Data reconciliation in gas networks

Jasper Verbree

May 2018

1 Introduction

During my internship at DNV GL I worked on multiple projects, all of which were concerned with finding ways of detecting the failure of sensors located in gas networks. The main project I worked on is called DVR and concerns a known method for the detection of systematic errors in measurements making use of the structure of the network. The aim of the project is to test this ability of the method to detect the presence of systematic errors in measurements in large-scale networks. The large-scale of the network means that we are considering networks that are both physically large, containing hunderds of kilometers of pipeline, as well as containing a lot of connections and measurement points, making for a complex network structure. As it is the main project I worked on during my internship, DVR will be the main subject of this report.
I also looked at another project before i started on DVR, which is called MECADA. This project aims at finding a way to assess the performance of a specific type of meter at the hand of measurement data and adjustment variables produced by the meter itself. I will briefly describe this project as well.
Lastly, I will give a short personal reflection on my internship here at DNV GL.

2 DVR

The main project I worked on concerned the testing of a method called Data Validation and Reconciliation (DVR). The method has previously been used by DNV GL to detect the presence of systematic errors in measurements in a small-scale gas network [1]. The question to be addressed was whether or not the method could still detect the presence of errors in measurements in larger networks. First we will give a short introduction about the motivation behind this project. Then we will give an overview of the structure and the relevant characteristics of the networks that we will be considering. Following that we will explain the concept of mass-balance equation, which we use to construct a simple model of the flow through a network in steady-state. Then we will give an overview of the problems and factors that need to be taken into consideration as we are working with physically large-scale networks. After that there is a section introducing the method DVR, and how it is used to make estimations about the flow through the network based on the measurements and the mass-balance equations. Finally we then discuss the testing procedure I used for the method, the results thereof, and the conclusions and discussion following from these results.

2.1 Project motivation

The primary motivation behind this project comes from questions asked by the Gasunie. The Gasunie monitors and controls a large gas network in the Netherlands. Due to errors in the measurements concerning the flow of gas through the network, an imbalance can occur between the measurement and actual amount of gas being delivered. The initial measurement of the amount of gas that is delivered to a client, may disagree with later estimations based on knowledge of additional measurements and other estimations. As a result the client has payed either too much or too little, and this is of course undesirable. Additionally, measurements

[1] I can't give a reference here, as I'm not allowed to distribute the documents involved

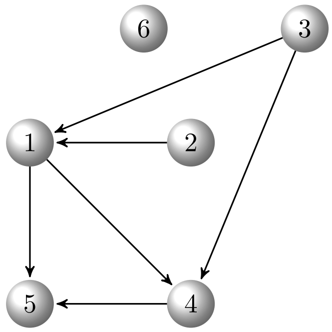

Figure 1: An example network.

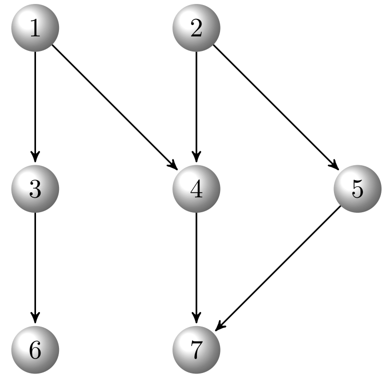

Figure 2: A layered network.

made are also used to determine the integrity of the network, to see whether or not maintenance is required and to estimate the amount of gas that needs to be pumped into the network to meet the demands of the clients. Systematic errors in the measurements can therefore lead to miss estimations of the amount of gas required in order to meet the demands on the network. To prevent these difficulties, and of course to prevent wrong measurements from going unnoticed, we want to detect the presence of systematic errors in the measurements as soon as possible.

2.2 Network structure

The type of systems we will consider concern networks of pipelines through which a gas flow is realised. A natural means of describing the network topology of such a system is with a graph as depicted in figure 1. The nodes or vertices of this graph represent metering stations in the network, where measurements concerning the gas flow are made. The arrows, or edges, of the graph represent possible gas flows between metering stations. The direction of the arrow indicates the direction in which the gas flows.

First, we will introduce some terminology here that we will need later on in this report. One important term concerning graphs that we will use often is 'neighbours', and specifically 'out-neighbours'. If we look at the example in figure 1 we see that vertex 1 is connected to the vertices 2, 3, 4 and 5. These are referred to as the neighbours of vertex 1. The arrows show that two of these connections, the connections to the vertices 2 and 3, are incoming connections, therefore we refer to vertices 2 and 3 as the in-neighbours of vertex 1. The other two connections, to vertices 4 and 5, are outgoing connections and thus we refer to the vertices 4 and 5 as the out-neighbours of vertex 1.

Another term often used when considering dynamical models of certain processes is the term 'steady-state'. This term describes the situation where the state of a (physical) system remains constant throughout time. When considering gas networks this would mean the flow-rate, the pressure and the other variables which influence the flow remain constant over a certain time period.

For the sake of clarity: we will often refer to the processes, variables and relations that influence the flow of gas through the network as the 'dynamics'. Lastly we will refer to the gas flowing into and out of the network as the 'input' of the system we are considering.

Now, the types of networks that we will consider are more structured than the example in figure 1. They consist of three layers of vertices, and a gas flow going down the layers, as can be seen in figure 2. The top layer consists of all vertices where gas enters the network and the bottom layer consists of all vertices where the gas exits the network. In the middle layer gas does not exit or enter the network, but measurements are taken. In addition to the clear structure of the network we also assume that the gas flow along the edges is directed, and the gas can only travel down the layers, never upwards or between two vertices in the same layer. This prevents the network from containing any loops, where gas can return to a vertex that it has already passed. For the method that we are considering the non-existence of loops is not necessary, but it is

a useful assumption when one tries to construct a model of the flow through a gas network. In our case we are looking at a type of network that is encountered in practice, which is why this assumption is made.
With these assumptions we have a set of networks that we can take into consideration. Since the goal of this project is to assess the performance of DVR in detecting errors in measurements in any large-scale networks of the described type, we will be looking at randomly generated networks which have the desired form. These networks are generated by setting the amount of vertices in each of the layers, and subsequently randomly generating links between the vertices of the first two layers and between the vertices of the last two layers.
What remains is to describe the relations between the measured variables in such a network. From a systems and control point of view we would like to describe these relations using a dynamic system of equations. However, modelling gas flows through networks of pipelines is, as we discovered, beyond the scope of this internship. Models for gas flow throughout a network are typically based on knowledge about the pressure and the flow-rate at certain points in the system, and are concerned with partial differential equations involving these variables. An example of this can be seen in [2]. In addition to these measurements, metering station also measure the temperature, gas density, and gas composition. These variables, although important for the dynamics of the gas flow, are not part of the dynamical model. Instead they are used to calculate parameters which influence the dynamics of the flow, and these parameters are often assumed to be constant over the considered period of time.

2.3 Flow based model

As previously mentioned, constructing a dynamic model describing the relations between the different variables in a gas flow network is outside the scope of this internship. Instead we need to look at different ways of describing these relations. One way to do that is to consider only the flow. The flow is the amount of gas that passes a metering station over a given period of time. This quantity can be calculated on the basis of measurements of flow-rate, density and temperature.
The relations between the flows at different measurement points can, under certain assumptions, be described by the intuitively simple rule: 'What comes in, must go out'. No gas can be created or destroyed within the network, and thus the amount of gas that flows through the first layer of vertices should be equal to the amount of gas that flows through the second layer and subsequently the third layer of the network. This gives us the so-called mass-balance equations.
The assumptions we need to make for the mass-balance equations to hold is that the system is in steady-state. As a consequence of this assumption the flow throughout the network is constant over time. We need to make this assumption since a change in flow at one vertex does not immediately influence the flow of the neighbours of that vertex. Consequently, if the flow changes at one point of measurement, for a certain time-period the mass-balance equations no longer hold.
For systems in steady-state, however, we can now easily describe the flow throughout the entire network. We simply divide the gas entering the system over the vertices in the first layer, and then divide the flow through a vertex in the network over all its out-neighbours, and so on. For the example network in figure 2 we could divide the flow as follows:

$$
\begin{aligned}
&f = 15, \\
&y_1 = 10, \;\; y_2 = 5 \\
&y_3 = 4, \;\; y_4 = 8, \;\; y_5 = 3 \\
&y_6 = 4, \;\; y_7 = 11
\end{aligned}
$$

where f is the amount of gas entering the system in the top layer.
In practice the way the flow is divided between the out-neighbours of a vertex is a result of the dynamics of the system and constant factors such as pipe diameter and the friction coefficient of the pipe. However, since we are looking at generic networks of the previously described three-layer form, we may randomly divide a flow through the network on the basis of the mass balance equations, without the need to define

the dynamics or other parameters which play a role. We can assume that the generated flow represents the steady-state of a generic network for which we do not know the dynamics and parameters, but only this steady-state flow.
The fact that we can only model the system in a steady-state is quite limiting, as in practice it may be the case that a system does not settle into a steady-state for a sufficiently long time-period. Additionally, not all situations we would like to take into account can be assumed to be a steady-state. This will be explained in more detail in section 2.4.
For an example of the mass-balance equations we look at the network in figure 2. The mass-balance equations describing the relations between the flows in the first and the second layer of vertices are given as follows:

$$y_1 + y_2 - y_3 - y_4 - y_5 = \begin{pmatrix} 1 & 1 & -1 & -1 & -1 & 0 & 0 \end{pmatrix} y = 0,$$
$$y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix}$$

where y_i is the measured value of the flow through node i and y is a vector containing these measurements. The mass-balance equation for the full network in figure 2 is given by

$$Ay = 0, \quad A = \begin{pmatrix} 1 & 1 & -1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & -1 \end{pmatrix}. \tag{1}$$

We see that the first row of $Ay = 0$ ensures that the amount of gas flowing through the first and the second layer are equal. The second row ensures that the amount of gas flowing through vertices 3 and 6 are equal, and the last row ensure that the flow through the vertices 4 and 5 is equal to the flow through vertex 7.

2.4 Considered factors

We want to test the performance of DVR in detecting errors in measurements in large-scale gas flow networks. Previously the method has already been applied to a smaller-scale gas network. As we consider networks that are both physically larger and contain more measurement points and connections, there are certain properties of the network that could be seen as negligible in small-scale systems, but can no longer be ignored.
The first of these properties is called the linepack, and it concerns the amount of gas that is stored within the network. For small-scale networks, the amount of gas that is stored within the pipelines of the network is negligible compared to the amount gas that flows through the network during a measuring period. As a result, small variations in the amount of stored gas have little to no effect on the dynamics of the flow and thus need not be taken into account. However, in networks containing hundreds of kilometres of pipeline, the amount of gas stored in the network, the linepack, may be considerable. Deviations in the stored amount of gas influence the pressure throughout the system and this in turn influences the dynamics of the flow. The aim of this project is mainly to evaluate the effects of the linepack on the performance of DVR.
Another property that comes into play when considering large-scale networks is that of delay. In small-scale networks, a change in flow-rate or pressure at one point is quickly noticed by the rest of the system as the resulting pressure wave travels through the relatively short pipelines. However, we are considering networks that potentially contain pipelines of hundreds of kilometres. The effect of for instance a change in pressure at one side of a hundred kilometre pipe takes roughly five minutes to reach the other side of the pipe [2].

[2] Such a pressure wave travels with approximately the speed of sound $C = 343ms^{-1}$ through the pipeline: $\frac{100.000m}{343ms^{-1}} = 291.55s \approx 5min$.

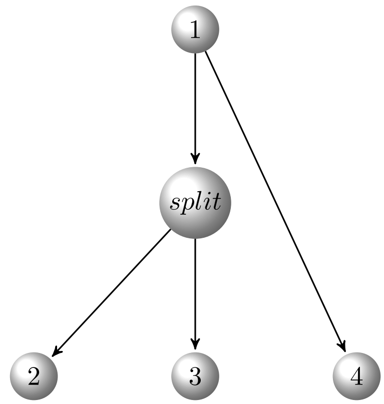

Figure 3: pipeline split.

Consequently, for the entire network to notice the change in the state at one measuring point may take a considerable amount of time. As a consequence it takes longer for the network to reach a steady-state, and for such a steady-state to be attained requires that the input of the system remains constant for long periods of time. Note that even if the input does remain constant for long periods, a steady-state may not be attained. The system could instead converge to a periodic solution. Since we are limited to a model of a steady-state we can not take the delay factor under consideration while evaluating the performance of DVR, but it is important to remember that steady-state may not be attained for long time-periods in large systems.

As we are dealing with large-scale networks, another thing to take into account is that the network itself may be subject to change. Pipelines may be shut down temporarily for maintenance, or connections may be added or removed. This of course influences the dynamics of the flow throughout the network. Once again, the static nature of the model that is available to us prevents us from testing the effects of these changing dynamics on DVR. We can however see whether it would be easy to adapt a method to a new network structure, or if incorporating the new structure into the method would be a difficult and time-consuming process. In networks where changes are common, ease of adaptation is a desirable property.

One more factor that is due to the large-scale of the network, is the potential complexity of the network. A large number of vertices and connections in the network lead to computationally heavy calculations in both the application of DVR and the generation of random network structures and flow. We have already made a simplification in order for computation time to remain reasonable. As we look at figure 2 we see that any flow between two metering stations is represented by a single edge. In reality such a connection is a pipeline and may pass intersections, junctions, splits and bends, which influences the dynamics of the flow, and can supply us with additional relations between flows. In figure 3 for instance, we can see that the flow from vertex 1 to vertices 2 and 3 are related to each-other, as in a steady-state the sum of the flows through the vertices 2 and 3 is equal to the total flow through the split in the pipeline. In our simplified model however, we do not take such relations into account.

The last factor we need to mention is that of measurement uncertainty. In practice measurements are never perfect. For instance, even if there is no systematic fault present, the true flow-rate at a measuring point probably differs from the measured flow-rate. As a consequence the same is true for the measured flow. The size of the difference between the true value and the measured value depends on the quality of the used sensors. The quality of the different sensors throughout the network may differ. As a result, some measurements are more reliable than others, and this is a factor we need to take into account as we try to detect faults in the measurements. The faults in the measurements which are not due to systematic errors can be simulated by taking the true flow through a metering station and perturbing them using a normal distribution centred around the true flow with a standard deviation depending on the precision of the sensors. We should mention that measurement uncertainty plays a role in small-scale systems as well.

2.5 DVR explained

If the system under consideration is in steady-state, then we know that the mass-balance equations should hold. The observed measurements however, will most likely not satisfy these equations, due to errors in measurement. DVR estimates the true value of the measured variables by finding values 'close' to the measurements, such that the mass-balance equations hold for this estimation. The term 'close' in this case means that the following cost function is minimized:

$$\min_{\hat{y}} (y - \hat{y})V^{-1}(y - \hat{y}) \tag{2}$$

subject to

$$A\hat{y} = 0. \tag{3}$$

In these equations, $y \in \mathbb{R}^n$ stands for the vector of measurements of the flow, $\hat{y} \in \mathbb{R}^n$ is the to be found vector of estimations of the flow, $A\hat{y} = 0 \in \mathbb{R}^m$ is the vector representation of the mass-balance equations following from the graph structure as in (1), and $V \in \mathbb{R}^{n \times n}$ is the co-variance matrix of the measurements. The co-variance matrix is used here to take into account the measurement uncertainties of different metering stations in the network. If no systematic error is present in the system, we assume that the errors in measurement are due to independent random noise processes working on the individual terms y_i of the vector y. As a consequence of this independence the co-variance between any two terms of y is equal to zero, which implies that V is a diagonal matrix with $v_{ij} = 0$ if $i \neq j$, $v_{ii} = var(y_i)$ $\forall i, j \in \{1, \cdots, n\}$. Here we denoted the element of V in the i-th row and j-th column with v_{ij}. It follows that V^{-1} is a diagonal matrix with the i-th diagonal element equal to $\frac{1}{var(y_i)}$. We see that the difference between the measurement y_i and the estimation $\hat{y}_i$ is weighted by $\frac{1}{var(y_i)}$ in the cost function of (2). It follows that the larger the variance of y_i, the more the estimation $\hat{y}_i$ is allowed to differ from y_i. The size of the variance of y_i is determined by the size of the flow through the related metering station and the uncertainty of the measurements made at that metering station. As a consequence the cost function in (2) allows for larger deviations between measurement and estimation at metering station with a higher measurement uncertainty or metering station with a larger flow through them. In this way the method of DVR incorporates the knowledge about the measurement uncertainties of the different metering stations.
Typically, minimising a cost function as in (2) would require the application of numerical algorithms. However, in this case, the solution to this problem can be found analytically. A short mathematical proof of this observation is given next. This proof requires some knowledge about linear-algebra, convexity an continuous optimization that is too extensive to explain in this report, and it will be explained rather briefly. For those who are not interested, I advise skipping the rest of this section. The only relevant part for the rest of the report is that the solution of the minimisation problem given by (2) subject to the constraints in (3) is given by (12).

2.5.1 Solution and proof

Since V is a co-variance matrix it is a positive semi-definite matrix. For this proof we need to make two not very constrictive assumptions. The first is that A is of full row-rank. This, however, can always be achieved, since adding dependent rows to the matrix A does not put any additional constrictions on the solutions. In addition we assume that V is positive definite, which is necessary for V^{-1} to exist. It follows that V and consequently, V^{-1} are positive definite matrices and thus that the cost function in (2) is convex. We rewrite the minimisation problem in a standard form for continuous optimisation problems.

$$\min_{\hat{y}} (y - \hat{y})V^{-1}(y - \hat{y}) \tag{4}$$

$$A\hat{y} \leq 0 \tag{5}$$

$$-A\hat{y} \leq 0. \tag{6}$$

The constraints of this optimization problem are now given by the inequalities (5) and (6). Since $A\hat{y}$ and $-A\hat{y}$ both describe a vector of affine functions, and as a consequence all constraints are affine, it follows that Slater's condition holds for this optimization problem[3](P. 284). Consequently we can look at the Lagrangian dual problem and know that the solutions to the original problem and the Lagrangian dual are the same [4](Thm 3.6):

$$L(\hat{y}, \lambda_+, \lambda_-) = (y - \hat{y})V^{-1}(y - \hat{y}) + \lambda_+^T A\hat{y} - \lambda_-^T A\hat{y} \tag{7}$$

$$\psi(\lambda_+, \lambda_-) = \inf_{\hat{y}} L(\hat{y}, \lambda_+, \lambda_-) \tag{8}$$

$$\sup_{\lambda_+ \in \mathbb{R}_+^m, \lambda_- \in \mathbb{R}_+^m} \psi(\lambda_+, \lambda_-). \tag{9}$$

Here $\lambda_+ \in \mathbb{R}_+^m$ and $\lambda_- \in \mathbb{R}_+^m$ are both vectors of dimension m with non-negative elements, and m denotes the number of rows of the matrix A. The Lagrangian dual problem consist of finding the supremum in (9) not subject to any constraints, and finding the value of $\hat{y}$ for which this supremum is attained. In order to do so, we first write out $\psi(\lambda_+, \lambda_-)$ for given values of λ_+ and λ_-. As a first step, observe that

$$\nabla L(\hat{y}, \lambda_+, \lambda_-) = -2V^{-1}(y - \hat{y}) + A^T\lambda_+ - A^T\lambda_-,$$
$$\nabla^2 L(\hat{y}, \lambda_+, \lambda_-) = 2V^{-1}.$$

Here ∇ denotes the Jacobian with respect to $\hat{y}$, and ∇^2 denotes the Hessian with respect to $\hat{y}$. Since V^{-1} is a positive definite matrix, it follows that the infimum of this function is attained for the value of $\hat{y}$ such that $\nabla L(\hat{y}, \lambda_+, \lambda_-) = 0$. This leads to the following implication:

$$-2V^{-1}(y - \hat{y}) + A^T\lambda_+ - A^T\lambda_- = 0 \tag{10}$$

$$\Rightarrow \hat{y} = y - \frac{1}{2}V(A^T\lambda_+ - A^T\lambda_-). \tag{11}$$

Writing out the Lagrangian function $L(\hat{y}, \lambda_+, \lambda_-)$ for this value of $\hat{y}$ gives us the following:

$$\begin{aligned}
L(\hat{y}, \lambda_+, \lambda_-) &= \frac{1}{4}(A^T\lambda_+ - A^T\lambda_-)^T V^T V^{-1} V(A^T\lambda_+ - A^T\lambda_-) \\
&+ \lambda_+^T A(y - \frac{1}{2}V(A^T\lambda_+ - A^T\lambda_-)) - \lambda_-^T A(y - \frac{1}{2}V(A^T\lambda_+ - A^T\lambda_-)) \\
&= \frac{1}{4}(\lambda_+^T A - \lambda_-^T A)V(A^T\lambda_+ - A^T\lambda_-) - \frac{1}{2}\lambda_+^T AV(A^T\lambda_+ - A^T\lambda_-) + \frac{1}{2}\lambda_-^T AV(A^T\lambda_+ - A^T\lambda_-) \\
&+ \lambda_+^T Ay - \lambda_-^T Ay \\
&= -\frac{1}{4}\lambda_+^T AVA^T\lambda_+ - \frac{1}{4}\lambda_-^T AVA^T\lambda_- + \frac{1}{2}\lambda_-^T AVA^T\lambda_+ + \lambda_+^T Ay - \lambda_-^T Ay \\
&= \psi(\lambda_+, \lambda_-)
\end{aligned}$$

In order to find the supremum of $\psi(\lambda_+, \lambda_-)$ we get the following:

$$\nabla\psi(\lambda_+, \lambda_-) = \begin{pmatrix} -\frac{1}{2}AVA^T\lambda_+ + \frac{1}{2}AVA^T\lambda_- + Ay \\ -\frac{1}{2}AVA^T\lambda_- + \frac{1}{2}AVA^T\lambda_+ - Ay \end{pmatrix}$$
$$\nabla^2\psi(\lambda_+, \lambda_-) = \begin{pmatrix} -\frac{1}{2}AVA^T & \frac{1}{2}AVA^T \\ \frac{1}{2}AVA^T & -\frac{1}{2}AVA^T \end{pmatrix}$$

Here ∇ denotes the Jacobian with respect to $(\lambda_+^T \ \lambda_-^T)^T$, and ∇^2 denotes the Hessian with respect to $(\lambda_+^T \ \lambda_-^T)^T$. Since V is a positive definite matrix, and since A is assumed to have full row rank it follows that $\frac{1}{2}AVA^T$ is positive definite as well. Consequently there exist a matrix Q such that $\frac{1}{2}AVA^T = QQ^T$. Thus we can write

$$\nabla^2\psi(\lambda_+,\lambda_-) = -\begin{pmatrix} Q \\ -Q \end{pmatrix}\begin{pmatrix} Q^T & -Q^T \end{pmatrix}.$$

It follows that $\nabla^2\psi(\lambda_+,\lambda_-)$ is a negative semi-definite matrix. From this it follows that $\psi(\lambda_+,\lambda_-)$ is a convex function over the entire domain. Since the Hessian is negative semi-definite we then know that the supremum of $\psi(\lambda_+,\lambda_-)$ is attained for the values of λ_+ and λ_- such that $\nabla\psi(\lambda_+,\lambda_-) = 0$. Then,

$$\begin{aligned}
&\begin{pmatrix} -\frac{1}{2}AVA^T\lambda_+ + \frac{1}{2}AVA^T\lambda_- + Ay \\ -\frac{1}{2}AVA^T\lambda_- + \frac{1}{2}AVA^T\lambda_+ - Ay \end{pmatrix} = 0 \\
&\Leftrightarrow \frac{1}{2}AVA^T\lambda_+ - \frac{1}{2}AVA^T\lambda_- + Ay = 0 \\
&\Leftrightarrow \frac{1}{2}AVA^T(\lambda_+ - \lambda_-) + Ay = 0
\end{aligned}$$

Since $\frac{1}{2}AVA^T$ is negative definite, we know that it is invertible. Thus we continue

$$\begin{aligned}
&\begin{pmatrix} -\frac{1}{2}AVA^T\lambda_+ + \frac{1}{2}AVA^T\lambda_- + Ay \\ -\frac{1}{2}AVA^T\lambda_- + \frac{1}{2}AVA^T\lambda_+ - Ay \end{pmatrix} = 0 \\
&\Leftrightarrow \lambda_+ - \lambda_- = 2(AVA^T)^{-1}Ay
\end{aligned}$$

We know that λ_+ and λ_- have non-negative elements. It follows that the possible choices for $\lambda_+ - \lambda_-$ span $\mathbb{R}^m$. Thus $\lambda_+ - \lambda_- = 2(AVA^T)^{-1}Ay$ can always be satisfied.
Plugging this relation into (11) we get the following:

$$\begin{aligned}
\hat{y} &= y - \frac{1}{2}V(A^T\lambda_+ - A^T\lambda_-) \\
&= y - \frac{1}{2}V(A^T(\lambda_+ - \lambda_-)) \\
&= y - VA^T(AVA^T)^{-1}Ay
\end{aligned}$$

We can summarise the conclusion of this proof as follows.
The solution of the optimization problem given by

$$\begin{aligned}
\min_{\hat{y}}(y-\hat{y})V^{-1}(y-\hat{y}) \\
A\hat{y} = 0
\end{aligned}$$

under the assumption that V is positive definite, and A has full row rank, is given by

$$\hat{y} = y - VA^T(AVA^T)^{-1}Ay. \tag{12}$$

2.6 Testing procedure

2.6.1 Data generation

To assess the performance of DVR in detecting whether or not systematic errors are present in the considered measurements, we first needed data to test the method on. In order to distinguish between the performance of the method on data with and without systematic errors present, we need data that is known to be reliable (although still subject to random noise disturbing the measurements), and data which is known to contain biases. In addition, we need to be sure that the data represents a steady-state, as otherwise the system does not need to satisfy the mass-balance equations that DVR enforces on the estimation. In order to ensure that the data satisfies these requirements we choose to generate the data ourselves, on the basis of randomly generated network structures and a flow divided throughout the network on the basis of the mass-balance equations. In order to be able to consider situations in which is gas is stored or drawn out of the system over a period of time, we add storage nodes in the system. This allows us to consider imbalances between the flow into and out of the system while the mass-blance equations are still satisfied.

The procedure for generating the data we used is as follows. First we set the amount of vertices in each layer of the graph. Most tests we performed on a network with 50 vertices in the top layer, 75 in the middle layer and 1000 in the bottom layer. The network structure is generated randomly using uniform distributions. For instance, for one vertex in the top layer we draw a number from a uniform distribution between 1 and 100 for a vertex in the middle layer. If the number exceeds a certain threshold, the two vertices are connected. Otherwise, they are not. We repeat this for all possible combinations of a vertex from the top layer and a vertex from the middle layer. In this way we generate connections between the top and the middle layer, and then we repeat the process for connections between the middle and the bottom layer. Once all vertices have been processed, we check whether or not there are vertices remaining that have no connections to the appropriate layers, and if there are, they are connected to a random vertex in the appropriate layers.

Once the network is generated we need to simulate a steady-state of the system. First we set the input, the amount of gas that flows into the system in the top layer, and the amount of gas that flows out of the system in the bottom layer, implicitly defining the amount of gas that is stored into or drawn out of the system. Most test were performed with a flow of 4000 (cubic meters of gas) coming into the network and a flow of 3800 exiting the network. The amount of gas that flows into the system is randomly divided over all vertices in the top layer. Next we use the mass-balance equations to divide the flow through one vertex between all its out-neighbours, with each out-neighbour receiving a random fraction of the total flow into the vertex. Once this has been calculated for all vertices in the top layer, we know the flow through each of the vertices in the middle layer, and we can use the same procedure to divide the flow over the bottom layer, finally resulting in a 'steady-state' of the network that satisfies the mass-balance equations.

The last thing to take into consideration is the amount of gas that is stored into or drawn out of the system, the linepack. Proportional to the flow through a vertex, part of the flow is not distributed to one of the out-neighbours, but instead assigned to a storage node. This is the case if gas is stored into the network. If instead gas is drawn out of the network, no gas is delivered to the storage nodes, but there is a flow from the storage nodes to the middle and bottom layer of the network, simulating gas being drawn out of the system. We chose to model the linepack in two different ways, and determining which method showed better result when applying DVR. The first option was to only use one storage node for the entire system, with a possible flow from the top layer to the storage if gas is stored, and connection from the storage to the bottom layer if gas is being drawn out. Examples of this option can be seen in figures 4 and 5. The other option we considered was to assign a storage node to every connection in the network, simulating gas being stored in the pipelines represented by that connection. Examples can be seen in the left side of the figures 6 and 7.

The flows to and from the storage nodes of the network are proportional to the flow through the relevant vertex compared to the total flow through the network. In the example of figure 4 for instance if one node in the top layer accounts for ten percent of the total flow through the top layer, then ten percent of the total amount of gas stored between layer 1 and layer 2 is accounted for by the same vertex; i.e. An amount of gas equal to ten percent of the total amount of stored gas between layer 1 and layer 2 will flow from the considered vertex to the storage node.

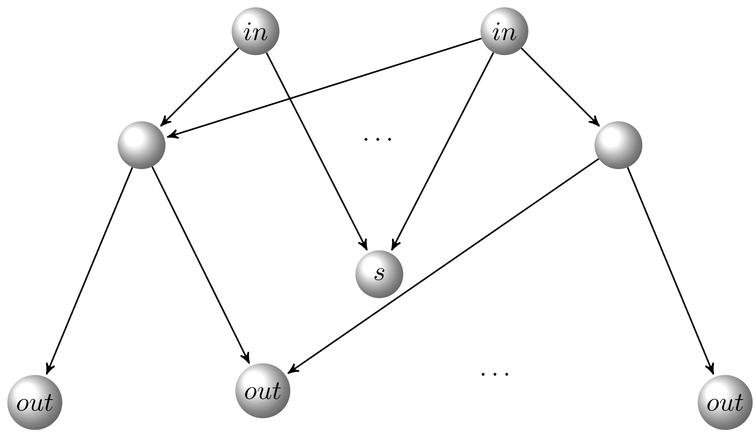

Figure 4: Gas being stored in a single storage node, denoted by s.

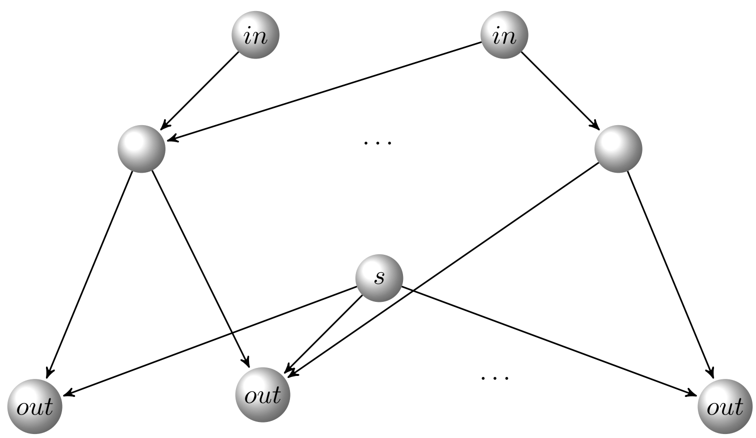

Figure 5: Gas drawn out of a single storage node, denoted by s.

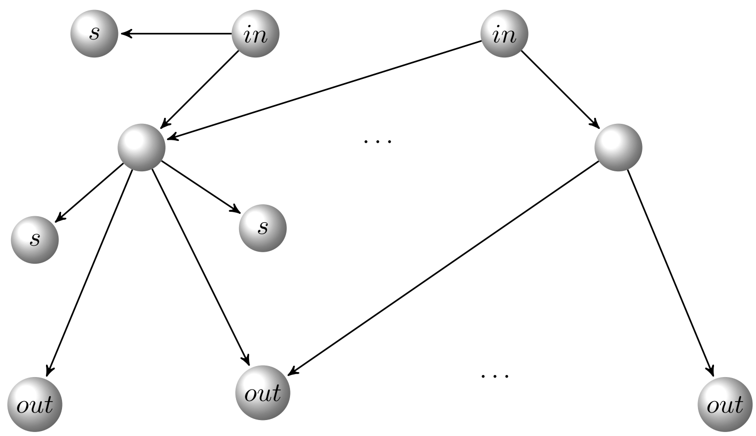

Figure 6: Gas stored in multiple storage nodes, indicated with s

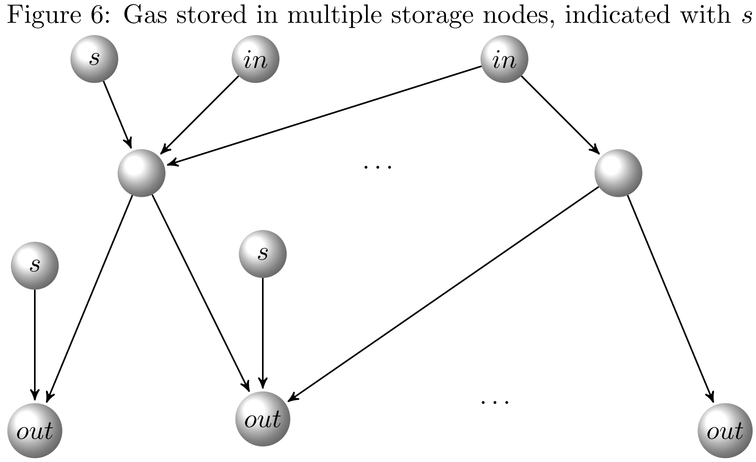

Figure 7: Gas drawn out of multiple storage nodes, indicated with s

Having generated the flows throughout the network and taking into account the storage of gas, we know the true steady-state of our simulated network. In order to generate measurement data, we need to perturb the true state using normal distributions, simulating the errors due to random noise. The measurements of the flow throughout the network are perturbed with normally distributed measurement noise around the true value with standard deviation equal to a certain percentage of the true value. The specific percentage used depends on the accuracy of the considered metering station, which is set in advance. On the basis of our new measurements, an estimation of the amount of gas that is stored into or drawn out of the system is made, and the flows to or from the storage nodes are updated such that the total amount of stored or drawn out gas is equal to the estimated value. (The flows to and from storage are of course unknown in reality, thus we can only estimate them on the basis of the difference between the measured flow into and out of the system, or alternatively, on the basis of pressure measurements used to estimate the linepack.)
Lastly, we added a systematic error to a randomly picked metering station by adding or subtracting a multiple of the standard deviation of that metering station to the measured value. We then applied DVR to the data with and without this systematic error, to see whether or not DVR could detect the presence of the systematic error.

2.6.2 Application of DVR

From the generated measurement data, we can obtain estimations using (12). Using our knowledge of the true state of the system we can perform several checks to evaluate the performance of DVR. The first check is to see if the estimation $\hat{y}$ is closer to the true value than the measurement y. For this we used the norm of the difference between $\hat{y}$, y and the true value as a measure of closenesss:

$$rec = ||x - y|| - ||x - \hat{y}|| \tag{13}$$

where x denotes the true value of the steady state. If rec in (13) is greater then zero, then we assume that the method is working correctly.
In order to detect the presence of systematic faults, we only want to use variables that would be available in practice. This means we can only use our measurement and estimation and try to detect a systematic error on the basis of the difference between those two. We constructed a vector containing the differences between the measurements and the estimations of the flow, relative to the sizes of the measured flow:

$$c_i = \frac{y_i - \hat{y}_i}{y_i}, \;\; i \in \{1, \cdots, n\}. \tag{14}$$

We compared the norm of the vector $c = (c_1, \cdots, c_n)^T \in \mathbb{R}^n$ for the data with a systematic error present, and the data without a systematic error:

$$d = \frac{||c_e||}{||c_n||},$$

where c_e denotes the vector c for the data with a systematic error and c_n denotes c for the data without a systematic error. If there was more then a factor 2 difference between these two we say that the method detects the presence of a systematic fault.

2.6.3 Networks containing sub-networks

A note that should be added to this is that the A matrix that we used in the implementation of DVR, was a simplified version of the true A matrix following from the mass-balance equations. To explain we look back to our example in figure 2. Instead of the A matrix given in 1 we could have chosen to use the following:

$$Ay = 0, \;\; A = \begin{pmatrix} 1 & 1 & -1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & -1 & -1 \end{pmatrix}. \tag{15}$$

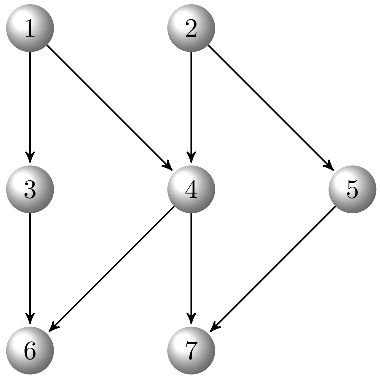

Figure 8: A layered network with no sub-networks in the bottom and middle layer.

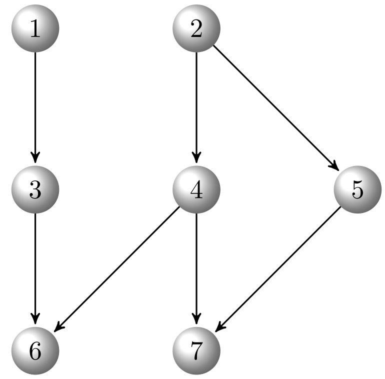

Figure 9: A layered network with two sub-networks in the top and middle layer.

This A matrix only caries part of the mass-balance equations, effectively reducing them to the the constraint that the flow through the first layer should be equal to the flow through the second layer, while disregarding that the flow through vertex 6 should equal the flow through vertex 3 and that the flow through vertex 7 should equal the sum of the flows through vertices 4 and 5. Thus information is lost.

The reason that there is a 'better' choice for A than the one given in (15) in this example is because there are subsets of vertices in the bottom and middle layer that have no neighbours in the middle or bottom layer outside of that subset. In the example in figure 2 we see that the vertices 3 and 6 have no neighbours other then each other in the bottom or middle layer. Similarly, the vertices 4, 5 and 7 have no neighbours besides each other in the bottom layer. We call such a subset of vertices not connected to any other vertices in the same layers a sub-network. If instead of the network in figure 2 we look at the network given in figure 8, no sub-networks exist and we would have to use the A matrix of (1). If instead we looked at the network in figure 9, there would be two sub-networks, one containing the vertices 1 and 3, and the other containing the vertices 2, 4 and 5.

The reason I chose to use the simplified A matrix is because otherwise I would first have to identify all possible sub-networks that existed in the network, and I estimated that the writing of an algorithm for that would take too much time.

Instead I chose to also construct a network in such a way that it contained such sub-networks, and run some additional test on that network.

The structure of that network is similar to that of the network in figure 10. The bottom layer has 1 vertex and the middle layer has 15 vertices. The top layer has 75 vertices, which are divided into 15 sets of 5 vertices. Each of these sets is only connected to one vertex in the middle layer. In these tests we chose to use the A matrix that carries all of the mass-balance equations. In addition we did not take into account the linepack in this case, so there are no storage nodes in the model, and the amount of gas that flows into the system is equal to the amount of gas that flows out of the system.

For these test we were also interested in whether or not the fault could be isolated, that is, whether or not the method could detect which of the measurements contained the systematic errors. In order to do so we added the systematic error to a vertex in the middle or top layer. We defined the values c_i as in equation 14 and selected a vertex j in the middle layer for which $|c_j| \geq |c_i|$ for all vertices i in the middle layer. If the systematic error was added to this vertex, or one of the in-neighbours of this vertex in the top layer, i.e. one of the 5 vertices in the top layer that are connected to the vertex j in the middle layer, then we say the fault was successfully isolated.

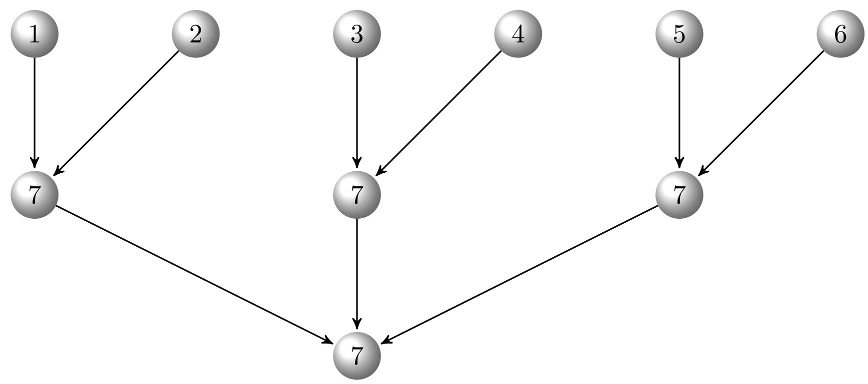

Figure 10: A layered network with three sub-networks in the top and middle layer.

2.7 Results

For most tests we generated a network consisting of 50 vertices in the top layer, 75 in the middle layer and 1000 in the bottom layer. The standard deviation for generating the measurements in the top layer is set as 1 percent of the true value of the flow, in the middle layer it is set as 10 percent of the true value of the flow and in the bottom layer as 2 percent of the true value of the flow. The flow into the top layer of the network we set equal to 4000 and the flow out of the bottom layer we set equal to 3800. These choices have been made to reflect knowledge about the network of the Gasunie. If we deviate from these values in any test, this will be indicated.

In order to simulate the fact that we would usually have multiple measurements over a certain time period at our disposal, we generate a number of measurements of the same steady state, by perturbing the true values multiple times to generate different measurements. For the data containing systematic errors, the error is added to the faulty sensor in each of the measurements. The resulting values are then averaged and used as the measurement vector for our application of DVR. Most test are performed using 60 measurements, if we deviate from this number this will be indicated.

First we ran a test to determine which of the two methods of modelling storage led to better results when using DVR for error detection. We used a systematic error of 2 times the standard deviation. The results can be seen in table 1. The method using a storage node for every connection in the network performs slightly better then the method using only a single storage node. However, we reset the seed used for the random generation in between the tests for the different cases. Consequently the generated network, flow, measurements and errors are exactly the same. We see that the second method gives better results in the exact same situation. Therefore we chose to use this model for the rest of our tests, so all subsequent test are performed using the model with a storage node for each connection in the network.

Next we ran tests for different sizes of the bias. The results can be seen in table 2. Even for a bias as small as 1 standard deviation of the measurement we see that there is some detection of error present. As we analysed the data we saw that this happens when the absolute error that is generated is large. This happens when the error is assigned to a vertex with a relatively large flow through it, and even more so when the measurement uncertainty of that vertex is high. The average generated measurement error in the case where an error was detected for a bias of 1 standard deviation was 489, while the overall average error in that case was 162. Additionally, we see that errors in the middle layer are detected more frequently than errors in the top or bottom layer.

To research this effect further, we ran test in which we put the error in a specific layer. The results can be seen in table 3. We see that the difference between proper detection of the fault is significantly higher in the middle layer. Notable as well is the extremely low rate of detection in the bottom layer. We also ran test in which we entered errors of different sizes in only the middle layer, to further investigate the performance of the method on errors in the middle layer. The results can be seen in table 4. We see that even with a relatively small error, detection of the fault is possible.

Next we varied the sizes of the flow throughout the network and the amount of gas stored into the network. The results can be seen in the tables 5 and 6. There is no clear trend present in these results.
Lastly, we also ran test for a different precision setting of the layers. Specifically, we set the standard variation in the measurements in the top and bottom layer equal to 20 percent of the true value, in order to see if the results would change if the bottom and top layer make more inaccurate measurements then the middle layer. The results can be seen in table 8. Detection of the faults in the top and bottom layer increase significantly, while detection of faults in the middle layer stays roughly the same.
We also ran some test on the network that we constructed to contain sub-networks. The results can be seen in table 9. The standard deviation of the measurements in the top and bottom layer is set to be 1 percent of the the true value of the true value of the flow, and the standard deviation of the measurements in the middle layer is set to be 2 percent of the true value of the flow. The total flow into (and out of) the network is set to be 4000. We see that these results are significantly better than the results when sub-networks are not considered.

2.8 Conclusion

2.8.1 Trends in the results

Despite the fact that we are limited to only steady-state simulations, the results of the previous section offer us some insight in the effectiveness of applying DVR on large-scale systems. The first observation that we make is that although the method does produce estimations that are closer to the true value than the measurements, the improvement is small. As we can see in table 1 the value of the variable *rec*, which can be seen as a measure for the closeness of the estimation to the true value relative to the closeness of the measurement to the true value, is small. This gives rise to doubts about the effectiveness of DVR in these large-scale networks, as it is the primary function of DVR to estimate the true value.
The next observation is that despite these doubts the method applied actually detects the presence of systematic error in the middle layer quite well. For a bias of 5 times the standard deviation we can see in tables 2, 3 and 4 that a large portion of the errors in the middle layer are detected. However, since the standard deviation of the measurements in the middle layer was set to be 10 percent of the true value, we need to remember that this means that the measurement deviates roughly 50 percent from the true value. For the top layer we can see from table 2 that a significant portion of the errors are detected when the error increases from 10 to 30 standard deviations. For the top layers this means a deviation of roughly 20 to 30 percent of the true value. In the bottom layer a significant portion of the errors is detected when we reach an error of 50 standard deviations, which constitutes an error in measurement of roughly 100 percent.
From tables 5 and 6 we see that the size of the flow through the system and the amount of gas that is stored or drawn out of the system do not greatly influence the effectiveness of the method in detecting the presence of systematic errors. We do see that the amount of detected errors is less when we look at situation where gas is drawn out of the system rather than stored.
We also see from figure table 7 that the effectiveness of the method increases as the number of measurements increase and we see from table 8 that strangely enough the method is more effective when the precision of the measurements in the top and bottom layer are reduced.
Lastly, in table 9 we see the results for different sizes of the bias in the network constructed to contain sub-networks. The result indicates that the method works significantly better when there are sub-networks present in the network and these sub-networks are taken into account when defining the A matrix. The method is in that case even capable of reliably indicating the vertex in the middle layer that is associated to the fault.

2.8.2 Considerations

Although some of the results from the previous section may seem counter-intuitive, I think they can be explained readily. The first thing to note is that the way I introduced systematic errors into the measurements was relative to uncertainty of the measurement to which the error was added. The result is that the larger

	single storage	multiple storages
$d > 2$	269	300
rec	0.28	0.29

Table 1: Results comparing different methods of modelling storage. the row labelled with $d > 2$ indicates the number of times an error was successfully detected, and the row labelled rec gives the average value of rec as in equation 13. The bias b is 2 times the standard deviation and the number of trials N is 1000.

	$b = 1$	$b = 2$	$b = 5$	$b = 8$	$b = 10$	$b = 20$	$b = 30$	$b = 40$	$b = 50$
$d > 2$	15	17	26	38	42	56	70	71	95
t	1	2	4	6	7	17	27	32	39
m	14	15	22	32	31	35	37	30	32
b	0	0	0	0	4	4	6	9	34

Table 2: Results for different sizes of the systematic error b. Sample size $N = 100$. The rows labelled t, m and b indicate how many of the detected errors were in the top, middle and bottom layer respectively.

	top	middle	bottom
$d > 2$	20	148	3

Table 3: Results for systematic faults present in different layers. $b = 5$, $N = 200$

	$b = 1$	$b = 2$	$b = 3$	$b = 4$	$b = 5$	$b = 6$
$d > 2$	33	52	63	69	72	80

Table 4: Results for different sizes of the systematic error b in the middle layer. Sample size $N = 100$.

in, out	$400, 380$	$4000, 3800$	$40000, 38000$	$380, 400$	$3800, 4000$	$38000, 40000$
$d > 2$	34	30	32	26	18	25

Table 5: Results for different sizes of the flow. $b = 5$, $N = 100$. The first row labelled in, out indicates the set amount of flow into the top layer and out of the bottom layer respectively.

in, out	$4000, 2000$	$4000, 3000$	$4000, 3800$	$2000, 4000$	$3000, 4000$	$3800, 4000$
$d > 2$	33	30	31	24	18	25

Table 6: Results for different sizes of the storage. $b = 5$, $N = 100$.

	$s = 4$	$s = 20$	$s = 60$	$s = 120$	$s = 360$	$s = 500$
$d > 2$	14	32	34	34	42	41

Table 7: Results for different number of measurements. $b = 5$, $N = 100$.

	$b = 1$	$b = 2$	$b = 5$	$b = 10$	$b = 20$
$d > 2$	34	50	92	77	84
t	21	27	37	27	26
m	13	19	29	38	34
b	0	4	26	12	24

Table 8: Results for different accuracies of the measurements. The top and bottom layer have a standard deviation of 20 percent of the true flow values, and the middle layer has a standard deviation of 10 percent of the true flow values in its measurements. $N = 100$.

	$b = 0$	$b = 1$	$b = 2$	$b = 5$	$b = 10$
$d > 2$ in	0	0	59	536	750
iso in	76	364	656	904	948
$d > 2$ mid	0	709	1000	1000	1000
iso mid	60	1000	1000	1000	1000

Table 9: Results for different size of the bias in the network containing sub-networks. The rows labelled 'iso in' and 'iso mid' indicates the number of times the method correctly identified the vertex in the middle layer associated to the systematic error when the error was added to the top and the middle layer respectively. $N = 1000$

the uncertainty, the larger the introduced error, in absolute terms. This can explain the results of table 8 among others. The sizes of the errors introduced in the tests considered in table 8 are the same relative to the measurement uncertainty in the middle layer compared to the tests related to table 2. However, in absolute terms, they are on average a factor 10 larger in the bottom layer, and in the top layer they have grown with a factor 20. This is due to the fact that we changed the measurement uncertainty of the bottom layer from 2 to 20 percent, and that of the top layer from 1 to 20 percent. If we for instance compare the results for the top layer in the column $b = 1$ in table 8 with the result for the top layer in column $b = 20$ in table 2 the results are a lot closer. Considering this we come to the hypothesis that an important factor in whether or not DVR can detect the presence of systematic errors in the measurements is the size of the error relative to the total flow throughout the system. This hypothesis is also in line with the results seen in table 5 and table 2. The results in table 5 show that the effectiveness of the method remains the same when we reduce the size of the flow and as a consequence also the absolute size of the error. The relative size of the error remains the same in all cases. The results in table 2 show that errors in the top and middle layer are easier to detect than errors in the bottom layer. We know that the flows through the vertices in the top layer are usually larger than the flows through the bottom layer, since the total amount of flow is roughly the same, but there are more vertices over which the flow is divided. The middle layer also has more vertices than the top layer, but the measurement uncertainty in this layer is higher. As a consequence an error added to the middle layer is usually still larger than an error added to the top layer, and errors added to the top and middle layer are usually larger then errors added to the bottom layer. The results given here are of course far from conclusive, and more research would be necessary to determine whether this hypothesis is valid.

If it is valid it does raise more questions about the viability of using DVR as a method of detecting systematic errors in measurements in these large-scale networks. If faults are detected only if they are larger than a certain fraction of the total flow through the network, then the method will never detect errors in measurements that concern only a small portion of the flow, and the accuracy of the made measurements plays no significant role in the detection of errors, which seems counter-intuitive.

Another important factor that may play a role in the effectiveness of DVR in detecting systematic errors in measurements is the position of the vertex to which the error is added. From the results we see that errors in the middle layer are detected more often than errors in the other layers. A possible explanation is that for a vertex in the middle layer, information about both the flow into, and out of the layer is available, as we know the flows through the layer above and below it. For the bottom and top layer we have only one layer with which we can compare the measured values. The result is that a deviation in the middle layer disagrees with more measurements, and will result in larger deviations from the total measurements when adjusting it.

The last consideration we make is that the method works significantly better if the potential sub-networks contained in the network are taken into account when defining the A matrix, even allowing for an indication of where the systematic error enters the network. Smaller faults can be detected then when sub-networks are not accounted for and with more reliability.

However, despite the positive results on the possibility of detecting errors in the middle layer, and on the detection and isolation of faults for more structured networks, there is quite some criticism possible on the methods that are employed here.

2.9 Discussion

The most important point of criticism is about the fact that the simulation I use is a steady-state simulation. This was the only method that was available to me, but in some aspects it is severely lacking. For instance, We tried to take the linepack into account using this steady-state simulation, however, storing gas into the network during the measuring period is inherently not a steady-state process. As more gas is stored in or drawn out of the network, the pressure rises in those parts of the network where there is now less or more gas stored. This in turn influences the dynamics of the flow. As a result the system is not in steady-state, contradicting the necessary assumption for the application of DVR. The conclusion is that DVR can not realistically be applied on a network which is subject to considerable changes in the stored amount of gas

in the network itself. Thus the results for some of the test that we ran, especially those in table 6 are not necessarily meaningful.
Related to this point is that fact that the two most important factors discussed in section 2.4, that of the linepack and the delay, need to be considered partly because they make it harder to achieve the steady-state that we require the system to be in. The application of a method which relies on a steady-state to be effective, on a system which has difficulty attaining such a steady-state, is perhaps not a viable option for detecting errors in the measurements of said system. This problem is underlined by the fact that DVR, although apparently capable of detecting errors in the middle layer reliably if the steady-state assumption holds, does not produce estimation that are much closer to the true value then the measurements. All in all it seems that applying DVR to these large-scale systems is not a very effective method.
For the other factors discussed in section 2.4, DVR has some benefits. First, DVR can be easily adapted to a changing network structure, as all that needs to be done to incorporate a change is to adapt the A matrix accordingly.
Second, the fact that DVR has an analytical solution available means that we do not need to use numerical algorithms to obtain estimations of the flow throughout the network, which reduces the necessary computation time. As the heaviest calculation that needs to be performed in order to obtain an estimation is the inverse in equation 11, computation time should not be a problem.
Lastly, DVR clearly takes into account the uncertainty of the individual measurements in order to arrive at an estimation.
As a final observation, the positive results concerning networks containing sub-networks may be very useful. For a system that can reasonably be assumed to be in steady state, it may be used to detect faults and isolate them reliably.

2.10 Concluding remarks

My conclusion is that DVR is most likely not an effective method for detecting systematic errors in measurements concerning large-scale gas networks. If the system can be assumed to be in steady-state for significant periods of time, then the method can be used to detect systematic errors in measurements concerning large flows, by comparing the difference between estimation and measurement to a preset threshold. The method is likely more successful in detecting systematic errors present in a measurement point that does not receive gas from or delivers gas to outside of the network. If such steady-states can be assumed to occur, DVR has some desirable properties: It is easily adapted to temporary or permanent changes in the structure of the network, it is not computationally heavy, and it makes use of available knowledge about the precision of different measurements.
However, since the method requires a steady-state, it can not be relied on as soon as linepack becomes a significant factor, or if the system is not often in steady-state.
Lastly, when sub-networks are present, and the method takes these sub-networks into consideration the results indicate that DVR is effective at detecting the vertex, or at least the associated sub-network, where the fault is present. This should be investigated further before any conclusions can be drawn.

3 MECADA

The first project I worked on is called MECADA. As explained in the introduction this project is concerned with trying to assess the performance of a meter by analysing the data that it produces. The first step of the project was to get familiar with the specific type of meter that we would be considering, a so-called ultrasonic meter, and the typical faults and difficulties encountered in gas flow metering. In this chapter I will explain shortly the motivation behind the project, the workings of the considered type of meters, and why I did not pursue this project any further.

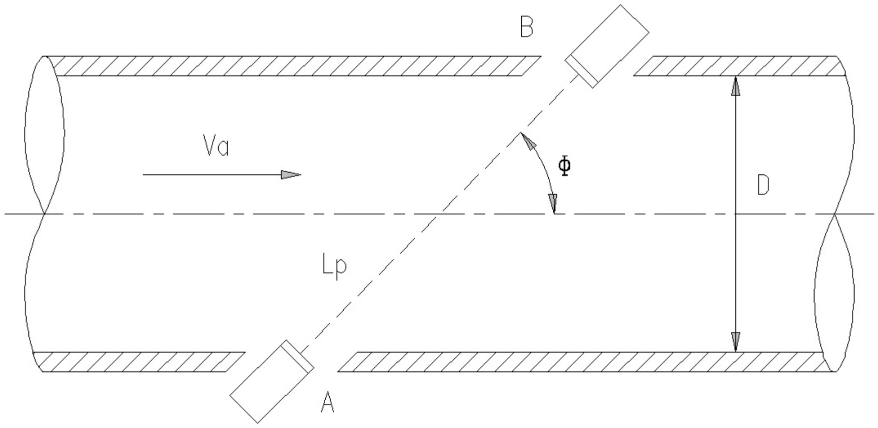

Figure 11: simplified set-up of an ultrasonic meter

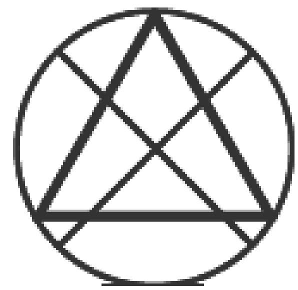

Figure 12: front-view of signal paths through a pipe, each line segment indicates a signal path, thus we see five paths here.

3.1 project motivation

The meters we are considering are used in gas flow networks to estimate the flow rate of gas that passes though a certain point. In large networks such as the gas distribution network of Gasunie in the Netherlands the amount of gas that passes through a meter station can be considerable, and small errors in the measurements can therefore lead to sizeable errors in the estimation of the amount of gas that has flowed into or out of the network at a certain point. This of course leads to the receiving party paying either too much or too little for the actual received amount of gas, and thus we see that it is in the interest of both parties to get accurate estimates of the gas flow. On the other hand the procedure to assess the performance of a meter involves removing it and transporting it to a testing facility, which is a costly process. A method that allows us to estimate the performance of a meter at the hand of the data it generates would allow us to more efficiently select the meters that should be tested or re-calibrated at a testing facility. This would remove the need to go to the costly testing procedure for meters that are performing well while detecting faulty sensor early on.

3.2 Ultrasonic meter

The ultrasonic meter measures the time it takes for a sound-wave to travel from one point in a pipeline to another. The principle is illustrated in figure 11. in this figure we see a pipeline with two sensors, A and B, installed. Both sensors emit and receive sound signals and the time difference between the sending and the receiving of a signal is measured. The relation between these measurements and the relevant variables is given by the following equations:

$$t_{AB} = \frac{L_p}{C + V\cos(\phi)} \tag{16}$$

$$t_{BA} = \frac{L_p}{C - V\cos(\phi)}. \tag{17}$$

Here t_{AB} is the travelling time of the signal from sensor A to sensor B, t_{BA} is the travelling time form sensor B to sensor A, L_p is the length of the path the signal has to travel, C is the speed of sound in the current gas composition, V is the velocity of the gas (in figure 11 given by V_a) and ϕ is the angle between the direction of the pipe and the path the signal has to travel.

Rewriting these equations allows us to estimate the speed of sound and the gas velocity on the basis of t_{AB} and t_{BA}:

$$V = \frac{L_p}{2\cos(\phi)}(\frac{1}{t_{AB}} - \frac{1}{t_{BA}}) \tag{18}$$

$$C = \frac{L_p}{2\cos(\phi)}(\frac{1}{t_{AB}} + \frac{1}{t_{BA}}) \tag{19}$$

In practice a meter consists of multiple sets of sensors, and the path between a set of sensors bounces off the side of the pipe several times, so as to cover a larger section of the pipe with one path. These paths are positioned in such a way that they cover different parts the cross section of a pipe; i.e. some of the paths are located towards the sides of the pipe, and other paths go through the middle, an example can be seen in figure 12. Since gas that travels close to the side of the pipe experiences friction it usually travels slower than the gas that flows through the middle of the pipe. By positioning the sensors in this way the measurements allow us to estimate the flow profile, that is the velocity of the gas throughout a cross section of the pipe.
Different types of disturbances are associated with different flow profiles and the estimations produced by a meter are used in this way to determine if and what kind of disturbances are present in the flow. In addition to the estimates of the speed of sound and the gas velocity, the meter also reports a gain factor for each sensor individually, which is the result of internal diagnostics, and is automatically adjusted by the meter to ensure that all signals are received.

3.3 Methodology

In order to find a method of detecting faulty behaviour in a meter we looked at supplied data for several meters, usually consisting of time-series data in several periods between 2011 and 2018, in combination with results of performance assessments of a meter when they were taken to a testing facility. As a starting point, and as a way of getting acquainted with the data, I applied some linear regression to the various variables in the data to see whether or not I could detect any trends. In addition I implemented subspace identification to see if there were any relations in the data that could be described by a (discrete-time) linear time-invariant system. As this is not the main project I worked on I will not go into the detail about this implementation of subspace identification. The full procedure can be found in chapter 9 of [1]. One condition for the application of the procedure to give valid results is that the dimension of the state of the generated system should be smaller than the number of measurements s used in the data equation. Since this was not achieved in several data sets for any choice of s I concluded that this method did not give any helpful results for the considered problem.
Later on I realised that the measurements given in the data were fifteen minutes apart, which makes it unlikely for the system identification method to find any valid relation between two sequential measurements, as any effect of the physical relationships between the measurements should have diminished to undetectable levels in that time-period.
Thus I went on to trying to see if any of the trends derived from the linear regression analysis, that were present in a time period between two performance assessments of a meter, were coupled to a deterioration (or improvement) of performance as concluded from the performance assessments. This is were I ran into a problem. The time-period between any known data of a meter and the closest performance assessment was usually in the range of several months to a year. There was only one case that I could find were an assessment was made directly following a period of available data, and this concerned very brief period of available data. Even more problematic was the fact that the results of a performance assessment were not always well-documented. If it was concluded that a re-calibration was necessary, the performance of the meter was only reported after the re-calibration, which means that there was no way to quantify the error development of the meter in the period in-between measurements.
On the basis of these findings I concluded that the data was most likely insufficient for determining a method of detecting faulty behaviour by looking at the produced measurements of a meter.

4 Personal reflection

In this last section I will give a short personal reflection on my internship at DNV GL.
As a starting remark: the working environment at DNV-GL was, for me at least, pleasant. Coming in I was unsure of what I could do for DNV GL and what would be expected of me, but my supervisor made clear that I should not worry about whether or not my internship would be a 'succes'. This put me at ease, but despite that, now that the internship is coming to an end I am not entirely satisfied with the work that I have done here.
I lost some time with finding a proper project to work on related to what I've learned in my study program so far. DNV GL supplied several projects for me to consider, and in an attempt to find some way in which I could use my knowledge to help with these projects I read a lot of articles concerning the modelling of gas flow networks, detecting sensor failure, and fault detection and isolation, with very few turning out to be useful. I struggled with thinking of a way to tackle the problems of the project without an available model. Although I think I did what I could without a model, I'm not very satisfied with the results. In hindsight I should have been more assertive and should have consulted more with my supervisor in order to find a way to work on the projects. That could have saved me time. The more positive results, concerning networks that contain sub-networks, I found on the last day of my internship, which sadly gave me no time to look into that further. If I had consulted with my supervisors more I might have realised earlier that I missed this idea, and this might have led to more useful and more interesting results.
On the other hand, I think the internship was very useful for me personally. I spent some time learning my way around the programming language python and I gained a scattering of knowledge about the physics and modelling used in the regulation of gas flow networks. Additionally, I learned a little about the working environment I can expect when I enter the job market. I learned that I need to gain a clear image of what a project entails, what the questions are that need to be answered, and what the goals and the available means are, before I start working on it. I learned that I tend to work on my own, and that it would be better to consult more with others, discussing the problems I've run into. As I write them down these seem to be revelations that are as clear as day, but perhaps they are best understood when learned from experience.

References

[1] M. Verhaegen, and V. Verdult. *Filtering and system identification: A least squares approach.* Cambridge University Press, Cambridge, 2007.

[2] A. Herrán-Gonzáles, J.M. De La Cruz, B. De Andrés-Toro, and J.L. Risco-Martín. *Modeling and simulation of a gas distribution pipeline network.* Applied Mathematical Modelling, 33, 1584-1600, March 2009.

[3] U. Faigle, W. Kern and G.Still. *Algorithmic principles of mathematical programming.* Kluwer Academic Publisher, 2002.

[4] E. de Klerk, C. Roos and T. Terklay. *Nonlinear optimization (CO 367).* Lecture notes, University of Waterloo, (2004).