University of
South-Eastern Norway

www.usn.no

FMH606 Master's Thesis 2018

Industrial IT and Automation

Model based estimation of fluid flow through a venturi channel: Kalman filters and observers

Om Prakash Chapagain

121179

Faculty of Technology, Natural sciences and Maritime Sciences
Campus Porsgrunn

University of South-Eastern Norway

www.usn.no

Course: FMH606 Master's Thesis, 2018

Title: Model based estimation of fluid flow through a venturi channel: Kalman filters and observers

Number of pages: 145

Keywords: State observer, Flow estimation, Saint Venant Equations, Linearization, State space model, Hurwitz, Unscented Kalman filter, Extended Kalman filter, Piping and Instrumentation Diagram, Sensitivity Index, Venturi flume

Student:	Chapagain, Om Prakash
Supervisor:	Sharma, Roshan
External partner:	Statoil, Kelda drilling controls ASA
Availability:	Open

The University of South-Eastern Norway takes no responsibility for the results and conclusions in this student report.

University of South-Eastern Norway

www.usn.no

Summary:

In oil drilling operation, the estimation of returning flow is vital for safe operation. The return flow gives an early warning of kick-loss phenomenon. Detecting kick-loss is important to prevent uncontrolled well-blowout. The estimated slow of the returning fluid is a primary indicator of a kick or a loss. The existing methods for flow estimation angle-based paddle sensor or Coriolis flow meter. These methods are either unreliable or too expensive.

The aim of this thesis work is to investigate other reliable flow estimation methods. Therefore, dynamic model-based flow estimation technique is proposed here. Using dynamic model, suitable estimators such as Luenberger observer and Kalman filters can be designed. Such technique reduces the operational and maintenance costs of using expensive flow measuring mechanical devices such as Coriolis flow meters.

A model of top-side open venturi channel is developed as a set of St. Venant equations for one spatial dimension which are a class of quasi-linear hyperbolic partial differential equations (PDEs). These PDEs are reduced to a set of nonlinear first order ordinary differential equation (ODEs) using orthogonal collocation methods and Lagrange interpolating polynomials. The nonlinear ODEs are linearized around a suitable operating point. Based on the linear ODEs, linear Luenberger full order state observer as well as linear Kalman filter (LKF) is designed. For designing nonlinear Kalman filters such as extended Kalman filter (EKF) and unscented Kalman filter (UKF), nonlinear ODEs are used.

Different types of state estimators are applied to model and real system. UKF outperforms all other estimators investigated during this thesis work. UKF converges faster and is more robust. It is able to filter out the noise. The error between the estimated flow rates and the measured flow rates is minimized. However, a proper tuning of process and measurement noise covariance matrices is necessary for UKF to provide optimal state estimation.

For increased accuracy in flow estimation, an improved version of the ODEs are recommended. PDE-observers are recommended for further improvement in the flow estimation. Based on the results of this thesis work, the future of model based flow estimation technique in oil industry looks promising.

The University of South-Eastern Norway takes no responsibility for the results and conclusions in this student report.

Preface

This work is a result of a master's thesis. It is a part of the two-years master's degree program in the field of Industrial IT and Automation at the University of South-Eastern Norway (USN). Being an engineer in the field of cybernetics, anything that has to do with modeling, control and simulation is naturally exciting. The proposed thesis work is based on a simple idea of flow estimation on an open channel setup. This thesis work is a part of on-going "Semi-kidd" project at USN. Both the theoretical and experimental aspect of the work is presented in this thesis. As the topics for the thesis work were published, this project was an eye-catcher. The purpose of this thesis work is to analyze the possibility of replacing expensive flow meters by introducing the mathematical model-based flow estimation techniques. Techniques such as state observers and Kalman filters are suitable for this project. That is an exciting field to work on. Prospect of being one of the first persons to step on to it is another tempting factor.

During this thesis work, I received an incredible support from my supervisor, Dr. Roshan Sharma. Without him, the project would not have been as fruitful. It was my pleasure to work with him. For the smooth running of the project, the help and contribution from one particularly important person cannot be forgotten, PHD research fellow at USN, Mr. Khim Chhantyal. I would also like to thank my family for their unconditional support during this hectic period. My special thank goes to my little niece, Erica, who has been the source of my happiness the whole time. It was my pleasure to discuss some aspects of the mathematics with mathematician Ms. Januka Subedi. The collective effort of these people propelled me to finalize this work and help me see the light at the end of the tunnel as my two-years master's program is coming to an end. I would, finally, like to thank all the teachers, members and staffs of the USN who are, directly or indirectly, involved in helping me get through this study program.

Porsgrunn, 15-05-2018

Om Prakash Chapagain

Contents

Figures

Tables

Abbreviations

Here is the list of abbreviations that are used throughout the report unless specified in the specific sections.

Abbreviations	Full form
1D	Unidirectional or 1 directional
ANN	Artificial Neural Network
ARE	Algebraic Riccati Equation
BOP	Blowout Preventer
C-DARE	Continuous Differential Algebraic Riccati Equation
DARE	Discrete Algebraic Riccati Equation
EiSj	Experiment “i”, setup “j” for data acquisition at the venturi rig. E1S1, E1S2, E2S1, E2S2, E3S1, E3S2, E4S1 and E4S2
EKF	Extended Kalman Filter
FT14A	Coriolis flow meter for measuring mass flow rate
GUI	Graphical user interface
HOT	Higher Order Terms
hM MODEL	Model based on level (h) and mass flow rate (M)
LHS	Left Hand Side
LKF	Linear Kalman Filter
LPF	Lowpass Filter
LT18	Ultrasonic level transmitter for measuring fluid level at collocation point 1
LT19	Radar level transmitter for measuring fluid level at collocation point 1
LTI	Linear Time Invariant
MEDF	Median filter. Built-in filter available in MATLAB. “medflt1.m”
MPC	Model Predictive Controller
ODE	Ordinary Differential Equations

P&ID	Piping and Instrumentation Diagram
PDE	Partial Differential Equations
PI	Proportional Integral
RHS	Right Hand Side
RK4	Runge-Kutta 4th order
SISO	Single Input Single Output
SS	State Space
SSPE	Sum Squared Prediction Error
SVE	Saint Venant Equations
UKF	Unscented Kalman Filter
USN	University of South-Eastern Norway
WMAF	Weighted Moving Average Filter
AQ MODEL	Model based on wetted cross-sectional area, (A) and volumetric flow rate, (Q)

Nomenclature

Here is the list of abbreviations that are used throughout the report unless specified in the specific sections.

SYMBOLS	DESCRIPTION	UNIT
$\hat{h}_2$	Estimated mass flow rate at the collocation point 1. First estimated state given by estimator, $\hat{y} = \hat{x}_1$	$[m]$
$\widehat{M}_1$	Estimated mass flow rate at the collocation point 1. Second estimated state given by estimator, $\hat{x}_2$	$\left[\frac{kg}{s}\right]$
$\widehat{M}_2$	Estimated mass flow rate at the collocation point 1. Third estimated state given by estimator, $\hat{x}_3$	$\left[\frac{kg}{s}\right]$
h_1	Level measured at the collocation point 1. Input of the venturi model, u	$[m]$
h_2	Level measured mass flow rate at the collocation point 2. First state and the output of the venturi model, $y = x_1$	$[m]$
I_1	Static hydrostatic pressure due to change in fluid cross-sectional area (A) at a given point along the channel	$[m^3]$
I_2	Static hydrostatic pressure due to width variation at a given point along the channel	$[m^3]$
M_1	Predicted mass flow rate at the collocation point 1. Second state of the venturi model, x_2	$\left[\frac{kg}{s}\right]$
M_2	Predicted mass flow rate at the collocation point 2. Third state of the venturi model, x_3	$\left[\frac{kg}{s}\right]$
S_b	Channel bed slope. Depends on φ	$[unitless]$
S_f	Friction slope at a given point along the channel	$[unitless]$
$S_s = S_L$	Side wall slope. Depends on α	$[unitless]$
T_f	Lowpass filter time constant	$[s]$
T_s	Sampling time. (data logging time for experiment)	$[s]$
$\dot{V} = Q$	Volumetric flow rate	$\left[\frac{m^3}{s}\right]$
c_1	Collocation point 1. Position of the input level measurement sensor (LT19 – radar)	$[m]$

c_2	Collocation point 2. Position of the output level measurement sensor (LT18 – ultrasonic)	$[m]$
k_s	Strickler friction coefficient, depends on n_M	$\left[\frac{\sqrt[3]{m}}{s}\right]$
$\dot{m} = M$	Mass flow rate	$\left[\frac{kg}{s}\right]$
n_M	Manning's roughness coefficient	$\left[\frac{s}{\sqrt[3]{m}}\right]$
n_u	Number of inputs to the system (model)	$[unitless]$
n_x	Number of states of the system (model)	$[unitless]$
n_y	Number of outputs from the system (model)	$[unitless]$
$\hat{x}$	Estimated states. In case of dynamic venturi model, $\hat{x}$, is a column vector of model states. $\hat{x} = [\hat{x}_1 \quad \hat{x}_2 \quad \hat{x}_3]^T = [\hat{h}_2 \quad \hat{M}_1 \quad \hat{M}_2]^T$	$\left[m, \frac{kg}{s}, \frac{kg}{s}\right]$
σ^2	Variance (squared standard deviation)	
$\equiv$	"Equivalent to" or "the same as"	
h	Fluid depth (level) at a given point along the channel	$[m]$
A	Wetted cross-sectional area at a given point along the channel	$[m^2]$
D	Rate of change of area w.r.t. level. Also, the scaling factor for rate of change of level w.r.t. time and rate of change of area w.r.t. time	$[m]$
L	Channel length between collocation point 1, c_1 and collocation point 2, c_2	$[m]$
P	Wetted perimeter at a given point along the channel	$[m]$
R	Hydraulic radius at a given point along the channel	$[m]$
W	Base width of the channel at a given point along the channel	$[m]$
g	Acceleration due to gravity	$\left[\frac{m}{s^2}\right]$
u	Input to the dynamic model of the venturi system as well as to the state estimators. $u = h_1$	$[m]$
v	Mean velocity of the liquid	$\left[\frac{m}{s}\right]$

α	Angle made by the side wall of the channel (rotating around x-axis) with the horizontal plane (xy-plane). Upward rotation of the side wall is the positive direction, increasing α.	$[degrees]$
β	Momentum correction factor (Boussinesq's coefficient)	$[unitless]$
θ	Angle made by the diverging wall of the channel (rotating around z-axis) with the vertical plane (xz-plane). Outward rotation of the diverging wall is the positive direction, increasing θ.	$[degrees]$
μ	Mean	
ρ	Density of the fluid	$\left[\frac{kg}{m^3}\right]$
σ	Standard deviation	
φ	Angle made by the channel base (rotating around y-axis) with the horizontal plane (xy plane). Upward rotation of the channel base is the positive direction, increasing φ.	$[degrees]$

1 Introduction

This chapter discusses the introductory part of the thesis. This is a transitional chapter towards the main objective of the project at hand which is analyzed in the subsequent chapters. It defines the scope of the document and the background of the problem that is to be tackled. It encompasses the big picture of the project for quick understanding of what the main goal is and the methodology used.

1.1 Scope

In this report, the focus is on developing and implementing the state observer and an optimal estimator to estimate the flow through an open channel. A mathematical model is used to design suitable observers and estimators. It is also important to analyze the stability of the model before proceeding with estimating techniques. Development and implementation of simulator is also discussed in detail. The report is divided into three main sections. Each section is subdivided into chapters that are closely related.

- Section 1: Structure of the venturi flume (open channel) and a mathematical model to describe the flow dynamics through it.
- Section 2: Flow estimation using the state observer and optimal estimator (Kalman filter) and implementation on the real process.
- Section 3: Discussions, future work and conclusion

The structure of the report follows a specific pattern. First part of a chapter focuses on the theory behind the topic of discussion. The second part elaborates and analyses the theory by implementing the idea on mathematical model. Implementation on the real system is discussed in the last contextual chapter about experimentation and result.

1.2 Background

Efficiency. One word that is a major goal of a company. The context in which efficiency is defined varies from industry to industry. Oil industry, for years, has been focusing on minimizing the operational cost and maximizing the oil production to meet the energy demand of the general population. Due to the imposed safety regulations, safety of the worker is extremely important. To meet these demands, oil industry is constantly seeking new ways to optimize the sales profit while abiding by the laws. Oil industry is playing a catch-up with the fast pace of the modern technology since they are still using the old sensor-based measuring techniques. There have been suggested different approaches on how to optimize the sales profit. For instance, by automating the flow and pressure measurements thus eliminating the manual manpower which allows them to cut the manual labor cost and increase the safety of the rig. In this thesis, the methodology proposed is based on a mathematical (physical) model with an aim to replace the expensive mechanical devices. There have been some works in this field previously to estimate the flow on the top side using mathematical approaches. As suggested in a paper, the flow can be estimated using the idea of minimum specific energy and Froude number at the critical depth [1]. Good old Bernoulli’s principle is at the heart of the methods that uses energy models. Model predictive technique such as artificial neural network is also in the works which produces complex mathematical model. However, there has been no effort, as of today to my knowledge, on estimating flow in an open channel using an observer or an estimator such as Kalman filter. In this case, the mathematical model is based on the dynamic model of the venturi flume that gives rise to the Saint Venant Equations (SVEs).

1.3 Overview

This section covers the overview, the big picture, of the task at hand. Mass flow rate through open channel (M) is estimated using observers and estimators such as Kalman filters. This approach, however, requires the level measurements over the top side of the open channel at two points along the channel c_1 and c_2. In this case, the input to the system is the level measured at a point c_1 along the venturi channel. The output of the system is the level measured at a point c_2 along the throat (constriction) of the channel as shown in Figure 1.1. The mass flow rate is estimated at both points where the level sensors $(LT19)$ and $(LT18)$ are installed as (M_1) and (M_2) respectively. At the steady state, both flow rates should be same. At the point c_1, LT19 measures the input level h_1. Similarly, At the point c_2, LT18 measures the input level h_2. It is important to note that the position for the input level measurement, c_1 must be at the subcritical region of the flow regime. In other words, level measured at the position 1, h_1 must be higher than the level measured at the position 2, h_2 as given by (10.10). LT18 and LT19 are the level transmitters used at the lab at USN. Selecting the position 2 at the throat of the venturi channel satisfies this condition given that the position 1 is in subcritical region.

$$h_1 > h_2 \tag{1.1}$$

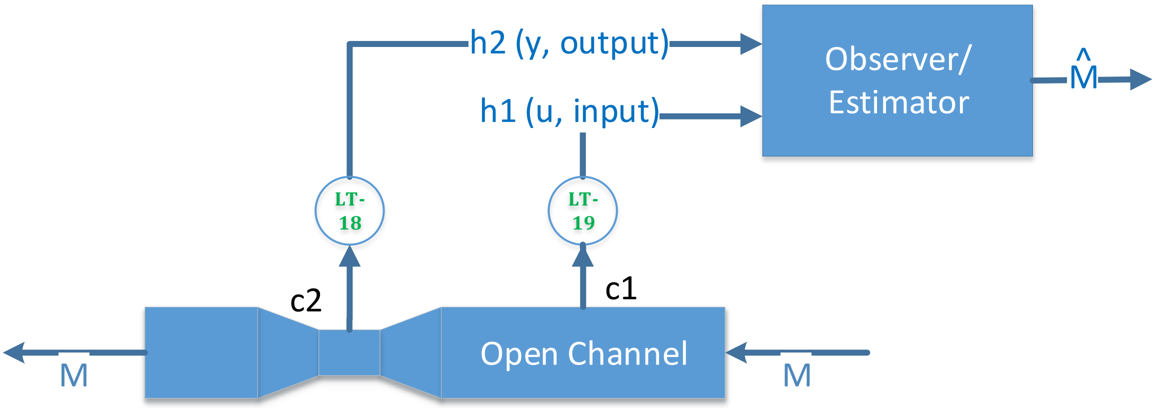

Figure 1.1: Overview of the flow estimation through open venturi channel (top view).

1.4 Objectives

The main objective of this thesis is to use mathematical model and design a suitable state observer or estimator for estimating the mud-flow through the open venturi channel. The aim of estimation of return flow is to detect an early warning of kick-loss that might occur during drilling operation. The early warning of kick-loss can prevent hazardous situations such as uncontrolled blowout. Such detection can also help mud engineer to determine the mud type required for efficient drilling operation. Importance of kick-loss detection is discussed in detail in later chapters. Therefore, flow estimation of the returning fluid is vital for safe operation. For reliable flow estimation, expensive Coriolis flow meters are used today. The benefit of using mechanistic model is that it helps to replace expensive flow measuring devices (sensors or flow meters) such as Coriolis flow meters. That will significantly reduce the installation, maintenance and operational cost. To achieve this objective, the task is divided into smaller subtasks as listed below:

- Linearize the nonlinear ODEs around a suitable operating point
- Develop and implement a linear and a nonlinear simulator and compare the outcome
- Perform the model parameter sensitivity analysis
- Determine the stability and the observability of the model
- Design a stable linear observer and compare with both linear and nonlinear simulator
- Design an optimal estimator in the form of Kalman filter and compare it with the linear observer, linear model and nonlinear model
- Perform experiments at the lab and log the appropriate data
- Test the linear observer and Kalman filter with the data from the real process
    - Investigate the effect of noise in the observer
    - Implement different filters for the input signal and discuss their performance and effect on the estimates
- If necessary, optimize the model parameters to reflect the behavior of the real system in the model
- Design and implement nonlinear estimator (Extended and Unscented Kalman filter)
- Implement and compare the nonlinear estimator with the model and the real data

These goals are also defined under the task description in Appendix 12.

1.5 Requirements

Designing an observer mathematically is a portion of the full extent of the task. For a full-fledged working observer, programming, testing and deployment is necessary. To achieve this, some software modules are used. For this thesis work, the software modules used are listed below:

- LabVIEW from National Instruments
    - With control design and simulation toolkit
- MATLAB from MathWorks
    - With Simulink module
- Visio – diagram drawing tool from Microsoft

1.6 Navigating through the report

A digital version such as ".docx" and ".pdf" of the report contains many references and cross-references to avoid the cluttering with same information repeating over again. References to chapters, sections, figures, equations, tables and other relevant contents are hyperlinked (cross-referred). Clicking on a link will take to the linked section. Going back to the previously viewed page in Adobe Reader DC (version 2018.011.20038 or newer) is done from the menu bar as: View>Page Navigation>Previous View or simply by using keyboard shortcut "left Alt + left arrow". The keyboard shortcut also works in case of the Microsoft Word 2016 or later.

SECTION – I

DRILLING OPERATION,

VENTURI FLUME AND ITS MATHEMATICAL MODEL

2 Oil well drilling operation

This chapter casts some light on the modern oil well drilling operation offshore. Standardized drill operation consists of specific sequence of operations. In this chapter, the focus is on offshore drilling method.

2.1 Drilling operation

A thick walled large diameter hollow tube called conductor tube is embedded into the sea-floor. A jet bit is inserted through the conductor to drill the rocks and sediments on the sea floor. The rock cuttings are removed by forcing high pressure sea water through the nozzles of the jet bit. The jet bit is removed after drilling a few hundred meters into the sea bed depending on type of the sea bed and the bed rock. A second conductor tube, with smaller diameter than the previous conductor, is inserted and is fixed in place by injecting concrete that forms the protective layer between the conductor tube and the well bored as shown in Figure 2.1. The concrete is injected to prevents the water from the surroundings to enter the well bore. A smaller diameter drill bit is then used to dig the well further. After a certain depth, a steel tubing called casing is inserted and is fixed in place by injecting concrete again. A riser, a tube through which a special kind of drilling fluid called drilling mud is returned, is lowered and locked in place along with the blow-out preventer valve (a set of advance high-pressure safety valves) [2]. This is the preliminary setup of the drilling operation offshore. From this stage onwards, the procedure is similar to the onshore drilling procedure.

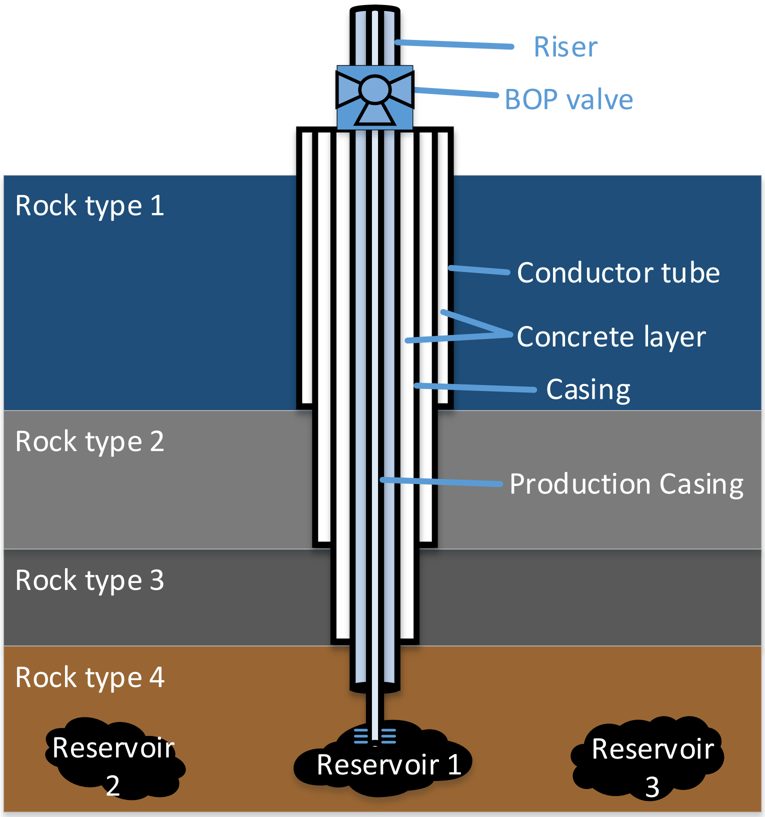

Figure 2.1: Oil well drilling overview (inside view)

The drilling continuous by inserting a smaller drill bit as a new rock types is encountered. During this stage, a clear drilling mud is injected into the well using a pump to take out the cuttings and chippings [3]. The mud is circulated continuously. Rock cuttings carried out by the mud is analyzed for the chemical composition of the mixture. It helps to determine the type of drilling mud required to tackle the pressure difference and prevent the blow out. The cuttings are removed from the mud using cutting removal equipment. The cleaned mud is put into the active mud pit for reuse. The closed loop system is illustrated in Figure 2.2. The process continuous until the oil reservoir is reached. Final casing is then inserted and cemented on place. The last part is then to insert the perforated production pipe. When the drilling is finished, the production begins.

Sometimes, it is difficult to pinpoint accurately the position of the reservoir. In such cases, directional drilling is preferred. It is also useful if the wells are distributed in a given area. Figure 2.1 shows the overview of the drill procedure with some of the rock layers found inside the earth crust and components used during drilling.

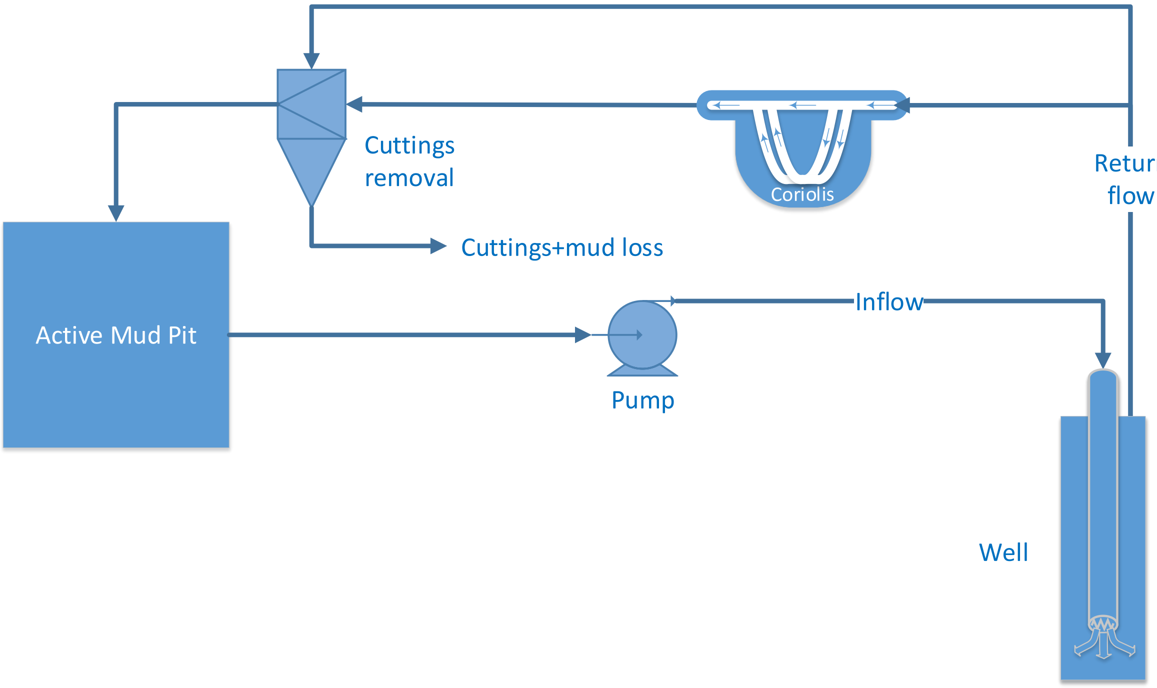

Figure 2.2: Closed loop drilling system with current flow measurement technique

2.2 Drilling mud and its importance

In the previous section, discussion about why the special fluid called drilling mud restricted to one function, remove the rock chipping formed during drilling. However, it is not only the reason for using the mud. The other benefit of the drilling mud is to cool the drill bit while also keeping the drill bit lubricated.

As shown in Figure 2.1, there are multiple layers of rock that need to be drilled through. The chemical composition of various types of rock varies. Returning mud shows the different composition of rocks being drilled. Some of these rock components may contaminate the drilling mud and prevent it from functioning properly. Therefore, checking viscosity and density constantly (in real time) is vital to operate safely and properly. If the layer of salt is present, the mud would cease to do its job and collapses the well which prevents the drilling. To combat the mud contamination, different types of mud needs to be circulated. Sometimes, a high-pressure gas encountered during drilling flows into the well and force out the mud which increases the return flow rate thus causing the blow out. The solution is to use the mud with higher density that counteracts the pressure exerted by the gas or other fluid in the well. Broadly speaking, higher density fluid maintains the pressure in the well as the mud exerts a hydrostatic pressure against the well wall.

To lift out the cuttings, mud must be highly viscous when it returns. The drawback of highly viscous fluid is that it is difficult to pump them into the well. Mud engineers wondered if only there exists a fluid whose viscosity changes. The inflowing fluid is required to be lightly viscous at high velocity making it easier to pump in and the returning fluid is highly viscous at low velocity making it able to take out the cuttings. Luckily, such fluid exists and are discussed in section 2.3.

2.3 Fluid types

Fluids can be classified into two main categories, Newtonian and non-Newtonian fluids. For incompressible fluids, the density, the heaviness of the fluid, remains constant. However, the density is not enough to uniquely characterize the behavior of the fluid. As the fluid start to flow, its behavior changes [4, p. 14]. To describe this behavioral change, the other important rheological property of the fluid can be used, namely viscosity. In simple words, viscosity is the resistance of a fluid to flow. On a technical term, viscosity can be defined using the Newtons law of viscosity as given by equations (2.1) and (2.2).

$$\tau = \mu\dot{\gamma} \tag{2.1}$$

$$\mu = \frac{\tau}{\dot{\gamma}} \tag{2.2}$$

Where, τ is the shearing stress, μ is the apparent viscosity of the fluid and $\dot{\gamma}$ is the rate of shearing strain (or simply, shear rate).

In one hand, if the shearing stress is linearly related to the rate of shearing strain, then the fluid is Newtonian. The apparent viscosity is constant for the Newtonian fluid. Most of the fluids found in the nature are Newtonian fluids. Water, air and crude oil are some examples of such fluid. For the equation (2.1) and (2.2) to be valid for Newtonian fluid, the apparent viscosity must be constant. For Newtonian fluid, increase in shear rate implies that the shearing stress must have been increased to keep the apparent viscosity constant [4, p. 16]. On the other hand, if the shearing stress is nonlinearly related to the shear rate, then the fluid is non-Newtonian. This suggests that the apparent viscosity of the non-Newtonian fluid varies with varying shear rate. A mixture of corn-starch and water is an example of non-Newtonian fluid.

Not all non-Newtonian fluids have same characteristics. Some act as solid with applied stress while other act more fluid. Thus, non-Newtonian fluids are further grouped into two classes; shear thinning, and shear thickening fluids based on the nonlinearity of the apparent viscosity. If the apparent viscosity increases with increasing shear rate, the fluid is shear thickening type as shown in Figure 2.3. On the contrary, if the apparent viscosity decreases with increasing shear rate, the fluid is shear thinning type. An ideal fluid for the drilling is the latter type.

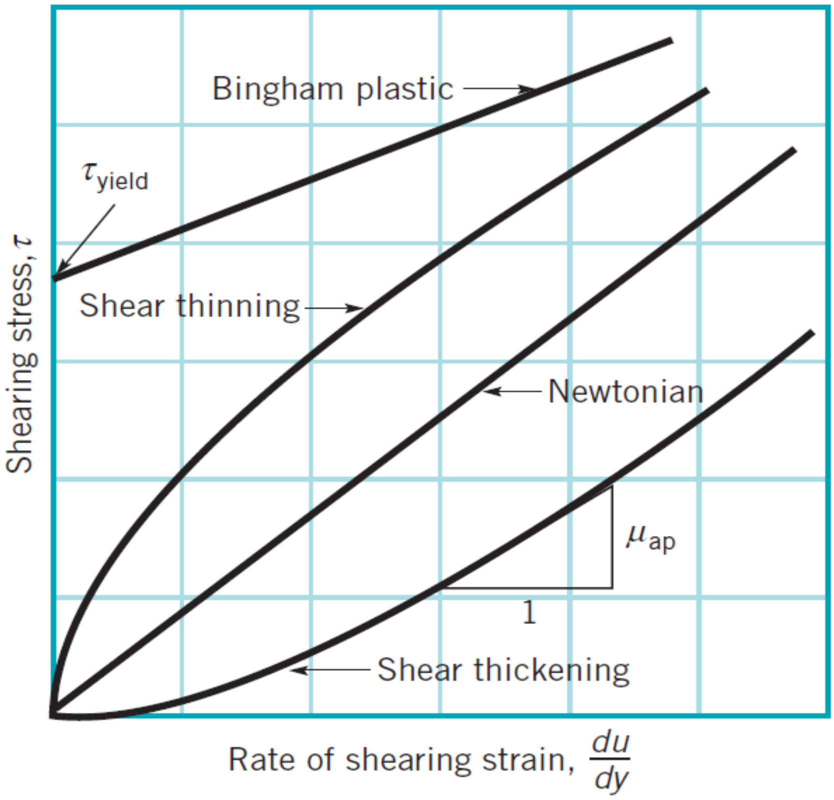

Figure 2.3: Property of the apparent viscosity shown as a relation between shearing stress and rate of shearing strain for different types of fluid. (Copied figure 1.7 from [4, p. 16]).

It is also important to discuss the relation between the fluid velocity, shear rate and shear stress. shows how these quantities profile develops at different radial section of the pipe (not along the pipe but at a given cross-section). In terms of the velocity and shear rate, shear thinning fluid is a type of fluid whose shear rate decreases with increasing fluid velocity. Such fluid is a good solution the mud engineers have been looking for.

In this thesis work, a shear thinning non-Newtonian fluid is used. The fluid is marked fluid1 at the lab at USN. It is a solution of potation carbonate (K_2CO_3) mixed with water. The solution has a density of $1340\,\frac{kg}{m^3}$.

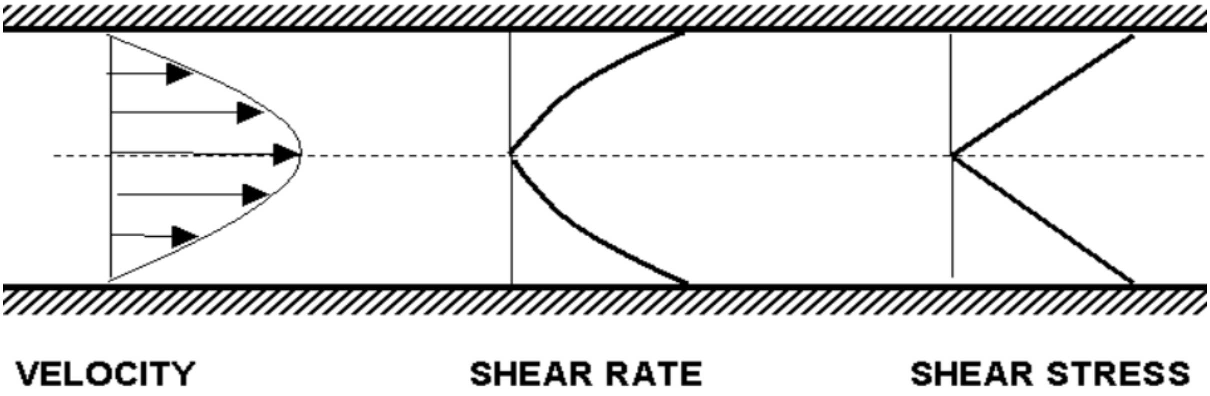

Figure 2.4: Velocity, rate of shearing strain (shear rate) and shear stress profile for a fluid flowing through a pipe [5].

2.4 Kick – loss detection

One of the most challenging part in an oil well drilling operation is maintaining the bottom hole pressure to prevent a blow-out. Accurate measurement of the returning flow is, thus, vital for a safe operation and for maintaining the bottom hole pressure. If the returning flow rate is greater than the injected flow rate, then probably some fluid (oil or gas or other substance) from the reservoir or surroundings has entered the well. This phenomenon is referred to as kick. Conversely, if the returning flow rate is less than the injected flow rate, then probably some mud has leaked into the reservoir. This phenomenon is referred to as a loss. In both cases, it can be an early warning of imbalance pressure in the bottom hole. One particularly disturbing problem with loss is that the mud might react with the fluid in the reservoir. This could block the well bore by forming a protective bung. This decreases the productivity of the well significantly or maybe altogether. Hence, it is important to detect kick-loss phenomenon as early as possible and deal with the situation. For this purpose, continuous flow measurement holds a significant importance for proper drilling operation. In the case of kick-warning, the heavier mud (with high density) should be pumped in to the well to counteract the pressure from the surrounding fluid.

3 Flow measuring technique

An introduction existing flow measuring technique as well as the model-based estimation technique is discussed here. Detailed analysis on design and the structure of the open channel is presented here. As discussed in section 1.3, it is vital to meet a condition defined by equation (10.10). To comply with that requirement, understanding the flow regime along the different section of the channel is also important. This is discussed in this chapter as well.

3.1 Current technique

In conventional drilling operation, paddle flow meter is used. Such flow meter measures the angle of the paddle introduced by the flowing fluid. Based on the measured angle, the volumetric flow rate is estimated. Such technique is unreliable since the flow is not uniform and thus the angle measurement is noisy. Therefore, the accuracy of the estimated return flow is severely undermined. A static flow measuring technique such as level measurement using a trip-tank is also used today. The level measurements are then used to determine the volumetric flow rate. This method is slow and unreliable.

A reliable flow measuring technique is to use Coriolis flow meter as shown in Figure 2.2. In managed drilling process, Coriolis is used. The returning mud is sent through the Coriolis flow meter, which estimates the flow of the returning fluid. The problem of such technique is that these flow meters are expensive to purchase, install and operate. In addition to that, these are mechanical devices that requires constant maintenance which is time consuming and expensive. Since the returning fluid is filled with rock cuttings and likely corrosive chemicals, these devices are prone to corrosion and internal damage. In worst case, the Coriolis flow meter may fail to estimate the flow or give unreliable estimates due to blockage internally. The blockage can be caused by the rock cuttings that stuck in the Coriolis pipes. This means that they may not last as long as their lifespan. Replacing them requires manual manpower and extended downtime for drilling. Time and again, they need to be recalibrated for proper functioning. To fulfill these tasks, more budget is required. Another shortcoming of Coriolis flow meter is that it fails to give reliable flow estimate in the presence of gas. Coriolis effect is best suitable for fluid flow and not for gas flow measurement. In oil well, gas pockets are likely to exist in and around the oil well. The gas trapped in these pockets is released when drilled through. The gas in return flow makes the Coriolis flow meter unreliable. Therefore, it makes sense to find a cheaper and effective solution. This thesis work hopes to pave a way to the future of oil drilling operation with a cheaper and effective solution then the existing one.

3.2 Proposed technique

The returning drilling mud is sent through an open venturi channel instead of the Coriolis flow meter. The levels are measured using either ultrasonic level sensor or radar level transmitter at two specific positions c_1 and c_2 as discussed briefly in section 1.3. Using the level measurements at these positions, the flow rate is estimated. The complete system loop with proposed flow measuring technique is shown in Figure 3.1. Coriolis flow meter in Figure 2.2 is replaced with an open venturi channel in Figure 3.1. The proposed technique is based on the mathematical model of the top flow in open venturi channel. Due to the dynamic nature of the model, the technique is dynamic as well. More importantly, this technique is not restricted to estimating the returning mud flow rate. Anyplace where the flow rate is measured using Coriolis meter or other flow measuring mechanical devices, this method is handy. Other important aspect of this technique is that the model can be used to design a suitable model predictive controller, MPC.

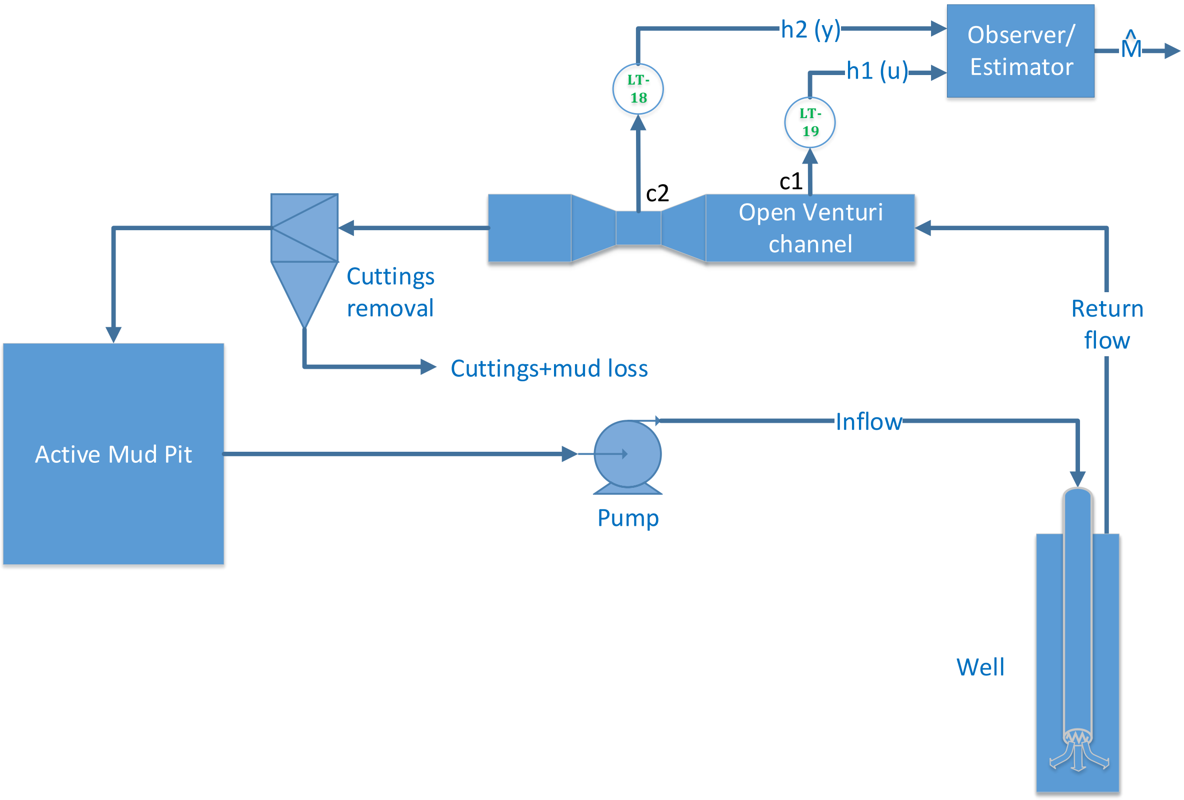

Figure 3.1: Closed loop drilling system with proposed flow measurement technique

3.3 Structure of the venturi flume

A venturi flume present at the lab at USN is a bisymmetrical trapezoidal flume pivot at the centroid of the throat section is shown in Figure 3.2. This apparatus is a nonprismatic channel, which means that the flow through it is non-uniform due to the change in the fluid velocity at different cross-sections of the channel. The change in the velocity is caused by the change in the cross-sectional area that is perpendicular to the direction of flow. There are five designated sections along the channel. Table 3.1 shows the classification of the different sections of the channel. Section 2 – section 4 is bisymmetric. p_0 is the reference point for length measurement and starts at 0 cm. The apparatus is a three-dimensional object. It is trapezoidal vertically at each point along the length of the channel as shown in Figure 3.3 (left). Figure 3.2 shows the top view (view on the x-y plane) of the entire channel. However, this view only reflects the schematic of the base width along the channel. Top width is determined by the fluid level and hence is not included in the top-view. Figure 3.3 (left) shows the cross-sectional view (viewed on the y-z plane) of the flume. Figure 3.3 (right) shows the longitudinal view (viewed on the x-z plane) of the flume. c_1 and c_2 are the point along the channel where the fluid level is measured using suitable level sensors. At these points, the mass flow rate is estimated.

Table 3.1: Different sections of the venturi flume with a corresponding shape.

Section #	1 $(p_0 - p_1)$	2 $(p_1 - p_2)$	3 $(p_2 - p_3)$	4 $(p_3 - p_4)$	5 $(p_4 - end)$
Type	Upstream	Converging	Throat	Diverging	Downstream
Shape	Rectangular	Trapezoidal	Rectangular	Trapezoidal	Rectangular
Range (cm)	0 – 132	132 – 147	147 – 167	167 – 182	182+

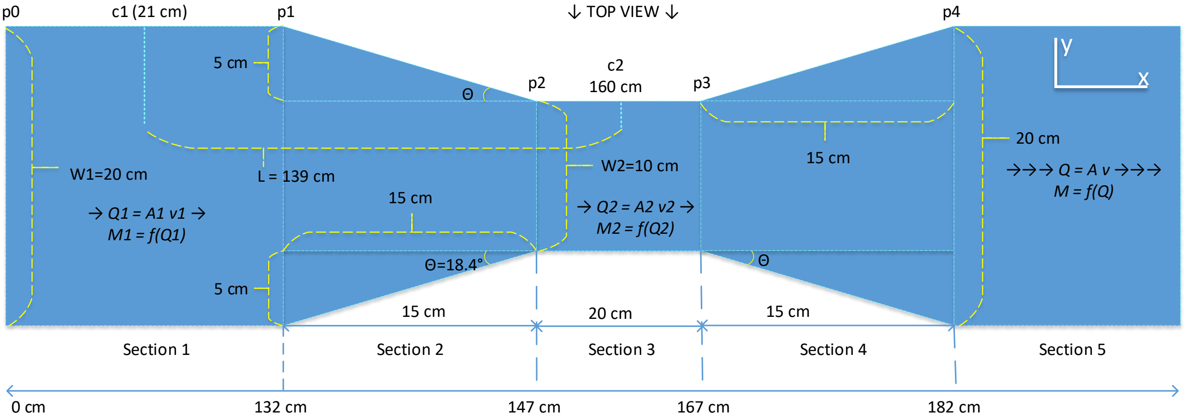

Figure 3.2: Top view of the bisymmetrical trapezoidal open venturi channel

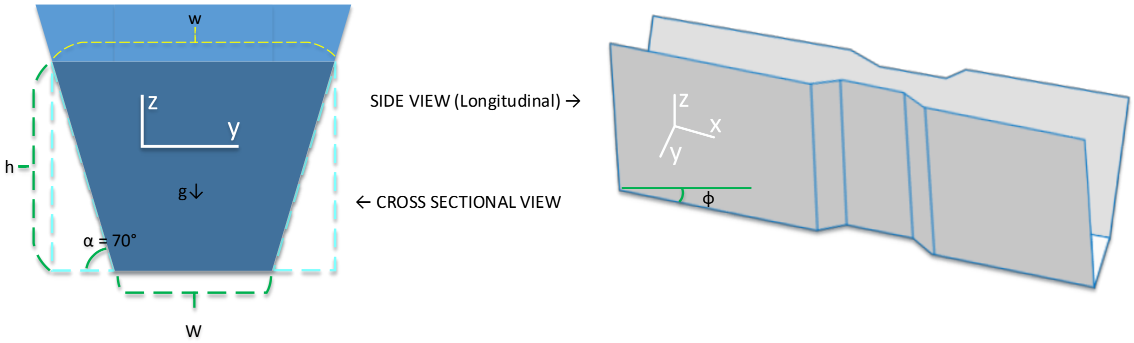

Figure 3.3: Cross-sectional and side (longitudinal) view of the venturi flume

3.3.1 Flow regime

A fluid flowing through open venturi channel has different velocity at different cross-section along the channel. Depending on the average velocity of the flow profile, flows can be classified into three types, critical, subcritical and supercritical flows. Flow through any arbitrary point is given by the Cross-sectional (wetted) area and the velocity of the fluid at the point as described by equation (3.1). Cross-sectional area is dependent on the fluid level. Assuming a steady flow condition, the velocity of the flow is inversely proportional to the wetted area as given by equation (3.2). Critical velocity occurs at the critical depth, which is a point along the channel where the specific energy of the fluid is minimum [6, pp. 530-532]. If the wetted are is larger than the critical wetted area, then the flow velocity is less than the critical velocity. Such flow is subcritical. Similarly, if the wetted are is smaller than the critical wetted area, the flow velocity is greater than the critical velocity. Such flow is supercritical. These conditions are presented in Table 3.2.

$$Q = A \cdot v \tag{3.1}$$

$$v = \frac{Q}{A}, \qquad v_c = \frac{Q}{A_c}, \qquad A = f(h) \tag{3.2}$$

$$Q = \frac{dV}{dt} = \frac{d\frac{m}{\rho}}{dt} = \frac{1}{\rho}\frac{dm}{dT} = \frac{M}{\rho} \tag{3.3}$$

Where, Q is the volumetric flow rate, A is cross-sectional area, v is the flow velocity at a given point. v_c and A_c are critical velocity and critical area at the critical point of the channel respectively. V is the volume of the fluid, m is the mass of the fluid and M is the mass flow rate. Equation (3.3) shows the relation between mass flow rate and volume flow rate.

Table 3.2: Different types of flow regime as classified by the flow velocity in an open channel

Flow condition	Subcritical	Critical	Supercritical
Wetted area	$A > A_c$	$A = A_c$	$A < A_c$
Flow velocity	$v < v_c$	$v = v_c$	$v > v_c$

In an open venturi channel setup as that at the lab at USN, as shown in Figure 3.2, critical flow usually occurs at the throat section (section 3) and the subcritical flow occurs at section 1 and section 2. Super-critical flow occurs at the section 4. This is true when the bed slope (angle made by the channel bed with the horizontal x-axis) is 0 and the mass flow rate of the fluid is less than 10 $\frac{kg}{s}$. This information is vital in choosing the position for level measurements as discussed in section 1.3; to satisfy the condition given by (10.10). Figure 3.4 illustrates the different flow conditions. Velocity and area line is used to illustrate how velocity and wetted cross-sectional area changes along the channel. Thickness of the line quantifies the velocity, while the slant quantifies the wetted area.

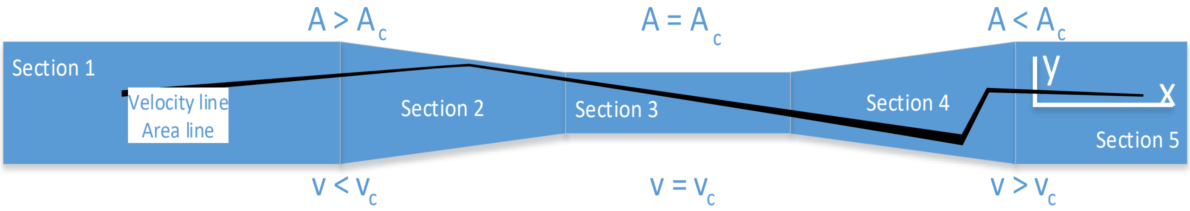

Figure 3.4: Development of the fluid velocity and wetted area across the open venturi channel

3.4 Experimental setup at the lab

Non-Newtonian fluid made by mixing potassium carbonate and water is stored in a primary tank just like the mud pit shown in Figure 3.1. The fluid passes through a buffer tank and enters the venturi channel. Ultrasonic and radar level transmitters are mounted over the channel. These sensors can be moved freely in the horizontal direction to measure the flow at a chosen position. The discharged fluid then returns to a secondary tank that is connected to the primary tank by a valve. There are different sensors mounted along the flow line to measure different process variables such as temperature, pressure, level and flow rate. There is also a Coriolis flow meter mounted along the inflow line. This device can measure mass flow rate and density. The Measurement by this device is, however, only used for the comparison purpose. Figure 3.5 shows the P&ID of the setup. The important sensors for this thesis work are level transmitters and Coriolis flow meter. Therefore, to simplify the diagram, only these sensors along with actuators such as pump are shown. A pump is controlled using a PI controller. A list of instruments, their symbol and the type devices used at the lab at USN are presented in Table 3.3. The list corresponds to the P&ID shown in Figure 3.5.

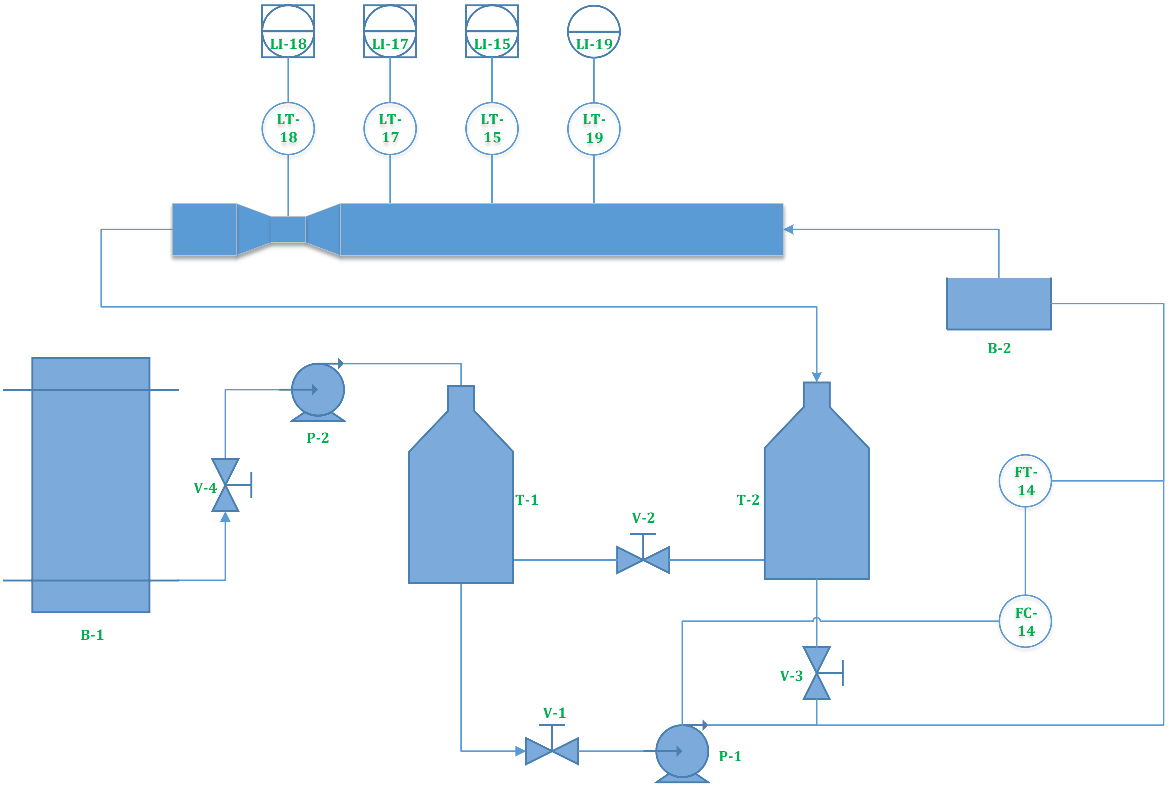

Figure 3.5: P&ID of the experimental setup at USN lab

Table 3.3: List of sensors, controllers, actuators and other equipment present at the USN lab.

Instrument	Symbol	Type
Radar level transmitter	LT19	Sensor
Ultrasonic level transmitters	LT15	Sensor

	LT17 LT18	
Coriolis transmitter (old)	FT14A – for mass flow rate FT14B – for density	Sensor
Flow controller	FC14	Controller, PI
Main pump	P-1	Actuator, pump
Fluid replacement pump	P-2	Actuator, pump
Buffer storage tank	B-1	Tank
Intermediate buffer tank	B-2	Tank
Primary storage tank	T-1	Tank
Secondary storage tank	T-2	Tank
Main valve	V-1	Actuator, Valve
Discharge valve	V-2	Actuator, Valve
Connection valve	V-3	Actuator, Valve
Fluid replacement valve	V-4	Actuator, Valve
Level indicator for ultrasonic sensors present in the front panel	LI15 LI17 LI18	Indicator
Level indicator for the radar sensor, mounted on the instrument	LI19	Indicator

4 Modeling of flow through an open venturi channel

By this stage, all the recipe for developing a mathematical model of the flow through open channel is in place. This chapter discusses the mathematical modeling of the flow through an open venturi channel. Development of the model as a partial differential equation (PDEs) and converting them into ordinary differential equations (ODEs) is discussed here briefly. In addition to that, the linearization of the nonlinear ODEs is explained in detail in this chapter.

4.1 Saint Venant Equation

During transient (change in flow rate), the flow through nonprismatic open channel becomes unsteady. This means that the fluid properties such as pressure, velocity, wetted area changes with spatial variable and time. Such dynamics can be explained by so called shallow water equations. These equations of continuity and motion are derived by using an idea of mass and momentum balance. Shallow water equations in one dimension are called Saint-Venant Equations (SVEs). SVEs in for one spatial dimension are a class of quasi-linear hyperbolic PDE. For the classification of PDEs, please refer [7, pp. 5-7].

4.1.1 Model for nonprismatic, 1D unsteady, open channel flow

Since the open channel in this case has one entry and one exit for the volumetric discharge, the lateral flow rate and the fluid loss is none. The fluid is incompressible and flows in one direction. Considering these facts and the pressure distribution is hydrostatic, following set of PDEs, as given by (4.1) and (4.2), are obtained. These equations are deduced at USN by fellow PHD students. The derivation of these equations is discussed in detail here [8]. Equation (4.1) is the continuity equation in conservation form. This statement simply means that the volume of fluid flowing through the channel is conserved (neither loss or gain). Equation (4.2) and (4.3) are the momentum equations.

$$\frac{\partial A}{\partial t}+\frac{\partial Q}{\partial x}=0 \tag{4.1}$$

$$\frac{\partial Q}{\partial t}+\frac{\partial}{\partial x}\left(\beta\frac{Q^2}{A}+gI_1\cos(\phi)\right)=gA\left(\sin(\phi)-S_f\right)+gI_2 \tag{4.2}$$

For small bed slope angle ϕ, $\cos(\phi)=1$. Replace $\sin(\phi)$ with S_b and rewrite the equation (4.2) as:

$$\frac{\partial Q}{\partial t}+\frac{\partial}{\partial x}\left(\beta\frac{Q^2}{A}+gI_1\right)=gA\left(S_b-S_f\right)+gI_2 \tag{4.3}$$

Expressions for some variables present in equations (4.1) – (4.3) are given below.

$$I_1=h^2\left(\frac{W}{2}+h\frac{S_s}{3}\right) \tag{4.4}$$

$$I_2 = h^2\left(\frac{1}{2}\frac{dW}{dx} + \frac{h}{3}\frac{dS_s}{dx}\right) \tag{4.5}$$

The side wall of the open channel, in this case, has a uniform side slope i.e., $\frac{dS_s}{dx} = 0$. Substituting this in equation (4.5), the result becomes equation (4.6).

$$I_2 = \frac{h^2}{2}\frac{dW}{dx} \tag{4.6}$$

$$S_s = \cot(\alpha) \tag{4.7}$$

$$A = (S_s h + W)h \tag{4.8}$$

$$D = 2S_s h + W \tag{4.9}$$

$$P = W + 2h\sqrt{1 + S_s^2} \tag{4.10}$$

$$R = \frac{A}{P} \tag{4.11}$$

$$n_m = \frac{1}{k_s} \tag{4.12}$$

$$S_f = \frac{Q|Q|n_M^2 P^{\frac{4}{3}}}{A^{\frac{10}{3}}} \quad \Leftrightarrow \quad S_f = \frac{Q|Q|n_M^2}{A^2 R^{\frac{4}{3}}} \tag{4.13}$$

$$h = \frac{-W + \sqrt{W^2 + 4AS_s}}{2S_s} \tag{4.14}$$

4.2 Nonlinear ordinary differential equations

Equations (4.1) and (4.3) are a set of PDEs. However, the state estimator and observer that is being designed during this thesis work is based on the ordinary differential equations. Hence, the PDEs must be represented by a set of ODEs before proceeding. A method used to accomplish this is called the orthogonal collocation method.

4.2.1 Orthogonal collocation method

States are the variables that describe the dynamics of the system. Simply stated, states are the variables that are differentiated with respect to time. Here the states are wetted cross-sectional area (A) that is perpendicular to the discharge direction (x-direction) and the volumetric flow rate (Q) as given by equations (4.1) and (4.3). It is possible to approximate these states at chosen collocation points in spatial domain by using interpolation technique. Lagrange interpolating technique is used for this purpose [9, pp. 809-812]. Collocation points are the roots of the interpolating polynomial in spatial domain. Hence, it is safe to say that these points describe the degree of the polynomial that interpolates the original function. Degree of polynomial defines the accuracy of the approximation. It is also required that the level at collocation point 1 (h_1) must be higher than the level at collocation point 2 (h_2) as given by (1.1). For this purpose, the first collocation point, c_1 is chosen at the upstream section (sub-critical flow region) and the second point, c_2 is chosen to be at the throat (critical-flow region) as shown in Figure 3.2. The number of collocation points used here is two (quadratic polynomial with two roots). A fellow PHD student at USN has produced a set of ODEs for nonprismatic channel for two collocation points [10]. Since there are two states, approximating them at two points yields four ODEs. The set of ODEs is given by equations (4.15) to (4.18).

$$\dot{A}_1 = -\frac{1}{L}(-Q_1 + Q_2) \tag{4.15}$$

$$\dot{A}_2 = -\frac{1}{L}(-Q_1 + Q_2) \tag{4.16}$$

$$\dot{Q}_1 = -\frac{\beta}{L}\left(-\frac{Q_1^2}{A_1} + \frac{Q_2^2}{A_2}\right) - \frac{g}{L}\left(-I_{1_1} + I_{1_2}\right) + \frac{gh_1^2}{2L}(-W_1 + W_2) + gA_1\left(S_b - S_{f_1}\right) \tag{4.17}$$

$$\dot{Q}_2 = -\frac{\beta}{L}\left(-\frac{Q_1^2}{A_1} + \frac{Q_2^2}{A_2}\right) - \frac{g}{L}\left(-I_{1_1} + I_{1_2}\right) + \frac{gh_2^2}{2L}(-W_1 + W_2) + gA_2\left(S_b - S_{f_2}\right) \tag{4.18}$$

Where subscript 1 corresponds to the collocation point 1. Similarly, subscript 2 corresponds to the collocation point 2. ODEs (4.15) - (4.18) describes the $AQ\ model$.

4.2.2 Change of variable

The set of ODEs represented by equations (4.15) to (4.18) can be simplified further by a simple change of variable. The new variables are chosen based on the actual measurands of the real process at the lab at USN. These variables are the level (h) measured using ultrasonic and radar sensor and mass flow rate (M) measured using Coriolis flow meter. This simplification makes it easier to linearize the model as well as implement them in a programming language such as MATLAB and LabVIEW. Another benefit, from the programming point of view, is that there is less number of unit conversions to perform. This increases the performance of the simulator, observer or estimator. Otherwise, the level measurements must be converted to cross-sectional area and back. This overhead is reduced by variable change. One discovery made is that the model represented using the new set of variables, h and M, is more robust during the initial run. Since the operating points for the system with A and Q as states, depend on the model parameters, their uncertainty affects the proper estimates of A and Q mathematically as they are not measured directly. Default parameter values are presented in Table 4.1. However, some of parameter values need to be adapted and optimize for the different operating conditions. More about this is discussed in later section.

Table 4.1: Default parameter values for model with changed variables, h and M

Params (θ)	$\alpha\ [deg]$	$\beta\ [\,]$	$ks\ \left[\frac{\sqrt[3]{m}}{s}\right]$	$L\ [m]$	$\phi\ [deg]$	$\rho\ \left[\frac{kg}{m^3}\right]$	$W_1\ [m]$	$W_2\ [m]$
Values	*70*	*0.66*	*56*	*1.39*	*0*	*1340*	*0.2*	*0.1*

The procedure on how to change the state variable and express the ODEs in terms of new variables is presented in detail in Appendix 1. The new ODEs are given by equations (4.19) to (4.22) in terms of h and M as shown in Part 1A. The model based on these ODEs is referred to as $hM\ model$.

$$\dot{h}_1 = \frac{(M_1 - M_2)}{\rho L D_1} \tag{4.19}$$

$$\dot{h}_2 = \frac{(M_1 - M_2)}{\rho L D_2} \tag{4.20}$$

$$\dot{M}_1 = \frac{\beta}{\rho L}\left(\frac{M_1^2}{A_1} - \frac{M_2^2}{A_2}\right) + \frac{\rho g}{L}\left(I_{1_1} - I_{1_2}\right) - \frac{\rho g h_1^2}{2L}(W_1 - W_2) + \rho g S_b A_1 - \frac{g}{\rho}\frac{M_1|M_1|n_m^2 P_1^{\frac{4}{3}}}{A_1^{\frac{7}{3}}} \tag{4.21}$$

$$\dot{M}_2 = \frac{\beta}{\rho L}\left(\frac{M_1^2}{A_1} - \frac{M_2^2}{A_2}\right) + \frac{\rho g}{L}\left(I_{1_1} - I_{1_2}\right) - \frac{\rho g h_2^2}{2L}(W_1 - W_2) + \rho g S_b A_2 - \frac{g}{\rho}\frac{M_2|M_2|n_m^2 P_2^{\frac{4}{3}}}{A_2^{\frac{7}{3}}} \tag{4.22}$$

Equations (4.4) to (4.14) are generic functions for different intermediate variables, meaning that these equations define the dynamics of the model at any given point. For two collocation points, these equations can be extended to form specific expressions as shown in Table 4.2.

Table 4.2: Equations for different parameters at two different collocation points

Collocation point 1, c_1 (upstream section)	Collocation point 2, c_2 (throat section)
$h_1 = \frac{-W_1 + \sqrt{W_1^2 + 4A_1S_s}}{2S_s}$	$h_2 = \frac{-W_2 + \sqrt{W_2^2 + 4A_2S_s}}{2S_s}$
$P_1 = W_1 + 2h_1\sqrt{1 + S_s^2}$	$P_2 = W_2 + 2h_2\sqrt{1 + S_s^2}$
$A_1 = (S_s h_1 + W_1)h_1$	$A_2 = (S_s h_2 + W_2)h_2$
$S_{f_1} = \frac{Q_1\|Q_1\|n_M^2 P_1^{\frac{4}{3}}}{A_1^{\frac{10}{3}}}$	$S_{f_2} = \frac{Q_2\|Q_2\|n_M^2 P_2^{\frac{4}{3}}}{A_2^{\frac{10}{3}}}$
$I_{1_1} = h_1^2\left(\frac{W_1}{2} + h_1\frac{S_s}{3}\right)$	$I_{1_2} = h_2^2\left(\frac{W_2}{2} + h_2\frac{S_s}{3}\right)$

$D_1 = 2S_s h_1 + W_1$	$D_2 = 2S_s h_2 + W_2$

4.3 Linearization of the venturi model

So far, the equation of continuity and motion as a set of hyperbolic PDEs have been derived. Using orthogonal collocation method, the PDEs have been reduced to a set of nonlinear ODEs. Nonlinear ODEs are used for nonlinear estimator, but a linear model is needed for designing linear observers and estimators. To find a linear model, it is important to define the inputs, states and the outputs of the system. There are four ODEs given by equations (4.19) – (4.22). Three of them are used to describe the system dynamics and one of the them is used as an input. Table 4.3 show the assignment of function names to the ODEs and define their type as input, states or output of the system. For model linearization, Tylor series expansion around a suitable linearization (operating) point is used. A general method of linearization of a multivariate nonlinear model is given in Appendix 2.

Table 4.3: Assigning function name for the ODEs derived by change of variable.

Equation #	(4.19)	(4.20)	(4.21)	(4.22)	
Function	$f_1 = \dot{h}_1$	$f_2 = \dot{h}_2$	$f_3 = \dot{M}_1$	$f_4 = \dot{M}_2$	$g = h_2$
Type	Input	State 1	State 2	State 3	Output

Based on Table 4.3, the venturi model is a deterministic system with one input (u), three states (x) and one output (y). The system is written in following a compact and standard form:

$$u = h_1, \qquad u_{op} = h_{1_{op}}$$

$$x = \begin{bmatrix} h_2 \\ M_1 \\ M_2 \end{bmatrix}, \qquad \dot{x} = \begin{bmatrix} \dot{h}_2 \\ \dot{M}_1 \\ \dot{M}_2 \end{bmatrix}, \qquad x_{op} = \begin{bmatrix} h_{2_{op}} \\ M_{1_{op}} \\ M_{2_{op}} \end{bmatrix}, \qquad \dot{x}_{\text{op}} = \begin{bmatrix} \dot{h}_{2_{op}} \\ \dot{M}_{1_{op}} \\ \dot{M}_{2_{op}} \end{bmatrix}$$

$$\dot{x} = f(x, u, \theta, t) \tag{4.23}$$

$$y = h_2, \qquad y_{op} = h_{2_{op}}$$

$$n_u = 1 \quad n_x = 3 \quad n_y = 1$$

Where θ is the model parameter vector, n_u represents the number of inputs, n_x represents the number of states, n_y represents the number of outputs. The simplest approach to find the operating point is to simulate the model until the steady state is reached rather than using a mathematical equation for operating point that is complicated to solve analytically. Note the steady state values as the values for operating point. However, even better way is to use the real measurement data. Operating points for some flow rates based on the real measurements logged at the lab are given in Table 4.4 and shown graphically in Figure 4.1. Figure 4.1 also shows the filtered operating point filtered using a median filter (more on filters in section 9.2). Operating point for area and volumetric flow are calculated using (4.8) and (a19) respectively.

Table 4.4: Operating value for input, states and output at different flow rates for a fluid with a density of $\rho = 1340\,\frac{kg}{m^3}$. Default operating value for the mass flow rate is $350\,\frac{kg}{min}$

$M_{op}\left[\frac{kg}{min}\right]$	$Q_{op}\left[\frac{m^3}{s}\right]$	$h_{1op}\ [mm]$	$A_{1op}\ [m^2]$	$h_{2op}\ [mm]$	$A_{2op}\ [m^2]$
275	*3.4 e-3*	*59.9*	*1.3 e-2*	*44.1*	*5.5 e-3*
350 (default)	*4.4 e-3*	*69.1*	*1.6 e-2*	*49.4*	*6.3 e-3*
425	*5.3 e-3*	*78.4*	*1.8 e-2*	*57.0*	*7.4 e-3*

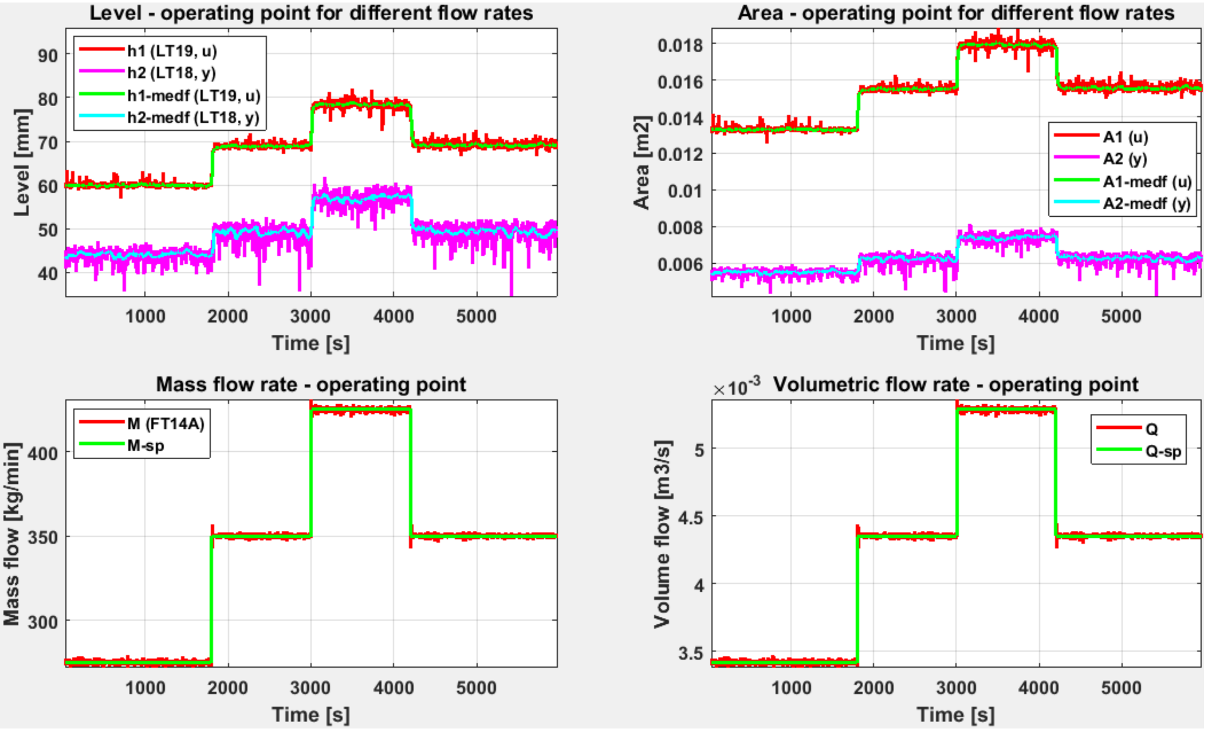

Figure 4.1: Operating point for h, A, M and Q for a fluid with a density of $\rho = 1340\,\frac{kg}{m^3}$

4.3.1 Linear state space model

There is not direct feedthrough term in the measurement equation $y = h_2$. This simply means that the output equation is not directly influenced by the input. The deterministic linear state space model is, thus, given by equations (4.24) and (4.25). Refer Appendix 2 for the complete derivation of the state space model including the declaration of the system matrices. The deviation variables are presented in (a2).

$$\delta\dot{x} = A_c\delta x + B_c\delta u \tag{4.24}$$

$$\delta y = C_c\delta x \tag{4.25}$$

Where A_c, B_c and C_c are system matrices derived from linearizing the nonlinear model. The states and outputs in full form are computed using (4.26) and (10.10) respectively.

$$x = \delta x + x_{op} \tag{4.26}$$

$$y = \delta y + y_{op} \tag{4.27}$$

4.3.2 System matrices

According to the Table 4.3, the system is a single-input-single-output (SISO) system with three states. Number of states defines the order of the system. Hence, the system order (n_x) is 3 for venturi model. Figure 4.2 visualizes the dimension of system (Jacobian) matrices. These matrices are time invariant (independent of time).

Table 4.5: The size of the system matrices

A_c	B_c	C_c
$[n_x \times n_x] = [3 \times 3]$	$[n_x \times n_u] = [3 \times 1]$	$[n_y \times n_x] = [1 \times 3]$

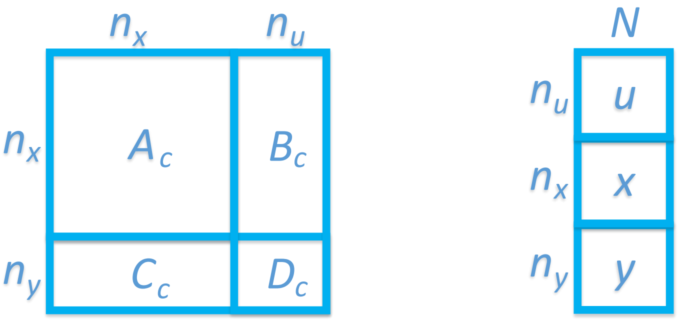

Figure 4.2: Dimension of the system matrices and input, states and output matrices or vectors

Where, N is the number of samples of inputs, states and outputs. The size of the system matrices applicable for this venturi model is given in Table 4.5. The system matrices, also known as the Jacobian matrices, are represented in following form:

$$A_c = \left.\frac{\partial f}{\partial x}\right|_{x_{op},u_{op}} \quad B_c = \left.\frac{\partial f}{\partial u}\right|_{x_{op},u_{op}} \quad C_c = \left.\frac{\partial g}{\partial x}\right|_{x_{op},u_{op}}$$

The derivation of these matrices is shown in detail in Appendix 2 (a3). For venturi model, f is a vector function given by (4.28) and g is a scalar function given by (4.29).

$$f = \begin{bmatrix} f_2 \\ f_3 \\ f_4 \end{bmatrix} = \begin{bmatrix} \dot{h}_2 \\ \dot{M}_1 \\ \dot{M}_2 \end{bmatrix} = \dot{x} \tag{4.28}$$

$$g = h_2 \tag{4.29}$$

$$A_c = \begin{bmatrix} \frac{\partial f_2}{\partial h_2} & \frac{\partial f_2}{\partial M_1} & \frac{\partial f_2}{\partial M_2} \\ \frac{\partial f_3}{\partial h_2} & \frac{\partial f_3}{\partial M_1} & \frac{\partial f_3}{\partial M_2} \\ \frac{\partial f_4}{\partial h_2} & \frac{\partial f_4}{\partial M_1} & \frac{\partial f_4}{\partial M_2} \end{bmatrix}, \qquad B_c = \begin{bmatrix} \frac{\partial f_2}{\partial h_1} \\ \frac{\partial f_3}{\partial h_1} \\ \frac{\partial f_4}{\partial h_1} \end{bmatrix}, \qquad C_c = \begin{bmatrix} \frac{\partial g}{\partial h_2} & \frac{\partial g}{\partial M_1} & \frac{\partial g}{\partial M_2} \end{bmatrix}$$

Where,

$$\frac{\partial f_2}{\partial h_2} = \frac{-2S_s(M_1 - M_2)}{\rho L D_2^2} \qquad A_c(1,1)$$

$$\frac{\partial f_3}{\partial h_2} = \frac{\beta D_2 M_2^2}{\rho L A_2^2} - \frac{\rho g A_2}{L} \qquad A_c(2,1)$$

$$\begin{aligned} \frac{\partial f_4}{\partial h_2} = \frac{\beta D_2 M_2^2}{\rho L A_2^2} - \frac{\rho g A_2}{L} - \frac{\rho g h_2 (W_1 - W_2)}{L} + \rho g S_b D_2 \\ - \frac{g M_2 |M_2| n_m^2 P_2^{\frac{1}{3}}}{3\rho A_2^{\frac{7}{3}}} \left(8\sqrt{1 + S_s^2} - \frac{7 P_2 D_2}{A_2} \right) \end{aligned} \qquad A_c(3,1)$$

$$\frac{\partial f_2}{\partial M_1} = \frac{1}{\rho L D_2} \qquad A_c(1,2)$$

$$\frac{\partial f_3}{\partial M_1} = \frac{2\beta M_1}{\rho L A_1} - \frac{2 g M_1^2 n_m^2 P_1^{\frac{4}{3}}}{\rho |M_1| A_1^{\frac{7}{3}}} \qquad A_c(2,2)$$

$$\frac{\partial f_4}{\partial M_1} = \frac{2\beta M_1}{\rho L A_1} \qquad A_c(3,2)$$

$$\frac{\partial f_2}{\partial M_2} = \frac{-1}{\rho L D_2} \qquad A_c(1,3)$$

$$\frac{\partial f_3}{\partial M_2} = \frac{-2\beta M_2}{\rho L A_2} \qquad A_c(2,3)$$

$$\frac{\partial f_4}{\partial M_2}=\frac{-2\beta M_2}{\rho L A_2}-\frac{2gM_2^2 n_m^2 P_2^{\frac{4}{3}}}{\rho|M_2|A_2^{\frac{7}{3}}}$$

$A_c(3,3)$

$$\frac{\partial f_2}{\partial h_1}=0$$

$B_c(1,1)$

$$\frac{\partial f_3}{\partial h_1}=\frac{-\beta D_1 M_1^2}{\rho L A_1^2}+\frac{\rho g A_1}{L}-\frac{\rho g h_1(W_1-W_2)}{L}+\rho g S_b D_1$$
$$-\frac{gM_1|M_1|n_m^2 P_1^{\frac{1}{3}}}{3\rho A_1^{\frac{7}{3}}}\left(8\sqrt{1+S_s^2}-\frac{7P_1D_1}{A_1}\right)$$

$B_c(2,1)$

$$\frac{\partial f_4}{\partial h_1}=\frac{-\beta D_1 M_1^2}{\rho L A_1^2}+\frac{\rho g A_1}{L}$$

$B_c(3,1)$

$$\frac{\partial g}{\partial h_2}=1$$

$C_c(1,1)$

$$\frac{\partial g}{\partial M_1}=0$$

$C_c(1,2)$

$$\frac{\partial g}{\partial M_2}=0$$

$C_c(1,3)$

Some of the axioms and rules of differentiation used during linearization of the nonlinear ODEs are given under Appendix 5. Alternative forms of the system matrices for venturi ODEs for different variables are given under Appendix 3.

5 Model analysis and simulation

Now that the nonlinear model and linear model are developed, the next step is, naturally, to simulate the model. This chapter deals with simulator development in detail. It is also important to analyze the stability of the model and parameter sensitivity. The model is finally implemented in MATLAB and LabVIEW.

5.1 Model analysis

A model needs to be analyzed for its stability, controllability and observability before designing any control system or observers. All these aspects are important in control engineering which are discussed briefly in this section.

5.1.1 Analytical solution

A system can be deterministic or stochastic system. On one hand, a deterministic system does not account for any randomness. A deterministic system always produces a specific final state from a specific initial state for a given input signal. On the other hand, a stochastic system accounts for randomness, be it with process noise or measurement noise or both. A general form of a stochastic state space model with direct feedthrough term in the output is given by equations (5.1) and (5.2).

$$\delta\dot{x} = A_c\delta x + B_c\delta u + Ew \tag{5.1}$$

$$\delta y = C_c\delta x + D_c\delta u + Fv \tag{5.2}$$

Where, w is the process noise and v is the measurement noise. If matrices E and F are zeros-matrices, the system become deterministic. Analytical solution to the state equation (5.1) is given by equation (5.3) and the solution to the output equation (5.2) is given by (5.4). These solutions are for special case of $t_0 = 0$ as derived in Part 6A of Appendix 6.

$$\delta x(t) = \Phi(t)\delta x(0) + \int_0^t \Phi(t-\tau)B_c\delta u(\tau)d\tau + \int_0^t \Phi(t-\tau)Ew(\tau)d\tau \tag{5.3}$$

$$\begin{aligned}\delta y(t) = C_c\Phi(t)\delta x(0) + C_c\int_0^t \Phi(t-\tau)B_c\delta u(\tau)d\tau + C_c\int_0^t \Phi(t-\tau)Ew(\tau)d\tau \\ + D_c\delta u + Fv\end{aligned} \tag{5.4}$$

Where $\Phi(t)$ is a time invariant transition matrix for LTI systems and is given by (a12). For a deterministic system, the solution (5.3) takes a special form given by (5.5). The output of the system without the direct feedthrough term is given by (5.6).

$$\delta x(t) = \Phi(t)\delta x(0) + \int_0^t \Phi(t-\tau)B_c\delta u(\tau)d\tau \tag{5.5}$$

$$\delta y(t) = C_c\Phi(t)\delta x(0) + C_c\int_0^t \Phi(t-\tau)B_c\delta u(\tau)d\tau \tag{5.6}$$

Analytical solution for state equation given by (5.5) is complicated to solve since it contains a convolution integral. Hence, for solving the state equation (4.24), a numerical method, fourth order Runge-Kutta (RK4), is used in this project. See Appendix 4 for RK4 algorithm.

5.1.2 Stability

Stability of a model is an important concept in mathematical modeling. A stable system is defined as a system with a specific bounded response to a specific bounded input [11, p. 387]. To design a controller, the closed-loop system must be stable. An unstable system is of little use. There are many approaches to analyze the stability of a dynamic system. Methods such as eigenvalues and pole-zero analysis, Routh-Hurwitz stability criterion, frequency response method such as bode stability criterion and Nyquist stability criterion and so on. The discussion in this section focuses on the eigenvalues and pole-zero analysis.

Poles of a controllable and observable system are the eigenvalues of the transition matrix A_c. A system can be asymptotically stable, marginally stable or unstable. Asymptotic stability means that a system comes back to equilibrium point even after an introduction of large disturbance or input. A marginally stable system neither converges to nor diverges from but oscillates around the equilibrium point. An unstable system, on the other hand, goes to infinity even for a small finite input change. In terms of eigenvalues, an asymptotically stable system has all its eigenvalues in the left side of complex plane. For marginally stable system, the eigenvalues lie on the imaginary plane. For unstable system, eigenvalues lie on the right side of the complex plane. Figure 5.1 show the different conditions of stability using poles plotted on the complex plane. Illustration is based on a system with three states.

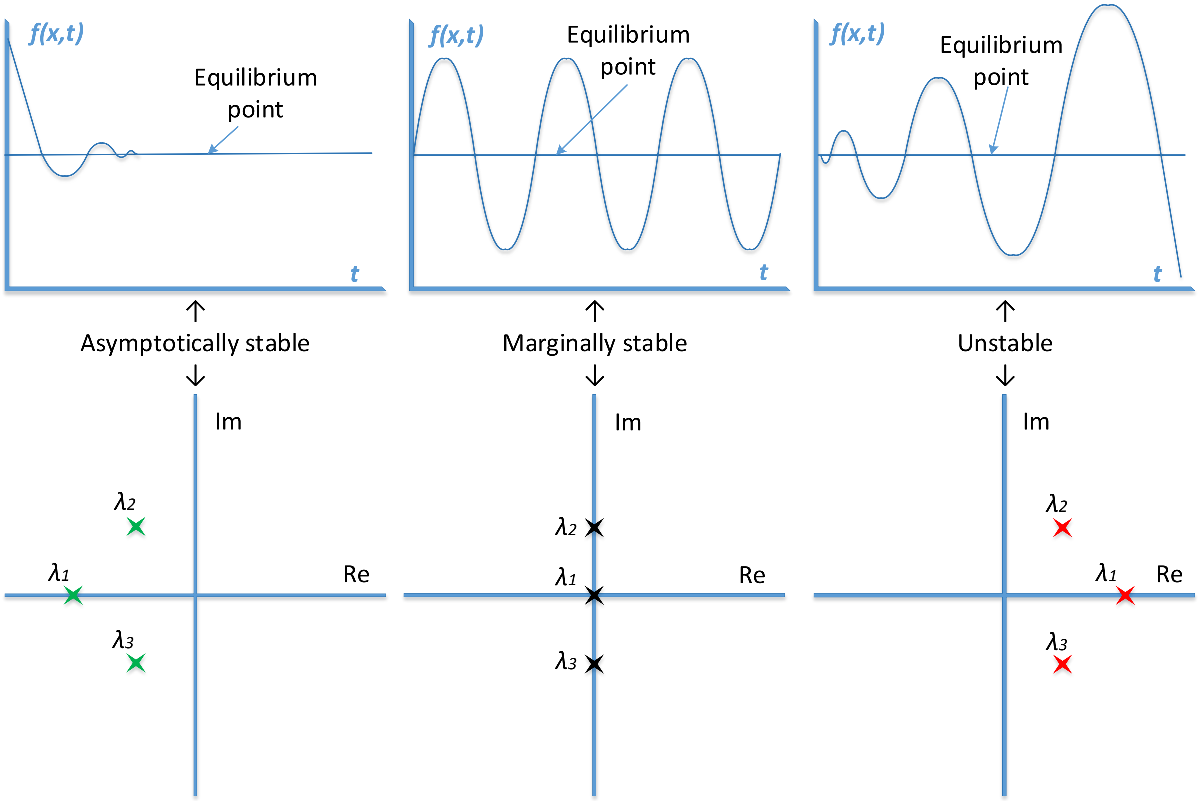

Figure 5.1: System stability in terms of poles (eigenvalues) of the system

For a system, following condition defines its stability:

- If all the eigenvalues of the system matrix A_c lie in the left side of the complex plane, the system is stable. In other words, for stable system, the real part of all the eigenvalues is negative as shown in Figure 5.1 (left).

$$real\big(eig(A_c)\big) < 0 \rightarrow stable\ system$$

- If the eigenvalues of the system matrix A_c lie on the imaginary axis of the complex plane, the system is marginally stable. In other words, for marginally stable system, the real part of the eigenvalues must be zero as shown in Figure 5.1 (center).

$$real\big(eig(A_c)\big) = 0 \rightarrow marginally\ stable\ system$$

- If the eigenvalues of the system matrix A_c lie in the left side of the complex plane, the system is unstable. In other words, for unstable system, the real part of the eigenvalues is positive as shown in Figure 5.1 (right).

$$real\big(eig(A_c)\big) > 0 \rightarrow unstable\ system$$

5.1.3 Controllability

It is important to identify if a system is controllable before designing a control system. Controllability aids in answering this question. Controllability is defined as an ability of system to transit from a specific initial state to a desired final state for an appropriate input signal. A system is controllable if the rank of the controllability matrix is the same as the number of states in the system. In other words, if the system has n-states, the controllability matrix must have n-independent column that spans the entire vector space. Simply stated, if the determinant of the controllability matrix is non-zero, then it is a full ranked matrix and the system is controllable. The controllability matrix is given by (5.7). Controllability of a continuous system is given by (5.8).

$$\Delta_\mathrm{c} = [B_c \quad A_cB_c \quad A_c^2B_c \quad \cdots \quad A_c^{n-1}B_c] \tag{5.7}$$

$$rank(\Delta_\mathrm{c}) = n \tag{5.8}$$

Where, n is the order of system.

5.1.4 Observability

The idea of this thesis is to design a suitable observer or an estimator. This can only be done if the system is observable. Determination of system observability is a reverse process. If the initial states can be predicted from the knowledge of past inputs and outputs, then the system is said to be observable. This, as with the controllability, can be determined using the rank of the observability matrix. If the rank of observability matrix is the same as the number of states in the system. In other words, if the system has n states, the observability matrix must have n-independent column that spans the entire vector space. If the observability matrix has a non-zero determinant, then it is a full ranked matrix and the system is observable. The observability matrix is given by (5.9). Observability of a continuous system is given by (5.10).

$$O_\mathrm{c} = [C_c^T \quad (C_cA_c)^T \quad (C_cA_c^2)^T \quad \cdots \quad (C_cA_c^{n-1})^T]^T \tag{5.9}$$

$$rank(\Phi_c) = n \tag{5.10}$$

Where, n is the order of system.

5.1.5 Stability, controllability and observability of venturi system

System stability for the venturi model is determined using the "eig.m" function in MATLAB. Controllability matrix of the venturi model is determined using "ctrb.m" function. The observability of the model is determined using "obsv.m" function. The rank is determined using "rank.m" function. The outcome is that both controllability and observability matrices are full ranked i.e., rank is equal to the system order. In this case, the system order is three. For the default parameter values as shown in Table 4.1, following system matrices are generated:

$$A_c = \begin{bmatrix} 0 & 0.0038 & -0.0038 \\ 0.3372 & 0.1438 & -0.7327 \\ 16.1390 & 0.2255 & -1.3128 \end{bmatrix}, \qquad B_c = \begin{bmatrix} 0 \\ 101.2385 \\ 164.0757 \end{bmatrix}, \qquad C_c = [1 \quad 0 \quad 0]$$

Eigenvalues (λ) of A_c of the venturi model is given by:

$$\begin{array}{lllll} \lambda_1 & = & -1.1299 & + & 0 & i \\ \lambda_2 & = & -0.0195 & + & 0.2686 & i, \\ \lambda_3 & = & -0.0195 & - & 0.2686 & i \end{array} \qquad \lambda_2 \xLeftrightarrow{complex\ conjugate} \lambda_3$$

$$real(\lambda_j) < 0 \rightarrow stable\ system, \qquad j = 1,2,\ldots,n$$

The controllability and observability matrices are given by:

$$\Delta_c = \begin{bmatrix} 0 & -0.2380 & 0.3291 \\ 101.2385 & -105.6634 & 25.8213 \\ 164.0757 & -192.5647 & 220.0777 \end{bmatrix}, \qquad O_c = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.0038 & -0.0038 \\ -0.1402 & -0.00031 & -0.0022 \end{bmatrix}$$

$$rank(\Delta_c) = rank(O_c) = 3 = number\ of\ states = system\ order$$

This shows that the system is both controllable and observable. The system is also stable as confirmed by Figure 5.2. This result is in par with the illustration of Figure 5.1

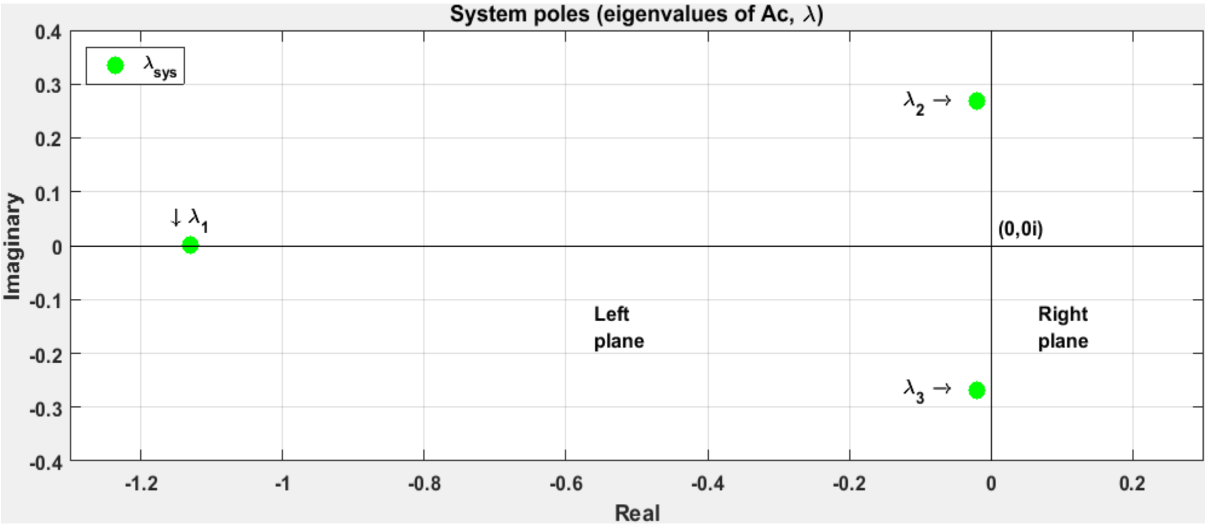

Figure 5.2: System poles (eigenvalues) for venturi model plotted on the complex plane

5.2 Parameter sensitivity analysis

The venturi model has several input parameters, as shown in Table 4.1, which defines the model output. Some of them are more influential than other. By influential, it means that the change in the input causes the change in the output. The change in the output is highly significant for some parameters than the others. Determination of which parameter is more influential and which parameter is less, is known as sensitivity analysis of the parameter. Sensitivity analysis is important to determine which parameter contributes most to the model uncertainty as the parameter uncertainty are propagated into the model. Such analysis is also a test to determine insignificant parameters that can be removed from the model [12]. To perform sensitivity analysis, it is important to define a probability density for each input factors. The probability density function for the uniform distribution is shown in Figure 5.3. Nominal values and uniform distribution and range for each input parameter is given in Table 5.1. The parameters are uniformly distributed around ±10 % of their nominal values. For an input parameter ϕ, the nominal value is 0. Therefore, the distribution is specified randomly.

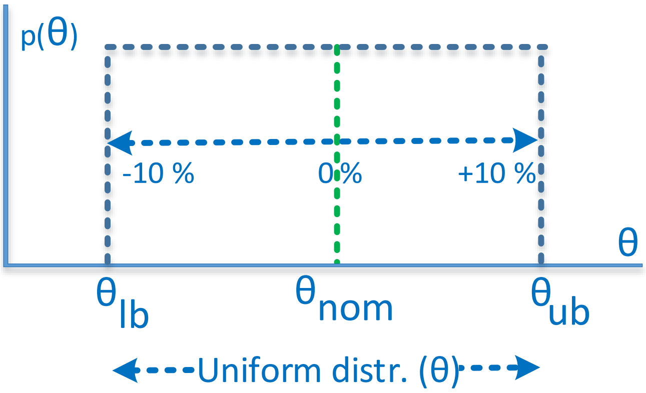

Figure 5.3: Uniform distribution around the nominal value for each parameter

Table 5.1: Nominal value, uniform distribution and range for each input parameter

Input Parameters (θ)	Nominal value (θ_n)	Distribution range (±10 %)		Unit
		Lower bound	Upper bound	
α	*70*	*63*	*77*	$[deg]$
β	*0.66*	*0.594*	*0.726*	$[-]$
h_1	*0.079*	*0.0711*	*0.0869*	$[m]$
k_s	*66*	*59.4*	*72.6*	$\left[\frac{\sqrt[3]{m}}{s}\right]$
L	*1.39*	*1.251*	*1.529*	$[m]$
ϕ	*0*	*-0.036*	*0.036*	$[deg]$

ρ	*1340*	*1206*	*1474*	$\left[\frac{kg}{m^3}\right]$
W_1	*0.2*	*0.18*	*0.22*	$[m]$
W_2	*0.1*	*0.9*	*0.11*	$[m]$

5.2.1 Differential sensitivity

Differential sensitivity determination technique is the most common and elegant way of defining the sensitivity of a parameter to the model output. The point here is to quantify how sensitive each parameter is to the model output, in this case, model states. The idea is to determine the sensitivity coefficient for each parameter and analyze them. The sensitivity coefficient is the ratio of the change in the output to the change in the input [12]. This coefficient is a measure of the percent change in the output to the percent change in the input as given by (5.11).

$$\varepsilon_j^i = \frac{\Delta x\%}{\Delta\theta\%} \quad \begin{matrix} i = 1,2,\dots,r \\ j = 1,2,\dots,n \end{matrix} \tag{5.11}$$

$$\varepsilon_j^i = \frac{\dfrac{x_j^i - x_0^i}{x_0^i} \times 100\%}{\dfrac{\theta_j^i - \theta_0^i}{\theta_0^i} \times 100\%}$$

$$\varepsilon_j^i = \left(\frac{x_j^i - x_0^i}{x_0^i}\right)\left(\frac{\theta_0^i}{\theta_j^i - \theta_0^i}\right)$$

$$\varepsilon_j^i = \left(\frac{x_j^i - x_0^i}{\theta_j^i - \theta_0^i}\right)\left(\frac{\theta_0^i}{x_0^i}\right) \tag{5.12}$$

Where, ε is the sensitivity coefficient and θ is an input parameter and x is the state. θ_0^i is the nominal parameter value and x_0^i is the nominal output. (r) is the number of input factors and (n) is the number of samples from distribution. Equation (5.12) gives the rate of change of output with respect to the parameter. ε, θ and x can be a scalar or a vector. For multivariable functions, (5.12) can be approximated by partial derivative of the output (states) with respect to the parameters as given by (5.13).

$$\varepsilon_j^i = \frac{\partial x^i}{\partial \theta^i}\left(\frac{\theta_0^i}{x_0^i}\right)\bigg|_j \tag{5.13}$$

The neat thing about the equation (5.13) is that the value of the coefficient is normalized by the ratio of nominal parameter and the nominal output value. This removes the effect of the parameter unit and the unit of output. Using the sensitivity coefficient for each parameter based on (n) samples and compute the mean (μ) and standard deviation (σ). The result is then plotted in a $\mu\sigma$ space for analysis. Steps taken for such analysis are listed below:

- Define an input parameter space by distributing each parameter around its nominal value and generate (n) samples.
    - As discussed earlier, uniform distribution is chosen here.
- Select a parameter, perform the following:
    - For a chosen parameter, keep other parameters to their nominal value
    - Select a sample from the distribution pool for the chosen parameter
    - Simulate the model for long enough time so that the steady state is reached
    - Note the steady state value for each output or state
    - Using (5.12), determine a sensitivity coefficient for the sample and store it
    - Repeat these steps for all the samples
    - From (n) coefficients, determine the mean and standard deviation and store it
- Repeat above steps for all parameters.
- By this stage there are (r) mean and standard deviation values for each output/state.
- Plot these in a $\mu\sigma$ space

One important thing to note about determining the sensitivity of an input parameter by differential sensitivity method, is that it only quantifies the local sensitivity. In other words, this method does not account for any potential interaction effect of the chosen parameter with the other input parameters. Simply stated, the effect is independent.

5.2.2 Sensitivity index

Another way to quantify the sensitivity of a parameter is by determining the sensitivity index. It is a measure of the ratio of the steady state model output space to the range of parameter space. For a parameter, (n) samples are generated using a probability distribution function. The model is then simulated until the steady state is achieved. The final steady state values are used to define the states space, i.e., the maximum variation in the states (x). This is shown graphically in Figure 5.4. The range is defined by the maximum and the minimum value reached by the output. Mathematically, the sensitivity index is given by equation (5.14).

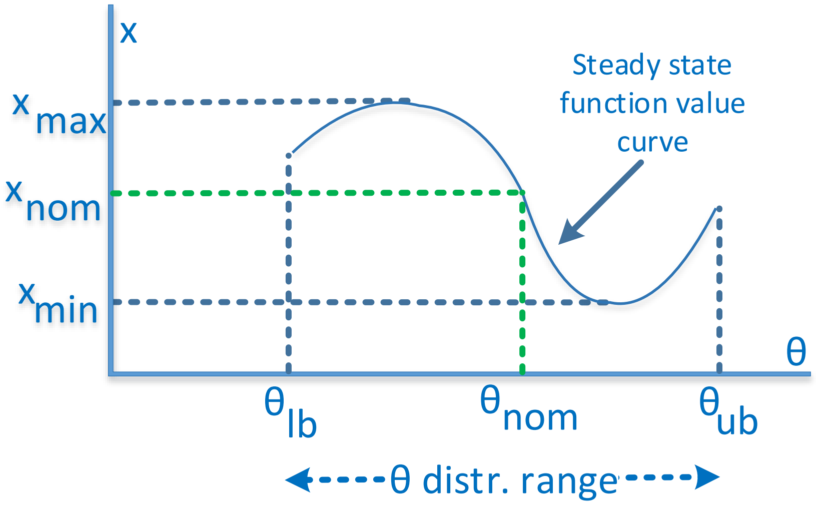

Figure 5.4: Sensitivity index as a ratio between output range to the input range

$$SI = \frac{x_{max} - x_{min}}{x_{max}} \tag{5.14}$$

5.2.3 Analysis of the venturi model parameters

In venturi model, there are eight model parameters and one input whose sensitivity on the model states (h_2, M_1, M_2) is to be determined. Table 5.1 shows these parameters and their nominal value along with the distribution range. Total 99 points are uniformly distributed around the nominal value of a parameter. Thus, the nonlinear venturi model is simulated 99 times. Equations (5.13) and (5.14) are used to compute sensitivity coefficients and sensitivity indices respectively. The results are discussed below. Figure 5.4 shows the mean and standard deviation plot of the sensitivity coefficient. The interpretation of $\mu\sigma$ plot gives rise to the following four conditions based on the magnitude, absolute value, of μ, σ:

1) Small μ and small σ
    - ➢ The parameter is less influential and is unimportant [13].
2) Large μ and small σ
    - ➢ The parameter is important, and the model output depends on it mostly linearly. Zero σ means that the parameter and the model output is fully linear [13].
3) Large μ and large σ
    - ➢ The parameter is important, and the model output depends on it highly nonlinearly. The parameter might also interact with the other parameters.
4) Small μ and large σ
    - ➢ The parameter is important, and the model output depends on it highly nonlinearly [14].

The bottom line is that if the μ is large, the parameter is highly important. If the σ is large, the parameter may be interacting with other parameters nonlinearly. Since there are three states, the interpretation is done individually. Figure 5.5, Figure 5.7 and Figure 5.6 are interpreted together. Figure 5.5 shows the effect of the parameter based on μ and σ of sensitivity coefficient. Figure 5.7 shows the sensitivity index for each parameter for all the states. Figure 5.6 shows the percent change in the outputs (states) influenced by the percent change in the parameter value.

- Level at the throat (collocation point 2) as state 1, (h_2)

From the Figure 5.5 (left subplot), the channel width at the collocation point 1, W_1 has the largest absolute value for μ and σ. Hence, W_1 is the most influential parameter for h_2. It may also be interacting with other model parameters. The effect of W_1 on h_2 is nonlinear. This interpretation is supported by Figure 5.6. Since all the parameters are distributed by the same amount ($\pm 10\%$) around their nominal values, the percent change in W_1 influences h_2 the most. The sensitivity index plot of Figure 5.7 cements this observation further as W_1 has the highest value. Similarly, he next influential parameter for h_2 is Strickler friction coefficient, k_s with relatively large μ and σ. The effect is nonlinear as displayed in Figure 5.6. Fluid level at the throat, W_2, on the other hand has large μ and average σ. This means that W_2 mildly nonlinear as seen in Figure 5.6. Fluid density, ρ is the least important factor for h_2 since it has the least value of μ and σ and least value for sensitivity index. It is also seen in Figure 5.6 that h_2 has no change for any value of ρ. Thus, ρ is the least influential parameter for h_2.

- Mass flow rate through the venturi channel as state 2 (M_1) and state 3 (M_2)

Similarly, M_2 is the most influential parameters for both M_1 and M_2. It has the highest value for μ, relatively large σ and largest sensitivity index. A particularly interesting parameter for flow rates is, h_1. It has second largest μ but extremely small σ (Figure 5.5). This means that h_1 an important parameter with large sensitivity index (Figure 5.7). However, due to small σ, the effect is fully linear. The linearity is clearly seen in Figure 5.6. The length between the collocation points (L) has less influence on (M_1) and (M_2) with comparatively small μ, σ and sensitivity index value. However, the least important parameter for the flow rates is channel bed angle, ϕ with least μ and σ values.

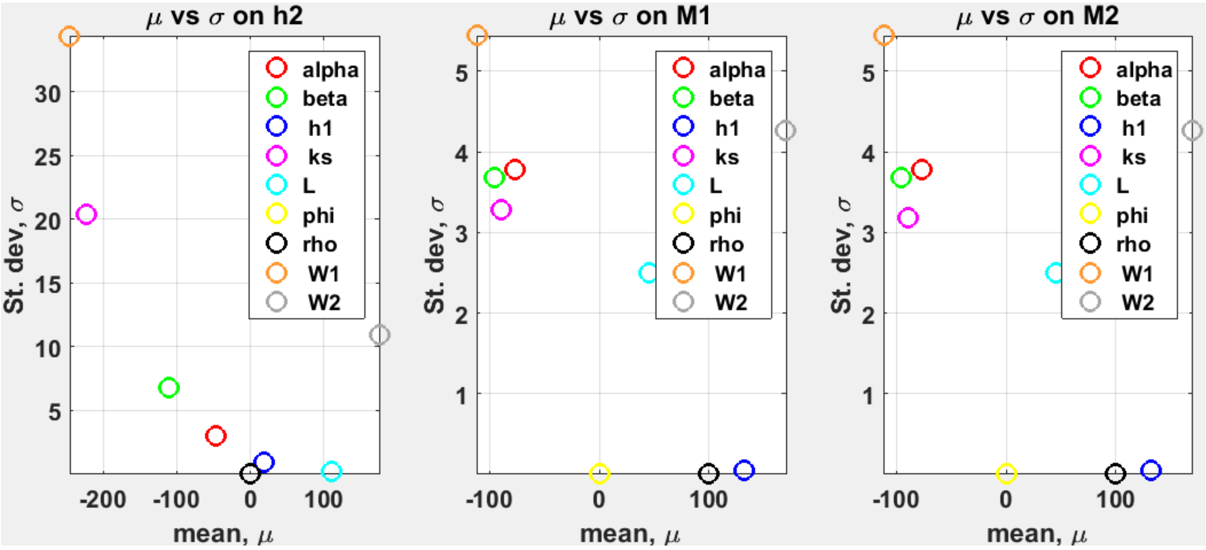

Figure 5.5: Mean and standard deviation of the sensitivity coefficient for the venturi model

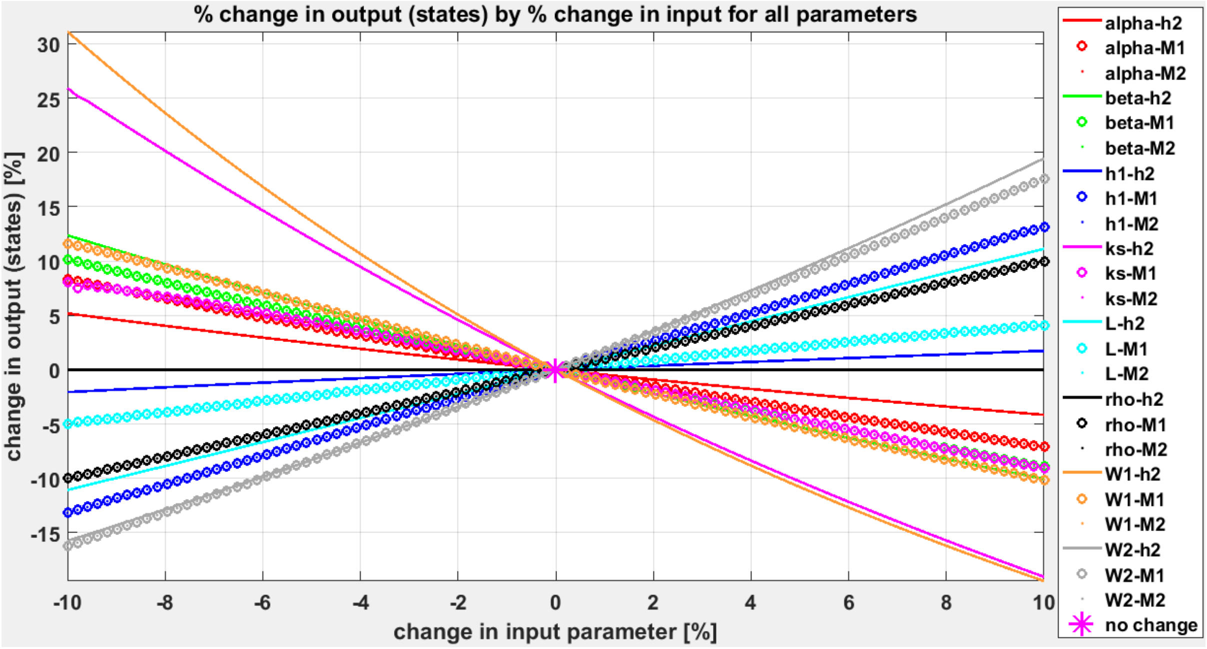

Figure 5.6: Percent change in the parameter vs the percent change in the output for all parameters for all states of the venturi model

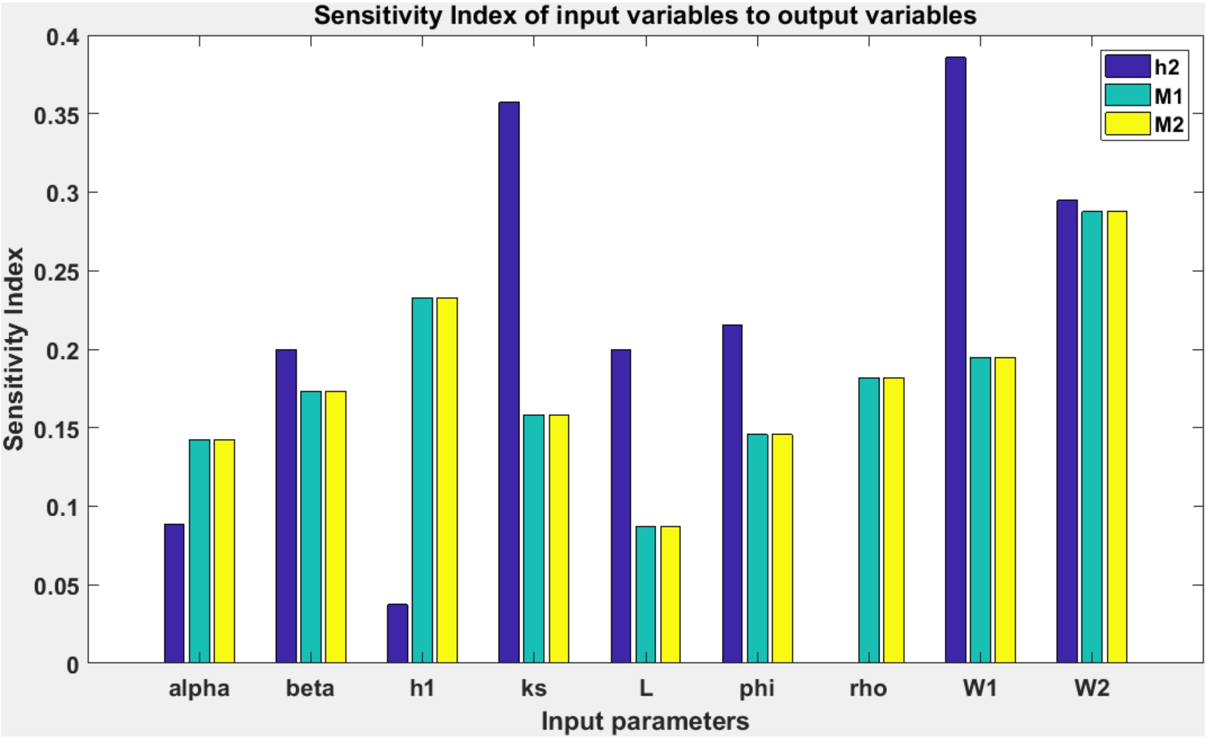

Figure 5.7: Sensitivity index for venturi model parameters to all states

In conclusion, W_1 is the most influential and ρ is the least influential parameter for h_2. W_2 is the most influential and ϕ is the least influential parameter for both M_1 and M_2. The complete summary of the analysis is presented in detail in Appendix 7 (Part 7A and Part 7B) for each state with additional relevant information.

5.2.4 Correlation

Derivative based sensitivity analysis is a great tool to analyze influential parameter in a model. However, it does not account the complete picture of the relation of the parameters with the model output. Correlation between the parameter and the states is, therefore, a great tool to view such relation in detail. Correlation is how two variables change with respect to each other. It shows the direction of change in output variables (positive or negative) with respect to the change in the model parameter. Correlation, however, does not specify by how much other variable change if one variable changes. Correlation value varies from -1 to 1. Interpretation of correlation is as follow:

- Correlation of -1:
  - Highly correlated
  - Negatively $(-ve)$ directed
    - As one variable increases, the other variable decreases by same rate
- Correlation of 0:
  - No correlation. Two variables are independent of each other
- Correlation of +1:
  - Highly correlated
  - Positively $(+ve)$ directed

- As one variable increases, the other variable increases as well, by the same rate

For venturi model, the correlation between each state and each parameter is determined based on the steady state values. Table 5.2 show the magnitude of correlation and the direction.

Table 5.2: Correlation between parameters and states

x, θ	α	β	h_1	k_s	L	ϕ	ρ	W_1	W_2
h_2	-0.998 $(-ve)$	-0.999 $(-ve)$	0.999 $(+ve)$	-0.997 $(-ve)$	*1* $(+ve)$	0.998 $(+ve)$	*0.152* (0)	-0.993 $(-ve)$	0.999 $(+ve)$
M_1	-0.999 $(-ve)$	-0.999 $(-ve)$	*1* $(+ve)$	-0.999 $(-ve)$	0.999 $(+ve)$	*1* $(+ve)$	*1* $(+ve)$	-0.999 $(-ve)$	*1* $(+ve)$
M_2	-0.999 $(-ve)$	-0.999 $(-ve)$	*1* $(+ve)$	-0.999 $(-ve)$	0.999 $(+ve)$	*1* $(+ve)$	*1* $(+ve)$	-0.999 $(-ve)$	*1* $(+ve)$

The result is consistent with the sensitivity analysis. To see the correlation between states and the parameters, refer Figure 5.6. The table shows large correlation between most of the parameters. This is because the model is not influenced with the noise. These values will alter, not drastically though, when working with the real measurements.

5.2.5 Parameter optimization

A mathematical model can only predict the output as accurately as the best measured or estimated parameters value. The model to depicts the real system accurately, determining the parameter values is important. An approach to determining the parameter values is by using hit and trial method. This can be done by simulating the model along with the real process in parallel and adjust the parameter values manually until satisfactory result is gained. Such methods are mind-numbingly tedious and time consuming. The better way is to minimize the error between the predicted and the real measurement. To do this, a cost function, also known as objective function is designed. The objective is to minimize the error between the predicted and real measurements. This can be done using sum-squared-prediction-error (SSPE) method. SSPE is given by (5.15).

$$\varepsilon = \sum_{i=1}^{N} e_i^T e_i \tag{5.15}$$

$$\min_{\theta} J = \sum_{i=1}^{N} e_i^T e_i \tag{5.16}$$

$$e_i = x_i - \bar{x}_i \quad [e_i] = [n_x \times 1] \tag{5.17}$$

Where, ε represents the SSPE that is being minimized. N is the number of measurements (samples) and e_i is the prediction error vector. x_i is the real measurement vector with one samples of each state at a given time i. $\bar{x}_i$ is the model predicted vector with one sample for each state. Since there are three states in the dynamic model of the venturi flume, e_i, x_i, $\bar{x}_i$ are column vectors with the same dimension of $[3 \times 1]$.

Predicted state vector x_i is the solution of the state equation (4.28) for the given time step (i). The state equation is discretized and solved by using RK4 (Appendix 4). The built-in optimization function, “fmincon.m”, of MATALB is then used to optimize the parameter. The result is given in Table 5.3. Using the optimal parameter values, the model is simulated to see how good the prediction becomes. Figure 5.8 shows the predicted vs actual measurement. SSPE for the new predicted states is 14.73, which is acceptable.

Table 5.3: Optimal parameter values determined by minimizing SSPE

$\theta \rightarrow$	α	β	k_s	L	ϕ	ρ	W_1	W_2
θ_{opt}	*70.7*	*0.65*	*52.7*	*1.4*	*-0.07*	*1353.4*	*0.199*	*0.119*

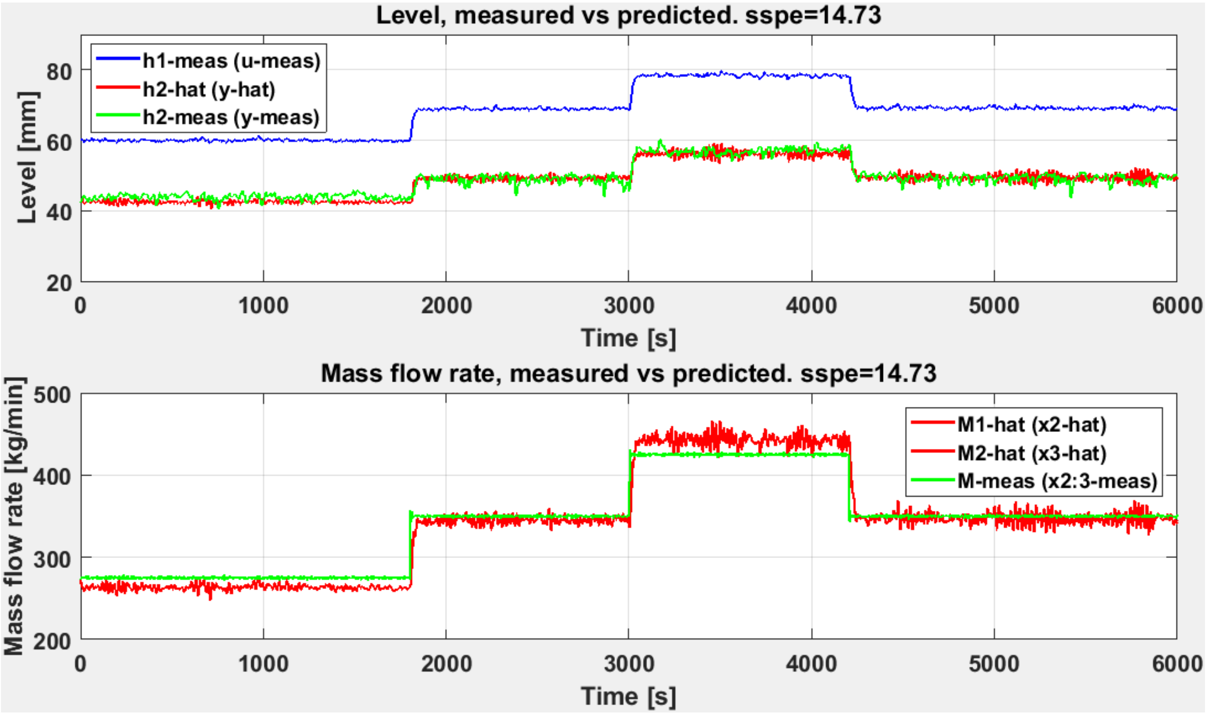

Figure 5.8: Predicted states using optimal parameter vs real measurement

5.3 Simulator development

The linear and nonlinear version of the venturi model is implemented in MATLAB. The simulation is open loop and the input (h_1) is controlled manually. The linear model is linearized around suitable operating point. The operating point is chosen based on the information from the real system. An operating point for input and each state in local and SI unit is given in Table

5.4. These values are determined based on Figure 4.1. Local units are used only for plotting purposes. During simulation, input, states, output and model parameters are converted to their SI units. For plotting purpose, the values are converted back to the local units.

Table 5.4: Operating point for input and each state for the Venturi flume based on the information from the real system

Type of unit	$M_{1_{op}} = M_{2_{op}}$	$h_{1_{op}}$	$h_{2_{op}}$
Local	$350 \left[\frac{kg}{min}\right]$	$80\ [mm]$	$45\ [mm]$
SI	$5.883 \left[\frac{kg}{s}\right]$	$0.08\ [m]$	$0.045\ [m]$

5.3.1 Comparison of $hM\ model$ and $AQ\ model$

Model based on level and mass flow rate-based is referred to as $hM\ model$ represented by (4.19) - (4.22) and the model based on area and volumetric flow rate is called the $AQ\ model$ represented by (4.14) - (4.18). The value of the model parameters for both nonlinear simulators is the same. The simulators are run in parallel in LabVIEW. Both simulators are excited by the same input (h_1) with added synthetic noise. The result is shown in Figure 5.9. In both cases, the outcome is exactly the same. However, implementing $hM\ model$ is simpler and easier than implementing $AQ\ model$. Hence, the $hM\ model$ is preferred for the rest of this thesis work.

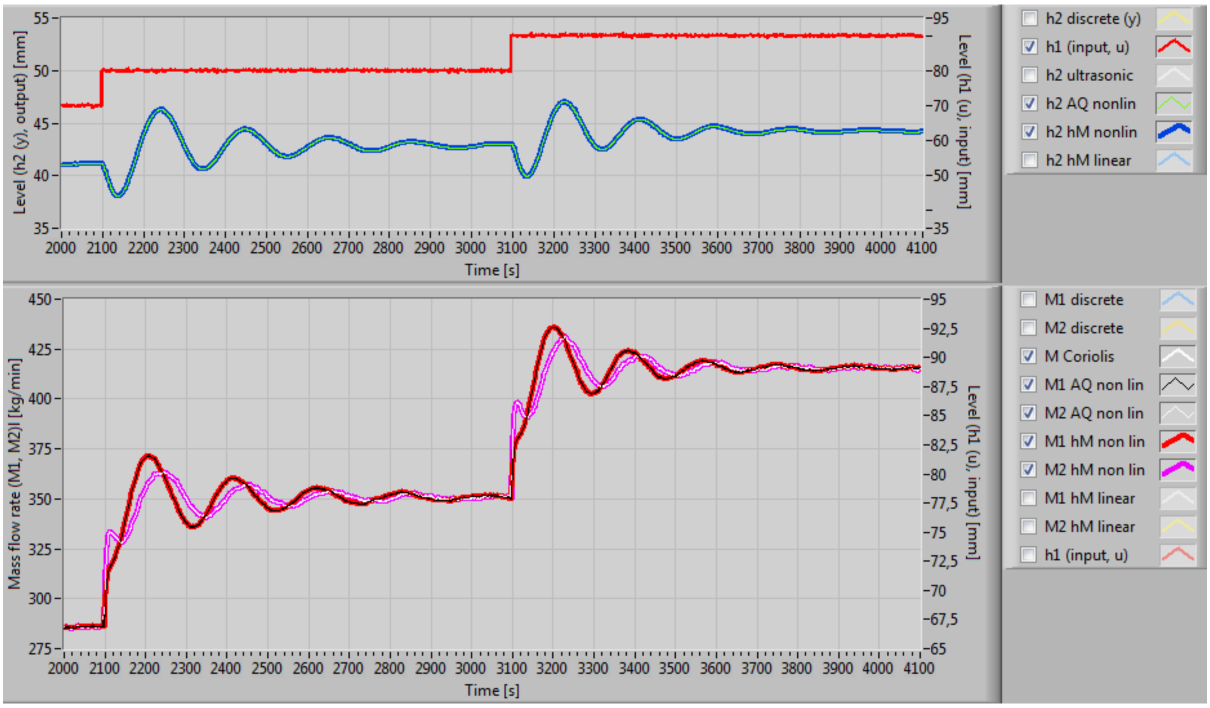

Figure 5.9: Simulation result of a model based on level (h) and the mass flow rate (M) along with a model based on area (A) and the volumetric flow rate (Q).

5.3.2 Nonlinear and linear model comparison

Linear and nonlinear models are simulated in parallel to compare how they fare alongside each other. The input is changed manually from the operating point and the system dynamics is observed. The result is displayed in Figure 5.10. First part of the simulation $(0\,s \leq t < 1200\,s)$ does not account for any noise in the input. The second part $(12000\,s \leq t \leq 2000\,s)$, the h_1 and h_2 are excited with noise signals w and v respectively. Since the input to the real system is level measurement, which are noisy, a synthetic noise is added to the simulator input to simulate the effect of noise on the model. The synthetic noise has the same variance, σ^2, as the measured input level measurements (h_1) taken from the venturi rig at the lab. The variance of the white noise processes used is 10^{-6}.

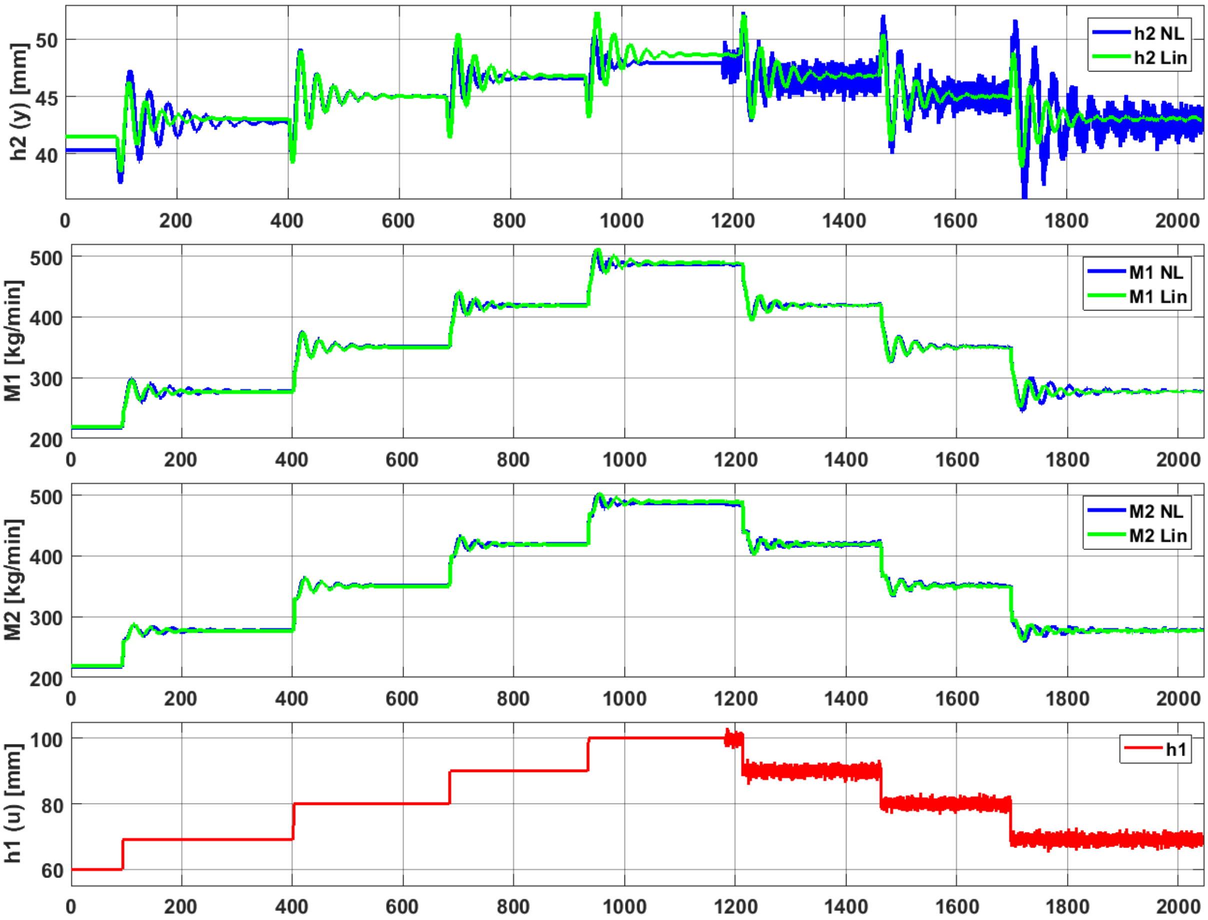

Figure 5.10: Linear vs nonlinear venturi model comparison.

As the simulation starts, noise-free input signal is away from the operating point (subplot 4). This introduces a deviation in the output (subplot 1). During linearization, the model is approximated by a series of polynomials around the operating point. The higher terms are neglected keeping only the linear first order terms. These higher order terms do carry some information of the model output. The removal of higher order term accounts for the deviation in the output. As the input is changed gradually towards the operating point, $h_1 = 80\,mm$, the deviation starts to disappear. When the noise is applied to the model, oscillation increases, and the amplitude of oscillation is also amplified. The noise affects all the states, including flow rates. The settling time for the states also increases due to the noise introduction. However, the mass flow rate deviates very little away from the linearization point. This could be because the model is developed using Lagrange interpolating polynomial at two points which produces linear ODE's. This means the model is a linear approximation of the PDE. Bottom line is that both models, linear and nonlinear, produce similar result around the operating point as the linear model is just an approximation of the nonlinear model around the linearization point.

SECTION – II

FLOW ESTIMATION

STATE ESTIMATION USING OBSERVERS AND KALMAN FILTERS

IMPLEMENTATION OF OBSERVER AND ESTIMATOR ON THE REAL PROCESS

6 Highway to state estimation

A basic idea of state estimation is discussed in this chapter briefly as a bridge between model simulation and state estimation. The aim is to cast light on why state estimation is required. The focus is set on importance of state estimation based on the oil well drilling operation.

6.1 Purpose of state estimation

State information of a process is useful in many ways. They can be used for designing a control system, analyzing the process for safety and productivity and predict the future behavior of the system. For this, all states must be measured. However, not all process variables are measured on a real process. This could be due to lack of available sensors or simply due to the cost of the sensors being too expensive to implement in the system. To tackle this constraint, techniques to predict the system states are required. Such techniques are the state estimation techniques. The idea is to estimate the system states mathematically using the available sensors. Figure 6.1 show the setup of a technique for estimating unmeasured states.

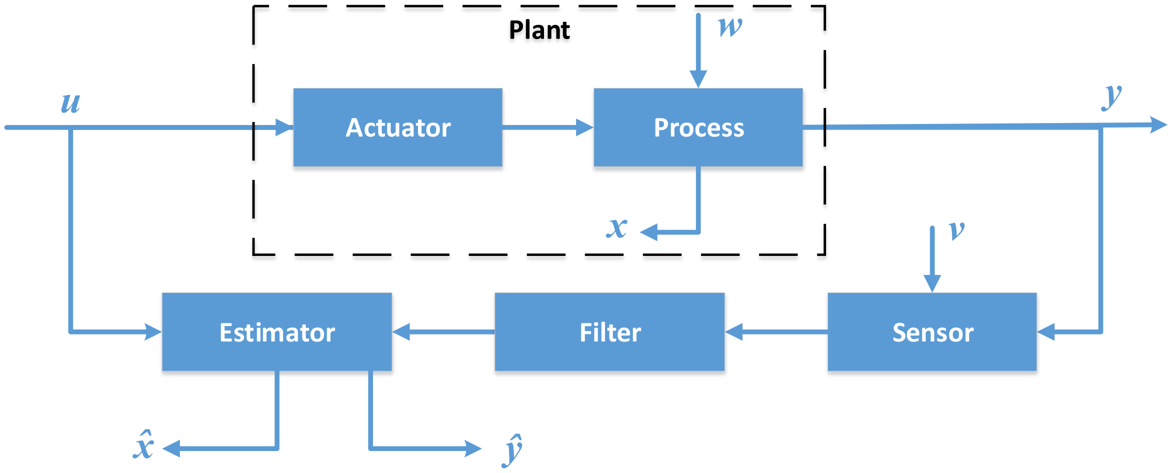

Figure 6.1: Overview of a measurement system with embedded state estimation system for a plant (real process)

6.1.1 Importance of state estimation in oil well drilling operation

In oil well drilling operation, it is important to detect kick or loss to prevent hazardous situation as discussed in section 2.4. This can be done by estimating the return mud flow. Since the return flow measurement requires expensive flow measuring techniques for a reliable flow estimation, other cheaper techniques are preferable. The aim is to remove the expensive flow measuring devices. This means that the flow must be estimated using suitable and reliable estimation technique.

6.2 An introduction to state estimation

An observer or an estimator is a system that deduce some information based on the available process measurements. The idea is to use the knowledge of past available measurements and extract some information about the unmeasured states. It is important to know the system dynamics before this idea can be applicable. A mathematical model of some form is required to describe the system dynamic. Unmeasured states can be estimated if the inputs of the system are known, the system dynamics is known and some of the states are measured. This is not enough, however, to estimate the states. States must be observable. This information is acquired by the observability analysis as discussed in section 5.1.4. A state estimation technique is only as good as the best mathematical model. In other words, a bad model gives a bad estimate which leads to a bad control system. This can happen when ODE's are used to describe a multivariable system rather than PDE's. A linearized model is used instead of a nonlinear model, see section 5.3.2 for comparison of linear and nonlinear model. Model parameter uncertainties also contribute to a bad model as discussed in section 5.2. There are always disturbances and stochastic noise processes influencing the real process as shown in Figure 6.1. Hence, the measurements will always have uncertainties. Therefore, the quality of measurements, type of mathematical model (ODE's or PDE's, linear or nonlinear) and the sensor accuracies impact the optimum estimation. In simple terms, a proper knowledge of the system is required for best estimation. More knowledge of the system there is, the better is the approximation of system dynamics. Take away from this is that state observer is a reverse process. It is about determining the initial states and the state dynamics based on the past knowledge.

6.2.1 Process of state estimation

An observer or an estimator runs in parallel with the real process. The input excites both the estimator and the real process. Initially, the estimated states are the same as the initial guess for the real process states since the measurement from the real process is not available. From the next available measurement, the states are estimated based on the error between the predicted output and the measured output. Simply stated, using the error information, state estimates are updated. In many cases, the measurements are too noisy that gives noisy estimates. To avoid that, the noises are filtered out by a suitable filter before feeding the measurements to the observer system (Figure 6.1). Hence, such system is of feedback type. The process continuous until process termination.

6.2.2 State estimation approach

There are different approaches for state estimation. Deterministic and stochastic approaches such as Luenberger observer design and Kalman filter respectively are the two of the most used approaches. Both approaches are presented in subsequent chapters. Linear as well as nonlinear versions of these approaches are available. This thesis work concentrates on linear and semi-nonlinear observer and linear and nonlinear Kalman filter for the ODEs representing the flow through the top side of the venturi flume.

7 State Observer

In this chapter, the design and stability of a linear state observer is discussed in detail. This chapter focuses on the idea behind the linear observer and the algorithm. Implementation of observers on the mathematical model at the end is also presented here.

7.1 Full order state observer

A full order state observer is a system that estimates all the states of a system, both measured and unmeasured states. It simply means that the order of an observer is equal to the order of the model. In this section, a linear full order state observer is derived. A linear state observer is designed based on the deterministic linear state space model given by (4.24) and (4.25). The real system is stochastic in nature. The better approximation of the real system (Figure 6.1) would be described by (7.1) and (7.2). The analytical solution for a such system is given by (7.3) and (7.4).

$$\delta\dot{x}_r = A_r\delta x_r + B_r\delta u + Ew \tag{7.1}$$

$$\delta y_r = C_r\delta x_r + Fv \tag{7.2}$$

$$\delta x_r(t) = \Phi_r(t)\delta x_r(0) + \int_0^t \Phi_r(t-\tau)B_r\delta u(\tau)d\tau + \int_0^t \Phi_r(t-\tau)Ew(\tau)d\tau \tag{7.3}$$

$$\begin{aligned}\delta y_r(t) = C_r\Phi_r(t)\delta x_r(0) + C_r\int_0^t \Phi_r(t-\tau)B_r\delta u(\tau)d\tau \\ + C_r\int_0^t \Phi_r(t-\tau)Ew(\tau)d\tau + Fv(t)\end{aligned} \tag{7.4}$$

Where, matrices and variables with subscript (r) represent the real system. Φ_r is the state transition matrix for real process and is defined as:

$$\Phi_r(t) = e^{A_r t}$$

Assume that the real system can be represented by a linear model of the form (7.1) and (7.2). There is some deviation between the real states (7.3) and the model states (5.5). The deviation is the error between the model states and the real states. Similarly, there is some deviation in the real output (7.4) and the model output (5.6). The error exists due to the presence of uncertain model parameters, stochastic noise processes (w, v) in the real system and initialization error $x_r(0) \neq x(0)$. Assume the stable system dynamics in both cases given by following condition:

$$\lim_{t\to\infty}\Phi_r(\mathrm{t}) \to 0 \equiv \lim_{t\to\infty} e^{A_r t} \to 0$$

$$\lim_{t\to\infty}\Phi(\mathrm{t}) \to 0 \equiv \lim_{t\to\infty} e^{A t} \to 0$$

Even with stable system dynamics, the error between the real states given by (7.3) and the model states given by (5.5), at steady state is still not eliminated due to the aforementioned reasons. Similarly, the error between the actual output (7.4) and the model output (5.6) exists as well.

$$\delta x_r - \delta x \neq 0$$

$$\delta y_r - \delta y \neq 0$$

There is no mechanism in the mathematical model to suppress the error. This is where an observer comes in. The observer utilizes the available measurements of the system to reduce the error between the true state and the model states. Hence, the observer equation has the same basic structure as the linear state space model with added error correction term, $L_c(\delta y - \delta\hat{y})$, as given by equation (7.5) and (7.6). The correction term is injected to update the states at each time instance when the measurement is available. Thus, the observer states resemble the real states.

$$\delta\dot{\hat{x}} = A_c\delta\hat{x} + B_c\delta u + L_c(\delta y - \delta\hat{y}) \quad [L_c] = \left[n_x \times n_y\right] \tag{7.5}$$

$$\delta\hat{y} = C_c\delta\hat{x} \tag{7.6}$$

Where, L_c is the observer gain matrix which is chosen by the designer to minimize the observation error, $\delta y - \delta\hat{y}$. Observation error is often called innovation process. The prediction term, $A_c\delta\hat{x} + B_c\delta u$, present to the LHS of (7.5) is called the apriori state estimate. The linear observer (7.5) defines a deterministic system with no stochastic noise processes. Using the injected sensor measurements into the observer system and the input signal, the states are estimated. Figure 7.1 illustrates this principle. State observer equation with injected measurement is given by (7.5) and (7.6).

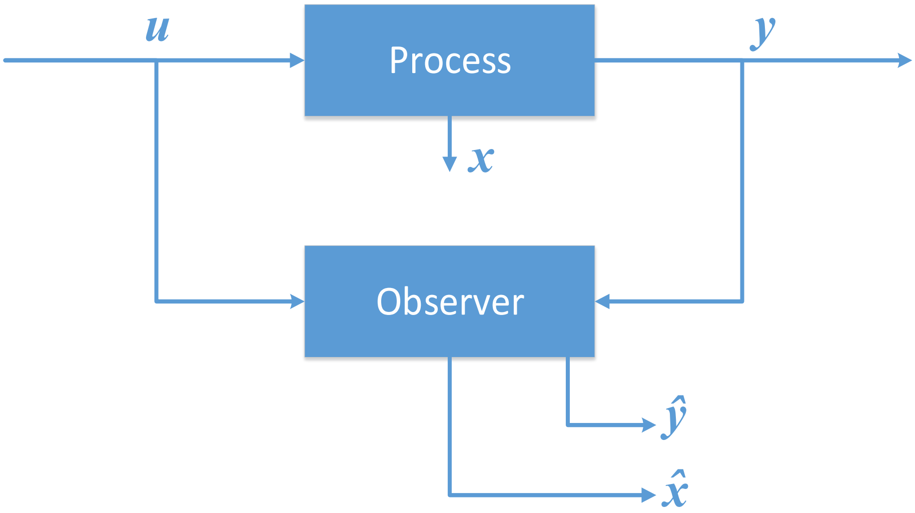

Figure 7.1: State observer overview with injected measurements

Further simplifying equation (7.5) gives:

$$\delta\dot{\hat{x}} = (A_c - L_cC_c)\delta\hat{x} + B_c\delta u + L_c\delta y$$

$$(A_c - L_cC_c) \stackrel{\text{def}}{=} \Lambda \quad [\Lambda] = [n_x \times n_x] \tag{7.7}$$

$$\delta\dot{\hat{x}} = \Lambda\delta\hat{x} + B_c\delta u + L_c\delta y \tag{7.8}$$

Λ is the system matrix for the observer. (7.8) is known as the continuous time linear full-order Luenberger state observer in deviation form. Analytical solution to the state observer equations (7.8) and (7.6) is given by (7.9) and (7.10). Derivation of the analytical solution is shown in detail in Appendix 6 Part 6B.

$$\delta\hat{x}(t) = \Omega(t)\delta\hat{x}(0) + \int_0^t \Omega(t-\tau)B_c\delta u(\tau)d\tau + \int_0^t \Omega(t-\tau)L_c\delta y(\tau)d\tau \tag{7.9}$$

$$\delta\hat{y}(t) = C_c\Omega(t)\delta\hat{x}(0) + C_c\int_0^t \Omega(t-\tau)B_c\delta u(\tau)d\tau + C_c\int_0^t \Omega(t-\tau)L_c\delta y(\tau)d\tau \tag{7.10}$$

Where $\Omega(\mathrm{t})$ is the state transition matrix for the observer and is given as:

$$\Omega(t) = e^{\Lambda t}$$

$$\Omega(t-\tau) = e^{\Lambda(t-\tau)}$$

Solutions (7.9) and (7.10) have incorporated the current measurements as the feedback term in the observer. This helps to suppress the error between the estimates and the real process.

7.1.1 Observer error dynamic

The observed states are likely to differ from the actual states of the real process due to model parameter uncertainties and process noise as discussed earlier. In such case, the dynamics of the real system may not be explained properly. Hence, the model states must be recalibrated. This can be done by utilizing the available real process measurements. We can express the system dynamics in terms of the error between the model states and the observer states. By defining the stable error dynamics, the estimated states approach the real states as given by (7.11).

$$\begin{gathered}\varepsilon \overset{\text{def}}{=} \delta x - \delta\hat{x} \\ \lim_{t\to\infty} \varepsilon \to 0 \quad \lim_{\varepsilon\to 0} \delta\hat{x} \to \delta x\end{gathered} \tag{7.11}$$

$$\dot{\varepsilon} \overset{\text{def}}{=} \delta\dot{x} - \delta\dot{\hat{x}} \tag{7.12}$$

Simplifying (7.12) further yields:

$$\dot{\varepsilon} = A_c\delta x + B_c\delta u - A_c\delta\hat{x} - B_c\delta u - L_c(\delta y - \delta\hat{y})$$

$$\dot{\varepsilon} = A_c\delta x - A_c\delta\hat{x} - L_cC_c\delta x + L_cC_c\delta\hat{x}$$

$$\dot{\varepsilon} = A_c(\delta x - \delta\hat{x}) - L_cC_C(\delta x - \delta\hat{x})$$

$$\dot{\varepsilon} = (A_c - L_cC_c)(\delta x - \delta\hat{x}) \tag{7.13}$$

Define a transition matrix using (7.7) and simplify the above equation using definition for error dynamics (7.11). The state error dynamics is formulated as (7.14).

$$\dot{\varepsilon} = \Lambda\varepsilon \tag{7.14}$$

7.2 Observer design by pole placement

It is vital to show that the error converges to zero to prove stability of the error dynamic. This can be done by analyzing the poles (eigenvalues) of the analytical solution for the equation (7.14). The stability analysis of a system using poles and eigenvalues is discussed in detail in section 5.1.2. The analytical solution for (7.14) is given by (7.15). The detailed process for finding the analytical solution is shown in Appendix 6 Part 6C.

$$\begin{gathered}\varepsilon(t) = e^{\Lambda t}\varepsilon(0) \\ \varepsilon(t) = \Omega(t)\varepsilon(0)\end{gathered} \tag{7.15}$$

(7.15) shows that the error dynamic is defined by the initial estimation error and the matrix Λ, which is constant. Hence, it is important to prove that for any initialization error, the error dynamic is stable. This is done by analyzing the matrix Λ.

7.2.1 Stability of a system using pole placement

Since A_c and C_c are constant matrices for an LTI state space model or the error dynamic, the initial error estimate is non-zero i.e., $\varepsilon(0) \neq 0$. Only L_c is unknown in (10.10), which can be adjusted to make error dynamic converge to zero as $t \to \infty$. In other words, chose L_c such that the error dynamic system matrix Λ is asymptotically stable, and the error dynamic is also stable. Stable Λ forces error to decay to zero for any initial value of error. Initialization error will, therefore, decay asymptotically to zero as $t \to \infty$.

$$\lim_{t\to\infty} e^{\Lambda t} \to 0 \xrightarrow{iff} \Lambda < 0$$

$$\lim_{t\to\infty} \varepsilon(t) \to 0 \xrightarrow{iff} \Lambda < 0$$

Above statement means that the system is stable if and only if Λ is Hurwitz. In other words, Λ must have negative eigenvalues for above condition to hold. Poles of a controllable and observable system are the eigenvalues of the system transition matrix. For error dynamics, the transition matrix is given by Λ. To make the error dynamic system stable, it is possible to force Λ to have negative eigenvalues if error dynamic is controllable. In other words, poles of the transition matrix Λ can be placed at any arbitrary location in the left half of the complex plane for a controllable error dynamic. It can be proven that the eigenvalues of Λ is the same as the eigenvalues of Λ^{T}.

$$eig(\Lambda) = eig(\Lambda^{\mathrm{T}})$$

$$\Lambda = A_c - L_cC_c \quad \equiv \quad \Lambda^{\mathrm{T}} = A_c^T - C_c^T L_c^T$$

For (7.14) to be controllable, Λ^{T} must have full rank i.e., $rank(\Lambda^{\mathrm{T}}) = n_x$. This can be determined using the controllability matrix (5.7) for the error dynamics as given below:

$$\Delta_n = [C_c^T \quad A_c^T C_c^T \quad \cdots \quad (A_c^{n-1})^T C_c^T]^T \equiv [C_c^T \quad (C_cA_c)^T \quad \cdots \quad (C_cA_c^{n-1})^T]^T = O_n \tag{7.16}$$

If $rank(\Delta_n) = n_x$, then the error dynamic is controllable. One important observation to make here is that the controllability matrix of the error dynamic given above is the same as the observability matrix of the linear model given by (5.9) [15]. Thus, for an observable linear model, the error dynamic is always controllable. The venturi model is already proven to be both controllable and observable (section 5.1.5). Since the matrix pair (A_c^T, C_c^T) is controllable, i.e., the pair (A_c, C_c) is observable for a particular choice of L_c.

L_c can be chosen such that Λ is Hurwitz. MATLAB has a built-in function "place.m" that returns the value of L_c for an arbitrary pole position. LabVIEW also has a built-in sub-vi called

"CD Pole Place.vi". The location of poles determines how fast the system responds. The rule of thumb is to place the poles of the Λ more towards the left of the system A_c poles. By doing so, the system becomes more responsive and converges faster. In theory, observer poles can be placed anywhere in the left half of the complex plane. A rule of thumb is to place them more to the left of the system poles.

$$real(\lambda_L) < real(\lambda_{sys})$$

$$real(eig(\Lambda)) < real(eig(A_c))$$

Since all poles of the venturi system lie in the left side of the complex plane (Figure 7.2), the system is stable. Based on the stable system poles, three different observers are designed. The poles of these observers are plotted together with the venturi system poles as shown in Figure 7.2. Table 7.1 shows how the poles for all the observers are positioned. All the observers are designed by placing the poles more towards the left of the system poles. Poles of the observer system matrix are the same as the poles of the error dynamic system matrix. Revisit equations (7.8) and (7.14).

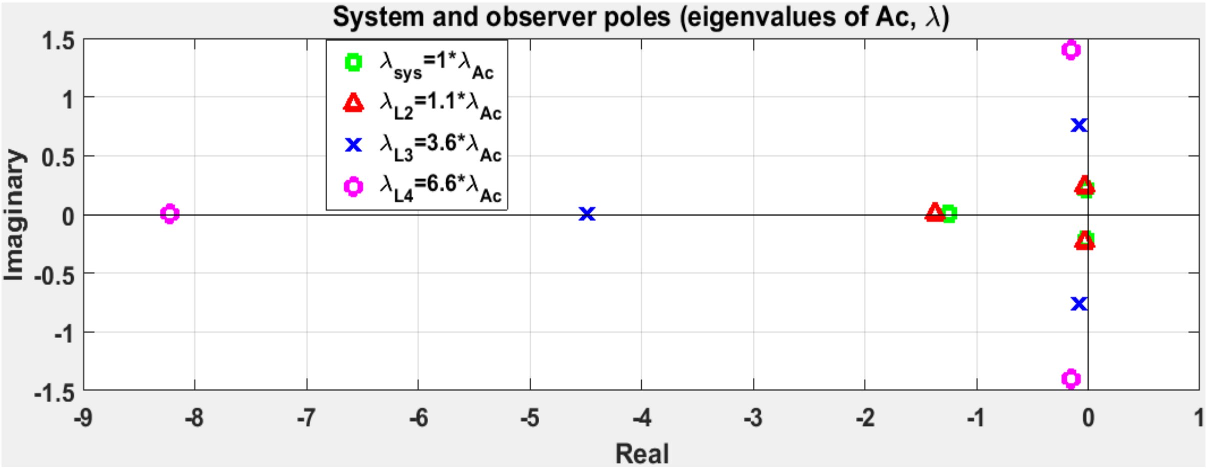

Figure 7.2: System and observer poles in complex plane

Table 7.1: Venturi model and state observer poles determined using pole placement method

Model	Multiplier (l)	Designator	Poles (λ)
Venturi model	*1*	$\lambda_{sys} = 1 \times \lambda_{A_c}$	$\lambda_1 = -1.246 + 0.000i$ $\lambda_2 = -0.024 + 0.212i$ $\lambda_3 = -0.024 - 0.212i$
Observer (L2) Λ_2	*1.1*	$\lambda_{L2} = 1.1 \times \lambda_{A_c}$	$\lambda_1 = -1.371 + 0.000i$ $\lambda_2 = -0.026 + 0.233i$ $\lambda_3 = -0.026 - 0.233i$
Observer (L3) Λ_3	*3.6*	$\lambda_{L3} = 3.6 \times \lambda_{A_c}$	$\lambda_1 = -4.485 + 0.000i$ $\lambda_2 = -0.085 + 0.762i$ $\lambda_3 = -0.085 - 0.762i$

Observer (L4) Λ_4	*6.6*	$\lambda_{L4} = 6.6 \times \lambda_{A_c}$	$\lambda_1 = -8.223 + 0.000i$ $\lambda_2 = -0.155 + 1.397i$ $\lambda_3 = -0.155 - 1.397i$

7.3 Semi-nonlinear observer

A tweaked version of the linear observer has been designed during this work to see if the estimation can be improved. The observer is named semi-nonlinear observer. This is because, L_c is determined using linear model matrices A_c and C_c. To determine L_c, the same pole placement technique is used for semi-nonlinear observer as for the linear observer. However, the nonlinear state equation is used instead of linear state equation for apriori state estimation. Linearization of nonlinear model removed the HOT in Tylor series approximation of the nonlinear model. This induced a deviation in the linear model away from the operating point. By introducing the nonlinear deterministic state equation $\dot{x} = f(x, u)$, the hope is to reduce the estimation error induced by the removal of HOT.

$$\dot{\hat{x}} = f(\hat{x}, u) + L_c\big(g(x) - g(\hat{x})\big) \tag{7.17}$$

Where, $f(\hat{x}, u)$ and $g(\hat{x})$ are nonlinear state and output equations respectively. In case of venturi model, $g(x)$ is a linear output equation given by (4.29). Therefore, $g(\hat{x})$ is $\hat{h}_2$. Based on the venturi model, the semi observer equation takes the form:

$$\dot{\hat{x}} = f(\hat{x}, u) + L_c(y - \hat{y}) \tag{7.18}$$

$$y = g(x) = Cx \tag{7.19}$$

$$\hat{y} = g(\hat{x}) = C\hat{x} \tag{7.20}$$

For semi-nonlinear observer, the error dynamic based on nonlinear model is given by (7.21).

$$\dot{\varepsilon} = \dot{x} - \dot{\hat{x}}$$

$$\dot{\varepsilon} = f(x, u) - f(\hat{x}, u) - L_c\big(g(x) - g(\hat{x})\big)$$

$$\dot{\varepsilon} = f(x - \hat{x}, u) - L_c(C_c x - C_c\hat{x})$$

$$\dot{\varepsilon} = f(x - \hat{x}, u) - L_c C_c(x - \hat{x})$$

$$\dot{\varepsilon} = f(\varepsilon, u) - L_c C_c \varepsilon \tag{7.21}$$

Equation (7.21) defines the nonlinear error dynamic based on the semi-nonlinear observer and nonlinear model of the system.

7.4 Linear observer comparison based on simulation

The linear and semi-nonlinear observer are compared here. Linear observer is implemented in Simulink and MATLAB and the semi-nonlinear observer is implemented in LabVIEW. The implementation is based on the mathematical model (ODEs) of the venturi flume.

7.4.1 Comparison of linear observers with different pole position

First part of the comparison is based on the noise-free input signal. The second part is based on the noisy input signal. The noise variance is determined based on the real level measurements at the collocation point 1.

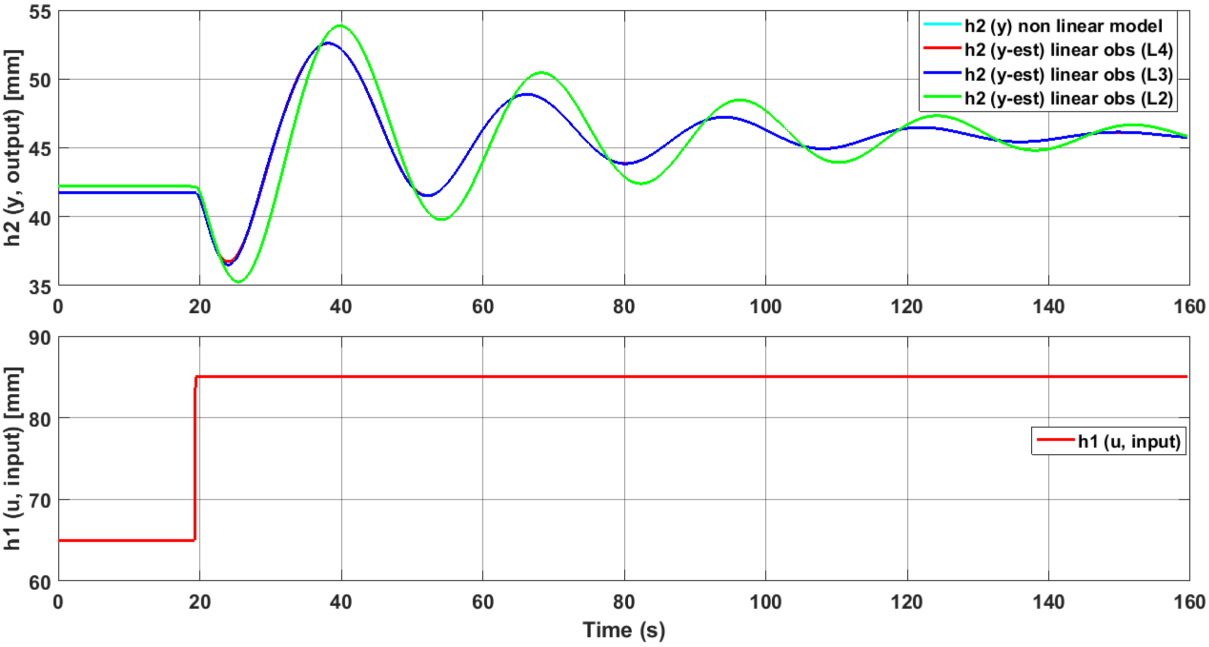

Figure 7.3: Comparison of linear observers with different pole position for h_2 (venturi model) excited with noise-free input signal h_1

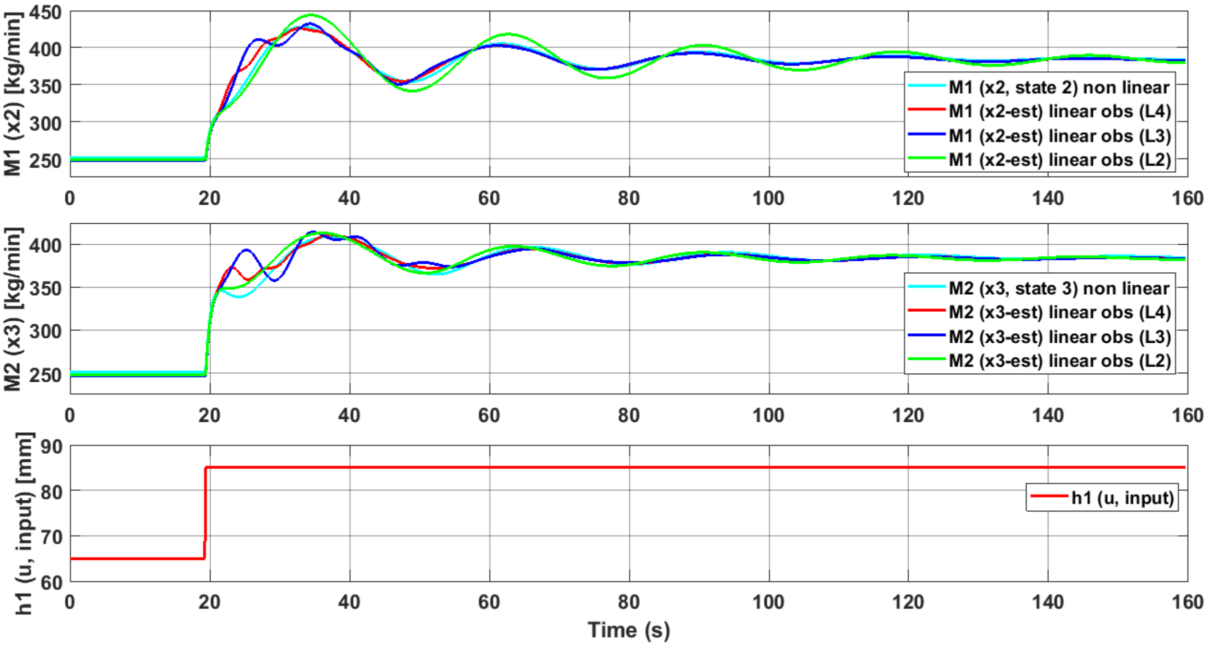

Figure 7.4: Comparison of linear observers with different pole position for M_1 and M_2 (venturi model) excited with noise-free input signal h_1

Figure 7.3 and Figure 7.4 shows the estimates from different observers for h_2 and M_1, M_2 respectively. All observers are excited by the same noise-free input signal. Observer designation and their corresponding poles are shown in Table 7.1. The observer (L4) with the left-most pole (Figure 7.2) responds faster than the observer with right-most pole (L2). The amplitude of the oscillation for L4 is smaller than the oscillation amplitude for (L2). The consequence is that there is less overshoot and undershoot in case of L4. L4 is more robust and faster than L2 for all states. All the observers are asymptotically stable, however, L4 converges faster. This means that if the poles are placed more towards left, the observer responds and converges faster. Since poles of L2 are close to the poles of nonlinear model, state estimates from L2 are overlapped with nonlinear states. As usual, the linear observers show some deviation away from the operating point.

There is a flaw in analyzing linear observers that are excited with noise-free inputs. There is no information about how noise would affect the observers. In real process, the sensor measurements are noisy. For noisy inputs and measurements, it is important to analyze carefully the effect of noise in the observer. Therefore, the synthetic noise is added to the input and output of the venturi model. Figure 7.5 and Figure 7.6 visualizes this circumstance. Nonlinear model and all linear observers are excited with same noisy input signals. Since L4 reacts to input changes faster than L2, the noise is amplified. From sensitivity analysis (Figure 5.7 from section 5.2.3), it is determined that h_1 has a small effect on h_2 whereas, the effect on M_1 and M_2 is large. This is seen in estimates as well. Flow rate estimates are noisier then level estimate. In the presence of noisy input, L2 estimates are less noisy than L4 estimates. Hence, it is a tradeoff between fast response, small over- undershoot and noisy estimates; and slow response, larger over- or undershoot and filtered estimates. One way to suppress the noise is to use a signal filter such as lowpass time constant filter, median filter or weighted moving average filter. These three filters are compared with each other using the real measurement data when implementing the observers on the real system.

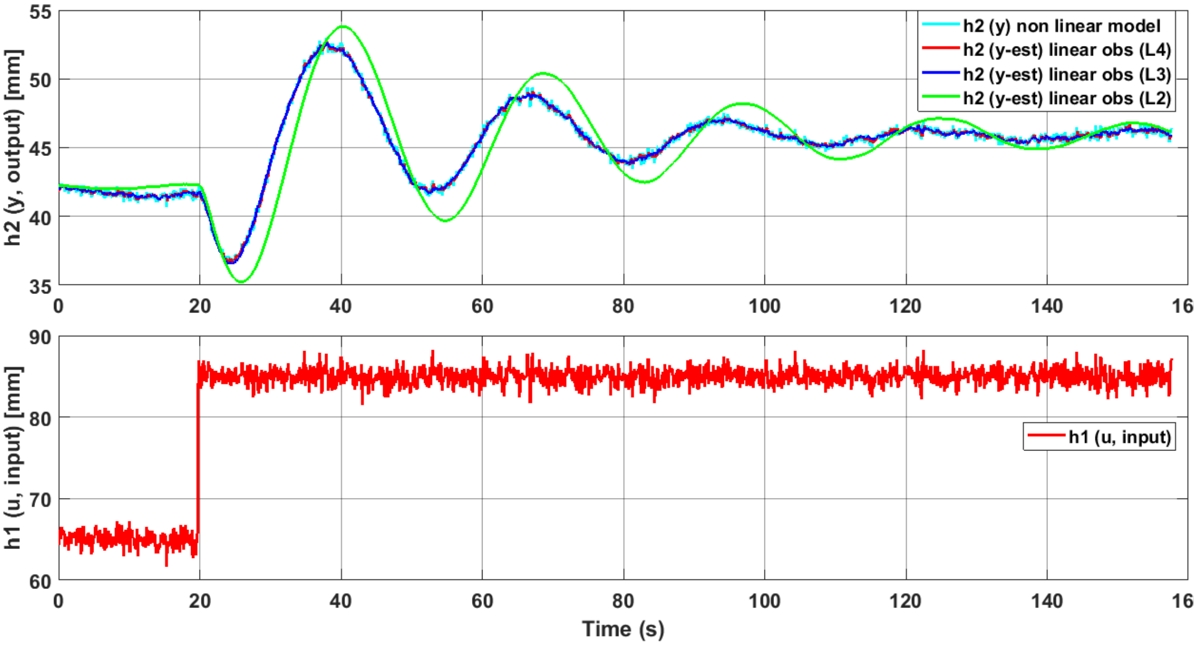

Figure 7.5: Comparison of linear observers with different pole position for h_2 (venturi model) with added synthetic noise in the input h_1 and the output h_2

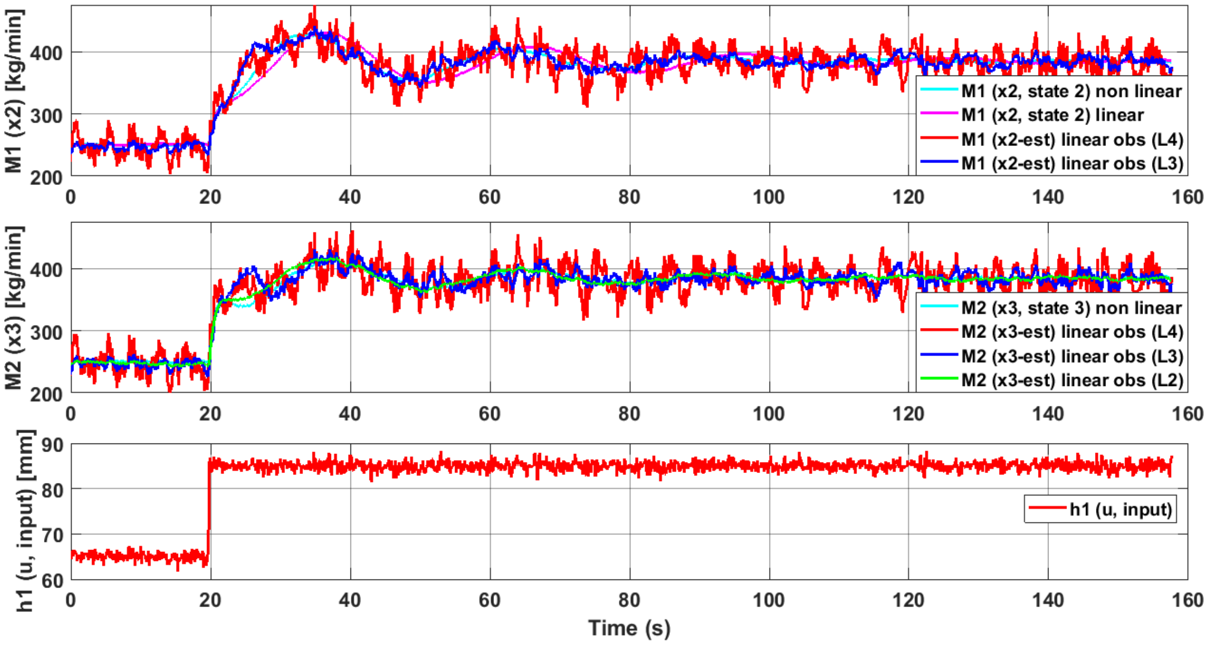

Figure 7.6: Comparison of linear observers with different pole position for M_1 and M_2 (venturi model) with added synthetic noise in the input h_1 and the output h_2

Conclusion here is that if the observer poles are too close to the system poles gives slow response and takes longer to converge. If the observer poles are placed too far to the left, the noise is amplified due to faster response. The period of oscillation also decreases with faster poles. Shorter oscillation period means faster convergence. Hence, during observer design, both points must be considered. In case of venturi model, an observer with a pole placed 1.6 times towards the left of the system pole position is deemed to be an optimal choice.

7.4.2 Comparison of Linear observer and semi-nonlinear observer

Linear observer gives good estimates of the states. However, the deviation that occurs away from the operating point is not eliminated entirely. Since linear model is only the first two terms of the infinite Tylor series approximation of the nonlinear model, the removal of HOT makes linear model imprecise as compared to the nonlinear model. Thus, the deviation occurs away from the linearization point. This is the reason for designing a semi-nonlinear observer. The observer gain matrix L_c, in both linear and semi-nonlinear cases, is chosen using the pole placement method and the poles of the observers are placed 1.6 times to the left of the system poles. Both observers are applied on the simulator. Hence, the states for the reference point in comparison are the simulated (nonlinear model) states. Both observers are implemented in LabVIEW.

Figure 7.7 shows the result of linear and semi-nonlinear observer applied on simulator for level (h_2) estimation. Similarly, Figure 7.8 shows the result of both observers for estimation of mass flow rates (M_1, M_2) at two collocation points c_1 and c_2. In both cases, the synthetic noise is added to the simulated input (h_1) and the simulated output (h_2) to represent the real process. The synthetic noise in the input and the output has the same variance as that noise in actual level measurements. The linear observer is able to filter out the noise in the estimated states of the system. However, there is a deviation at the steady state in linear observer case. The semi-nonlinear observer is able to reduce the deviation as well as filter out the noise. Level estimates has higher amplitude of oscillation (greater over- undershoot) in linear observer case than in the case of semi-nonlinear observer. This suggests that the semi-nonlinear observer gives more accurate estimates than the linear observer. Semi-nonlinear observer is also more robust than the linear observer.

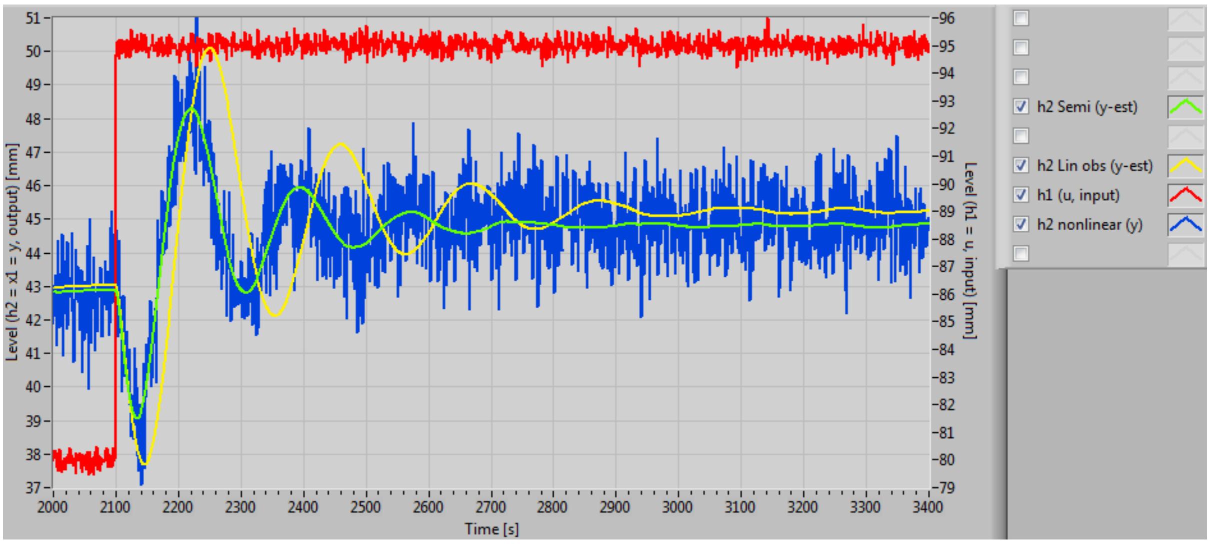

Figure 7.7: Observer comparison based on the simulated output of the nonlinear model $(y = x_1 = h_2)$ for noisy input signal (h_1)

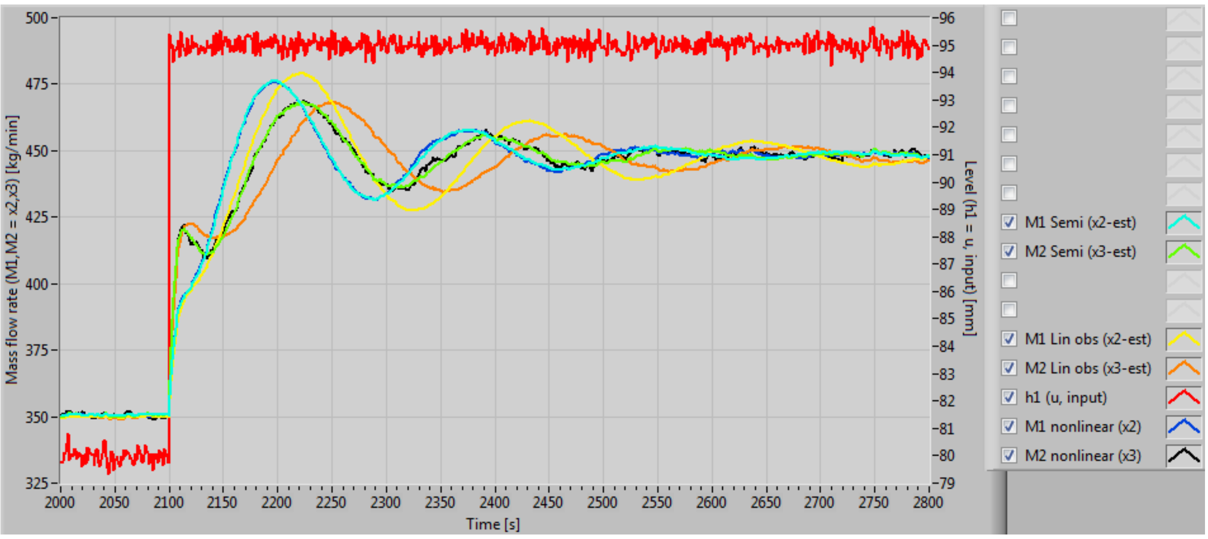

Figure 7.8: Observer comparison based on the simulated states of the nonlinear model $(x_2, x_3) = (M_1, M_2)$ for noisy input signal (h_1)

A better way of analyzing observers is using the error dynamics. Observer with the minimum error between the estimates and the actual states performs better. Figure 7.9 shows the error dynamics for both observers. The magnitude of semi-nonlinear observer error dynamic is much less compared to the magnitude of linear observer error dynamic. At the steady state, the state estimation error is negligibly small for semi-nonlinear observer since the deviation due to linearization is eliminated. In both cases, the level estimation error is small. The main difference is in the mass flow rate estimation. It is evident that the mass flow rate estimation error converges for both observer. However, for semi-nonlinear observer the error decays faster as $t \to \infty$. This concludes that the semi-nonlinear observer performs better than the linear observer. This is also evident in case of noise-free input in a pure simulator. Figure 7.10 and Figure 7.11 show the result of linear and semi-linear estimator for estimating the model states. As with the noisy system, the noise free system also proves that the semi-linear observer is more robust and faster with minimum estimation error than the linear observer. The deviation between the nonlinear model states (reference states) and the observer states are seen clearly in the figures.

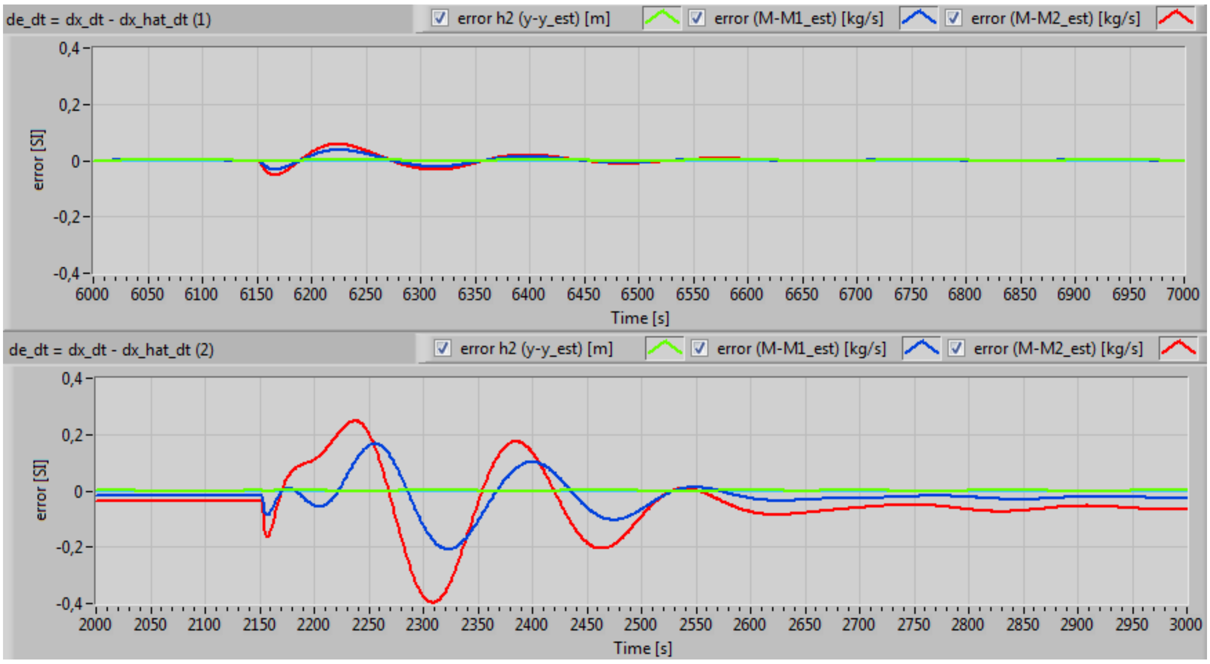

Figure 7.9: Comparison of estimation error dynamics using linear observer (bottom) and semi-nonlinear observer (top).

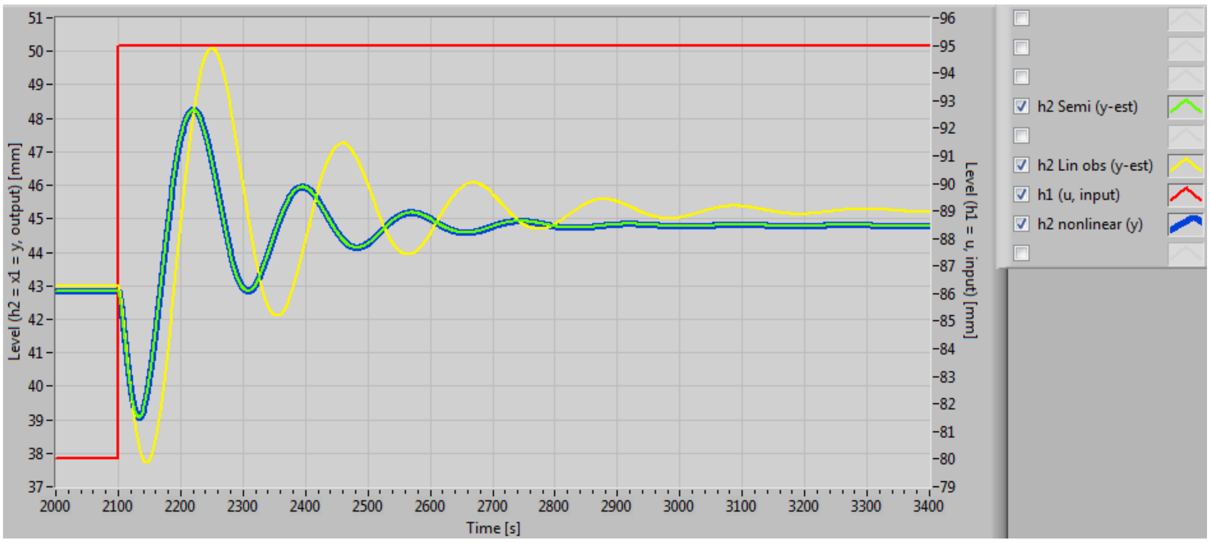

Figure 7.10: Comparison of linear observer and semi-nonlinear for estimating the level as the output ($y = x_1 = h_2$) in parallel with the simulator excited by noise-free input (h_1)

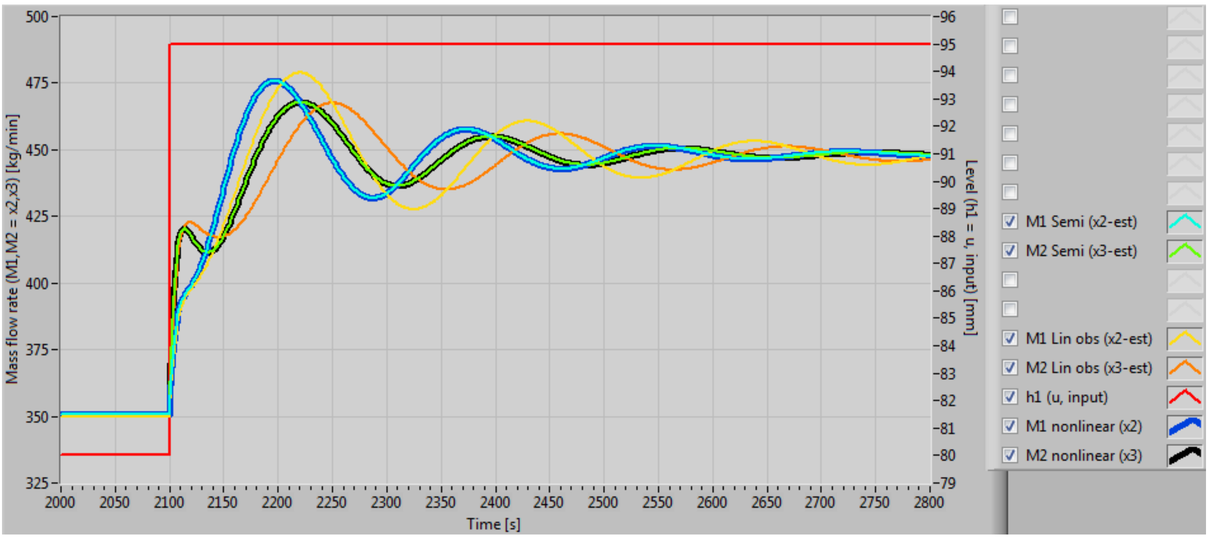

Figure 7.11: Comparison of linear observer and semi-nonlinear for estimating the flow rates as states $(x_2, x_3 = M_1, M_2)$ in parallel with the simulator excited by noise-free input (h_1)

8 Kalman filter

So far, linear observer and semi-nonlinear observer have been investigated. Both of those techniques are based on the deterministic approach. In real world, the systems are most likely to be stochastic in nature. Most process are constantly influenced by disturbances, uncertainties in sensor measurements and process noise. In this chapter, both linear and nonlinear version of Kalman filters are described. Kalman filter is stochastic approach as oppose to Luenberger observer of deterministic approach. In principle, Kalman filter is an optimal state estimator. In Kalman filter case, the main process is excited by stochastic processes, also known as random process noise, v and w as oppose to the deterministic process in observer case. An overview of Kalman filter for state estimation is shown in Figure 8.1 (compare with Figure 7.1 for observer overview).

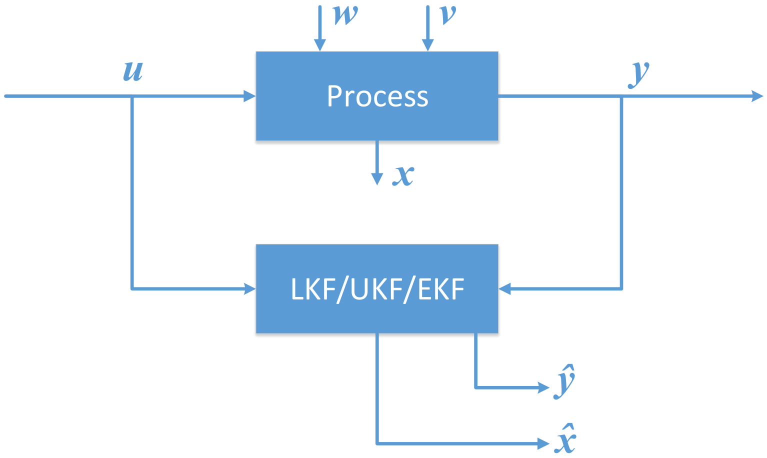

Figure 8.1: State estimation overview with injected measurements and stochastic process

The idea with Kalman filter is that the covariance and the mean of the real states is the same as the covariance and mean of the state estimates. Kalman filter assumes the gaussian distribution for the states. After linear transformation using the linear process model and measurement model, the distribution maintains its gaussian properties. This means that the mean and covariance of the true states and the estimated states are the same after undergoing the linear transformation using the linear model. For a system with gaussian distributed white noise process and measurement noise, the Kalman filter is an optimal linear filter [16, p. 124]. Kalman filter is optimal in a sense that the covariance of state estimation error is minimized. Kalman filter is able to estimate the covariance of the state estimation error influenced by the stochastic processes w, and v (Figure 8.1).

Mean and covariance for an arbitrary vector $\boldsymbol{z}$, is mathematically represented using the mean and covariance operator as (8.1) and (10.10).

$$\bar{z} = E\{\boldsymbol{z}\} = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \boldsymbol{z}_k \tag{8.1}$$

$$\boldsymbol{Z} = E\{\boldsymbol{z}\boldsymbol{z}^T\} = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} (\boldsymbol{z}_k - \bar{z})(\boldsymbol{z}_k - \bar{z})^T \tag{8.2}$$

Where, N is the number of samples. The noise and state estimation error covariance matrices based on (8.1) and (10.10) is given by:

$$X = E\{\varepsilon\varepsilon^T\} = P \quad W = E\{ww^T\} = Q \quad V = E\{vv^T\} = R \tag{8.3}$$

Where, $\varepsilon = x - \hat{x}$. Sometimes, X, W and V are used interchangeably as P, Q and R respectively. Q and R are known constant matrices. Dimension of these matrices are given in Table 8.1. Q and R should be properly tuned for Kalman filter to work properly.

Table 8.1: Size of the covariance matrices used in Kalman filter algorithm for general system (LHS) and for the model venturi flume (RHS).

$P = X$	$Q = W$	$R = V$
$[n_x \times n_x] = [3 \times 3]$	$[n_x \times n_x] = [3 \times 3]$	$[n_y \times n_y] = [1 \times 1]$

8.1 Linear Kalman filter

A continuous-time linear stochastic system with no direct feedthrough term in the output ($D_c = \mathbf{0}$), is described by (8.4) and (8.5).

$$\delta\dot{x} = A_c\delta x + B_c\delta u + w \tag{8.4}$$

$$\delta y = C_c\delta x + v \tag{8.5}$$

Where, w is the process noise and v is the measurement noise. w and v are uncorrelated, zero-mean, Gaussian distributed, white stochastic noise processes [17]. For gaussian distributed random processes, w and v, the sample mean, based on (8.1), is zero and is given as:

$$\bar{w} = E\{w\} = 0 \quad \bar{v} = E\{v\} = 0$$

With zero sample mean for random white noise processes, the mean of the true states is unaffected for large number of samples. Therefore, the estimated states will have the same mean as the true states after the linear transformation. A form of continuous-time linear Kalman filter (LKF) resembles the form of linear Luenberger observer (7.8). The only difference is that the observer gain, L_c is replaced by Kalman gain K_c as shown in (8.6) – (8.8).

$$\delta\dot{\hat{x}} = \Lambda\delta\hat{x} + B_c\delta u + K_c\delta y \tag{8.6}$$

$$\delta\hat{y} = C_c\delta\hat{x} \tag{8.7}$$

$$\Lambda \overset{\text{def}}{=} \mathrm{A_c} - K_cC_c \tag{8.8}$$

Observer gain, L_c, is determined using pole placement method which requires a computation of system eigenvalues. However, no information about the noise processes influencing the system is required. To determine Kalman gain, K_c, a different approach is required. It is often difficult to deduce the noise and disturbance information acting on the system. This makes it difficult to tune noise covariance matrices (Q and R) to find proper gain. Next section discusses the structure of K_c.

8.1.1 Time varying Kalman gain

For a continuous-time system, Kalman gain is computed continuously using, so called, continuous differential algebraic Riccati equation (C-DARE). C-DARE is given by (8.9). The Kalman gain is given by (10.10). Such gain K_c, is sometimes, called an online Kalman gain.

$$\dot{P} = A_c P + P A_c^T + Q - P C_c^T R^{-1} C_c P \tag{8.9}$$

$$K_c = P C_c^T R^{-1} \quad [K_c] = [n_x \times n_y] \tag{8.10}$$

Where, P, Q, R are covariance matrices given by (8.3). C-DARE is a matrix differential equation which needs to be solved to determine K_c. For the case of venturi system, RK4 is used as a solver. To solve a differential equation, a boundary condition is required. Often the initial state estimation error covariance matrix is unknown due to the lack of knowledge of the system states. In such cases, it is a good idea to define the initial state estimation error covariance matrix as a diagonal matrix with a large impulsive parameter, δ.

$$P(t_0) = P_0 = \delta I_{n_x}, \qquad \delta \gg 0$$

There is a built-in function in LabVIEW to solve C-DAREs called "CD Continuous Algebraic Riccati Equations.vi".

8.1.2 Steady state Kalman gain

As the estimation system reaches steady state, the covariance matrix converges to a constant value. This means that, at steady state $\dot{P} = \mathbf{0}$.

$$\lim_{t\to\infty} \dot{P} = \mathbf{0} \xrightarrow{implies} P \to \bar{P}$$

By setting $\dot{P} = \mathbf{0}$ in equation (8.9), C-DARE becomes an algebraic Riccati equation (ARE). Assuming that the P is constant at the steady state and matrices A, C, Q and R are known constant matrices, K_c becomes a constant matrix too. Thus, it is a good idea to compute the steady state Kalman gain offline. Such gain K_c, is sometimes referred to as an offline Kalman gain. Determining the Kalman gain offline reduces the number of steps to execute when implementing the Kalman filter. There is no need to solve CARE to compute K_c. To compute steady state Kalman gain in MATLAB, a function called "kalman.m" can be used. In LabVIEW, a function that fulfills the purpose is "CD Kalman Gain.vi". To determine the Kalman gain, matrices Q and R must be tuned properly. For venturi flume, the values of Q and R is determined using the data measured in the LAB. Using equations (8.2) and (8.3) and real measurements, the knowledge of Q and R can be deduced. Q and R are time invariant constant covariance matrices. So, the proper tuning of these matrices is vital for proper state estimation.

8.1.3 Stability of linear Kalman filter

After determining the Kalman gain, the poles of the transition matrix (8.8) for Kalman filter is determined. Figure 8.2 shows the poles of the Kalman filter plotted together with the system poles and observer poles. Kalman poles are extremely close to the actual system poles. Since all the poles of the linear Kalman filter lie in the left plane of the complex plane and slightly towards the left of the system poles, the estimator is stable.

For controllability, it is sufficient to check the rank of the transition matrix (8.8) for Kalman filter. If the rank is a full rank, then the Kalman filter is controllable i.e., $rank(A_c - K_c C_c) = n_x$. Using MATLAB, the rank was determined to be full rank. Hence, the Kalman filter is controllable. The controllability matrix of Kalman filter is the same as that of the linear observer (10.10), provided that L_c is replaced with K_c.

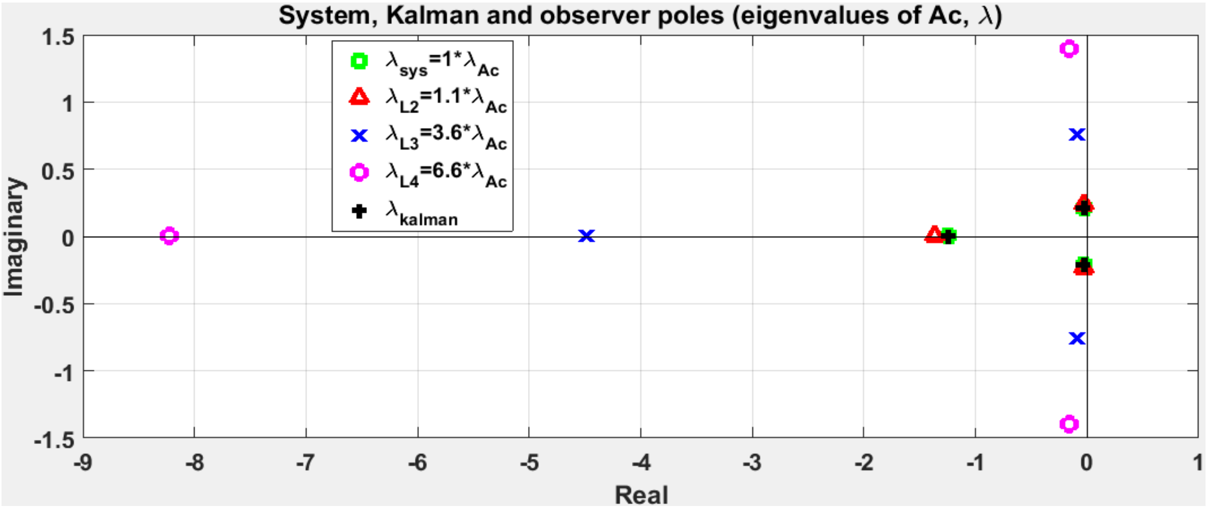

Figure 8.2: Poles of the Kalman filter and venturi model plotted in complex plane

8.2 Nonlinear Kalman filter

Linear model works fine for nonlinear models that can be approximated by a linearized model around an operating point. For highly nonlinear model, the linearization tends to give improper model approximation; specially, when operated away from the operating point. Thus, for highly nonlinear models, linear Kalman filter (LKF) estimates are not optimal. Not optimal in a sense that the covariance of the estimation error is large. For a linear model, the mean and covariance of the state estimate after linear transformation remains the same as that of the true states due to an assumed gaussian distribution. For nonlinear model however, the distribution after the state transformation may not be gaussian. In such case, nonlinear state estimator is required. Two of the nonlinear state estimator are extended Kalman filter (EKF) and unscented Kalman filter (UKF). The algorithm for EKF and UKF for this thesis work is based on the discrete system given by (8.11) and (10.10). The discrete nonlinear model is stochastic in nature with additive process noise (w_k) and measurement noise (v_k). The dimension of each vector for a given time instance is also presented below. For a given time instance, only one sample is considered.

$$\begin{gathered} x_{k+1} = f(x_k, u_k, \theta, t_k) + w_k \\ [x_{k+1}] = [x_k] = [w_k] = [n_x \times 1] \end{gathered} \tag{8.11}$$

$$\begin{gathered} y_k = g(x_k, u_k, \theta, t_k) + v_k \\ [y_k] = [v_k] = [n_y \times 1] \end{gathered} \tag{8.12}$$

Where, θ is a vector of the model parameters as listed in Table 5.3. The covariance matrices of the white noise process are given by (8.3).

8.2.1 Extended Kalman filter

In LKF, the model Jacobians A_c and C_c are computed once and are time invariant (constant). In EKF, the model matrices change since the linearization is performed around the mean of the current state estimates. This gives better approximation of the true states when linearized around the current state estimates. Thus, A_c and C_c are time variant. The probability distribution around the state estimates is, thus, gaussian. After the transformation, gaussian properties is upheld by the estimated states [18]. Note that the transformation is still linear, but the linear model is the best approximation of the nonlinear model around the mean of the current state estimates. This idea is presented below as the step by step discrete EKF algorithm. The algorithm is based on the discrete time nonlinear system (8.11) and (10.10). The algorithm can be directly implemented in any programming language. LabVIEW is used for this thesis.

Some notations used in the EKF algorithm:

- Apriori state estimate: $\hat{x}_k^-$
- Aposteriori state estimate: $\hat{x}_k^+$
- Apriori state covariance matrix: P_k^-
- Aposteriori state covariance matrix: P_k^+
- Size of a matrix or a vector is given inside square braces. $[A] = [r \times c]$. Where, A is an arbitrary variable. r and c represent the number of rows ann columns respectively.

Step by step EKF algorithm:

Step 0. Initialization step (runs only once)

0.1. Define process noise covariance matrix W and measurement noise covariance matrix V given by (8.3). The dimensions of these matrices are:

$$W = E\{ww^{\mathrm{T}}\} \quad [W] = [n_x \times n_x]$$

$$V = E\{ww^{\mathrm{T}}\} \quad [V] = \left[n_y \times n_y\right]$$

0.2. Initialize apriori state estimate based on (8.1)

$$\hat{x}_0^- = E(x_0) \quad [\hat{x}_0^-] = [n_x \times 1]$$

$$\hat{x}_k^- = E(x_k) \quad [\hat{x}_k^-] = [n_x \times 1]$$

0.2.1. It is a good idea to set the initial apriori state estimate to be equal to the initial value for the state based on the knowledge of the system

$$\hat{x}_k^- = \hat{x}_0^- = x_0$$

0.3. Define and initialize state covariance matrix

$$P_0^- = E\{(x_0 - \hat{x}_0^-)(x_0 - \hat{x}_0^-)^T\} \quad [P_0^-] = [n_x \times n_x]$$

$$P_k^- = E\{(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T\} \quad [P_k^-] = [n_x \times n_x]$$

0.3.1. If the initial state covariance is not known, use large scalar δ to multiply an identity matrix of proper size (given below)

$$P_k^- = P_0^- = \delta I_{n_x}$$

Step 1. Predict the output using the apriori state estimate and the measurement equation (10.10)

$$\hat{y}_k = g(\hat{x}_k^-, u_k, \theta, t_k) \quad [\hat{y}_k] = [n_y \times n_x]$$

Step 2. Compute the matrix C_k

2.1. For nonlinear measurement equation (10.10), the matrix C_k must be determined by linearizing the equation around the mean of the apriori state estimate.

$$C_k = \left.\frac{\partial g(x_k, u_k, \theta, t_k)}{\partial x_k^T}\right|_{(\hat{x}_k^-, u_k)} \qquad [C_k] = [n_y \times n_x]$$

2.2. For linear measurement equation, C_k can be constant based. If so, linearization is not necessary, and the step is skipped by using the constant C_k

Step 3. Compute the Kalman gain

$$K_{f_k} = P_k^- C_k^T (C_k P_k^- C_k^T + V)^{-1} \quad [K_{f_k}] = [n_x \times n_y]$$

Step 4. Aposteriori state estimate using the Kalman gain

$$\hat{x}_k^+ = \hat{x}_k^- + K_{f_k}(y_k - \hat{y}_k) \quad [\hat{x}_k^+] = [n_x \times 1]$$

Step 5. Apriori state update

5.1. Using the nonlinear process model (8.11) and aposteriori state estimate, predict the apriori state

$$\hat{x}_{k+1}^- = f(\hat{x}_k^+, u_k, \theta, t_k) \quad [\hat{x}_{k+1}^-] = [n_x \times 1]$$

Step 6. Determine the time varying system matrix A_k

6.1. Linearize the nonlinear process model (8.11) around the aposteriori state estimate

$$A_k = \left.\frac{\partial f(x_k, u_k, \theta, t_k)}{\partial x_k^T}\right|_{(\hat{x}_k^+, u_k)} \qquad [A_k] = [n_x \times n_x]$$

Step 7. Update covariance matrix

7.1. Predict the aposteriori state estimation covariance matrix

$$P_k^+ = (I_{n_x} - K_{f_k} C_k) P_k^- (I_{n_x} - K_{f_k} C_k)^T + K_{f_k} V_k K_{f_k}^T \quad [P_k^+] = [n_x \times n_x]$$

7.2. Predict the aposteriori state estimation covariance matrix

$$P_{k+1}^- = A_k P_k^+ A_k^T + W_k \quad [P_{k+1}^-] = [n_x \times n_x]$$

Step 8. Slide one step forward

8.1. Update apriori state estimates

$$\hat{x}_k^- = \hat{x}_{k+1}^-$$

8.2. Update the apriori state estimation covariance matrix

$$P_k^- = P_{k+1}^-$$

8.3. Update the time instance k (slide 1 step forward)

$$k + 1 := k$$

Repeat the step 1 to 8 as long as required. EKF is explained in detail in [16, p. 400] and [17].

8.2.2 Unscented Kalman filter

Linearization may not be a good approximation around the point of high nonlinearity. For complex nonlinear model, the partial derivatives may not exist and computing Jacobians maybe computationally demanding. These are some of the shortcomings of EKF. Instead of linearizing the nonlinear model, as an EKF does, UKF approximates the probability distribution function around the mean of the estimates. UKF generates a set of sample points that capture the mean and covariance of the state estimates. These points are called the sigma points. Sigma points are symmetrically distributed around the mean of the state estimation. These individual signa points are easier to transform using the nonlinear model. Thus, the nonlinear transformation of the signa points are performed. Such nonlinear transformation of signa points is known as the unscented transformation. After the unscented transformation, the mean and covariance of the transformed sigma points is calculated. The sample mean and the covariance of the transformed signa points are the best estimate of the true mean and covariance. The signa points are generated for both state equation and measurement equation. UKF uses the nonlinear models which eliminates the error of linearization. The idea is presented below as a step by step algorithm that can be used to implement directly in a programming language. LabVIEW is used for implementing UKF.

Some important notations used in the UKF algorithms:

- Apriori is denoted by superscript $-$
  - State estimate: $\hat{x}_k^-$
  - Apriori state covariance: P_k^-
  - A small perturbation to generate sigma points based on apriori state estimates and aposteriori state covariance matrix: $\tilde{x}_i^-$
- Aposteriori is denoted by superscript $+$
  - Aposteriori state estimate: $\hat{x}_k^+$
  - Aposteriori state covariance: P_k^+
  - A small perturbation to generate sigma points based on aposteriori state estimates and aposteriori state covariance matrix: $\tilde{x}_i^+$
- Sigma points: $\hat{x}_k^{\sigma}$
  - Transformed sigma points using state equation: $\hat{x}_k^{\vec{\sigma}}$
  - Transformed sigma points using measurement equation: $\hat{y}_k^{\vec{\sigma}}$

The UKF algorithm, step by step:

Step 0. Initialization (runs only once)

0.1. Define process noise covariance matrix W and measurement noise covariance matrix V given by (8.3) with proper dimensions

$$W = E\{ww^{\mathrm{T}}\} \quad [W] = [n_x \times n_x]$$

$$V = E\{ww^{\mathrm{T}}\} \quad [V] = [n_y \times n_y]$$

0.2. Define an initial expected value of the aposteriori state estimate at time k. Expected value is the mean of the state estimate based on (8.1)

$$\hat{x}_0^+ = E(x_0) \quad [\hat{x}_0^+] = [n_x \times 1]$$

$$\hat{x}_k^+ = E(x_k) \quad [\hat{x}_k^+] = [n_x \times 1]$$

$$\hat{x}_k^+ = \hat{x}_0^+$$

0.3. Define an initial expected covariance matrix of the aposteriori state covariance matrix at time k

$$P_0^+ = E\{(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T\} \quad [P_0^+] = [n_x \times n_x]$$

$$P_k^+ = E\{(x_k - \hat{x}_k^+)(x_k - \hat{x}_k^+)^T\} \quad [P_k^+] = [n_x \times n_x]$$

0.4. If the initial state covariance is not known, use large scalar δ to multiply an identity matrix of proper size (given below)

$$P_k^+ = P_0^+ = \delta I_{n_x}$$

Step 1. Generate sigma points

1.1. Number of signa points (n_s) to generate is related to the number of states (n_x) in the system. To distribute symmetrically around the mean, twice the number of states is enough to capture the mean and covariance of the state estimate

$$n_s = 2 \cdot n_x$$

1.2. Define matrix square root of the aposteriori covariance matrix

$$sq_{nP}^+ = \sqrt{n_x P_k^+} \quad [sq_{nP}^+] = [n_x \times n_x]$$

1.3. Take a column from the matrix square root to symmetrically distribute the sigma points

$$sq_{nP\,j}^+ = sq_{nP}^+(j,:) \quad j = 1,2,\dots n_x \quad \left[sq_{nP\,j}^+\right] = [1 \times n_x]$$

1.4. Define small perturbation from the mean of the current state estimate.

$$\tilde{x}_i^+ = a \cdot sq_{nP\,j}^{+\ T} \begin{cases} j = i & a = 1 \quad 1 \le i \le n_x \\ j = i - n_x & a = -1 \quad n_x < i \le n_s \end{cases} \quad i = 1,2,\dots n_s$$

$$\left[sq_{nP\,j}^{+\ T}\right] = [n_x \times 1] \quad [\tilde{x}_i^+] = [n_x \times 1] \quad [\tilde{x}^+] = [n_x \times n_s]$$

1.5. Add the perturbation to the aposteriori state mean to generate a set of sigma points

$$\hat{x}_k^{\sigma(i)} = \tilde{x}_i^+ + \hat{x}_k^+ \quad \left[\hat{x}_k^{\sigma(i)}\right] = [n_x \times 1] \quad [\hat{x}_k^{\sigma}] = [n_x \times n_s]$$

Step 2. Perform unscented transformation to the sigma points using the nonlinear state equation. The outcome is the set of transformed sigma points for estimated states

$$\hat{x}_{k+1}^{\vec{\sigma}\ (i)} = f\left(\hat{x}_k^{\sigma(i)}, u_k, \theta, t_k\right) \quad i = 1,2,\dots n_s \quad \left[\hat{x}_{k+1}^{\vec{\sigma}\ (i)}\right] = [n_x \times 1] \quad \left[\hat{x}_{k+1}^{\vec{\sigma}}\right] = [n_x \times n_s]$$

Step 3. Determine the apriori mean and covariance of the transformed sigma points which is the good estimation of the true mean and true covariance of the states

3.1. Apriori mean of the transformed signa points (using (8.1))

$$\hat{x}_{k+1}^- = \frac{1}{n_s}\sum_{i=1}^{n_s} \hat{x}_{k+1}^{\vec{\sigma}\ (i)} \quad [\hat{x}_{k+1}^-] = [n_x \times 1]$$

3.2. Apriori state covariance matrix of the transformed sigma points (using (8.2)). The process noise covariance is added to take care of the process noise.

$$P_{k+1}^{-} = \frac{1}{n_s}\sum_{i=1}^{n_s}\left(\hat{x}_{k+1}^{\vec{\sigma}}{}^{(i)} - \hat{x}_{k+1}^{-}\right)\left(\hat{x}_{k+1}^{\vec{\sigma}}{}^{(i)} - \hat{x}_{k+1}^{-}\right)^{T} + W \quad i = 1,2,\dots n_s \quad [P_{k+1}^{-}] = [n_x \times n_x]$$

The matrix multiplication takes care of the Σ sign above. The proof is given in Appendix 8. Working in matrix form, especially in MATLAB, the above algorithm can be written as:

$$P_{k+1}^{-} = \frac{1}{n_s}\left(\hat{x}_{k+1}^{\vec{\sigma}} - \hat{x}_{k+1}^{-}\right)\left(\hat{x}_{k+1}^{\vec{\sigma}} - \hat{x}_{k+1}^{-}\right)^{T} + W$$

Step 4. Generate a new set of sigma points using the apriori covariance matrix

4.1. Determine the matrix square root of apriori covariance matrix

$$sq_{nP}^{-} = \sqrt{n_x P_{k+1}^{-}} \quad [sq_{nP}^{-}] = [n_x \times n_x]$$

4.2. Take a column from the matrix square root to symmetrically distribute the sigma points

$$sq_{nP}^{-}{}_{j} = sq_{nP}^{-}(j,:) \quad j = 1,2,\dots n_x \quad \left[sq_{nP}^{-}{}_{j}\right] = [1 \times n_x]$$

4.3. Define small perturbation from the mean of the current state estimate.

$$\tilde{x}_i^{-} = a \cdot sq_{nP}^{-}{}_{j}^{T}\begin{cases} j = i & a = 1 \quad 1 \le i \le n_x \\ j = i - n_x & a = -1 \quad n_x < i \le n_s \end{cases} \quad i = 1,2,\dots n_s$$

$$\left[sq_{nP}^{-}{}_{j}^{T}\right] = [n_x \times 1] \quad [\tilde{x}_i^{-}] = [n_x \times 1] \quad [\tilde{x}^{-}] = [n_x \times n_s]$$

4.4. Add the perturbation to the apriori state mean to generate a new set of sigma points

$$\hat{x}_{k+1}^{\sigma}{}^{(i)} = \tilde{x}_i^{-} + \hat{x}_{k+1}^{-} \quad \left[\hat{x}_{k+1}^{\sigma}{}^{(i)}\right] = [n_x \times 1] \quad [\hat{x}_{k+1}^{\sigma}] = [n_x \times n_s]$$

Step 5. Perform unscented transformation to the sigma points using the nonlinear measurement equation. The outcome is the set of transformed sigma points for estimated output

$$\hat{y}_k^{\vec{\sigma}(i)} = g\left(\hat{x}_{k+1}^{\sigma}{}^{(i)}, u_k, \theta, t_k\right) \quad i = 1,2,\dots n_s \quad \left[\hat{y}_k^{\vec{\sigma}(i)}\right] = \left[n_y \times 1\right] \quad \left[\hat{y}_k^{\vec{\sigma}}\right] = \left[n_y \times n_s\right]$$

Step 6. Determine the mean and covariance of the transformed sigma points which is the good estimation of the true mean and true covariance of the output

6.1. Mean of the transformed signa points (using (8.1))

$$\hat{y}_k = \frac{1}{n_s}\sum_{i=1}^{n_s}\hat{y}_k^{\vec{\sigma}(i)} \quad [\hat{y}_k] = \left[n_y \times 1\right]$$

6.2. Measurement covariance matrix of the transformed sigma points (using (8.2)). The measurement noise covariance is added to take care of the noise

$$P_y = \frac{1}{n_s}\sum_{i=1}^{n_s}\left(\hat{y}_k^{\vec{\sigma}(i)} - \hat{y}_k\right)\left(\hat{y}_k^{\vec{\sigma}(i)} - \hat{y}_k\right)^{T} + V \quad \left[P_y\right] = \left[n_y \times n_y\right]$$

6.3. Determine the cross covariance between the states and the outputs

$$P_{xy} = \frac{1}{n_s}\sum_{i=1}^{n_s}\left(\hat{x}_{k+1}^{\vec{\sigma}\,(i)} - \hat{x}_{k+1}^{-}\right)\left(\hat{y}_k^{\vec{\sigma}\,(i)} - \hat{y}_k\right)^T \quad [P_{xy}] = [n_x \times n_y]$$

The matrix multiplication takes care of the Σ sign above. The proof is given in Appendix 8. Working in matrix form, especially in MATLAB, the above algorithm can be written as:

$$P_y = \frac{1}{n_s}\left(\hat{y}_k^{\vec{\sigma}} - \hat{y}_k\right)\left(\hat{y}_k^{\vec{\sigma}} - \hat{y}_k\right)^T + V$$

$$P_{xy} = \frac{1}{n_s}\left(\hat{x}_{k+1}^{\vec{\sigma}} - \hat{x}_{k+1}^{-}\right)\left(\hat{y}_k^{\vec{\sigma}} - \hat{y}_k\right)^T$$

Step 7. Determine the Kalman gain

$$K_{f_k} = P_{xy}P_y^{-1} \quad \left[K_{f_k}\right] = [n_x \times n_y]$$

Step 8. Predict the aposteriori state estimate and state covariance matrix

8.1. Aposteriori state estimate using the measurement

$$\hat{x}_{k+1}^{+} = \hat{x}_{k+1}^{-} + K_{f_k}(y_k - \hat{y}_k) \quad [\hat{x}_{k+1}^{+}] = [n_x \times 1]$$

8.2. Aposteriori state covariance matrix

$$P_{k+1}^{+} = P_{k+1}^{-} - K_{f_k}P_yK_{f_k}^T \quad [P_{k+1}^{+}] = [n_x \times n_x]$$

Step 9. Slide one step forward

9.1. Update apriori state estimates

$$\hat{x}_k^{+} = \hat{x}_{k+1}^{+}$$

9.2. Update the apriori state estimation covariance matrix

$$P_k^{+} = P_{k+1}^{+}$$

9.3. Update the time instance k (slide 1 step forward)

$$k + 1 \coloneqq k$$

Repeat the step 1 to 9 as long as required. UKF is explained in detail in [16, p. 433] and [17].

8.3 Comparison between Kalman filters and observers based on simulation

Kalman filters (both nonlinear and linear) and observers (both semi-nonlinear and linear) are implemented in LabVIEW. The comparison of these estimators is presented in this section.

8.3.1 Comparison of online and offline Kalman gain for LKF

Figure 8.3 and Figure 8.4 shows the output of the Kalman filter for online and offline gain respectively. An input with additive gaussian distributed white noise excites the both system. The output is also influenced by an additive white noise. Kalman filter is able to filter out the noise in both cases. There is not significant difference between both applications. This shows that the Kalman gain calculated offline is just as good as the online gain. However, in case of linear Kalman filter with online Kalman gain, more equations are involved that needs to be solved such as CARE and Kalman gain itself repeatedly since state estimation is performed in loop. By removing these overheads, the implementation comes down to a two-equation algorithm given by equations (8.6) and (8.7). Thus, the need for continuous calculation of the gain is avoided and the efficiency of the system increases.

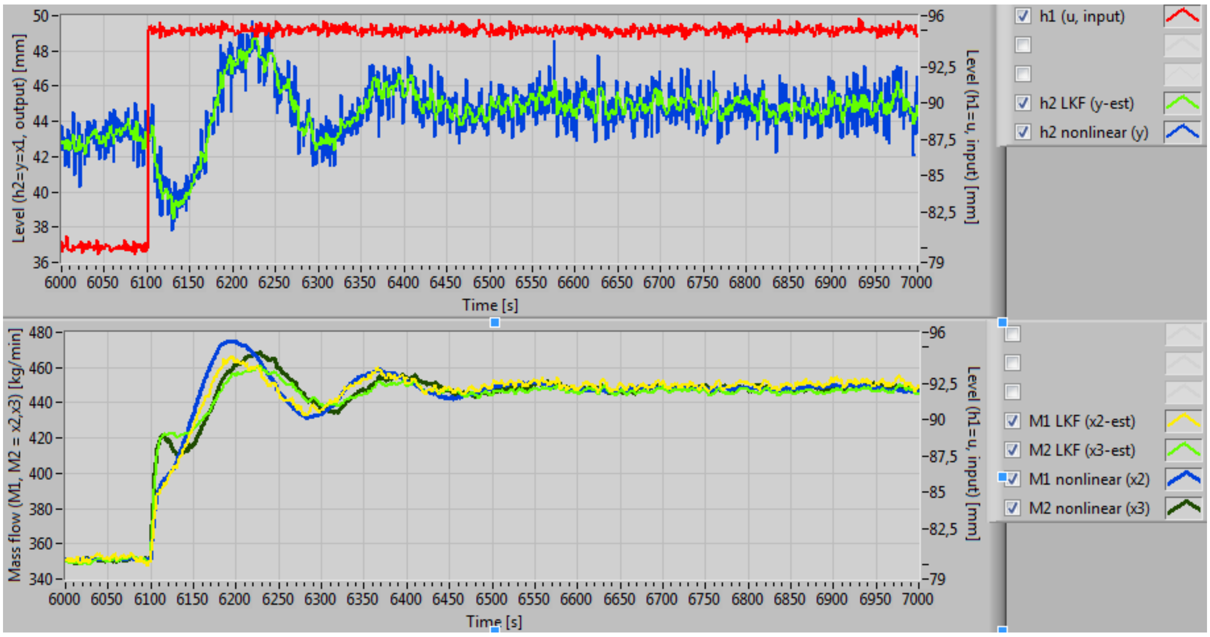

Figure 8.3: State estimation applied on the venturi model using Kalman gain estimated online with added synthetic noise in the input and the output

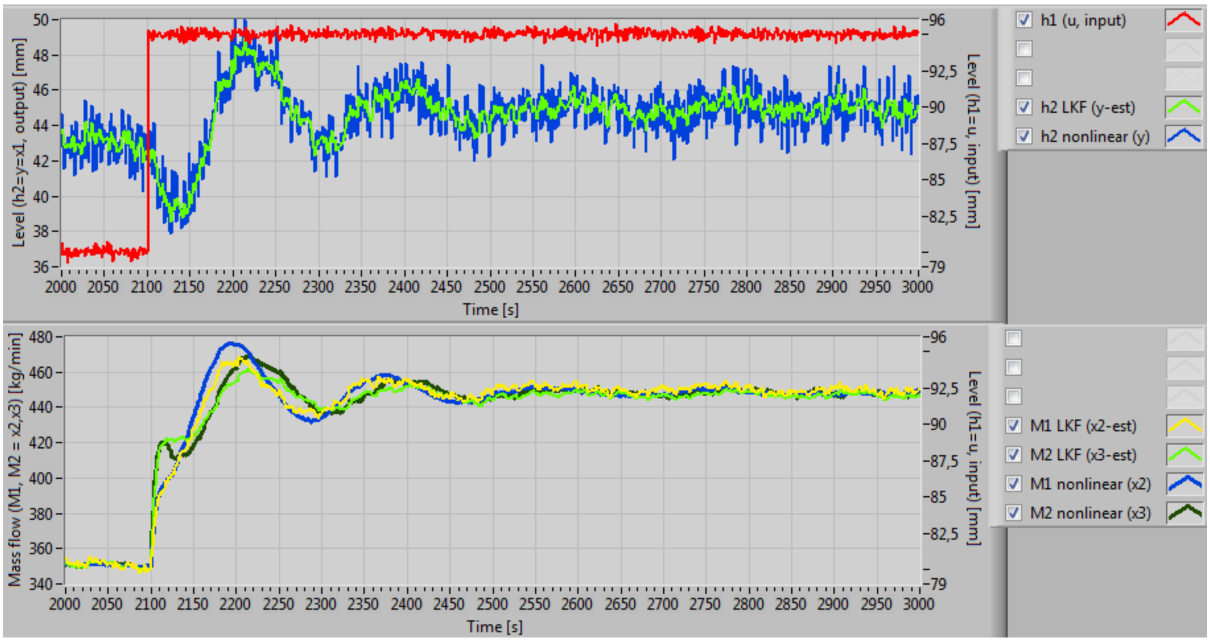

Figure 8.4: State estimation applied on the venturi model using Kalman gain calculated offline with added synthetic noise in the input and output

Comparison of online and offline Kalman gain on noise-free simulator is presented in Appendix 9 Part 9A.

When using LKF, EKF and UKF, covariance matrix for process noise Q and measurement noise R must be tunes properly. For venturi model, since the synthetic noise resembles real noise in level measurement at collocation point 1, it makes sense to define covariance matrices is Q and R based on the real measurements. The values are presented in Table 8.2.

Table 8.2: Covariance matrices and noise variances used in Kalman filters

Kalman filter	$\sigma^2(w)$	$\sigma^2(v)$	Q	R
LKF, EKF, UKF	10^{-6}	10^{-4}	*diag(1,0E-5;1,0E-2; 1,0E-2)*	*6,0E-7*

8.3.2 Comparison of LKF, EKF and UKF

The result of state estimation using LKF, EKF and UKF based on the dynamic simulator of the venturi flume is shown in Figure 8.5 and Figure 8.6. The reference states for comparison are the nonlinear model states. Figure 8.5 shows the level estimates and level predicted by the model whereas Figure 8.6 shows the mass flow rate estimates and the mass flow rate predicted by the model. All Kalman filters are able converge to a steady state while also reducing the noise in system. In case of level estimation, the noise negligible. Noise level in mass flow rate is also minimal. However, LKF has a deviation since it is operating away from the linearization point. In EKF and UKF, the deviation is non-existence. Based on Figure 8.5 and Figure 8.6, it is clear that, the UKF performs better than other two version of the Kalman filter for all states. UKF is better in a sense that it responds faster and converges faster than LKF and EKF. LKF has the largest overshoot and undershoot which can lead to unreliable initial estimates during transient if the change in input is large. At steady state, UKF is noisier than LKF or EKF. This is due to the fact that UKF responds faster to the input change. Since, the input is noisy, UKF responds to it and the estimates are noisier than LKF or EKF.

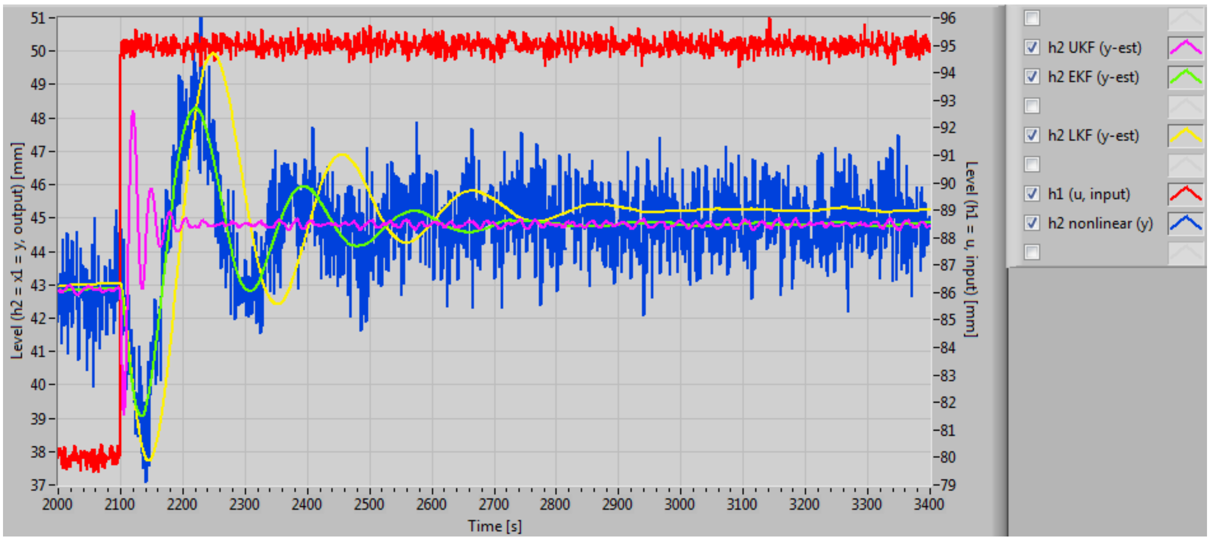

Figure 8.5: Comparison of Kalman filters based on level estimation $(\hat{y} = \hat{x}_1 = \hat{h}_2)$. Simulator is excited by synthetic noise in the input $(u = h_1)$ and the output $(y = h_2)$

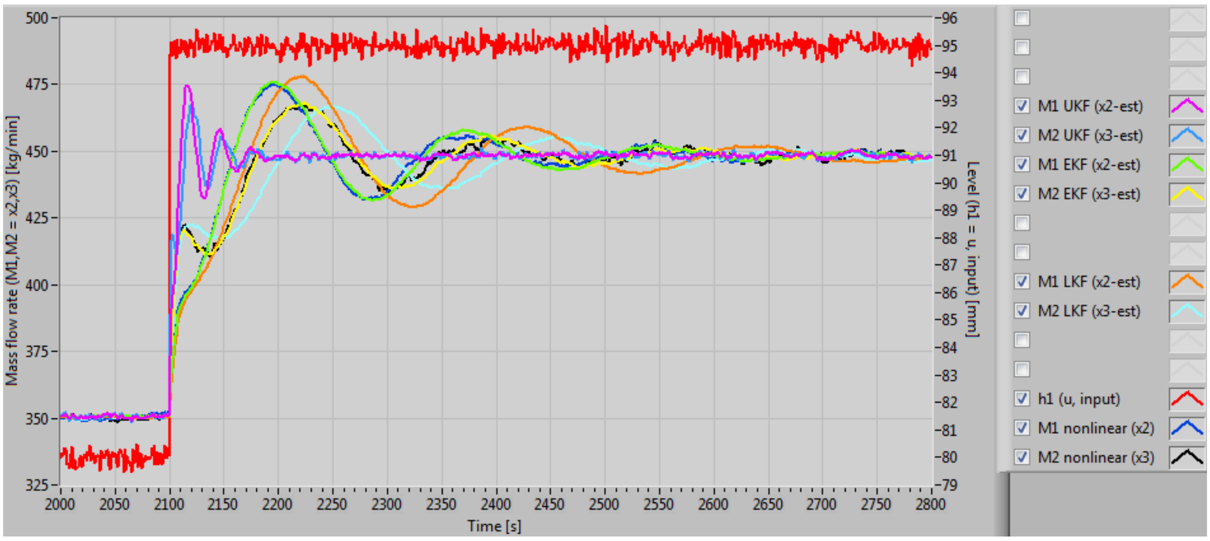

Figure 8.6: Comparison of Kalman filters based on estimation of mass flow rate $(\hat{x}_2 = \widehat{M}_1, \hat{x}_3 = \widehat{M}_2)$. Simulator is excited by synthetic noise in the input $(u = h_1)$ and the output $(y = h_2)$

In above case, the comparison is based on the noisy input and noisy output. For comparison of Kalman filters on a system excited by noise-free input is given in Appendix 9 Part 9B. In all cases, the EKF is better than LKF. UKF is better than EKF and LKF. Figure 8.7 shows the complete picture of performance for all the Kalman filters based on mass flow rate estimation. Input is changed multiple times. All Kalman filters perform ok but the performance of the UKF stands out as it is robust, fast and stable than other version of Kalman filters.

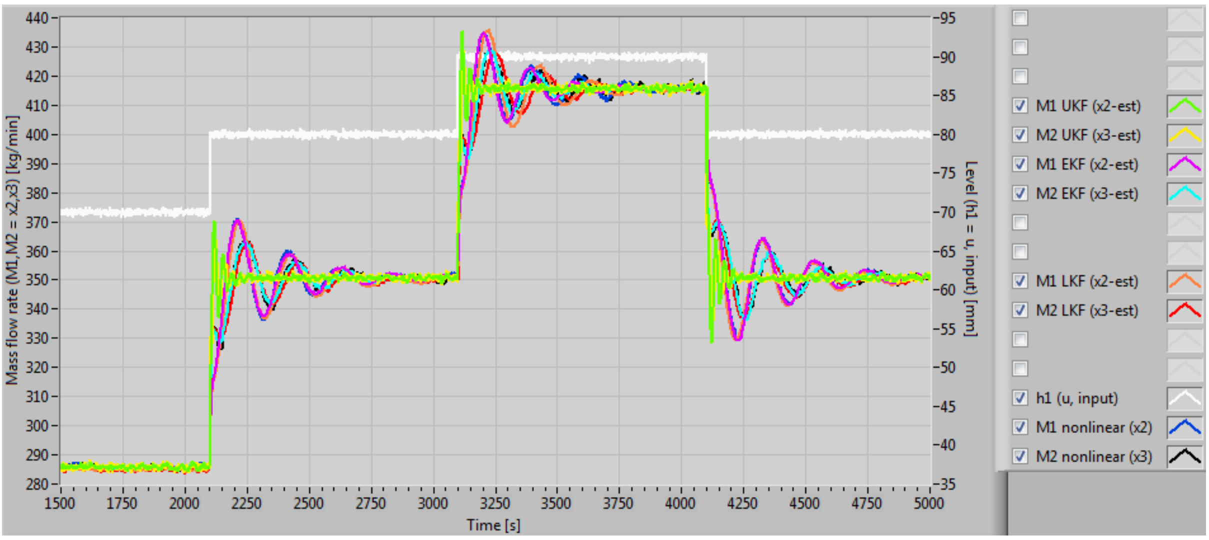

Figure 8.7: Flow estimation using LKF, EKF and UKF when input changes multiple times

8.3.3 Comparison of LKF and linear observer

So far, different types of observers are compared with each other. So is done with the different types of Kalman filters as well. In this section, linear estimators, linear observer and the linear Kalman filter, are compared. The result of comparison is shown in Figure 8.8 and Figure 8.9

based on level estimation and mass flow rate estimation respectively. Both observers are run in parallel with the nonlinear venturi model (simulator). Synthetic noise is added in input and the output of the simulator.

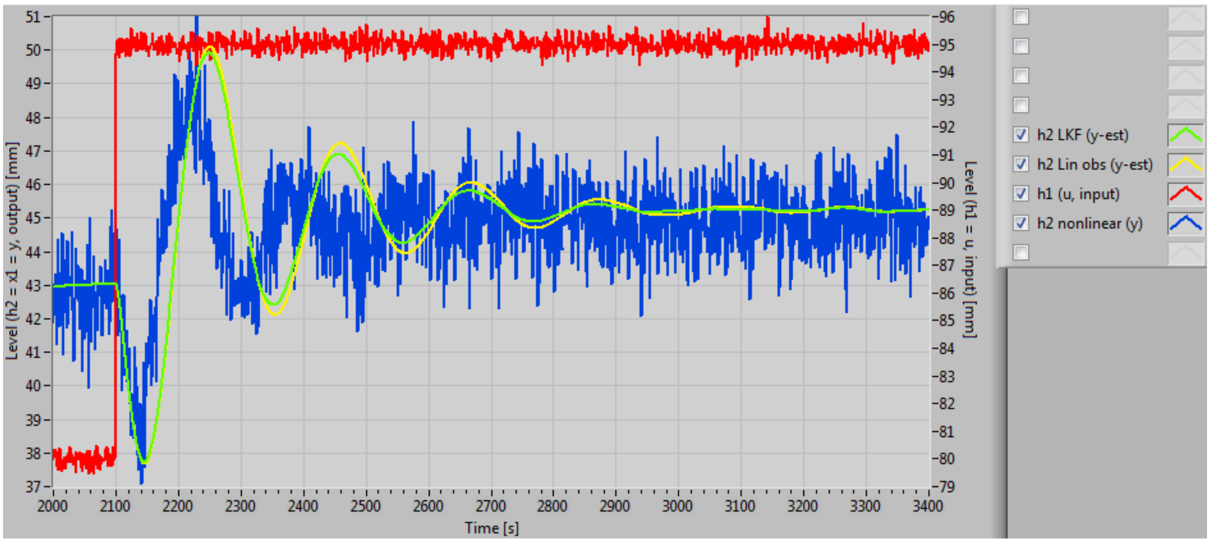

Figure 8.8: Comparison of linear estimators based on level estimation $(\hat{y} = \hat{x}_1 = \hat{h}_2)$. Simulator is excited by synthetic noise in the input $(u = h_1)$ and the output $(y = h_2)$

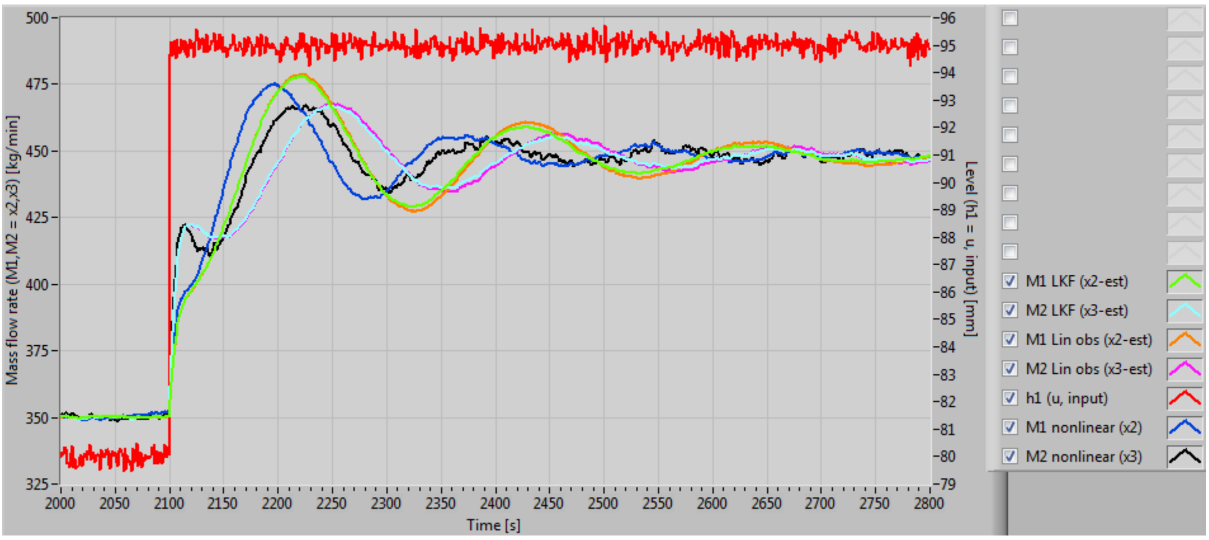

Figure 8.9: Comparison of linear estimators based on estimation of mass flow rate $(\hat{x}_2 = \widehat{M}_1, \hat{x}_3 = \widehat{M}_2)$. Simulator is excited by synthetic noise in the input $(u = h_1)$ and the output $(y = h_2)$

Both types of estimators are able to converge to a steady state value for both level and mass flow rate. The linear observer is placed 1.6 times to the left of the system poles. The LKF has its pole close to the system poles as discussed in section 8.1.3 (Figure 8.2). Thus, the amplitude of oscillation for linear observer is larger. Both estimators are able to filter out the noise significantly. The noise in level estimates are almost non-existence whereas the noise in the estimated mass flow rate is minimal. At steady state, both estimators show some deviation given that they are operating away from the linearization point. In terms of noise handling, LKF is slightly better than the linear observer; which is again, due to the faster poles in linear observer that amplifies the noise. Although LKF performs better in terms of noise handling, the design and implementation of linear observer is easier. Since, LKF needs properly tuned covariance matrices Q and R. The process of tuning these matrices is cumbersome. The price to pay here is the ease of design and implementation to the less noisy estimates. Since the noise level in linear observer is negligible, although worse than LKF, the process of tuning Q and R is avoided. Hence, linear observer is preferable than LKF. For additional comparison of linear estimators excited by noise free input, see Appendix 9 Part 9C.

8.3.4 Comparison of EKF, UKF and semi-nonlinear observer

From previous comparison and discussion, it is shown that nonlinear estimator outperforms their linear counterpart. Comparison of linear version of estimators showed that the estimation is relatively similar. Due to ease of designing linear observer, it is preferred. This section compares the semi-nonlinear observer to the nonlinear version of Kalman filters. Figure 8.10 is the level estimation and Figure 8.11 is the mass flow rate estimation. In both cases, all the nonlinear version of estimators is able to filter out the noise just like their linear counterpart. All estimators are able to converge to a steady state with no deviation which was seen in their linear counterpart. Among nonlinear estimators, UKF performs better. One important observation, though, is that the EKF and semi-nonlinear observer perform exactly same. Even though, the observer is semi-nonlinear, it is as good as non-linear EKF. Based on these facts, the conundrum is to choose between UKF or semi-nonlinear observer. In case of a system with large number of states, semi-nonlinear observer can be easier to implement. In case of highly, nonlinear system, UKF takes the upper ground since it can handle any nonlinear models.

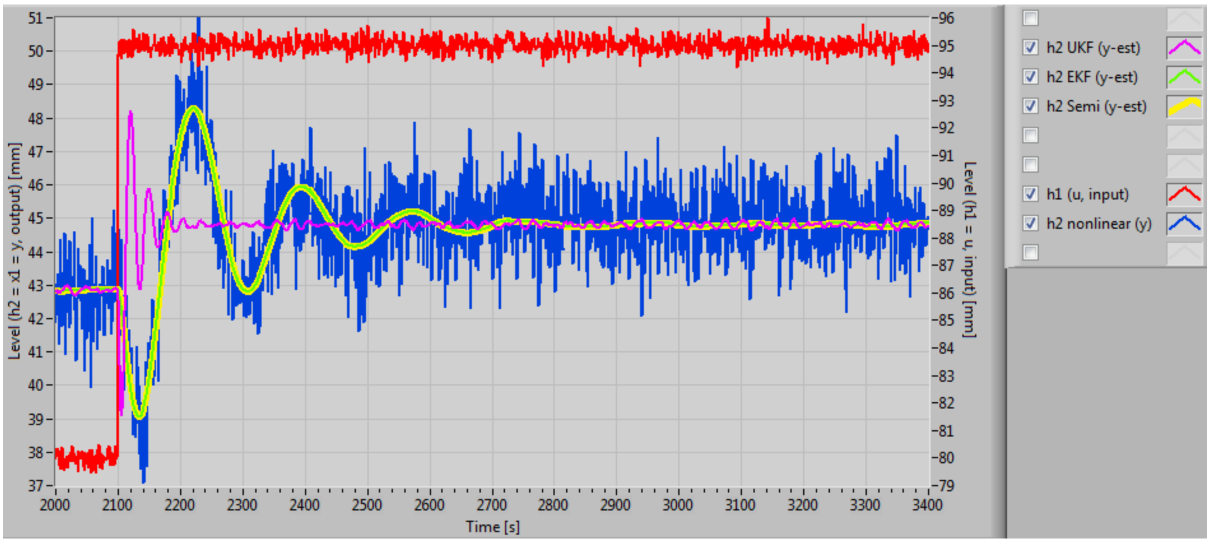

Figure 8.10: Comparison of nonlinear estimators based on level estimation $(\hat{y} = \hat{x}_1 = \hat{h}_2)$. Simulator is excited by synthetic noise in the input $(u = h_1)$ and the output $(y = h_2)$

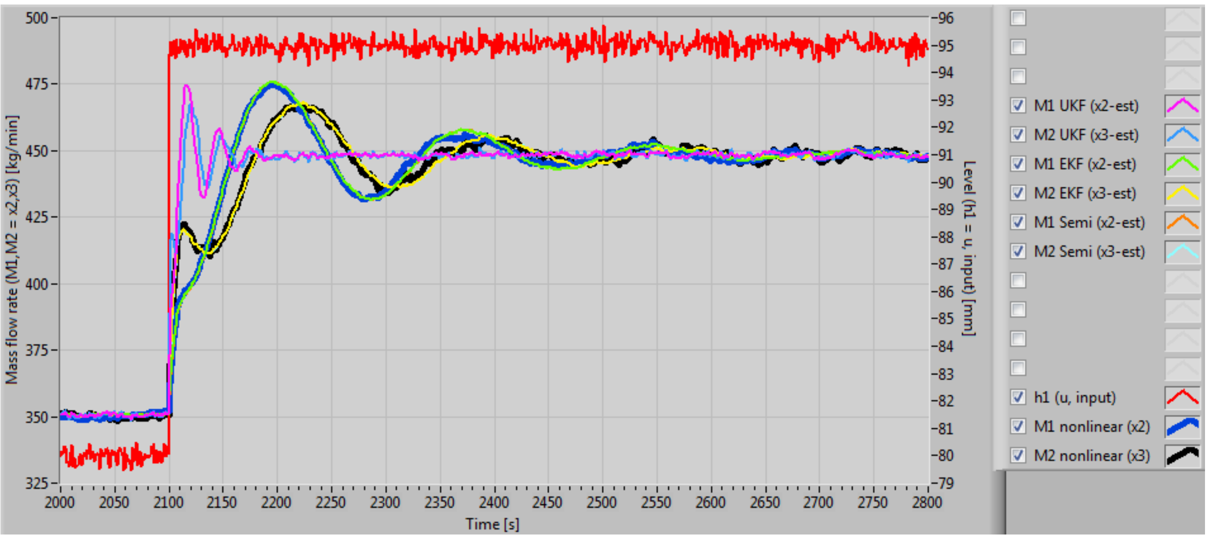

Figure 8.11: Comparison of linear estimators based on estimation of mass flow rate $\left(\hat{x}_2 = \widehat{M}_1, \hat{x}_3 = \widehat{M}_2\right)$. Simulator is excited by synthetic noise in the input $(u = h_1)$ and the output $(y = h_2)$

9 Experimentation and results

So far, the state estimator and observer are implemented on the mathematical model and the result is promising. However, the goal is to implement the observer and estimator on the real system. This chapter casts light on how experiments are performed at the lab. A thorough analysis of the performance of the state estimator on the experimental data is discussed here. Some discussions about the different types of filters to filter out the highly noisy measurements is presented as well.

9.1 Experiment

The experimental setup at the lab is shown in a P&ID in Figure 3.5. The non-Newtonian fluid used for the experiment has a density of $1340 \frac{kg}{m^3}$ as measured by the Coriolis meter. This fluid gives a lot of bubbles at higher flow rates. To remove these bubbles, a mechanical filter is used. Due to the constraint of the pump, the minimum flow rate the data can be acquired is $275 \ \frac{kg}{min}$ and the maximum flow rate possible is $450 \frac{kg}{min}$. Total of four experiments are performed. Each experiment has two setups; up measurements and down measurements. For each setup, the mud is sent through the venturi channel from a minimum flow rate to the maximum flow rate incremented by either 50 or $75 \frac{kg}{min}$ based on the experiment and then from maximum to minimum decremented by the same amount. Increment of 50 or $75 \frac{kg}{min}$ is chosen so that the change in level can be noticed during analysis as well as to stay within the range of allowed mass flow rate. Table 9.1 shows the procedure used for up-down measurements for experiments 3 and 4.

Table 9.1: Experimental setup for up-down measurements at the lab

Exp. #	Up mass flow rate	Log time (min)	# of samples	Down mass flow rate	Log time (min)	# of samples
E3S1	300 350 400	9 9 9	540 540 540	400 350 300	9 9 9	540 540 540
E3S2	275 350 425	9 9 9	540 540 540	425 350 275	9 9 9	540 540 540
E4S1	300 350 400	20 30 10	1200 1800 600	400 350 300	10 30 –	600 1800 –
E4S2	275 350 425	30 20 10	1800 1200 600	425 350 275	10 30 –	600 1800 –

9.1.1 Data acquisition

The data is logged 1 sample per second using a data logging application which is designed to log only one sample for a sensor per second. For all of the experiments, levels at the two positions are measured and the mass flow rate is measured. The position of LT19 and LT18 are $c_1 = 21\ cm$ and $c_2 = 160\ cm$ respectively. The length of between the two sensor positions is $L = c_2 - c_1 = 1.39\ m$. These positions are shown in Figure 3.2. Sensors used to acquire the data are listed below:

- LT19 – radar level sensor
  - Fluid level at the subcritical region.
  - This is referred to as collocation point 1.
  - The level measured here are the input (h_1) to the system.
- LT18 – ultrasonic level sensor
  - Fluid level at the throat section (usually critical region).
  - This is referred to as collocation point 2.
  - The measured levels are the output (h_2) of the system.
- FT14A – Coriolis flow meter
  - Mass flow rate (M) through the discharge pipe that connects that pump and the channel.
  - The measurements are only used for the comparison purpose. The measurements are not used in the observer or model at all.

The setup of the venturi rig and the placement of these sensors at the lab at USN is shown in a P&ID diagram (Figure 3.5).

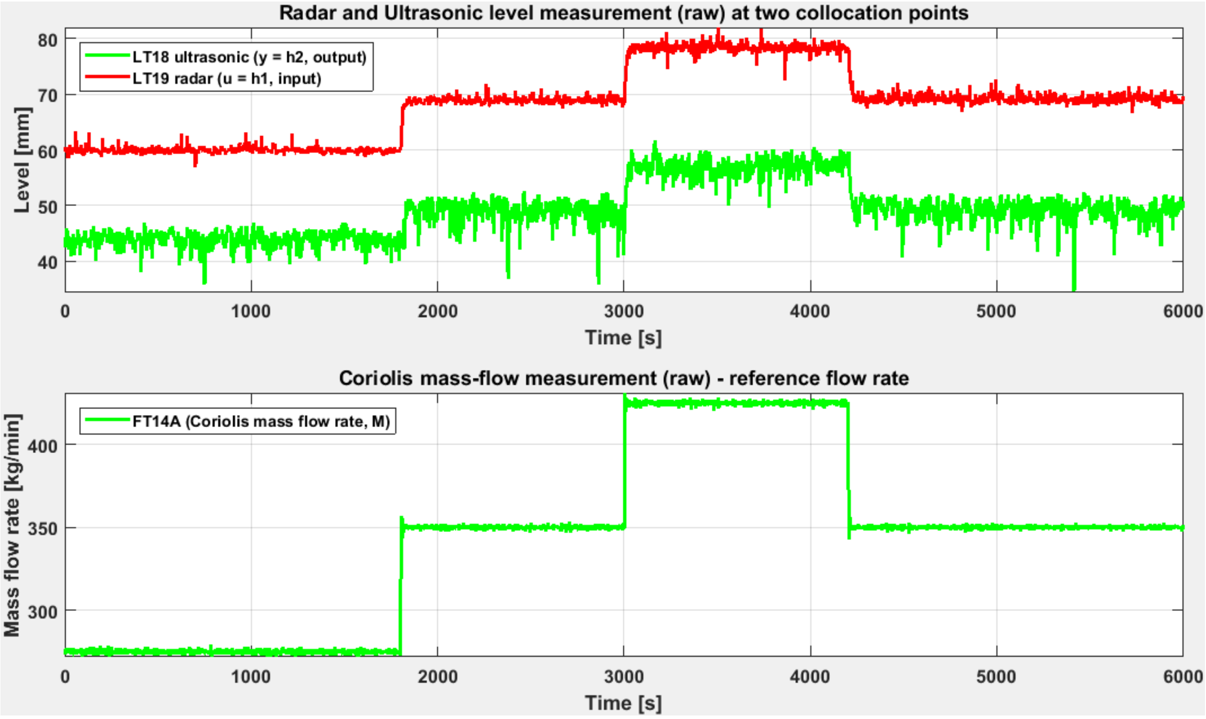

Figure 9.1: Visualization of raw measurements for level at two collocation points (top subplot) and mass flow rate measurement (bottom subplot)

Level and mass flow measurements using up-down technique is displayed in Figure 9.1. The figure shows the unprocessed raw data from the experiment performed at the lab. Based on the raw data plot, the radar sensor is less immune to noise than the ultrasonic sensors. The noisy level measurement by the ultrasonic sensor is partly due to the fact that the point of measurement (collocation point 2) lies in the throat section where the velocity of the fluid starts to increase rapidly which causes some turbulence in the region.

9.2 Filters

Measurements are noisy as seen in Figure 9.1. Noisy measurements give noisy estimates. It is, therefore, a good idea to filter out the noise prior to sending the data through to the estimator, Kalman filter or Luenberger observer, as shown in Figure 6.1. In this thesis work, three of the filters are investigated closely. The filters used are:

- Weighted moving average filter (WMAF)
- Lowpass filter (LPF)
- Median filter

Median filter is used directly from the built-in function in MATLAB called “medfilt1.m”. WMAF and LPF are designed from the ground up.

9.2.1 Weighted moving average filter

For WMAF, weights have been assigned, $w_1, w_2, \dots, w_n$. The sum of these weights is equal to 1 (or 100%) as given by (10.10). The purpose of these weights is to weight the measurements. The current measurements are the most important ones. The oldest measurements are the least important ones. Hence, the current measurements are weighted more than the past measurements. Formulation of WMAF is given by (9.1) – (10.10).

$$\sum_{i=1}^{n_w} w_i = 1 = 100\% \tag{9.1}$$

$$N = \sum_{j=1}^{n_w} n_j \tag{9.2}$$

$$x^k = \boldsymbol{x}(k{:}N + k) \quad k = 1,2,3,\dots,\infty \tag{9.3}$$

$$\bar{x}_k = \sum_{j=1}^{n_w} \frac{w_j}{n_j} \sum_{i=1}^{n_j} x_i^k \tag{9.4}$$

Where, N is the number of samples to consider. n_w is the number of measurement sets N samples are divided into. n_j is the number of data samples for a given set and w_j is the weight for the set. Number of data samples for a given set can be different. $\bar{x}_k$ is the weighted average of the N samples for a given time instance k. For venturi rig N is chosen to be 48 samples, n_w is chosen to be 3. n_1 is chosen to be 8 samples, n_2 is chosen to be 16 samples and n_3 is chosen to be 24 samples as shown in Table 9.2. The largest subscript number corresponds to the most recent measurements. $w_1 = 0.2 = 20\%$, $w_2 = 0.3 = 30\%$ and $w_3 = 0.5 = 50\%$ is given as weights for each set. k is shifted by 1 for when a new measurement is available, $k \coloneqq k + 1$.

Table 9.2: Distribution of data samples for weighted moving average filter

$N = 48$ divided into $n_w = 3$			$w = 1$ (100%) divided into $n_w = 3$		
n_1	n_2	n_3	w_1	w_2	w_3
8	16	24	$0.2 = 20\%$	$0.3 = 30\%$	$0.5 = 50\%$

9.2.2 Time constant lowpass filter

Transfer function for a first order time constant lowpass filter is given by (10.10).

$$\frac{y_f(s)}{y(s)} = \frac{1}{T_f s + 1} \tag{9.5}$$

Where, y_f is the filtered output, y raw signal as an input to the filter and T_f is the filter time constant. Equation (10.10) can be written as an ODE using inverse Laplace transformation as:

$$y_f(s) T_f s + y_f(s) = y(s)$$

$$T_f \mathcal{L}^{-1}\big(y_f(s)s\big) + \mathcal{L}^{-1}\left(y_f(s)\right) = \mathcal{L}^{-1}\big(y(s)\big)$$

$$T_f \dot{y}_f(t) + y_f(t) = y(t)$$

$$\dot{y}_f = \frac{(y - y_f)}{T_f} \tag{9.6}$$

Equation (10.10) is the first order ODE for continuous time - time constant lowpass filter. Using backward Euler differentiation method, a discrete form of the LPF is determined. The reason for using backward Euler differentiation method is because of the fact that only the previous measurements are available.

$$\dot{y}_{f_k} = \frac{y_{f_k} - y_{f_{k-1}}}{T_s} \tag{9.7}$$

$$\frac{y_{f_k} - y_{f_{k-1}}}{T_s} = \frac{(y_k - y_{k_t})}{T_f}$$

After simple algebraic simplification, the final version of discrete LPF is given as:

$$y_{f_k} = a y_k + (1 - a) y_{f_{k-1}} \tag{9.8}$$

$$a = \left(\frac{T_s}{T_f + T_s}\right) \quad 1 - a = \left(\frac{T_f}{T_f + T_s}\right) \tag{9.9}$$

A simple rule of thumb for choosing T_f is:

$$T_s \leq \frac{T_f}{5} \tag{9.10}$$

9.2.3 Filter comparison based on real measurements

Lowpass filter (LPF) and weighted moving average filter (WMAF) is implemented in MATLAB. LPF and WMAP are applied to the raw level measurements only since the mass flow rates measured by the Coriolis are only used for comparison purpose. A built-in median filter from MATLAB, "medfilt1.m" is also compared here. The order of median filter is 66 which means that the median from 66 samples are calculated. The number of samples corresponds to the one minute of measurements with 6 extra samples to handle the measurement delays. Since the sampling time (data logging time) for the experiments is 1 second, the LPF time constant should be at greater than 5 seconds according to equation (10.10). In this case, 16 second is used. The parameters for WMAF is given in Table 9.2. After assigning necessary parameters for each filter, they are applied to the real measurements. The result of the filter application on raw data is shown in Figure 9.2. Median filter is aggressive. WMAF is slower than LPF and MEDF. All of these filters are good enough to use in this project as all of them remove high frequency noise. MEDF is to aggressive which makes it difficult to analyze the filtering ability of the estimators. WMAF is slow. Thus, LPF is chosen for the rest of the thesis work. Therefore, LPF is also implemented in LabVIEW. The introduction of filter removes the noise; however, it comes at a cost. The cost to bear when using a filter is that it introduces a delay in the system and hence the state estimates are delayed.

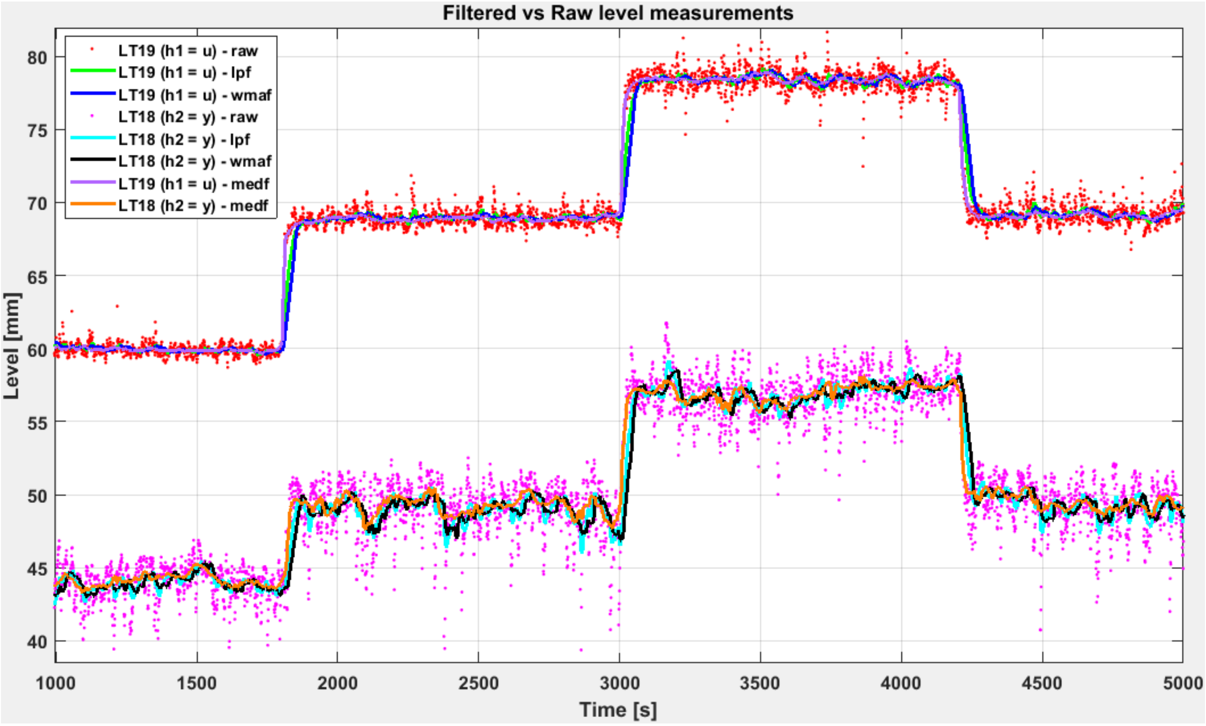

Figure 9.2: Deployment of different filters on the raw level measurements (real data)

9.3 Comparison between the nonlinear model and the real system

Until this stage, the mechanistic model of flow through venturi flume is formulated to represent the real system. State estimators are designed based on the dynamic model. Even though the model is stable, controllable and observable, it is important to compare it with the real system to determine its validity. Thus, it is natural to perform open loop simulation and verify the dynamic model of the system. The measured data represents the real system. In an open loop simulation, the simulator based on dynamic model is run in parallel with the real system. The simulator is excited by the real input level measurement. The states of the simulator are compared directly with the states of the real system. States of the real system are the level measured at the collocation point two (also the output of the system) and the Coriolis mass flow measurements as shown in Figure 9.1. The result of the open loop simulation is shown in Figure 9.3 and Figure 9.4 for raw input level measurements and filtered input level measurement to excite the simulator respectively. The upper subplot on both figures shows the level measurements at two collocation points, one (h_1) for representing the input to the system and the other (h_2) to represent the output of the system. The lower subplot represents the mass flow measurements at the aforementioned points. There is only one Coriolis flow meter used for flow measurements. From these figures, it is clear that the predicted mass flow rate is a good representation of the real mass flow rate. However, the level prediction is way off. This is due to imprecise and uncertain model parameters. It is also important to realize that the dynamic model is determined using two-point collocation method as discussed in section 4.2.1. The model precision can be increased by using higher order interpolating polynomials i.e., using multiple orthogonal collocation points or PDE itself. Thus, open loop simulation is inadequate. This is the reason why estimators are designed.

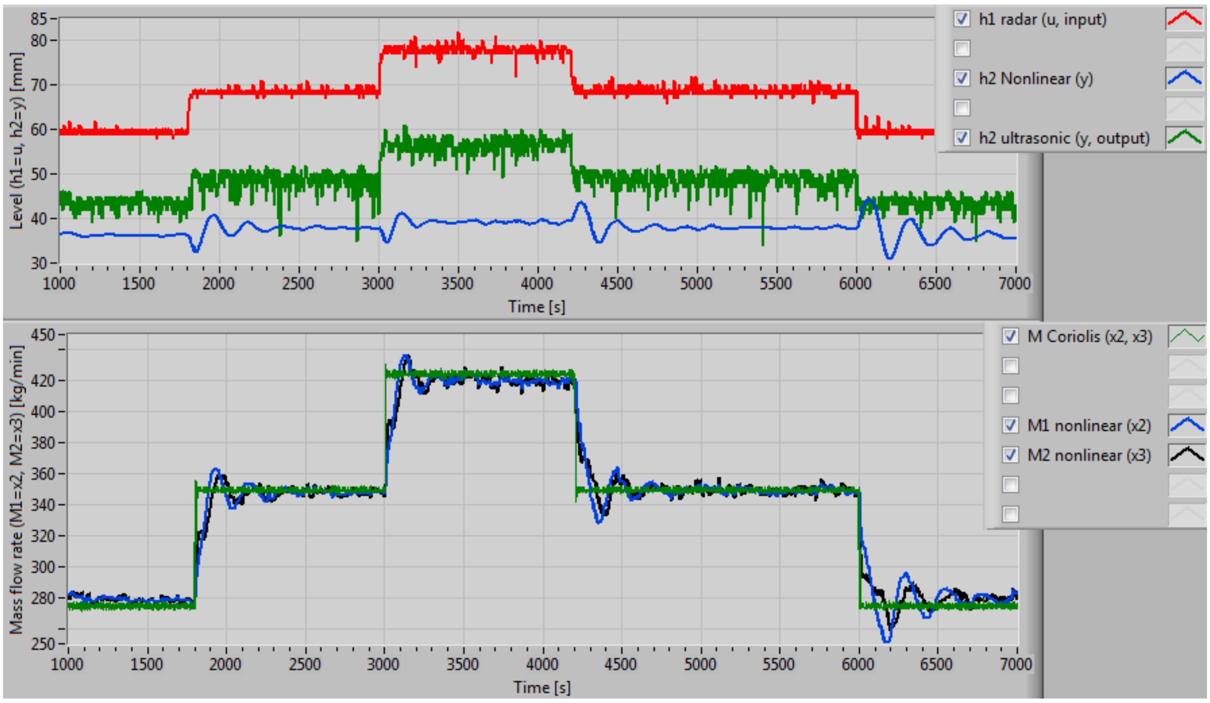

Figure 9.3: Open loop simulation of real system and the model in parallel with unfiltered (raw) input and output level measurements

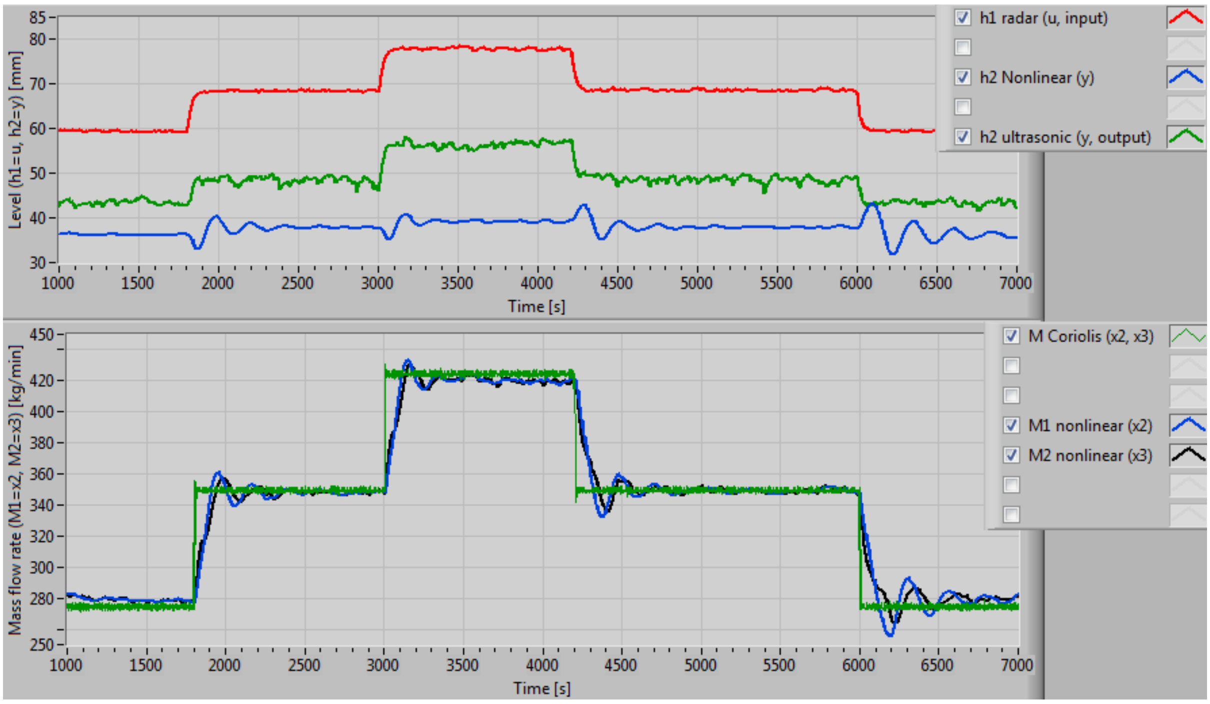

Figure 9.4: Open loop simulation of real system and the model in parallel with filtered input level and output measurements

9.4 State estimator comparison on real system

The main goal of this thesis is to see if mathematical model of the flow through venturi flume gives satisfactory flow estimation thus replacing expensive mechanical flow measuring devices. The estimated flow can be used as a signal for kick-loss detection and much more. This section discusses the implications of the flow estimation using linear and nonlinear estimators. In earlier chapters, estimators are used on the dynamic model (simulator). The result is promising. This is usually the case that the estimators work fine with simulator but with the real system, the story is different. The model does not represent the real system with absolute perfection due to uncertain model parameters and disturbances, as elaborated in previous section. As with the simulator, linear and nonlinear estimators are applied to the real system. The result is discussed in subsequent sub-sections below.

9.4.1 Comparison between LKF and Linear observer

LKF and linear observer are applied on the real system. The level measured by LT18 is used as the feedback to the estimators. The level measured by LT19 is used as the input to the system. The result of estimation is shown in Figure 9.5 and Figure 9.6 for raw input and the filtered input signal respectively. Both estimators are able to minimize the error between model predicted states and the real states. In terms of level estimation, both estimator produces good result given how imprecise the model is according to the open simulation of Figure 9.3. In terms of mass flow rate estimation, LKF performs better. The linear observer has faster poles which amplifies the noise in the system. LKF is able to filter out the noise in the estimates due to finely tuned noise covariance matrices Q and R. When the LPF is applied to the input and output measurements, the noise amplification by the observer is reduced. However, there is a deviation in all cases when operated away from the operating point. Therefore, investigating nonlinear estimators is important to see if the deviation due to linearization can be eliminated. It is also important to note that the filtered input signal using LPF gives more robust and reliable estimates (Figure 9.6). Thus, by filtering the high frequency noise in the input and the output using LPF prior to feeding them to the estimator, the performance of estimators can be improved. Although, the introduction of LPF introduces a small delay in the state estimates.

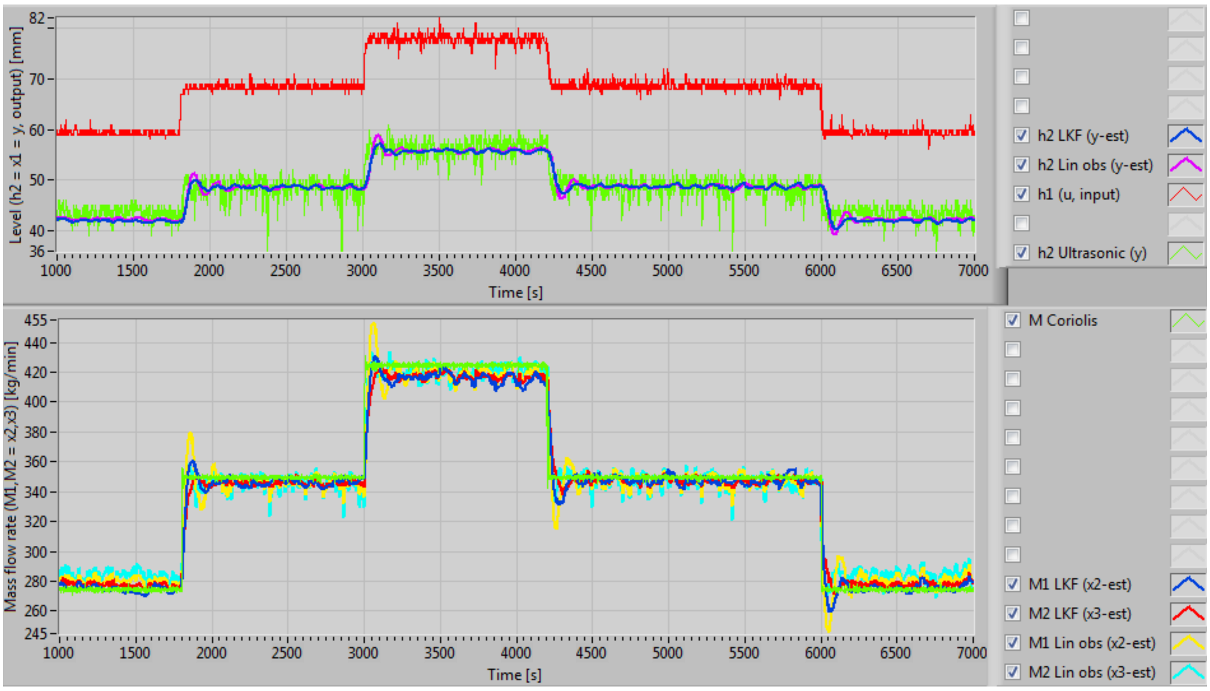

Figure 9.5: Comparison of linear estimators based on real raw (unfiltered) measurements

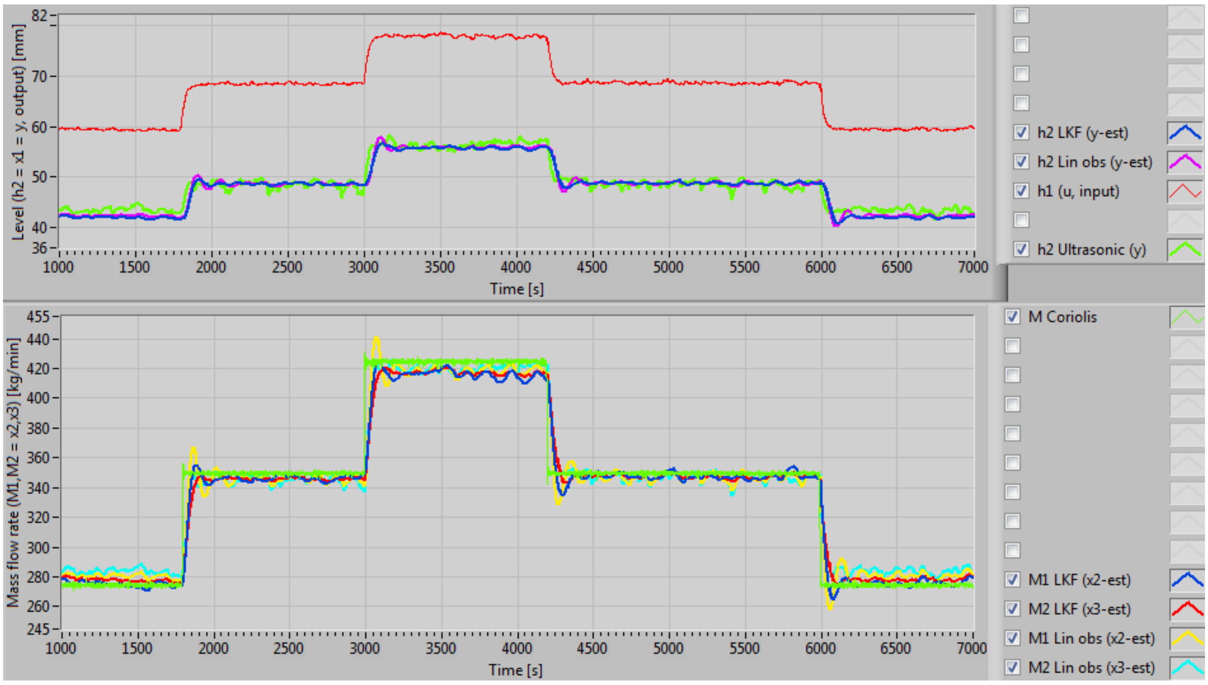

Figure 9.6: Comparison of linear estimators based on real filtered measurements

9.4.2 Comparison between EKF and UKF

The EKF and UKF are excited by raw input level measurements (h_1) measured at the collocation point 1, c_1. The feedback data provided to the Kalman filters is the raw output level measurements (h_2) measured at the collocation point 2, (c_2). The estimation result of EKF and UKF on raw (unfiltered) real system is shown Figure 12.2 and Figure 12.3 respectively.

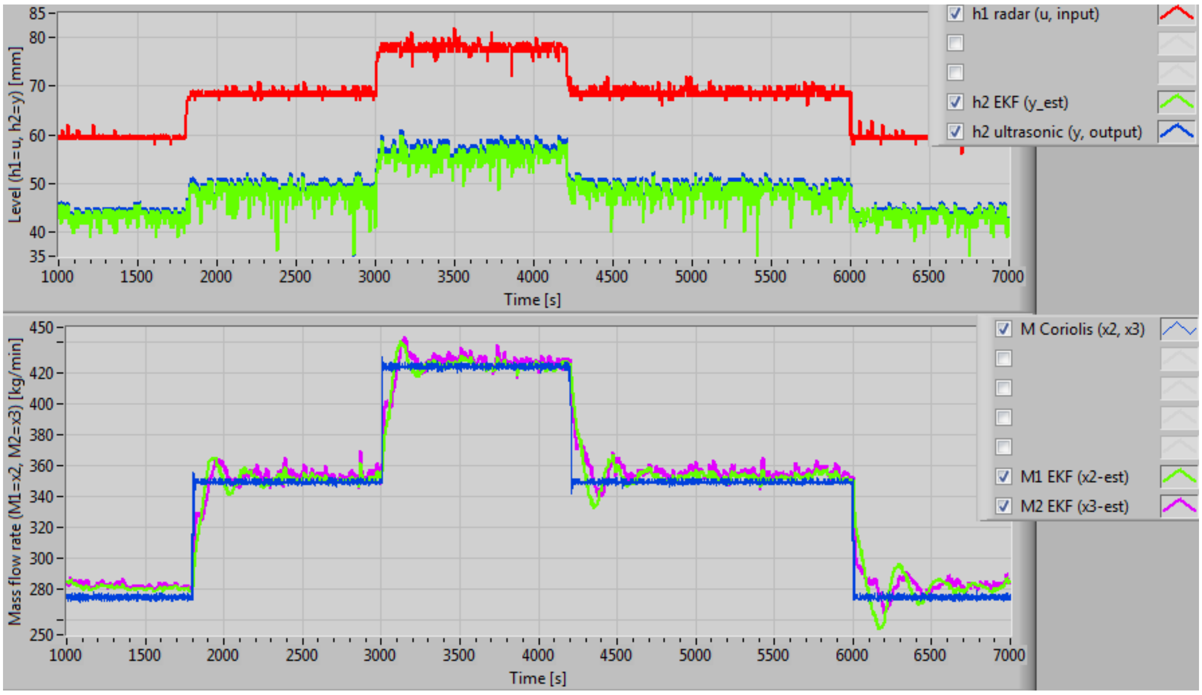

Figure 9.7: State estimation using EKF with feedback from the real raw measurements

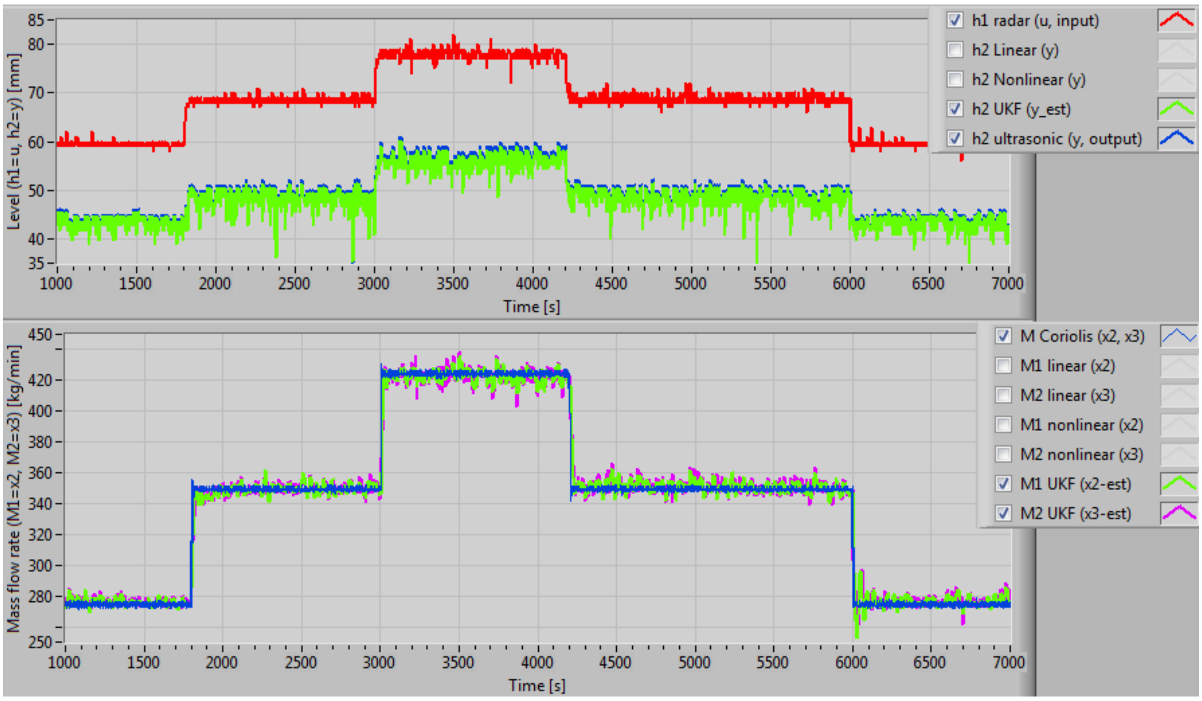

Figure 9.8: State estimation using UKF with feedback from the real raw measurements

Analysis of Figure 9.7 and Figure 9.8 gives an idea on the performance of nonlinear Kalman filters. The error between the mass flow rate estimated by UKF and the Coriolis measurements is smaller than the error between mass flow rate estimated by EKF and the Coriolis measurements. In case of EKF, there seems to be some deviation in the mass flow estimates for each flow rate measured by Coriolis flow meter. The deviation in UKF is negligible whereas, the deviation in EKF is significant. Estimates given by UKF oscillate around the actual measurements. At higher flow rate, due to near-turbulent flow occurring in venturi flume during experiments, the level measurements at both points are noisy. This is partly due to the bubbles that are produced when using the non-Newtonian fluid. This is reflected in the flow estimation as well.

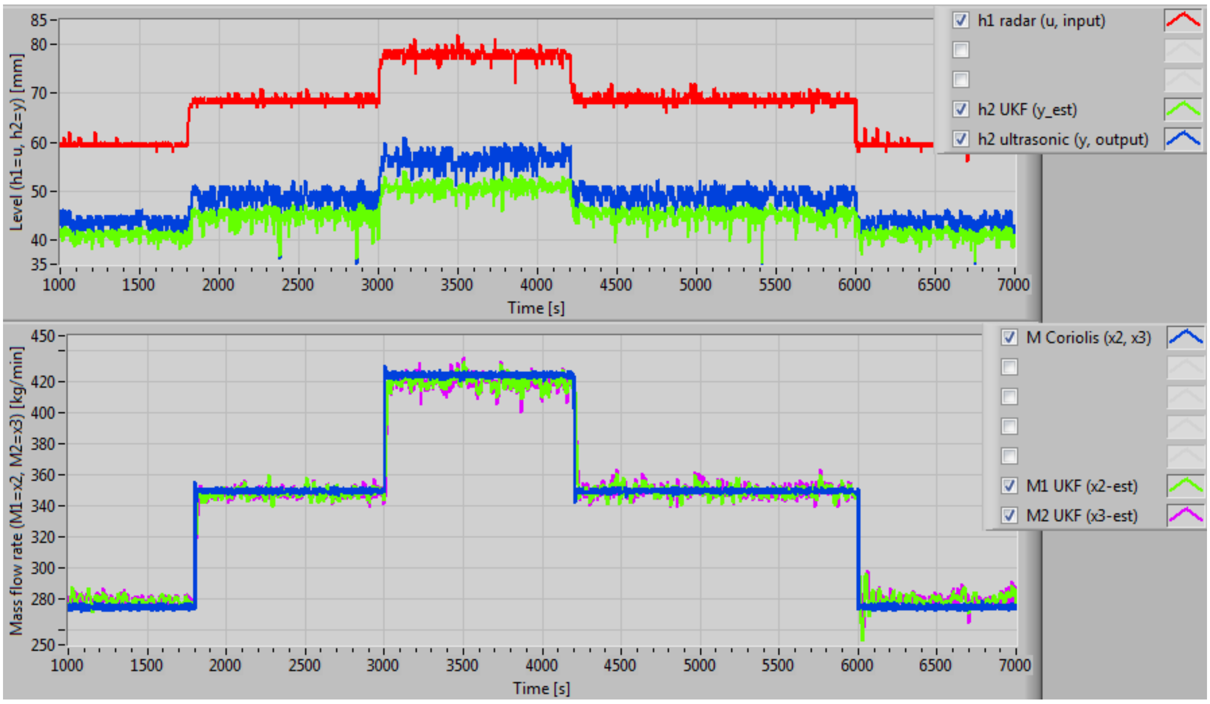

Figure 9.9: Result of adjusted Q and R to minimize the noise in level estimates using UKF

When it comes to level estimates, both EKF and UKF give similar results. Notice that the level estimates are too noisy. This is because the level predicted by the dynamic model is way off the actual level measured at c_2 (see Figure 9.4). To minimize the noise in the estimates, noise covariance matrices must be adjusted. When doing so, the noise is minimized but the deviation of the estimated states and the real states increases. Trying to minimize the deviation between the estimated states and the real states, the noise reappears. Adjustment to Q and R reduces noise in the level estimates but the deviation is increased (see Figure 9.9). Further adjustment gives almost no noise in the level estimates but the deviation between the estimated states and the real state is maximized. This is true for both EKF and UKF. The solution is to use the LPF to remove high frequency noises in the input and the output. This technique undermines the very concept of the Kalman filter, which is to estimate the states as well as filter out the noise. When the model is so much deviated, filtering out the output is a very small price to pay. It is not possible to keep the filtering properties of Kalman filters and give precise estimates for a deviated model. In addition to that, the goal of this thesis work is to estimate the mass flow rate. So, filtering out the input and output measurements is reasonable if the flow estimates can be improved. Thus, LPF is applied to both input and output level measurements. The filtered signals are then used in the estimators. The result of applying filtered input and output to nonlinear Kalman filters is shown in Figure 9.11 and Figure 9.12. In both cases, the noise is reduced and deviation between level estimates and the real level measurements is minimized. The estimated mass flow rate, in any case, is undeterred. Even with the introduction of slight delay in the state estimates due to the use of LPF, UKF converges faster than EKF.

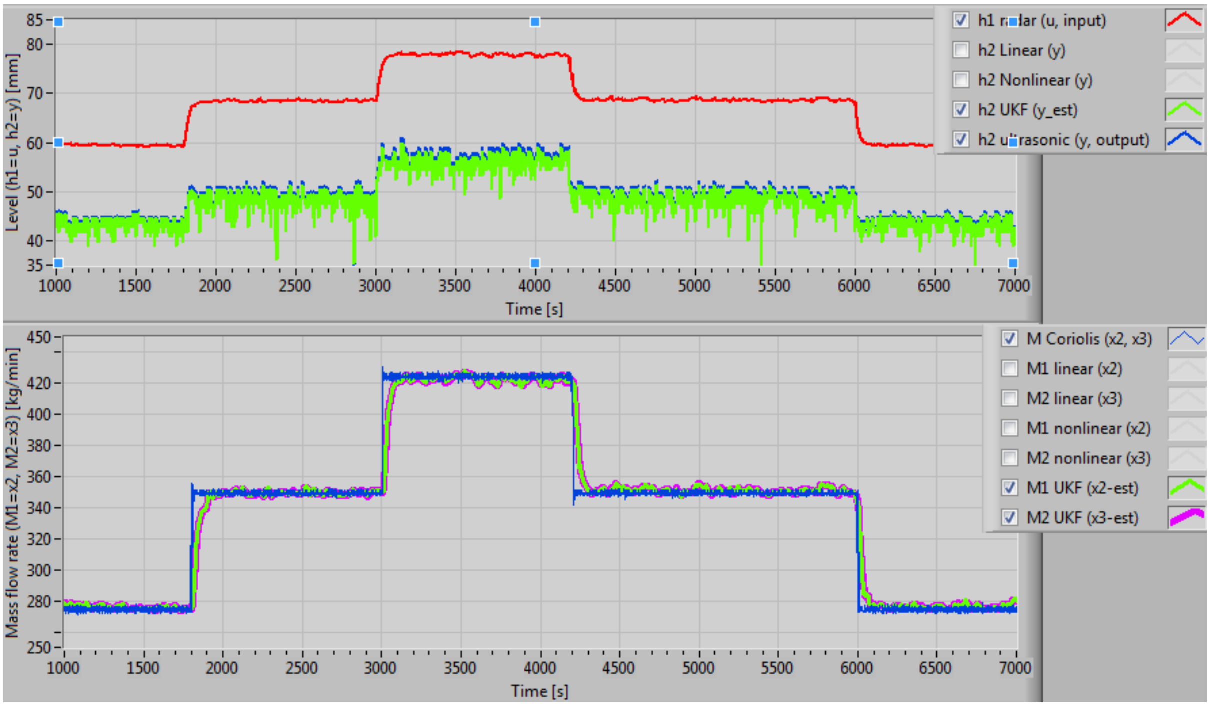

Figure 9.10: State estimation using UKF with filtered input and output measurement data

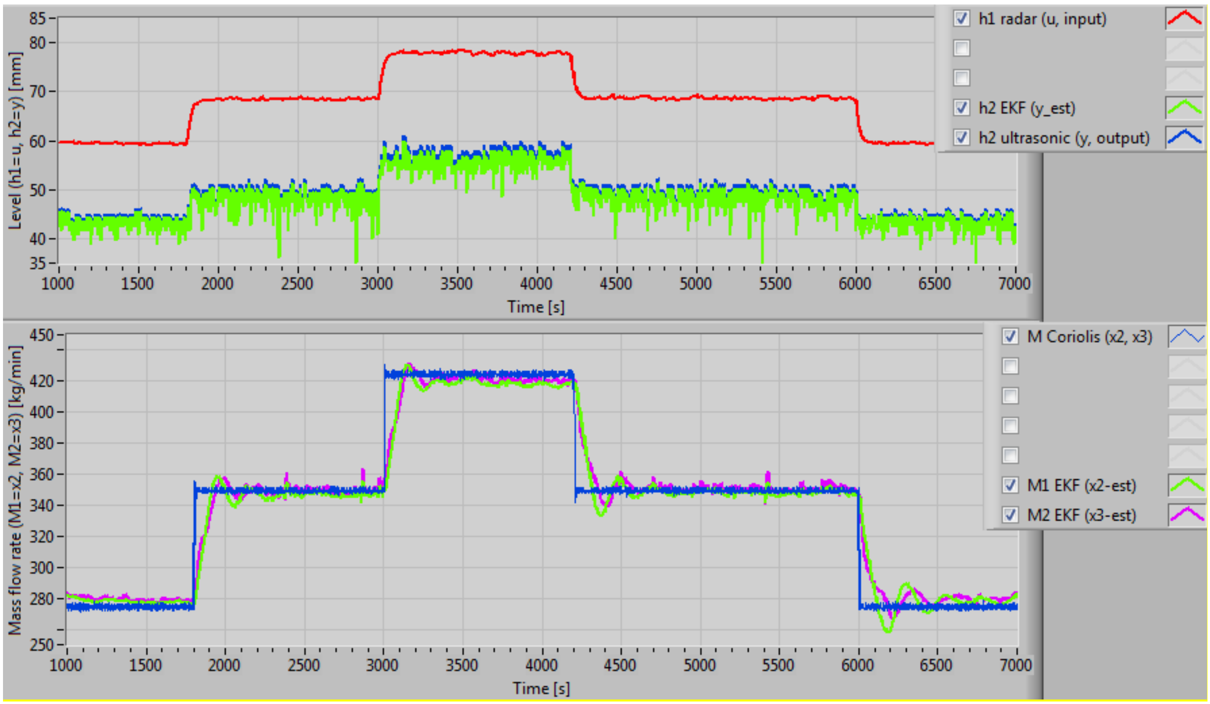

Figure 9.11: State estimation using EKF with filtered input and output measurement data

9.4.3 Comparison between linear and nonlinear Kalman filters

UKF and EKF are compared in previous section. The verdict is that UKF gives better, reliable and robust estimates than EKF. As with EKF and UKF, an LPF is applied to the input and output before feeding the signals to the LKF. The result of filter on LKF is shown in Figure 9.12. Comparing Figure 9.12 with Figure 9.10 and Figure 9.11, it is simple to conclude that LKF gives reliable estimates around the operating point. As the operation moves away from the operating point, the accuracy of estimates reduces. Based on faster response, less deviation between estimated states and real system and robustness of the estimator, UKF is superior to LKF or EKF. Hence, UKF is preferable.

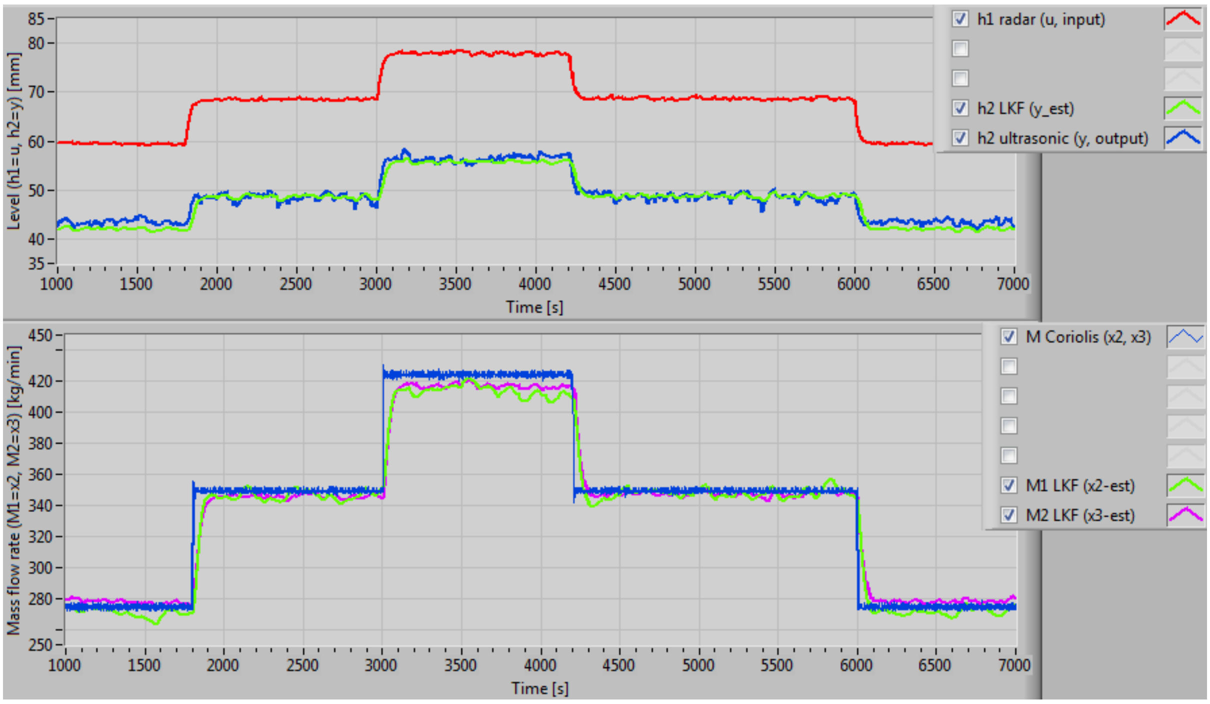

Figure 9.12: State estimation using LKF with filtered input and output measurement data

SECTION – III

DISCUSSION, CONCLUSION AND RECOMMENDATIONS FOR FUTURE WORK

10 Discussion

This section discusses the limitations and possibilities of the methods used in this report. Some insight to the future work on flow estimation is also suggested in this chapter. An alternative methods of flow estimation using artificial neural network (ANN) is also proposed here.

10.1 Limitation of the proposed flow measuring technique

Although, Coriolis is expensive, it not only measures the flow rate (both mass and volumetric), it also measures the density of the fluid and, in some cases, viscosity. The proposed mathematical model is restricted to estimating the flow rates (both mass and volumetric). Some modern flow meters come equipped with temperature sensor as well. This information can be harnessed to analyze the condition in the bottom hole. It is also important to remember that the physical model presented here assumes that the density of the returning fluid is constant. This is unlikely to be the case in the real-world scenario. Two-point collocation method can be improved by using higher order Lagrange interpolating polynomials to represent the PDEs. Two-point ODEs are the simplest form of the nonlinear model. Thus, there is a room for improvement. In case of level estimation, the model does not depict the real system. This must be re-calibrated if the two-point ODEs are to be used in further studies.

10.2 Use of state estimation in control system

In oil-industry, flow estimation is important to detect kick-loss phenomenon for safe operation. The linear and nonlinear estimators can do the job. However, this is not the only use of estimators such as Kalman filters or Luenberger observer. The estimated state can be used further in designing a control signal for mud-pump. An estimator can also help in designing a predictive controller such as MPC or aid in designing an optimal feedback controller such as PID. This section discusses the use of state estimator in a control system.

Let’s start with a brief introduction to a controller. A simple controller is a machine that generates a control signal, digital or analog, based on the feedback data from available sensor measurements, again analog or digital. An optimal control signal, therefore, depends on the measured states as given by a relation (10.1). The control signal is used to excite the system (10.2) with the help of an actuator to bring the system from an initial state to a desired final state. Due to lack of sensors to measure some of the states, a state estimator is used. Estimated states are then fed back to the controller. Using the available measurements and the control signal for a given time instance, states are estimated. The estimated states can then be used to design a suitable controller such as MPC or PID. The controller generates a control signal based on a relation (10.10). A block diagram of a closed loop feedback control system with state estimator is given by Figure 10.1.

$$u = -Gx \tag{10.1}$$

$$\dot{x} = f(x, u, \theta, t) \tag{10.2}$$

$$u = -\hat{G}\hat{x} \tag{10.3}$$

Where, G is a state feedback gain matrix (time invariant constant matrix) and $\hat{G}$ is the new state feedback gain matrix based on the estimated states

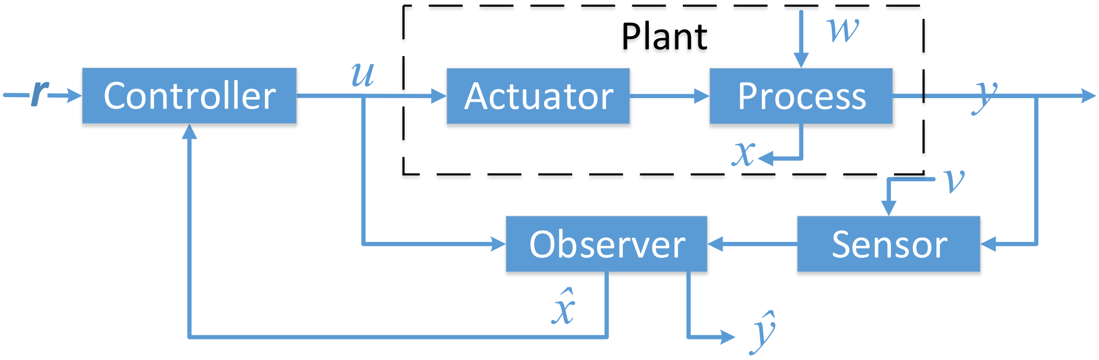

Figure 10.1: State feedback control system with estimated states

10.3 Reduced order observer

In most of the system, only some of the states are measured. For such system, it makes sense to estimate only the states that are unmeasured. This means that the full-state vector is decomposed into measured and unmeasured states. The observer is designed only for the unmeasured states. Thus, the name reduced order observer. The reason for doing this is that computation time for estimation is reduced if the system order (number of states), n_x is large. For computational efficiency, reduced order observer is preferable. The derivation of a reduced order observer is given in Appendix 10. A brief summary of the reduced order observer is presented here. Consider a state vector x with n_x states. It is composed of measured states, x_p and unmeasured states x_q. Thus, x can be decomposed as shown in (10.10).

$$\dot{x} = \begin{bmatrix} \dot{x}_p \\ \dot{x}_q \end{bmatrix} \quad x = \begin{bmatrix} x_p \\ x_q \end{bmatrix} \tag{10.4}$$

A linear state space model and linear state observer in decomposed form looks like:

$$\dot{x}_p = A_{pp}x_p + A_{pq}x_q + B_p u \quad \dot{\hat{x}}_p = A_{pp}\hat{x}_p + A_{pq}\hat{x}_q + B_p u + L_p y - L_p\hat{y} \tag{10.5}$$

$$\dot{x}_q = A_{qp}x_p + A_{qq}x_q + B_q u \quad \dot{\hat{x}}_q = A_{qp}\hat{x}_p + A_{qq}\hat{x}_q + B_q u + L_q y - L_q\hat{y} \tag{10.6}$$

$$y = C_p x_p + C_q x_q \quad \hat{y} = C_p\hat{x}_p + C_q\hat{x}_q \tag{10.7}$$

Where, x_p is a vector of measured states and x_q is a vector of unmeasured states, n_x is the number of total states (system order), $n_p = n_y$ is the number of measured states and n_q is the number of unmeasured states.

$$n_x = n_p + n_q$$

After performing some algebraic manipulation, following observer state equations in reduced order form is determined.

$$\dot{\hat{z}} = (A_{qq} - L_q A_{pq})\hat{z} + (B_q - L_q B_p)u + (A_{qq}L_q - L_q A_{pq}L_q + A_{qp} - L_q A_{pp})x_p \tag{10.8}$$

Where,

$$z \overset{\text{def}}{=} x_q - L_q x_p, \qquad \hat{z} \overset{\text{def}}{=} \hat{x}_q - L_q x_p, \qquad \dot{\hat{z}} \overset{\text{def}}{=} \dot{\hat{x}}_q - L_q \dot{x}_p, \qquad \hat{x}_q \overset{\text{def}}{=} \hat{z} + L_q x_p$$

Thus, the unmeasured states are given by:

$$\hat{x}_q = \hat{z} + L_q x_p \tag{10.9}$$

The error dynamic for reduced order observer is given by (10.10).

$$\dot{\varepsilon}_q = \left(A_{qq} - L_q A_{pq}\right)\varepsilon_q \tag{10.10}$$

Where,

$$\varepsilon_q \overset{\text{def}}{=} x_q - \hat{x}_q$$

If the error dynamic is asymptotically stable, then the observer is stable. For the error dynamic to be asymptotically stable, matrix $\left(A_{qq} - L_q A_{pq}\right)$ must be Hurwitz.

The reduced order observer is ready to be implemented in a programming language of choice. For venturi system, the level at c_2 is measurable state. Hence the number of measured states is 1. The number of unmeasured states is two, mass flow rates at two collocation points. Therefore, the reduced order observer state takes the form of:

$$\dot{x}_p = \left[\dot{h}_2\right] \quad \dot{x}_q = \begin{bmatrix}\dot{M}_1 \\ \dot{M}_2\end{bmatrix} \quad \dot{\hat{x}}_q = \begin{bmatrix}\dot{\hat{M}}_1 \\ \dot{\hat{M}}_2\end{bmatrix}$$

10.4 Artificial Neural Network

The world is moving faster than ever when it comes to new technologies. The talking point in today's tech-world, everyone's lips is ANN. In this section, a simple back propagation-based ANN algorithm is used. Rather than going into the math behind ANN, a built-in MATLAB tool is used.

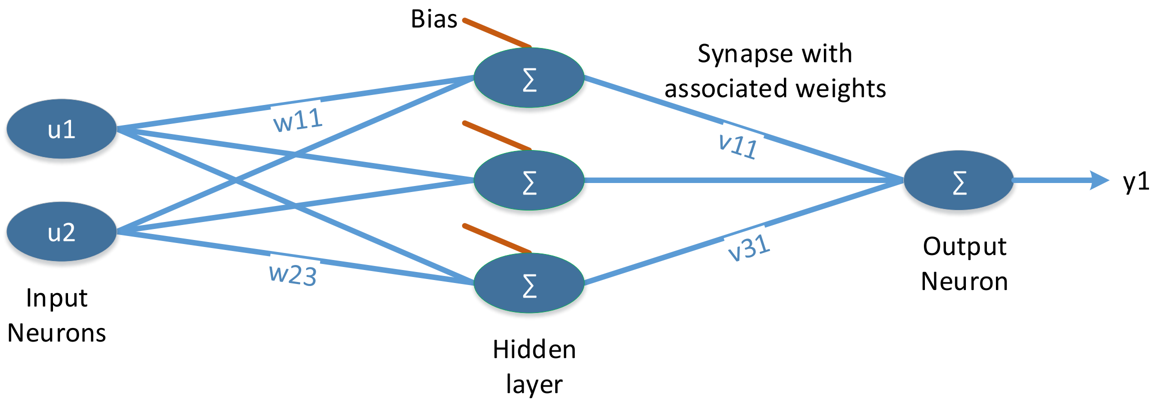

Figure 10.2: A simple sketch of artificial neural network

The main overview of ANN consists of input neurons, synapse with associated weights, hidden layer and output neurons. An ANN with two input neurons, three hidden neurons and one output neuron is shown in Figure 10.2. The value of each input neuron is weighted with (multiplied by) an associated weight. Each neuron in hidden layer sums the weighted values from the input neuron and sends the sum through a sigmoid function to scale the value between 0 and 1. The same procedure follows to the output neuron. The sum of the weighted values from previous layer is passed through the sigmoid function again. The result is the predicted output. In backpropagation algorithm, the predicted output is subtracted from the actual output and the error is then propagated backward in the network to adjust the weights. This is similar to an optimization (error minimization) problem as discussed in section 5.2.5.

For venturi model, two input neurons, two hidden neurons and one output neurons are selected. ANN tool of MATLAB initializes the weights randomly Input neurons are represented by the level measured at c_1 and c_2 as h_1 and h_2 respectively. The output is the predicted mass flow rate. After training the network using Bayesian Regularization algorithm, an ANN prediction model is produced. Using the model, the flow rate is estimated. To avoid over fitting, the training is done using data set from experiment 4, “E4S2”. The prediction is based on the data measured using experiment 2, “E2S1”. The result is shown in Figure 10.3. The mass flow rate predicted by ANN with two hidden layers is better than Kalman filters or observers. However, this model maybe overfitting and more dynamic data needs to be recorded to rely on the prediction. Dynamic data means that the data for different fluid types with different viscosity and density as well as level measurements at different flow rates.

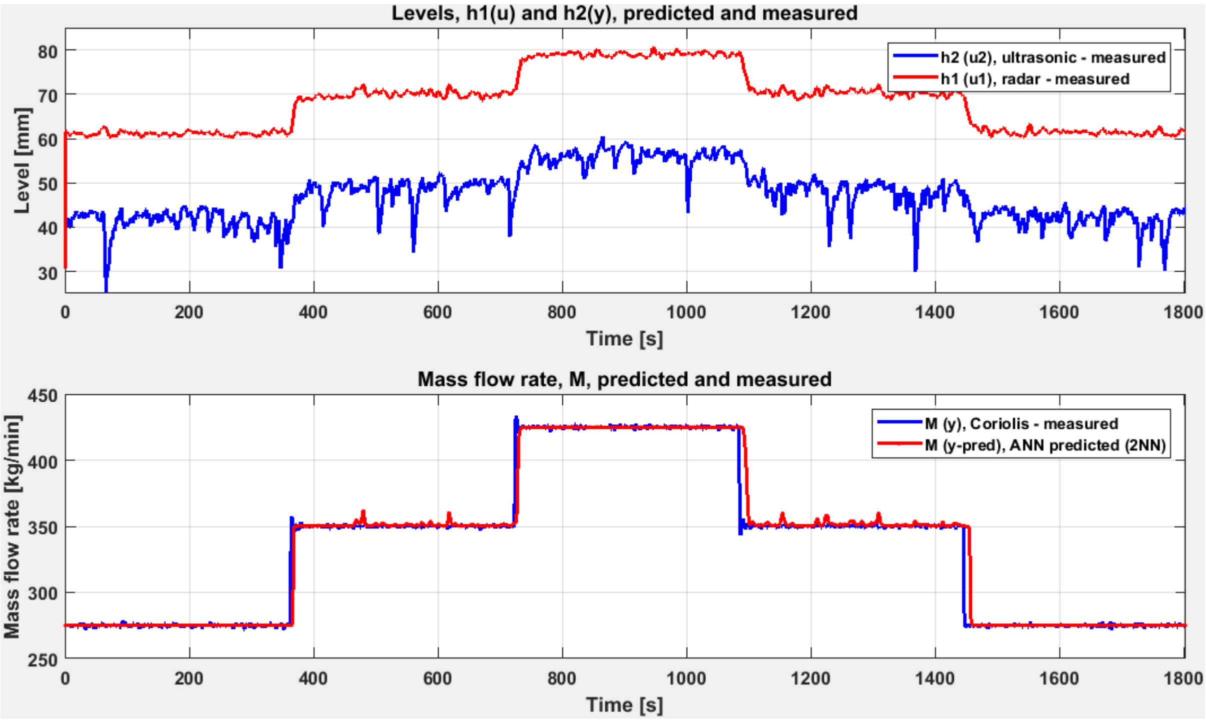

Figure 10.3: Mass flow rate predicted by ANN with two hidden layers. LT19 as input neuron 1, LT18 as input neuron 2.

10.5 Future work recommendations

Analysis of ODEs for two collocation points shows that the model describes the venturi system dynamics in an acceptable way. However, the model improvements are necessary. Specially, for predicting the level at collocation point 2. This can be done by using ODEs for 3-point collocation method currently in the work at USN by a fellow PHD student. Some of the recommended works are listed below:

- Implement the reduced order observer for estimating the mass flow rates for venturi channel since the level are already measured
- Design and implement adaptive observer
    - Since the model parameters are uncertain, they can be adapted so that the estimation error is minimized
    - There are some parameters in the venturi model that can be adjusted for different flow condition and fluid types as well as the type of the channel. For instance, β and k_s can be adjusted for different flow rates. Thus, observer that adapts these parameters is a good way to move forward
- Design and implement nonlinear observers
- Design and implement PDE observers
    - The best possible model for the venturi channel is the PDE model. An observer based on PDE model could give better estimation
- Prediction error methods (PEMs) can help in system identification (identification of state space model)
- Incorporate ANN with deep-learning
- Incorporate mechanism to handle the computational time delay and filter time delay by augmenting them with output delay model
- If case of failed measurements, the apriori state estimates can still be used for approximate knowledge of the system. Reading storing apriori state estimates, is thus, a good idea. This is helpful specially during drill-bit changing process

10.6 Codes and programs

All of the programs and codes used throughout this thesis work are given in Appendix 11. The primary programming language is LabVIEW and MATLAB with Simulink. GUIs are developed for user interactions.

11 Literature review on design of observer based on PDE

Some versions of the linear and nonlinear estimators have been investigated during this thesis work. It is well known by now that the nonlinear model can be improved, and the estimation can also be improved. The best form of estimator out of the investigated methods is the nonlinear estimator, specially UKF. However, UKF was unable to filter out the level estimates due to deviated dynamic model. Deviated in a sense that the level predicted by the mathematical model is way off when simulated in open loop with the real system. To make the deviation minimum, ODEs derived using higher order Lagrange interpolating polynomials such as 3-pt ODEs are required. Even better would be to use PDEs itself given by SVEs of (4.1) and (4.3). Therefore, observer based on PDEs should be an optimal way of flow estimation.

There have been some works on designing PDE observers for quasi-linear hyperbolic PDEs. These observers are suitable for venturi system since the PDEs for describing the top-flow for venturi channel is of type quasi-linear hyperbolic PDEs as discussed in section 4.1. Here is the list of previous work that can be beneficial for future work on observer design for PDEs:

1. PDE observer design for counter – current heat flows in a heat-exchanger. [19]
    a. This work focuses on estimating the temperature profile along the pipes of a plate heat-exchanger. The PDEs used to describe this system is first order and hyperbolic in nature.
2. Adaptive observer design for parabolic PDEs. [20]
    a. This work discusses the observer design for 1D parabolic PDEs based on backstepping methodology to handle the model parameter uncertainties. This idea can be useful since some of the model parameters such as β and k_s can be adapted.
    b. The approach used in this work suggests recasting PDEs as convex optimization problem
3. Backstepping observers for a class of parabolic PDEs. [21]
    a. This paper focuses on designing exponentially convergent observers for parabolic partial integro-differential equations (PIDEs). It has some connection with hyperbolic PDE which is the case in venturi models.
    b. Useful with process with boundary conditions
4. Boundary observers for linear and quasi-linear hyperbolic systems with application to flow control. [22]
    a. Considers the boundary observer design for 1D strictly hyperbolic systems
    b. A technique based on Lyapunov functions
    c. Considers both static and dynamic boundary controls for the observer design
5. An Efficient Implementation of Backstepping Observers for Time-Varying Parabolic PDEs. [23]
    a. A computationally efficient implementation of backstepping-based state observer for 1D PDEs
    b. It produces a set of first order ODEs which are solved using Euler backward differentiation method for computational efficiency
    c. Evaluation of impact of this method is described using numerical simulations

6. Backstepping-based extended Luenberger observer design for a Burgers-type PDE for multi-agent deployment. [24]
   a. A form of an extended Luenberger observer for a burgers-type PDE
   b. It considers the boundary condition for solving PDEs which is suitable for venturi systems as well
   c. It uses backstepping technique to stabilize the parameter observer error dynamics
   d. The concept is based on feedback control strategy

12 Conclusion

Final concluding remarks are summarized in this chapter. This thesis work covers vast array of information from system stability to observer design. The idea of this work is to design suitable observer that can aid in estimating mass flow rate so that the expensive Coriolis flow meters can be removed from the system. To investigate different estimators that can be used in oil industry, a background of the problem is necessary. All of this information is divided into chapters sequentially above. Each chapter acts as a link between the two adjacent chapters. Thus, chapter-wise conclusions are presented below to conclude the results and findings of this report.

- Oil well drilling operation
    - In oil well drilling operation, non-Newtonian fluid is used to lift out the rock cuttings as well as maintain the well pressure in order to prevent the well-blow out. The other important purpose of drill mud is to cool the drill bit.
    - Knowledge of the returning flow is vital for safe operation as it helps to detect kick-loss phenomenon. Information about kick-loss event can aid in determining the mud type to be used so that the bottom hole pressure is maintained, and potential hazardous situation is avoided
- Flow measuring technique
    - Current flow measuring technique used in oil industry such as angle-based paddle meter or Coriolis flow meter are either unreliable or expensive
    - A model-based estimation technique is cheaper and reliable since they do not have mechanical parts that need repair or maintenance
        - A dynamic model of the venturi flume is a good candidate for flow estimation
    - Proposed flow measuring technique is based on open channel venturi flume which is trapezoidal in shape
        - The returning mud from the well is sent through the venturi flume where the levels are measured. Using the measured levels, flow can be estimated
        - Levels are measured at specific points called collocation points. These points are chosen to satisfy a condition (1.1). One point must be at a sub-critical region of the flow and the other point must be around critical region such that the condition (1.1) is satisfied.
    - Venturi rig is built at the lab at USN. The data logged from the rig are represented as the real system for this thesis work
        - LT18 is ultrasonic level transmitter that measures the level at a point and acts as output to the system
        - LT19 is radar level sensor that measures the level at the other point and acts as input to the system
        - FT14A is the Coriolis flow meter that measures the mass flow rate which is only used as a reference during analysis
- Modeling of flow through an open venturi channel
    - A dynamic model for 1D unsteady flow through nonprismatic open venturi channel is developed

  - The model is described SVEs. SVEs for one spatial dimension are a class of quasi-linear hyperbolic of PDEs.
  - Using orthogonal collocation method, PDEs are reduced to a set of ODEs that are used to design suitable nonlinear and semi-nonlinear estimators such as semi-nonlinear Luenberger observer, EKF, UKF
  - The ODEs are linearized using Tylor series expansion. The linearized ODEs are used for designing suitable linear estimators such as linear Luenberger observer and linear Kalman filter
- Model analysis and simulation
  - Linear ODEs are analyzed for stability, controllability and observability
  - Stability analysis is performed based on the analysis of the system poles (eigenvalues). The ODEs for venturi channel are found out to be stable
  - Linear model is both controllable and observable since the rank of controllability and observability matrix is full rank
  - Venturi model consists of multiple model parameters. Their sensitivity is analyzed using differential sensitivity technique, sensitivity index, and correlation
    - Parameter W_1 is the most influential and ρ is the least influential for h_2
    - Parameter W_2 is the most influential and ϕ is the least influential parameter for both M_1 and M_2
  - Linear model works better around the operating point. As the operation moves away from the operating point, the linear model prediction becomes less accurate. Thus, it is important to choose a proper linearization (operation) point during linearization
- Highway to state estimation
  - Not all states are measurable. However, all states are important to describe a dynamic system. Thus, it is important to estimate the states
  - Estimated flow rate can be used to detect early warning of kick-loss event in oil well drilling operation
  - Out of many state estimation technique, this thesis focuses on designing Luenberger observer and Kalman filter
- State Observer
  - A state observer is a system that estimates the states of a system. A full state observer estimates all the states, both measurable and unmeasurable states
  - The purpose of state observer is to estimate states and minimize the error between the true states and the estimated states. This can be done by defining estimation error dynamic and forcing it to go to zero as $t \to \infty$
  - Luenberger linear state observer is designed using pole placement technique
    - Luenberger observer are based on deterministic system
    - Observer poles are placed at any arbitrary location in the left half of the complex plane such that the error dynamic is asymptotically stable
    - Observer gain is determined by placing the poles of the observer at the left half of the complex plane.
      - Further to the left the poles are placed, the faster the observer responds. However, this amplifies the noise in the system.

        - For stability of error dynamic, the poles (eigenvalues) of the error dynamic system matrix is determined. Since, the real part of the eigenvalues is negative, error dynamic is stable
        - Observer must be observable as well.
            - Controllability matrix of an observer state space model is the same as the observability matrix of a linear state space model
            - Since, the linear model is observable, the observer is controllable as well i.e., rank of observer controllability matrix is full rank.
    - A semi-nonlinear observer uses the observer gain determined based on the linear model matrices but uses nonlinear model to predict the apriori state estimates.
    - Both linear observer and semi-nonlinear observer produced satisfactory estimates based on simulator. However, semi-nonlinear observer worked better when operating away from the linearization point.
    - Linear and semi-nonlinear observer are able to reduce the noise in the system. However, if the poles are placed far to the left, the noise is amplified.
    - Error dynamic in terms of linear observer showed deviation when operated away from the linearization point, as expected. The error dynamic in terms of semi-nonlinear observer is asymptotically stable at any point of operation.
        - Thus, semi-nonlinear observer is a better choice than linear observer when working with the mathematical model
- Kalman filter
    - Kalman filter are based on stochastic system meaning that it incorporates the information of the process and measurement noise in the system
    - Linear Kalman filter resembles the linear Luenberger observer. Both estimators are designed based on linear ODEs
    - Kalman gain is determined either offline (steady-state) or online (time-varying)
        - In case of time-varying Kalman gain, a C-DARE must be solved. In this thesis work, RK4 is used to solve C-DARE to compute Kalman gain
        - It is found out that time-varying Kalman gain very quickly converges to a steady state value. Thus, it can be calculated offline to simplify the linear Kalman filter algorithm
    - Nonlinear Kalman filters such as EKF and UKF are designed using nonlinear-ODEs
        - EKF linearizes the ODE around apriori state estimates. Therefore, the model matrices are time-varying in contrast to the time-invariant model matrices for LKF.
        - UKF uses unscented transformation technique. It generates a set of sample points that capture the means and covariance of the state estimates and transforms them using the nonlinear ODEs. The mean of the transformed points gives a better approximation of the true states.
    - UKF is faster than EKF and LKF in case of model. All Kalman filters are able to reduce the noise in the system. There is minimal deviation between the true state estimates and the real states when UKF or EKF is used. The deviation is evident when LKF is used. Thus, EKF is better than LKF in a sense that EKF reduces deviation. UKF is better than EKF in a sense that it converges faster, and the estimates are more robust
    - EKF and semi-nonlinear observer perform exactly for the simulator

    - LKF and linear observer performed similarly but noise is amplified by faster observer poles. LKF is able to reduce the noise more than observer because noise covariance matrices for LKF are tuned properly
        - Linear observer is easier to implement as it does not rely on tuning noise covariance matrices which can be cumbersome
- Experimentation and results
    - Data is logged 1 sample per second from LT18, LT19 and FT14A
    - Since measurements are noisy, three types of filters, MEDF, WMAF and LPF are compared
        - LPF proved to perform better
    - Comparison of nonlinear model and real system shows that the level predicted by model are way off. This is potentially due to improper model parameters
    - LKF and linear observer performed relatively well around the operating point
    - UKF converges faster than EKF even with the delay introduced by the use of LPF. Thus, UKF is better than EKF. Although, EKF is easier to implement and computationally less heavy than UKF
    - The price to pay when choosing UKF is that it is computationally heavy. It has more algorithms to execute than EKF or LKF. Even though the computers are extremely powerful today, for a system with multiple states, the execution time has a huge impact on the system performance. Although, for a venturi system described by two-point collocation method, UKF is preferable
- EKF is suitable for mildly nonlinear systems. It may fail for highly nonlinear systems. UKF does not rely on linearization. Thus, it is an optimal choice unless computation time is of concern.

A final statement based on the result and observation is that the flow estimates by 2-pt models are promising. A better version of the dynamic model such as 3-pt models can improve the estimates. Therefore, it is safe to say that for a proper dynamic model, the proposed flow estimation technique can replace expensive Coriolis flow meter. The potential of UKF is huge. For better estimation, observer based on PDEs are recommended.

References

[1] K. Chhantyal, H. Viumdal and S. Mylvaganam, "Online Drilling Fluid Flowmetering in Open Channels with Ultrasonic Level Sensors using Critical Depths," in *Linköping Electronic Conference Proceedings*, Reykjavik, 2017.

[2] JAMSTEC, "Scientific Deep Sea Drilling and Coring Technology," JAMSTEC, [Online]. Available: http://www.jamstec.go.jp/chikyu/e/gallery/movie/. [Accessed March 2018].

[3] "Offshore deepwater drilling process," [Online]. Available: https://www.youtube.com/watch?v=c0bHP3yYVuk. [Accessed February 2018].

[4] B. R. Munson, D. F. Young, T. H. Okiishi and W. W. Huebsch, Fundamentals of Fluid Mechanics, 6th ed., John Wiley & Sons, Inc., 2009, p. 778.

[5] J. Vlachopoulos and D. Strutt, "The Role of Rheology in Polymer Extrusion," in *Extrusion Minitec and Conference*, Hamilton, Ontario, 2003.

[6] J. F. Douglas, J. M. Gasiorek, J. A. Swaffield and L. B. Jack, Fluid Mechanics, 5th ed., Essex: Pearson Education Limited, 2005, p. 993.

[7] S. J. Farlow, Partial Differential Equations for Scientists and Engineers, Reprint ed., New York: Dover Publications, Inc, 1993, p. 429.

[8] I. Pirir, A. Jinasena and R. Sharma, "Model based flow measurement using venturi flumes for return flow during drilling," *Modeling, Identification and Control,* vol. 38, no. 3, pp. 135--142, 2017.

[9] E. Kreyszig, Advanced Engineering Mathematics, 10th ed., John Wiley & Sons Inc., 2011, p. 1278.

[10] A. Jinasena, G.-O. Kaasa and R. Sharma, "Use of Orthogonal Collocation Method for a Dynamic Model of the Flow in a Prismatic Open Channel: For Estimation Purposes," in *Linköping Electronic Conference Proceedings*, Reykjavik, 2017.

[11] R. C. Dorf and R. H. Bishop, Modern Control Systsm, 12th ed., New Jersey: Prentice Hall, 2011, p. 1111.

[12] D. Hamby, "A review of techniques for parameter sensitivity analysis of environmental models," *Environmental Monitoring and Assessment,* vol. 32, no. 135, 1994.

[13] B. Sudret and C. Mai, "Computing derivative-based global sensitivity measures using polynomial chaos expansions," *Reliability Engineering and System Safety,* vol. 134, pp. 241-250, 2015.

[14] R. Sharma and B. Glemmestad, "Optimal control strategies with nonlinear optimization for an Electric Submersible Pump lifted oil field," *International Journal of Modeling and Optimization,* vol. 4, no. 4, pp. 278-286, 2014.

[15] Z. Gajic, "http://www.ece.rutgers.edu," [Online]. Available: http://www.ece.rutgers.edu/~gajic/psfiles/observers.pdf. [Accessed January 2018].

[16] D. Simon, Optimal State Estimation, New Jersey: John Wiley & Sons, Inc., 2006, p. 550.

[17] D. D. Ruscio, "Telememark University College (HIT) - Now USN," [Online]. Available: http://www2.hit.no/tf/fag/scev3006/kompendium/main_pc_e.pdf. [Accessed March 2018].

[18] "Understanding Kalman Filters Part 5: Nonlinear State Estimators," MATLAB, [Online]. Available: https://www.youtube.com/watch?v=Vefia3JMeHE. [Accessed March 2018].

[19] F. Zobiri, E. Witrant and F. Bonne, "PDE Observer Design for Counter-Current Heat Flows in a Heat-Exchanger," *IFAC-PapersOnLine,* vol. 50, no. 1, pp. 7127-7132, July 2017.

[20] P. A. Ascencio, «Adaptive Observer Design for Parabolic Partial Differential Equations,» London, 2017.

[21] A. Smyshlyaev and M. Krstic, "Backstepping observers for a class of parabolic PDE," *Systems & Control Letters,* vol. 54, pp. 613-625, 2005.

[22] F. C. Buenaventura, E. Witrant, C. Prieur and L. Dugard, "Boundary Observers for Linear and Quasi-linear Hyperbolic Systems with Application to Flow Control," *Automatica,* vol. 49, pp. 3180-3188, 2013.

[23] L. Jadachowski, T. Meurer and A. Kugi, "An Efficient Implementation of Backstepping Observers for Time-Varying Parabolic PDEs," *IFAC Proceedings Volumes,* vol. 45, no. 2, pp. 798-803, January 2012.

[24] G. Freudenthaler, F. Göttsch og T. Meurer, «Backstepping-based extended Luenberger observer design for a Burgers-type PDE for multi-agent deployment,» *IFAC-PapersOnLine,* vol. 50, nr. 1, pp. 6780-6785, July 2017.

Appendices

Appendix 1 ODEs in terms of changed variables

Part 1A *ODEs in terms of h and M*

Changing wetted cross-sectional area (A) and volumetric flow rate (Q)as states with level (h) and mass flow rate (M).

$$\begin{matrix} A & \rightarrow & h \\ Q & \rightarrow & M \end{matrix} \Rightarrow \begin{matrix} \dot{A} & \rightarrow & \dot{h} \\ \dot{Q} & \rightarrow & \dot{M} \end{matrix}$$

Procedure to change the states:

- From equation (4.8), area is a function of level

$$A = S_s h^2 + Wh$$

- Take the time derivative on both sides

$$\frac{dA}{dt} = S_s \frac{dh^2}{dt} + W\frac{dh}{dt}$$

$$\frac{dA}{dt} = S_s \frac{dh^2}{dh}\frac{dh}{dt} + W\frac{dh}{dt}$$

$$\frac{dA}{dt} = 2S_s h\frac{dh}{dt} + W\frac{dh}{dt}$$

$$\frac{dA}{dt} = (2S_s h + W)\frac{dh}{dt}$$

$$\frac{dA}{dt} = D\frac{dh}{dt} \Rightarrow \dot{A} = D\dot{h}$$

  - From equation (4.9), scaling variable D is defined as:

$$D = 2S_s h + W$$

- Express the level as state

$$\frac{dh}{dt} = \frac{1}{D}\frac{dA}{dt} \Rightarrow \dot{h} = \frac{\dot{A}}{D}$$

- Expressing volumetric flow rate as mass flow rate
  - Volume and mass relation

$$V = \frac{m}{\rho}$$

  - Take the time derivative on both sides

$$\frac{dV}{dt} = \frac{1}{\rho}\frac{dm}{dt}$$

$$Set, \quad \frac{dV}{dt} = Q, \quad \frac{dm}{dt} = M$$

$$Q = \frac{1}{\rho}M \tag{a1}$$

- Take the time derivative on both sides again

$$\frac{dQ}{dt} = \frac{1}{\rho}\frac{dM}{dt} \Longrightarrow \dot{Q} = \frac{1}{\rho}\dot{M}$$

- Express mass flow rate as the new state

$$\frac{dM}{dt} = \rho\frac{dQ}{dt} \Longrightarrow \dot{M} = \rho\dot{Q}$$

Table 12.1: States defined at different collocation points (1, 2) using the new state variables

$\dot{A}_1 = D_1\dot{h}_1$	$\dot{A}_2 = D_2\dot{h}_2$	$\dot{Q}_1 = \frac{1}{\rho}\dot{M}_1$	$\dot{Q}_2 = \frac{1}{\rho}\dot{M}_2$
$\dot{h}_1 = \frac{\dot{A}_1}{D_1}$	$\dot{h}_2 = \frac{\dot{A}_2}{D_2}$	$\dot{M}_1 = \rho\dot{Q}_1$	$\dot{M}_2 = \rho\dot{Q}_2$

- Express the ODEs given by (4.15) to (4.18) with new states by substituting equations from Table 12.1 the new ODEs are derived below.
  - Changing $\dot{A}_1$ to $\dot{h}_1$

$$\dot{A}_1 = -\frac{1}{L}(-Q_1 + Q_2)$$

$$\dot{h}_1 D_1 = -\frac{1}{L}\left(-\frac{M_1}{\rho} + \frac{M_2}{\rho}\right)$$

$$\dot{h}_1 = \frac{(M_1 - M_2)}{\rho L D_1}$$

  - Changing $\dot{A}_2$ to $\dot{h}_2$

$$\dot{A}_2 = -\frac{1}{L}(-Q_1 + Q_2)$$

$$\dot{h}_2 D_2 = -\frac{1}{L}\left(-\frac{M_1}{\rho} + \frac{M_2}{\rho}\right)$$

$$\dot{h}_2 = \frac{(M_1 - M_2)}{\rho L D_2}$$

  - Changing $\dot{Q}_1$ to $\dot{M}_1$

$$\dot{Q}_1 = -\frac{\beta}{L}\left(-\frac{Q_1^2}{A_1} + \frac{Q_2^2}{A_2}\right) - \frac{g}{L}\left(-I_{1_1} + I_{1_2}\right) + \frac{gh_1^2}{2L}(-W_1 + W_2) + gA_1\left(S_b - S_{f_1}\right)$$

$$\dot{Q}_1 = \frac{\beta}{L}\left(\frac{Q_1^2}{A_1} - \frac{Q_2^2}{A_2}\right) + \frac{g}{L}\left(I_{1_1} - I_{1_2}\right) - \frac{gh_1^2}{2L}(W_1 - W_2) + gS_bA_1 - gA_1S_{f_1}$$

$$\dot{Q}_1 = \frac{\beta}{L}\left(\frac{Q_1^2}{A_1} - \frac{Q_2^2}{A_2}\right) + \frac{g}{L}\left(I_{1_1} - I_{1_2}\right) - \frac{gh_1^2}{2L}(W_1 - W_2) + gS_bA_1 - gA_1\frac{Q_1|Q_1|n_m^2P_1^{\frac{4}{3}}}{A_1^{\frac{10}{3}}}$$

$$\frac{\dot{M}_1}{\rho} = \frac{\beta}{L}\left(\frac{1}{A_1}\left(\frac{M_1}{\rho}\right)^2 - \frac{1}{A_2}\left(\frac{M_2}{\rho}\right)^2\right) + \frac{g}{L}\left(I_{1_1} - I_{1_2}\right) - \frac{gh_1^2}{2L}(W_1 - W_2) + gS_bA_1 - g\frac{\frac{M_1}{\rho}\left|\frac{M_1}{\rho}\right| n_m^2 P_1^{\frac{4}{3}}}{A_1^{\frac{7}{3}}}$$

$$\dot{M}_1 = \frac{\beta}{\rho L}\left(\frac{M_1^2}{A_1} - \frac{M_2^2}{A_2}\right) + \frac{\rho g}{L}\left(I_{1_1} - I_{1_2}\right) - \frac{\rho g h_1^2}{2L}(W_1 - W_2) + \rho g S_b A_1 - \frac{g}{\rho}\frac{M_1|M_1|n_m^2 P_1^{\frac{4}{3}}}{A_1^{\frac{7}{3}}}$$

- Changing $\dot{Q}_2$ to $\dot{M}_2$

$$\dot{Q}_2 = -\frac{\beta}{L}\left(-\frac{Q_1^2}{A_1} + \frac{Q_2^2}{A_2}\right) - \frac{g}{L}\left(-I_{1_1} + I_{1_2}\right) + \frac{gh_2^2}{2L}(-W_1 + W_2) + gA_2\left(S_b - S_{f_2}\right)$$

$$\dot{Q}_2 = \frac{\beta}{L}\left(\frac{Q_1^2}{A_1} - \frac{Q_2^2}{A_2}\right) + \frac{g}{L}\left(I_{1_1} - I_{1_2}\right) - \frac{gh_2^2}{2L}(W_1 - W_2) + gS_bA_2 - gA_2S_{f_2}$$

$$\dot{Q}_2 = \frac{\beta}{L}\left(\frac{Q_1^2}{A_1} - \frac{Q_2^2}{A_2}\right) + \frac{g}{L}\left(I_{1_1} - I_{1_2}\right) - \frac{gh_2^2}{2L}(W_1 - W_2) + gS_bA_2 - gA_2\frac{Q_2|Q_2|n_m^2 P_2^{\frac{4}{3}}}{A_2^{\frac{10}{3}}}$$

$$\frac{\dot{M}_2}{\rho} = \frac{\beta}{L}\left(\frac{1}{A_1}\left(\frac{M_1}{\rho}\right)^2 - \frac{1}{A_2}\left(\frac{M_2}{\rho}\right)^2\right) + \frac{g}{L}\left(I_{1_1} - I_{1_2}\right) - \frac{gh_2^2}{2L}(W_1 - W_2) + gS_bA_2 - g\frac{\frac{M_2}{\rho}\left|\frac{M_2}{\rho}\right| n_m^2 P_2^{\frac{4}{3}}}{A_2^{\frac{7}{3}}}$$

$$\dot{M}_2 = \frac{\beta}{\rho L}\left(\frac{M_1^2}{A_1} - \frac{M_2^2}{A_2}\right) + \frac{\rho g}{L}\left(I_{1_1} - I_{1_2}\right) - \frac{\rho g h_2^2}{2L}(W_1 - W_2) + \rho g S_b A_2 - \frac{g}{\rho}\frac{M_2|M_2|n_m^2 P_2^{\frac{4}{3}}}{A_2^{\frac{7}{3}}}$$

Part 1B *ODEs in terms of h and Q*

$$\dot{A}_1 = -\frac{1}{L}(-Q_1 + Q_2)$$

$$\dot{h}_1(2S_sh_1 + W_1) = -\frac{1}{L}(-Q_1 + Q_2)$$

$$\dot{h}_1 = -\frac{(-Q_1 + Q_2)}{L(2S_sh_1 + W_1)} = f_1$$

$$\dot{A}_2 = -\frac{1}{L}(-Q_1 + Q_2)$$

$$\dot{h}_2 = -\frac{(-Q_1 + Q_2)}{L(2S_s h_2 + W_2)} = f_2$$

$$\dot{Q}_1 = -\frac{\beta}{L}\left(-\frac{Q_1^2}{A_1} + \frac{Q_2^2}{A_2}\right) - \frac{g}{L}\left(-I_{1_1} + I_{1_2}\right) + \frac{g h_1^2}{2L}(-W_1 + W_2) + gA_1\left(S_b - S_{f_1}\right) = f_3$$

$$\begin{aligned}\dot{Q}_1 = {} & -\frac{\beta}{L}\left(-\frac{Q_1^2}{S_s h_1^2 + W_1 h_1} + \frac{Q_2^2}{S_s h_2^2 + W_2 h_2}\right) - \frac{g}{L}\left(-h_1^2 \frac{W_1}{2} - h_1^3 \frac{S_s}{3} + h_2^2 \frac{W_2}{2} + h_2^3 \frac{S_s}{3}\right) \\ & + \frac{g h_1^2}{2L}(-W_1 + W_2) + gS_b(S_s h_1^2 + W_1 h_1) \\ & - g\frac{Q_1|Q_1|n_M^2\left(W_1 + 2h_1\sqrt{1 + S_s^2}\right)^{\frac{4}{3}}}{(S_s h_1^2 + W_1 h_1)^{\frac{7}{3}}} = f_3\end{aligned}$$

$$\dot{Q}_2 = -\frac{\beta}{L}\left(-\frac{Q_1^2}{A_1} + \frac{Q_2^2}{A_2}\right) - \frac{g}{L}\left(-I_{1_1} + I_{1_2}\right) + \frac{g h_2^2}{2L}(-W_1 + W_2) + gA_2\left(S_b - S_{f_2}\right) = f_4$$

$$\begin{aligned}\dot{Q}_2 = {} & -\frac{\beta}{L}\left(-\frac{Q_1^2}{S_s h_1^2 + W_1 h_1} + \frac{Q_2^2}{S_s h_2^2 + W_2 h_2}\right) - \frac{g}{L}\left(-h_1^2 \frac{W_1}{2} - h_1^3 \frac{S_s}{3} + h_2^2 \frac{W_2}{2} + h_2^3 \frac{S_s}{3}\right) \\ & + \frac{g h_2^2}{2L}(-W_1 + W_2) + gS_b(S_s h_2^2 + W_2 h_2) \\ & - g\frac{Q_2|Q_2|n_M^2\left(W_2 + 2h_2\sqrt{1 + S_s^2}\right)^{\frac{4}{3}}}{(S_s h_2^2 + W_2 h_2)^{\frac{7}{3}}} = f_4\end{aligned}$$

Appendix 2 Tylor series expansion and Jacobian matrices

The nonlinear state and the measurement equations for a deterministic system are given by:

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

$$y = g(x,u)$$

- u could be a vector with n_u elements or a scalar with one element
- x could be a vector with n_x elements or a scalar with one element
- y could be a vector with n_y elements or a scalar with one element
- f could be a vector function with n_x member functions or a scalar with one function
- g could be a vector function with n_y member functions or a scalar with one function
- f and g depends on the number of states and the number of outputs in the system

Function values around the linearization (operating) point x_{op}, u_{op}, y_{op}:

$$\dot{x}_{op} = f\left(x_{op}, u_{op}\right)$$

$$y_{op} = g\left(x_{op}, u_{op}\right)$$

Linearizing the state and measurement equations (RHS) around the operating point x_{op}, u_{op}, y_{op} using Tylor series approximation:

$$f(x,u) = f\left(x_{op},u_{op}\right) + \left.\frac{\partial f\left(x_{op},u_{op}\right)}{\partial x}\right|_{x_{op},u_{op}} \left(x - x_{op}\right) + \left.\frac{\partial f\left(x_{op},u_{op}\right)}{\partial u}\right|_{x_{op},u_{op}} \left(u - u_{op}\right) + HOT$$

$$g(x,u) = g\left(x_{op},u_{op}\right) + \left.\frac{\partial g\left(x_{op},u_{op}\right)}{\partial x}\right|_{x_{op},u_{op}} \left(x - x_{op}\right) + \left.\frac{\partial g\left(x_{op},u_{op}\right)}{\partial u}\right|_{x_{op},u_{op}} \left(u - u_{op}\right) + HOT$$

$$f(x,u) - f\left(x_{op},u_{op}\right) = \left.\frac{\partial f\left(x_{op},u_{op}\right)}{\partial x}\right|_{x_{op},u_{op}} \left(x - x_{op}\right) + \left.\frac{\partial f\left(x_{op},u_{op}\right)}{\partial u}\right|_{x_{op},u_{op}} \left(u - u_{op}\right) + \cancel{HOT}$$

$$g(x,u) - g\left(x_{op},u_{op}\right) = \left.\frac{\partial g\left(x_{op},u_{op}\right)}{\partial x}\right|_{x_{op},u_{op}} \left(x - x_{op}\right) + \left.\frac{\partial g\left(x_{op},u_{op}\right)}{\partial u}\right|_{x_{op},u_{op}} \left(u - u_{op}\right) + \cancel{HOT}$$

Substitute the following and remove the higher order terms (HOT):

$$f(x,u) = \dot{x}, \qquad f\left(x_{op},u_{op}\right) = \dot{x}_{op}$$

$$g(x,u) = y, \qquad g\left(x_{op},u_{op}\right) = y_{op}$$

The new refined equation becomes:

$$\dot{x} - \dot{x}_{op} = \left.\frac{\partial f}{\partial x}\right|_{x_{op},u_{op}} \left(x - x_{op}\right) + \left.\frac{\partial f}{\partial u}\right|_{x_{op},u_{op}} \left(u - u_{op}\right)$$

$$y - y_{op} = \left.\frac{\partial g}{\partial x}\right|_{x_{op},u_{op}} \left(x - x_{op}\right) + \left.\frac{\partial g}{\partial u}\right|_{x_{op},u_{op}} \left(u - u_{op}\right)$$

Define deviation variables:

$$x - x_{op} = \delta x, \qquad \dot{x} - \dot{x}_{op} = \delta\dot{x}, \qquad u - u_{op} = \delta u, \qquad y - y_{op} = \delta y \tag{a2}$$

Where,

$$\left.\frac{\partial f}{\partial x}\right|_{x_{op},u_{op}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_{n_x}} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_{n_x}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{n_x}}{\partial x_1} & \frac{\partial f_{n_x}}{\partial x_2} & \cdots & \frac{\partial f_{n_x}}{\partial x_{n_x}} \end{bmatrix}, \quad \left.\frac{\partial f}{\partial u}\right|_{x_{op},u_{op}} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_{n_u}} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_{n_u}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{n_x}}{\partial u_1} & \frac{\partial f_{n_x}}{\partial u_2} & \cdots & \frac{\partial f_{n_x}}{\partial u_{n_u}} \end{bmatrix}$$

$$\left.\frac{\partial g}{\partial x}\right|_{x_{op},u_{op}} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_{n_x}} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_{n_x}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_{n_y}}{\partial x_1} & \frac{\partial g_{n_y}}{\partial x_2} & \cdots & \frac{\partial g_{n_y}}{\partial x_{n_x}} \end{bmatrix}, \quad \left.\frac{\partial g}{\partial u}\right|_{x_{op},u_{op}} = \begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \frac{\partial g_1}{\partial u_2} & \cdots & \frac{\partial g_1}{\partial u_{n_u}} \\ \frac{\partial g_2}{\partial u_1} & \frac{\partial g_2}{\partial u_2} & \cdots & \frac{\partial g_2}{\partial u_{n_u}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_{n_y}}{\partial u_1} & \frac{\partial g_{n_y}}{\partial u_2} & \cdots & \frac{\partial g_{n_y}}{\partial u_{n_u}} \end{bmatrix}$$

Define Jacobian matrices (system matrices):

$$A_c = \left.\frac{\partial f}{\partial x}\right|_{x_{op},u_{op}} \quad B_c = \left.\frac{\partial f}{\partial u}\right|_{x_{op},u_{op}} \quad C_c = \left.\frac{\partial g}{\partial x}\right|_{x_{op},u_{op}} \quad D_c = \left.\frac{\partial g}{\partial u}\right|_{x_{op},u_{op}} \tag{a3}$$

- The dimension of these matrices is:

$$[A_c] = [n_x \times n_x] \quad [B_c] = [n_x \times n_u] \quad [C_c] = [n_y \times n_x] \quad [D_c] = [n_y \times n_u]$$

State and measurement equations in deviation form for deterministic system

$$\begin{aligned} \delta\dot{x} &= A_c\delta x + B_c\delta u \\ \delta y &= C_c\delta x + D_c\delta u \end{aligned} \tag{a4}$$

Full form of state and measurement equations for deterministic system

$$\begin{aligned} \dot{x} &= \dot{x}_{op} + A_c(x - x_{op}) + B_c(u - u_{op}) \\ y &= y_{op} + C_c(x - x_{op}) + D_c(u - u_{op}) \end{aligned} \tag{a5}$$

Appendix 3 Linear model matrices, alternative form

Part 3A *System matrices in terms of A and Q*

$$A_c = \begin{bmatrix} \frac{\partial f_2}{\partial A_2} & \frac{\partial f_2}{\partial Q_1} & \frac{\partial f_2}{\partial Q_2} \\ \frac{\partial f_3}{\partial A_2} & \frac{\partial f_3}{\partial Q_1} & \frac{\partial f_3}{\partial Q_2} \\ \frac{\partial f_4}{\partial A_2} & \frac{\partial f_4}{\partial Q_1} & \frac{\partial f_4}{\partial Q_2} \end{bmatrix}, \qquad B_c = \begin{bmatrix} \frac{\partial f_2}{\partial A_1} \\ \frac{\partial f_3}{\partial A_1} \\ \frac{\partial f_4}{\partial A_1} \end{bmatrix}, \qquad C_c = \begin{bmatrix} \frac{\partial g_1}{\partial A_2} & \frac{\partial g_1}{\partial Q_1} & \frac{\partial g_1}{\partial Q_2} \end{bmatrix}$$

Where;

$$f_2 = equation\ (4.16) \quad f_3 = equation\ (4.17) \quad f_4 = equation\ (4.18) \quad g = A_2$$

$$\left.\frac{dh_1}{dA_1}\right|_{op} = \frac{1}{\sqrt{W_1^2 + 4A_{1_{op}}S_s}}$$

$$\left.\frac{dh_2}{dA_2}\right|_{op} = \frac{1}{\sqrt{W_2^2 + 4A_{2_{op}}S_s}}$$

$$\left.\frac{dP_1}{dA_1}\right|_{op} = \frac{2\sqrt{1+S_s^2}}{\sqrt{W_1^2 + 4A_{1_{op}}S_s}} = 2\sqrt{1+S_s^2}\left.\frac{dh_1}{dA_1}\right|_{op}$$

$$\left.\frac{dP_2}{dA_2}\right|_{op} = \frac{2\sqrt{1+S_s^2}}{\sqrt{W_2^2 + 4A_{2_{op}}S_s}} = 2\sqrt{1+S_s^2}\left.\frac{dh_2}{dA_2}\right|_{op}$$

$$A_{c_{11}} = \frac{\partial f_2}{\partial A_2} = 0$$

$$A_{c_{21}} = \frac{\partial f_3}{\partial A_2} = \frac{\beta}{L}\frac{Q_{2_{op}}^2}{A_{2_{op}}^2} - \frac{gA_{2_{op}}}{L}\left.\frac{dh_2}{dA_2}\right|_{op}$$

$$A_{c_{31}} = \frac{\partial f_4}{\partial A_2} = \frac{\beta}{L}\frac{Q_{2_{op}}^2}{A_{2_{op}}^2} - \frac{g}{L}\left(A_{2_{op}} + h_{2_{op}}W_1 - h_{2_{op}}W_2\right)\left.\frac{dh_2}{dA_2}\right|_{op} + gS_b$$
$$- \left(4\left.\frac{dP_2}{dA_2}\right|_{op} - \frac{7\mathrm{P}_{2_{op}}}{A_{2_{op}}}\right)\frac{gQ_{2_{op}}\left|Q_{2_{op}}\right| n_M^2 \sqrt[3]{\mathrm{P}_{2_{op}}}}{3A_{2_{op}}^{\frac{7}{3}}}$$

$$A_{c_{12}} = \frac{\partial f_2}{\partial Q_1} = \frac{1}{L}$$

$$A_{c_{22}} = \frac{\partial f_3}{\partial Q_1} = \frac{2\beta Q_{1_{op}}}{LA_{1_{op}}} - \frac{2gQ_{1_{op}}^2 n_M^2 \mathrm{P}_{1_{op}}^{\frac{4}{3}}}{\left|Q_{1_{op}}\right| A_{1_{op}}^{\frac{7}{3}}}$$

$$A_{c_{32}} = \frac{\partial f_4}{\partial Q_1} = \frac{2\beta Q_{1_{op}}}{LA_{1_{op}}}$$

$$A_{c_{13}} = \frac{\partial f_2}{\partial Q_2} = \frac{-1}{L}$$

$$A_{c_{23}} = \frac{\partial f_3}{\partial Q_2} = \frac{-2\beta Q_{2_{op}}}{LA_{2_{op}}}$$

$$A_{c_{33}} = \frac{\partial f_4}{\partial Q_2} = \frac{-2\beta Q_{2_{op}}}{LA_{2_{op}}} - \frac{2gQ_{2_{op}}^2 n_M^2 \mathrm{P}_{2_{op}}^{\frac{4}{3}}}{\left|Q_{2_{op}}\right| A_{2_{op}}^{\frac{7}{3}}}$$

$$B_{c_{11}} = \frac{\partial f_2}{\partial A_1} = 0$$

$$B_{c_{21}} = \frac{\partial f_3}{\partial A_1} = \frac{-\beta}{L}\frac{Q_{1_{op}}^2}{A_{1_{op}}^2} + \frac{g\left(A_{1_{op}} - h_{1_{op}}W_1 + h_{1_{op}}W_2\right)}{L}\left.\frac{dh_1}{dA_1}\right|_{op} + gS_b$$
$$- \left(4\left.\frac{dP_1}{dA_1}\right|_{op} - \frac{7\mathrm{P}_{1_{op}}}{A_{1_{op}}}\right)\frac{gQ_{1_{op}}\left|Q_{1_{op}}\right| n_M^2 \sqrt[3]{\mathrm{P}_{1_{op}}}}{3A_{1_{op}}^{\frac{7}{3}}}$$

$$B_{c_{31}} = \frac{\partial f_4}{\partial A_1} = \frac{-\beta}{L}\frac{Q_{1_{op}}^2}{A_{1_{op}}^2} + \frac{gA_{1_{op}}}{L}\left.\frac{dh_1}{dA_1}\right|_{op}$$

$$C_{c_{11}} = \frac{\partial g}{\partial A_2} = 1$$

$$C_{c_{12}} = \frac{\partial g}{\partial Q_1} = 0$$

$$C_{c_{13}} = \frac{\partial g}{\partial Q_2} = 0$$

Part 3B ***System matrices in terms of h and Q***

$$A_c = \begin{bmatrix} \frac{\partial f_2}{\partial h_2} & \frac{\partial f_2}{\partial Q_1} & \frac{\partial f_2}{\partial Q_2} \\ \frac{\partial f_3}{\partial h_2} & \frac{\partial f_3}{\partial Q_1} & \frac{\partial f_3}{\partial Q_2} \\ \frac{\partial f_4}{\partial h_2} & \frac{\partial f_4}{\partial Q_1} & \frac{\partial f_4}{\partial Q_2} \end{bmatrix}, \qquad B_c = \begin{bmatrix} \frac{\partial f_2}{\partial h_1} \\ \frac{\partial f_3}{\partial h_1} \\ \frac{\partial f_4}{\partial h_1} \end{bmatrix}, \qquad C_c = \begin{bmatrix} \frac{\partial g}{\partial h_2} & \frac{\partial g}{\partial Q_1} & \frac{\partial g}{\partial Q_2} \end{bmatrix}$$

Where;

f_2, f_3 and f_4 are given by ODEs in Part 1B. $g = h_2$

$$\frac{\partial f_2}{\partial h_2} = \frac{2S_s\left(-Q_{1_{op}} + Q_{2_{op}}\right)}{L\left(2S_s h_{2_{op}} + W_2\right)^2}, \qquad Q_{1_{op}} = Q_{2_{op}}$$

$$\frac{\partial f_3}{\partial h_2} = \frac{\beta}{L} \frac{Q_{2_{op}}^2 \left(2S_s h_{2_{op}} + W_2\right)}{A_{2_{op}}^2} - \frac{g}{L}\left(W_2 h_{2_{op}} + S_s h_{2_{op}}^2\right)$$

$$\begin{aligned}\frac{\partial f_4}{\partial h_2} = {} & \frac{\beta}{L} \frac{Q_{2_{op}}^2 \left(2S_s h_{2_{op}} + W_2\right)}{A_{2_{op}}^2} - \frac{g}{L}\left(W_2 h_{2_{op}} + S_s h_{2_{op}}^2\right) + \frac{g h_{2_{op}}}{L}(-W_1 + W_2) \\ & + g S_b \left(2S_s h_{2_{op}} + W_2\right) - \frac{8}{3} g Q_{2_{op}} \left|Q_{2_{op}}\right| n_M^2 \frac{\sqrt{1 + S_s^2}(Pw_2)^{\frac{1}{3}}}{A_{2_{op}}^{\frac{7}{3}}} \\ & + \frac{7}{3} g Q_{2_{op}} \left|Q_{2_{op}}\right| n_M^2 \frac{\left(W_2 + 2h_{2_{op}}\sqrt{1 + S_s^2}\right)^{\frac{4}{3}} \left(2S_s h_{2_{op}} + W_2\right)}{\left(S_s h_{2_{op}}^2 + W_2 h_{2_{op}}\right)^{\frac{10}{3}}}\end{aligned}$$

$$\frac{\partial f_2}{\partial Q_1} = \frac{-1}{L\left(2S_s h_{2_{op}} + W_2\right)}$$

$$\frac{\partial f_3}{\partial Q_1} = \frac{2\beta Q_{1_{op}}}{L\left(S_s h_{1_{op}}^2 + W_1 h_{1_{op}}\right)} - 2g \frac{Q_{1_{op}}^2 n_M^2 \left(W_1 + 2h_{1_{op}}\sqrt{1 + S_s^2}\right)^{\frac{4}{3}}}{\left|Q_{1_{op}}\right| \left(S_s h_{1_{op}}^2 + W_1 h_{1_{op}}\right)^{\frac{7}{3}}}$$

$$\frac{\partial f_4}{\partial Q_1} = \frac{2\beta Q_{1_{op}}}{L\left(S_s h_{1_{op}}^2 + W_1 h_{1_{op}}\right)}, \qquad h_{1_{op}} \neq 0$$

$$\frac{\partial f_2}{\partial Q_2} = \frac{1}{L\left(2S_s h_{2_{op}} + W_2\right)}$$

$$\frac{\partial f_3}{\partial Q_2} = \frac{-2\beta Q_{2_{op}}}{L\left(S_s h_{2_{op}}^2 + W_2 h_{2_{op}}\right)}, \qquad h_{2_{op}} \neq 0$$

$$\frac{\partial f_4}{\partial Q_2} = \frac{-2\beta Q_{2_{op}}}{L\left(S_s h_{2_{op}}^2 + W_2 h_{2_{op}}\right)} - 2g \frac{Q_{2_{op}}^2 n_M^2 \left(W_2 + 2h_{2_{op}}\sqrt{1 + S_s^2}\right)^{\frac{4}{3}}}{\left|Q_{2_{op}}\right| \left(S_s h_{2_{op}}^2 + W_2 h_{2_{op}}\right)^{\frac{7}{3}}}$$

$$\frac{\partial f_2}{\partial h_1} = 0$$

$$\begin{aligned}\frac{\partial f_3}{\partial h_1} = &-\frac{\beta Q_{1_{op}}^2\left(2S_s h_{1_{op}} + W_1\right)}{L\left(S_s h_{1_{op}}^2 + W_1 h_{1_{op}}\right)^2} + \frac{g}{L}\left(h_{1_{op}} W_1 + h_{1_{op}}^2 S_s\right) + \frac{g h_{1_{op}}}{L}(-W_1 + W_2) \\ &+ g S_b\left(2S_s h_{1_{op}} + W_1\right) \\ &- \frac{8}{3} g Q_{1_{op}}\left|Q_{1_{op}}\right| n_M^2 \frac{\sqrt{1+S_s^2}\left(W_1 + 2h_{1_{op}}\sqrt{1+S_s^2}\right)^{\frac{1}{3}}}{\left(S_s h_{1_{op}}^2 + W_1 h_{1_{op}}\right)^{\frac{7}{3}}} \\ &+ \frac{7}{3} g Q_{1_{op}}\left|Q_{1_{op}}\right| n_M^2 \frac{\left(W_1 + 2h_{1_{op}}\sqrt{1+S_s^2}\right)^{\frac{4}{3}}\left(2S_s h_{1_{op}} + W_1\right)}{\left(S_s h_{1_{op}}^2 + W_1 h_{1_{op}}\right)^{\frac{10}{3}}}\end{aligned}$$

$$\frac{\partial f_4}{\partial h_1} = \frac{-\beta Q_{1_{op}}^2\left(2S_s h_{1_{op}} + W_1\right)}{L\left(S_s h_{1_{op}}^2 + W_1 h_{1_{op}}\right)^2} + \frac{g}{L}\left(W_1 h_{1_{op}} + S_s h_{1_{op}}^2\right)$$

$$\frac{\partial g_1}{\partial h_2} = 1$$

$$\frac{\partial g_1}{\partial Q_1} = 0$$

$$\frac{\partial g_1}{\partial Q_2} = 0$$

Appendix 4 Runge-Kutta 4th order (RK4)

The detailed algorithm on RK4 is discussed here. To explain properly, a visual description of the RK4-idea is also given by Figure 12.1. Runge-Kutta fourth order algorithm uses the standard Euler forward differentiation scheme four times to approximate a function value at the point of interest. However, the name RK4 is not given based on 4 times Euler approximations. An important scheme must be understood before going through the RK4 steps. Assume a first order system described by a multivariable function with x as an independent variable and y as a dependent variable. A general first order ODE for such system is given by:

$$\frac{dy}{dx} = f(x,y) = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x}$$

$$f(x,y) = rate\ of\ change\ of\ y\ at\ (x,y) = slope\ at\ (x,y)$$

Euler forward scheme can be represented by different step sizes. Two of them are used by RK4 scheme:

- For a full step, Euler forward approximation, as shown in Figure 12.1, is:

$$y_{i+1} = y_i + \Delta y = y_i + \Delta x \cdot \frac{\Delta y}{\Delta x} = y_i + h \cdot \frac{dy}{dx} = y_i + h \cdot f(x,y)$$

$$h = x_{i+1} - x_i = \Delta x$$

- For a half step, Euler forward approximation is:

$$y_{i+\frac{1}{2}} = y_i + \Delta y = y_i + \Delta x \cdot \frac{\Delta y}{\Delta x} = y_i + \frac{h}{2} \cdot \frac{dy}{dx} = y_i + \frac{h}{2} \cdot f(x,y)$$

$$\frac{h}{2} = x_{i+\frac{1}{2}} - x_i = \Delta x$$

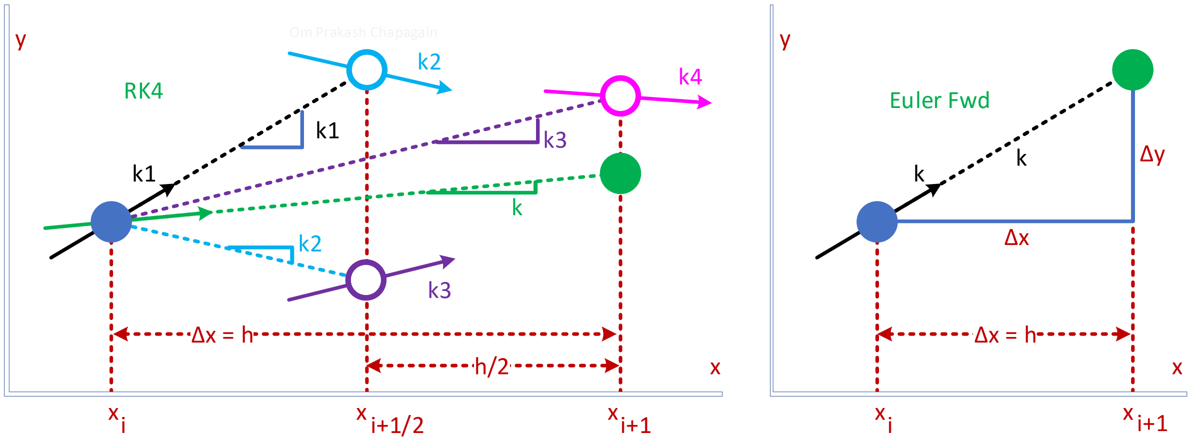

Figure 12.1: Visualization of Runge-Kutta fourth order (RK4) (left) and Euler forward (right) numerical method

RK4 algorithm for a multivariable ODE explained in steps

1. Compute the slope of the function at the point (x_i, y_i)

$$k_1 = f(x_i, y_i), \qquad k_1 = \frac{dy}{dx}\ at\ (x_i, y_i)$$

2. Using the slope k_1, predict the next value of y at the half-way (mid-way) $y_{i+\frac{1}{2}}$. This step is simply Euler forward approximation

$$y_{i+\frac{1}{2}} = y_i + \frac{h}{2} \cdot f(x_i, y_i) = y_i + \frac{h}{2} \cdot k_1$$

3. Compute the slope k_2 at this predicted value of y at the point $\left(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}\right)$

$$k_2 = f\left(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}\right), \qquad k_2 = \frac{dy}{dx}\ at\ \left(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}\right)$$

$$note\ that:, \qquad x_{i+\frac{1}{2}} = x_i + \frac{h}{2}$$

4. Using the slope k_2, predict the new value of at the mid-way (half-way) $y_{i+\frac{1}{2}}$ as above step. This step is once again Euler forward approximation. Note that the slope is calculated at the mid-way point, but it is used to predict a new value at the mid-way point from the original point.

$$y_{i+\frac{1}{2}} = y_i + \frac{h}{2} \cdot f\left(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}\right) = y_i + \frac{h}{2} \cdot k_2$$

5. Compute the slope k_3 using the new predicted value of y, again at the mid-way $\left(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}\right)$

$$k_3 = f\left(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}\right), \qquad k_3 = \frac{dy}{dx}\ at\ \left(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}\right)$$

$$note\ that: x_{i+\frac{1}{2}} = x_i + \frac{h}{2}$$

6. Using the slope k_3, predict the new value of y_{i+1} at a full step. This step is once again Euler forward approximation. Note again that the slope is calculated mid-way point, but it is used to predict a new value at the full step from the original point

$$y_{i+1} = y_i + h \cdot f\left(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}\right) = y_i + h \cdot k_3$$

7. Compute the slope k_4 at the new predicted value of y (again) at the full-step (x_{i+1}, y_{i+1})

$$k_4 = f(x_{i+1}, y_{i+1}), \qquad k_4 = \frac{dy}{dx}\ at\ (x_{i+1}, y_{i+1}), \qquad note\ that: x_{i+1} = x_i + h$$

8. Compute the average slope k using all the slopes found above. Since k_2 and k_3 are computed at the half-way, weight them twice

$$k = \frac{(k_1 + k_2 + k_2 + k_3 + k_3 + k_4)}{6} = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \qquad average\ of\ 6\ slopes$$

9. Finally, using the averaged slope k, predict the actual value of y_{i+1} at a full step. This step is, as usual, Euler forward approximation

$$y_{i+1} = y_i + h \cdot f(x_{i+1}, y_{i+1}) = y_i + h \cdot k$$

Appendix 5 Differentiation axioms including absolute valued function

Some important axioms in calculus:

- Chain rule

$$\frac{d}{dx}\Big(u\big(p(x)\big)\Big) = \frac{du}{dp}\frac{dp}{dx} \tag{a6}$$

  - Example

$$\frac{d}{dx}\frac{1}{(ax+b)} = \frac{d(ax+b)^{-1}}{d(ax+b)}\frac{d(ax+b)}{dx} = -a(ax+b)^{-2} = -\frac{a}{(ax+b)^2}$$

- Quotient rule

$$\frac{d}{dx}\left(\frac{u\big(p(x)\big)}{v\big(q(x)\big)}\right) = \frac{v\frac{du}{dp}\frac{dp}{dx} - u\frac{dv}{dq}\frac{dq}{dx}}{v^2} \tag{a7}$$

  - Example

$$\frac{d}{dx}\frac{(ax+b)^2}{\sqrt{cx}} = \frac{\sqrt{cx}\frac{d(ax+b)^2}{d(ax+b)}\frac{d(ax+b)}{dx} - (ax+b)^2\frac{d\sqrt{cx}}{d(cx)}\frac{d(cx)}{dx}}{cx}$$

$$= \frac{2a(ax+b)\sqrt{cx} - \frac{c(ax+b)^2}{2\sqrt{cx}}}{cx} = \frac{2a(ax+b)}{\sqrt{cx}} - \frac{c(ax+b)^2}{2\sqrt{(cx)^3}}$$

- Product rule

$$\frac{d}{dt}(uv) = u\frac{dv}{dt} + \frac{du}{dt}v = u\dot{v} + \dot{u}v \tag{a8}$$

  - Example

$$\frac{d}{dt}(at+b)(ct)^2 = (at+b)\frac{d}{dt}(ct)^2 + (ct)^2\frac{d}{dt}(at+b)$$

$$= (at+b)\frac{d(ct)^2}{d(ct)}\frac{d(ct)}{dt} + (ct)^2\frac{d}{dt}(at+b) = 2c^2t(at+b) \ + a(ct)^2$$

- Proof of differentiation of absolute valued function

$$\frac{d}{dx}(x\cdot|x|) = \frac{d}{dx}\left(\sqrt{x^2}\cdot\sqrt{x^2}\right) = \frac{d}{dx}\left(\sqrt{x^4}\right) = \frac{d}{dx}(x^4)^{\frac{1}{2}}$$

$$= \frac{d}{d(x^4)}(x^4)^{\frac{1}{2}}\frac{dx^4}{dx} = \frac{1}{2}(x^4)^{\frac{1}{2}-1}4x^3 = \frac{2x^3}{(x^4)^{\frac{1}{2}}} = \frac{2x^3}{\sqrt{x^4}} = \frac{2x^3}{\sqrt{x^2}\cdot\sqrt{x^2}} = \frac{2x^3}{x\cdot|x|} = \frac{2x^2}{|x|}$$

$$\frac{d}{dx}(x\cdot|x|) = \frac{2x^2}{|x|} \tag{a9}$$

All the axioms above are used during linearization of the nonlinear ODEs.

Appendix 6 Derivation of analytical solution to linear LTI system

LTI system means linear time invariant system. Time invariant means that the model matrices (A, B, E, C, D, F) are constant for all time instances. In other words, LTI system has known model matrices that are independent of the time.

- For a non-singular matrix Λ, following expression holds true

$$e^{\Lambda t}\Lambda = \Lambda e^{\Lambda t} \tag{a10}$$

- Differentiation of matrix exponential

$$(-\Lambda)e^{-\Lambda t} = \frac{d}{dt}e^{-\Lambda t} = \left(e^{\dot{-\Lambda t}}\right) \tag{a11}$$

Part 6A *Analytical solution to linear state space model*

For a combined deterministic and stochastic LTI system, state equation the form:

$$\dot{x} = Ax + Bu + Ew$$

$$y = Cx + Du + Fv$$

Steps to find the analytical solution:

1. Rewrite the above equation in the form:

$$\dot{x} - Ax = Bu + Ew$$

2. Multiply both sides with e^{-At}

$$e^{-At}\dot{x} - e^{-At}Ax = e^{-At}Bu + e^{-At}Ew$$

3. Use the axiom (a10) rewrite the above equation as:

$$e^{-At}\dot{x} + (-A)e^{-At}x = e^{-At}Bu + e^{-At}Ew$$

4. Use the axiom (a11) rewrite the above equation as:

$$e^{-At}\frac{dx}{dt} + \left(\frac{d}{dt}e^{-At}\right)x = e^{-At}Bu + e^{-At}Ew$$

5. Using the axiom (a8), rewrite the above equation as:

$$\frac{d}{dt}\left(e^{-A\tau}x(t)\right) = e^{-At}Bu(t) + e^{-At}Ew(t)$$

6. Integrate both sides with respect to time. To avoid ambiguity, lets integrate both sides with an arbitrary time τ.

$$\int_{t_0}^{t}\frac{d}{d\tau}\left(e^{-A\tau}x(\tau)\right)d\tau = \int_{t_0}^{t}e^{-A\tau}Bu(\tau)d\tau + \int_{t_0}^{t}e^{-A\tau}Ew(\tau)d\tau$$

$$\left(e^{-A\tau}x(\tau)\right)\Big|_{t_0}^{t} = \int_{t_0}^{t}e^{-A\tau}Bu(\tau)d\tau + \int_{t_0}^{t}e^{-A\tau}Ew(\tau)d\tau$$

$$e^{-At}x(t) - e^{-At_0}x(t_0) = \int_{t_0}^{t}e^{-A\tau}Bu(\tau)d\tau + \int_{t_0}^{t}e^{-A\tau}Ew(\tau)d\tau$$

$$e^{-At}x(t) = e^{-At_0}x(t_0) + \int_{t_0}^{t} e^{-A\tau}Bu(\tau)d\tau + \int_{t_0}^{t} e^{-A\tau}Ew(\tau)d\tau$$

$$x(t) = e^{At}e^{-At_0}x(t_0) + e^{At}\int_{t_0}^{t} e^{-A\tau}Bu(\tau)d\tau + e^{At}\int_{t_0}^{t} e^{-A\tau}Ew(\tau)d\tau$$

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau + \int_{t_0}^{t} e^{A(t-\tau)}Ew(\tau)d\tau$$

7. Define transition matrix, $\Phi(t)$

$$\Phi(t) = e^{At}, \qquad \begin{matrix} \Phi(t-t_0) = e^{A(t-t_0)} \\ \Phi(t-\tau) = e^{A(t-\tau)} \end{matrix} \tag{a12}$$

$$x(t) = \Phi(t-t_0)x(t_0) + \int_{t_0}^{t} \Phi(t-\tau)Bu(\tau)d\tau + \int_{t_0}^{t} \Phi(t-\tau)Ew(\tau)d\tau$$

8. The solution to the LTI system from above can be directly used to compute the output

$$y(t) = C\Phi(t-t_0)x(t_0) + C\int_{t_0}^{t} \Phi(t-\tau)Bu(\tau)d\tau + C\int_{t_0}^{t} \Phi(t-\tau)Ew(\tau)d\tau + Du(t) + Fv(t)$$

9. To simplify the solution, it is a good idea to set $t_0 = 0$ and see at some special cases.

$$x(t) = \Phi(t)x(0) + \int_{0}^{t} \Phi(t-\tau)Bu(\tau)d\tau + \int_{0}^{t} \Phi(t-\tau)Ew(\tau)d\tau$$

$$y(t) = C\Phi(t)x(0) + C\int_{0}^{t} \Phi(t-\tau)Bu(\tau)d\tau + C\int_{0}^{t} \Phi(t-\tau)Ew(\tau)d\tau + Du(t) + Fv(t)$$

- Case 1: Deterministic system i.e., $E = F = \mathbf{0}$ matrix

$$x(t) = \Phi(t)x(0) + \int_{0}^{t} \Phi(t-\tau)Bu(\tau)d\tau$$

  - 1a: With direct feedthrough term in the output equation i.e., $D \neq \mathbf{0}$

$$y(t) = C\Phi(t)x(0) + C\int_{0}^{t} \Phi(t-\tau)Bu(\tau)d\tau + Du(t)$$

  - 1b: Without direct feedthrough term in the output i.e., $D = \mathbf{0}$ matrix

$$y(t) = C\Phi(t)x(0) + C\int_{0}^{t} \Phi(t-\tau)Bu(\tau)d\tau$$

Part 6B *Analytical solution to linear state observer using duality principle*

$$\dot{\hat{x}} = A\hat{x} + Bu + L(\bar{y} - \hat{y})$$

$$\hat{y} = C\hat{x}$$

$$\dot{\hat{x}} = A\hat{x} + Bu + L\bar{y} - L(C\hat{x})$$

$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + L\bar{y}$$

$$(A - LC) \overset{\text{def}}{=} \Lambda \tag{a13}$$

$$\dot{\hat{x}} = \Lambda\hat{x} + Bu + L\bar{y}$$

$$\dot{\hat{x}} - \Lambda\hat{x} = Bu + L\bar{y}$$

- Solution to the state observer can be found by using duality principle as shown in Table 12.2. Duality principle states that; if the solution to the state space model is known, then it is possible to directly write the solution to the state observer model.

Table 12.2: Duality with between linear stochastic state space equation and observer equation

Combined deterministic and stochastic system	Observer state equation
$\dot{x} - Ax = Bu + Ew$	$\dot{\hat{x}} - \Lambda\hat{x} = Bu + L\bar{y}$
$y = Cx$	$\hat{y} = C\hat{x}$
$\dot{x}$	$\dot{\hat{x}}$
A	Λ
x	$\hat{x}$
E	L
w	$\bar{y}$
y	$\hat{y}$
$\Phi(t-t_0) \overset{\text{def}}{=} e^{A(t-t_0)}$ $\Phi(t-\tau) \overset{\text{def}}{=} e^{A(t-\tau)}$	$\Omega(t-t_0) \overset{\text{def}}{=} e^{\Lambda(t-t_0)}$ $\Omega(t-\tau) \overset{\text{def}}{=} e^{\Lambda(t-\tau)}$ (a14)
$x(t) = \Phi(t-t_0)x(t_0) + \int_{t_0}^{t}\Phi(t-\tau)Bu(\tau)d\tau + \int_{t_0}^{t}\Phi(t-\tau)Ew(\tau)d\tau$	$\hat{x}(t) = \Omega(t-t_0)\hat{x}(t_0) + \int_{t_0}^{t}\Omega(t-\tau)Bu(\tau)d\tau + \int_{t_0}^{t}\Omega(t-\tau)L\bar{y}(\tau)d\tau$
$y(t) = C\Phi(t-t_0)x(t_0) + C\int_{t_0}^{t}\Phi(t-\tau)Bu(\tau)d\tau + C\int_{t_0}^{t}\Phi(t-\tau)Ew(\tau)d\tau$	$\hat{y}(t) = C\Omega(t-t_0)\hat{x}(t_0) + C\int_{t_0}^{t}\Omega(t-\tau)Bu(\tau)d\tau + C\int_{t_0}^{t}\Omega(t-\tau)L\bar{y}(\tau)d\tau$

Part 6C *Analytical solution to the estimation error dynamics*

Steps:

1. Define error dynamics

$$\varepsilon \overset{\text{def}}{=} x - \hat{x} \quad \dot{\varepsilon} \overset{\text{def}}{=} \dot{x} - \dot{\hat{x}}$$

$$\varepsilon \overset{\text{def}}{=} x - \hat{x} \quad \lim_{t\to\infty} \varepsilon \to 0 \quad \lim_{\varepsilon\to 0} \hat{x} \to x$$

2. Determine expression for the error dynamics

$$\dot{\varepsilon} = Ax + Bu - A\hat{x} - Bu - L(y - \hat{y})$$

$$\dot{\varepsilon} = Ax - A\hat{x} - LCx + LC\hat{x}$$

$$\dot{\varepsilon} = A(x - \hat{x}) - LC(x - \hat{x})$$

$$\dot{\varepsilon} = (A - LC)(x - \hat{x})$$

3. Using axiom (a13), rewrite above equation incorporating the error transition matrix Λ

$$\dot{\varepsilon} = \Lambda\varepsilon$$

4. Multiply both sides with $e^{-\Lambda t}$ and move everything to the left side

$$e^{-\Lambda t}\dot{\varepsilon} = e^{-\Lambda t}\Lambda\varepsilon \equiv e^{-\Lambda t}\dot{\varepsilon} - e^{-\Lambda t}\Lambda\varepsilon = 0$$

5. Using axiom (a10) and (a11), express error dynamics as:

$$e^{-\Lambda t}\dot{\varepsilon} + (-\Lambda)e^{-\Lambda t}\varepsilon = 0$$

$$e^{-\Lambda t}\dot{\varepsilon} + (\dot{e^{-\Lambda t}})\varepsilon = 0$$

6. Write the compact form of error dynamics using (a8)

$$\frac{d}{dt}(e^{-\Lambda t}\varepsilon) = 0$$

7. Integrate both sides w.r.t. time

$$\int_{t_0}^{t} \frac{d}{d\tau}\left(e^{-\Lambda\tau}\varepsilon(\tau)\right) d\tau = 0$$

$$\left(e^{-\Lambda\tau}\varepsilon(\tau)\right)\Big|_{t_0}^{t} = 0$$

$$e^{-\Lambda t}\varepsilon(t) - e^{-\Lambda t_0}\varepsilon(t_0) = 0$$

$$e^{-\Lambda t}\varepsilon(t) = e^{-\Lambda t_0}\varepsilon(t_0)$$

$$\varepsilon(t) = e^{\Lambda t}e^{-\Lambda t_0}\varepsilon(t_0)$$

$$\varepsilon(t) = e^{\Lambda(t-t_0)}\varepsilon(t_0)$$

8. Using axiom (a14), rewrite above equation in compact form. The result is the solution of error dynamic

$$\varepsilon(t) = \Omega(t - t_0)\varepsilon(t_0)$$

- Case 1: when $t_0 = 0$

$$\varepsilon(t) = \Omega(t)\varepsilon(0)$$

Appendix 7 Sensitivity analysis of all the parameters on all the states

Colum representation for subsequent tables:

- Parameters = θ_i
- Mean of states based on the nominal values, x = μ_x
- Standard deviation of states based on the nominal values, x = σ_x
- Mean of sensitivity coefficient, ε = μ_ε
- Standard deviation of sensitivity coefficient, ε = σ_ε
- Sensitivity index = SI
- Correlation = r
- Percent change in the state $(\Delta x\%)$ induced by -10% change in parameter value based on nominal value = $\Delta\theta\%^- \rightarrow \Delta x\%$
- Percent change in the state $(\Delta x\%)$ induced by -10% change in parameter value based on nominal value = $\Delta\theta\%^+ \rightarrow \Delta x\%$

Part 7A ***Parameter sensitivity analysis on h_2***

θ_i	μ_ε $\times 10^{-2}$	σ_ε $\times 10^{-2}$	μ_x $[mm]$	σ_x $[mm]$	SI	r	$\Delta\theta\%^-$ $\rightarrow \Delta x\%$	$\Delta\theta\%^+$ $\rightarrow \Delta x\%$
α	*-46.3*	*3.1*	*34.8*	*0.9*	*8.9*	*-0.998*	*5.2*	*-4.2*
β	*-111.6*	*6.8*	*34.8*	*2.3*	*20*	*-0.998*	*12.4*	*-10.1*
h_1	*18.8*	*0.9*	*34.7*	*0.4*	*3.7*	*0.999*	*-2.1*	*1.7*
k_s	*-223.4*	*20.4*	*35.1*	*4.6*	*35.7*	*-0.996*	*25.9*	*-19.1*
L	*111.2*	*0.3*	*34.7*	*2.3*	*20*	*1*	*-11.1*	*11.1*
ϕ	*0*	*0*	*34.8*	*2.5*	*21.5*	*0.998*	*:*	*:*
ρ	*0*	*0*	*34.7*	*0*	*0*	*0.152*	*0*	*0*
W_1	*-245.7*	*34.4*	*35.4*	*5.1*	*38.6*	*-0.993*	*31.1*	*-19.4*
W_2	*175.2*	*11*	*34.9*	*3.6*	*29.5*	*0.998*	*-15.7*	*19.5*

Parameter values for both flow rates at the steady state is same. Hence, one table is enough.

Part 7B ***Parameter sensitivity analysis on M_1 and M_2***

θ_i	$\mu_\varepsilon \times 10^{-2}$	$\sigma_\varepsilon \times 10^{-2}$	μ_x $\left[\frac{kg}{min}\right]$	σ_x $\left[\frac{kg}{min}\right]$	SI	r	$\Delta\theta\%^- \rightarrow \Delta x\%$	$\Delta\theta\%^+ \rightarrow \Delta x\%$
α	*-76.6*	*3.8*	*370.8*	*16.7*	*14.2*	*-0.999*	*8.4*	*-7.1*
β	*-95.5*	*3.7*	*370.8*	*20.7*	*17.3*	*-0.999*	*10.2*	*-8.9*
h_1	*131.7*	*0.1*	*370*	*28.6*	*23.3*	*1*	*-13.2*	*13.2*
k_s	*-89.2*	*3.3*	*369.4*	*19*	*15.8*	*-0.999*	*8*	*-9.1*
L	*45.5*	*2.5*	*369.5*	*9.9*	*8.7*	*0.998*	*-5*	*4.1*
ϕ	*0*	*0*	*370.5*	*17.1*	*14.6*	*0.999*	*:*	*:*
ρ	*100*	*0*	*370*	*21.7*	*18.2*	*1*	*-10*	*10*
W_1	*-111.9*	*5.4*	*371.1*	*24*	*19.5*	*-0.999*	*11.6*	*-10.1*
W_2	*170.3*	*4.3*	*370.9*	*36.9*	*28.8*	*0.999*	*-16.2*	*17.6*

Appendix 8 Matrix product and summation

This appendix proves that the sum of a vector square is the same as the matrix multiplication. The vector is extracted from a matrix. The matrix multiplication takes care of the summation sign of sigma notation. In other words, the sum of the product of each column of a matrix with its transpose can be written in sigma notation or as a matrix product.

- Take an example of a rectangular matrix E given as:

$$E = \begin{bmatrix} e_{(1,1)} & \cdots & e_{(1,n)} \\ \vdots & \ddots & \vdots \\ e_{(m,1)} & \cdots & e_{(m,n)} \end{bmatrix}$$

- Take a column vector from the matrix E as.

$$\mathbf{e}_i = E_{(:,i)} = \begin{bmatrix} e_{(1,i)} \\ \vdots \\ e_{(m,i)} \end{bmatrix}$$

$$\mathbf{e}_i = column\ vector\ extracted\ from\ E$$

- Sum of the squared vector (multiplied itself by its transpose) is written as:

$$V = \sum_{i=1}^{n} \mathbf{e}_i \cdot \mathbf{e}_i^T = \sum_{i=1}^{n} E_{(:,i)} \cdot E_{(:,i)}{}^T$$

- The proof:
  - Expand the vector product inside of the signa notation

$$V = \mathbf{e}_1 \cdot \mathbf{e}_1^T + \cdots + \mathbf{e}_n \cdot \mathbf{e}_n^T$$

$$V = E_{(:,1)} \cdot E_{(:,1)}{}^T + \cdots + E_{(:,n)} \cdot E_{(:,n)}{}^T$$

  - Rewrite the expanded sum in compact form

$$V = \sum_{i=1}^{n} \begin{bmatrix} e_{(1,i)} \\ \vdots \\ e_{(m,i)} \end{bmatrix} \begin{bmatrix} e_{(1,i)} & \cdots & e_{(m,i)} \end{bmatrix}$$

$$V = \begin{bmatrix} e_{(1,1)} \\ \vdots \\ e_{(m,1)} \end{bmatrix} \begin{bmatrix} e_{(1,1)} & \cdots & e_{(m,1)} \end{bmatrix} + \cdots + \begin{bmatrix} e_{(1,n)} \\ \vdots \\ e_{(m,n)} \end{bmatrix} \begin{bmatrix} e_{(1,n)} & \cdots & e_{(m,n)} \end{bmatrix}$$

$$V = \begin{bmatrix} e_{(1,1)}e_{(1,1)} & \cdots & e_{(1,1)}e_{(m,1)} \\ \vdots & \ddots & \vdots \\ e_{(m,1)}e_{(1,1)} & \cdots & e_{(m,1)}e_{(m,1)} \end{bmatrix} + \cdots + \begin{bmatrix} e_{(1,n)}e_{(1,n)} & \cdots & e_{(1,n)}e_{(m,n)} \\ \vdots & \ddots & \vdots \\ e_{(m,n)}e_{(1,n)} & \cdots & e_{(m,n)}e_{(m,n)} \end{bmatrix}$$

$$V = \begin{bmatrix} e_{(1,1)}e_{(1,1)} + \cdots + e_{(1,n)}e_{(1,n)} & \cdots & e_{(1,1)}e_{(m,1)} + \cdots + e_{(1,n)}e_{(m,n)} \\ \vdots & \ddots & \vdots \\ e_{(m,1)}e_{(1,1)} + \cdots + e_{(m,n)}e_{(1,n)} & \cdots & e_{(m,1)}e_{(m,1)} + \cdots + e_{(m,n)}e_{(m,n)} \end{bmatrix}$$

$$V = \begin{bmatrix} e_{(1,1)} & \cdots & e_{(1,n)} \\ \vdots & \ddots & \vdots \\ e_{(m,1)} & \cdots & e_{(m,n)} \end{bmatrix} \begin{bmatrix} e_{(1,1)} & \cdots & e_{(m,1)} \\ \vdots & \ddots & \vdots \\ e_{(1,n)} & \cdots & e_{(m,n)} \end{bmatrix} = EE^T \quad Q.E.D$$

$$V = \sum_{i=1}^{n} E_{(:,i)} \cdot E_{(:,i)}{}^T = EE^T \tag{a15}$$

Appendix 9 Estimator applied on simulator excited by noise-free input signal

Some of the figures to supplement the analysis performed in the previous chapters are given here.

Part 9A *Comparison of Online and Offline Kalman filter gain*

Comparison of Figure 12.2 and Figure 12.3 proves that the difference in the estimated states is negligible when using online or offline Kalman gain in LKF. However, offline Kalman gain is efficient when running the system continuously (in loop), which is always the case. It is efficient because, Kalman gain need not be calculated for each iteration in contrast to online Kalman gain which is calculated during each iteration.

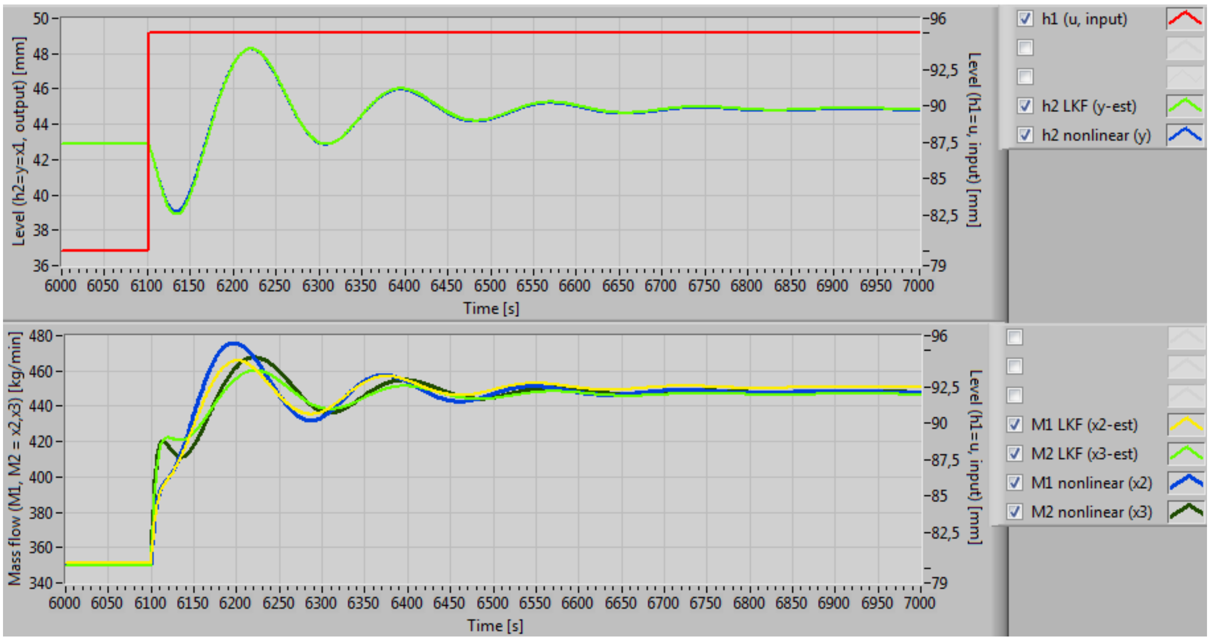

Figure 12.2: Estimated states using LKF with offline calculated Kalman gain

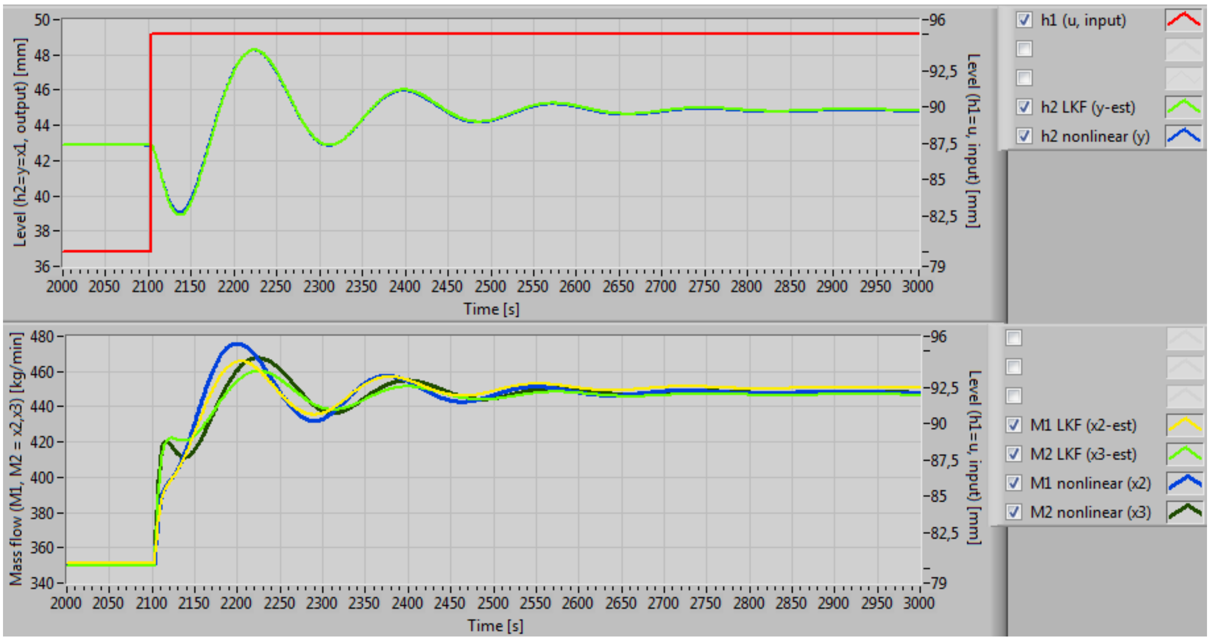

Figure 12.3: Estimated states using LKF with online calculated Kalman gain

Part 9B *Comparison of LKF, EKF and UKF*

As with the noisy input and output case (discussed in 8.3.2), EKF is faster than LKF. There is no deviation in case of EKF and UKF. UKF is superior to both EKF and LKF in terms of convergence rate. Since no noise is added in the input signal, the filtering property of these filters is not evident. One important observation based on Figure 12.4 and Figure 12.5 is that EKF estimates are exactly the same as the nonlinear model prediction.

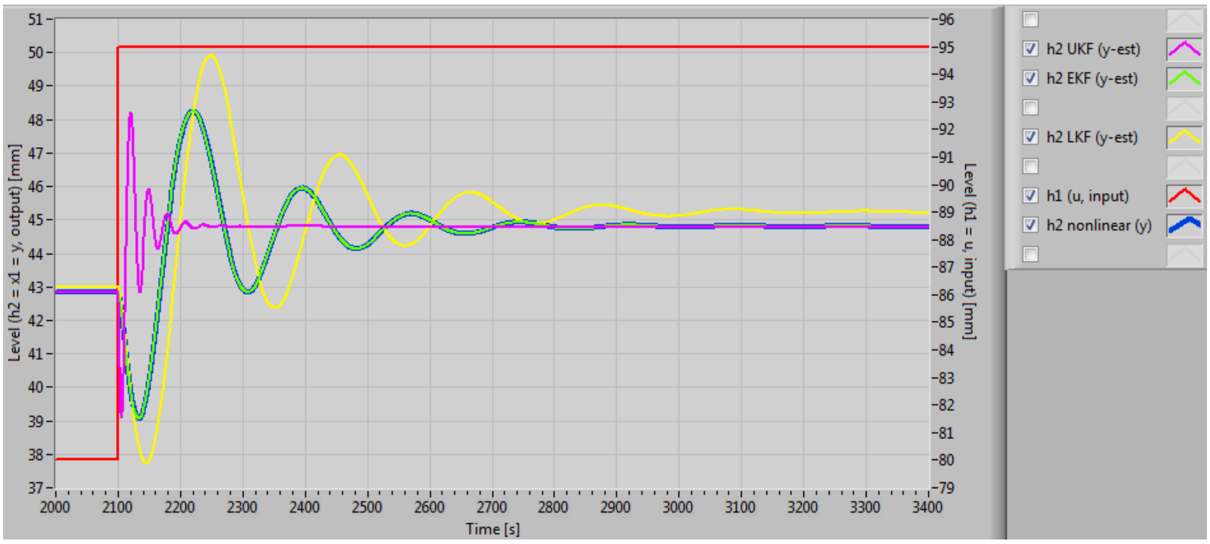

Figure 12.4: Comparison of Kalman filters based on level estimation $(\hat{y} = \hat{x}_1 = \hat{h}_2)$. Simulator is excited by the noise-free input $(u = h_1)$

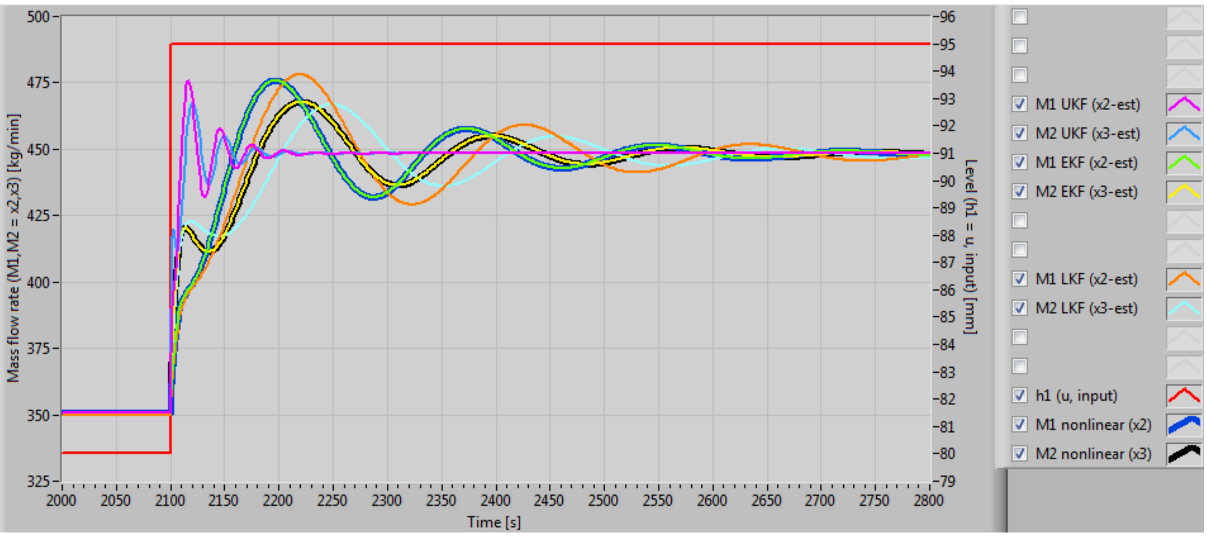

Figure 12.5: Comparison of Kalman filters based on estimation of mass flow rate $(\hat{x}_2 = \widehat{M}_1, \hat{x}_3 = \widehat{M}_2)$. Simulator is excited the noise-free input $(u = h_1)$

Part 9C *Comparison of linear estimators (LKF and linear observer)*

Figure 12.6 and Figure 12.7 show the result of state estimation based on noise-free system input. These figures complement the result of comparison for system excited with noisy system as discussed in section 8.3.3. Both LKF and linear observer perform relatively similarly and both estimators show the deviation away from the linearization point.

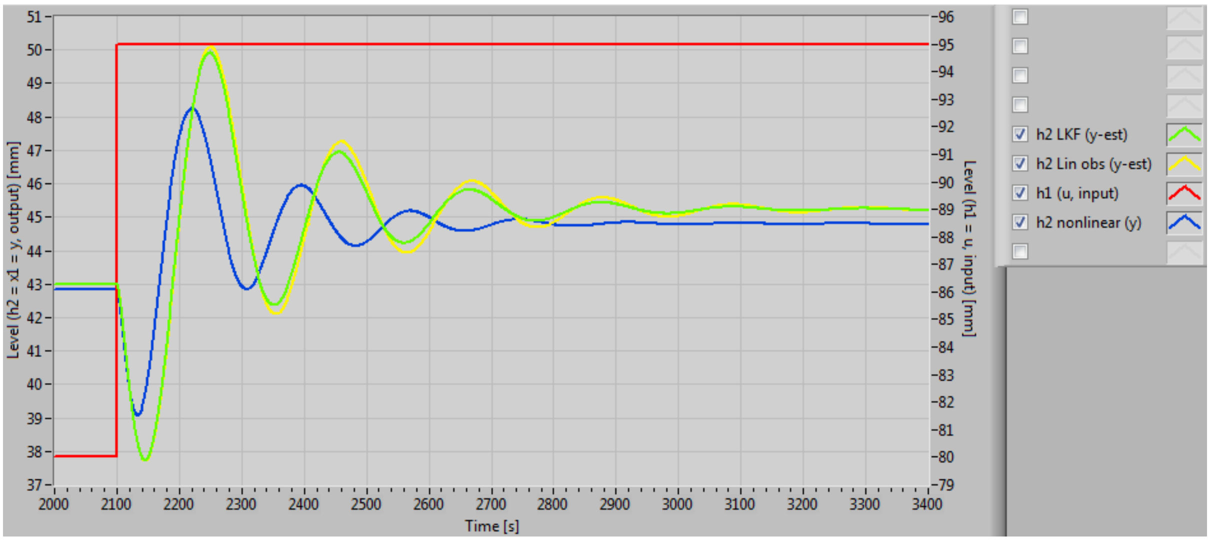

Figure 12.6: Comparison of linear estimators based on level estimation $(\hat{y} = \hat{x}_1 = \hat{h}_2)$. Simulator is excited by the noise-free input $(u = h_1)$

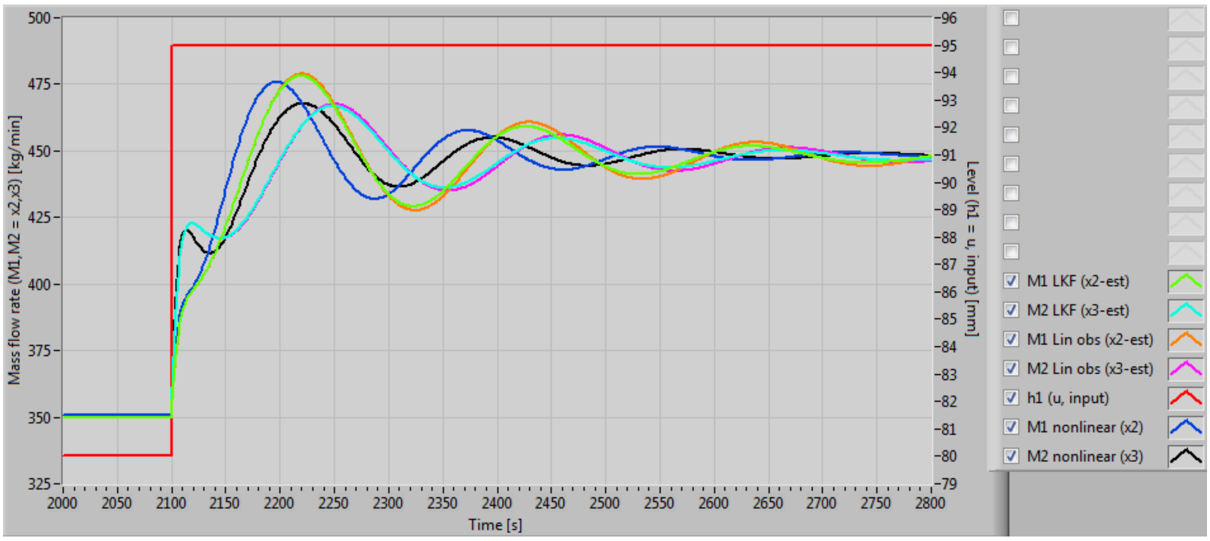

Figure 12.7: Comparison of linear estimators based on estimation of mass flow rate $(\hat{x}_2 = \widehat{M}_1, \hat{x}_3 = \widehat{M}_2)$. Simulator is excited the noise-free input $(u = h_1)$

Part 9D ***Comparison of nonlinear estimators (EKF, UKF and semi-nonlinear observer)***

Nonlinear estimators for noisy system are compared in detail in section 8.3.4. In case of noise-free system, EKF and semi-nonlinear observers performed exactly the same. Their signal in the Figure 12.8 and Figure 12.9 are super imposed with the predicted states by the nonlinear venturi model. Even in noise-free system, UKF outperforms EKF and semi-nonlinear observer. All nonlinear estimators show no deviation from the model states.

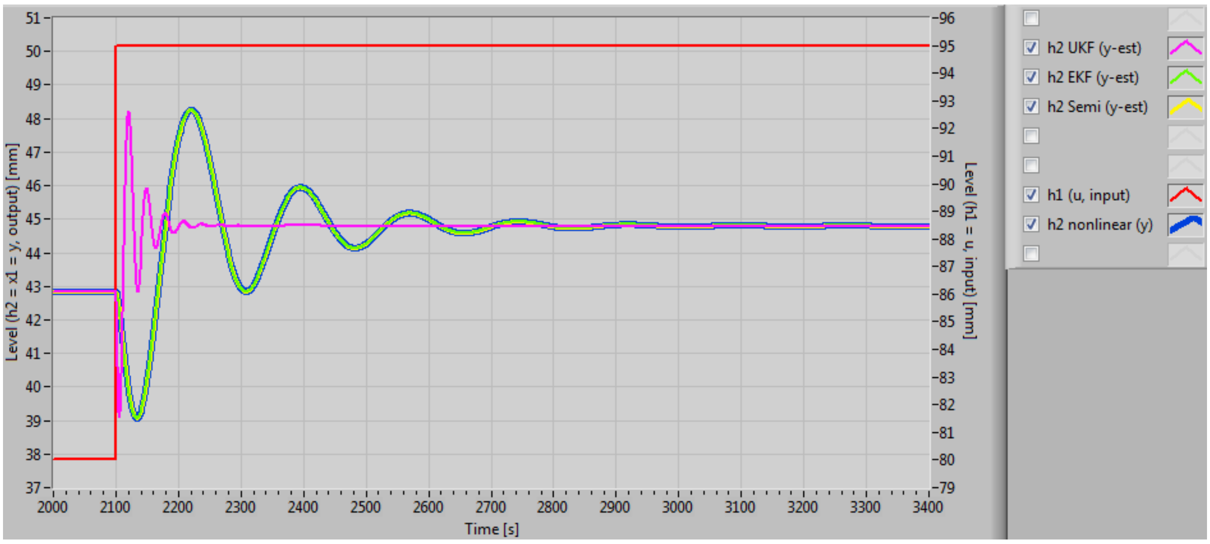

Figure 12.8: Comparison of nonlinear estimators based on level estimation $\left(\hat{y} = \hat{x}_1 = \hat{h}_2\right)$. Simulator is excited by the noise-free input $(u = h_1)$

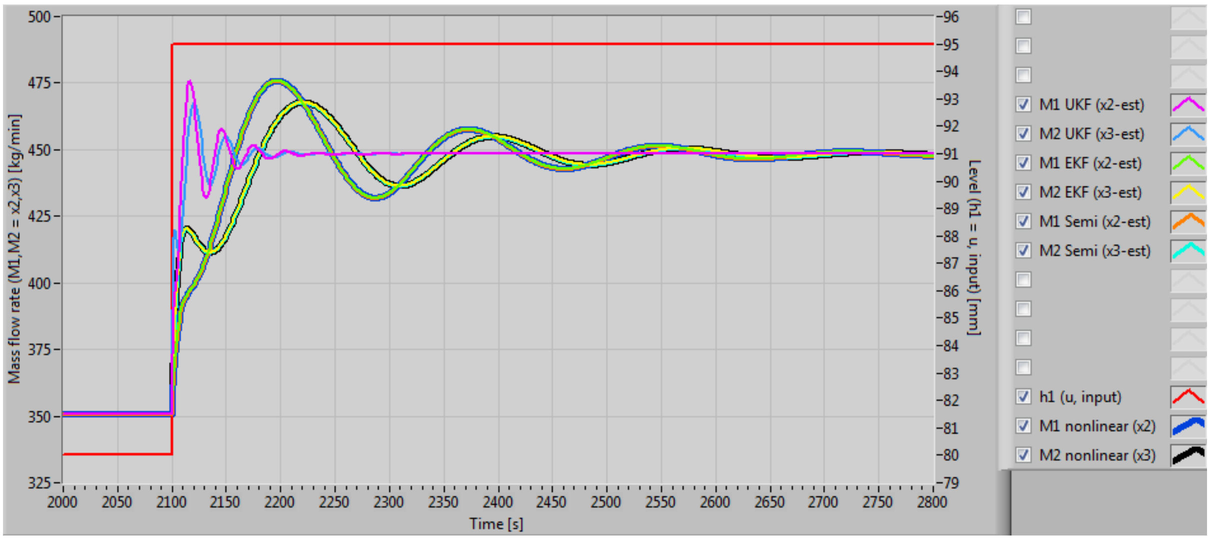

Figure 12.9: Comparison of nonlinear estimators based on estimation of mass flow rate $\left(\hat{x}_2 = \widehat{M}_1, \hat{x}_3 = \widehat{M}_2\right)$. Simulator is excited the noise-free input $(u = h_1)$

Appendix 10 Reduced order observer derivation

Consider a linear state space model given below:

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

Decomposing the state equation into measured and unmeasured states.

$$\begin{bmatrix} \dot{x}_p \\ \dot{x}_q \end{bmatrix} = \begin{bmatrix} A_{pp} & A_{pq} \\ A_{qp} & A_{qq} \end{bmatrix} \begin{bmatrix} x_p \\ x_q \end{bmatrix} + \begin{bmatrix} B_p \\ B_q \end{bmatrix} [u]$$

$$[y] = \begin{bmatrix} C_p & C_q \end{bmatrix} \begin{bmatrix} x_p \\ x_q \end{bmatrix}$$

$$n_x = n_p + n_q$$

Where, x_p is a vector of measured states and x_q is a vector of unmeasured states, n_x is the number of total states (system order), $n_p = n_y$ is the number of measured states and n_q is the number of unmeasured states.

Expanding the linear model equations, we get:

$$\dot{x}_p = A_{pp}x_p + A_{pq}x_q + B_p u$$

$$\dot{x}_q = A_{qp}x_p + A_{qq}x_q + B_q u$$

$$y = C_p x_p + C_q x_q$$

Since only some of the states are directly measured, in this case x_p, the output equation can be simplified by setting $C_q = zeros(n_y, n_q)$.

$$y = C_p x_p$$

Output could be the combination of multiple measured states (soft-sensing). In case of a single measured state and single output system, $C_p = zeros(1, n_p)$. For generality, lets proceed with the equations above. C_p could be scalar, vector or a matrix. When C_p is a vector or a rectangular matrix, the pseudo-inverse is required to represent x_p in terms of y.

$$\left(C_p^T C_p\right)^{-1} C_p^T y = x_p$$

$$x_p = D_p y, \qquad D_p = \left(C_p^T C_p\right)^{-1} C_p^T$$

In case of the linear observer equations, the expansion can be done in the similar way as the model equation.

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y})$$

$$\hat{y} = C\hat{x}$$

$$\begin{bmatrix} \dot{\hat{x}}_p \\ \dot{\hat{x}}_q \end{bmatrix} = \begin{bmatrix} A_{pp} & A_{pq} \\ A_{qp} & A_{qq} \end{bmatrix} \begin{bmatrix} \hat{x}_p \\ \hat{x}_q \end{bmatrix} + \begin{bmatrix} B_p \\ B_q \end{bmatrix} [u] + \begin{bmatrix} L_p \\ L_q \end{bmatrix} (y - \hat{y})$$

$$[y] = \begin{bmatrix} C_p & C_q \end{bmatrix} \begin{bmatrix} \hat{x}_p \\ \hat{x}_q \end{bmatrix}$$

Expanding the observer equations, we get:

$$\dot{\hat{x}}_p = A_{pp}\hat{x}_p + A_{pq}\hat{x}_q + B_p u + L_p y - L_p\hat{y}$$

$$\dot{\hat{x}}_q = A_{qp}\hat{x}_p + A_{qq}\hat{x}_q + B_q u + L_q y - L_q\hat{y}$$

$$\hat{y} = C_p\hat{x}_p + C_q\hat{x}_q$$

The purpose of the reduced order observer is to estimate only the states that are not measured. Hence, $\hat{x}_p$ can be removed from the observer output equation as well as it can be replaced with x_p. It also makes sense to use the model equation instead of the first state estimation equation which does not need to be estimated.

$$\dot{x}_p = A_{pp}x_p + A_{pq}x_q + B_p u$$

$$\dot{\hat{x}}_q = A_{qp}x_p + A_{qq}\hat{x}_q + B_q u + L_q y - L_q\hat{y}$$

$$\hat{y} = C_q\hat{x}_q$$

Keeping everything known in LHS, above equations can be simplified further.

$$Define, \qquad y = \dot{x}_p - A_{pp}x_p - B_p u = A_{pq}x_q = Cx$$

$$y = \dot{x}_p - A_{pp}x_p - B_p u, \qquad C = A_{pq}, \qquad x = x_q$$

$$y = A_{pq}x_q$$

$$\dot{\hat{x}}_q = A_{qq}\hat{x}_q + A_{qp}x_p + B_q u + L_q y - L_q\hat{y}$$

$$\hat{y} = C_q\hat{x}_q$$

$$\hat{y} = A_{pq}\hat{x}_q, \qquad C_q = C = A_{pq}$$

Comparing these equations with the linear reduced order state-observer equation with the similar form to that of the full order observer, we get:

$$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}u + \bar{L}(y - \bar{y})$$

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

$$\dot{\bar{x}} = \dot{\hat{x}}_q$$

$$\bar{A} = A_{qq}$$

$$\bar{x} = \hat{x}_q$$

$$\bar{B}u = A_{qp}x_p + B_q u$$

$$\bar{L} = L_q$$

$$y = \dot{x}_p - A_{pp}x_p - B_p u$$

$$\bar{y} = \hat{y}$$

$$\bar{C} = C_q = A_{pq}$$

Substituting above equality equations in the reduced order observer, we get:

$$\dot{\hat{x}}_q = A_{qq}\hat{x}_q + A_{qp}x_p + B_q u + L_q\dot{x}_p - L_q A_{pp}x_p - L_q B_p u - L_q A_{pq}\hat{x}_q$$

$$\dot{\hat{x}}_q - L_q\dot{x}_p = \left(A_{qq} - L_q A_{pq}\right)\hat{x}_q + \left(B_q - L_q B_p\right)u + \left(A_{qp} - L_q A_{pp}\right)x_p$$

Define new state variable such that the derivative term to the measured state variable is eliminated from the model:

$$z = x_q - L_q x_p, \qquad \hat{z} = \hat{x}_q - L_q x_p, \qquad \dot{\hat{z}} = \dot{\hat{x}}_q - L_q \dot{x}_p, \qquad \hat{x}_q = \hat{z} + L_q x_p$$

$$\dot{\hat{z}} = \left(A_{qq} - L_q A_{pq}\right)\left(\hat{z} + L_q x_p\right) + \left(B_q - L_q B_p\right)u + \left(A_{qp} - L_q A_{pp}\right)x_p$$

$$\dot{\hat{z}} = \left(A_{qq} - L_q A_{pq}\right)\hat{z} + \left(A_{qq} - L_q A_{pq}\right)L_q x_p + \left(B_q - L_q B_p\right)u + \left(A_{qp} - L_q A_{pp}\right)x_p$$

$$\dot{\hat{z}} = \left(A_{qq} - L_q A_{pq}\right)\hat{z} + \left(B_q - L_q B_p\right)u + \left(A_{qq} L_q - L_q A_{pq} L_q + A_{qp} - L_q A_{pp}\right)x_p \qquad \text{(a16)}$$

Solution to the $\dot{\hat{z}}$ gives out $\hat{z}$, from which the missing states can be extracted simply by using the equation:

$$\hat{x}_q = \hat{z} + L_q x_p \qquad \text{(a17)}$$

Define the error dynamics:

$$\varepsilon_q \overset{\text{def}}{=} x_q - \hat{x}_q \quad \dot{\varepsilon}_q \overset{\text{def}}{=} \dot{x}_q - \dot{\hat{x}}_q \qquad \text{(a18)}$$

$$\dot{x}_q - \dot{\hat{x}}_q = A_{qp} x_p + A_{qq} x_q + B_q u - \left(A_{qq}\hat{x}_q + A_{qp} x_p + B_q u + L_q y - L_q \hat{y}\right)$$

$$\dot{x}_q - \dot{\hat{x}}_q = A_{qp} x_p + A_{qq} x_q + B_q u - \left(A_{qq}\hat{x}_q + A_{qp} x_p + B_q u + L_q A_{pq} x_q - L_q A_{pq}\hat{x}_q\right)$$

$$\dot{x}_q - \dot{\hat{x}}_q = A_{qq} x_q - A_{qq}\hat{x}_q - L_q A_{pq} x_q + L_q A_{pq}\hat{x}_q$$

$$\dot{x}_q - \dot{\hat{x}}_q = \left(A_{qq} - L_q A_{pq}\right)\left(x_q - \hat{x}_q\right)$$

$$\dot{\varepsilon}_q = \left(A_{qq} - L_q A_{pq}\right)\varepsilon_q \qquad \text{(a19)}$$

If the error dynamic is asymptotically stable, then the observer is stable. For the error dynamic to be asymptotically stable, matrix $\left(A_{qq} - L_q A_{pq}\right)$ must be Hurwitz.

Appendix 11 Codes and programs

All of the codes and programs, including graphical user interfaces used in this work is zipped and presented to the supervisor for archive.

Appendix 12 Master’s thesis task description, 2018

See next pages

University College of Southeast Norway

Faculty of Technology, Natural Sciences and Maritime Sciences, Campus Porsgrunn

FMH606 Master's Thesis

Title: Model based estimation of fluid flow through a venturi channel: Kalman filters and observers

HSN supervisor: Assoc. Prof. Roshan Sharma

External partner: Statoil, Kelda drilling controls ASA

Task background:

This master thesis is a part of the on-going "Semi-Kidd" project at USN (which involves 4 PhD students working with the project) financed by Statoil and Norwegian Research Council.

During drilling, clean drill fluid is injected into the well being drilled using mud pump as shown in Figure 1. This drill fluid will carry away the drill cuttings from the bottom of the well and transport them back to the top through the return flow line.

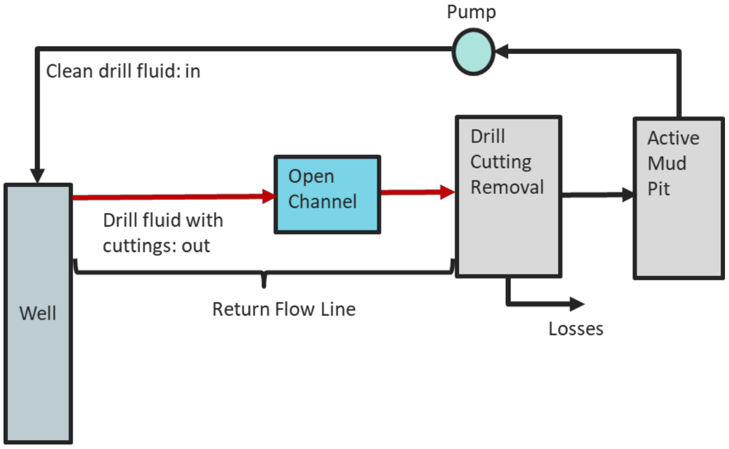

Figure 1: Schematic of fluid flow in drilling

On the top side, the drill fluid is cleaned with drill cutting removal equipment and re-injected into the well. Accurate and real time measurement of the flow of drill fluid in the return line is vital for a safe operation. In practice, Coriolis flow meters are placed on the return flow line to measure the flow rate of the return fluid. However, they are very expensive and unreliable in the presence of gases. So, in this thesis, we focus on answering the following questions:

- *Can we instead use a suitable mathematical model of the fluid flow to estimate the mud flow in the return line?*
- *Can we replace the expensive sensor with model based measurement/estimation?*

The knowledge of how much drill fluid is flowing in the return line can provide an early warning about a kick or a loss. If the flow in the return line is more than what is injected into the well, probably some fluid from the reservoir have entered into the well i.e. a kick might have occurred. Conversely, if the flow in the return line is less than what is injected into the well, probably some of the fluid from the well have leaked into the reservoir zone i.e. a loss might have occurred. Thus, flow rate of the fluid in the return line is a primary indicator of a kick or a loss. An uncontrolled kicked is known as “well blow out” and is disastrous and highly unwanted situation.

The proposal at the University College of Southeast Norway is that, a venturi channel (which is an open channel) can be placed in the return flow line to measure the flow rate of the fluid. If this is successful, then the Coriolis meters can be replaced by a simpler and a cost-effective solution. At USN, a venturi rig has been constructed to measure the flow rate of the return drill fluid.

The main work for this thesis is to use the state estimation principles to design suitable model based estimators and/or observers for estimating the flow rate through the channel. The available measurements are the level of the fluid in the channel. By using an appropriate model, observers/estimators should be designed.
The student will be provided with the necessary mathematical model of the process.

Task description:
Design and development of model based estimators/observers is the main task.

1. Give an overview of the drilling process and explain the importance of early kick/loss detection.
2. Understand and explain the Saint Venant equations for open channel flow. The student will be provided with the equations but understanding the equations is vital.
3. Understand and explain the reduced order model for open channel flow (orthogonal collocation). The model will be provided to the student as full set of ODEs (Ordinary Differential equations), but understanding the equations is vital.
4. Linearize the nonlinear ODE model and simulate it together with the nonlinear model. Discuss thoroughly your observations and results. Perform sensitivity analysis if necessary.
5. Design suitable estimators (Kalman filter or observer) using the linear model. Perform observability analysis.
6. Perform experiments at the venturi rig at USN and test the linear estimator with the rig data.
7. Design suitable nonlinear estimators (non linear versions of the Kalman filter or non linear observers) using the nonlinear ODE model of the flow through a venturi channel.
8. Perform experiments at the venturi rig at USN and test the non linear estimator with the rig data.
9. If time permits, perform a literature survey on observer design for PDE (partial differential equation) models for fluid flow through open channels.
10. Report the work in the Master’s Thesis. Present the thesis work.

Student category: Open only for students from IIA

Practical arrangements:

Perform experiments in the venturi rig at USN.

Signatures:

Student (date and signature):

Supervisor (date and signature):