e, u_{op}\}$, and the velocity LQR algorithm will penalize deviations from the point $\{x_e, u_{op}, r\}$, thus regulating the system to r while deviating minimally from the vicinity of the equilibrium $\{x_e, u_{op}\}$. We note additionally - and importantly - that the control law resulting from Eq. (139) is a control *increment* law, i.e. that:

$$\Delta u^*(k) = -\Delta K(k)\Delta x(k) \quad (142)$$

and that the actual control applied to the system at any time k is:

$$\begin{aligned} \Delta u^*(k) &= -\Delta K(k)x(k) \\ u^*(k) &= \sum_{i=1}^k \Delta u^*(i) \end{aligned} \quad (143)$$

6.6 Disturbance-Accommodating Linear Quadratic Regulator for Exogenous Inputs

We now address a final LQR-related wrinkle. As shown in Eq. (88), the consumer demand flows are modelled as exogenous, uncontrolled inputs. Standard LQR does not accommodate this construction, but may be modified to accommodate exogenous inputs by adding an additional term to the optimal control law [10] such that:

$$u(k) = u^*(k) - B^\dagger B \delta(k) \quad (144)$$

where B^\dagger is the Moore-Penrose pseudoinverse of B and $\delta(k)$ is the exogenous input at time k . We note that this new control law remains optimal with respect to the disturbance-free system, but is *not* optimal with respect to the disturbed dynamics.

6.7 Implementation and Simulation Study

In this section, we detail the implementation and simulation study of a velocity-form LQR controller for the tank pressure dynamics described by Eq. (88). We choose a discretisation time $t_s = 100s$, and the tank constant is $\tau = -0.000096 \frac{\text{bar}}{\text{m}^3}$, such that $T = 0.0096$. This results in the system:

$$\begin{aligned} p_\tau(k+1) &= Ap_\tau(k) + B_p d_p(k) + B_c d_c(k) \\ A &= 1, \quad B_p = B_c = [0.0096 \quad 0.0096], \quad C = 1, \end{aligned} \quad (145)$$

Using the velocity-form transform from Eq. (134), the system becomes:

$$\begin{aligned} \begin{bmatrix} \Delta p_\tau(k+1) \\ p_\tau(k+1) - p_r \end{bmatrix} &= \tilde{A} \Delta p_\tau(k) + \tilde{B}_p \Delta d_p(k) + \tilde{B}_c \Delta d_c(k) \\ \tilde{A} &= \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad \tilde{B}_p = \tilde{B}_c = \begin{bmatrix} 0.0096 & 0.0096 \\ 0.0096 & 0.0096 \end{bmatrix}, \quad \tilde{C} = [0 \ 1] \end{aligned} \quad (146)$$

6. Exactly, in fact, in some indeterminately sized region

We now simulate the system under a variety of conditions, including significant model error and an output disturbance. In every case, a significant state disturbance is applied to the system.

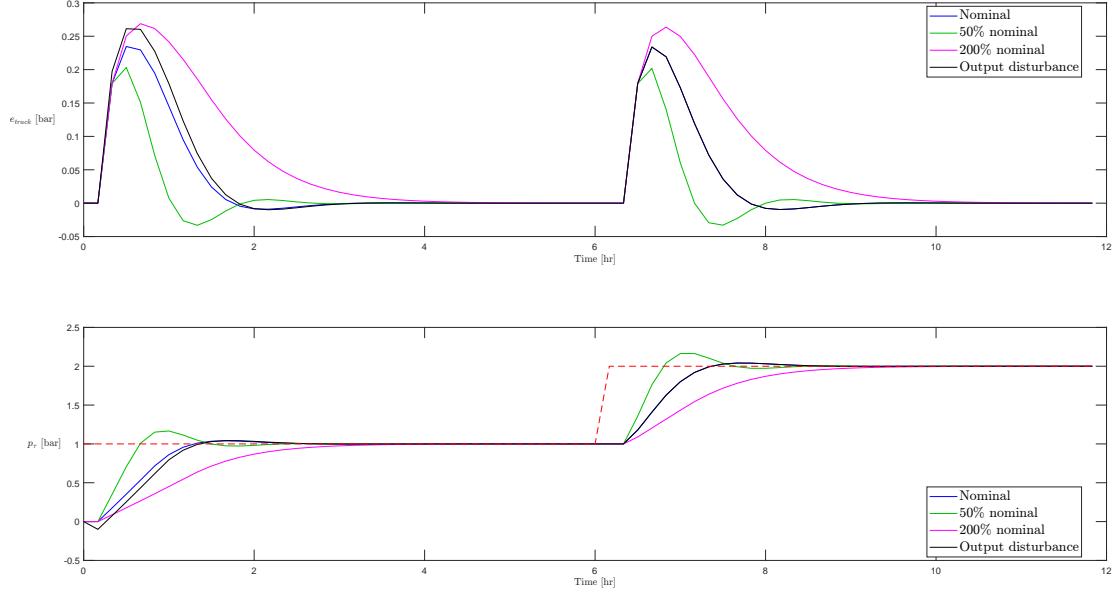


Fig. 12. Tracking performance of the VF-LQR controller

It is clear that the proposed controller is capable of compensating for the disturbance and tracking, even under significant model error or output disturbance.

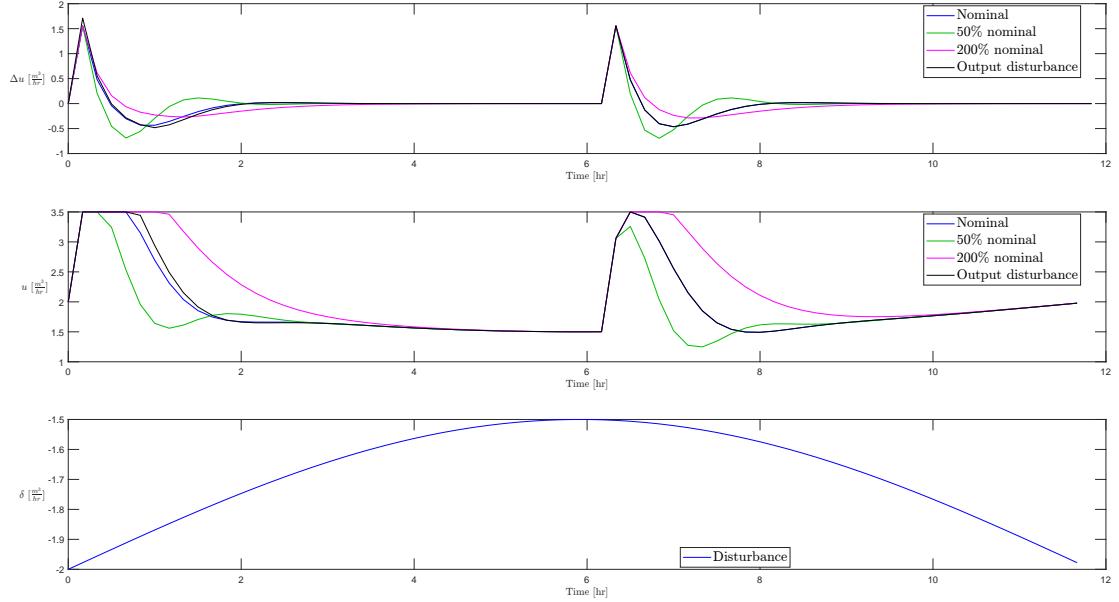


Fig. 13. Control effort of the VF-LQR controller

We see on Fig. 13 that rejection of the state disturbance is, as expected based on Eq. (144), accomplished by reverse-phase injection of the disturbance signal. This also reveals a fundamental limitation of VF-LQR with the proposed disturbance rejection strategy; its tracking ability is asymptotically bounded by the steady-state error in the disturbance estimate.

7 DISTURBANCE ESTIMATOR

This section concerns the estimator used for estimation of the consumer demand flows.

7.1 Estimator Objective

The LQR controller relies on knowledge of the current consumer demand flows. The consumption flow is not measured directly but can be calculated from the combined measurements of tank flow d_τ and pump flows d_p . If no leakage is present, all nodal demands sum to zero (mass conservation) and thus $d_c = d_p + d_\tau$. Noise will be present on all flow sensor measurements which will affect LQR controller performance. Thus it is favourable to estimate the actual underlying consumer flow, attenuating measurement noise.

7.2 Estimator Model

An initial solution for estimating the consumer flow is to utilise the full fast dynamic system matrix. This requires the ability to accurately model the full system, which has been done in this case, but is not feasible for real, large WDNs. Therefore another solution is explored.

If prior knowledge of the behaviour of the process is available, this can be taken advantage of. A new state space model describing the behaviour of the process can be made. The consumption can be approximated by a truncated Fourier series. From the consumption pattern data reviewed in Section 7.4.3 it becomes apparent that a fourth-order approximation is appropriate. This approximation is seen in Eq. (147).

$$d_{c_{approx}}(t) = k_0 + k_1 \cos(\omega t + \phi_1) + k_2 \cos(2\omega t + \phi_2) + k_3 \cos(3\omega t + \phi_3) + k_4 \cos(4\omega t + \phi_4) \quad (147)$$

The principles behind development and utilisation of a such Fourier-series model are adequately understood from a second-order example. Thus only a model of this order is further examined here. Such a model can be seen in Eq. (148).

$$d_{c_{estimate}} = K + k_1 \cos(\omega t) + k_2 \cos(2\omega t) \quad (148)$$

This process model output should be achievable as a combination of states. As such the following states are chosen:

$$x_{desired} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} K \\ k_1 \cos(\omega t) \\ k_2 \cos(2\omega t) \end{bmatrix} \quad (149)$$

As with any state-space system these states need to be expressed in terms of their derivatives ($\dot{x} = Ax$) which are:

$$\dot{x}_{desired} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ -\omega k_1 \sin(\omega t) \\ -2\omega k_2 \sin(2\omega t) \end{bmatrix} \quad (150)$$

No linear combination of our current states can yield this derivative. Thus we need to expand our states such that this can be achieved. Our state vector expands to:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} K \\ k_1 \cos(\omega t) \\ k_1 \sin(\omega t) \\ k_2 \cos(2\omega t) \\ k_2 \sin(2\omega t) \end{bmatrix} \quad (151)$$

Which has the state derivative:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\omega k_1 \sin(\omega t) \\ \omega k_1 \cos(\omega t) \\ -2\omega k_2 \sin(2\omega t) \\ 2\omega k_1 \cos(2\omega t) \end{bmatrix} \quad (152)$$

The system matrix then needs to be:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\omega & 0 & 0 \\ 0 & \omega & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2\omega \\ 0 & 0 & 0 & 2\omega & 0 \end{bmatrix} \quad (153)$$

For $y = Cx$ to be equal to Eq. (148) the C-matrix becomes:

$$C = [1 \ 1 \ 0 \ 1 \ 0] \quad (154)$$

If one wishes to have a signal model of greater harmonic orders, this can be accomplished by block-diagonal concatenation of harmonic signal models, like seen in Eq. (155). This can be performed for an arbitrary amount of harmonics.

$$\begin{aligned} \dot{x} &= A_\delta x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega \\ 0 & \omega & 0 \end{bmatrix} x \\ y &= C_\delta x = [1 \ 1 \ 0] x \end{aligned} \quad (155)$$

Such a concatenation would, for a second-order approximation, look like so:

$$\begin{aligned} \begin{bmatrix} K \\ \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & A_{\delta_1} & 0 \\ 0 & 0 & A_{\delta_2} \end{bmatrix} \begin{bmatrix} K \\ x_1 \\ x_2 \end{bmatrix}, \quad y = C_\delta \begin{bmatrix} K \\ x_1 \\ x_2 \end{bmatrix} \\ x_1, x_2 &\in \mathbb{R}^{2 \times 1} \\ \forall i : A_{\delta_i} &\in \mathbb{R}^{2 \times 2} \\ C_\delta &\in \mathbb{R}^{1 \times 2n+1} \end{aligned} \quad (156)$$

7.3 Estimator Type

Since the disturbance can be described as a linear system, we need a linear estimator. The optimal affine (and hence also linear) estimator is the Kalman filter, if the following conditions are fulfilled

- 1) The consumer flow disturbance can be modelled as a dynamic system excited by white zero-mean Gaussian noise
- 2) The flow measurement noise can be considered white zero-mean Gaussian noise
- 3) The model uncertainty can be modelled as zero-mean white Gaussian noise
- 4) The measurement noise and model uncertainty are uncorrelated

It is the optimal affine estimator given some assumptions about the precision of the model of the process and the knowledge of the process and measurement noise, which is further described in Section 7.4.

7.4 The Kalman Filter

The Kalman filter is an optimal estimator, where the optimality criterion with respect to the Mean Squared Error (MSE) of the residual $x - \hat{x}$. It estimates an unknown process by a combination of a system model and a measurement that is related to the unknown processes. The Kalman filter recursively finds the optimal *Kalman gain* for the given system to minimize the residual MSE. Finding the optimal gain relies on assuming that the model and measurement noise are uncorrelated white noise processes with known covariances. The performance of the Kalman filter relies heavily on the guess of the covariance of measurement and model noise. This means in practice that an online tuning of covariances is desirable to avoid bad guesses of the covariances [11, p. 232].

7.4.1 Mathematical Formulation of the Kalman Filter in State Space

The Kalman Filter described in this report will based on a discrete time state space representation:

$$x(k+1) = Ax(k) + Bu(k) + w(k) \quad (157)$$

$$y(k) = Cx(k) + v(k) \quad (158)$$

where

$x(k)$ is the state vector at time index k

$u(k)$ is the input at time index k

$w(k) \sim \mathcal{N}(0, Q)$ is the model noise at time index k

$v(k) \sim \mathcal{N}(0, R)$ is the measurement noise at time index k

$y(k)$ is the observable output

A is the state transition matrix

B is the control-input matrix

C is the observation matrix,

To simplify the presentation for the readers of this report the Kalman equations will be presented in the form of 8 equations, after which they can be combined to yield the traditional and more dense form of 5 equations. The Kalman equations will further be divided into two phases: the prediction stage and the update stage.

Prediction stage

$$\hat{x}(k|k-1) = A\hat{x}(k-1|k-1) + Bu(k-1) \quad (159)$$

$$\hat{y}(k|k-1) = C\hat{x}(k|k-1) \quad (160)$$

$$P(k|k-1) = AP(k-1|k-1)A^T + Q \quad (161)$$

where

$\hat{x}(k|k-1)$ is the state estimate at time index k given all $k-1$ samples

$\hat{x}(k-1|k-1)$ is the state estimate at time index $k-1$ given all $k-1$ samples

$\hat{y}(k|k-1)$ is the output estimate at time index k given all $k-1$ samples

$P(k|k-1)$ is the estimate covariance matrix at time index k given all $k-1$ samples

$P(k-1|k-1)$ is the updated estimate covariance matrix at time index $k-1$ given all $k-1$ samples

Q is the process noise covariance matrix

All three prediction equations predicts the a priori k th (next) estimate based on the $k-1$ previous samples and the model knowledge.

Update Stage

$$e(k) = y(k) - \hat{y}(k|k-1) \quad (162)$$

$$S(k) = CP(k|k-1)C^T + R \quad (163)$$

$$K(k) = P(k|k-1)C^T S^{-1}(k) \quad (164)$$

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)e(k) \quad (165)$$

$$P(k|k) = (I - K(k)C)P(k|k-1) \quad (166)$$

where

$e(k)$ is the innovation term for time index k

$y(k)$ is the observed output at time index k

$S(k)$ is the innovation covariance for time index k

R is the observation noise covariance

$K(k)$ is the Kalman gain for time index k

$\hat{x}(k|k)$ is the estimate of the state vector at time index k given all k samples

$P(k|k)$ is the updated estimate covariance matrix at time index k given all k samples

I is the identity matrix

All the equations in the update stage estimates the posteriori estimate of the k sample based on all k th samples and the model knowledge. A few expressions to give some intuitions to what the equations actually represent. The estimate covariance, the prediction covariance and innovation covariance can respectively be represented as

$$P(k|k) = \text{cov}(x(k) - \hat{x}(k|k)) \quad (167)$$

$$P(k|k-1) = \text{cov}(x(k) - \hat{x}(k|k-1)) \quad (168)$$

$$S(k) = \text{cov}(e(k)) \quad (169)$$

Compact Form

Now collecting some of the above equations gives the more compact form of the Kalman equations. Equations (159) and (161) remain in the same form, but omitting $Bu(k-1)$ yields Eq. (170) and Eq. (171). $Bu(k-1)$ can be omitted as it is not part of the system model in Eq. (155).

Equations (163) and (164) are combined yielding Eq. (172). Equation (160) is combined with Eq. (162) and Eq. (165) yielding Eq. (173), while Eq. (166) remains the same. All these combined presents the Kalman equations as in [12]

Prediction

$$\hat{x}(k|k-1) = A\hat{x}(k-1|k-1) \quad (170)$$

$$P(k|k-1) = AP(k-1|k-1)A^T + Q \quad (171)$$

Update

$$K(k) = P(k|k-1)C^T(CP(k|k-1)C^T + R)^{-1} \quad (172)$$

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)(y(k) - C\hat{x}(k|k-1)) \quad (173)$$

$$P(k|k) = (I - K(k)C)P(k|k-1) \quad (174)$$

7.4.2 LTI Kalman Filter

If we assume that the covariances and system are time invariant, i.e. $\forall k : \{A(k), B(k), Q(k), R(k)\} = \{A, B, Q, R\}$, the estimate covariance and Kalman gain can be calculated offline as they don't depend on measurements, only on A, C, Q, R . As such, the Kalman filter itself becomes an LTI filter. The steady state prediction covariance can be found by solving its Algebraic Riccati equation (ARE) - it will now be derived in the discrete form.

In steady-state where $k \rightarrow \infty$ we can assume that $P(k|k) = P(k-1|k-1)$. Furthermore the prediction covariance, the estimation covariance, and the Kalman gain converge to a constant value:

$$\begin{aligned}\lim_{k \rightarrow \infty} P(k|k-1) &= \Pi \\ \lim_{k \rightarrow \infty} P(k|k) &= P \\ \lim_{k \rightarrow \infty} K(k) &= K\end{aligned}\tag{175}$$

Equations Eq. (172), Eq. (173) and Eq. (174) are rewritten with their steady state counterparts:

$$K = \Pi C^T (C\Pi C^T + R)^{-1}\tag{176}$$

$$P = (I - KC)\Pi\tag{177}$$

$$\Pi = APA^T + Q\tag{178}$$

We go on the hunt for the damned Riccati equation of Π . We substitute RHS of Eq. (177) into Eq. (178).

$$\begin{aligned}\Pi &= APA^T + Q \\ &= A(I - KC)\Pi A^T + Q \\ &= A\Pi A^T - AKC\Pi A^T + Q\end{aligned}\tag{179}$$

We now substitute RHS of Eq. (176) in for K in Eq. (179) and factorise respectively A and A^T .

$$\begin{aligned}\Pi &= A\Pi A^T - A\Pi C^T (C\Pi C^T + R)^{-1} C\Pi A^T + Q \\ &= A(\Pi - \Pi C^T (C\Pi C^T + R)^{-1} C\Pi) A^T + Q\end{aligned}\tag{180}$$

And let The Matrix Inversion lemma Eq. (183) be used to further simplify what's left in the parenthesis of RHS of Eq. (180) by letting

$$A^{-1} = \Pi, B = C^T, C^{-1} = R, D = C\tag{181}$$

enables

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1}\tag{182}$$

$$(\Pi^{-1} + C^T R^{-1} C)^{-1} = (\Pi - \Pi C^T (C\Pi C^T + R)^{-1} C\Pi)\tag{183}$$

yielding the desired result: The Algebraic Riccati equation in a compact form, Eq. (184).

$$\Pi = A(\Pi^{-1} + C^T R^{-1} C)^{-1} A^T + Q\tag{184}$$

Eq. (184) can be solved for Π numerically and substituted into Eq. (176) yielding the *LTI Kalman Gain* K which is then substituted into Eq. (177) to obtain *LTI Estimate Covariance* P .

$$K = \Pi C^T (C\Pi C^T + R)^{-1}\tag{185}$$

$$P = (I - KC)\Pi\tag{186}$$

An attempt to identify the Algebraic Riccati equation for the Steady-State Estimate Covariance matrix was also attempted:

$$\begin{aligned}P &= (I - KC)\Pi \\ &= \Pi - KC\Pi\end{aligned}\tag{187}$$

Substituting RHS of Eq. (185):

$$P = \Pi - \Pi C^T [C\Pi C^T + R]^{-1} C\Pi\tag{188}$$

Again applying the Matrix Inversion lemma by letting:

$$A^{-1} = \Pi, B = C^T, C^{-1} = R, D = C\tag{189}$$

yielding:

$$P = [\Pi + C^T R^{-1} C]^{-1}\tag{190}$$

substituting RHS of Eq. (178):

$$P = [(APA^T + Q) + C^T R^{-1} C]^{-1} \quad (191)$$

Which indeed is the Algebraic Riccati equation for the *Steady State Estimate Covariance* matrix P . Now the foundation is derived for starting to implement an actual model of the system in the following sections.

7.4.3 Kalman Simulation

Real water consumption data is analysed in order to justify approximating the consumption pattern as a harmonic series. A historical dataset showing typical daily water consumption during a three month period has been obtained from Grundfos A/S. Two arbitrary days from said data set is seen in Fig. 14. Note that the unit of consumption is unknown, and we will normalise this data to fit the expected maximum flow in the laboratory. Using frequency analysis we construct a model containing the four most influential frequency components; this represents the best least-squares fit of the data in a sparse Fourier basis.

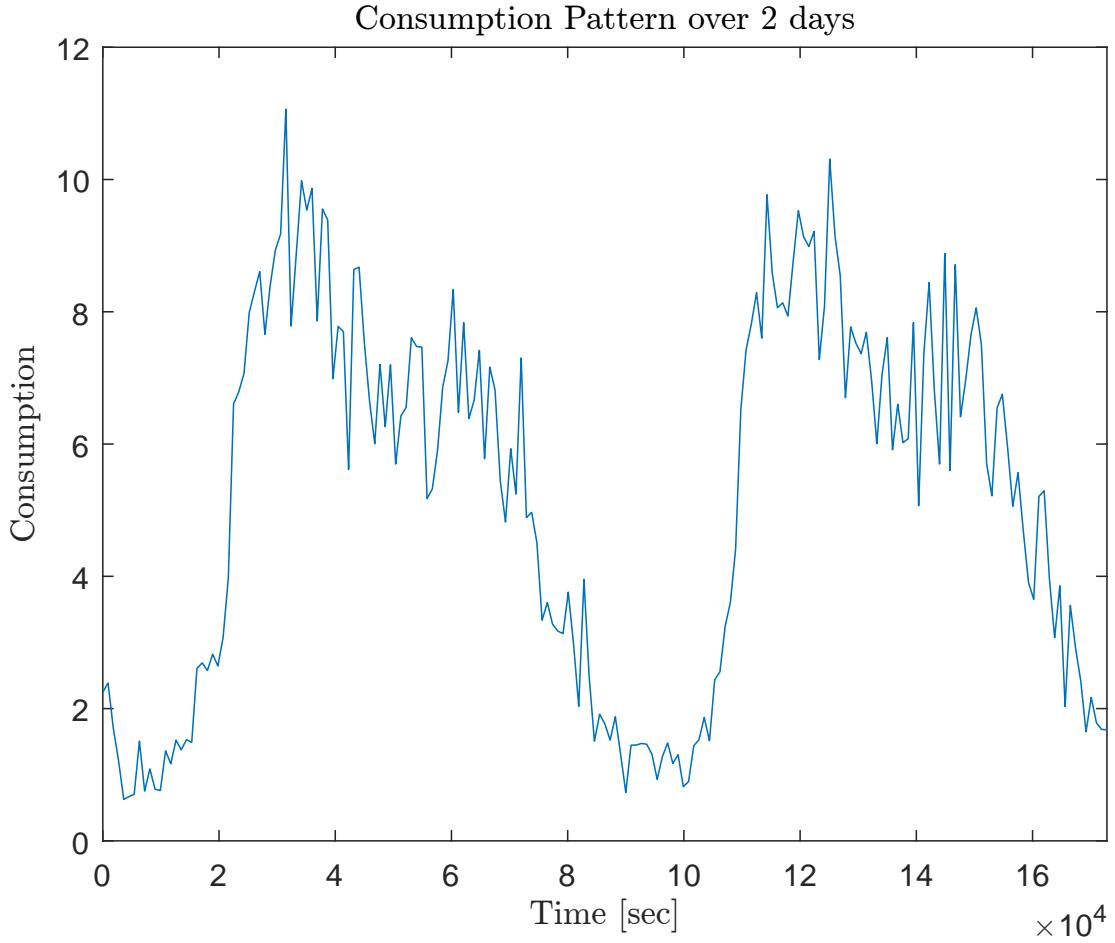


Fig. 14. Consumption pattern over two days

FFT analysis yields amplitude and phase plots as seen in Fig. 15. The amplitude plot clearly shows a bias and the most influential components. To construct the correct signal we also need to identify the phase shift of each component. These can be extracted from the phase plot by identifying which indices correspond to the largest-amplitude frequency components.

The resulting model is compared to the raw historical data in Fig. 16, showing moderate coincidence. Estimate accuracy in a specific application will benefit from more thorough design of the model, possibly including additional harmonics and season-specific consumption data from the local area. Therefore, we are satisfied with the model as a general example of a consumer pattern, without requirements on validity or application. A minor correction is made based on visual inspection of the plot, resulting in a slight timeshift of the entire model to match peaks of the fundamental frequency. This is done to make up for the lack of timeshift information from the FFT.

Simulations of the Kalman filter allows us to verify its ability to track the model and simultaneously allows us to make initial guesses on covariance matrices. We have established that a model may accurately represent consumer patterns and therefore allow the estimator to place greater trust in the model than the measurements. This also improves our ability to detect a leakage, as the error between estimate and measurement becomes more prominent when the model is the primary

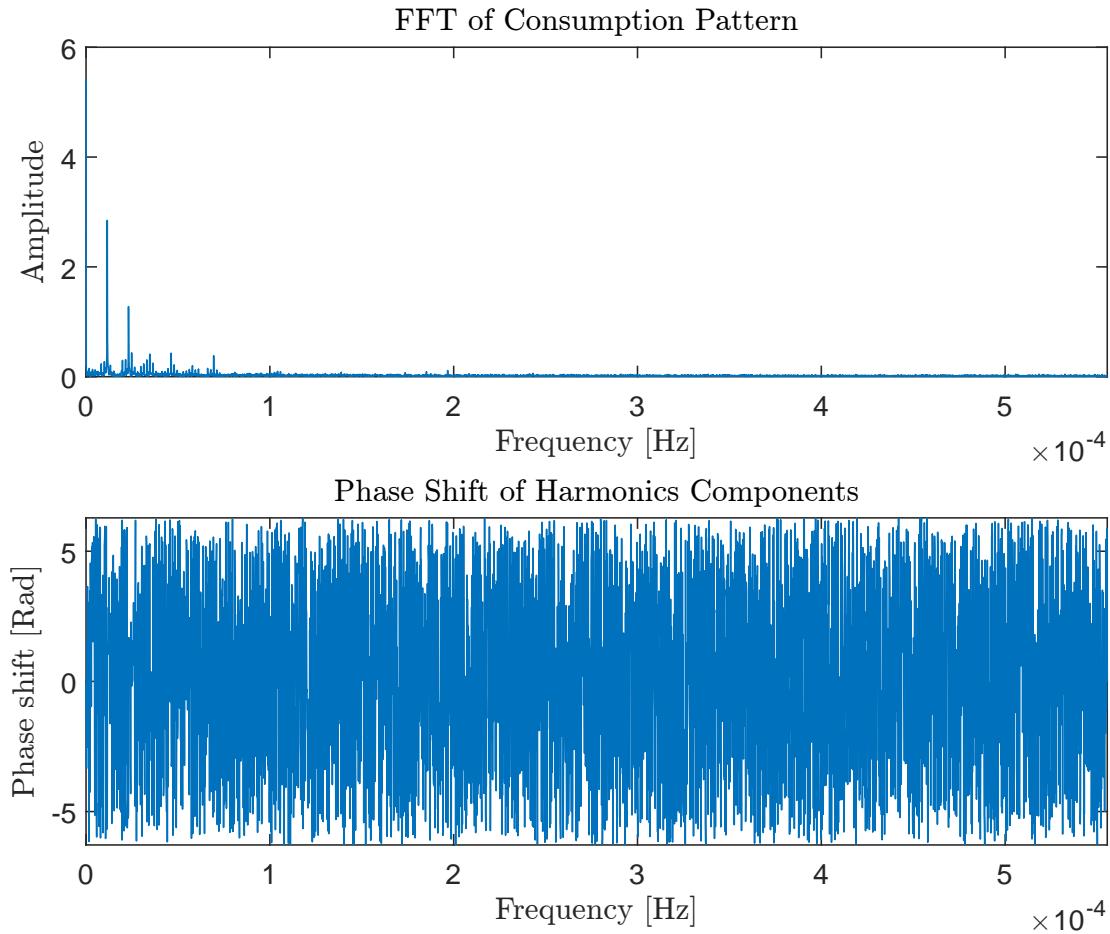


Fig. 15. FFT results on consumption pattern

contributor in the filter. This leads to somewhat arbitrary initial covariance guesses where the consumer model is trusted ten times more than the measurement.

A large white noise is applied to the signal; we assume this noise to be much greater than any practical occurrence, and therefore assume the filter will perform better than simulation results in any real application. The model from which measurements are drawn is also altered to simulate a discrepancy between Kalman model and real measurements. This alteration is introduced by increasing amplitude of the first two sinusoidal components by 20%, and reducing the third by 50%.

First plot in Fig. 17 shows the altered and unaltered model, the Kalman estimate and a leak introduced at $t = 95000$ s. Using covariance values ($R = 0.1, Q = 1$), the Kalman filter tracks the altered model accurately, but it preserves a lot of noise. As such we wish to decrease our trust in measurement; this is also advantageous in terms of detecting the leak, as deviation from the expected behaviour given by the model is concealed because the filter tracks the deviation too well. This phenomenon introduces a tradeoff where we wish to accurately estimate the consumer pattern, i.e. trust measurements, such that minor changes in the pattern are tracked, and still rely enough on the model to make larger deviations stand out. The residual is shown in the second plot of Fig. 17, which depicts this relationship between model accuracy and ability to detect a leak. We want to find a Q value such that the plot shows a clear residual increase during the leak while retaining good tracking during periods without a leak.

Increasing to $Q = 100000$ yields Fig. 18, where the slow dynamics of the consumer pattern is still tracked, and the residual plot shows a noticeable spike when the leak is introduced. The leak gradually develops over one hour, with a maximum magnitude equivalent to 33% of max flow. Quite a large leak is chosen such that the residual tendency can be seen visually; a moving average algorithm is shown to easily find smaller leaks in the next section.

A value of $Q = 100000$ is used in the laboratory after based on simulation results. Algorithmic leakage detection is not implemented, but we present a simple algorithm in simulation. Before proceeding, the leak is slowed down to develop over four hours instead of one, which means it will appear much more similar to the consumer model. The results are shown in Fig. 19, which also clearly shows that the leak is indistinguishable without computer processing. Fig. 20 shows a moving average algorithm which computes the average residual over two hours. The figure clearly shows the average residual increasing after the leak, and a threshold could be set to automatically detect a leak. The beginning of the plot should be disregarded as it is distorted due to lacking prior information for the moving average algorithm, and the Kalman

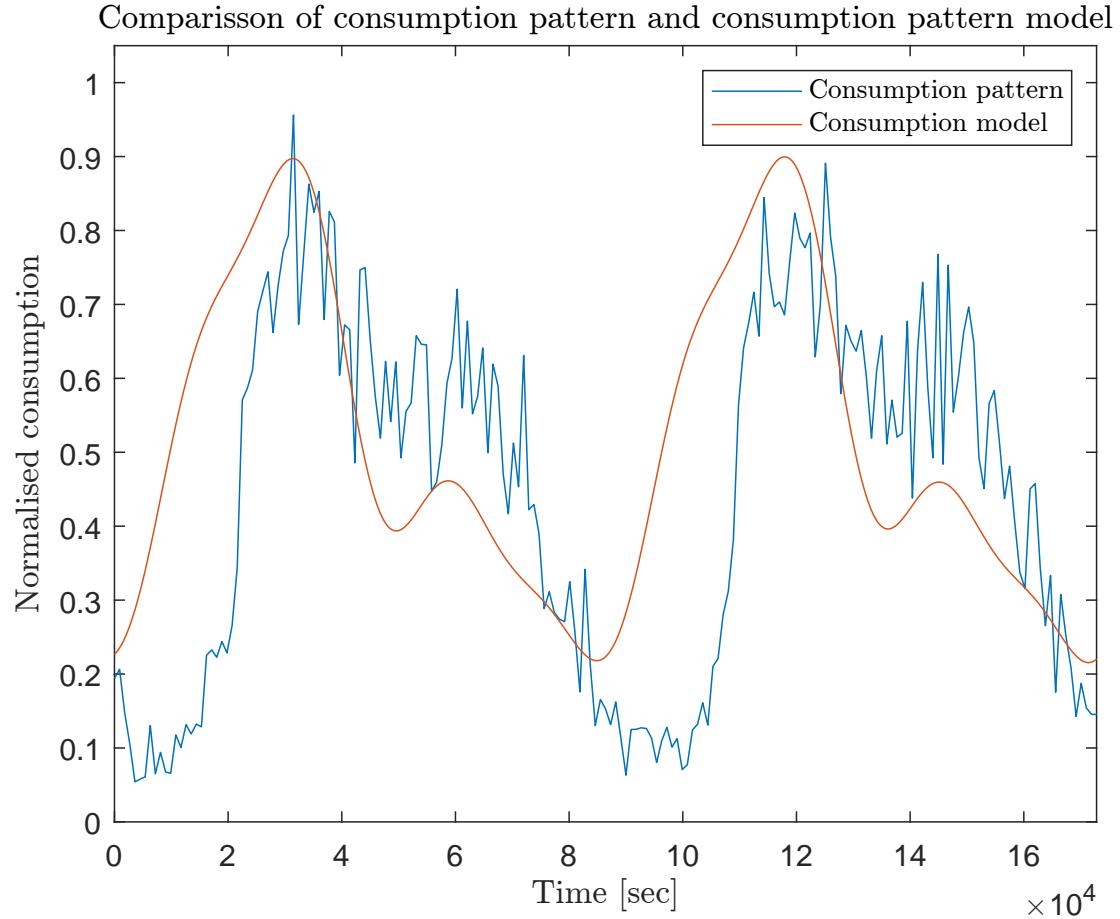


Fig. 16. Comparison of raw historical data, and model

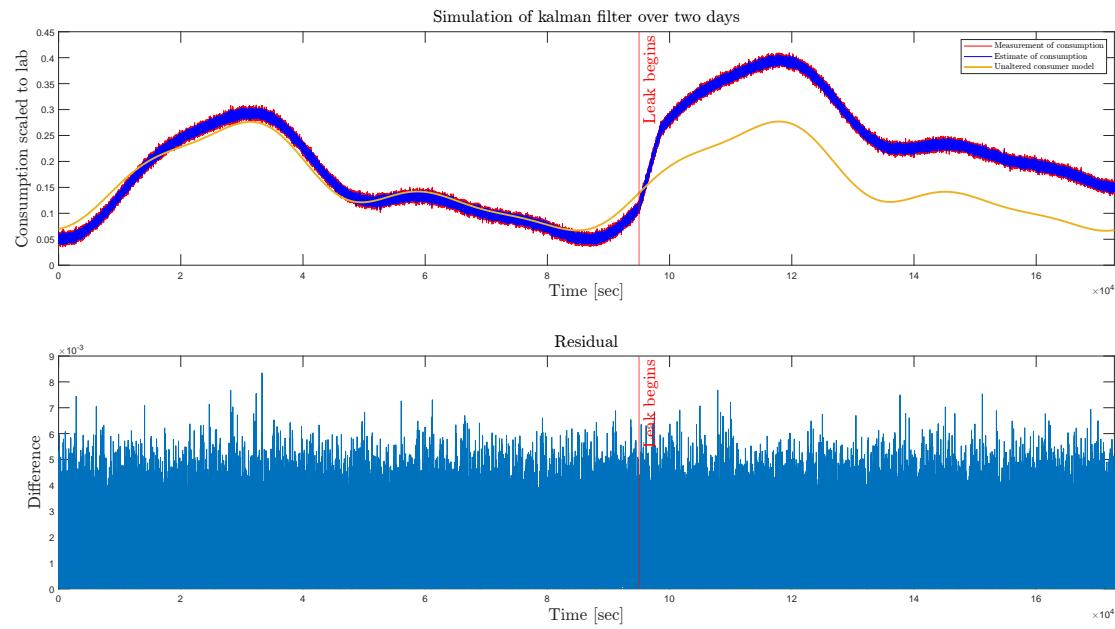


Fig. 17. Simulation of the Kalman filter over two days where $Q = 0.1$ and $R = 1$.

estimate needs time to settle.

In conclusion, we are confident that a leak can be detected using this method. The nature of the leak, in terms of magnitude and development is, however, important. A fast leak should be very easy to detect given the high contrast to the slow dynamics of the consumer pattern. As such, we consider a slowly developing leak. The effectiveness of this

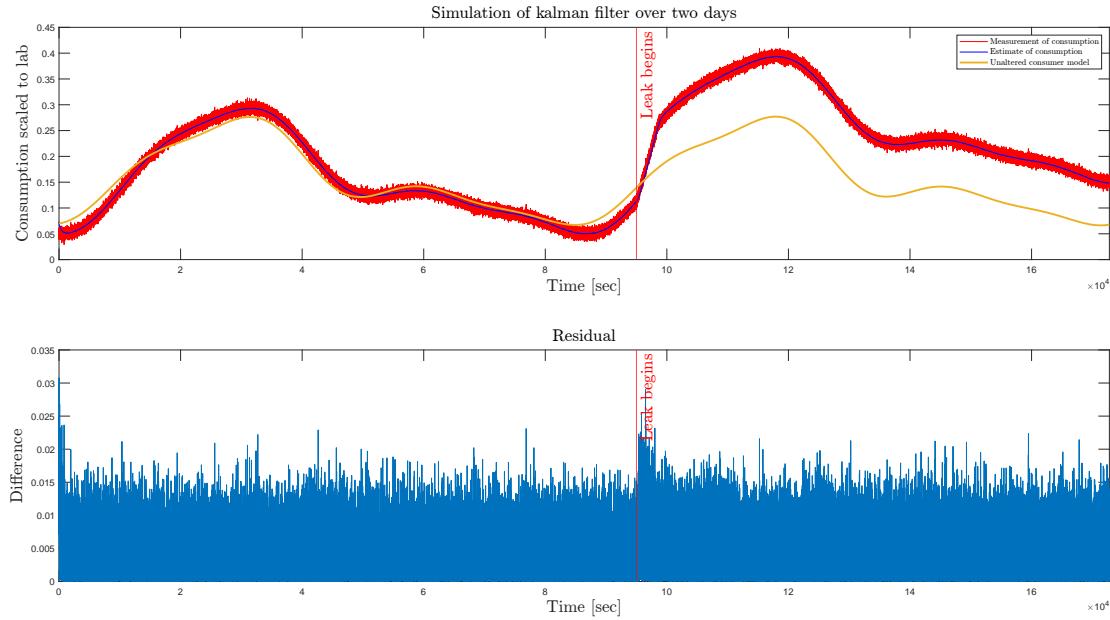


Fig. 18. Simulation of the Kalman filter over two days where $Q = 0.1$ and $R = 100000$.

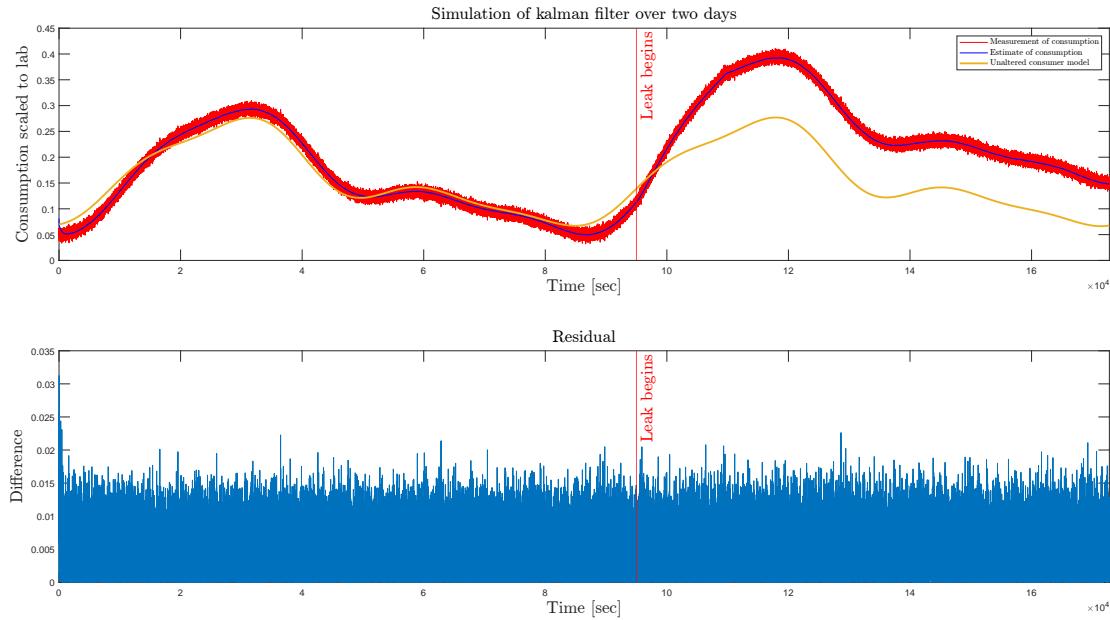


Fig. 19. Leak development slowed down to four hours.

method depends on a series of variables: the correctness of the model, leak properties, and magnitude of measurement noise. We have made assumptions on these variables considering a general proof of concept, which suggests this method will work adequately for many applications, but acknowledge the importance of knowing the relevant variables.

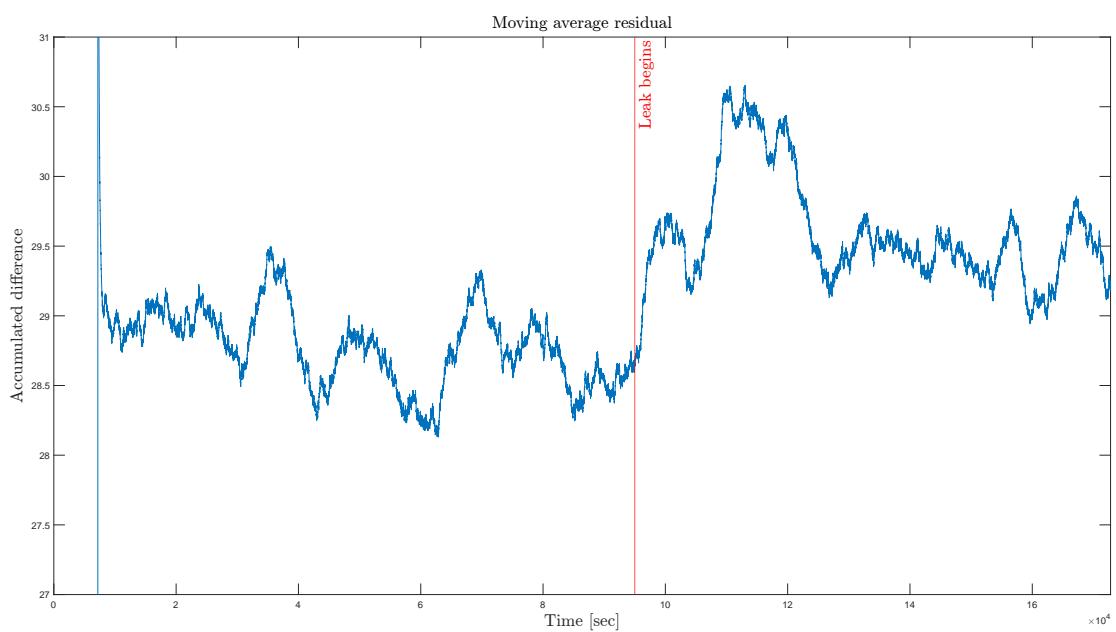


Fig. 20. Moving average algorithm applied to residual

8 MODELLING NETWORK EFFECTS

In the design of both the inner PI controllers in Section 5 and outer VF-LQR in Section 6, we have not accounted for potential effects induced by control over some kind of communication network. While this is a reasonable assumption in the inner loop, which is intended as a local controller that is connected to sensors and actuators over high-reliability, low-latency physical connections, this is *not* a reasonable assumption in the outer loop, which is conceived of as a central controller unit that may physically distal to the local controllers that it commands.

To adequately account for the effects of networked control, we make the following set of assumptions:

- Traffic induced by the control structure is low relative to the available network bandwidth of the system, and thus traffic-induced delay and loss can be neglected entirely.
- The control structure is located in a dense urban environment, and losses on account of this and physical distance cannot be neglected.

A recent study by Anmon Sheth et al. is instructive with respect to the latter assumption [13]. It is clear that losses in an urban environment are significant, and scale exponentially with distance. At distances of 20 km, packet losses of well over 50% may be expected when communication happens over WiFi. This poses an obvious stability risk, depending on which loss-handling protocol is assumed. In this case, we will assume a so-called Try-Once-Discard (TOD) protocol, as proposed by Hu and Yan [14]. This protocol is attractive in combination with VF-LQR as the protocol effectively tells the controller that it has settled at the reference whenever data is unavailable, leading to simple maintenance of the existing control action. Additionally, if this protocol is assumed, and the nominal feedback loop $A - BK$ is stable, [14] allows us to calculate whether the system is stable in the mean-square sense for a given packet loss ratio α . Specifically, the condition is:

$$\mathcal{S}(\alpha A \otimes A + (1 - \alpha)(A - BK) \otimes (A - BK)) < 1 \quad (192)$$

where \mathcal{S} is the spectral radius operator and \otimes is the Kronecker tensor product. This condition follows from the fact that

$$(\alpha A \otimes A + (1 - \alpha)(A - BK) \otimes (A - BK)) \quad (193)$$

can be shown to describe the propagation matrix of the state covariances of the system, and intuitively this must be dissipative - i.e.:

$$\lim_{k \rightarrow \infty} \mathbb{E}[x(k)^2] = 0 \quad (194)$$

for the system to be stable. Furthermore, [14] gives an upper bound Ξ on α for which the system will remain stable, which can be computed as:

$$\begin{aligned} \Xi &= \frac{1}{\|\sigma_+(V)\|_\infty} \\ V &= \begin{bmatrix} (S \otimes \hat{S} + \hat{S} \otimes S)(I - S \otimes S)^{-1} & \hat{S} \otimes \hat{S} \\ (I - S \otimes S)^{-1} & 0 \end{bmatrix} \\ S &= (A - BK) \otimes (A - BK), \quad \hat{S} = A \otimes A - S \end{aligned} \quad (195)$$

where σ_+ is the positive spectral set and $\|\cdot\|_\infty$ is the infinity norm. In the specific case of VF-LQR control of EWRs with dynamics given by Eq. (87), Eq. (195) can be used to show that a such control structure is marginally stable for arbitrary packet loss in the zero limit of control. Specifically, consider that for a VF-LQR control structure and the given dynamics, we have:

$$\begin{aligned} \tilde{A} &= \begin{bmatrix} A & 0 \\ AC & I \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \\ \tilde{B} &= \begin{bmatrix} B \\ CB \end{bmatrix} = \begin{bmatrix} \mathcal{T} & \mathcal{T} \\ \mathcal{T} & \mathcal{T} \end{bmatrix} \\ \tilde{K} &\in \mathbb{R}^{2 \times 2} \end{aligned} \quad (196)$$

The covariance propagation matrix in Eq. (192) can then be written as:

$$(\alpha \tilde{A} \otimes \tilde{A} + (1 - \alpha)(\tilde{A} - \tilde{B}\tilde{K}) \otimes (\tilde{A} - \tilde{B}\tilde{K})) \quad (197)$$

and it follows that:

$$\lim_{\tilde{K} \rightarrow 0} (\alpha \tilde{A} \otimes \tilde{A} + (1 - \alpha)(\tilde{A} - \tilde{B}\tilde{K}) \otimes (\tilde{A} - \tilde{B}\tilde{K})) = (\alpha \tilde{A} \otimes \tilde{A} + (1 - \alpha)\tilde{A} \otimes \tilde{A}) \quad (198)$$

Clearly, the right-hand side of this corresponds to $\tilde{A} \otimes \tilde{A}$ for any value of $\alpha \in \{0 \dots 1\}$. By Eq. (196) we then have:

$$\forall \alpha \in \{0 \dots 1\} : \quad \left(\alpha \tilde{A} \otimes \tilde{A} + (1 - \alpha) \tilde{A} \otimes \tilde{A} \right) = \tilde{A} \otimes \tilde{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (199)$$

The spectrum of which is $\sigma = \{1, 1, 1, 1\}$. Thus *any* EWR is marginally stable for *any* physically meaningful value of packet loss in the zero limit of VF-LQR control, which is a beautifully intuitive result that agrees with the basic physics of a water reservoir.

9 EXPERIMENTAL SETUP AND RESULTS

The proposed control structure is tested in the AAU Smart Water Infrastructure Lab (SWIL). This modular laboratory consists of a number of units pumping units (PU), consumer units (CU), and piping units (PiU) that can be used for small-scale emulation of a real WDN.



Fig. 21. Picture of the AAU SWIL

The network topology in Fig. 3 is emulated via two PUs, a CU with two adjustable one-way valves each emulating a consumer, and a CU with an open bidirectional valve acting as the EWR, interconnected via two PiUs. Geodesic heights are emulated by pressurising the CUs. Experiments run for 12 hours, with consumer demand curves based on real data and compressed to a fundamental frequency of 4 hours such that each experiment corresponds to 3 days. The controller attempts to follow a constant level reference throughout, starting 15mm beneath it. After 4 hours, a system leakage is simulated by fully opening the consumer valves for 30 seconds. After 8 hours, 50% packet loss is introduced in the outer loop, which operates on a simulated TOD protocol. The tank level is seen on Fig. 22, while a snapshot of pump behaviour is seen on Fig. 23. The disturbance estimation scheme and true consumer flows can be seen on Fig. 24, with a zoomed-in view around the leakage seen on Fig. 25.

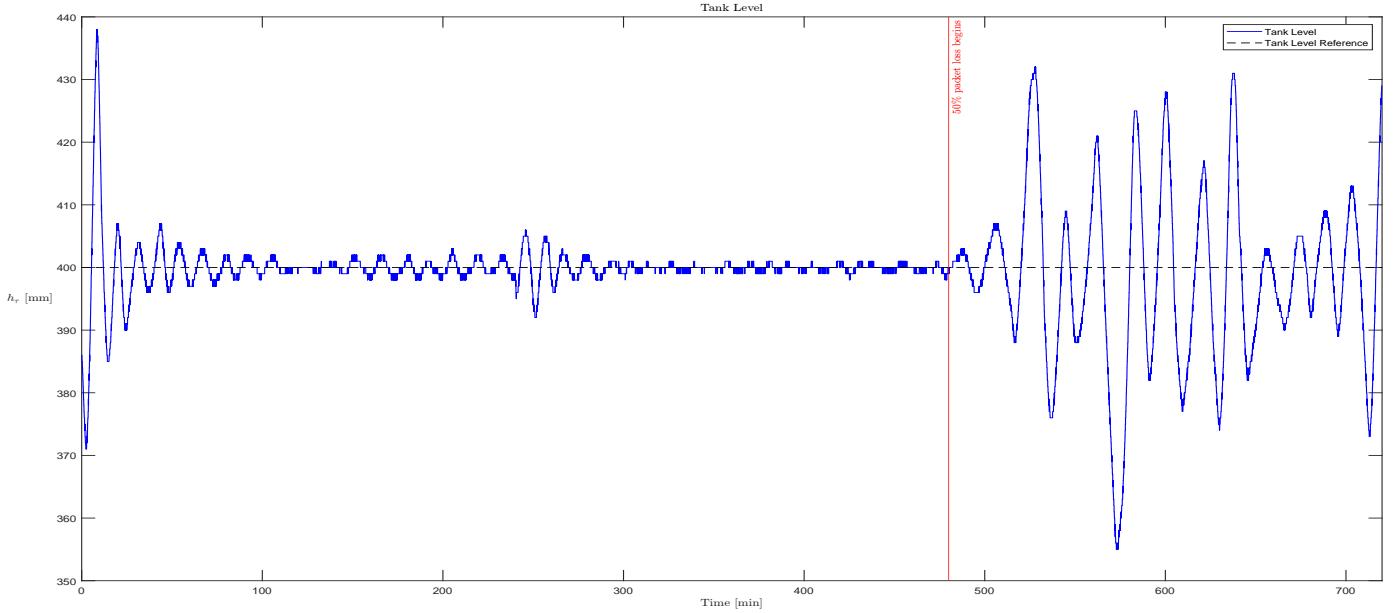


Fig. 22. Outer loop controller performance.

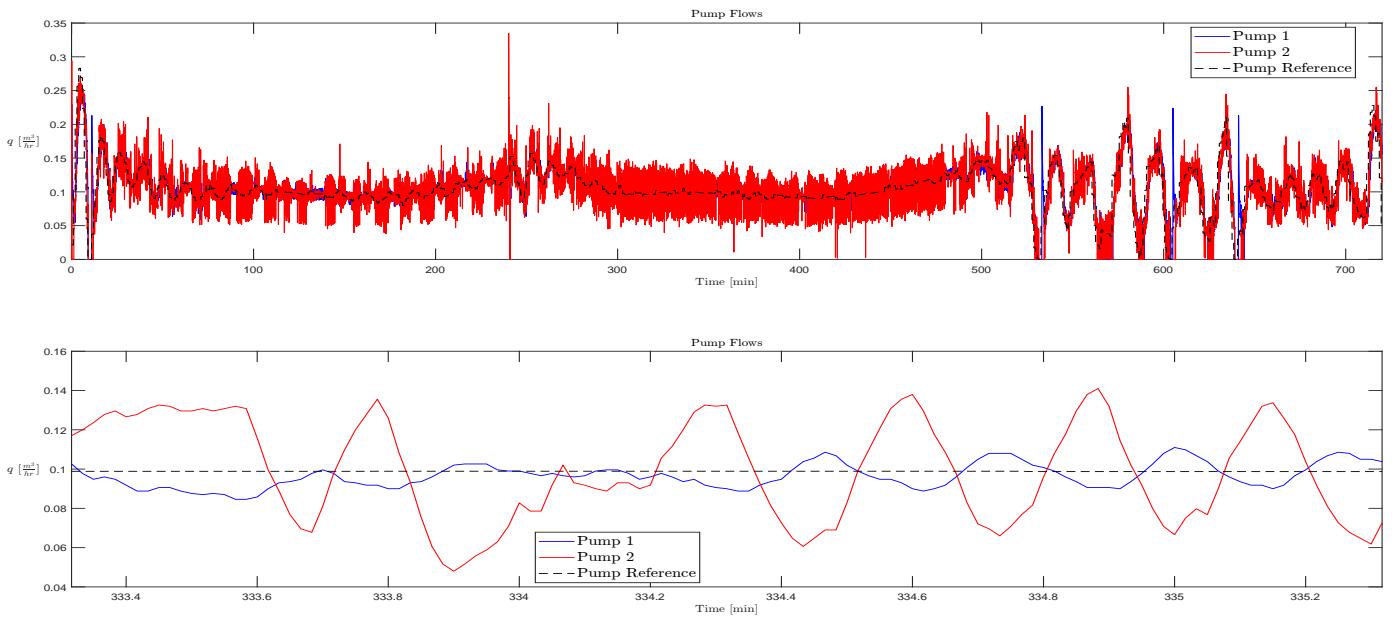


Fig. 23. Inner loop controller performance.

It can be seen on Fig. 22 that the outer loop eventually settles to within the quantisation error of the level sensor, i.e. tracks with zero measurable offset. It remains at this level for several hours, until 50% packet loss is introduced after 8 hours. A small transient is seen at 4 hours, induced by the leakage, from which the system recovers with little issue. We note on Fig. 25 that it is clearly possible to catch this leakage based on evaluation of the residual between measurement and the Kalman filter.

We note furthermore that the system clearly cannot converge to the reference and remain there for an extended period until the Kalman filter has converged to roughly the trajectory of the true consumer disturbance. This is mostly clearly seen by the difference between the short period from $\approx \{120 \dots 160\}$ min, and the longer period from ≈ 300 min until packet loss is introduced:

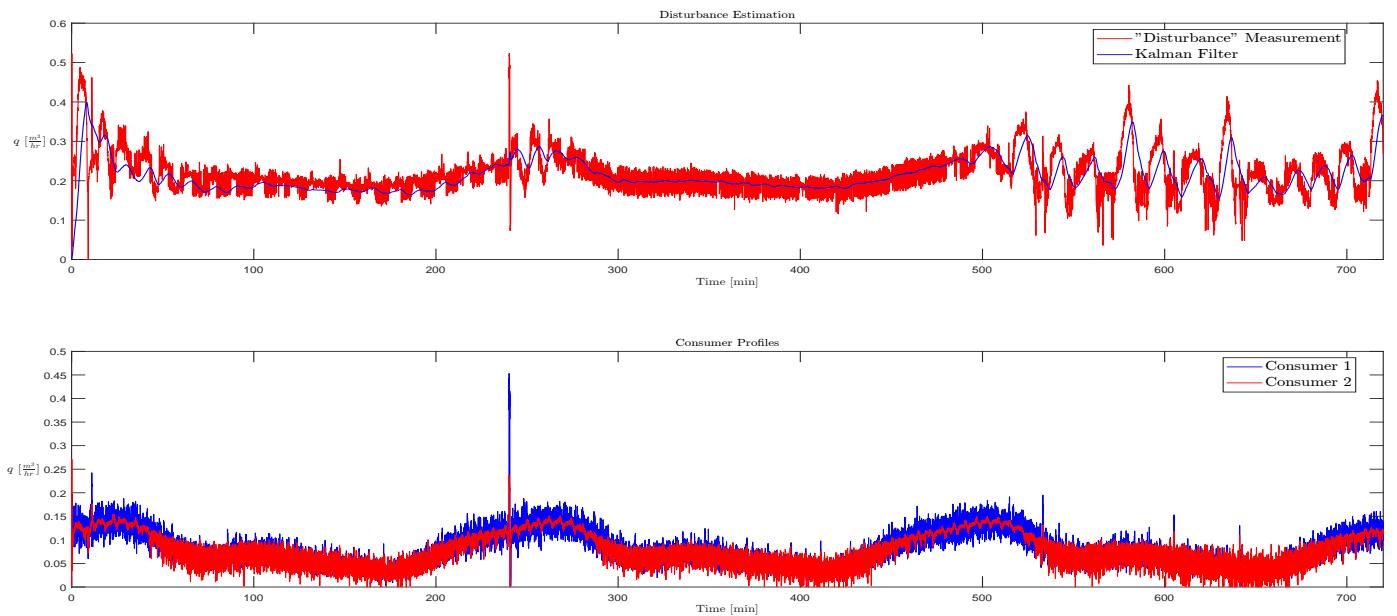


Fig. 24. Kalman filter performance and consumer disturbances.

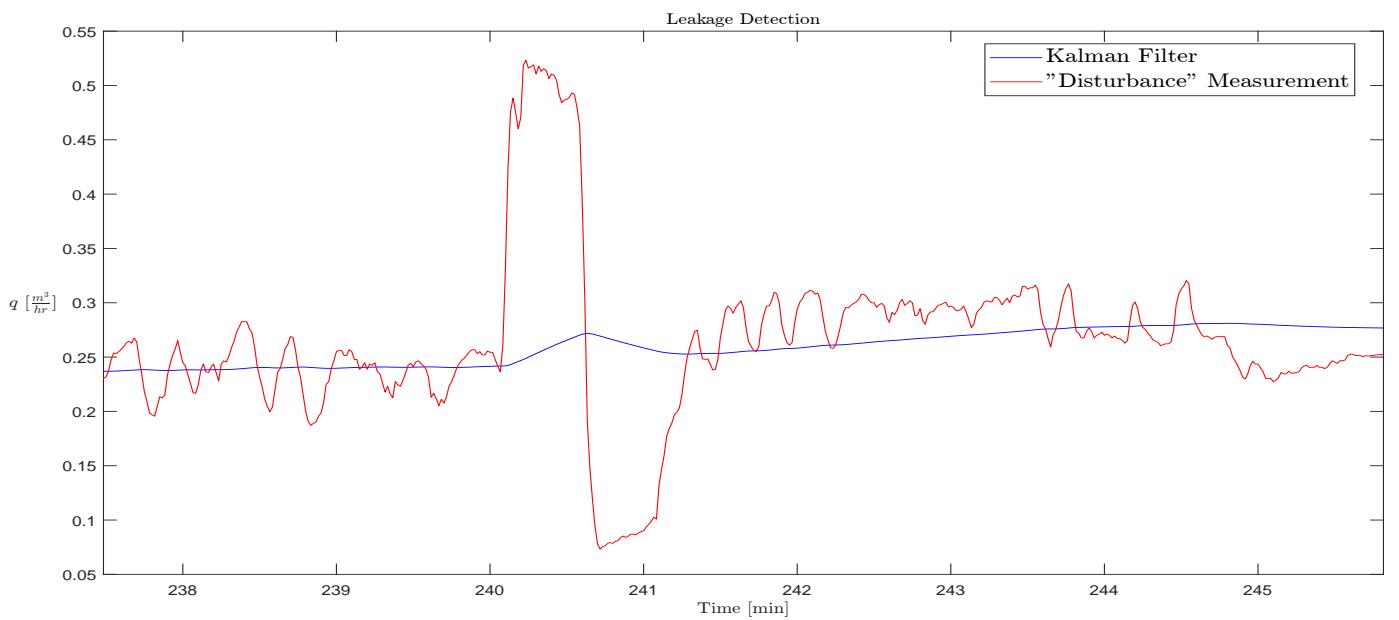


Fig. 25. Kalman filter-based leakage detection.

- In the former case, the system reaches the reference level and remains there for some period of time. However, inspecting the Kalman filter on Fig. 24, it is clear that this has yet to settle to the true disturbance trajectory, and still exhibits significant oscillations. These oscillations eventually drive the outer loop away from the reference again.
- In the latter case, the Kalman filter has converged to approximately the true trajectory of the consumer-induced disturbance. Per Fig. 23 the pump reference trajectory induced by the VF-LQR controller tracks it approximately, rather than oscillating as in the former case.

Interestingly, this performance is possible despite the fact that it is largely impossible to achieve the exact reference specified in the inner loop. Charitably, the pumps tend to achieve the reference in an averaged sense, but as evidenced by Fig. 23, the pumps have very clear cross-coupling that is somehow not reflected in the linearised model, and it is no surprise that two SISO controllers with no decoupling strategy cannot achieve the desired controller performance in this case, but instead give rise to a periodic steady-state solution. We note additionally that the model derived from first-principles is quite inaccurate; PI controller parameters had to be adjusted upwards by a factor of around 4 via manual tuning in the lab. Controller performance in the inner loop is further hindered by the frankly *dreadful* state of the SWIL flow sensors; these have a deadband of almost $0.1 \frac{\text{m}^3}{\text{hr}}$, and thus PI controllers working solely off their measurements cannot actually drive the flows in the system to 0. To compensate for this serious sensor deficiency, we instead implemented forced 0-clamping of the pump speeds whenever the reference delivered by the VF-LQR is 0, with an accompanying complete reset of the PI integrator terms. While this solves the sensor issue, it obviously comes at cost to the performance of the controller.

Tuning of the outer loop is similarly difficult in practice. The primary issue arises from the coupling between the outer loop and the Kalman filter, which is induced by the use of the measurement $\hat{d}_c = d_p - d_\tau$. This coupling means that transients in the outer loop are mapped into the Kalman filter, and vice-versa. Commensurately, these must be tuned extremely carefully to achieve the kind of performance seen in Fig. 22. Specifically, estimator transients induced by the controller must fall outside the passband of the Kalman filter, while the controller transients induced by the estimation error must either be insignificant or fall outside the bandwidth of the controller. A clear example of this can be seen on Figs. 26 and 27.

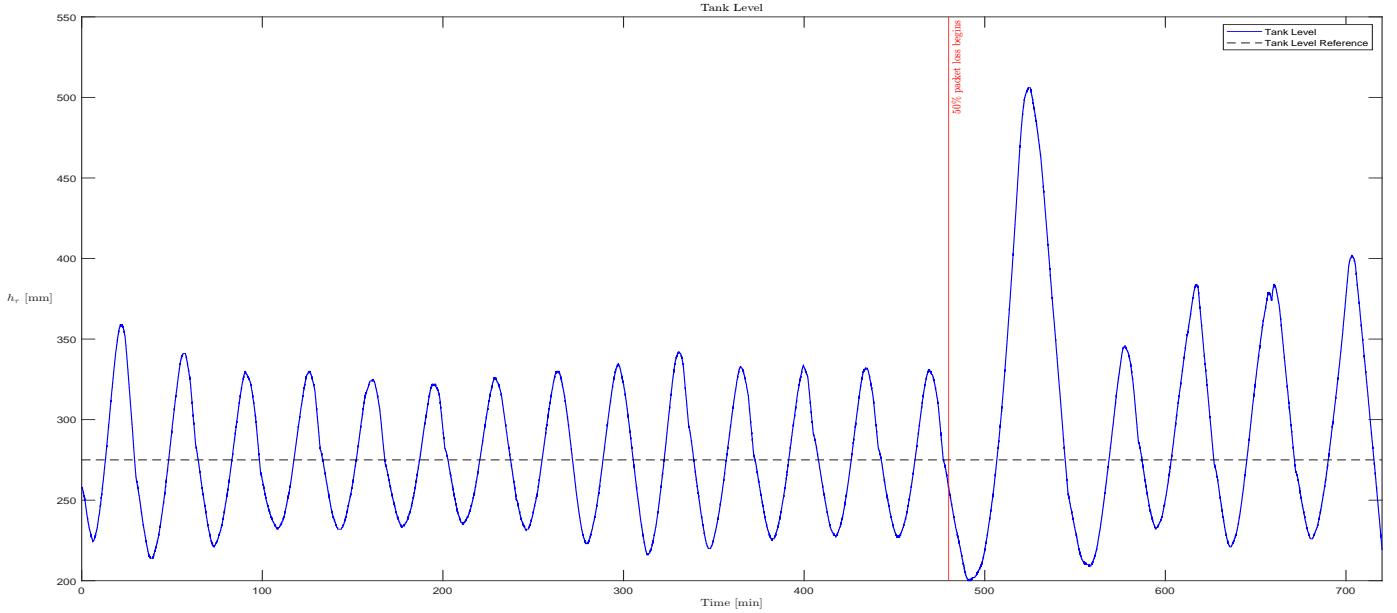


Fig. 26. Outer loop controller performance, poor tuning.

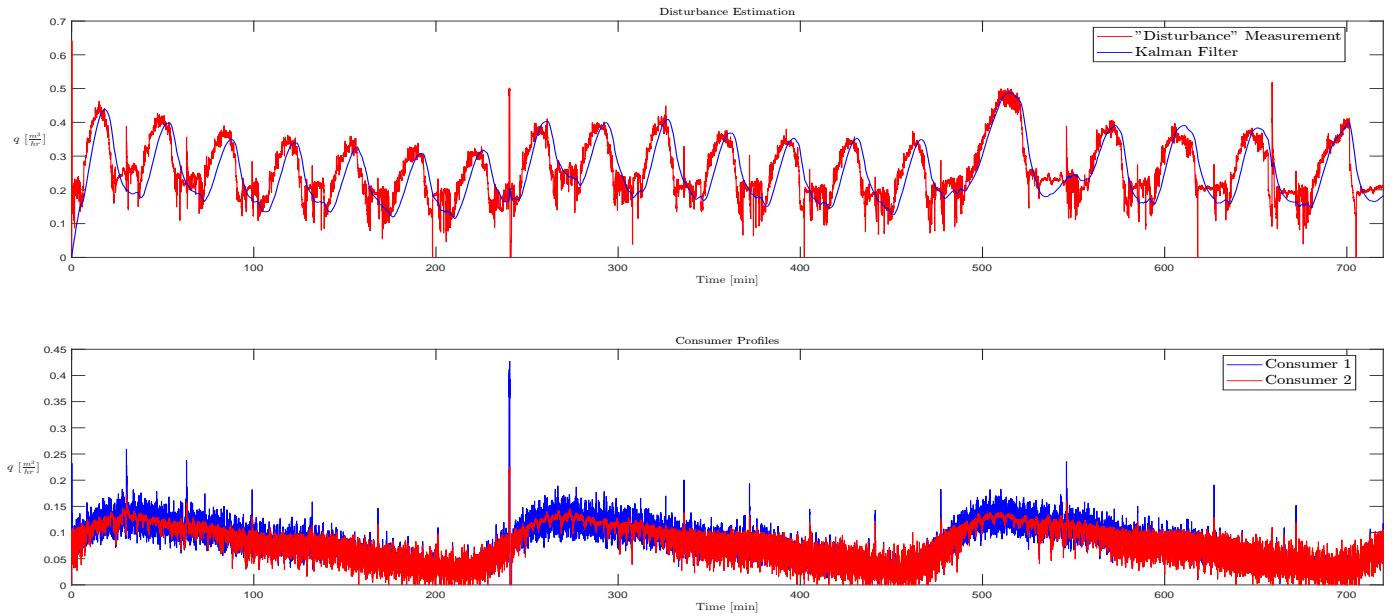


Fig. 27. Kalman filter performance and consumer disturbances, poor tuning.

We see that in this case, the controller and estimator mutually perturb each other, leading to performance that arguably resembles a pair of coupled oscillators. This is unfortunately a quite common result of the proposed control strategy; performance as in Fig. 22 is very much confined to a small "Goldilocks" zone, at least for the given consumer disturbance profiles and size of the tank. We note, however, that the tank in this system is very small compared to what might be expected of a real-world WDN EWR, and therefore it is not unreasonable to consider these results close to worst-case conditions.

10 CONCLUSION

This document has demonstrated the viability of a cascaded structure for control of the water level of an EWR in a small-scale WDN. Models of the two dynamical regimes of the system have been derived from first principles, and the non-linear regime has been linearised to acquire an LTI model. Based on these models, outer and inner control loops have been designed, employing respectively an optimal VF-LQR strategy with disturbance rejection in the outer loop, and a pair of PI controllers in the inner loop. Furthermore, a harmonic model of consumer behaviour has been developed based on a truncated Fourier series and real consumer data, and an LTI Kalman filter has been designed based on this model. Finally, an analytical result regarding the mean-square stability of an arbitrary EWR under VF-LQR control has been derived.

Experimental results validate the design of the controller, which performs excellently and eventually settles at the specified level reference, and remains stable, in an averaged sense, around the reference after packet loss is introduced. Results have also revealed that performance of the designed control and estimation structure is very sensitive to tuning of respectively the outer loop and Kalman filter, as these may interact as coupled oscillators due to the choice of measurement in the Kalman filter. However, this is expected to be a limited issue in more realistic WDNs - the tank in the laboratory WDN is uncharacteristically small and thus has very fast dynamics relative to the dynamics of the network proper. Furthermore, results reveal that it is likely impossible to design high-quality controllers for the pumps based on simple model linearisation and naïve SISO techniques, as cross-coupling between the pumps appears to be lost during the linearisation process, and further study including the application of MIMO control techniques and possibly non-linear or data-driven methods is encouraged by authors.

REFERENCES

- [1] S. S. Rathore, "Optimal Control in Water Distribution Network," 2019.
- [2] M. Mølgaard and B. G. Pétursson, "Energy Optimization of Water Distribution Networks," Master's thesis, Aalborg University, 2015.
- [3] T. N. Jensen, "Modelling of open hydraulic networks," pp. 1–8.
- [4] N. Deo, *Graph theory with applications to engineering and computer science*. Dover Publications, Inc., 1974. [Online]. Available: <https://store.doverpublications.com/0486807932.html>
- [5] L. Perko, *Differential Equations and Dynamical Systems*, ser. Texts in Applied Mathematics. New York, NY: Springer New York, 1968, vol. 22, no. 102. [Online]. Available: <http://link.springer.com/10.1007/978-1-4613-0003-8>
- [6] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 2nd ed. John Wiley and Sons Ltd, 2005, vol. 1.
- [7] D. Liberzon, *Calculus of Variations and Optimal Control Theory*. Princeton University Press, oct 2019.
- [8] G. Pannocchia and J. Rawlings, "The velocity algorithm LQR: a survey," *Texas-Wisconsin Modeling and Control Consortium*, no. September, pp. 1–21, 2001. [Online]. Available: <http://jbrwww.che.wisc.edu/tech-reports/twmcc-2001-01.pdf>
- [9] D. D. Ruscio, "Discrete LQ optimal control with integral action: A simple controller on incremental form for MIMO systems," *Modeling, Identification and Control*, vol. 33, no. 2, pp. 35–44, 2012.
- [10] A. K. Singh and B. C. Pal, "An extended linear quadratic regulator for LTI systems with exogenous inputs," *Automatica*, vol. 76, pp. 10–16, feb 2017.
- [11] R. Doraiswami, C. Diduch, and M. Stevenson, *Modeling and Identification of Physical Systems*, 2014.
- [12] S. M. Bozic, "Digital and Kalman Filtering: An Introduction to Discrete-Time Filtering and Optimum Linear Estimation," 1994.
- [13] A. Sheth, S. Nedevschi, R. Patra, S. Surana, E. Brewer, and L. Subramanian, "Packet loss characterization in wifi-based long distance networks," *Proceedings - IEEE INFOCOM*, pp. 312–320, 2007.
- [14] S. Hu and W. Y. Yan, "Stability robustness of networked control systems with respect to packet loss," *Automatica*, vol. 43, no. 7, pp. 1243–1248, jul 2007.

11 APPENDIX

This section is concerned with the appendices of the WorkSheets

11.1 PumpCoefficients

A model of the pump is based on the following relationship between the pressure drop over the pump and the pump speed, flow through the pump and the pump coefficients.

The full pump model:

$$\Delta p_{pump} = a_0 \cdot \omega^2 + a_1 \cdot q \cdot \omega + a_2 \cdot q^2 \quad (200)$$

The pump coefficients are estimated from the pump performance curves from the data-sheets. These describe the exact relationship described in Eq. (200).

The pressure drops were read at the following flows:

$$q = [0 \ 0.5 \ 1 \ 1.5 \ 2 \ 2.5 \ 3 \ 3.5] \quad (201)$$

At these flows the pressure drop across the pump were read from the y-axis at three different pump speeds, namely 80%, 69% and 48%. The recorded pressure drops were the following:

$$p = \begin{bmatrix} p_{80} \\ p_{69} \\ p_{48} \end{bmatrix} = \begin{bmatrix} 5.57 & 5.85 & 5.9 & 5.7 & 5.1 & 4.25 & 3.3 & 2.4 \\ 4.1 & 4.2 & 4.2 & 4 & 3.55 & 3 & 2.3 & 1.55 \\ 1.75 & 1.85 & 1.7 & 1.3 & 0.95 & & & \end{bmatrix} \quad (202)$$

Only 5 pressure readings were made from the 48% pump speed.

The coefficients are found by finding the least square solution to following equation:

$$Ax = b \leftrightarrow x = (A^T A)^{-1} A^T b \quad (203)$$

where A is a matrix defined as follows:

$$A = [\omega^2 \ q \cdot \omega \ q^2] \quad (204)$$

With number of rows corresponding to the amount of data points in p , i.e. the first 8 rows corresponds to measurements made for $\omega = 80$, the next 8 rows corresponds to measurements made for $\omega = 69$ etc.

and $b = [p_{80} \ p_{69} \ p_{48}]^T$ from Eq. (202).

The solution yields the following coefficients:

$$\begin{aligned} a_0 &= 0.0001 \\ a_1 &= 0.0004 \\ a_2 &= -0.0323 \end{aligned} \quad (205)$$

We confirm our pump model by comparing it to actual data points on the pump curve from the data sheet.

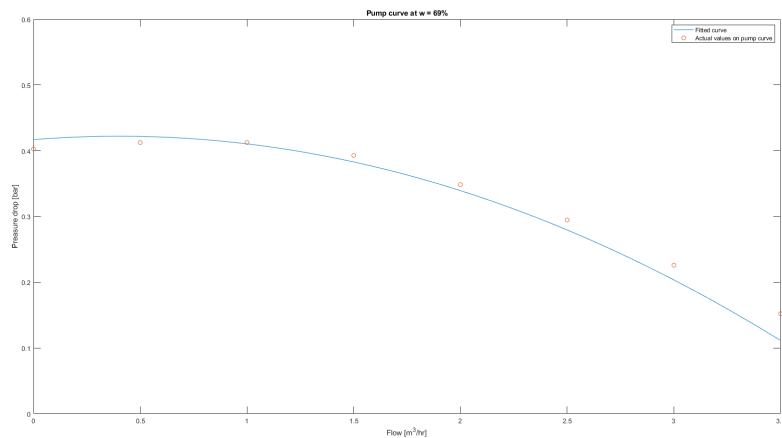


Fig. 28. Comparing model of pump curve with actual values from data sheet

12 MODEL LEGEND

Model:

$$\Phi \mathcal{J} \Phi^T \dot{q} = -\Phi \left(\lambda(q_n) + \mu(q_n, OD) + \alpha(q_n, \omega) \right) + \Psi(\bar{h} - \mathbf{1}h_0) + \mathcal{I}(p_\tau - \mathbf{1}p_0)$$

where p_τ evolves according to:

$$\dot{p}_\tau = -\mathcal{T} \dot{d}_\tau, \quad \mathcal{T} = \text{diag}(\tau_i)$$

and the matrices Φ, Ψ, \mathcal{I} are defined as:

$$\Phi \triangleq \begin{bmatrix} I & -\bar{H}_C^T \bar{H}_T^{-T} \\ 0 & \bar{F}^T \bar{H}_T^{-T} \\ 0 & \bar{G}^T \bar{H}_T^{-T} \end{bmatrix}, \quad \Psi \triangleq \begin{bmatrix} 0 \\ \bar{F}^T \\ \bar{G}^T \end{bmatrix}, \quad \mathcal{I} \triangleq \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix}$$

Φ and Ψ contains:

F	Open-node matrix - Extracts the pumps and consumer nodes	Dim = $n \times e$
G	Tank-node matrix/vector - Extracts the tank node.	Dim = $n \times 1$ (one tank node)

$$F = \begin{array}{c} d_{f_1} \quad d_{f_2} \quad d_{f_3} \quad d_{f_4} \\ \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right], \quad G = \begin{array}{c} d_{\tau_1} \\ \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right] \end{array} \end{array}$$

$$\bar{F} = F \setminus F_{6*} \wedge \bar{G} = G \setminus G_{6*}$$

\mathcal{J} is the mass inertia of the water in all pipes [kg/m^4]. Dim = $m \times m$

$\lambda(q_n)$ is the pressure drop across all components due to pipe friction [Pa]

$$\lambda(q) = \left(f \cdot \frac{8 \cdot L \cdot q^2}{\pi^2 \cdot g \cdot D^5} + k_f \cdot \frac{8 \cdot q^2}{\pi^2 \cdot g \cdot D^4} \right) \cdot g \cdot \rho$$

f	Friction constant	[·]
L	Length of pipe	[m]
g	Gravitational acceleration	[$\frac{m}{s^2}$]
D	Diameter of pipe	[m]
ρ	Density of water	[$\frac{kg}{m^3}$]

$\mu(q_n, OD)$: is the pressure drop across all components in the system due to valve friction [Pa]

$$\mu(q, OD) = \frac{1}{K_{valve}(OD)^2} \cdot |q| \cdot q$$

OD	Opening degree	[%]
K_{valve}	Constant: flow in $\frac{m^3}{h}$ at OD = 100 % at 1 bar Δp	[·]

$\alpha(q_n, \omega)$: Pressure drop across the pump (Δp_p) [Pa]

$$\alpha(q, \omega) = a_0 \cdot \omega^2 + a_1 \cdot \omega \cdot q - a_2 \cdot |q| \cdot q$$

ω	Pump angular velocity	[$\frac{rad}{s}$]
$a_0 - a_2$	tuple of coefficients that describe the pump's characteristic curve	[·]

\bar{h} : Height of all nodes excluding the reference node [m]

h_0 : The reference node height [m]

p_τ : The pressure at the bottom of the tank [Pa].

p_0 : The reference node pressure [Pa]