Bare Demo of IEEEtran.cls for IEEE Conferences

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

Abstract

The abstract goes here.

1 Introduction

Put introduction here

2 Graph Theory

2.1 System modelling

This section describes the interconnected component in a water distributed network (WDN) using Graph Theory. Furthermore thee component classes will be examined, ie. pipes, pumps and valves.

2.2 Graph Theory

When using graph theory as an analytical tool the incident matrix H comes in handy when describing the connection between edges and nodes. The H matrix is defined as follows:

2.2.1 The Incident Matrix

$$H_{i,j} = \begin{cases} -1 & \text{If the } j^{th} \text{ edge enters the } i^{th} \text{ node} \\ 0 & \text{If the } j^{th} \text{ edge is not connected to the } i^{th} \text{ node} \\ 1 & \text{If the } j^{th} \text{ edge s leaving the } i^{th} \text{ node} \end{cases} \tag{1}$$

2.2.2 Spanning Tree

The spanning tree is the connected graph but with no loops, i.e you can not find a route around the graph where you start and end in the same node without entering a node more than one time. When adding one chord to the graph, exactly one loop is created.

2.2.3 The Loop Marix

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

The loop-matrix can be calculated in one of two ways. The first one is shown in the equation below

$$B = \begin{bmatrix} I & -\bar{H}_C^T \cdot \bar{H}_T^{-T} \end{bmatrix} \tag{2}$$

With $\bar{H}_C$ being the matrix containing only chord columns of the incident matrix. The same goes for $\bar{H}_T$, containing only the non-chord columns. The bar indicates that the row corresponding to the reference node has been removed from the incident matrix.

The other method is more graphical but can be formulated as follows:

Adding one chord to the spanning tree, creates exactly one loop. Writing up the edges in the loop, with signs corresponing to the incident matrix creates a row in the loop matrix B.

2.3 Simplified system

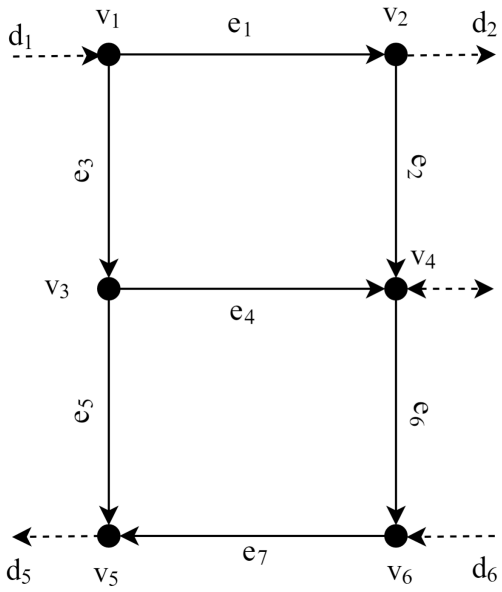

Fig. 1. Graph of simplified WDN network [1]

When applying the rules shown above for the simplified graph model of the WDN the incident matrix in eq. (3).

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{3}$$

The reduced incident matrix by taking an arbitrary vertex as a reference, and removing that vertex-row from eq. (3). We chose the 4^{th} vertex, which results in the following reduced incident matrix:

$$\bar{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{4}$$

Chords and edges of the spanning tree

$$\begin{aligned} E_C &= \{e_1, e_4\} \\ E_T &= \{e_2, e_3, e_5, e_6, e_7\} \end{aligned} \tag{5}$$

$$B = \begin{bmatrix} 1 & 0 & 1 & -1 & -1 & 1 & 1 \\ 0 & 1 & 0 & 0 & -1 & 1 & 1 \end{bmatrix} \tag{6}$$

2.4 Detailed system

In this section we present a graph of the water distribution network.

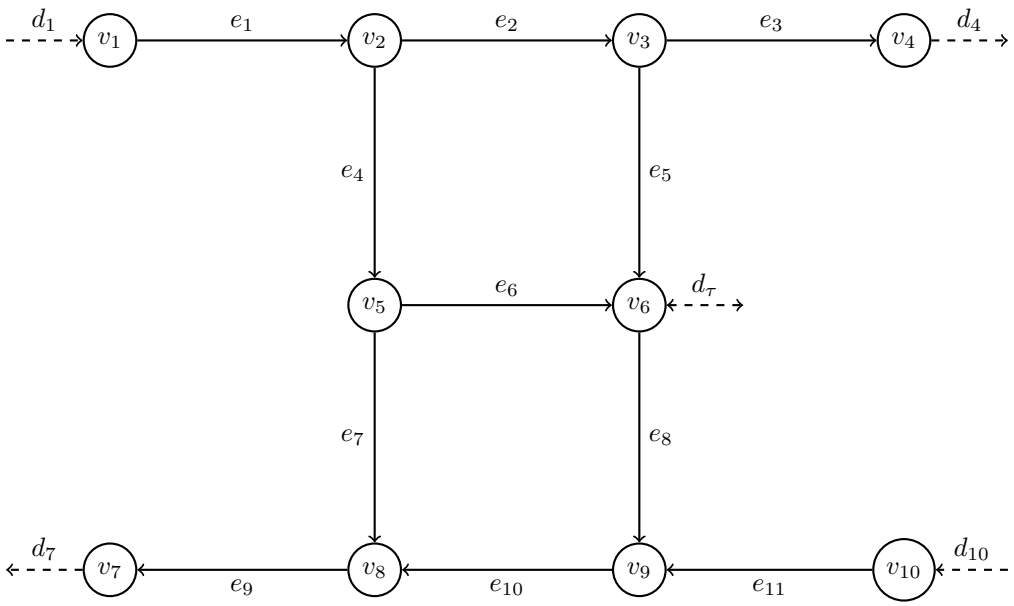

Fig. 2. Graph network of water distributed network

d_1 and d_{10} represent the waterflow into the system from the pumps, which are represented by edge e_1 and e_{11} respectively. d_4 and d_7 represent the waterflow out of the system from the valves, which are represented by edge e_3 and e_9 respectively. d_τ represent the waterflow in an out of the tank. The incident matrix for the graph on **??** is shown blow in eq. (12).

$$H = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \end{array} \begin{array}{c} \begin{array}{ccccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} & e_{11} \end{array} \\ \left[\begin{array}{ccccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array} \tag{7}$$

The chords and the spanning tree is chosen from **??** [1]

$$E_C = \{e_2, e_6\}$$

$$E_T = \{e_1, e_3, e_4, e_5, e_7, e_8, e_9, e_{10}, e_{11}\}$$

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

By choosing the xxx node as a reference, and removing it from the incident matrix we obtain:

$$\bar{H} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \end{array} \begin{array}{c} \begin{array}{ccccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} & e_{11} \end{array} \\ \left[\begin{array}{ccccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array} \tag{8}$$

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

$$F = \begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \\ d_9 \\ d_{10} \end{array} \begin{array}{c} \begin{array}{cccc} d_{f_1} & d_{f_2} & d_{f_3} & d_{f_4} \end{array} \\ \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \end{array}, \qquad G = \begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \\ d_9 \\ d_{10} \end{array} \begin{array}{c} \begin{array}{c} d_{\tau_1} \end{array} \\ \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{array}\right] \end{array} \tag{9}$$

with their reference-respective equivalents given by:

$$\bar{F} = F \setminus F_{6\star} \wedge \bar{G} = G \setminus G_{6\star} \tag{10}$$

where the notation $X_{6\star}$ denotes the entire 6th row[2] 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 $d \vee \bar{d}$.

2.5 Implementation of Detailed system

The H matrix:

2. Correspondingly, $X_{\star 6}$ would denote the entire 6th column

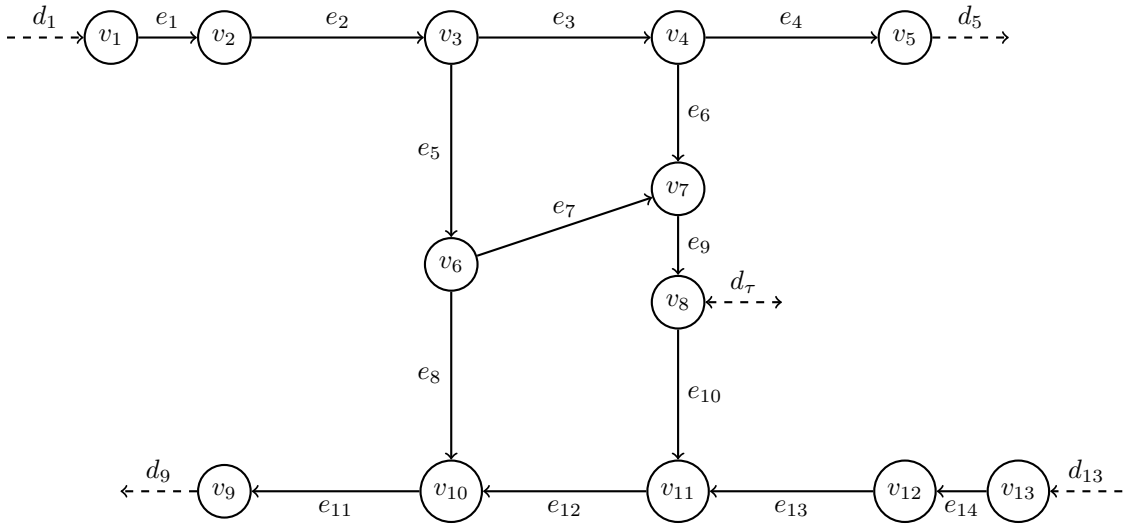

Fig. 3. Graph network of implemented water distributed network

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

The reduced incident matrix $\bar{H}$ for the system given by eq. (11) is obtained by choosing the node 13 as reference:

$$\bar{H} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \\ v_{11} \\ v_{12} \end{array} \begin{array}{c} \begin{array}{cccccccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} & e_{11} & e_{12} & e_{13} & e_{14} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \end{array} \tag{12}$$

The B matrix (not even started):

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

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:

$$\begin{bmatrix} \Delta p_k \\ q_k \end{bmatrix} = \begin{bmatrix} p_i - p_j \\ q_k \end{bmatrix} \tag{13}$$

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:

$$\Delta \mathrm{p_k} = J_k \cdot \dot{q_k} + \lambda_\mathrm{k}(q_k) - \Delta \mathrm{z_k} \tag{14}$$

Δp_k	The differential pressure across the k^{th} component	[Pa]
J_k	Is the mass inertia of the water in the k^{th} pipe	$[\mathrm{kg/m^4}]$
q_k	is the flow of water trough the k_{th} pipe	$[\mathrm{m^3/s}]$
$\lambda_\mathrm{k}(q_k)$	is the drop in pressure due to friction in the k^{th} pipe	[Pa]
$\Delta \mathrm{z_k}$	is the drop in pressure due to geodesic level	[Pa]

The mass inertia of water can be describes as follows:

$$J = \frac{L \cdot \rho}{A} \tag{15}$$

L	is the length of the pipe	[m]
ρ	is the density of water	$[\mathrm{kg/m^3}]$
A	is the cross sectional area of water	$[\mathrm{m^2}]$

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

The causes of flow friction $\lambda_\mathrm{k}(q_k)$ are surface resistance h_f and form resistance h_m. The surface resistance can be describes with the Darcy-Weisbach equation:

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

h_f	is the head loss from surface resistance	[m]
f	is the pipe friction factor	$[\cdot]$
D	is the pipe diameter	[m]
g	is the gravitational constant	$[\mathrm{m/s^2}]$

Under the assumption of turbulent flow f can be is given by:

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

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

The form resistance can be given by the following:

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

h_m	is the head loss from form resistances	$[\mathrm{m}]$
k_f	is coefficient form loss	$[\cdot]$

The drop in pressure due to geodesic level difference:

$$\Delta z_k = \rho \cdot g \cdot \Delta h_k \tag{19}$$

Δh_k	is the level difference across the terminals of the k^{th} pipe	$[\mathrm{m}]$

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.

$$p = \rho \cdot g \cdot h \tag{20}$$

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

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

Inserting and reducing into eq. (14)

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

The absolute value of one of the flow component 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:

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

We then express the conductivity function $K_{valve}(OD)$ as, 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.:

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

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

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

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:

$$\Delta p = a_0 \cdot \omega^2 + a_1 \cdot \omega \cdot q - a_2 \cdot |q| \cdot q \tag{26}$$

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:

$$p_\tau \propto \zeta \tag{27}$$

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

$$\dot{V} = d_\tau \tag{28}$$

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

$$\dot{\zeta} \propto \dot{V} \Leftrightarrow \dot{\zeta} \propto d_\tau \tag{29}$$

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

$$\dot{p}_\tau \propto d_\tau \tag{30}$$

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

$$\dot{p}_\tau = -\tau \cdot d_\tau \tag{31}$$

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	[$\mathrm{Pa/m^3}$]
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.	[$\mathrm{m^3/s}$]

The tank parameter is given by:

$$\tau = \rho \cdot g \frac{1}{A} \tag{32}$$

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 is obeys Kirchoff's Nodal Law (KNL), i.e. that:

$$Hq = d \tag{33}$$

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

$$B\Delta p = BH^T p = 0 \wedge B\Delta h = BH^T h = 0 \tag{34}$$

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.:

$$d_n = -\sum_{i=1}^{n-1} d_i \tag{35}$$

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

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

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:

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

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:

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

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:

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

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:

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

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

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

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

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

where $\mathcal{J}, \lambda, \mu, \alpha$ are as defined in section 3[3] and q is the vector of edge flows. We now invoke KML as per eq. (34) on eq. (42), obtaining:

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

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

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

We now develop this expression further via the definition of B in eq. (2) and via a chord/tree-partition of $\mathcal{J}$ such that:

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

$$B\mathcal{J}B^T\dot{q}_C + \begin{bmatrix} I_C & -\bar{H}_C^T\bar{H}_T^{-T} \end{bmatrix} \begin{bmatrix} \mathcal{J}_C & 0 \\ 0 & \mathcal{J}_T \end{bmatrix} \begin{bmatrix} 0 \\ \bar{H}_T^{-1} \end{bmatrix} (\bar{F}\dot{d}_f + \bar{G}\dot{d}_\tau) = -B(\lambda(q) + \mu(q) + \alpha(q)) \tag{45}$$

This simplifies to:

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

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:

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

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

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

We now exploit that:

$$H_T\bar{H}_T^{-1} = \bar{H}_T^{-T}H_T^T \tag{49}$$

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

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

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

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

which implies that:

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

and finally by restating eq. (33) as:

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

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

Dimensions doesnt add up (H-bar on right side have dim 12x12 and [pbar p0]is 13x1) in

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

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 $p_\tau - \mathbf{1}p_0$:

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

which analogously to eq. (54) becomes:

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

Now, defining the state vector $q_n = \begin{bmatrix} q_C^T & d_f^T & d_\tau^T \end{bmatrix}^T$ 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:

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

where p_τ evolves according to:

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

and the matrices $\Phi, \Psi, \mathcal{I}$ are defined as:

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

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.

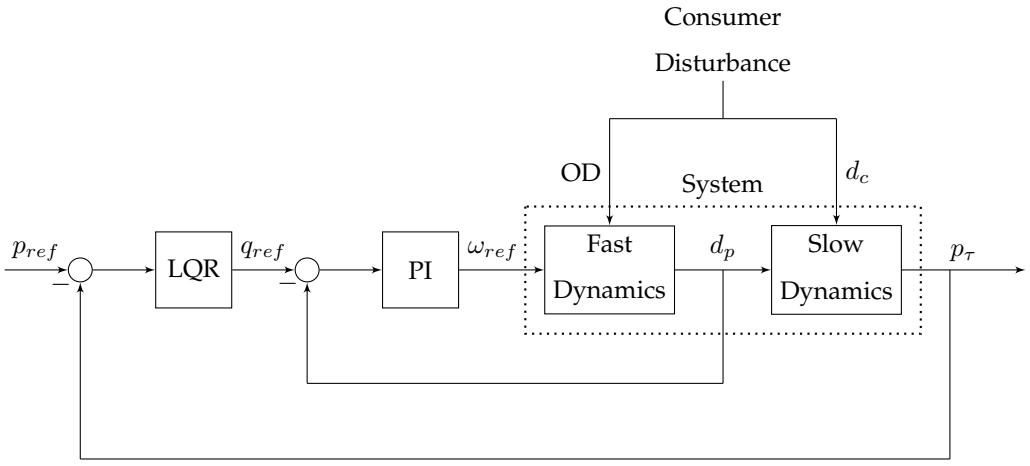

Fig. 4. Control Structure

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 hyberbolic:

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

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 nonlinear system, there exists a *linear* system that behaves in a generally identical manner when evolved in time. 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 may not 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 nonlinear 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 allows us to rewrite eq. (57) as:

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

We now make the additional observation that $\Psi(\bar{h} - \mathbf{1}h_0)$ is constant for all time, $\mathcal{I}(p_\tau - \mathbf{1}p_0)$ do 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:

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

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:

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

where:

$$x_0 = \{q_0, OD_0, \omega_0, p_\tau\} \tag{64}$$

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

$$\tilde{q_n} = q_n - q_0 \tag{66}$$

$$\tilde{OD} = OD - OD_0 \tag{67}$$

$$\tilde{\omega} = \omega - \omega_0 \tag{68}$$

$$\tilde{p_\tau} = p_\tau - p_{\tau_0} \tag{69}$$

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

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

$$\left.\frac{\partial \Omega}{\partial OD}\right|_{x_0} = -|q_0| q_0 \frac{2}{K_v^2 OD_0^3} \tag{71}$$

$$\left.\frac{\partial \Omega}{\partial \omega}\right|_{x_0} = a_1 q_0 + 2a_0\omega_0 \tag{72}$$

$$\left.\frac{\partial \Omega}{\partial p_\tau}\right|_{x_0} = 1 \tag{73}$$

and the complete Taylor expansion becomes:

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

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 $\frac{\partial \Omega}{\partial \tilde{OD}}$ is:

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

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

$$\dot{q}_n \approx -\mathcal{P}\Phi\left(a_1\omega_0 + \Big(|q_0| + \text{sign}(q_0)q_0\Big)\left(K_\lambda + a_2 + \frac{1}{(K_v O D_0)^2}\right)\tilde{q}_n\right) - \mathcal{P}\Phi\left(\Big(a_1 q_0 + 2a_0\omega_0\Big)\tilde{\omega}\right) \quad (77)$$

5.3 Linearised model

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

5.3.1 State space model - pump dynamics

Before linearizing the system model, presented in eq. (57), we need to identify the equilibrium points. This is done by evaluating the before mentioned model, at fixed ω and OD, with dynamic states set to zero. We choose pump speeds as an average of those used in section 8.1, which is 66% for each pump. We set OD to 0.5 for each valve. This yields the following equilibrium point:

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

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

$$d_t^* = 0 \tag{80}$$

The linearised model of the system can be expressed on the standard state space form given in eq. (81) by applying the equilibrium to eq. (77)

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

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

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

The output matrix C can be constructed by the fact that the two outputs are; the flow pump at the non reference pump and the flow to/from the tank.

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 & -1 \end{bmatrix} \tag{84}$$

The inputs to our system will be the pump speeds. The outputs will be the measured pressure at the tank node and the flows of the pumps. This yields

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

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:

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

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

$$\begin{gathered} p_\tau(k+1) = p_\tau(k) + \tau \left(d_1(k) + d_{13}(k) + d_5(k) + d_9(k) \right) t_s = T \left(d_1(k) + d_{13}(k) + d_5(k) + d_9(k) \right) \\ T = \tau \cdot t_s \end{gathered} \tag{87}$$

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

$$\begin{gathered} p_\tau(k+1) = A p_\tau(k) + B_p d_p(k) + B_c d_c(k) = A p_\tau(k) + \left(B_p \begin{bmatrix} d_1(k) \\ d_{13}(k) \end{bmatrix} + B_c \begin{bmatrix} d_5(k) \\ d_9(k) \end{bmatrix} \right) \\ A = 1, \; B_p = \begin{bmatrix} T & T \end{bmatrix}, \; B_c = \begin{bmatrix} T & T \end{bmatrix} \end{gathered} \tag{88}$$

In practice the flows may not all be measurable, in which case they must be estimated with an observer. As eq. (88) is clearly linear, either a simple Luenberger observer or a Kalman filter will do. Additionally, we note the presence of $B_c d_c$, an exogenous input over which we have no control. This will be addressed in section 6.

5.4 Choice of state variables

5.5 Choice of inputs and output variables

5.6 Cascaded control

5.7 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 possible bandwidth of a system to be

$$\omega_c < \frac{1}{\theta} \tag{89}$$

This naturally sets a limitation as to how fast the inner loop of the control structure can be. As the inner loop sets a limitation of how fast the outer loop can be, i.e. 5-10 timer slower than the inner loop. This is a nessesary limitation, since the inner loop needs to be sufficiently fast for it to be used to linearize the outer loop.

5.8 Design of pump controllers

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

5.8.1 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 systems are investigated with magnitude plots. For this analysis the tank is assumed to have a constant pressure. This is part of the seperation of the inner and outer loop, where the outer loop is assumed to be sufficienly slow, so it can 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.

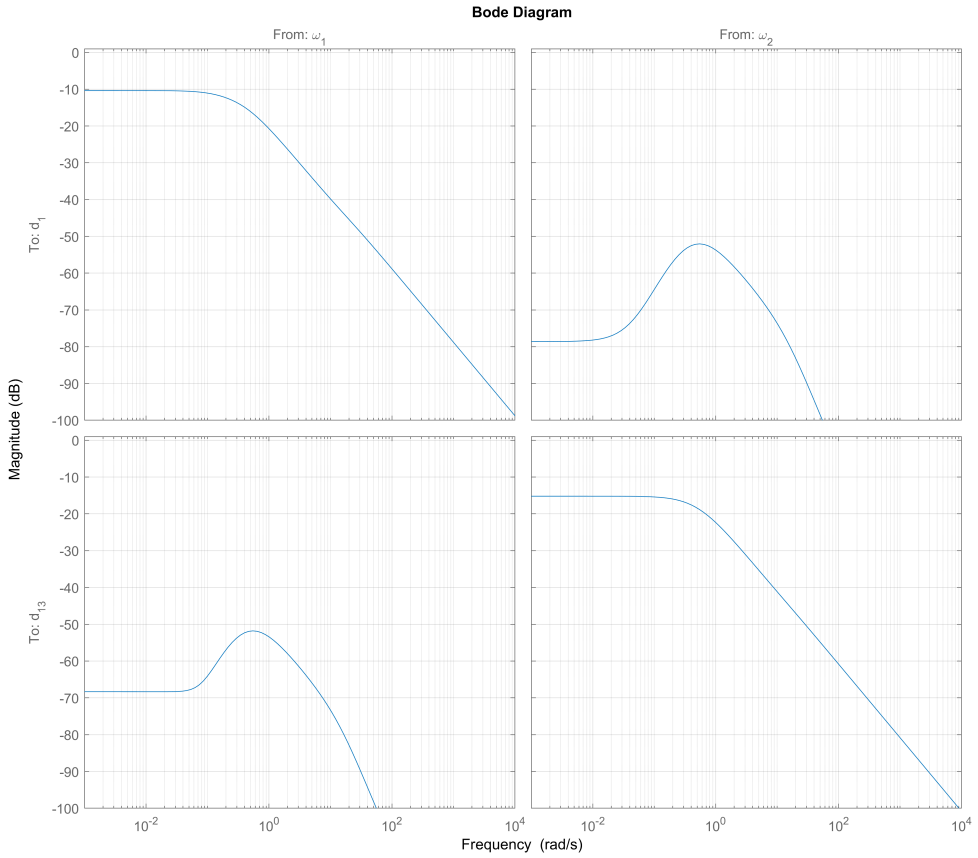

Fig. 5. Magnitude Plot from inputs to outputs

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 is the gain from ω_2 to d_1 for all frequencies less than -50 dB, 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 $1/50$. Thus the system will be assumed to have no interaction/coupling, which allows for design of controllers independently of the interactions, and solely from the diagonal transfer functions in the transfer matrix.

5.8.2 Similarities within the diagonal systems and simplification

The diagonal transfer functions are considered to be identical, based on the poles and zeros of the two diagonal transfer functions seen in eq. (90):

$$\begin{aligned} z_{11} &= \begin{bmatrix} -18.9173 & -13.9942 & -0.5571 & -0.3357 & -0.1944 \end{bmatrix} \\ z_{22} &= \begin{bmatrix} -20.7216 & -14.1600 & -0.4079 & -0.3639 & -0.1597 \end{bmatrix} \\ p_{11} = p_{22} &= \begin{bmatrix} 20.7703 & -14.8635 & -0.5624 & -0.3930 & -0.3337 & -0.1597 \end{bmatrix} \end{aligned} \tag{90}$$

Considering the two systems identical enables the design of only one controller for both the subsystems. Realising that many of the poles more or less cancels out with the zeros can simplify the model very much for the root locus design, yielding poles and zeros eq. (91):

$$\begin{aligned} z &= \begin{bmatrix} -0.1944 \end{bmatrix} \\ p &= \begin{bmatrix} -0.3930 & -0.1597 \end{bmatrix} \end{aligned} \tag{91}$$

5.8.3 Modelling of time delay

The timedelay of the pumps can be modelled with a pade approximation, using Matlab. The order of the approximation is not unimportant. The higher the order, the closer to a true time delay the result will be. It comes however at the cost of clarity, as a nth-order approximation places n LHP poles and n RHP zeroes, to simulate the delay. This is shown in fig. 6.

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 shows the closed loop pole locations when using a 1st, 2nd and 10th order pade approximation. The plots show a noticable difference from 1st to 2nd order, but no significance above that.

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

5.8.4 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 achived by ensuring that the closed loop poles of the system are placed in the left half plane on the s-plane. Simultaniously the inner loop of the system should be at least 5 times faster than the slow outer system [we have to come up with a reasonable estimate of this]. The desired BW of the inner loop is chosen to be 0.1 rad/s. This in turn is another restriction on the closed loop pole locations.

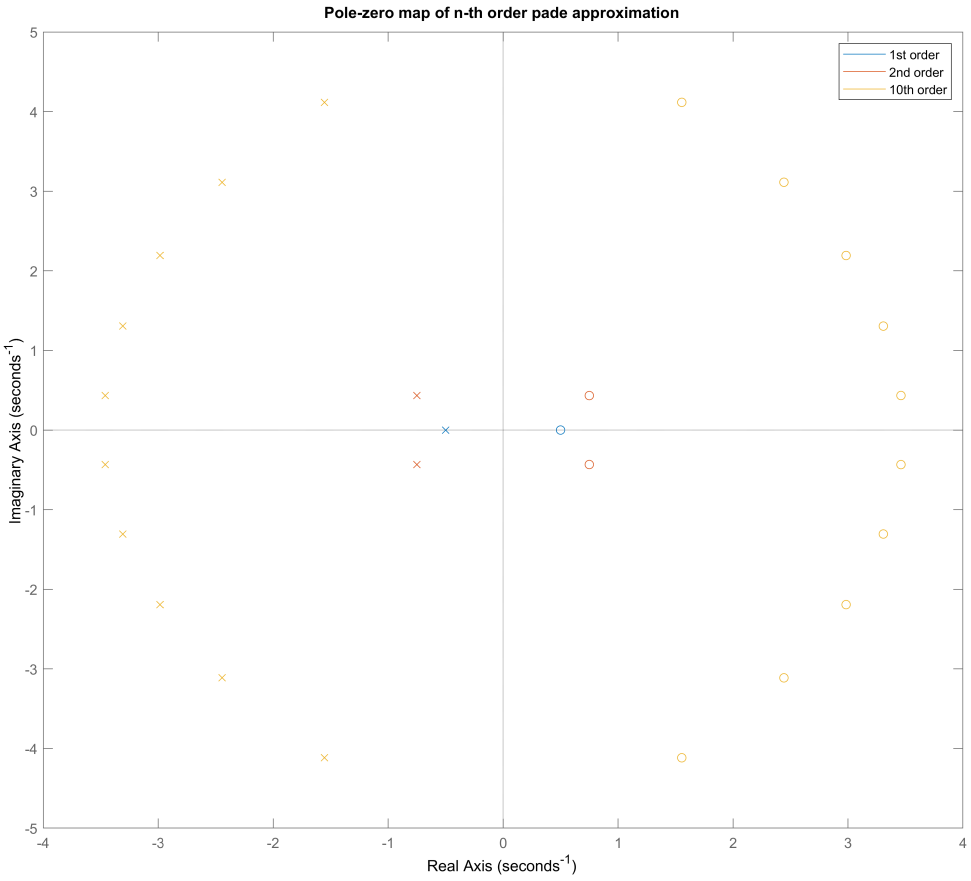

Fig. 6. Pole-zero map of various order pade approximations of a 4 second delay

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 oscilations 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 pade 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 must 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 rad/s.
- The closed loop system must have a time constant of approximately $\frac{1}{bw} = \frac{1}{0.1} = 10s$.

5.8.5 Choice of controller type

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

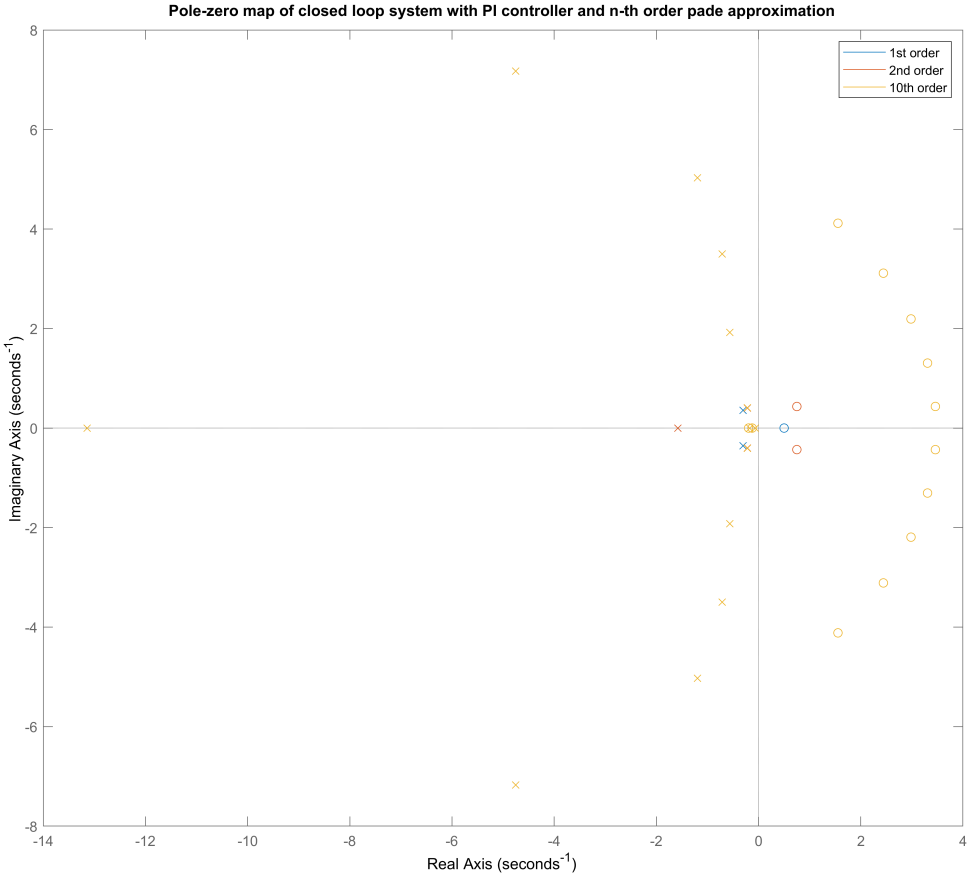

Fig. 7. Pole-zero map of closed loop pole locations with various order pade approximations of a 4 second delay

5.8.6 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 pade 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 loci where the possible closed loop poles are all undesireable.

For step 3, the gain was increased until the bandwidth of 0.1 rad/s was achieved with no oscilations.

As step 4 implies, the PI controller was designed after some iterations, resulting in the following controller:

$$G_{PI} = 1.8\frac{s + 0.125}{s} \tag{92}$$

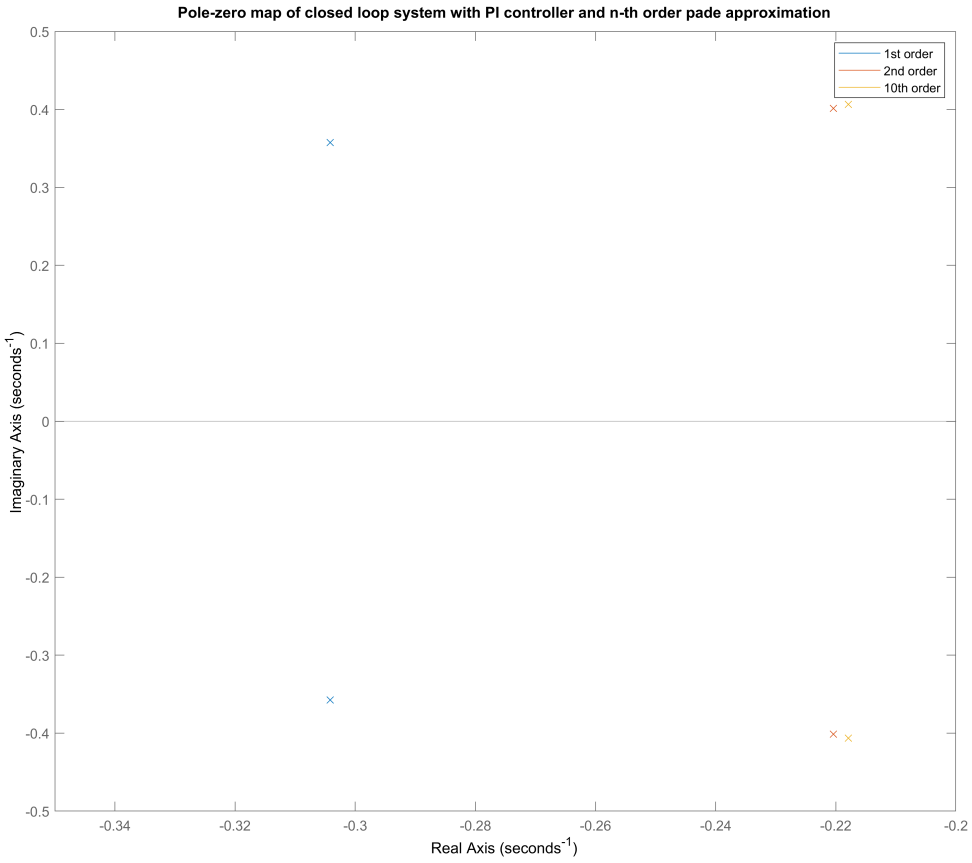

Fig. 8. Pole-zero map of closed loop pole locations with various order pade approximations of a 4 second delay. Zoomed in on the dominant complex conjugate pole pair.

With this controller the root locus can be seen on fig. 9. The plot show 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 pade 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 that 10th order.

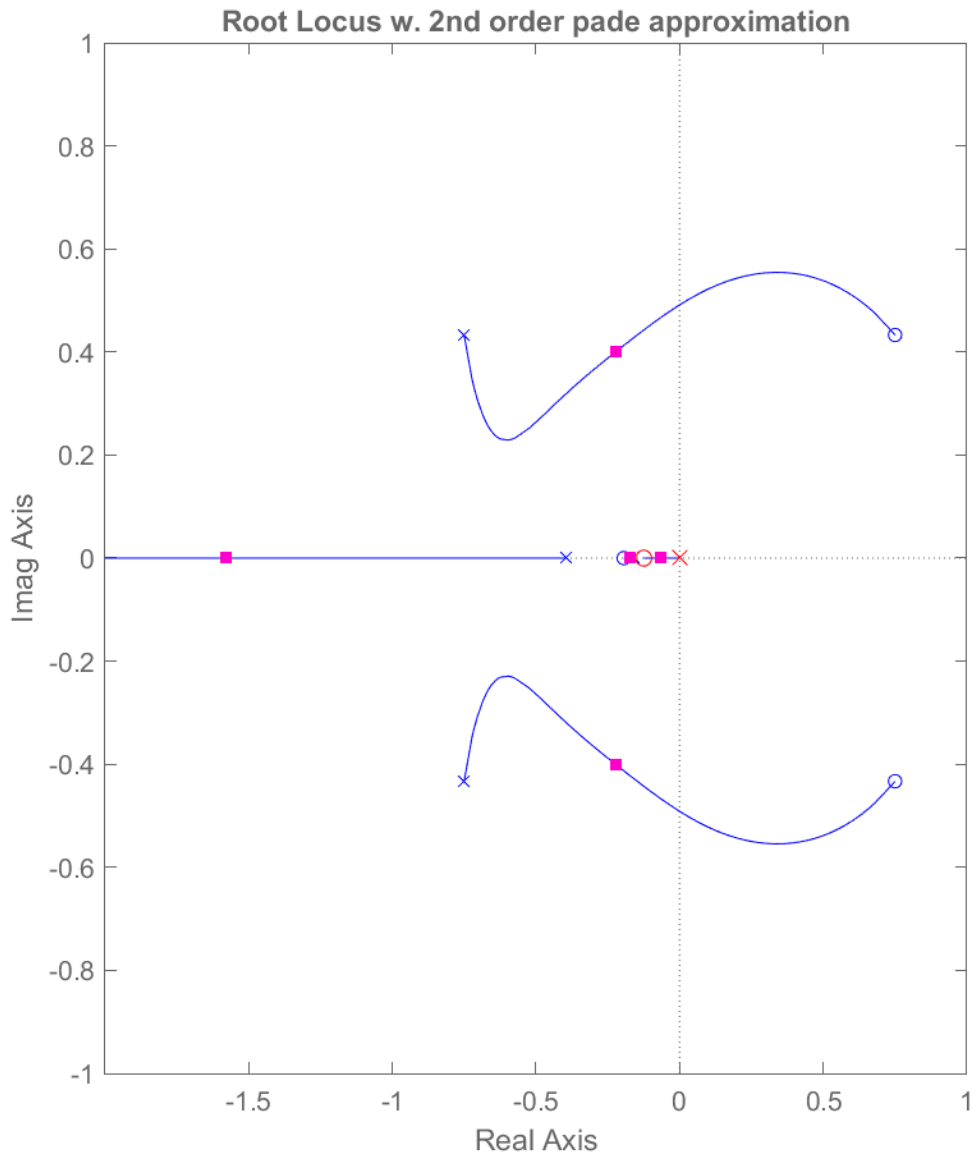

Fig. 9. Root locus of delayed pump transfer function with PI controller

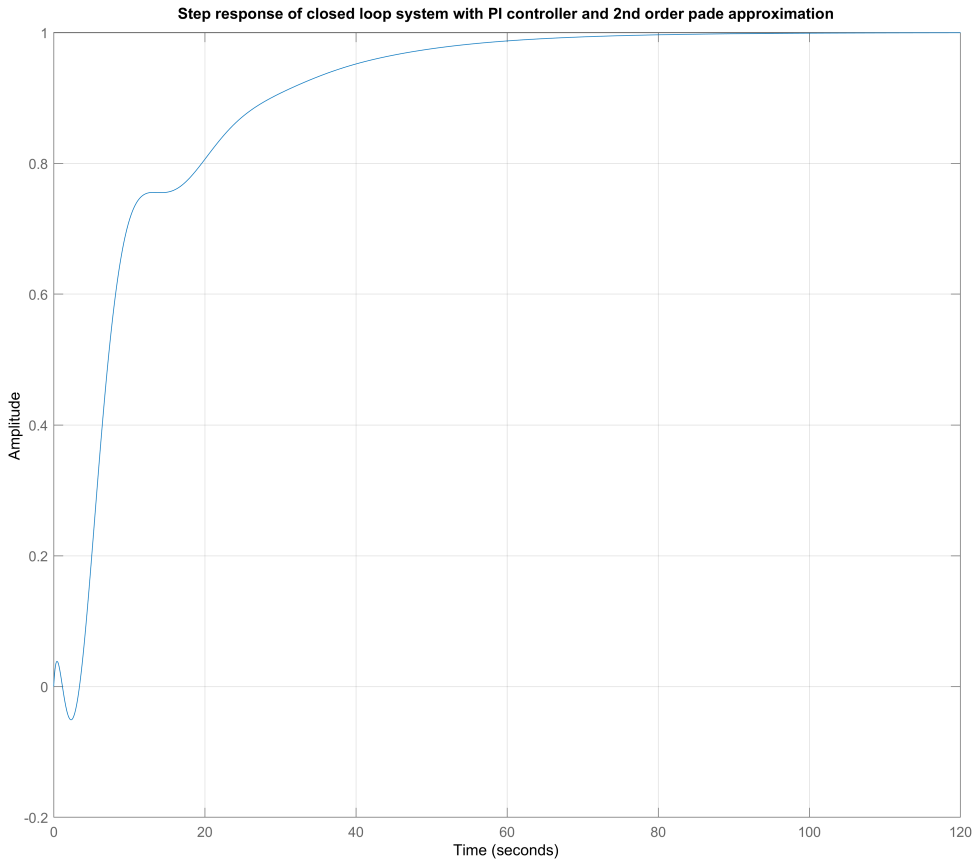

Fig. 10. Step response of the closed loop system with a 2nd order pade approximation

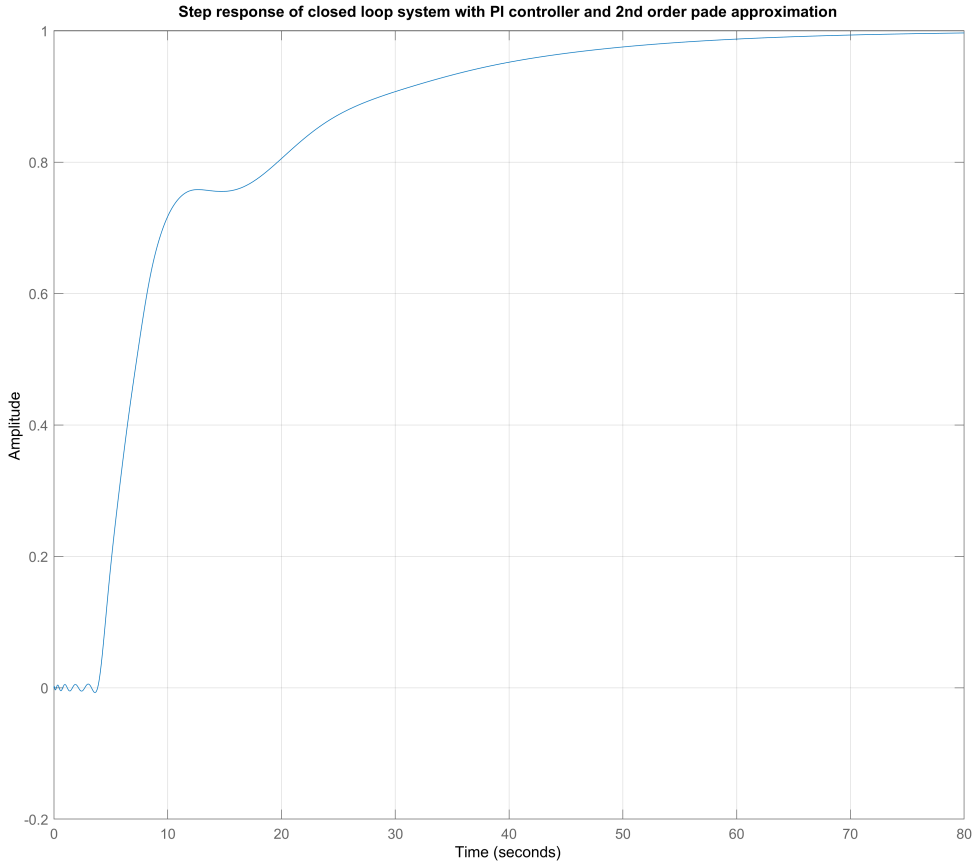

Fig. 11. Step response of the closed loop system with a 10th order pade approximation

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:

$$\dot{x} = f(x, u, t) \tag{93}$$

$$x(t_0) = x_0,\, x \in \mathbb{R}^n,\; u \in U \in \mathbb{R}^m \tag{94}$$

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:

$$J = \mathcal{M}(x(T)) + \int_0^T \mathcal{L}(x(t), u(t))dt \tag{95}$$

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. (94), we can formulate a function known as the *Hamiltonian*:

$$\mathcal{H}(x(t), u(t), \lambda(t), t) = \lambda^T(t) f(x(t), u(t)) + \lambda_0 \mathcal{L}(x(t), u(t)) \tag{96}$$

where λ^T are *Lagrange multipliers* that are commonly referred to as the *costates* of eq. (93) 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:

$$\forall t \in [0, T] \wedge \forall u \in U :\; \mathcal{H}(x^*(t), u^*(t), \lambda^*(t), t) \leq \mathcal{H}(x^*(t), u(t), \lambda^*(t), t) \tag{97}$$

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

$$-\dot{\lambda}^T(t) = \frac{\partial}{\partial x}\mathcal{H}(x^*(t), u^*(t), \lambda(t), t) = \frac{\partial}{\partial x}\Big(\lambda^T f(x^*(t), u^*(t)) + \lambda_0 \mathcal{L}(x^*(t), u^*(t))\Big) \tag{98}$$

$$\dot{x} = \frac{\partial}{\partial \lambda}\mathcal{H}(x^*(t), u^*(t), \lambda(t), t) = \frac{\partial}{\partial \lambda}\Big(\lambda^T f(x^*(t), u^*(t)) + \lambda_0 \mathcal{L}(x^*(t), u^*(t))\Big) \tag{99}$$

and the costates must fulfil the terminal condition:

$$\lambda^T(T) = \mathcal{M}_x(x(T)) \tag{100}$$

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:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \tag{101}$$

$$x(t_0) = x_0 \tag{102}$$

and let a cost functional be given by:

$$J = \int_{t_0}^{t_1} \big(x^T(t)Q(t)x(t) + u^T(t)R(t)u(t)\big)dt + x^T(t_1)\mathcal{M}x(t_1) \tag{103}$$

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:

$$\mathcal{L}(x(t), u(t)) = x^T(t)Q(t)x(t) + u^T(t)R(t)u(t) \tag{104}$$

and per [7] we may freely choose $\lambda_0 = -1$[4] such that the Hamiltonian becomes:

$$\mathcal{H}(x(t), u(t), \lambda(t), t)) = \lambda^T(t)\big(A(t)x(t) + B(t)u(t)\big) - x^T(t)Q(t)x(t) - u^T(t)R(t)u(t) \tag{105}$$

Intuitively, the control gradient $\frac{\partial \mathcal{H}}{\partial u}$ of eq. (105) must vanish along the optimal control trajectory $u^*(t)$ if we are to satisfy the first condition of Pontryagin's Minimum Principle in eq. (97). Taking the partial differential, we obtain:

$$\frac{\partial \mathcal{H}}{\partial u} = B^T(t)\lambda^*(t) - 2R(t)u^*(t) \tag{106}$$

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

$$u^*(t) = \frac{1}{2}R^{-1}(t)B^T(t)\lambda^*(t) \tag{107}$$

It can then be shown - doing so is expressly outside the scope of this report, but relies on the remaining conditions eqs. (98) to (100) - that $\lambda^*(t)$ must satisfy a relation of the form:

$$\lambda^*(t) = -2P(t)x^*(t) \tag{108}$$

We see that this is quite clearly true at the boundary $t = T$, where per eq. (100) we must satisfy that:

$$\lambda^T(T) = \mathcal{M}_x(x(T)) = -2\mathcal{M}x^*(T) \tag{109}$$

The relation in eq. (108) then gives a state feedback control law in the form of:

$$u^*(t) = -R^{-1}(t)B^T(t)P(t)x^*(t) \tag{110}$$

4. 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.

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

$$\dot{P}(t) = -P(t)A(t) - A^T(t)P(t) - Q(t) + P(t)B(t)R^{-1}(t)B^T(t)P(t) \tag{111}$$

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:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{112}$$

$$x(t_0) = x_0 \tag{113}$$

and let the cost functional be:

$$J = \int_{t_0}^{\infty} \big(x^T(t)Qx(t) + u^T(t)Ru(t)\big)dt \tag{114}$$

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

$$\mathcal{H}(x(t), u(t), \lambda(t), t)) = \lambda^T(t)\big(Ax(t) + Bu(t)\big) - x^T(t)Qx(t) - u^T(t)Ru(t) \tag{115}$$

As before, the control gradient:

$$\frac{\partial \mathcal{H}}{\partial u} = B^T\lambda^*(t) - 2Ru^*(t) \tag{116}$$

must vanish along the optimal trajectory, necessitating that:

$$u^*(t) = \frac{1}{2}R^{-1}B^T\lambda^*(t) \tag{117}$$

and it can analogously to section 6.2 be shown that:

$$\lambda^*(t) = -2Px^*(t) \tag{118}$$

such that:

$$u^*(t) = -R^{-1}B^TPx^*(t) \tag{119}$$

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

$$PA + A^TP + Q - PBR^{-1}B^TP = 0 \tag{120}$$

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. (112) be given by:

$$y = Cx \tag{121}$$

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

$$J(u) = \int_{t_0}^{\infty} \left(x^T(t) C^T C x(t) + u^T(t) R(t) u(t)\right) dt \tag{122}$$

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. (114) 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:

$$J = \int_{t_0}^{\infty} \left(\hat{x}^T Q \hat{x} + \hat{u}^T R \hat{u}\right) dt \tag{123}$$

We can by a similar ploy convert eq. (123) into an *output* tracking problem, which is perhaps the most common practical control problem, as C is simply a linear transformation (i.e a coordinate translation) on the states:

$$J = \int_{t_0}^{\infty} \left((C\hat{x})^T Q_y (C\hat{x}) + \hat{u}^T R \hat{u}\right) dt = \int_{t_0}^{\infty} \left(\hat{y}^T Q_y \hat{y} + \hat{u}^T R \hat{u}\right) dt \tag{124}$$

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:

$$u = -\bar{K}\bar{x} \tag{125}$$

$$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}u + B_r r \tag{126}$$

$$y = \bar{C}\bar{x} \tag{127}$$

$$\bar{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}, \ \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \ B_r = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ \bar{C} = \begin{bmatrix} C & 0 \end{bmatrix}, \ \bar{K} = -\begin{bmatrix} K & -K_i \end{bmatrix}, \tag{128}$$

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:

$$\Delta x_k = x_k - x_{k-1},\ \Delta y_k = y_k - r_k,\ \Delta u_k = u_k - u_{k-1} \tag{129}$$

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

$$\tilde{x}_k = \begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix},\ \tilde{u}_k = \Delta u_k,\ \tilde{A} = \begin{bmatrix} A & 0 \\ CA & I \end{bmatrix},\ \tilde{B} = \begin{bmatrix} B \\ CB \end{bmatrix},\ \tilde{C} = \begin{bmatrix} 0 & I \end{bmatrix} \tag{130}$$

and the velocity-form dynamics:

$$\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + \tilde{B}\tilde{u}_k \tag{131}$$

$$\Delta y_k = \tilde{C}\tilde{x}_k \tag{132}$$

$$\tag{133}$$

We now note a number of interesting properties about this velocity-form algorithm. Consider first the origin regulation problem, i.e. $\tilde{x} \to 0$. Clearly, driving $\tilde{x}$ to 0 must drive the system to settle at exactly r, per the definitions in eq. (129), as:

$$\begin{aligned} \tilde{x} = 0 &\Rightarrow \Delta x = 0 \\ \tilde{x} = 0 &\Rightarrow \Delta y = 0 \Rightarrow y = r \end{aligned} \tag{134}$$

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

$$J = \sum_{k_0}^{\infty} \left(\tilde{x}^T Q \tilde{x} + \tilde{u}^T R \tilde{u} \right) = \sum_{k_0}^{\infty} \left(\begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix}^T Q \begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix} + \Delta u_k^T R \Delta u_k \right) \tag{135}$$

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:

$$\begin{aligned} f(x,u) &\approx \frac{\partial f}{\partial x}\tilde{x} + \frac{\partial f}{\partial u}\tilde{u}, \\ \tilde{x} &= x - x_e,\ \tilde{u} = u - u_{op} \end{aligned} \tag{136}$$

which lends itself to the interpretation that:

$$\Delta\tilde{x} = x - x_e,\ \Delta\tilde{u} = u - u_{op} \tag{137}$$

This is a quite attractive quality when contextualised by the Hartman-Grobman theorem, as we expect linearised model dynamics to closely[5] 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. (135) is a control *increment* law, i.e. that:

$$\Delta u^*(k) = -\Delta K(k)\Delta x(k) \tag{138}$$

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

$$\begin{aligned} u(k) &= -K(k)x(k) \\ K(k) &= \sum_{i=1}^{k} \Delta K(i) \end{aligned} \tag{139}$$

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:

$$u^*(k) = -Kx(k) - B^\dagger B\delta(k) \tag{140}$$

where $B^\dagger$ is the Moore-Penrose pseudoinverse of B and $\delta(k)$ is the exogenous input at time k.

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 $t_s = 10\text{s}$, and the tank constant is $\tau = -0.000096\frac{\text{bar}}{\text{m}^3}$, such that $T = 0.00096$. This results in the system:

$$\begin{aligned} p_\tau(k+1) &= Ap_\tau(k) + B_p d_p(k) + B_c d_c(k) \\ A = 1,\ B_p = B_c &= [0.00096\ 0.00096],\ C = 1, \end{aligned} \tag{141}$$

Using the velocity-form transform from eq. (130), the system becomes:

$$\begin{aligned} \begin{bmatrix} \Delta p_\tau(k+1) \\ p_\tau(k+1) - p_r \end{bmatrix} &= \tilde{A}\Delta p_\tau(k) + \tilde{B}_p \Delta d_p(k) + \tilde{B}_c \Delta d_c(k) \\ \tilde{A} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix},\ \tilde{B}_p = \tilde{B}_c &= \begin{bmatrix} 0.00096 & 0.00096 \\ 0.00096 & 0.00096 \end{bmatrix},\ \tilde{C} = [0\ 1] \end{aligned} \tag{142}$$

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

7 CONCLUSION

The conclusion goes here.

8 APPENDIX

This section is concerned with the appendices of the WorkSheets

8.1 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:

$$\Delta p_{pump} = a_0 \cdot \omega^2 + a_1 \cdot q \cdot \omega + a_2 \cdot q^2 \tag{143}$$

The pump coefficients are estimated from the pump performance curves from the data-sheets. These describe the exact relationship described in eq. (143).

The preassure drops were read at the following flows:

$$q = \begin{bmatrix} 0 & 0.5 & 1 & 1.5 & 2 & 2.5 & 3 & 3.5 \end{bmatrix} \tag{144}$$

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:

$$p = \begin{bmatrix} p_{80} \\ p_{69} \\ p_{48} \end{bmatrix} = \begin{bmatrix} 5.57 & 5.85 & 5.9 & 5.7 & 5.1 & 4.25 & 3.3 & 2.4 \\ 4.1 & 4.2 & 4.2 & 4 & 3.55 & 3 & 2.3 & 1.55 \\ 1.75 & 1.85 & 1.7 & 1.3 & 0.95 & & & \end{bmatrix} \tag{145}$$

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

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

$$Ax = b \leftrightarrow x = (A^T A)^{-1} A^T b \tag{146}$$

where A is a matrix defined as follows:

$$A = \begin{bmatrix} \omega^2 & q \cdot \omega & q^2 \end{bmatrix} \tag{147}$$

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

and $b = \begin{bmatrix} p_{80} & p_{69} & p_{48} \end{bmatrix}^T$ from eq. (145).

The solution yields the following coefficients:

$$\begin{aligned} a_0 &= 0.0001 \\ a_1 &= 0.0004 \\ a_2 &= -0.0323 \end{aligned} \tag{148}$$

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

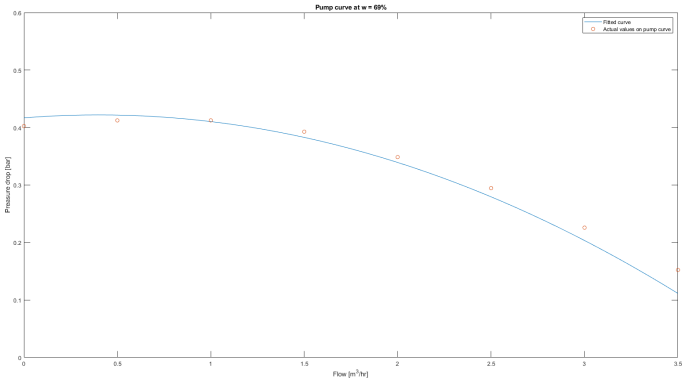

Fig. 12. Comparing model of pump curve with actual values from data sheet

9 Model Legend

Model:

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

where p_τ evolves according to:

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

and the matrices $\Phi, \Psi, \mathcal{I}$ are defined as:

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

Φ and Ψ contains:

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

$$F = \begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \\ d_9 \\ d_{10} \end{array} \begin{array}{c} \begin{array}{cccc} d_{f_1} & d_{f_2} & d_{f_3} & d_{f_4} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}, \qquad G = \begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \\ d_9 \\ d_{10} \end{array} \begin{array}{c} \begin{array}{c} d_{\tau_1} \end{array} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{array}$$

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

$\mathcal{J}$ is the mass inertia of the water in all pipes [$\mathrm{kg/m^4}$]. Dim = m x 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	[$\frac{rad}{s}$]
$\alpha_0 - \alpha_2$	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]

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] S. S. Rathore, "Optimal Control in Water Distribution Network," 2019.

[2] P. K. Swamee and A. K. Sharma, *Design of Water Supply Pipe Networks*. Wiley-Interscience, 2008.

[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. [Online]. Available: https://www.wiley.com/en-us/Multivariable+Feedback+Control%3A+Analysis+and+Design%2C+2nd+Edition-p-9780470011676

[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.