Leif Andreas Hirsti

Estimation of wax and hydrate deposition in a Cold Flow cooling system.

Modeling, simulation and implementation.

Master's thesis in Industrial Cybernetics
Supervisor: Jan Tommy Gravdahl
June 2019

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Leif Andreas Hirsti

Estimation of wax and hydrate deposition in a Cold Flow cooling system.

Modeling, simulation and implementation.

Master's thesis in Industrial Cybernetics
Supervisor: Jan Tommy Gravdahl
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Abstract

Wax and hydrate deposition have been a major problem for the petroleum industry for some time. EMPIG AS have created a Cold flow cooling system to make sure that wax and hydrate deposition are removed in a 300 m pipeline to fit inside a container. They use a mechanical arm to find and heat up wax and hydrates. They have had a problem finding wax and hydrate location and thickness. This is to save resources and avoiding heating the whole pipeline.

In this thesis, an algorithm to estimate the location and wax thickness has been created. Starting with a model in COMSOL Multiphysics to make three models of a 150 m flowline and simulate pressure profile with wax deposition at 50 m, 75 m and 100 m. Furthermore, creating state-space equations of pressure and flow states to later be implemented into MATLAB has been done. Stability and observability were areas that also had to be covered. The flowline was divided into `n` control volumes and was iteratively added a wax deposition of 0.5mm and 1mm. To estimate the location and thickness was done with a discrete Kalman filter. The results showed that estimation in the middle gave a result with the smallest error. Discussion and future work were conducted about possible errors and what EMPIG and others must do for getting a better estimation. In conclusion, it is stated that the estimation for location and thickness works but need further work to become accurate enough for EMPIG's cooling system.

Sammendrag

Voks- og hyrdratavleiring har vært et problem i petroleumsbransjen en stund. EMPIG AS har utviklet et Cold flow kjøling system. Det er et 300 m langt rørsystem plassert inne i en container for å ha et kontrollert område hvor voks og hydrater kan bygge seg opp, for så bli varmet opp av en robotarm. EMPIG har hatt problemer med finne lokasjonen av voksavleiring og tykkelse i dette kjøling systemet. Dette er for å spare ressurser slik at de slipper å varme opp hele røret.

I denne masteroppgaven har det blitt utviklet en algoritme for å estimere vokslokasjon og tykkelse. Det startet med å modellere og simulere tre 150 m lange strømlinjer og plassere voks på 50 m, 75 m og 100 m. Vokstykkelsen ble satt til 0.5 mm og 1mm. En tilstandsrommodell av strømlinjen ble laget med trykk og strømning tilstander med `n` noder. Dette ble implementert i MATLAB. Tilstandsrommodellen ble sjekket for stabilitet og om den er observerbar. Iterativt fikk hver kontroll volum et mindre areal for å tilsvare vokstykkelse på 0.5mm og 1mm. Etter dette så ble et diskre Kalman filter utredet og implementert. Resultatet viste seg den beste estimering var i midten av strømlinjen. Diskusjon og framtidig arbeidet har vært utredet der eventuelle feilkilder og hva EMPIG og andre aktører må forbedre før man kan teste estimeringen i fullskala. Konklusjonen er at voks og hydrat avleiring og tykkelse kan bli estimert, men man gjøre noen modifisering for at modellen skal bli lik kjøling systemet.

Preface

The material presented in this report is the results of master thesis for the degree in Master of Science in Industrial Cybernetics at the Norwegian University of Science and Technology (NTNU) and was conducted during the spring of 2019. The project was conducted and in cooperation for EMPIG AS in Trondheim.

MATLAB and SIMULINK have been used in the project for larger and time-consuming computations. COMSOL Multiphysics is a modeling and simulation software from the Faculty of Engineering at NTNU that has been used frequently in this thesis.

This thesis is based on my preliminary study, “Introduction to Cold Flow – an understanding of wax and hydrate deposition in cooling systems”, which was conducted during the fall of 2018. Part of chapter 1 and chapter 2 is based on outlines of that work.

Leif Andreas Hirsti
Trondheim, 4th June 2019

Acknowledgment

There are several persons that have contributed academically and with support during this project. I would therefore firstly like to thank my supervisor at NTNU, professor Jan Tommy Gravdahl. His great motivation and guidance have helped a lot in the project, and he has shown great interest in my work.

From EMPIG AS, I would like to thank the co-supervisors Fredrik Lund and Per Jonny Nutudal. They have been giving guidance and technical information since the preliminary study in the fall of 2018.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Wax and hydrate deposition have been a problem for the oil and gas industry for many years. Clogged pipelines occur when seawater cools down the multiphase flow from the well. This have caused an economic disadvantage for the industry.

In the last years prestudies by SINTEF about Flow Assurance (FA) and specific Cold Flow (CF) has been done in small scale testing to see if hydrates can be removed with a feedback loop with seed crystals (SINTEF, 2010). This had a good effect for hydrates, and not so good for wax.

Furthermore, in 2017 EMPIG AS was granted funding through the Large-scale Programme for Petroleum Research (PETROMAKS 2) by the Research Council of Norway (Empig, 2019a). EMPIG AS has their main office in Trondheim and has been working with CF since 2011. The spring of 2019 EMPIG has been testing their product for removal of wax and hydrates on a large-scale at SINTEF Multiphase-laboratory. It is a cooler that has a mechanical arm that heats up the pipeline where wax and hydrates occur, a schematic illustration is shown in 1.1.

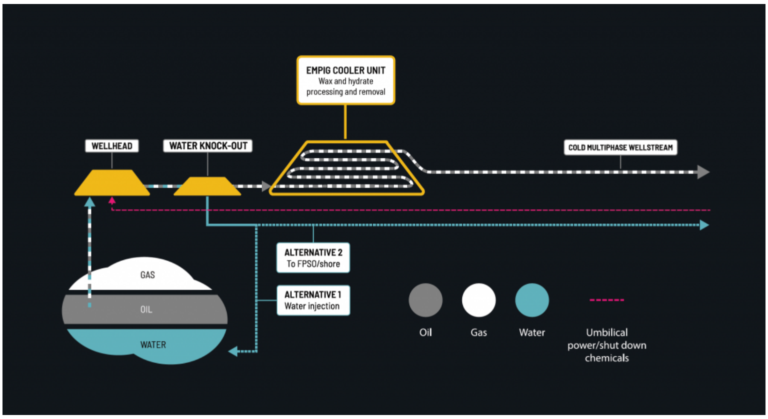


Figure 1.1: Schematic illustration of the EMPIG technology – compact cooler (Empig, 2019b)

1.1 Potential and problem formulation

During the large-scale testing at SINTEF Multiphase-laboratory there have been a usage of many instruments, like temperature, pressure and flow sensors shown in fig. 1.2. This has worked well, when the mechanical arm heats up the wax to make the wellstream flow smoothly. The problem is that the pressure and flow instruments are intrusive sensors, which has a great probability for leakage. Furthermore, intrusive sensors have a higher price than non-intrusive sensors. It is beneficial that the number of intrusive sensors is as low as possible, and still detects where the wax and hydrate are located.

The following problem formulation has been constructed to optimize the usage of instrumentation to the cooler.

> *Finding an algorithm to locate wax and hydrates inside the Cold flow cooling system. Using software-based methods due to minimization of instrumentation on the cooler. Furthermore, making an algorithm to the wax and hydrate concentration at the estimated positions. Lastly, selecting values for the initial conditions to the algorithm.*

1.2 Prestudies

This thesis report is based on a prestudy, conducted by the author himself during the fall of 2018 and described in the unpublished work, Introduction to Cold Flow – an understanding of wax and hydrate deposition in cooling systems, (Hirsti, 2018). The objective in the prestudy was to find different low-cost approaches to detect wax and hydrates depositions.

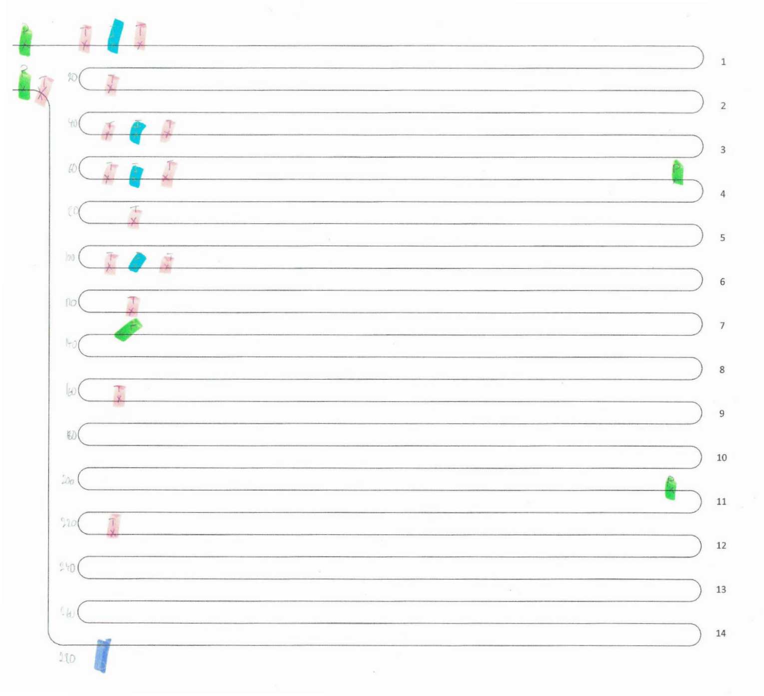


Figure 1.2: Illustration of the EMPIG instrument usage for the cooler. Green is pressure, blue is flow and red is temperature.

As mentioned in the introduction SINTEF has done studies about Flow assurance, named Saturn Cold flow project which gave good results for hydrates been removed by seeding. Moreover, other approaches such as ultrasound was investigated as a method to see the thickness of the wax deposition. Developed deposition sensors was also considered for the cooling system. The details of why the concepts was not used will be explained in the next chapter.

“Leak Detection in Pipelines by the use of State and Parameter Estimation” by (Oven, 2014) is a master thesis that uses flow and pressure equations to estimate flow and pressure loss. An attempt to modify the state and parameter may be used to estimate the wax and hydrate deposition in the cooler.

1.3 Scope

The assumption that the previous research by (Oven, 2014) can be used to solve this problem is not enough. These questions must be answered in this thesis.

1. We need to find a state-space model for pressure and flow.
2. Check if the state-space model is observable with the observability matrix.
3. Making a simulation for a pipeline using COMSOL Multiphysics and implement the results into MATLAB.
4. Creating a Kalman-filter and implement the simulation to verify that the method can be used for simulation in a controlled environment.

1.4 Outline

Chapter 2 is about the background research from the preliminary study that was done in fall of 2018. Investigating in what type of existing technology and methods that are already developed and might be used to the cooling system to EMPIG.

In chapter 3, gives a description of the modeling and simulation of a pipeline that can be postprocessed into MATLAB. The simulation includes pressure measurements that are going to be compared to estimated values.

Furthermore, in chapter 4 is about deriving the state-space equations by (Oven, 2014) and modify them for this problem. An algorithm for the state-space equations that is easy to change in size will be shown in this chapter. The observability matrix for the state-space equations has also been set up to see if the system is observable.

Chapter 5 is about the discrete Kalman filter, how it is derived and how to implement it to SIMULINK.

Moreover, in chapter 6 the results and plots of the estimated pressure. The COMSOL simulation are used to compare the estimation.

In chapter 7, the discussion of possible errors and future work that the author himself has noticed but did not have the opportunity to add or change.

The last chapter is a summary and a short conclusion of the work in this thesis.

Chapter 2

Background

During the fall of 2018, the specialization project was meant to give an understanding of how the Cold Flow system worked and what kind of methods had been used in previous years. Research of getting an understanding of CF was also crucial in creating a system that can work to measure wax deposition. Moreover, this chapter will also give a recapitulation on what technology was available and explain if they were considered for future work or not.

2.1 Saturn Cold Flow

Cold Flow started with SINTEF research and the purpose to make oil extraction environmental friendly (SINTEF, 2010). To be defined as CF, these three criteria were set by SINTEF (Larsen, 2008, slide 5):

1. No use of chemicals to prevent deposition, neither for hydrates or wax, and no "emulsifikatorer"
2. No warm up of pipelines or components
3. No isolation of the pipelines

The principle is to create seeds that are going in a feedback loop to the hot stream well. Moreover, the seeds contribute such that hydrates does not clog the pipeline. A simple figure of the CF concept is shown in fig. 2.1 (Hirsti, 2018, p. 7).

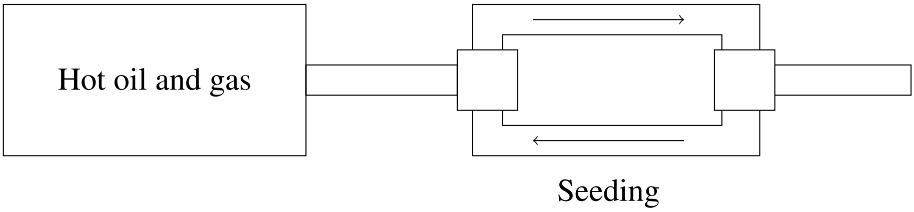
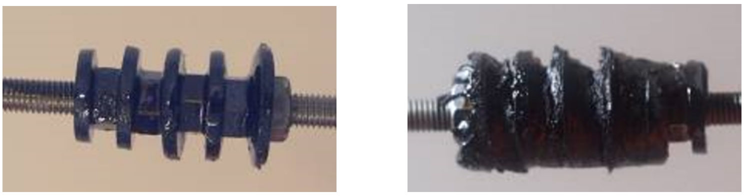


Figure 2.1: A simple set up for the concept of Cold Flow. Where the arrows indicate the seeding direction.

The method gave a good result for hydrate deposition removal in fig. 2.2. Pigging showed that the hydrates was almost gone with CF, and without CF it is a large amount of hydrate deposition which will after a while clog the pipe. EMPIG used this principle to make a large-scale cooler that removes wax and hydrate deposition. In addition, EMPIG heats up the pipeline in the cooler with a mechanical arm for a faster removal. Knowing the precise location will optimize the need of the mechanical arm and unnecessary power consumption.



(a) Pigging of the test pipe using CF.

(b) Pigging of the test pipe not using CF.

Figure 2.2: SINTEF test of CF process (Larsen, 2008, slide 9).

2.2 The principle of leak detection

In 2014, it was developed a method to estimate leakage in pipelines. Leakage can damage the environment and economical loss (Aamo, 2016). The software-based approach uses pressure and flow equations to estimate the leakage with a quick convergence. In fig. 2.3 shows the plot of the leakage estimation. As mentioned in section 1.3 if the same method may be used to estimate the wax and hydrate deposition.

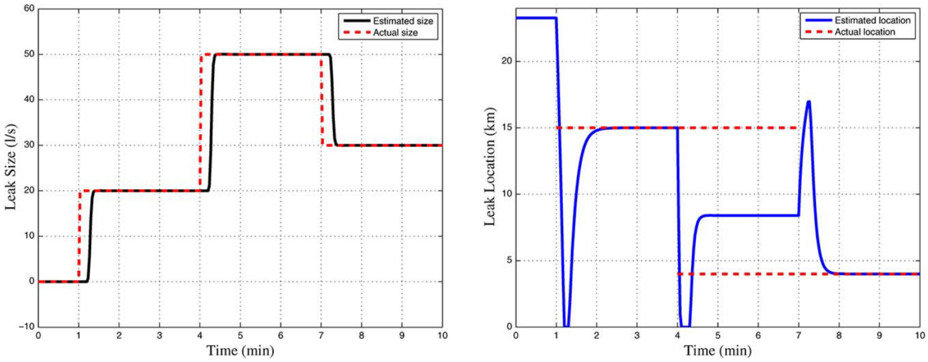
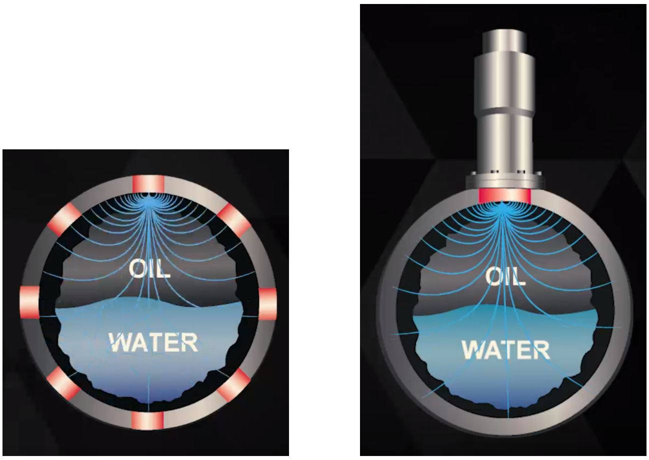


Figure 2.3: Plots of the simulation (Aamo, 2016, p. 250)

2.3 Deposition sensors

Rocsole makes deposition sensors for the petroleum industry. The pipe and plug in fig. 2.4 are intrusive sensors attached to the pipeline, with a flanged connection on both sides of the sensor and intrusive installation on top of the pipeline, respectively. Both sensors are using voltage injecting cycles. When all the injections are done, measurements form the electrodes give an image of deposition and liquids (Hirsti, 2018, p. 9).

Since these are permanently installed to the pipeline makes them not ideal for the cooling system to EMPIG. The static placement will only measure deposition at that specific location. EMPIG wanted a dynamic measurement because of the unpredictable mixture of multiphase flow in the well stream.



(a) Pipe sensor (b) Plug sensor

Figure 2.4: Rocsole sensors (Rocsole, 2018)

2.4 Thickness estimation

In 2013, Statoil petroleum patented a product for estimating the thickness of deposited material on a surface. The main purpose is to use it for Subsea, located on the Topside of a facility (Statoil Petroleum AS and World Intellectual Property Organization, 2014, p. 6). Furthermore, the patent is illustrated in fig. 2.5, where (1) is a pipeline that has a fluid containing pumped oil and gas. Seawater is pumped up by (8) to pipe (7). Adjusting the flow rate of the seawater makes the same thermal conditions. Inside (7) a heat pulse is sent through the pipe to measure the response (Hirsti, 2018, p. 10). Because the installation is on the topside of a facility makes it not compatible with EMPIG cooler.

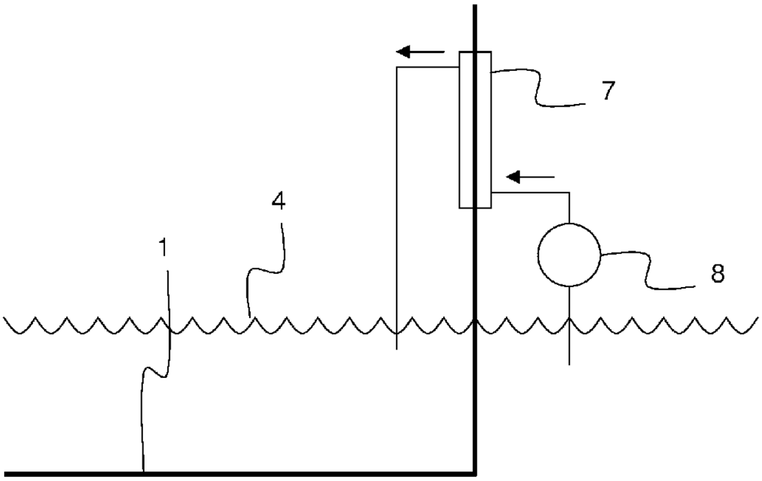


Figure 2.5: Illustrates schematically a sea-water filled annulus disposed around an insulated pipeline (Statoil Petroleum AS and World Intellectual Property Organization, 2014, p. 1)

2.5 Ultrasound instrumentation

Ultrasound is widely used in the industry and health services. Using soundwaves to see objects that are not visible, like human organs or thickness in materials. EMPIG asked if this was an approach that could be used see the wax and hydrate deposition.

Installing ultrasound to the cooling system will not work in this case, the reason is reflection coefficient (RF). RF is calculated by solving eq. (2.1) where, Z_1 and Z_2 are acoustic impedance of crude oil and steel, respectively. Acoustic impedance (Z) is the conductivity of sound (Brekke, 2018). It is defined as density (ρ) times the speed of sound (c) of the material. Since Z_1 and Z_2 are so unequal the soundwave will reflect almost immediately at the boarder between the steel pipe and fluid. The image processing of the fluid will visualize close to nothing.

$$RF = \frac{Z_2 - Z_1}{Z_2 + Z_1} \tag{2.1}$$

Chapter 3

Pipe flow model

Modeling the estimator directly on the cooler may result in many errors. The cooler EMPIG has constructed is 300 m long. It is constructed such that it can fit inside a standardized container. The CAD model in fig. 3.1 shows the setup. Furthermore, a shorter pipeline with a controlled environment must be constructed to verify that the estimator gives a good result.

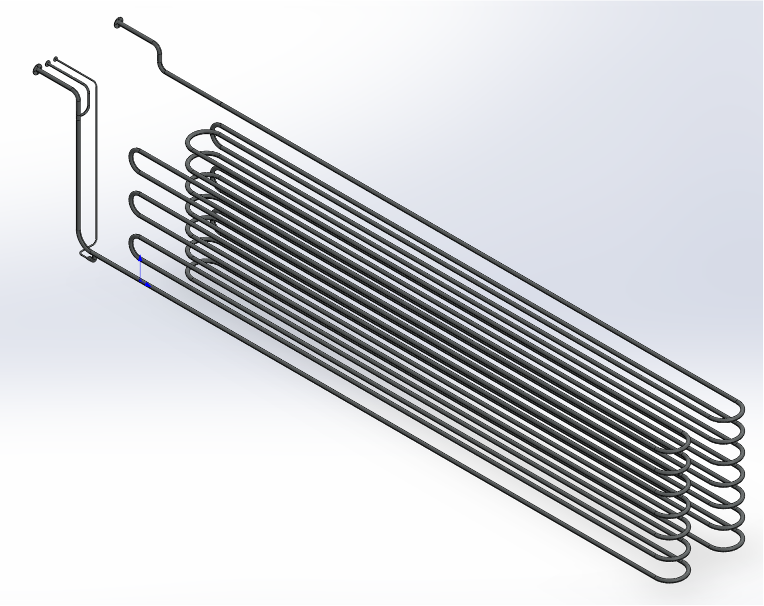


Figure 3.1: The CAD of the pipeline cooler. Courtesy by EMPIG.

3.1 COMSOL Multiphysics

COMSOL Multiphysics is a software for modeling and simulation of physics such as acoustics, chemical, fluid, heat and mechanics. These physics can be used individually or combined to get an estimated result. The more complexity added to the model will increase the computation time of the simulation. For this thesis, a high-performance laptop still required a simple model with one physic.

3.2 COMSOL geometry

The geometry was created by making a 150 m pipeline with a placement of the deposition. Because of the long computation time of the simulation, three geometries were created with various placement of deposition at the middle, front and end of the pipeline, respectively. Moreover, in the fall of 2018, EMPIG listed the parameters for the dimensions of the pipeline in table 3.1. It was also stated what wax deposition thickness (r_wax) is acceptable before heating up the specific section of the pipeline. The additional parameters where added to make the implementation easier in COMSOL. In table 3.1 there are different sizes of the dimensions. COMSOL has a default setting in meters and converts every other dimension to meters automatically.

In this thesis, dimensioning of the thickness of pipelines is not part of the scope. COMSOL is making simplifications given only dimensions for the flowline. The software assumes that everything on the outside is a wall on the boarder. In the simulation this is accounted for and calculates that the speed of the fluid is 0 m/s at the wall (r_inner).

3.3 Materials

Since the flowline includes multiphase flow makes the model increase in complexity. For simplification purposes, a single phase is used. Selecting the correct material to the flowline has an impact on the flow and pressure equations. COMSOL has a material library for gas, fluids and solids. Since crude oil is not a part of this library, another material was selected with close material properties. Engine Oil has similar density with crude oil and

Table 3.1: Parameter for COMSOL with deposition placed in the middle.

Parameter	Value	Description
r_inner	25.4[mm]	Radius of inner diameter of the pipeline
r_outer	30.4[mm]	Radius of outer diameter of the pipeline, not used
h_pipe	150[m]	Length of the pipeline
r_wax	24.9[mm]	0.5 mm wax deposition thickness
h_wax	10[cm]	Length of wax deposition
wax_place	74.95[m]	Center placement of wax deposition
x0	0[m]	Origo x
y0	0[m]	Origo y
z0	0[m]	Origo z

similar dynamic viscosity (Lundberg, 2018) and (Biltema, 2019, p. 8).

3.4 Physical

In the prestudies, it was verified that the flow inside the pipeline is turbulent (Hirsti, 2018, p. 22). COMSOL has several flow physics and many turbulent solvers. The most commonly used turbulence model is the k-ϵ. It is a more stable and converge easier to the solution (Lyu, 2016, 00:19:50). Selecting this physic model is a safe choice because of the common usage. Moreover, the initial conditions were set in physic setup. The inlet flow where specified by EMPIG (Hirsti, 2018, p. 23). Pressure inlet and outlet are selected with the values in table 3.2, this is to prevent negative pressure in the simulation. Gravity was an added feature to make the models more realistic. All the models have the same initial conditions, but the only dif-

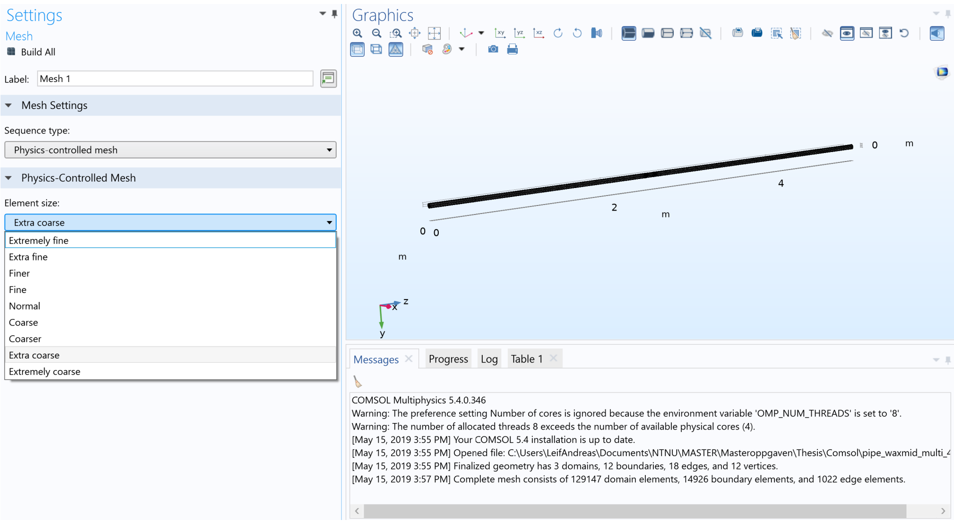
Table 3.2: Selected initial values for COMSOL Multiphysics.

Parameter	Value	Description
Gravity	g_const	Gravity value in COMSOL placed in positive y-direction
Inlet pressure	75 [Bar]	Assumed value for simulation purpose
Outlet pressure	72 [Bar]	For not getting negative values in COMSOL

ference are the wax deposition locations at 50, 75 and 100 meters. These are midpoints with wax covering 5 cm on each side.

3.5 Meshing

Meshing has a major impact on the computation of a simulation. Smaller mesh with many will intuitively have a longer computation time and use more storage. The result will be more accurate. COMSOL has an automatic meshing sequence with several options, see fig. 3.2. Alternatively, custom made meshing can be made in COMSOL. If the geometry is complex and some parts of the geometry is more important than others. For this case, a fine mesh with big element sizes has been selected. Where the deposition has a smaller cross-section, an extremely fine mesh has been created. COMSOL had problems with the fine mesh at the smaller cross-section where the deposition placement is located. This was solved by making the minimum element size smaller and worked well. The mesh has been set to minimize the computation time for the simulation.



Label: Mesh 1

Sequence type: Physics-controlled mesh

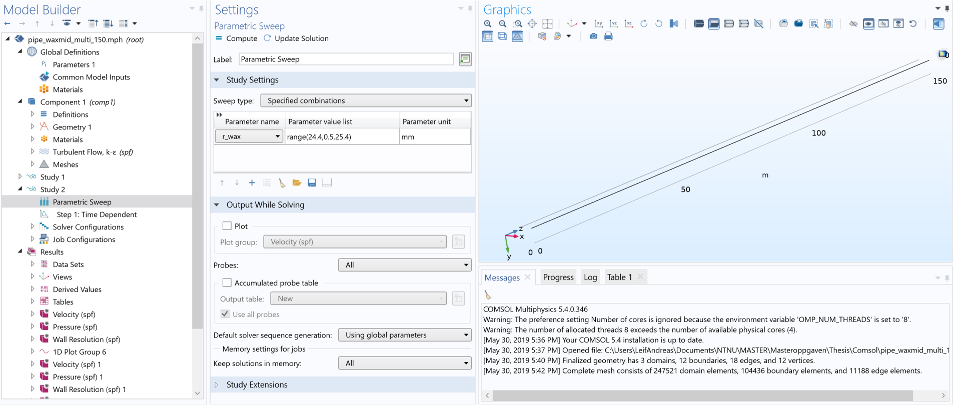
Element size: Extra coarse

```
COMSOL Multiphysics 5.4.0.346
Warning: The preference setting Number of cores is ignored because the environment variable 'OMP_NUM_THREADS' is set to '8'.
Warning: The number of allocated threads 8 exceeds the number of available physical cores (4).
[May 15, 2019 3:55 PM] Your COMSOL 5.4 installation is up to date.
[May 15, 2019 3:55 PM] Opened file: C:\Users\LeifAndreas\Documents\NTNU\MASTER\Masteroppgaven\Thesis\Comsol\pipe_waxmid_multi_
[May 15, 2019 3:55 PM] Finalized geometry has 3 domains, 12 boundaries, 18 edges, and 12 vertices.
[May 15, 2019 3:57 PM] Complete mesh consists of 129147 domain elements, 14926 boundary elements, and 1022 edge elements.
```

Figure 3.2: The meshing options to the model in COMSOL Multiphysics.

3.6 Studies

Studies in COMSOL are the settings before computing the simulation. There are two studies that can be chosen in COMSOL, stationary or time dependent. The correct simulation is time dependent study. Normally, a time dependent study would have used a lot of computation time. Since the flowline was long and with fine mesh with big element sizes made it converge quickly. Furthermore, it was also used parametric sweep, see fig. 3.3. Parametric sweep is a setting in COMSOL that allows you to select different parameter values and simulate them in one computation session. The parametric sweep was set to have a flowline with a deposition thickness r_wax equal 0 mm, 0.5 mm and 1 mm and the time dependent worked simultaneously and calculated pressure at 0 seconds, 30 seconds and 60 seconds, respectively. This gives different pressure and flow values for each parameter setting. This is a simplified solution to test the estimator and how quick the response is.



Label: Parametric Sweep

Sweep type: Specified combinations

Parameter name	Parameter value list	Parameter unit
r_wax	range(24.4,0.5,25.4)	mm

☐ Plot
Plot group: Velocity (spf)
Probes: All
☐ Accumulated probe table
Output table: New
☑ Use all probes
Default solver sequence generation: Using global parameters
Keep solutions in memory: All

```
COMSOL Multiphysics 5.4.0.346
Warning: The preference setting Number of cores is ignored because the environment variable 'OMP_NUM_THREADS' is set to '8'.
Warning: The number of allocated threads 8 exceeds the number of available physical cores (4).
[May 30, 2019 5:36 PM] Your COMSOL 5.4 installation is up to date.
[May 30, 2019 5:37 PM] Opened file: C:\Users\LeifAndreas\Documents\NTNU\MASTER\Masteroppgaven\Thesis\Comsol\pipe_waxmid_multi_1
[May 30, 2019 5:40 PM] Finalized geometry has 3 domains, 12 boundaries, 18 edges, and 12 vertices.
[May 30, 2019 5:42 PM] Complete mesh consists of 247521 domain elements, 104436 boundary elements, and 11188 edge elements.
```

Figure 3.3: The studies options to the model before simulation in COMSOL Multiphysics.

3.7 COMSOL results

Results can be found by using “More Derived Values” in COMSOL Multiphysics. These are options that find the lowest, highest and average values for pressure, temperature and velocity etc. (COMSOL, n.d.). Creating a 1-D plot group and placing a line through the pipeline to find the pressure values and convert them to `.dat`-files. The files can be further postprocessed in MATLAB, such as fig. 3.4, 3.5 and 3.6. Postprocessing in MATLAB creates an opportunity to compare later with a possible estimation.

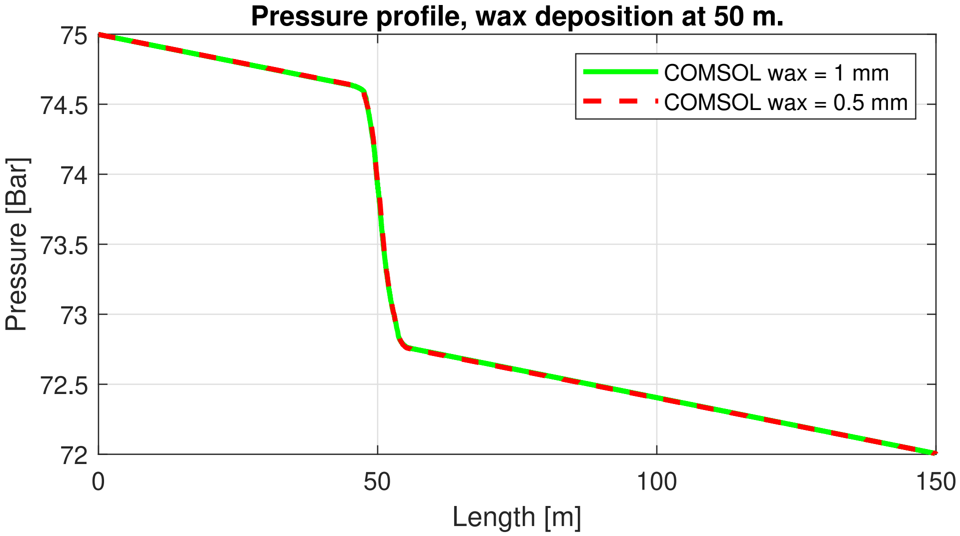


Figure 3.4: COMSOL plot of wax deposition at 50 m.

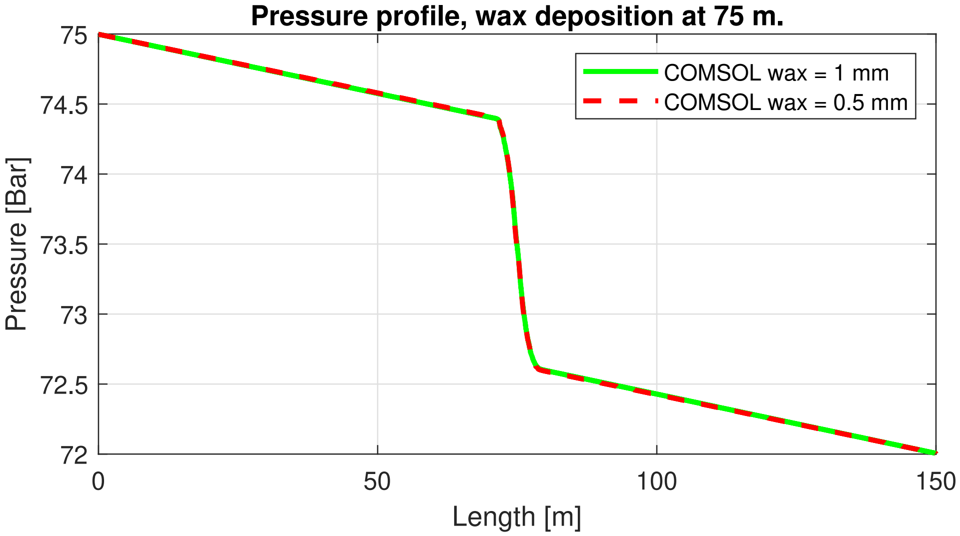


Figure 3.5: COMSOL plot of wax deposition at 75 m.

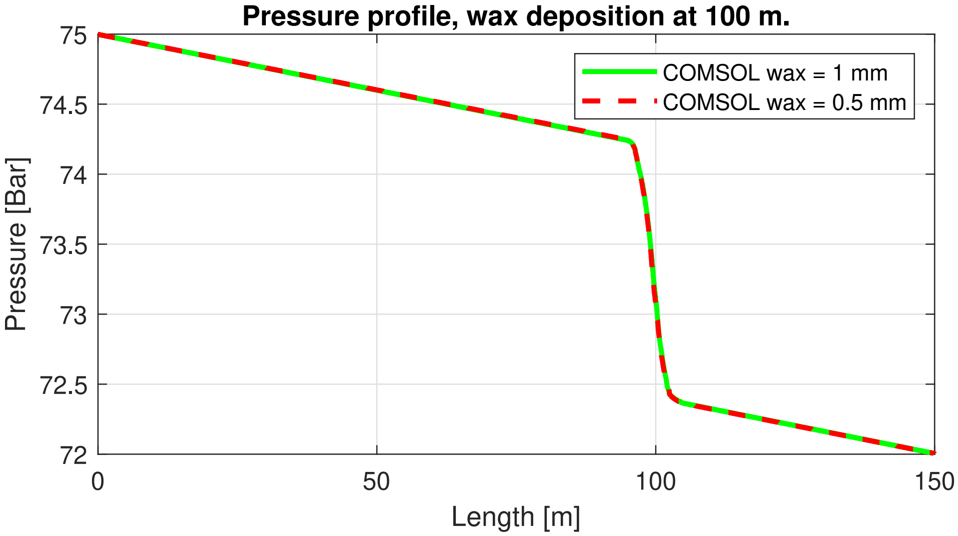


Figure 3.6: COMSOL plot of wax deposition at 100 m.

Chapter 4

Deposition detection

This chapter, we will investigate point one and two of the scope in section 1.3. In section 1.2, it was stated that a previous study by (Oven, 2014) may be used with a modification to estimate pressure loss due to wax and hydrate deposition.

4.1 Hydraulic transmission line

The transmission line equations (4.1) and (4.2) are discretized to pressure states, p, and flow states, q, (Oven, 2014, p. 14). Figure 4.1 illustrates that a pipeline with length L (m), has been divided in to pipe segments or control volumes with the length l. The terms has been simplified and set to $k_1 = \frac{\beta}{Al}$ and $k_2 = \frac{A}{l\rho}$. Where A is cross-section area of the pipeline, β is the bulk modulus for water (Pa) and ρ is the average density of water (kg/m^3).

$$\begin{aligned} \dot{p_1} &= k_1(q_{in} - q_1) \\ &\vdots \\ \dot{p_N} &= k_1(q_{N-1} - q_N) \end{aligned} \tag{4.1}$$

$$\begin{aligned} \dot{q_1} &= k_2(p_1 - p_2) - fq_1 \\ &\vdots \\ \dot{q_N} &= k_2(p_N - p_{out}) - fq_N \end{aligned} \tag{4.2}$$

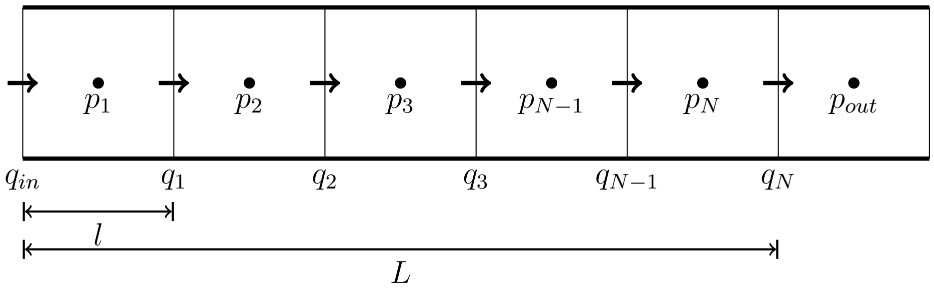


Figure 4.1: Control volumes for (Oven, 2014) work on his master thesis.

Furthermore, the continuous-time (CT) linear time-invariant (LTI) state-space equations are given by (Chen, 2013, p. 105). Equation (4.1) and eq. (4.2) can be rewritten to state-space equations such as eq. (4.4a). This gives the state matrix $\mathbf{A}$ and input-to-state matrix $\mathbf{B}$, respectively.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{4.3a}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{4.3b}$$

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{p_1} \\ \dot{p_2} \\ \vdots \\ \dot{p_N} \\ \dot{q_1} \\ \dot{q_2} \\ \vdots \\ \dot{q_N} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dots & 0 & -k_1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & k_1 & -k_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & -k_1 \\ k_2 & -k_2 & \dots & 0 & -f & 0 & \dots & 0 \\ 0 & k_2 & \dots & 0 & 0 & -f & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & k_2 & 0 & 0 & \dots & -f \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \\ q_1 \\ q_2 \\ \vdots \\ q_N \end{bmatrix} \tag{4.4a}$$

$$+ \begin{bmatrix} k_1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & -k_2 \end{bmatrix} \begin{bmatrix} q_{in} \\ p_{out} \end{bmatrix} \tag{4.4b}$$

The last node of flow, and the last node of pressure are used as measurements of the pipeline, which can be described as

$$\mathbf{y} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ p_N \\ 0 \\ 0 \\ \vdots \\ q_N \end{bmatrix} \tag{4.5}$$

The measurements are expanded to give state-to-output matrix $\mathbf{C}$. The feedthrough matrix $\mathbf{D}$ is zero.

$$\mathbf{y} = \mathbf{C}\mathbf{x} = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \\ q_1 \\ q_2 \\ \vdots \\ q_N \end{bmatrix} \tag{4.6}$$

4.2 Modified hydraulic transmission line

For this thesis a modified setup is necessary get the pressure measurement. Figure 4.2 shows how the new discretized model with different initial values.

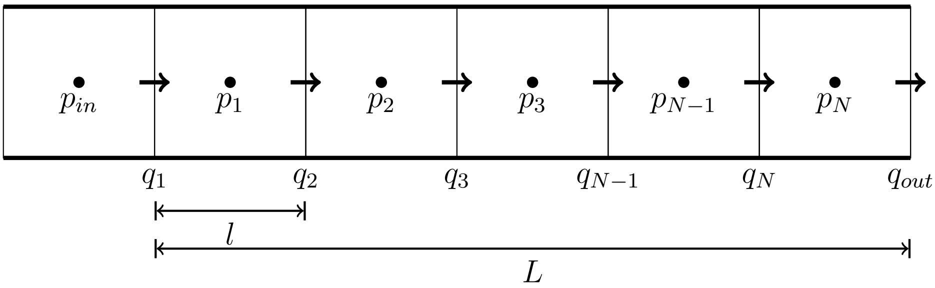


Figure 4.2: The control volumes for this thesis.

Using fig. 4.2 to write the model in to discretized equations (4.7) and (4.8). Where $k_1 = \frac{\beta}{Al}$ and $k_2 = \frac{A}{l\rho}$ and A is cross-section area of the pipeline, β is the bulk modulus for water (Pa) and ρ is the average density of crude oil (kg/m^3). The friction factor d is very important for the flow equations. The value will prevent that the system becomes unstable or marginal stable, more on that in section 4.3.

$$\begin{aligned} \dot{p_1} &= k_1(q_1 - q_2) \\ &\vdots \\ \dot{p_N} &= k_1(q_N - q_{out}) \end{aligned} \tag{4.7}$$

$$\begin{aligned} \dot{q_1} &= k_2(p_{in} - p_1) - dq_1 \\ &\vdots \\ \dot{q_N} &= k_2(p_{N-1} - p_N) - dq_N \end{aligned} \tag{4.8}$$

Moreover, the measurements for those equations are the first and last pressure state in eq. (4.9). The state-space equations for this case can be rewritten as in equations (4.10) and (4.11). This are the state matrix $\mathbf{A}$, input-to-state matrix $\mathbf{B}$ and state-to-output matrix $\mathbf{C}$, respectively.

$$\mathbf{y} = \begin{bmatrix} p_1 \\ 0 \\ \vdots \\ p_N \end{bmatrix} \tag{4.9}$$

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{p_1} \\ \dot{p_2} \\ \vdots \\ \dot{p_N} \\ \dot{q_1} \\ \dot{q_2} \\ \vdots \\ \dot{q_N} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dots & 0 & k_1 & -k_1 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & k_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & k_1 \\ -k_2 & 0 & \dots & 0 & -d & 0 & \dots & 0 \\ k_2 & -k_2 & \dots & 0 & 0 & -d & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -k_2 & 0 & 0 & \dots & -d \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \\ q_1 \\ q_2 \\ \vdots \\ q_N \end{bmatrix} \tag{4.10a}$$

$$+ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & -k_1 \\ k_2 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_{in} \\ q_{out} \end{bmatrix} \tag{4.10b}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \\ q_1 \\ q_2 \\ \vdots \\ q_N \end{bmatrix} \tag{4.11}$$

4.3 Stability

Stability for the system is important for getting a solution. The system was oscillating and, in the beginning, could not use an automatic solver in MATLAB/SIMULINK. The Rosenbrock method had to be used to make the solution converge. Rosenbrock can be used for stiff systems without Newton iterations in the stage computations (Egeland and Gravdahl, 2002, p. 575). There were two areas that was observed when the system was constructed. What happens if the friction was removed and how does the system behave with a large numerical difference between k_1 and k_2?

4.3.1 Friction

A further simplification of the system which includes removing the friction factor d will cause in marginal or unstable system. It is essential to make the system stable. Consider the state-space equations (4.3) with and Laplace transformation of system to make it a transfer function. Equation (4.12) shows the approach to use the state-space equations to a transfer function (Egeland and Gravdahl, 2002, p. 11).

$$\mathcal{L}\{\dot{\mathbf{x}}\} \Rightarrow s\mathbf{x}(s) = \mathbf{A}\mathbf{x}(s) + \mathbf{B}\mathbf{u}(s) \tag{4.12a}$$

$$\mathbf{x}(s) = (s\mathbb{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{u}s \tag{4.12b}$$

$$\mathcal{L}\{\mathbf{y}\} \Rightarrow \mathbf{y}(s) = \mathbf{C}\mathbf{x}(s) + \mathbf{D}\mathbf{u}(s) \tag{4.12c}$$

$$\mathbf{y}(s) = \mathbf{C}(s\mathbb{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{u}(s) + \mathbf{D}\mathbf{u}(s) \tag{4.12d}$$

$$\frac{\mathbf{y}}{\mathbf{u}}(s) = \mathbf{H}(s) = \mathbf{C}(s\mathbb{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \tag{4.12e}$$

Furthermore, consider one control volume with inlet pressure and outlet flow fig. 4.3. Equations (4.13) and (4.14) shows the systems with and without the friction factor d. The transfer function to eq. (4.13) shows that the system is undamped. This means that system will oscillate and be difficult to estimate. Equation (4.14) has a damping ratio. Tuning the parameter correct will give a stable system with real negative eigenvalues.

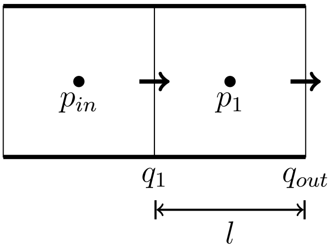


Figure 4.3: One control volume.

$$\dot{p_1} = k_1(q_1 - q_{out}) \tag{4.13a}$$

$$\dot{q_1} = k_2(p_{in} - q_1) \tag{4.13b}$$

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & k_1 \\ -k_2 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ q_1 \end{bmatrix} + \begin{bmatrix} 0 & -k_1 \\ k_2 & 0 \end{bmatrix} \begin{bmatrix} p_{in} \\ q_{out} \end{bmatrix} \tag{4.13c}$$

$$\mathbf{y} = \begin{bmatrix} p_1 \\ q_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ q_1 \end{bmatrix} \tag{4.13d}$$

$$\mathbf{H}(s) = \begin{bmatrix} \frac{k_1\,k_2}{s^2+k_1\,k_2} & -\frac{k_1\,s}{s^2+k_1\,k_2} \\ \frac{k_2\,s}{s^2+k_1\,k_2} & \frac{k_1\,k_2}{s^2+k_1\,k_2} \end{bmatrix} \tag{4.13e}$$

$$\dot{p_1} = k_1(q_1 - q_{out}) \tag{4.14a}$$

$$\dot{q_1} = k_2(p_{in} - p_1) - dq_1 \tag{4.14b}$$

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & k_1 \\ -k_2 & -d \end{bmatrix} \begin{bmatrix} p_1 \\ q_1 \end{bmatrix} + \begin{bmatrix} 0 & -k_1 \\ k_2 & 0 \end{bmatrix} \begin{bmatrix} p_{in} \\ q_{out} \end{bmatrix} \tag{4.14c}$$

$$\mathbf{H}(s) = \begin{bmatrix} \frac{k_1\,k_2}{s^2+d\,s+k_1\,k_2} & -\frac{k_1\,(d+s)}{s^2+d\,s+k_1\,k_2} \\ \frac{k_2\,s}{s^2+d\,s+k_1\,k_2} & \frac{k_1\,k_2}{s^2+d\,s+k_1\,k_2} \end{bmatrix} \tag{4.14d}$$

Selecting friction factor d can be done by solving the second order oscillatory system (Balchen, Andresen and Foss, 2016, p. 144). Equation (4.15c) shows that the damping can be tuned as wanted, if $\zeta = 1$ means that is critically damped.

$$H(s) = \frac{k_1\,k_2}{s^2 + d\,s + k_1\,k_2} \tag{4.15a}$$

$$H(s) = \frac{k_1\,k_2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{4.15b}$$

$$\zeta = \frac{d}{2\sqrt{k_1\,k_2}} \tag{4.15c}$$

4.3.2 Stiff system and scaling

Some systems have a large spread in eigenvalues are referred as stiff systems. Stiff systems give problems with simulation time and accuracy (Egeland and Gravdahl, 2002, p. 535). This system has a large spread in eigenvalues. Implicit Runge-Kutta methods can be used, but it is easier to scale the matrix. The discretized pressure and flow equations are calculated in SI-units. Converting the pressure values from Pa to Bar will give a less stiff system. Equations (4.16) and (4.17) are showing that k_1 and k_2 can be divided and multiplied by 10^5, respectively. Furthermore, stiff systems gives also accuracy problems when implemented to the Kalman filter (Kulikov and Kulikova, 2018).

$$\dot{p}_{1Pa} = k_1(q_1 - q_{out}) \tag{4.16a}$$

$$\dot{p}_{1Bar}10^5 = k_1(q_1 - q_{out}) \tag{4.16b}$$

$$\dot{p}_{1Bar} = \frac{k_1}{10^5}(q_1 - q_{out}) \tag{4.16c}$$

$$\dot{q_1} = k_2(p_{inPa} - p_{1Pa}) - dq_1 \tag{4.17a}$$

$$\dot{q_1} = k_2(10^5 p_{inBar} - 10^5 p_{1Bar}) - dq_1 \tag{4.17b}$$

$$\dot{q_1} = 10^5 k_2(p_{inBar} - p_{1Bar}) - dq_1 \tag{4.17c}$$

4.4 Wax deposition and MATLAB algorithm

When wax deposition builds up in the pipeline it narrows the cross-section area. This must be accounted for. Figure 4.4 shows a smaller cross-section area for one control volume. The values k_1 and k_2 has to be changed iteratively for each control volume. The observability must be calculated for no wax and every control volume with 0.5 mm and 1 mm wax thickness.

Furthermore, since the dimension of the observable matrixes are unknown, an algorithm in MATLAB is created. The algorithm 4.1 will change in size and still have the same structure as derived above. This allows us to experiment with different lengths of the pipeline and number of states to know what is observable. The entire code can be found in appendix C.

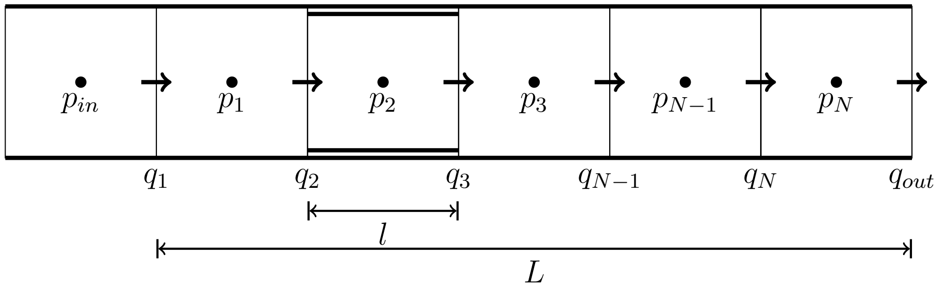


Figure 4.4: Wax deposition in control volume 2.

Algorithm 4.1: MATLAB state-space equation generator – no wax

```
%% Observability
clear all
close all

%% Impliment values
scale = 1e5; % Scaling to prevent stiff system

beta = 1.05e9/scale; % Bulk modulus for Benzene [
    Bar]
a = pi*0.0254^2; % Cross-section of the pipeline
    [m^2]
density = 900; % Density of fluid [kg/m^3]
l = 14; % Length of pipesegment [m]

k1 = beta/(a*l*scale); % Simplifcation to the
    code with scaling
k2 = (a*scale)/(l*density);
```

```
d = 1; % Friction factor for pipe roughness coefficient

n = 22; % Number of states, half pressure, half flow

An = zeros(n,n); % Making an A mastrix with n by n dimension of zeros
p = n/2+1;
 for m = 1:n/2-1 % Adding k1 and -k2*d to the state-space model
     An(m,p) = k1;
     An(m,p+1) = -k1;
     An(p,p) = -d;
     p = p+1;
 end
An(n/2,n) = k1; % Adding k1 to qN
An(n/2+1,1) = -k2;     % Adding -k2 to p1
An(n,n) = -d;
p = 1;
 for m = n/2+2:n % Adding k2 from 2 to n-1
     An(m,p) = k2;
     An(m,p+1) = -k2;
     p = p+1;
 end

Cn = zeros(n,n);  % Making a C matrix for
```

```
   pressure
Cn(1,1) = 1;
Cn(n/2,n/2) = 1; % Adding measurement points

Bn = zeros(n,2);% Creating the B matrix
Bn(n/2,2) = -k1;
Bn(n/2+1,1) = k2;
```

4.5 Observability

Considering an n-dimensional p-input and q-output state-space equation on the form $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$ and $\mathbf{y} = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$ with the dimensions of $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ being $n \times n$, $n \times p$, $q \times n$ and $q \times p$, respectively (Chen, 2013, p. 194). The system described by these equations is said to be observable if the $nq \times n$ dimension observability matrix defined in eq. (4.18) has full column rank; $rank(\mathcal{O}) = n$ (Chen, 2013, p. 197).

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{4.18}$$

An addition to algorithm 4.1 in MATLAB the observability matrix can be generated by `obsv` (Chen, 2013, p. 197). Furthermore, using `rank` gives the rank of the observability matrix. If $rank(\mathcal{O}) = n$ it is concluded that the system is observable.

Algorithm 4.2: Observability matrix and rank calculation

```
obser_estimation = obsv(A_estimation, C_estimation); % Caluclating the Observability matrix
rank_estimation = rank(obser_estimation); % The rank of Observability matrix
```

Chapter 5

The Kalman filter

The Kalman filter has been used since the 1960s (Brown and Hwang, 2012, p. 141). The filter receives random signals such as disturbance from measurements and estimates the states in the system. Implementing a Kalman filter will give us an estimate of pressure profiles of where the wax deposition is located.

5.1 Discrete Kalman filter

The most commonly version of the Kalman filter is the discrete Kalman filter. Equation (5.1) shows how the discrete Kalman filter is working iteratively to estimate the states.

$$
\begin{aligned}
\mathbf{L}_k &= \mathbf{P}_k^- \mathbf{C}_k^\top (\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^\top + \mathbf{R}_k)^{-1} && \text{(5.1a)}\\
\hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{L}_k(\mathbf{y} - \mathbf{C}_k \hat{\mathbf{x}}_k^-) && \text{(5.1b)}\\
\mathbf{P}_k &= (\mathbb{I} - \mathbf{L}_k \mathbf{C}_k)\mathbf{P}_k^- (\mathbb{I} - \mathbf{L}_k \mathbf{C}_k)^\top + \mathbf{L}_k \mathbf{R}_k \mathbf{L}_k^\top && \text{(5.1c)}\\
\hat{\mathbf{x}}_{k+1} &= \mathbf{A}_k \hat{\mathbf{x}}_k && \text{(5.1d)}\\
\mathbf{P}_{k+1}^- &= \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^\top + \mathbf{Q}_k && \text{(5.1e)}
\end{aligned}
$$

Where $\mathbf{L}_k$ is the Kalman gain, $\mathbf{P}_k$ is the error covariance, $\mathbf{R}_k$ is the measurement noise covariance and $\mathbf{Q}_k$ is the process noise covariance (Brown and Hwang, 2012, p. 147/165). $\mathbf{A}_k$, $\mathbf{B}_k$ and $\mathbf{C}_k$ are discretized matrixes of the state-space equations. The terms with $^-$ are the previous values that are updated after each iteration.

5.2 Implementing the discrete Kalman filter

Implementing the discrete Kalman filter can be created in SIMULINK. Figure 5.1 show how the model and the Kalman filter was created. The function block has the Kalman filter implemented with the same algorithm as in eq. (5.1). The discrete Kalman function is described in algorithm C.2.

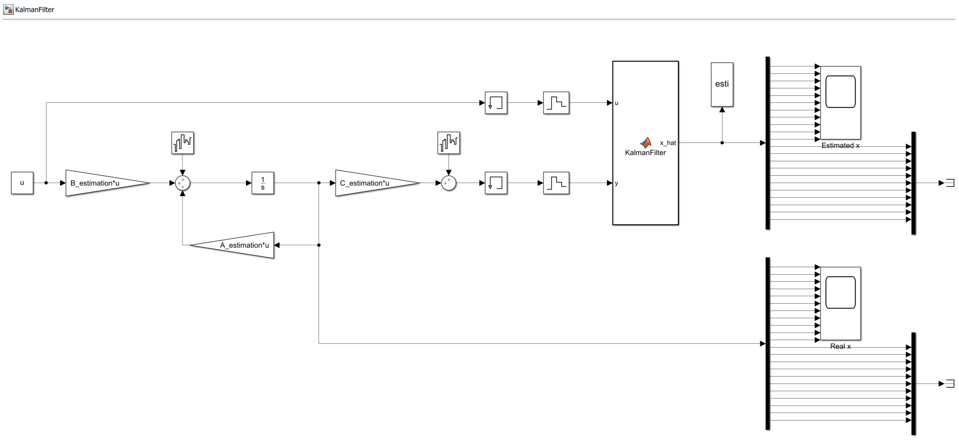


Figure 5.1: SIMULINK set up of the Kalman filter

Chapter 6

Results

In this chapter we will look at the results of the questions in the scope, the postprocessing COMSOL Multiphysics in MATLAB and implementing the Kalman filter.

6.1 Stability and Observability matrix

The first item from section 1.3 was to create the state-space equation for the system. This is derived in chapter 4. Selecting number of nodes to `n` = 22, `l` = 14 and `d` = 1 shows that all the systems with and without wax deposition are stable and observable. The scaling of the system improved the stability and made it more observable. Changing an integer number with ± 1 can cause some unobservable control volumes. If the system is unobservable, it often occurs in the middle of the pipeline with wax deposition, far from the measurements. Furthermore, fig. 6.1 illustrates how the control volumes are divided.

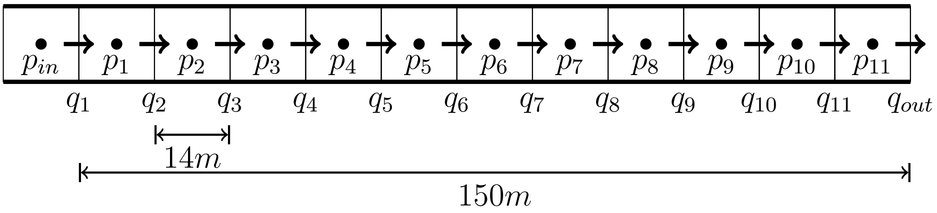


Figure 6.1: The model where every state is stable and observable.

6.2 Wax deposition at 50 m

The results in fig. 6.2 and 6.3 show that for a 1 mm wax deposition at 50 meters, the control volume converge after 5000 seconds and at 10000 seconds the estimation is below simulated value by 10 – 20 kPa. For 0.5 mm wax deposition it is 20 – 30 kPa.

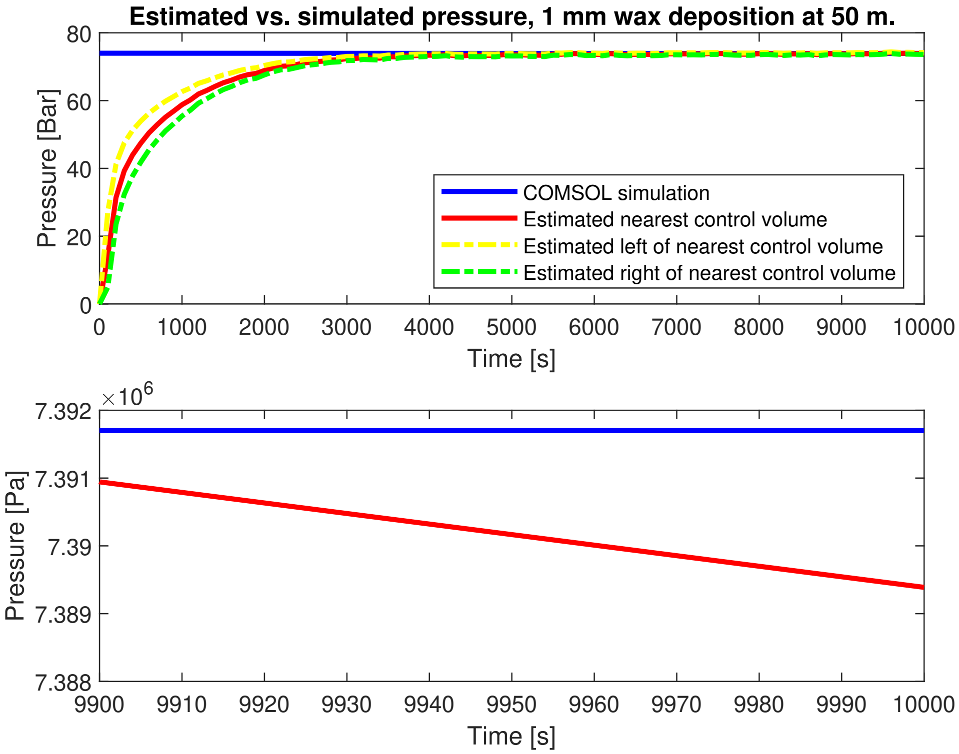


Figure 6.2: 1 mm of wax deposition at 50 m and estimated plot.

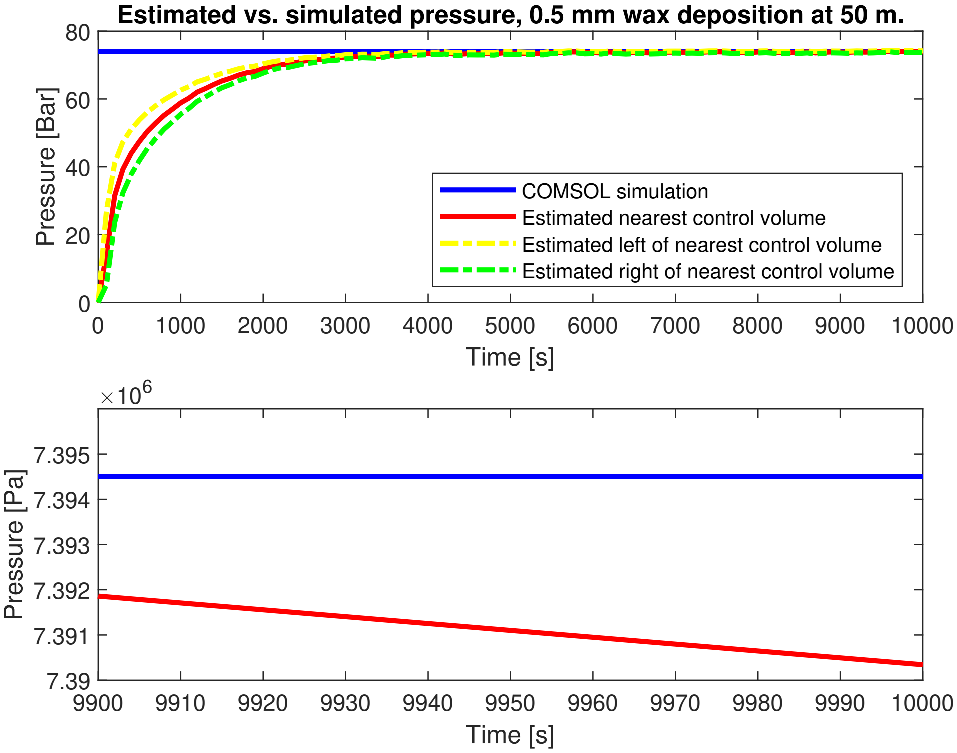


Figure 6.3: 0.5 mm of wax deposition at 50 m and estimated plot.

The pressure profiles in fig. 6.4 show that 1 mm wax deposition is closer to the simulated COMSOL model than an empty pipe. Wax deposition with 0.5 mm thickness is similar to an empty pipe.

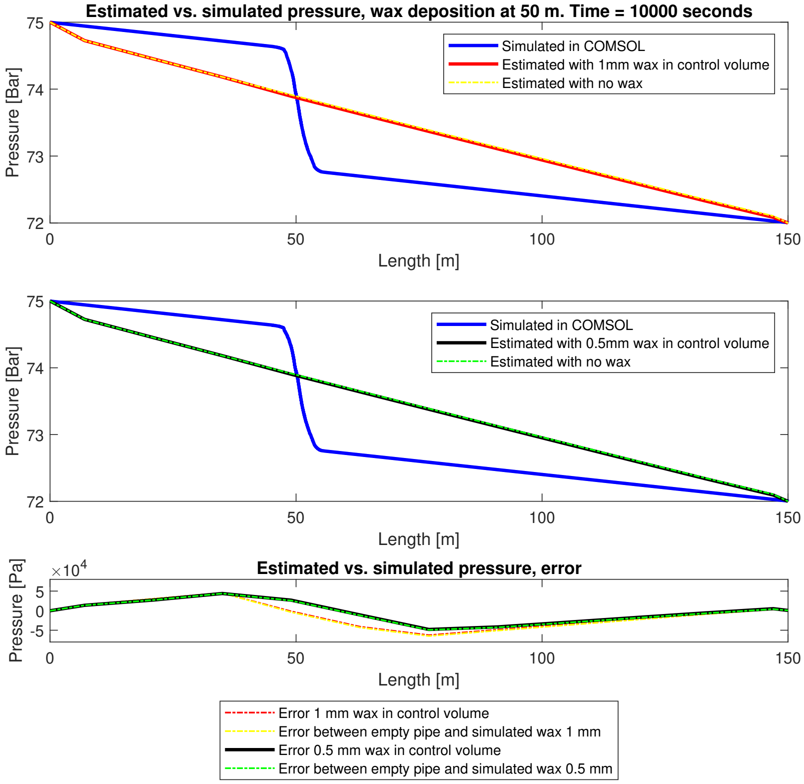


Figure 6.4: Comparing and error at 50 m and estimated plot.

6.3 Wax deposition at 75 m

As in section 6.2 the estimation is similar, with the same deviation.

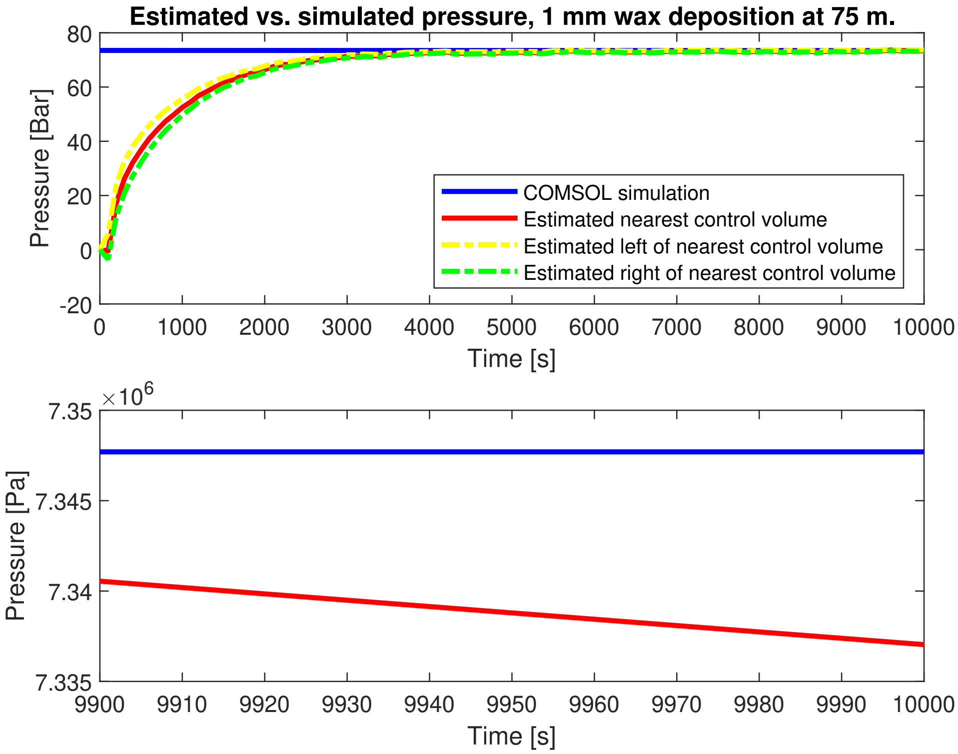


Figure 6.5: 1 mm of wax deposition at 75 m and estimated plot.

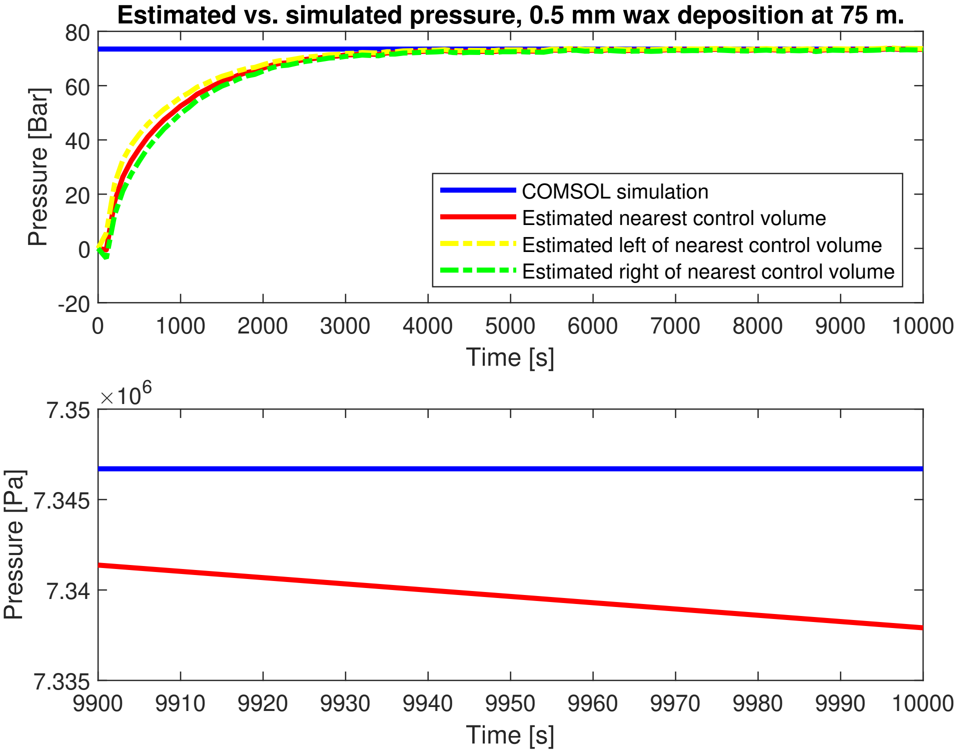


Figure 6.6: 0.5 mm of wax deposition at 75 m and estimated plot.

The pressure profile shows that the wax depositions are the best estimated at 75 meters.

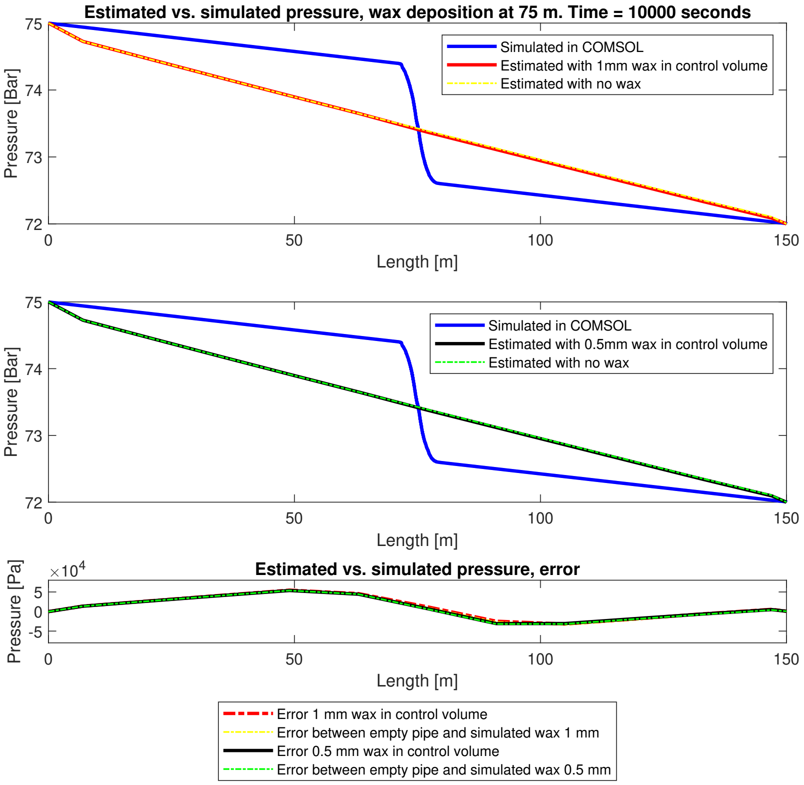


Figure 6.7: Comparing and error at 75 m and estimated plot.

6.4 Wax deposition at 100 m

As in section 6.2 and 6.3 the estimation is similar, with the same deviation.

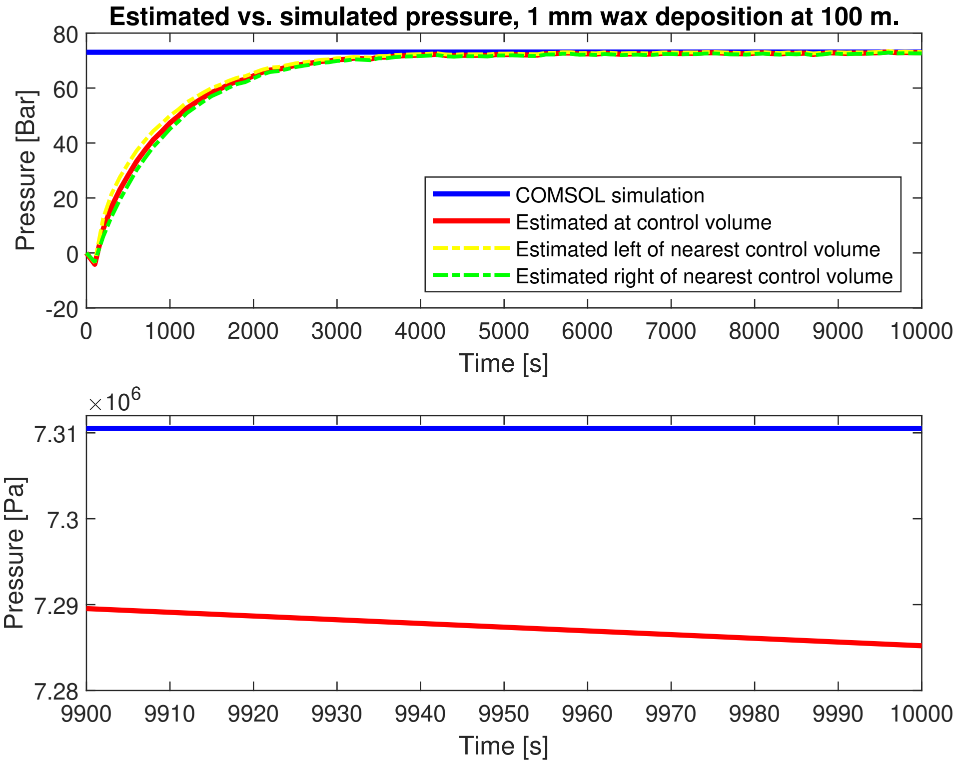


Figure 6.8: 1 mm of wax deposition at 100 m and estimated plot.

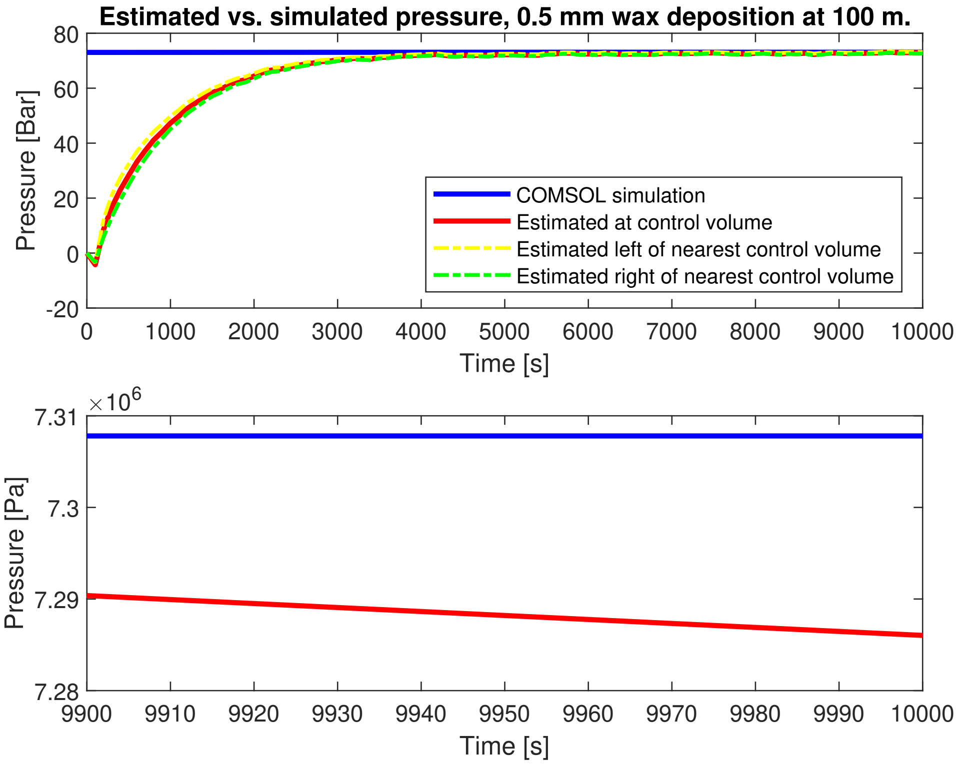


Figure 6.9: 0.5 mm of wax deposition at 100 m and estimated plot.

A 1 mm wax deposition has the biggest error. The 0.5 mm wax deposition estimation and have the best estimation at 100 meters.

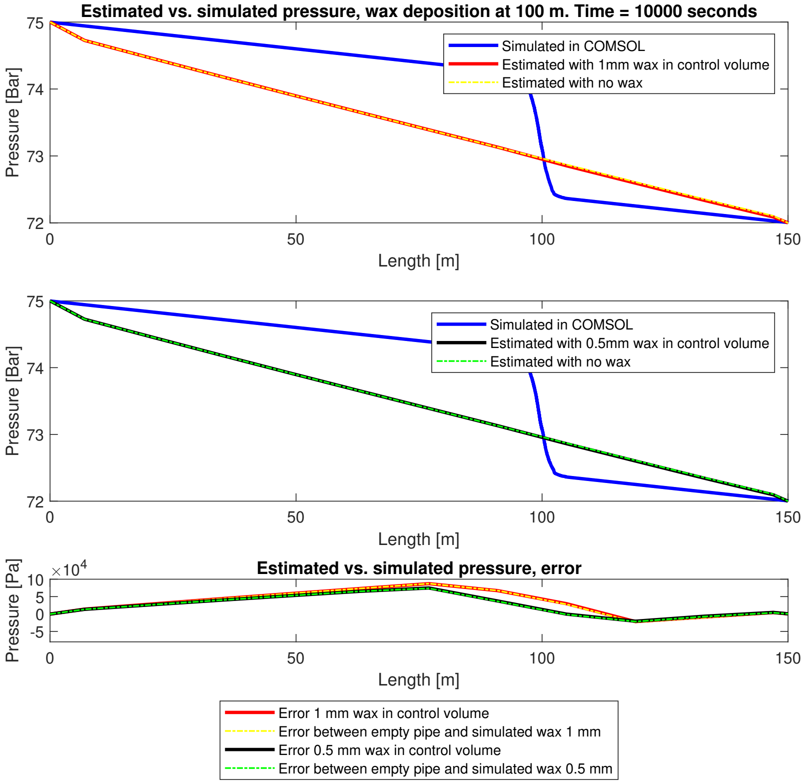


Figure 6.10: Comparing and error at 100 m and estimated plot.

Comparing the estimations show that the control volumes that will become unobservable if number of nodes are changed will give the best estimates.

Chapter 7

Discussion and Future work

This chapter will merge the discussion throughout the thesis. Errors and the choices made will be presented from chapter 3 to chapter 6. In addition, with recommendations for future work to EMPIG and others who are going to work with this topic later.

7.1 COMSOL Multiphysics

COMSOL Multiphysics was learned during the spring of 2019. It is a complex software with many options.

7.1.1 Geometry

When creating geometry in COMSOL was not designed for creation of wax deposition. The geometry was divided into three parts and merged together to make a flowline with wax deposition. Geometry could also be constructed in computed assisted design (CAD) software such as Autodesk Inventor or SolidWorks. COMSOL does not understand that a pipeline created around a flowline are two different components, this is most likely an error from the user.

7.1.2 Selection of constants

Selecting the initial values and constants were most a guessed value by the author himself and some values such as inlet pressure and flow where given by EMPIG. Selection of material was discussed section 3.3. Therefore, will this give an error. It is recommended that full specification of the crude oil is implemented into COMSOL or another modeling and simulation software.

7.1.3 Meshing

The flowline was 150 meters long. The length of the flowline caused big elements and therefore making the computation time shorter. When wax deposition (r_wax = 24.9 mm) was added gave COMSOL problems because it needed smaller elements where the cross-section area is 0.5 mm smaller than the rest of the flowline. This had to be custom made. COMSOL approved the specification, but COMSOL changed the cross-section to square shaped instead of circular. To hopefully make the meshing and geometry correct, it is recommended to use a computer with more memory for such computations.

7.1.4 Simulation

The time dependent study might have been to long to set for 10000 seconds with a 5000 seconds step interval. EMPIG wanted to optimize detection and removal of wax deposition (Lund, 2018, personal communication, 15. August). A comparison with real Cold Flow system would have given an accurate description of wax deposition.

It was intentional to have a smaller flowline in the begin of this thesis, the elements were smaller and the computation time with parametric sweep took several hours. After changing the length and creating larger elements the simulation was shortened to under 90 minutes for each geometry. The big element sizes are most likely to shorten the computation time. Again, a more efficient computer to handle this type of problems would have given a better result.

7.2 State-space equations

There are several possible errors in the state-space equations that has influenced the results.

7.2.1 Model and selection of constants

The state-space was based on the work by (Oven, 2014). Implementing the model was added directly from that thesis. Selecting the bulk modulus, density was taken from (The Engineering ToolBox, n.d.[a]) and (The Engineering ToolBox, n.d.[b]), respectively. The friction factor was selected to get a stable system. For future work the material must be specified and getting a more detailed technical data on friction for an accurate solution.

7.2.2 Scaling and Stiff system

In this thesis, there were only scaling of the pressure with 10^5. There were other parameters that could be scaled to soften the system such as flow. To convert flow from m^3/s to l/min would give a ratio of $6 \cdot 10^4$. Changing the scaling in flow or pressure or use both combined to see what happened to the system is a topic for future work.

7.2.3 Observability

When the `l` must be a bit smaller than `n` showed that the system was observable. Selecting more nodes will demand an increase of the pipe segment length. This is not fortunate because of the user options to adapt to their needs. To optimize the pipe segment and number nodes are recommended for next steps for this problem.

7.3 Kalman filter and simulation

The Kalman filter worked to estimate the pressure loss at the specific positions, but it could not estimate the entire pressure profile.

7.3.1 Accuracy

Using the Kalman filter is working for where the thickness of wax deposition is 1mm. This might have something to do with scaling and using Bar as measurement. The plot had a small difference to kPa and in this case is almost nothing. Adapting the filter such that a result with Pa is the measurement can give a more accurate solution.

7.3.2 Time step discrete state-space equations and Covariance

The sampling time was set to 0.1 second and this is a selected value from the author himself. This will have impact on the continuous state-space equations converted to discrete system. Further investigation about how this could impact the result is needed.

The covariance values process, measurement and error covariance were selected. The error covariance was updated in the Kalman filter and changed after each iteration. The process and measurement noise covariance remained as constants. These values will affect the estimation and it is recommended to change these values to get an amount of noise that is more realistic to the Cold Flow system instrumentation.

7.3.3 Simulation length

The simulation time length that was done in SIMULINK was a direct consequence of the simulation of the COMSOL models. The Kalman filter estimated the states after simulated 5000 seconds. For future work it is recommended to investigate if this is necessary.

7.4 Results

The results showed that the estimation was close to the wax deposition with small oscillations of 0.1-0.3 Bar. It is not certain if the value is accurate enough because every single estimation was not compared with the simulation. This must be done before EMPIG can use this further.

Pressure profile shows that the pressures have the similar error until the wax deposition appears. This show that modeled wax deposition can be estimated and using the smallest error to pinpoint location and size of the estimation. Future work to optimize this solution is to adapt to recommendations in the previous sections of this chapter.

Chapter 8

Summary and Conclusion

The purpose of this thesis is estimate wax and hydrate deposition to the Cold Flow system for EMPIG AS. In the prestudies it was documented that existing technology and the methods for wax and hydrate deposition was not compatible with EMPIG's system.

COMSOL Multiphysics was used to create a 150 m long flowline to simulate wax depositions at 50 m, 75 m and 100 m and varying deposition of 0 mm, 0.5 mm and 1 mm. The simulations were time dependent with 0, 5000 and 10000 seconds.

Creating the state-space equations based on works by (Oven, 2014) was used and modified to serve the need to estimate wax and hydrate deposition. The equations had to be scaled to prevent a stiff system. Argumentations for the friction factor was derived and showed that without the factor the system will become marginal stable. The MATLAB script including the state-space equation was coded with the observability matrix.

The discrete Kalman filter was derived in chapter 5. Showing how the discrete state-space equations was used to estimate wax and hydrate deposition. The implementation was created in SIMULINK.

The results showed that the Kalman filter is able to estimate the point where the wax deposition is. It has an error of 10-30 kPa. The pressure profile for the whole pipeline is not equal. The middle of the pipeline estimates with their respective control volume and deposition were closest to the simulated value.

There are several possible errors in this thesis that has to be accounted before EMPIG can use the estimation. Implementing the correct fluid properties and values such as friction factor and noise covariance are among the topics to make this work for the Cold flow system.

8.1 Conclusion

It is concluded as in the problem formulation that an algorithm to estimate location of the Cold flow cooling system is found. It is also possible to estimate the thickness of the wax deposition. The best estimations are in the middle of the flowline. Furthermore, the questions in section 1.3 has been answered and fulfilled, but before using the algorithm EMPIG or others have to add the constants that is more accurate with the existing Cold flow cooler.

Bibliography

Aamo, Ole Morten (2016). 'Leak Detection, Size Estimation and Localization in Pipe Flows'. eng. In: *Automatic Control, IEEE Transactions on* 61.1, pp. 246–251. ISSN: 0018-9286.

Balchen, Jens G., Trond Andresen and Bjarne A. Foss (2016). *Reguleringsteknikk*. nor. 6. utg. Trondheim: Institutt for teknisk kybernetikk. ISBN: 978-82-7842-202-1.

Biltema (2019). *Sikkerhetsdatablad Kombiolje 10W–30*. Available: `https://www.biltema.no/BiltemaDocuments/SecuritySheets/nb/36994_SDB.pdf`, (Acsessed: 08/05/2019).

Brekke, Magne (2018). *Akustisk Impedans*. Available: `https://sml.snl.no/akustisk_impedans`, (Acsessed: 18/11/2018).

Brown, Robert Grover and Patrick Y.C. Hwang (2012). *Introduction to random signals and applied Kalman filtering: with MATLAB exercises*. eng. 4th ed. Hoboken, N.J: Wiley. ISBN: 978-0-470-60969-9.

Chen, Chi-Tsong (2013). *Linear system theory and design*. eng. International 4th ed. The Oxford series in electrical and computer engineering. New York: Oxford University Press. ISBN: 9780199964543.

COMSOL (n.d.). *Postprocessing the Results from a Parametric Sweep*. Available: `https://www.comsol.com/video/postprocessing-results-parametric-sweep`, (Acsessed: 15/05/2019).

Egeland, Olav and Jan Tommy Gravdahl (2002). *Modeling and simulation for control*. eng. Report (Norges teknisk-naturvitenskapelige universitet. Institutt for teknisk kybernetikk: trykt utg.) Trondheim: Department of Engineering Cybernetics, Norwegian University of Science and Technology.

Empig (2019a). *Empig*. Available: `https://empig.no/`, (Acsessed: 06/05/2019).

– (2019b). *Empig Technology*. Available: `https://empig.no/technology/`, (Acsessed: 06/05/2019).

Hirsti, Leif Andreas (2018). *Introduction to Cold Flow – an understanding of wax and hydrate deposition in cooling systems*. Norwegian University of Science and Technology – Department of Engineering Cybernetics.

Kulikov, Gennady Yu. and Maria V. Kulikova (2018). *Accuracy Issues in Kalman Filtering State Estimation of Stiff Continuous-Discrete Stochastic Models Arisen in Engineering Research*. Available: `https://ieeexplore.ieee.org/document/8540695`, (Acsessed: 24/05/2019).

Larsen, Roar (2008). *SATURN Cold Flow – transport av brønnstrøm uten kjemikalier og oppvarming*. Available: https://www.sintef.no/globalassets/upload/konsern/media/sintef-seminar-foredrag/foredrag-roar-larsen-25-sept-2008.ppt, (Acsessed: 06/11/2018).

Lund, Fredrik (2018). CTO, Senior Mechanical – Empig AS.

Lundberg, Nils H. (2018). *Råolje*. Available: https://snl.no/r%C3%A5olje, (Acsessed: 15/12/2018).

Lyu, Peter (2016). *Simulating Turbulent Flow in COMSOL Multiphysics®*. Available: https://www.comsol.com/video/simulating-turbulent-flow-in-comsol-multiphysics, (Acsessed: 08/05/2019).

Oven, Sindre (2014). *Leak Detection in Pipelines by the use of State and Parameter Estimation*. nor. Norwegian University of Science and Technology – Department of Engineering Cybernetics. Available: http://hdl.handle.net/11250/261148.

Rocsole (2018). *Deposition Watch*. Available: https://www.rocsole.com/products/applications/deposition-watch, (Acsessed: 07/11/2018).

SINTEF (2010). *Cold Flow*. Available: https://www.sintef.no/en/projects/saturn-cold-flow/, (Acsessed: 31/10/2018).

Statoil Petroleum AS and World Intellectual Property Organization (Nov. 2014). ‘Estimating a Thickness of a Deposited Material on a Surface’. WO 2014/177210 A1.

The Engineering ToolBox (n.d.[a]). *Bulk Modulus and Fluid Elasticity*. Available: `https://www.engineeringtoolbox.com/bulk-modulus-elasticity-d_585.html`, (Acsessed: 03/06/2019).

– (n.d.[b]). *Liquid Densities*. Available: `https://www.engineeringtoolbox.com/liquids-densities-d_743.html`, (Acsessed: 03/06/2019).

Appendix

Appendix A

Nomenclature

Nomenclature

β	Bulk modulus
ρ	Density material, $\mathrm{kg/m^3}$
A	Cross-section area, $\mathrm{m^2}$
c	Speed of sound
d	Friction factor
f	Friction factor
g	Gravitational acceleration, $9,81\mathrm{m/s^2}$
$H(s)$	Transfer function
k	Constant
l	Length of pipesegment, m
L	Length, m
$\mathcal{L}$	Laplace transformation
n	Number of nodes
$\mathcal{O}$	Observability
p	Pressure, Pa
q	Flow, $\mathrm{m^3/s}$
r_wax	Thickness wax, m
Z	Acoustic impedance, $\mathrm{kg/m^2s}$

Appendix B

Acronym

Acronym

CAD	Computed Assisted Design
CF	Cold Flow
CFD	Computational Fluid Dynamics
IP	Intellectual Property
FA	Flow assurance
RF	Reflection coefficient

Appendix C

MATLAB codes

Algorithm C.1: Full MATLAB state-space equation generator

```matlab
%% Observability
clear all
close all

%% Impliment values
scale = 1e5; % Scaling to prevent stiff system

beta = 1.05e9/scale; % Bulk modulus for Benzene [Bar]
a = pi*0.0254^2; % Cross-section of the pipeline [m^2]
density = 900; % Density of fluid [kg/m^3]
l = 14; % Length of pipesegment [m]

k1 = beta/(a*l*scale); % Simplifcation to the code with scaling
k2 = (a*scale)/(l*density);
d = 1; % Friction factor for pipe roughness coefficient

n = 22; % Number of states, half pressure, half flow

An = zeros(n,n); % Making an A mastrix with n by n dimension of zeros
p = n/2+1;
```

```
 for m = 1:n/2-1 % Adding k1 and -k2*d to the state-space model
     An(m,p) = k1;
     An(m,p+1) = -k1;
     An(p,p) = -d;
     p = p+1;
 end
An(n/2,n) = k1; % Adding k1 to qN
An(n/2+1,1) = -k2;     % Adding -k2 to p1
An(n,n) = -d;
p = 1;
 for m = n/2+2:n % Adding k2 from 2 to n-1
     An(m,p) = k2;
     An(m,p+1) = -k2;
     p = p+1;
 end

Cn = zeros(n,n);  % Making a C matrix for pressure
Cn(1,1) = 1;
Cn(n/2,n/2) = 1; % Adding measurement points

Bn = zeros(n,2);% Creating the B matrix
Bn(n/2,2) = -k1;
Bn(n/2+1,1) = k2;


```

```
%% Inital values
p_in = (75*1e5)/scale; % u1 in [Bar]
q_out = 2*a; % Cross-section * 2 m/s [m^3/s]
u = [p_in;q_out]; % u for B matrix

C_estimation = Cn; % Make it easier

Q = 1e-3; % Process noise covariance
R = 1e-4; % Measurement noise covariance
Ts = 0.1; % Sampling time [s]
P_ = Q*diag(ones(n,1)); % Guessed error covariance
x_hat_ = zeros(n,1); % Estimated values set to 0

SimTime = 10000; % Simulation time
filename = 'KalmanValues.mat'; % Filename for saving data
SimFile = 'KalmanFilter'; % Simulink file

%% Smaller cross-section at control volume, wax thickness 0.5 mm
a_05 = pi*0.0249^2; % Cross-section of the section with 0.5mm wax thickness [m^2]
k1_a_05 = beta/(a_05*l*scale); % New k1 value with 0.5mm wax thickness
k2_a_05 = (scale*a_05)/(l*density); % New k2 value with 0.5mm wax thickness
```

```

for m = 0:n/2
    A_estimation = An; % Copy of state-space
       equation
    B_estimation = Bn; % Copy of state-space
       equation
    if m == 1
        A_estimation(1,n/2+1) = k1_a_05;
        A_estimation(1,n/2+2) = -k1_a_05;
        A_estimation(n/2+1,1) = -k2_a_05;
        B_estimation(n/2+1,1) = k2_a_05;
    elseif m == n/2
        A_estimation(m,n) = k1_a_05;
        A_estimation(n,n/2) = -k2_a_05;
        A_estimation(n,n/2-1) = k2_a_05;
        B_estimation(n/2,2) = -k1_a_05;
    elseif m == 0
        A_estimation = An;
        B_estimation = Bn;
    else
        A_estimation(m,n/2+m) = k1_a_05;
        A_estimation(m,n/2+1+m) = -k1_a_05;
        A_estimation(n/2+m,m-1) = k2_a_05;
        A_estimation(n/2+m,m) = -k2_a_05;
    end

    [A_d,B_d] = c2d(A_estimation,B_estimation,Ts)
```

```
       ; % Discrete A and B
    C_d = C_estimation; % Discrete C

    obser_estimation = obsv(A_estimation,
       C_estimation); % Caluclating the
       Observability matrix
    rank_estimation = rank(obser_estimation); %
       The rank of Observability matrix
    fprintf('%d. control volume, %d states. Rank
       of Observability matrix %d.\n',m,n,
       rank_estimation)
    stable = eig(A_estimation) % See if the A
       matrix is stable

%% Starting the simulation
    KalmanData = struct('A',A_d,'B',B_d,'C',C_d,'
       x',x_hat_,'P',P_,'Q',Q,'R',R); % Making a
       struct to impliment in the Kalman fiter
       with in SIMULINK
    simOut = sim(SimFile,SimTime); % Simulate in
       SIMULINK
%% Saving the values
    esti_value = simOut.get('esti');
    esti_value = esti_value.data;
    eval(['estimated_' num2str(m) '= esti_value
       (1:round(end/100):end,1:11)']);
    save(filename,['estimated_' num2str(m)],'-
```

```
        append');
end


%% Smaller cross-section at control volume, wax
   thickness 1 mm
a_1 = pi*0.0245^2; % Cross-section of the section
    with 1mm wax thickness [m^2]
k1_a_1 = beta/(a_1*l*scale); % New k1 value with
   0.5mm wax thickness
k2_a_1 = (scale*a_1)/(l*density); % New k2 value
   with 0.5mm wax thickness

for o = 1:n/2
    A_estimation = An; % Copy of state-space
       equation
    B_estimation = Bn; % Copy of state-space
       equation
    if o == 1
        A_estimation(1,n/2+1) = k1_a_1;
        A_estimation(1,n/2+2) = -k1_a_1;
        A_estimation(n/2+1,1) = -k2_a_1;
        B_estimation(n/2+1,1) = k2_a_1;
    elseif o == n/2
        A_estimation(o,n) = k1_a_1;
        A_estimation(n,n/2) = -k2_a_1;
        A_estimation(n,n/2-1) = k2_a_1;
```

```
        B_estimation(n/2,2) = -k1_a_1;
    else
        A_estimation(o,n/2+o) = k1_a_1;
        A_estimation(o,n/2+1+o) = -k1_a_1;
        A_estimation(n/2+o,o-1) = k2_a_1;
        A_estimation(n/2+o,o) = -k2_a_1;
    end

    [A_d,B_d] = c2d(A_estimation,B_estimation,Ts); % Discrete A and B
    C_d = C_estimation; % Discrete C

    obser_estimation = obsv(A_estimation,C_estimation); % Caluclating the Observability matrix
    rank_estimation = rank(obser_estimation); % The rank of Observability matrix
    fprintf('%d. control volume, %d states. Rank of Observability matrix %d.\n',o,n,rank_estimation)
    stable = eig(A_estimation) % See if the A matrix is stable

%% Starting the simulation
    KalmanData = struct('A',A_d,'B',B_d,'C',C_d,'x',x_hat_,'P',P_,'Q',Q,'R',R); % Making a struct to impliment in the Kalman fiter
```

```
        with in SIMULINK
    simOut = sim(SimFile,SimTime); % Simulate in SIMULINK
%% Saving the values
    esti_value = simOut.get(’esti’);
    esti_value = esti_value.data;
    eval([’estimated_’ num2str(n/2+o) ’= esti_value(1:round(end/100):end,1:11)’]);
    save(filename,[’estimated_’ num2str(n/2+o)],’-append’);
end
```

Algorithm C.2: MATLAB function Kalman filter

```
function x_hat = KalmanFilter(u,y,KalmanData)
    persistent init A B C Q R P_ x_
        if isempty(init)
        init = 1; % Initialize the Kalman Filter
        [A,B,C,x_,P_,Q,R] = deal(KalmanData.A, KalmanData.B, KalmanData.C, KalmanData.x, KalmanData.P, KalmanData.Q, KalmanData.R);
        end
        L = (P_*C')*pinv(C*P_*C' + R); % Kalman gain
        x = x_ + L*(y-C*x_); % Update estimate with measurement y
        P = (eye(22)-L*C)*P_*(eye(22)-L*C)' + L*R*L'; % Compute error covariance for updated estimate
        x_ = A*x + B*u; % Project ahead
        P_ = A*P*A' + Q; % Predict error covariance
        x_hat = x;
end
```

Algorithm C.3: Plots generated from estiamation and COMSOL

```
%% COMSOL and estimation plots
clear all
close all

Pressure_COMSOL_waxmid = load('waxmid.dat');
Pressure_COMSOL_waxstart = load('waxstart.dat');
Pressure_COMSOL_waxend = load('waxend.dat');

Pressure_estimate = load('KalmanValues.mat');

%% wax = 1 mm @ 50m
t_r1_50m = [0 5e3 1e4];
y_r1_50m = [Pressure_COMSOL_waxstart(1914,2)
   Pressure_COMSOL_waxstart(7856,2)
   Pressure_COMSOL_waxstart(13798,2)]/1e5;

t_hat_r1_50m_15 = 0:100:10000;
hat_r1_50m_15_15 = Pressure_estimate.estimated_15
   (:,4);
hat_r1_50m_15_14 = Pressure_estimate.estimated_15
   (:,3);
hat_r1_50m_15_16 = Pressure_estimate.estimated_15
   (:,5);

fig50m1mm = figure('Name','COMSOL values waxstart
    1mm','Color','White');
```

```
subplot(2,1,1)
plot(t_r1_50m,y_r1_50m,'b','LineWidth',2)
hold on
plot(t_hat_r1_50m_15,hat_r1_50m_15_15,'r','
   LineWidth',2)
plot(t_hat_r1_50m_15,hat_r1_50m_15_14,'y-.','
   LineWidth',2)
plot(t_hat_r1_50m_15,hat_r1_50m_15_16,'g-.','
   LineWidth',2)
xlabel('Time [s]'); ylabel('Pressure [Bar]')
legend('COMSOL simulation','Estimated nearest
   control volume','Estimated left of nearest
   control volume','Estimated right of nearest
   control volume','location','southeast')
title('Estimated vs. simulated pressure, 1 mm wax
    deposition at 50 m.')
subplot(2,1,2)
plot(t_r1_50m,y_r1_50m*1e5,'b','LineWidth',2)
hold on
plot(t_hat_r1_50m_15,hat_r1_50m_15_15*1e5,'r','
   LineWidth',2)
plot(t_hat_r1_50m_15,hat_r1_50m_15_14*1e5,'y-.','
   LineWidth',2)
plot(t_hat_r1_50m_15,hat_r1_50m_15_16*1e5,'g-.','
   LineWidth',2)
xlim([9900 10000])
ylim([73.88 73.92]*1e5)
```

```
xlabel(’Time [s]’); ylabel(’Pressure [Pa]’)

%% wax = 0.5 mm @ 50m
t_r05_50m = [0 5e3 1e4];
y_r05_50m = [Pressure_COMSOL_waxstart(20390,2)
    Pressure_COMSOL_waxstart(27416,2)
    Pressure_COMSOL_waxstart(34442,2)]/1e5;

t_hat_r05_50m_4 = 0:100:10000;
hat_r05_50m_4_4 = Pressure_estimate.estimated_4
    (:,4);
hat_r05_50m_4_3 = Pressure_estimate.estimated_4
    (:,3);
hat_r05_50m_4_5 = Pressure_estimate.estimated_4
    (:,5);

fig50m05mm = figure(’Name’,’COMSOL values
    waxstart 0.5mm’,’Color’,’White’);
subplot(2,1,1)
plot(t_r05_50m,y_r05_50m,’b’,’LineWidth’,2)
hold on
plot(t_hat_r05_50m_4,hat_r05_50m_4_4,’r’,’
    LineWidth’,2)
plot(t_hat_r05_50m_4,hat_r05_50m_4_3,’y-.’,’
    LineWidth’,2)
plot(t_hat_r05_50m_4,hat_r05_50m_4_5,’g-.’,’
    LineWidth’,2)
```

```
xlabel('Time [s]'); ylabel('Pressure [Bar]')
title('Estimated vs. simulated pressure, 0.5 mm
    wax deposition at 50 m.')
legend('COMSOL simulation','Estimated nearest
    control volume','Estimated left of nearest
    control volume','Estimated right of nearest
    control volume','location','southeast')
subplot(2,1,2)
plot(t_r05_50m,y_r05_50m*1e5,'b','LineWidth',2)
hold on
plot(t_hat_r05_50m_4,hat_r05_50m_4_4*1e5,'r','
    LineWidth',2)
plot(t_hat_r05_50m_4,hat_r05_50m_4_3*1e5,'y-.','
    LineWidth',2)
plot(t_hat_r05_50m_4,hat_r05_50m_4_5*1e5,'g-.','
    LineWidth',2)
xlim([9900 10000])
ylim([73.9 73.96]*1e5)
xlabel('Time [s]'); ylabel('Pressure [Pa]')

%% wax = 1 mm @ 75 m
t_r1_75m = [0 5e3 1e4];
y_r1_75m = [Pressure_COMSOL_waxmid(3200,2)
    Pressure_COMSOL_waxmid(9569,2)
    Pressure_COMSOL_waxmid(15938,2)]/1e5;

t_hat_r1_75m_17 = 0:100:10000;
```

```
hat_r1_75m_17_17 = Pressure_estimate.estimated_17
    (:,6);
hat_r1_75m_17_16 = Pressure_estimate.estimated_17
    (:,5);
hat_r1_75m_17_18 = Pressure_estimate.estimated_17
    (:,7);

fig75m1mm = figure('Name','COMSOL values waxmid 1
   mm','Color','White');
subplot(2,1,1)
plot(t_r1_75m,y_r1_75m,'b','LineWidth',2)
hold on
plot(t_hat_r1_75m_17,hat_r1_75m_17_17,'r','
   LineWidth',2)
plot(t_hat_r1_75m_17,hat_r1_75m_17_16,'y-.','
   LineWidth',2)
plot(t_hat_r1_75m_17,hat_r1_75m_17_18,'g-.','
   LineWidth',2)
xlabel('Time [s]'); ylabel('Pressure [Bar]')
legend('COMSOL simulation','Estimated nearest
   control volume','Estimated left of nearest
   control volume','Estimated right of nearest
   control volume','location','southeast')
title('Estimated vs. simulated pressure, 1 mm wax
    deposition at 75 m.')
subplot(2,1,2)
plot(t_r1_75m,y_r1_75m*1e5,'b','LineWidth',2)
```

```
hold on
plot(t_hat_r1_75m_17,hat_r1_75m_17_17*1e5,'r','
    LineWidth',2)
plot(t_hat_r1_75m_17,hat_r1_75m_17_16*1e5,'y-.','
    LineWidth',2)
plot(t_hat_r1_75m_17,hat_r1_75m_17_18*1e5,'g-.','
    LineWidth',2)
xlim([9900 10000])
ylim([73.35 73.5]*1e5)
xlabel('Time [s]'); ylabel('Pressure [Pa]')

%% wax = 0.5 mm @ 75 m
t_r05_75m = [0 5e3 1e4];
y_r05_75m = [Pressure_COMSOL_waxmid(22334,2)
    Pressure_COMSOL_waxmid(28859,2)
    Pressure_COMSOL_waxmid(35384,2)]/1e5;

t_hat_r05_75m_6 = 0:100:10000;
hat_r05_75m_6_6 = Pressure_estimate.estimated_6
    (:,6);
hat_r05_75m_6_5 = Pressure_estimate.estimated_6
    (:,5);
hat_r05_75m_6_7 = Pressure_estimate.estimated_6
    (:,7);

fig75m05mm = figure('Name','COMSOL values waxmid
    0.5mm','Color','White');
```

```
subplot(2,1,1)
plot(t_r05_75m,y_r05_75m,'b','LineWidth',2)
hold on
plot(t_hat_r05_75m_6,hat_r05_75m_6_6,'r','LineWidth',2)
plot(t_hat_r05_75m_6,hat_r05_75m_6_5,'y-.','LineWidth',2)
plot(t_hat_r05_75m_6,hat_r05_75m_6_7,'g-.','LineWidth',2)
xlabel('Time [s]'); ylabel('Pressure [Bar]')
legend('COMSOL simulation','Estimated nearest control volume','Estimated left of nearest control volume','Estimated right of nearest control volume','location','southeast')
title('Estimated vs. simulated pressure, 0.5 mm wax deposition at 75 m.')
subplot(2,1,2)
plot(t_r05_75m,y_r05_75m*1e5,'b','LineWidth',2)
hold on
plot(t_hat_r05_75m_6,hat_r05_75m_6_6*1e5,'r','LineWidth',2)
plot(t_hat_r05_75m_6,hat_r05_75m_6_5*1e5,'y-.','LineWidth',2)
plot(t_hat_r05_75m_6,hat_r05_75m_6_7*1e5,'g-.','LineWidth',2)
xlim([9900 10000])
ylim([73.35 73.5]*1e5)
```

```
xlabel('Time [s]'); ylabel('Pressure [Pa]')

%% wax = 1 mm @ 100m
t_r1_100m = [0 5e3 1e4];
y_r1_100m = [Pressure_COMSOL_waxend(4029,2)
    Pressure_COMSOL_waxend(9971,2)
    Pressure_COMSOL_waxend(15913,2)]/1e5;

t_hat_r1_100m_19 = 0:100:10000;
hat_r1_100m_19_8 = Pressure_estimate.estimated_19
    (:,8);
hat_r1_100m_19_7 = Pressure_estimate.estimated_19
    (:,7);
hat_r1_100m_19_9 = Pressure_estimate.estimated_19
    (:,9);

fig100m1mm = figure('Name','COMSOL values waxend
    1mm','Color','White');
subplot(2,1,1)
plot(t_r1_100m,y_r1_100m,'b','LineWidth',2)
hold on
plot(t_hat_r1_100m_19,hat_r1_100m_19_8,'r','
    LineWidth',2)
plot(t_hat_r1_100m_19,hat_r1_100m_19_7,'y-.','
    LineWidth',1.5)
plot(t_hat_r1_100m_19,hat_r1_100m_19_9,'g-.','
    LineWidth',1.5)
```

```
xlabel('Time [s]'); ylabel('Pressure [Bar]')
legend('COMSOL simulation','Estimated at control volume','Estimated left of nearest control volume','Estimated right of nearest control volume','location','southeast')
title('Estimated vs. simulated pressure, 1 mm wax deposition at 100 m.')
subplot(2,1,2)
plot(t_r1_100m,y_r1_100m*1e5,'b','LineWidth',2)
hold on
plot(t_hat_r1_100m_19,hat_r1_100m_19_8*1e5,'r','LineWidth',2)
plot(t_hat_r1_100m_19,hat_r1_100m_19_7*1e5,'y-.','LineWidth',2)
plot(t_hat_r1_100m_19,hat_r1_100m_19_9*1e5,'g-.','LineWidth',2)
xlim([9900 10000])
ylim([72.8 73.12]*1e5)
xlabel('Time [s]'); ylabel('Pressure [Pa]')

%% wax = 0.5 mm @ 100m
t_r05_100m = [0 5e3 1e4];
y_r05_100m = [Pressure_COMSOL_waxend(22289,2) Pressure_COMSOL_waxend(29315,2) Pressure_COMSOL_waxend(36341,2)]/1e5;

t_hat_r05_100m_8 = 0:100:10000;
```

```
hat_r05_100m_8_8 = Pressure_estimate.estimated_8(:,8);
hat_r05_100m_8_7 = Pressure_estimate.estimated_8(:,7);
hat_r05_100m_8_9 = Pressure_estimate.estimated_8(:,9);

fig100m05mm = figure('Name','COMSOL values waxend 0.5mm','Color','White');
subplot(2,1,1)
plot(t_r05_100m,y_r05_100m,'b','LineWidth',2)
hold on
plot(t_hat_r05_100m_8,hat_r05_100m_8_8,'r','LineWidth',2)
plot(t_hat_r05_100m_8,hat_r05_100m_8_7,'y-.','LineWidth',1.5)
plot(t_hat_r05_100m_8,hat_r05_100m_8_9,'g-.','LineWidth',1.5)
xlabel('Time [s]'); ylabel('Pressure [Bar]')
title('Estimated vs. simulated pressure, 0.5 mm wax deposition at 100 m.')
legend('COMSOL simulation','Estimated at control volume','Estimated left of nearest control volume','Estimated right of nearest control volume','location','southeast')
subplot(2,1,2)
plot(t_r05_100m,y_r05_100m*1e5,'b','LineWidth',2)
```

```
hold on
plot(t_hat_r05_100m_8,hat_r05_100m_8_8*1e5,'r','
    LineWidth',2)
plot(t_hat_r05_100m_8,hat_r05_100m_8_7*1e5,'y-.',
    'LineWidth',2)
plot(t_hat_r05_100m_8,hat_r05_100m_8_9*1e5,'g-.',
    'LineWidth',2)
xlim([9900 10000])
ylim([72.8 73.1]*1e5)
xlabel('Time [s]'); ylabel('Pressure [Pa]')

%% Values from comsol

x_r1_50m_full = Pressure_COMSOL_waxstart
    (11885:17826,1);
y_r1_50m_full = Pressure_COMSOL_waxstart
    (11885:17826,2)/1e5;
x_r05_50m_full = Pressure_COMSOL_waxstart
    (31879:38904,1);
y_r05_50m_full = Pressure_COMSOL_waxstart
    (31879:38904,2)/1e5;
x_r1_75m_full = Pressure_COMSOL_waxmid
    (12739:19107,1);
y_r1_75m_full = Pressure_COMSOL_waxmid
    (12739:19107,2)/1e5;
x_r05_75m_full = Pressure_COMSOL_waxmid
    (32158:38682,1);
```

```
y_r05_75m_full = Pressure_COMSOL_waxmid
    (32158:38682,2)/1e5;
x_r1_100m_full = Pressure_COMSOL_waxend
    (11885:17826,1);
y_r1_100m_full = Pressure_COMSOL_waxend
    (11885:17826,2)/1e5;
x_r05_100m_full = Pressure_COMSOL_waxend
    (31879:38904,1);
y_r05_100m_full = Pressure_COMSOL_waxend
    (31879:38904,2)/1e5;

length = [0 7:14:147 150];
y_empty = [75 Pressure_estimate.estimated_0(end
    ,:) 72];
hat_y_r1_50m = [75 Pressure_estimate.estimated_15
    (end,:) 72];
hat_y_r05_50m = [75 Pressure_estimate.estimated_4
    (end,:) 72];
hat_y_r1_75m = [75 Pressure_estimate.estimated_17
    (end,:) 72];
hat_y_r05_75m = [75 Pressure_estimate.estimated_6
    (end,:) 72];
hat_y_r1_100m = [75 Pressure_estimate.
    estimated_19(end,:) 72];
hat_y_r05_100m = [75 Pressure_estimate.
    estimated_8(end,:) 72];

```

```matlab
fig_compare50 = figure('Name','COMSOL profile vs MATLAB profile','Color','White');
subplot(3,1,1)
plot(x_r1_50m_full,y_r1_50m_full,'b','LineWidth',2)
hold on
plot(length,hat_y_r1_50m,'r','LineWidth',2)
plot(length,y_empty,'y-.','LineWidth',1)
xlabel('Length [m]'); ylabel('Pressure [Bar]')
title('Estimated vs. simulated pressure, wax deposition at 50 m. Time = 10000 seconds')
legend('Simulated in COMSOL','Estimated with 1mm wax in control volume','Estimated with no wax')
subplot(3,1,2)
plot(x_r05_50m_full,y_r05_50m_full,'b','LineWidth',2)
hold on
plot(length,hat_y_r05_50m,'k','LineWidth',2)
plot(length,y_empty,'g-.','LineWidth',1)
xlabel('Length [m]'); ylabel('Pressure [Bar]')
legend('Simulated in COMSOL','Estimated with 0.5 mm wax in control volume','Estimated with no wax')
subplot(3,1,3)
plot(length,Pressure_COMSOL_waxstart(11885:495:17826,2)'-hat_y_r1_50m*1e5,'r-.','LineWidth',1)
```

```
hold on
plot(length,Pressure_COMSOL_waxstart
    (11885:495:17826,2)'-[75 Pressure_estimate.
    estimated_0(end,:) 72]*1e5,'y-.','LineWidth',1)
plot(length,Pressure_COMSOL_waxstart
    (31879:585:38904,2)'-hat_y_r05_50m*1e5,'k','
    LineWidth',2)
plot(length,Pressure_COMSOL_waxstart
    (31879:585:38904,2)'-[75 Pressure_estimate.
    estimated_0(end,:) 72]*1e5,'g-.','LineWidth',1)
ylim([-0.8 0.8]*1e5)
xlabel('Length [m]'); ylabel('Pressure [Pa]')
title('Estimated vs. simulated pressure, error')
legend('Error 1 mm wax in control volume','Error
    between empty pipe and simulated wax 1 mm','
    Error 0.5 mm wax in control volume','Error
    between empty pipe and simulated wax 0.5 mm','
    location','southoutside')

fig_compare75 = figure('Name','COMSOL profile vs
    MATLAB profile','Color','White');
subplot(3,1,1)
plot(x_r1_75m_full,y_r1_75m_full,'b','LineWidth'
    ,2)
hold on
plot(length,hat_y_r1_75m,'r','LineWidth',2)
plot(length,y_empty,'y-.','LineWidth',1)
```

```
xlabel('Length [m]'); ylabel('Pressure [Bar]')
title('Estimated vs. simulated pressure, wax deposition at 75 m. Time = 10000 seconds')
legend('Simulated in COMSOL','Estimated with 1mm wax in control volume','Estimated with no wax')
subplot(3,1,2)
plot(x_r05_75m_full,y_r05_75m_full,'b','LineWidth',2)
hold on
plot(length,hat_y_r05_75m,'k','LineWidth',2)
plot(length,y_empty,'g-.','LineWidth',1)
xlabel('Length [m]'); ylabel('Pressure [Bar]')
legend('Simulated in COMSOL','Estimated with 0.5 mm wax in control volume','Estimated with no wax')
subplot(3,1,3)
plot(length,Pressure_COMSOL_waxmid(12739:530:19107,2)'-hat_y_r1_75m*1e5,'r-.','LineWidth',2)
hold on
plot(length,Pressure_COMSOL_waxmid(12739:530:19107,2)'-[75 Pressure_estimate.estimated_0(end,:) 72]*1e5,'y-.','LineWidth',1)
plot(length,Pressure_COMSOL_waxmid(32158:543:38682,2)'-hat_y_r05_75m*1e5,'k','LineWidth',2)
plot(length,Pressure_COMSOL_waxmid
```

```
   (32158:543:38682,2)'-[75 Pressure_estimate.
   estimated_0(end,:) 72]*1e5,'g-.','LineWidth',1)
ylim([-0.8 0.8]*1e5)
xlabel('Length [m]'); ylabel('Pressure [Pa]')
title('Estimated vs. simulated pressure, error')
legend('Error 1 mm wax in control volume','Error
   between empty pipe and simulated wax 1 mm','
   Error 0.5 mm wax in control volume','Error
   between empty pipe and simulated wax 0.5 mm','
   location','southoutside')


fig_compare100 = figure('Name','COMSOL profile vs
    MATLAB profile','Color','White');
subplot(3,1,1)
plot(x_r1_100m_full,y_r1_100m_full,'b','LineWidth
   ',2)
hold on
plot(length,hat_y_r1_100m,'r','LineWidth',2)
plot(length,y_empty,'y-.','LineWidth',1)
xlabel('Length [m]'); ylabel('Pressure [Bar]')
title('Estimated vs. simulated pressure, wax
   deposition at 100 m. Time = 10000 seconds')
legend('Simulated in COMSOL','Estimated with 1mm
   wax in control volume','Estimated with no wax')
subplot(3,1,2)
plot(x_r05_100m_full,y_r05_100m_full,'b','
```

```
LineWidth',2)
hold on
plot(length,hat_y_r05_100m,'k','LineWidth',2)
plot(length,y_empty,'g-.','LineWidth',1)
xlabel('Length [m]'); ylabel('Pressure [Bar]')
legend('Simulated in COMSOL','Estimated with 0.5 mm wax in control volume','Estimated with no wax')
subplot(3,1,3)
grid on
plot(length,Pressure_COMSOL_waxend(11885:495:17826,2)'-hat_y_r1_100m*1e5,'r','LineWidth',2)
hold on
plot(length,Pressure_COMSOL_waxend(11885:495:17826,2)'-[75 Pressure_estimate.estimated_0(end,:) 72]*1e5,'y-.','LineWidth',1)
plot(length,Pressure_COMSOL_waxend(31879:585:38904,2)'-hat_y_r05_75m*1e5,'k','LineWidth',2)
plot(length,Pressure_COMSOL_waxend(31879:585:38904,2)'-[75 Pressure_estimate.estimated_0(end,:) 72]*1e5,'g-.','LineWidth',1)
ylim([-0.8 1]*1e5)
xlabel('Length [m]'); ylabel('Pressure [Pa]')
title('Estimated vs. simulated pressure, error')
legend('Error 1 mm wax in control volume','Error
```

```
    between empty pipe and simulated wax 1 mm','
    Error 0.5 mm wax in control volume','Error
    between empty pipe and simulated wax 0.5 mm','
    location','southoutside')

%% Saving plots

saveas(fig50m1mm,'plots/plot50m1mm','epsc')
saveas(fig50m05mm,'plots/plot50m05mm','epsc')
saveas(fig75m1mm,'plots/plot75m1mm','epsc')
saveas(fig75m05mm,'plots/plot75m05mm','epsc')
saveas(fig100m1mm,'plots/plot100m1mm','epsc')
saveas(fig100m05mm,'plots/plot100m05mm','epsc')
saveas(fig_compare50,'plots/plotcompare50','epsc'
    )
saveas(fig_compare75,'plots/plotcompare75','epsc'
    )
saveas(fig_compare100,'plots/plotcompare100','
    epsc')
```

Leif Andreas Hirsti

Pressure loss detection and estimation of a Cold Flow cooling system.

NTNU
Norwegian University of
Science and Technology