University College of Southeast Norway
Faculty of Technology
–

Master's Thesis
Study programme: FMH606
Spring 2016

Thanushan Abeywickrama

Mass Flow-rate estimation in an open venturi channel using system identification

A Study of Mud flow measurements in open venturi channel

University College of Southeast Norway
Faculty of Technology
Institute of EIK
PO Box 235
NO-3603 Kongsberg, Norway

http://www.usn.no

MASTER'S THESIS, COURSE CODE FMH606

Student:	**Thanushan Abeywickrama**
Thesis title:	**Flow-rate estimation in an open venturi channel**
Signature:	Thanushan ....
Number of pages:	<54>
Keywords:	Venturi channel, coriolis meter, system identification, subspace methods, Deterministic and Stochastic system identification and Realization (DSR), Prediction Error Methods (PEM), Neural Networks (NN), Real-time estimations, State Space PEM (SSPEM), N4SID.

Supervisor:	Håkon Viumdal	Sign.: ....
2nd supervisor:	Saba Mylvaganam	Sign.: ....
Censor:	<name>	Sign.: ....
External partner:	Geir Elseth	Sign.: ....
Availability:	<Open/Secret>	

Archive approval (supervisor signature): Sign.: .... **Date :** 30.05.2016

Abstract:

Drilling operations in oil and gas industry are getting more and more advanced and complicated. One of the main reasons is that the real time monitoring and control of the processes are getting complex in their executions. As the wells constructed are getting more complex, it leads to new type of drilling methods with an increase in need of various tools and equipment. Therefore now a day's more and more research work is carried out and as a part of these research work, the flow rate estimations using level measurements in open channel venturi rigs are evaluated with mud flow applications.

This study mainly reveals the different approaches that can be used with respect to system identification, in order to achieve better flow rate estimator models. Deterministic and Stochastic system identification and Realization (DSR) method, Prediction Error Methods (PEM), N4SID with PEM method, State Space PEM (SSPEM) method and Neural Network approach were mainly discussed and evaluated in this report. The achieved models from these approaches were validated with two different sets of experimental data so that the reliability of these models were assured.

However the obtained models estimated the mass flow rate with the flow depth level measurements as the input variables. The best estimator model was obtained with N4SID algorithm together with the Prediction Error Method. This model only needs the level measurements of the flow depth in the open channel venturi. The model validation results provided the flow rate estimations with a mean percentage error of 2% and with a Root mean Square error of 8kg/min. Therefore further discussions were made with the use of this model instead of the Coriolis meter as a real-time flow rate estimator in offshore drilling industry.

Table of figures

Table of Contents

Preface

This thesis is about real-time estimations of mud flow rate in an open channel venturi with respect to level measurements based on different approaches in system identification techniques. This project is an extension of the project work carried out during the autumn 2015 named, Mud flow measurements in open venturi channel (Thanushan Abeywickrama, Jeremiah Ejimofor, Minh Hoang, Aderonke OKoro, 2015). The earlier project was concluded with obtaining a neural network model with a Mean Square Error of 114.24 kg^2/min^2. The validation results of these models were mainly compared with the newly obtained models with different system identification methods. The main focus of this thesis research is to achieve a precise measurement of drilling mud flow in order to control the bottom-hole pressure in a well, which should stay between the formation's fracture and pore pressure. All the presented results in this report were programmed with MATLAB and simulated with the same.

The thesis work is a compulsory part of System and Control master programs at Univeristy College of Southeast Norway. The research work is carried out under the close supervision of Håkon Viumdal (main-supervisor), Saba Mylvaganam (co-supervisor), Khim Chhantyal (co-supervisor), David Di Ruscio (co-supervisor) and Geir Elseth (external supervisor) from Statoil. I would like to thank all my supervisors for their generosity in helping me to get through this project and Statoil for giving this project to USN.

1 Introduction

Drilling operations in oil and gas industry are getting more and more advanced and complicated. One of the main reasons is that the real time monitoring and control of the processes are getting complex in their executions. As the wells constructed are getting more complex, it leads to new type of drilling methods with an increase in need of various tools and equipment. Therefore now a day's more and more research work is carried out on improving the drilling operations by improving the sensors and control systems. As a part of these research work, the flow rate estimations using level measurements in open channel venturi rigs are evaluated with mud flow applications. Considering the flow rate measurements in drilling operations, there are many instruments used in measuring the drilling mud flow rate. Among them the main standard devices are flow paddle meter and Coriolis meter. When considering the paddle wheel flow meters, these devices are designed to be inserted inside a pipe fitting or 'in- line'. The paddle flow meters have a lot of limitations when it comes to flow measurements in drilling mud. The main limitation is that these meters operates best with clean fluids. However the drilling mud is not a clean fluid at all. It has a lot of drill cuttings as well as other particles. Other than the paddle wheel flow meters, Coriolis flow meter is widely used in measuring the flow rate of drilling mud. The Corilis flow meters are somehow expensive and are in need of frequent replacement as of the damages occurred with the drill cuttings that comes with drilling mud flow.

These limitations drives the research work in obtaining estimators for drilling mud flow rate in offshore drilling operations. However the concept of estimation of mud flow with respect to the level measurements in an open channel venturi is not used frequently used in the industry. Therefore there are limited number of research work carried out in this scenario. Among these a mechanistic model for mudflow measurements was developed by Agu in his thesis (Agu, 2014). The research was done with respect to one dimensional Saint Venant equation for open channel flow for non-Newtonian fluids. There are many limitations in using this model as a real-time estimator. This mechanistic model needs the fluid property parameters before it predicts the flow rate. Fluid consistency index (n), fluid behavior index (K) and etc. However the results from this mechanistic model was not able to estimate the dynamics of the changes in the flow rate with the time. Therefore the mechanistic model was not taken into consideration from here onwards. This point was observed in the research carried out by Abeywickrama et al., 2015 in their Master Project. In the same project an empirical neural network model was obtained to estimate the drilling mud flow rate. This model has 3 input parameters. Two of them are ultrasonic level measurements in the open venturi channel and the last was a density measurement of the fluid, measured by a Coriolis sensor. The main drawback in this model is that it needs the density measurement which is obtained with the installed Coriolis meter. The main objective of obtaining a soft sensor model for flow rate estimations is to get rid of the expensive Coriolis equipment. However the comparison of the validation results of this neural network model is carried out in this report. Furthermore the idea of using subspace system identification methods plays a big role in developing the models for flow rate estimations in open channel venturi rigs. This aspect of obtaining a model with these subspace methods is mainly concerned in this report.

2 Problems/Objectives

The main goal of this project is to obtain a model to estimate the flow rate with focus on the ultrasonic level measurements. The report is structured with 6 main chapters.

- Experimental procedure (Chapter 3): This chapter mainly describes the experimental procedure used in obtaining the experimental data as well as the validation data.
- Methods and model formulation (Chapter 4): This chapter mainly describes the theoretical background of different subspace methods used in developing the models.
- Result and analysis (Chapter 5): This chapter mainly describes and discusses about the results of the obtained models with respect to the validation data sets. Also at the end of this chapter the best model is selected among everything.
- Conclusion (Chapter 6): This chapter summarizes the work done and the feasibility of using the obtained model as an estimator in control systems used in off shore drilling operations.

3 Experimental Procedure

The experiments were carried out with the use of the open channel venturi rig at the process hall at University College of Southeast Norway. The venturi rig is trapezoidal in its shape and consists of three main sections which are the converging section, the upstream section and the downstream section. A pump is used to pump the fluid from the reservoir. Many different sensors were installed to take different measurements. With respect to the task descriptions of this project, the main concern was to take the flow rate measurements with a higher priority towards the level measurements. Therefore the following measurements were taken into consideration.

Table 1 Measurements taken from Open channel Venturi

Measurements	**Units**
Density	kgm^{-3}
Upstream ultrasonic level measurement LT15	mm
Level measurement at the throat section LT17	mm
Downstream level measurement LT18	mm
Differential pressure reading PDT	mbar
Pump outlet pressure PT	bar
Mass flow rate	kg/min

Three different data sets were obtained with the experiments.

Model Calibration Data set – used for calibration of models

Data set 1 and Data set 2 – used for validation

These experiments are carried out in a similar manner followed in the previous project work presented by Abeywickrama et al., 2015. In both Model Calibration Data set and

Data set 1 used for model validation, the flow rate measurement was varied from 250kg/min to 550kg/min while keeping a set point flow rate for some specific time. Data set 2 was obtained by changing the set point flow rate more frequently.

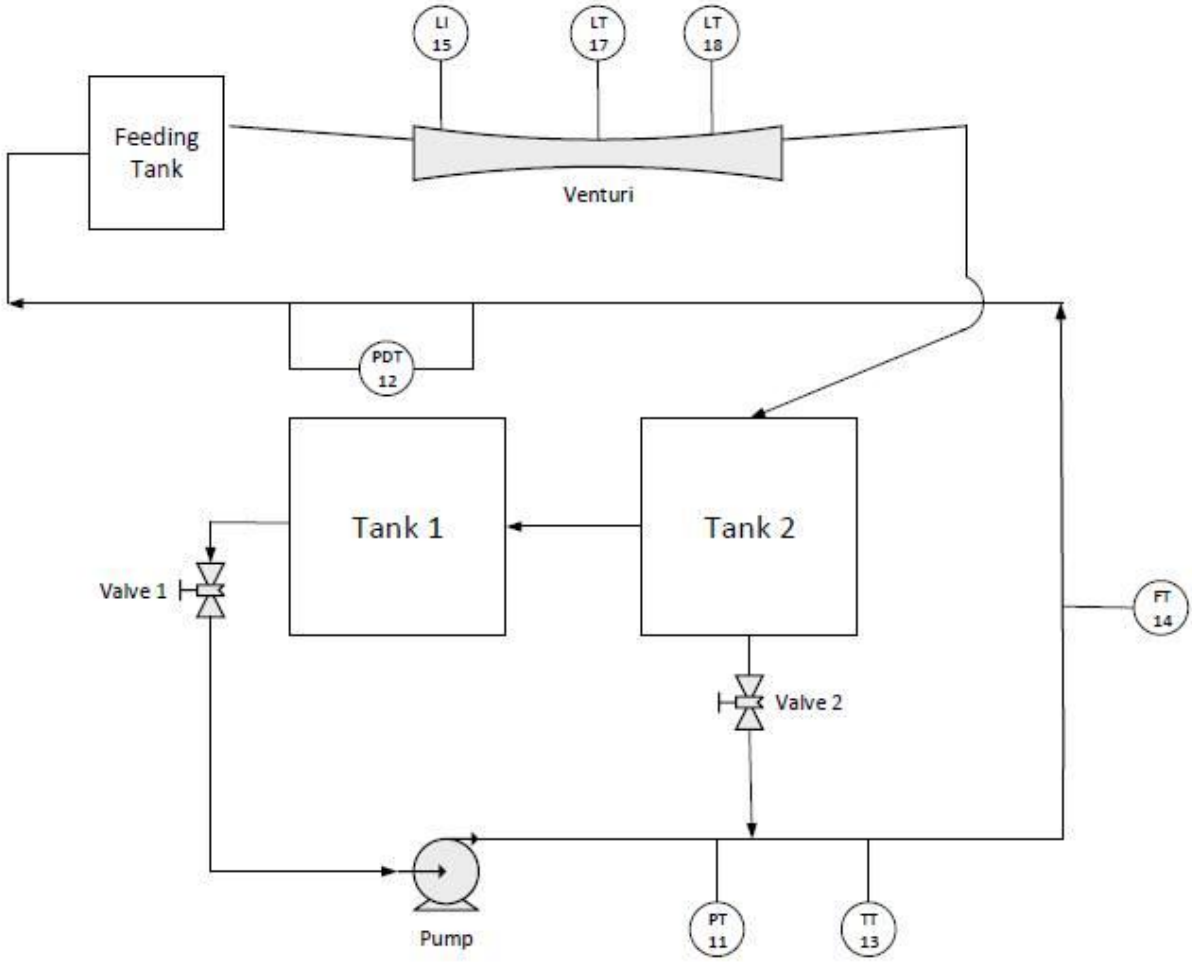

Figure 3.1 P&ID diagram of the Venturi rig

Figure 3.1 shows the Piping and Instrumentation Diagram of the Venturi rig.

LT – Ultrasonic level transmitters

FT – Coriolis meter

PT – Pressure transmitter

TT – Temperature transmitter

PDT – Differential pressure transmitter

The experiments were conducted with two fluids which are water and fluid 1. The properties and characteristics of the fluids are given in the Table 2.

Table 2 Fluid properties

	Water	Fluid 1
Density	1000 kg/m^3	1165 kg/m^3
pH	7	11.91
Characteristic	Low density & low viscosity	Low density & high viscosity
Recipe		Potassium Carbonate and Xanthan Gum mixed with water

4 Methods and model formulation

In this chapter we will briefly discuss about the methods used in the formulation of models for flow rate estimations in the open channel venturi.

4.1 Principal Component Analysis (PCA)

PCA is considered as the decomposition of the initial raw data matrix (X), into an important structure part and a noise section.

$$X = TP^T + E = Structure + Noise$$

The main objective of PCA is to represent the initial raw data which is in the form of p number of original variables, into a new coordinate space which consists of Principal Components (PCs). In this manner there's an advantage as we can drop the noisy higher order principal components. This ultimately transforms the original coordinate system to a more relevant coordinate system with principal components (Esbensen, 2010). The selection of relevant number of principal components is depended on the calculation method of PCs. In this report there are two different methods used to carry out the Principal Component Analysis, which are Singular Value Decomposition (SVD) method and the Non-linear Iterative Projections by Alternating Least Squares (NIPALS) method. The choice of appropriate number of PCs provides a comfortable way to break a data matrix into simple and meaningful pieces.

4.2 Deterministic and Stochastic system identification and Realization (DSR) method

The DSR algorithm is mainly focused on writing an Extended State Space Model (ESSM). Therefore the elimination of unknown states is carried out from the problem. The extended state space model indicates the relationship between the state space model matrices and the known data matrices. Therefore the DSR algorithm has no problems with unknown initial values and unknown states. After obtaining the state space model matrices from the extended state space model, the DSR algorithm determines the stochastic part of the model (innovations covariance matrix and the Markov parameters) from a projection of known data matrices. This estimation is done without any recursive procedure on the non-linear matrix equations like the Riccati Equation. (Ruscio, Subspace System Identification, 1995)

The DSR method is mainly based on a linear discrete time invariant state space model and the algorithm obtains the system order (n) and the matrices (A, B, D, E, CF, and F) in the following model form.

The following is a brief description about the MATLAB function for DSR which is used in estimating the flow rate in the open venturi channel.

$$[A, B, C, D, E, CF, F, x_0] = dsr(Y, U, L)$$

Y is an N by m matrix with output observations.

U is an N by r matrix with input observations.

L is an integer specifying the future horizon that is used for predicting the system order. Choice of L should be L>0 where the assumed system order satisfy $n \leq Lm$ where L = 1 is the default value. (Ruscio, An introduction to MATrix LABoratory, 2006)

$$x_{k+1} = Ax_k + Bu_k + Ce_k$$

$$y_k = Dx_k + Eu_k + e_k$$

$k \geq 0$ is the discrete time.

$x \in R^n$ is the state vector. Initial value is x_0.

$y \in R^m$ is the output of the system.

$u \in R^r$ is the system input.

$e \in R^m$ is the unknown innovations process of white noise.

A - State transition matrix.

B – External input matrix.

C – Kalman gain matrix.

D – Output matrix

E – Direct control input to output matrix.

$C = CF.F^{-1}$ is the Kalman filter gain matrix.

$E(e_t e_t^T) = FF^T$ is the innovations noise covariance matrix.

Furthermore an assumption is made as the pair (D, A) is an observable pair. More explanations regarding the formulation of the DSR algorithm can be found in (Ruscio, Subspace System Identification, 1995).

4.3 Prediction Error Methods (PEM)

Following will be an overview of prediction error methods.

Given below is the state space model on innovations form.

$$\bar{x}_{k+1} = A\ \bar{x}_k + Bu_k + Ke_k$$

$$y_k = D\ \bar{x}_k + Eu_k + e_k$$

$$\bar{y}_k = D\ \bar{x}_k + Eu_k$$

$\Delta = E(e_k e_k^T)$ is the covariance matrix.

$\bar{x}_1$ is the initial state predicted. The model is now parameterized so that the parameters in A, B, K, D, E, x_1 are organized to a vector θ which is a parameter vector. The goal is to find out the best θ vector from the known input output matrices which are Y and U.

Define the Prediction Error as

$\epsilon_k(\theta) = y_k - \bar{y}_k(\theta)$.

A better model results in a small prediction error. More explanations on prediction error methods can be found in (Ruscio, Model Predictive Control and optimization, 2001).
In MATLAB, PEM is using numerical optimization in minimizing the cost function and a weighted norm of the PE which can be considered as the following for scalars.

$$V_N(G,H) = \sum_{t=1}^{N} e^2(t)$$

$e(t)$ is the difference of measured and predicted output from the model. If we consider a linear model the error can be considered as follows.

$$e(t) = H^{-1}(q)[y(t) - G(q)u(t)]$$

Here $e(t)$ is the vector and $V_N(G,H)$ is a scalar. G and H are the respective transfer functions in rational form. N indicates the cost function is of the number of data samples. This becomes more precise for higher values of N. (MathWorks, MathWorks, 2006)

Furthermore when considering about the subspace approach in identifying the state space models with Prediction error methods it has the following steps. Estimate the k step ahead with the use of least square algorithm. Then select the state vector from the achieved results and ultimately estimate the state space matrices with the states and least squares method. (Ljung, 2009)

4.3.1 State Space Prediction Error Method (SS-PEM) Algorithm

This algorithm was developed by David Di Ruscio in 2001. (Ruscio, Model Predictive Control and optimization, 2001). This toolbox is used in appropriate with the DSR toolbox which is described earlier.

Initial parameter vector θ_1 is calculated by DSR algorithm to calculate initial state space Kalman Filter model which provides (A, B, D, E, K, x1) matrices as indicated by the model in section 4.3. Then the model is converted to observability canonical form by DSR function ss2cf.m.

The minimizing parameter vector is computed with the function fminunc.m.

[A, B, D, E, K, x1] = sspem(Y, U, n)

The system order n should be known before running this algorithm. The solution for this is to run the DSR algorithm and estimate the system order by that. The identified model is then represented as an observability canonical form. Further details on this algorithm can be found in (Ruscio, Model Predictive Control and optimization, 2001).

4.3.2 Discrete-time state-space model using N4SID

N4SID is a numerical algorithm used for subspace state space system identification. This is highly useful for higher order multivariable systems. It is mentioned that with N4SID algorithm most of the a-priori parameterization problems can be overcome. As in SS-PEM algorithm the order of the system is essential and this can be achieved by DSR algorithm initially. N4SID algorithm is non-iterative, with no involvement in non-linear optimization. Usually typical iterative algorithms suffer from no guaranteed convergence, sensitivity to initial estimates and local minima of the objective. Furthermore when using N4SID algorithms there is no variation between zero and non-zero initial states (Overschee, 1992).

System Identification toolbox in MATLAB provides the N4SID algorithm which estimates an n order state space model using the given observed input and output data. (MathWorks, n4sid, 2016)

SYS = n4sid (data, n)

SYS is a state space model with identifiable parameters (idss model), and it is a discrete time model with no specific sample time and element of state disturbance. (MathWorks, idss, 2016)

The following model is represented by SYS.

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

$$y(t) = Cx(t) + Du(t) + e(t)$$

A, B, C, D are state space matrices.

K is the disturbance matrix.

u(t) is the input observations and y(t) is the output observations.

x(t) is the state vector of n states and e(t) is disturbance. (MathWorks, n4sid, 2016)

5 Results and analysis

This chapter will mainly discuss about the obtained results from different system identification approaches.

5.1 Principle Component Analysis (PCA)

First of all a Principal Component Analysis was carried out with respect to the Model Calibration Data set to find out the most important variables with respect to the Mass Flow, which is considered as the output variable in all the models from here onwards. The other variables were taken as input variables, which are level measurements (LT15, LT17, and LT18), differential pressure measurement (PDT) and outlet pressure of the pump (PT). The PCA showed the importance of the input variables to the output variable, with respect to each principal component. The PCA was done with two different methods which are Singular Value Decomposition (SVD) technique and the NIPALS technique.

5.1.1 Singular Value Decomposition (SVD) calculation results

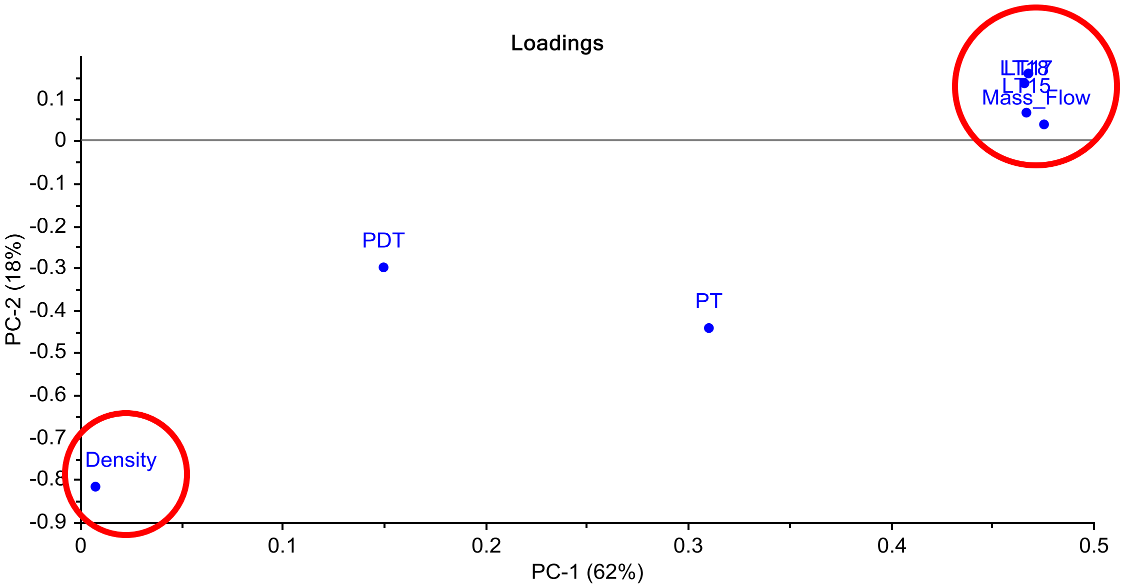

Figure 5.1 PCA variable loadings obtained with SVD

A MATLAB program was used to obtain the singular values of the data set. The results obtained with are shown as following. The Model Calibration Data set was centered and scaled before the analysis.

Singular Values =

141.8312

77.0870

65.5450

46.1205

16.3910

12.7009

8.0784

With respect to the above obtained singular values the number of components used for PCA is only four. The reason why is that it seems like the singular values after 46.1205 do not express a higher quantitative value with respect to the first four. Therefore the following Loading Vector was obtained.

Loading vector with density =

	PC1	PC2	PC3	PC4
Mass Flow	0.4774	-0.0273	0.0009	-0.1532
LT15	0.4664	-0.0798	0.1165	-0.1899
LT17	0.4704	-0.1587	0.0370	-0.1009
LT18	0.4685	-0.1239	-0.0538	-0.0968
PDT	0.1298	0.3531	-0.9188	0.0190
PT	0.3114	0.4360	0.2304	0.8120
Density	0.0017	0.7985	0.2912	-0.5110

According to Figure 5.1 and the loading vector values obtained by the MATLAB program, with respect to the most important principal component which is PC1 we can clearly see that LT15 LT17 and LT18 are highly correlated with the Mass Flow. According to our data sets, Density of the fluid is not related to Mass Flow at all. Therefore from here onwards these three level measurements are considered as inputs and for Mass Flow estimation model formations. This fact is further proved with NIPLAS algorithm calculations in the following Figure 5.2.

5.1.2 NIPALS calculation results

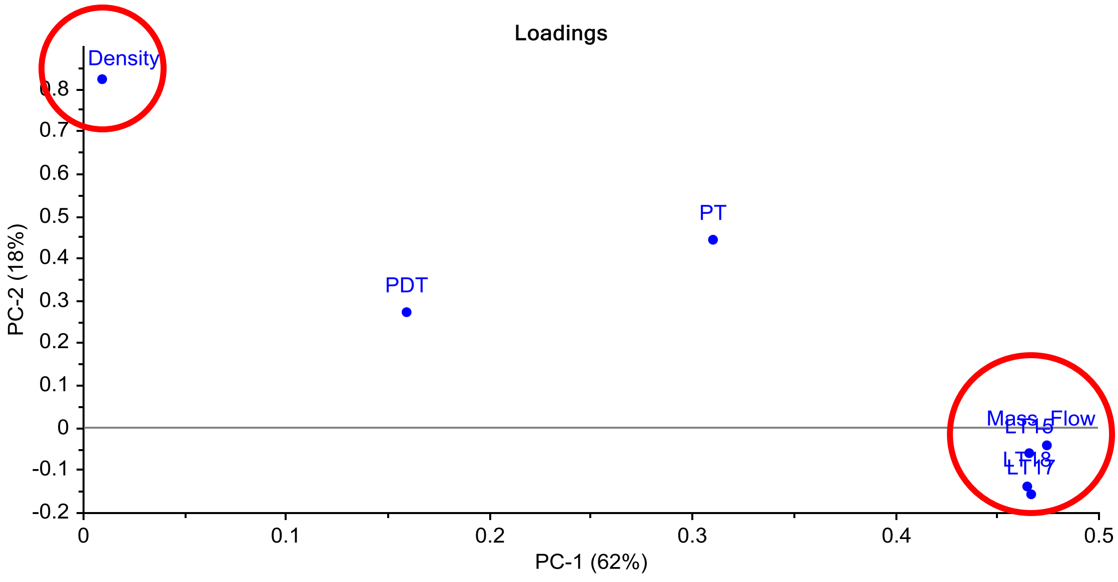

Figure 5.2 PCA variable loadings obtained with NIPALS

The same Model Calibration Data set was fed to Unscrambler software to carry out a PCA with NIPALS algorithm. The same loading values were obtained as it was in SVD. Therefore the results from the SVD calculations are validated with the NIPALS calculations.

LT15, LT17 and LT18 level measurements were the selected as input variables for obtaining the models further.

5.1.3 Noise reduction of the Input Data

This was carried out with the use of the singular values and loading vector values obtained with SVD. New set of singular values are obtained only with the selected inputs LT15, LT17 and LT18. The data set was only centered. The scaling was not done as all the inputs are level measurements and all the measurements are in millimeters.

Singular Values =

1.0e+03 *

1.5072
0.2298
0.1624

Looking at the above set of singular values we can clearly see that only one singular value which relates to the 1st PC is enough to obtain the new principal component vector space. Therefore the following loading vector is obtained with respect to the first principal component.

Loading vector =

0.5967

0.5987

0.5344

With these calculations the new input signals (noise reduced) were obtained with respect to the 1st PC. The effect of noise reduction can be clearly seen in the following Figure 5.3.

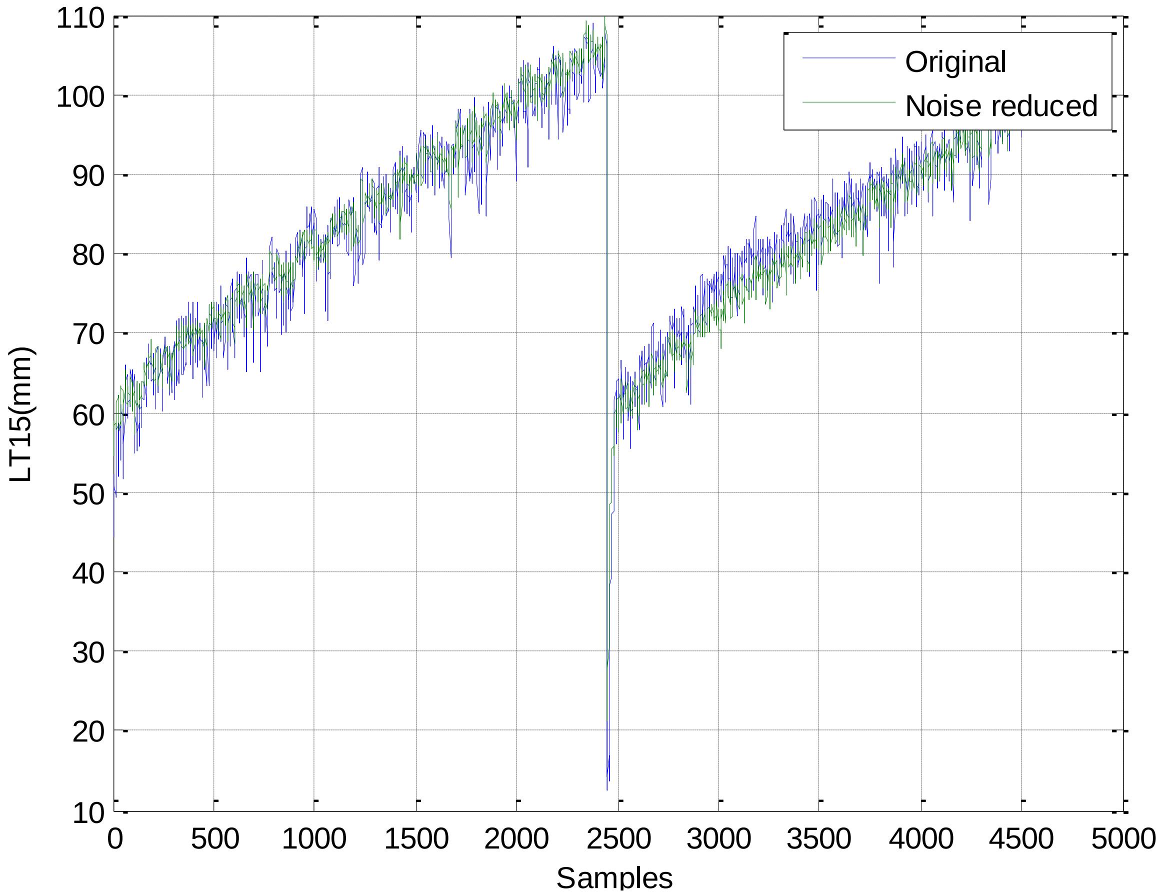

Figure 5.3 Comparison of Original and Noise reduced input

From here onwards the following input and output data matrices will be used to obtain the models.

Input U = [LT15, LT17, LT18]

Output Y = [Mass Flow]

5.2 DSR Model

5.2.1 Model obtained with the Model Calibration dataset

5.2.1.1 Without Noise reduction of the Inputs

The following DSR model was obtained with the raw data from the Model Calibration dataset. The inputs were not treated with singular value decomposition method to reduce the noise.

The obtained model was a first order one with L =1 (where L is the number of block rows). The error values of the above estimation is shown as follows.

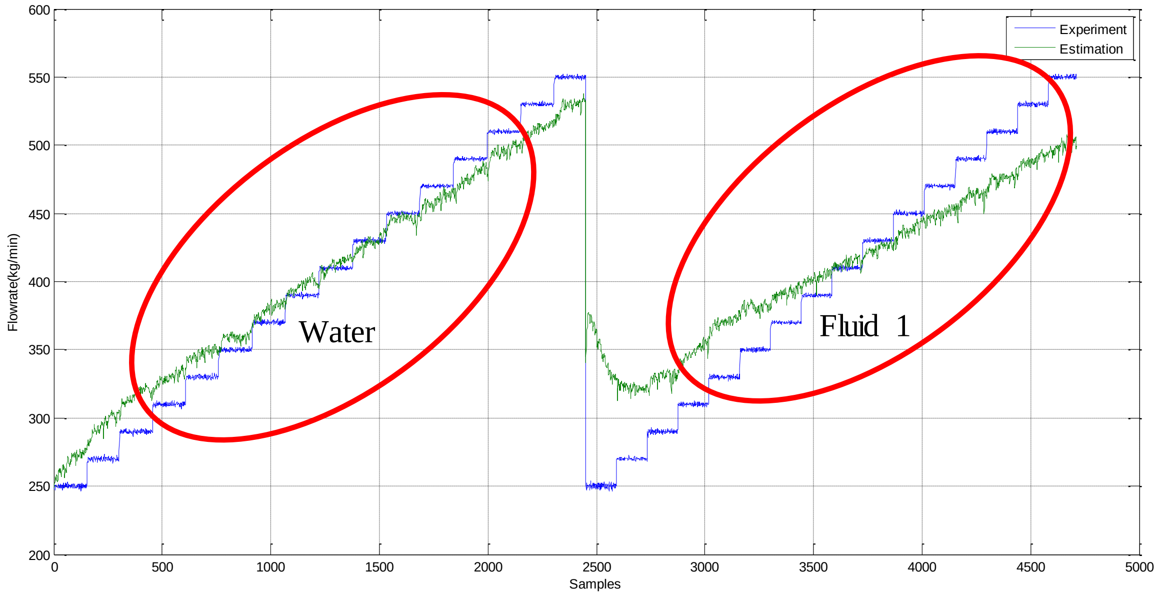

Figure 5.4 Estimation results from DSR Model (Raw data)

Figure 5.4 illustrates the time response of the following discrete time linear system.

$$x_{t+1} = Ax_t + Bu_t$$

$$y_t = Dx_t + Eu_t$$

A = 0.9905, B = [-0.0450 0.0346 -0.0245], D = -1, E = [0.6587 1.1774 0.3302]

Mean Error Percentage = 6.6315 %. RMSE = 30.4191 (kg/min)

With respect to Figure 5.4 we can clearly see that the calibrated model estimation is having a trend line approach compared to the experimental results. The original dynamics of the Mass flow rate are not well tracked by the model. However the model was validated with the two new data sets as follows.

5.2.1.1.1 Validation of Model with Data set 1

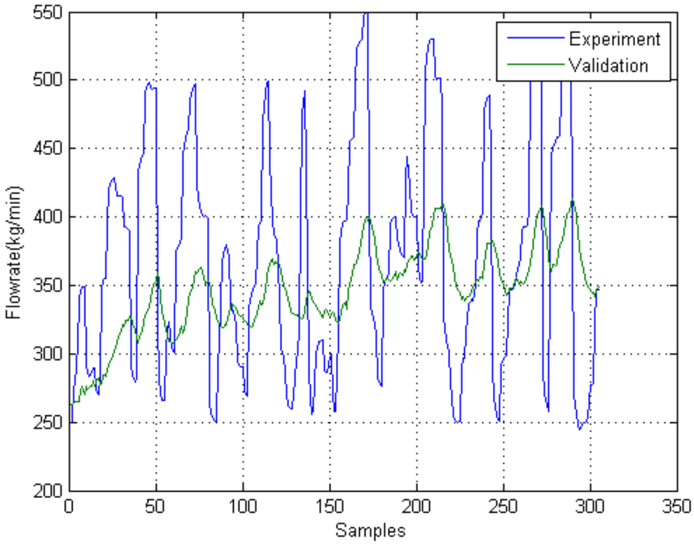

Figure 5.5 Validation results from DSR Model (Raw data) with Data set 1

Mean error Percentage = 18.6130%. RMSE = 83.2397 (kg/min)

5.2.1.1.2 Validation of Model with Data set 2

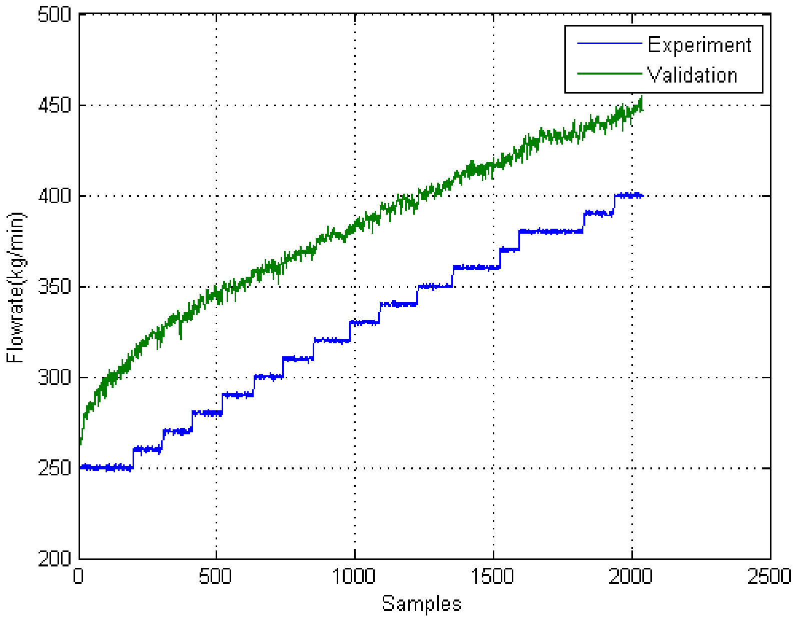

Figure 5.6 Validation results from DSR Model (Raw data) with Data set 2

Mean error Percentage = 17.0416%. RMSE = 54.6456 (kg/min)

With respect to the validation results obtained, Figure 5.5 and Figure 5.6 shows that the DSR model obtained with raw input data doesn't provide a good model. Therefore the noise reduction of the input variables were carried out with singular value decomposition technique.

5.2.1.2 With Noise reduction of the Inputs

The following DSR model was obtained with the singular value decomposed input variables from the Model Calibration dataset. The inputs were treated with singular value decomposition with respect to the 1st principal component to reduce the noise.

The obtained model was a first order one with L =1 (where L is the number of block rows). The error values of the above estimation is shown as follows.

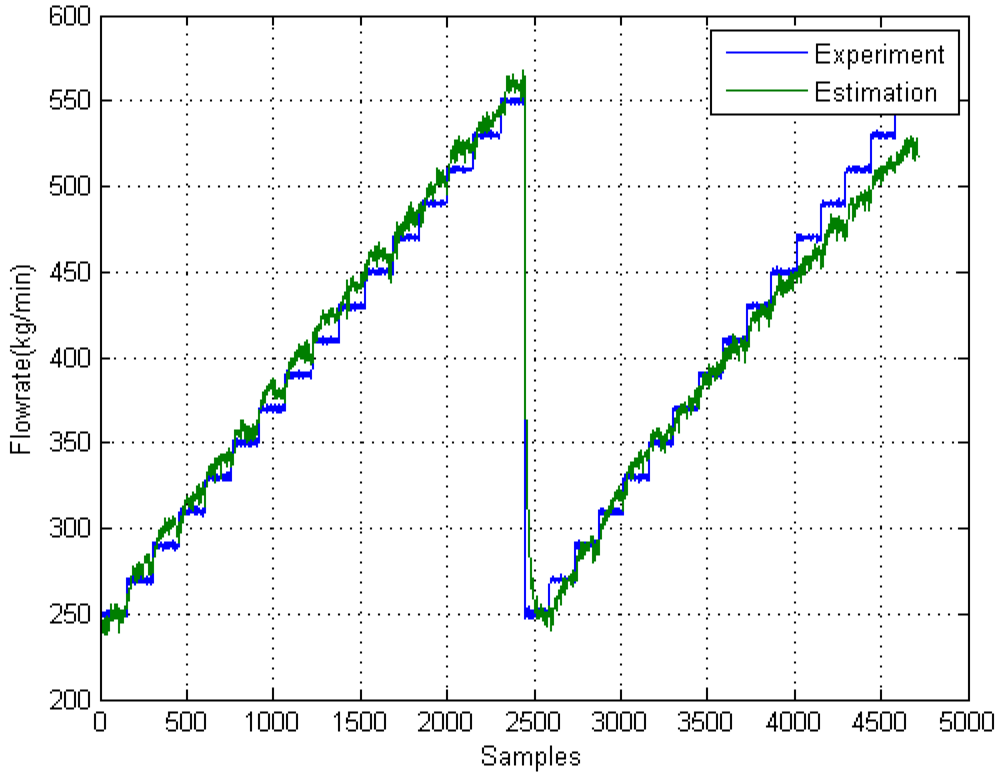

Figure 5.7 Estimation results from DSR Model (SVD noise reduces data)

Figure 5.7 illustrates the time response of the following discrete time linear system.

$$x_{t+1} = Ax_t + Bu_t$$

$$y_t = Dx_t + Eu_t$$

A = 0.9684, B = [1.0e+12 * -2.8098 1.9969 0.8998], D = -1

E = [1.0e+12 * -2.5239 -2.0408 5.1045]

Mean error percentage = 2.4940%. RMSE = 13.1398 (kg/min)

With respect to Figure 5.7 we can clearly see that the calibrated model estimation is have less a trend line approach compared to the 1st DSR model obtained earlier. The original dynamics of the Mass flow rate are somewhat well tracked by the model. However this model was also validated with the two new data sets as follows.

5.2.1.2.1 Validation of Model with Data set 1

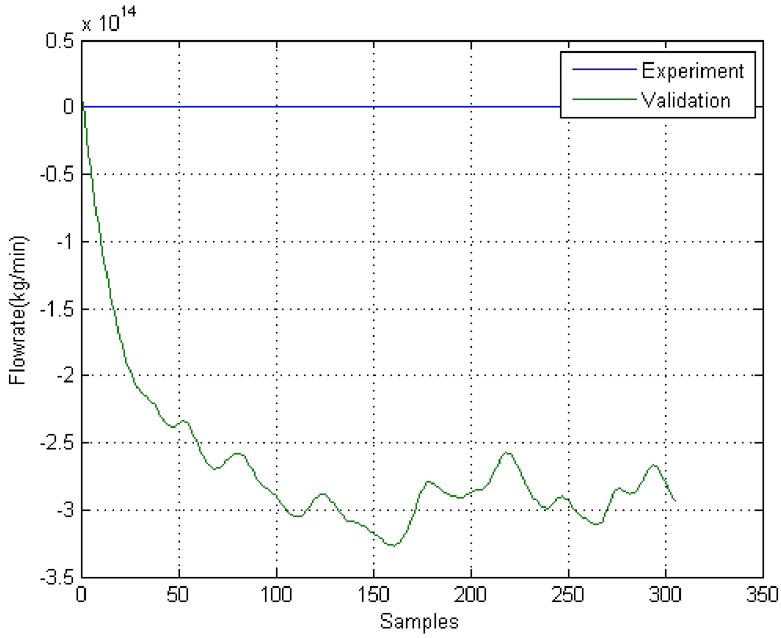

Figure 5.8 Validation results from DSR Model (SVD noise reduced data) with Data set 1

Mean error percentage = 7.6032e+13%. RMSE = 2.7366e+14 (kg/min)

5.2.1.2.2 Validation of Model with Data set 2

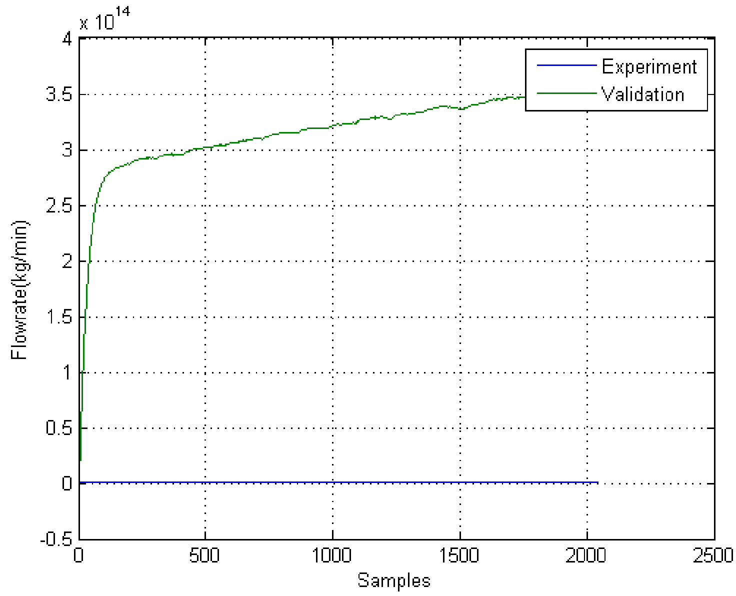

Figure 5.9 Validation results from DSR Model (SVD noise reduced data) with Data set 2

Mean error percentage = 9.7714e+13 %. RMSE = 3.1814e+14 (kg/min)

With respect to the validation results obtained, Figure 5.8 and Figure 5.9 shows that the DSR model obtained with noise reduced inputs doesn't provide a good model (concerning the error values).

5.3 SSPEM Model

5.3.1 Model obtained with the Model Calibration dataset

5.3.1.1 Without Noise reduction of the Inputs

The following SSPEM model was obtained with the raw data from the Model Calibration dataset. The inputs were not treated with singular value decomposition method to reduce the noise. This is a second order discrete time state space model. The error values of the model estimation is shown as follows.

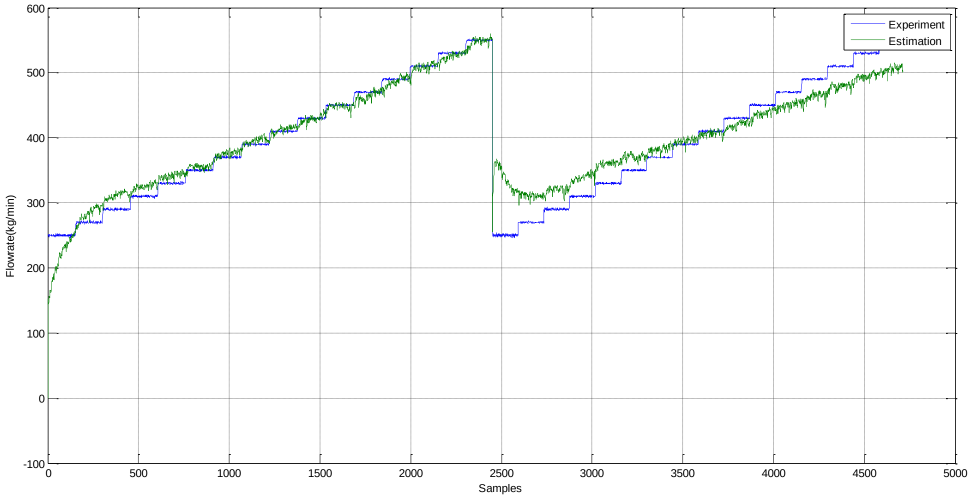

Figure 5.10 Estimation results from SSPEM Model (Raw data)

Figure 5.10 illustrates the time response of the following discrete time linear system.

$$x_{t+1} = Ax_t + Bu_t$$

$$y_t = Dx_t + Eu_t$$

A = [0 1.0000; 0.1075 0.8802]

B = [-0.1551 -0.2242 -0.1593; 0.0531 -0.0286 0.0785]

D = [1 0] , E = [1.0070 1.7299 0.6920]

Mean_error_percentage = 5.3968%. RMSE = 26.4535 (kg/min)

With respect to Figure 5.10 we can clearly see that the calibrated model estimation is having a minor trend line approach compared to the experimental results. The original dynamics of the Mass flow rate are not that well tracked by the model. The obtained SSPEM model was validated with the two new data sets as follows.

5.3.1.1.1 Validation of Model with Data set 1

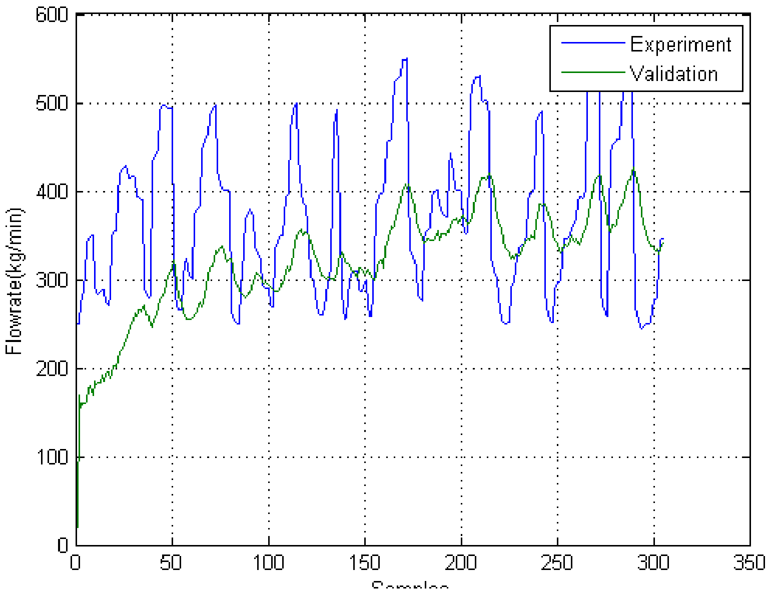

Figure 5.11 Validation results from SSPEM Model (Raw data) with Data set1

Mean error percentage = 21.7515%. RMSE = 99.4608 (kg/min)

5.3.1.1.2 Validation of Model with Data set 2

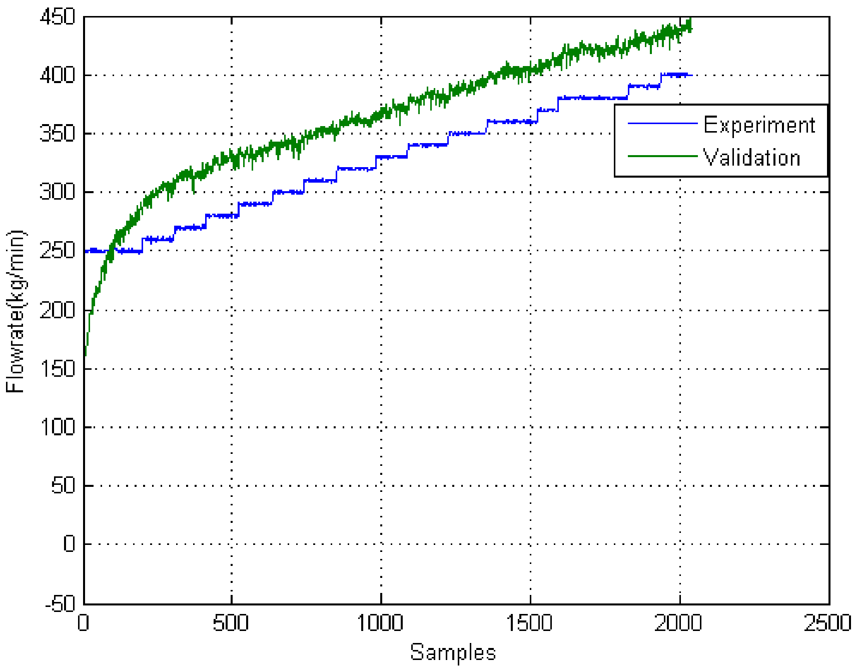

Figure 5.12 Validation results from SSPEM Model (Raw data) with Data set2

Mean error percentage = 12.4483%. RMSE = 41.0635 (kg/min)

With respect to the validation results obtained, Figure 5.11 and Figure 5.12 shows that the SSPEM model obtained with raw data inputs doesn't provide a good model (concerning the error values).

5.3.1.2 With Noise reduction of the Inputs

The following SSPEM model was obtained with the singular value decomposed input variables from the Model Calibration dataset. The inputs were treated with singular value decomposition with respect to the 1st principal component to reduce the noise. This is also a third order discrete time state space model. The error values of the model estimation is shown as follows.

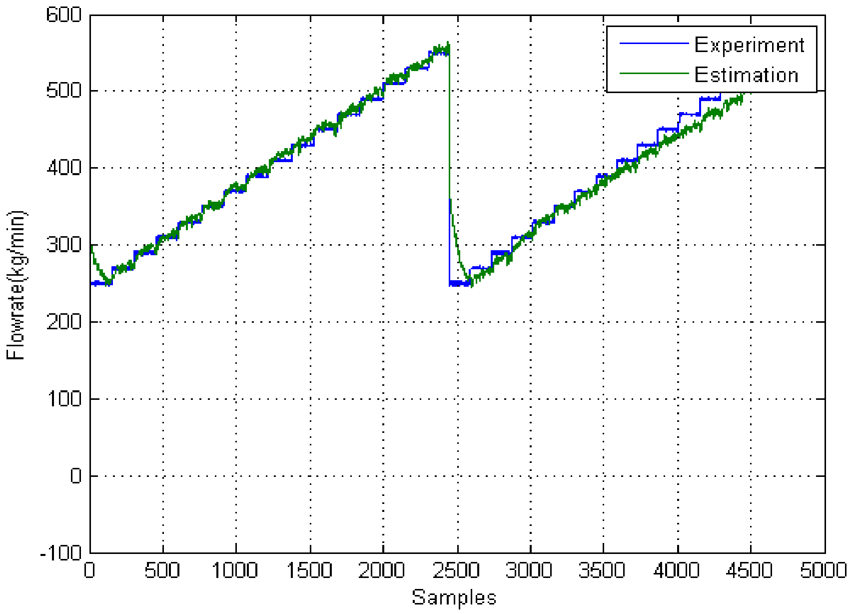

Figure 5.13 Estimation results from SSPEM Model (SVD noise reduces data)

Figure 5.13 illustrates the time response of the following discrete time linear system.

$$x_{t+1} = Ax_t + Bu_t$$

$$y_t = Dx_t + Eu_t$$

A = [0   1.0000   0;   0   0   1.0000;   -0.3523   0.2247   1.1186]

B = [1.0e+11 *   -0.9703   0.7328   0.2624;   -0.0265   0.0200   0.0072;   1.0636   -0.8032   -0.2876]

D = [1   0   0]

E = [0.0000   2.1120   0.0000]

Mean error percentage =   2.8508%. RMSE =   16.4539 (kg/min)

With respect to Figure 5.13 we can see that the calibrated model estimation is have a lesser trend line approach compared to the 1st SSPEM model obtained earlier. The original dynamics of the Mass flow rate are somewhat well tracked by the model. However this model was also validated with the two new data sets as follows.

5.3.1.2.1 Validation of Model with data set 1

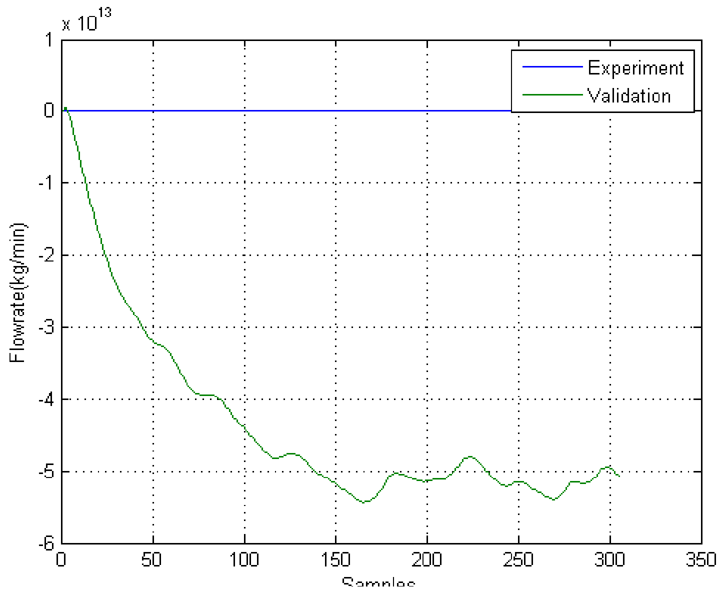

Figure 5.14 Validation results from SSPEM Model (SVD noise reduced data) with Data set 1

Mean error percentage = 1.2285e+13%. RMSE = 4.5192e+13 (kg/min)

5.3.1.2.2 Validation of Model with data set 2

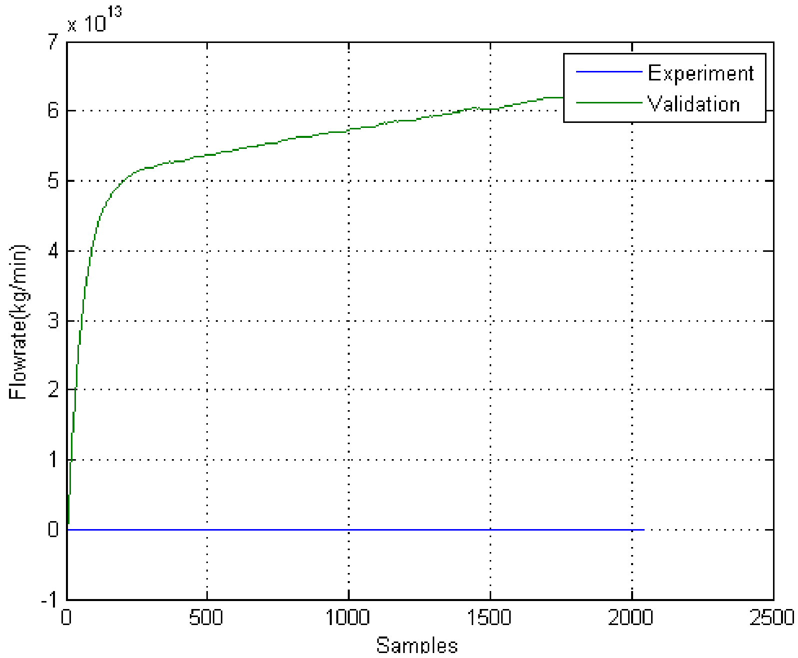

Figure 5.15 Validation results from SSPEM Model (SVD noise reduced data) with Data set 2

Mean error percentage = 1.7174e+13%. RMSE = 5.6342e+13 (kg/min)

With respect to the validation results obtained, Figure 5.14 and Figure 5.15 shows that the SSPEM model obtained with noise reduced inputs doesn't provide a good model (with concern to the error values).

5.4 Model with Discrete-time state-space model using N4SID and PEM

5.4.1 Model obtained with the calibration dataset

5.4.1.1 Without Noise reduction of the Inputs

The following model was obtained with the use of N4SID algorithm and PEM in MATLAB system identification toolbox, with the raw data from the Model Calibration dataset. The inputs were not treated with singular value decomposition method to reduce the noise. This is a second order discrete time state space model.

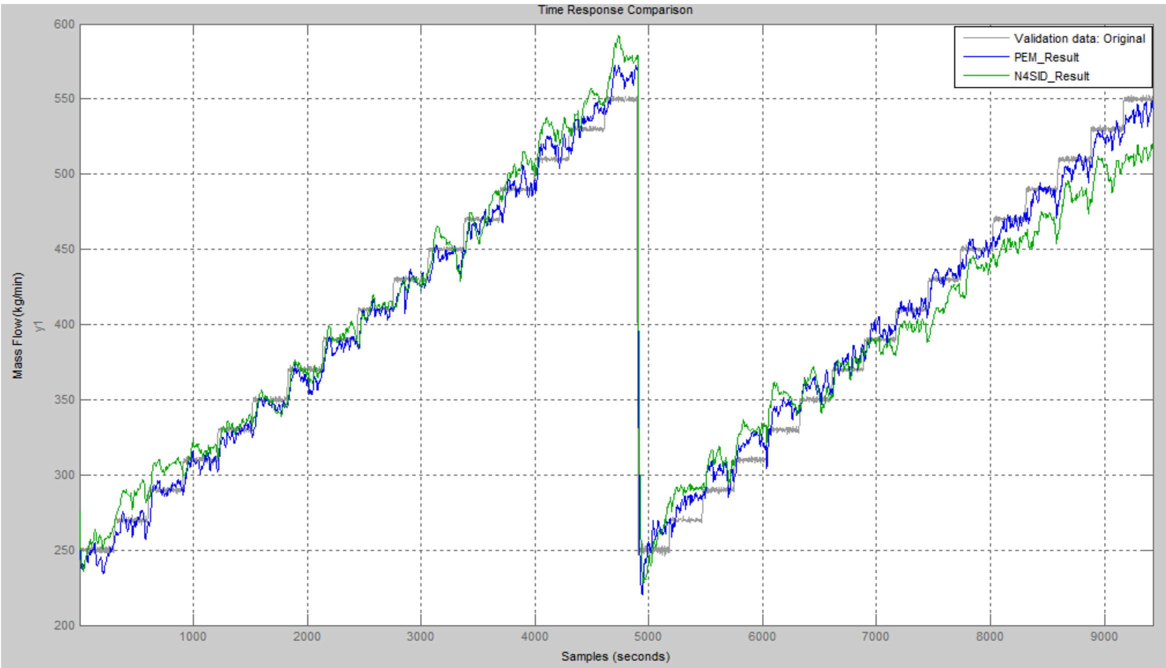

Figure 5.16 Estimation results and response comparison from N4SID & PEM Models (Raw data)

Figure 5.16 illustrates the time response comparison of three different scenarios. Which are the original experimental data, PEM results and N4SID results.

The time response of the following discrete time linear system is simulated with the obtained A, B, D and E matrices from the N4SID algorithm.

$$x_{t+1} = Ax_t + Bu_t$$

$$y_t = Dx_t + Eu_t$$

A = [0.9542 0.0111; 0.0027 0.9577]

B = 0.0000 0.0004 -0.0000; 0.0002 -0.0015 0.0009]

D = [1.0e+03 * 1.6648 0.1180] , E = [0 0 0]

Figure 5.17 was obtained as the time response simulation of the second order discrete time state space model with the above system's state space matrices (A, B, D & E). The error values of the model estimation is shown as follows.

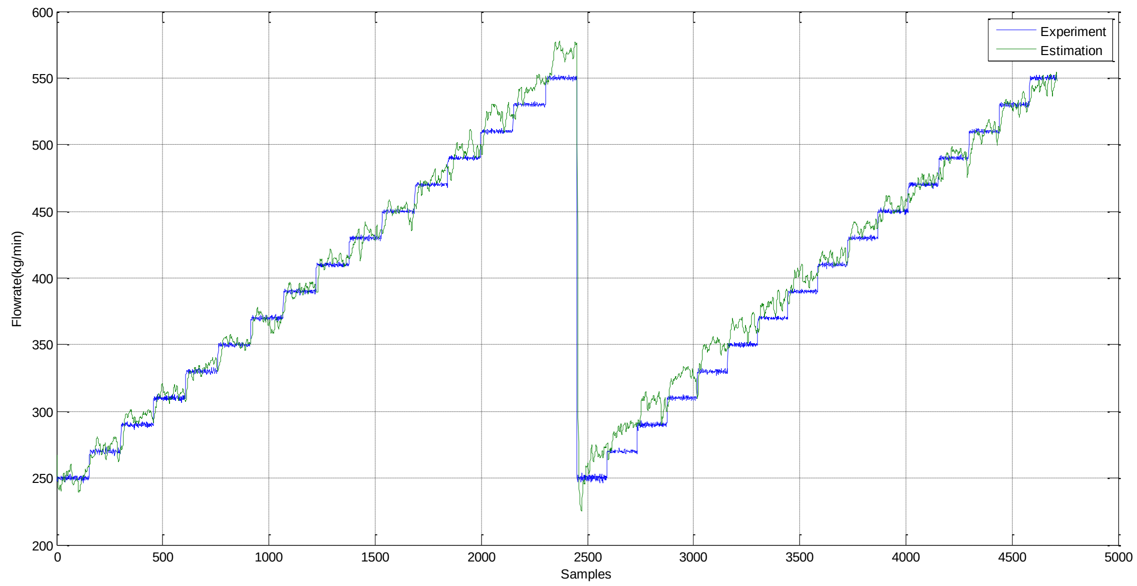

Figure 5.17 Estimated simulation results from N4SID Model (Raw Data)

Mean error percentage = 2.3464 %. RMSE = 12.7317 (kg/min)

With respect to Figure 5.17 we can see that the calibrated model estimation is having the least trend line approach compared to the all the models obtained up to now. The original dynamics of the Mass flow rate are greatly tracked by the model estimations. Let's validate this model with the two new data sets.

5.4.1.1.1 Validation of Model with data set 1

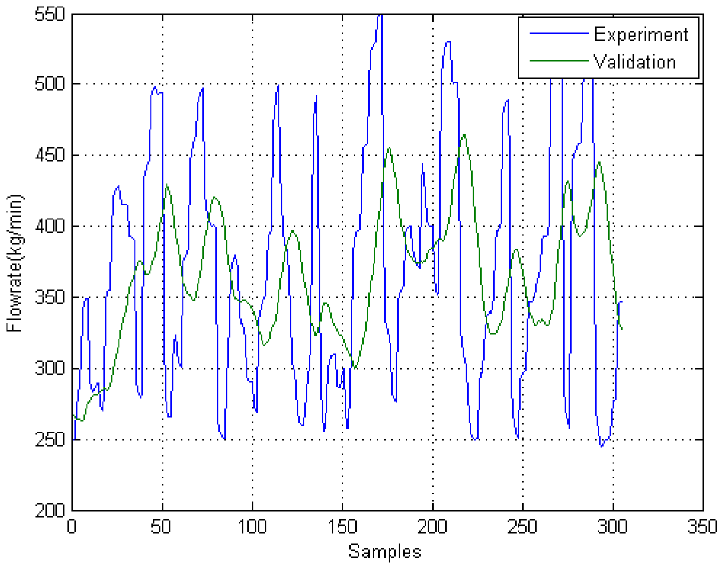

Figure 5.18 Validation results from N4SIDModel (Raw data) with Data set 1

Mean error percentage = 22.7785%. RMSE = 95.5065 (kg/min)

5.4.1.1.2 Validation of Model with data set 2

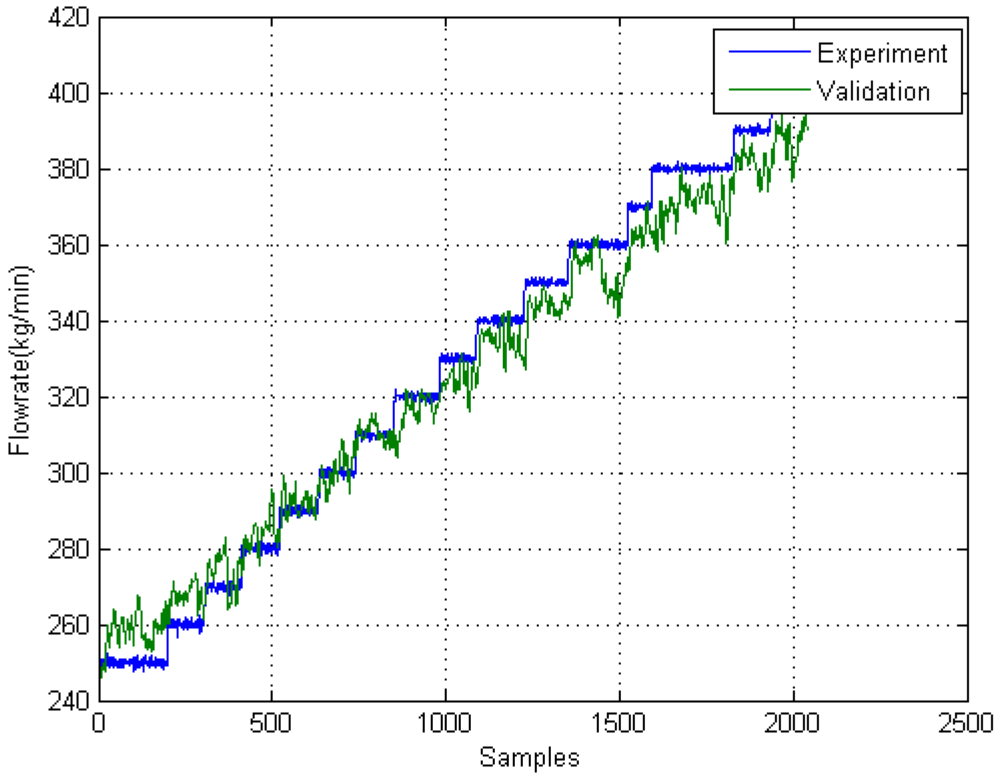

Figure 5.19 Validation results from N4SIDModel (Raw data) with Data set 2

Mean error percentage = 2.0472%. RMSE = 8.0770 (kg/min)

With respect to the validation results obtained, Figure 5.18 and Figure 5.19 shows that the N4SID model obtained with raw data inputs provides a fairly good model (concerning the error values). The mean error percentage of the model validation with respect to Data set 2 is almost 2% and the root mean square error value (RMSE) is around 8.077 kg/min. When considering the dynamics of the model validation the N4SID model proves better model dynamics. This is vivid when looking at Figure 5.19. This model can be taken into consideration for further analysis. However the model validation didn't work well with the Data set 1. The reasons can be that the variations of the set point, which is the Mass Flow, is too frequently changed and the height levels measured with the ultrasonic sensors might not be in the equilibrium state. Therefore there can be a small problem with the experimental Data set 1.

5.4.1.2 With Noise reduction of the Inputs

The following model was obtained with the use of N4SID algorithm and PEM in MATLAB system identification toolbox, with the singular value decomposed input variables from the Model Calibration dataset. The inputs were treated with singular value decomposition with respect to the 1st principal component to reduce the noise. This is a second order discrete time state space model.

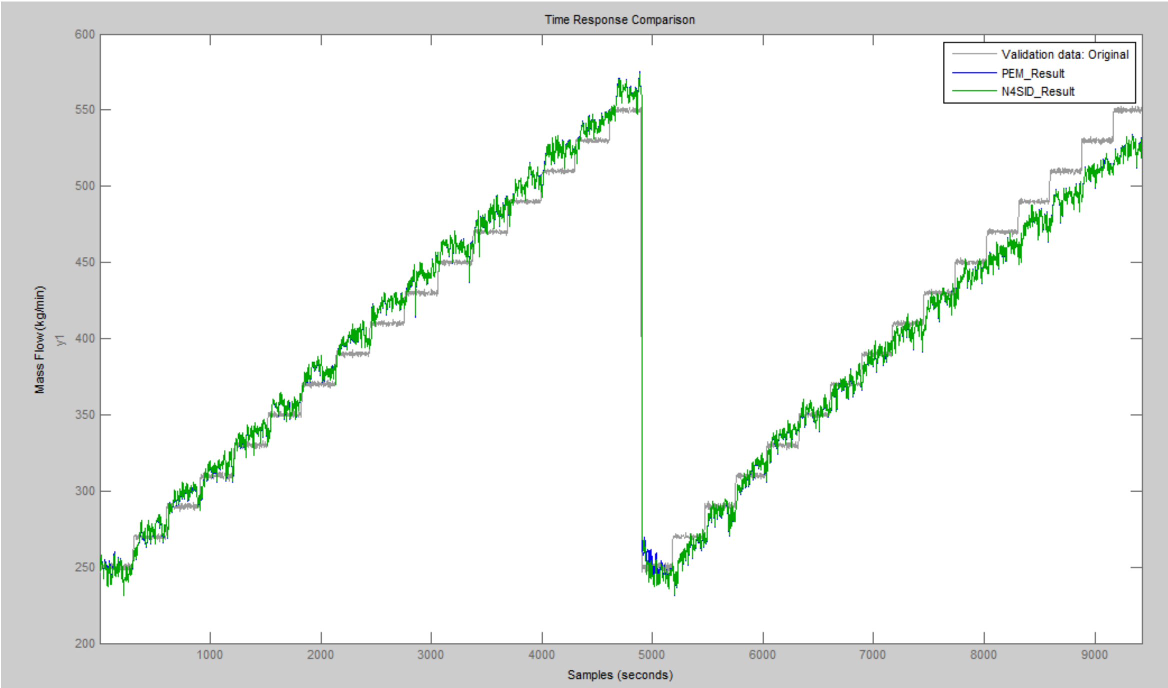

Figure 5.20 Estimation results and response comparison from N4SID & PEM Models (SVD noise reduced data)

Figure 5.20 illustrates the time response comparison of three different scenarios. Which are the original experimental data, PEM results and N4SID results.

The time response of the following discrete time linear system is simulated with the obtained A, B, D and E matrices from the N4SID algorithm.

$$x_{t+1} = Ax_t + Bu_t$$

$$y_t = Dx_t + Eu_t$$

A = [0.9371 0.1501; 0.1427 0.2123]

B = [0.0167 0.0878 -0.1835; -0.0835 -0.4406 0.9213]

D = [1.0e+03 * 1.2053 0.2491] , E = [0 0 0]

Figure 5.17 was obtained as the time response simulation of the second order discrete time state space model with the above system's state space matrices (A, B, D & E). The error values of the model estimation is shown as follows.

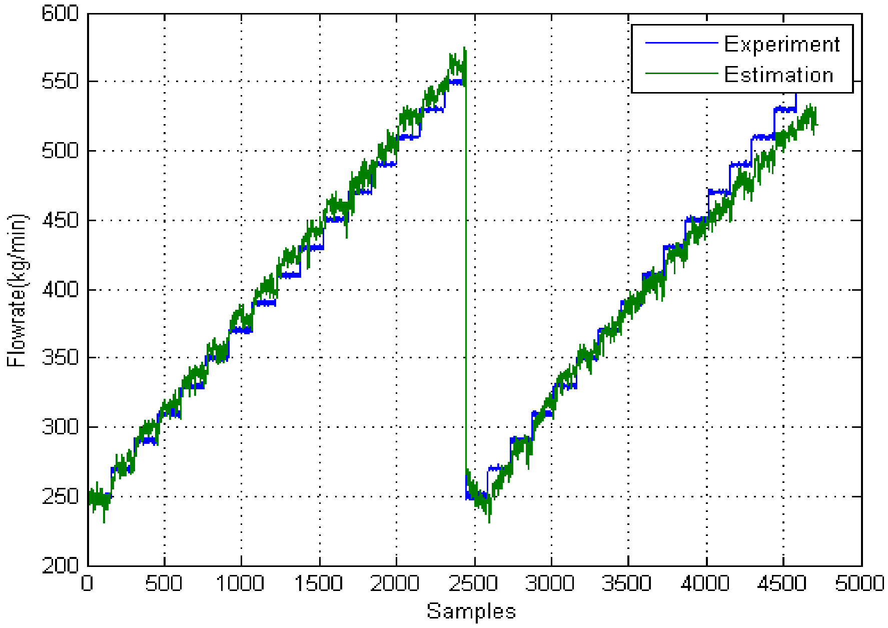

Figure 5.21 Estimated simulation results from N4SID Model (SVD noise reduced data)

Mean error percentage = 2.4171%. RMSE = 12.8742 (kg/min)

With respect to Figure 5.21 we can see that the calibrated model estimation is having a similar approach compared to the N4SID model results obtained with the raw input data. The original dynamics of the Mass flow rate are also well tracked by the model estimations. Let's validate this model with the two new data sets.

5.4.1.2.1 Validation of Model with data set 1

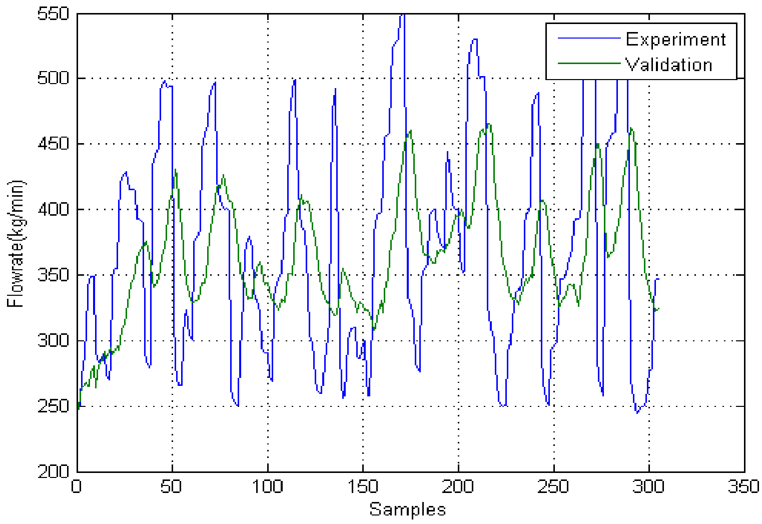

Figure 5.22 Validation results from N4SIDModel (SVD noise reduced data) with Data set 1

Mean error percentage = 20.5631%. RMSE = 86.5930 (kg/min)

5.4.1.2.2 Validation of Model with data set 2

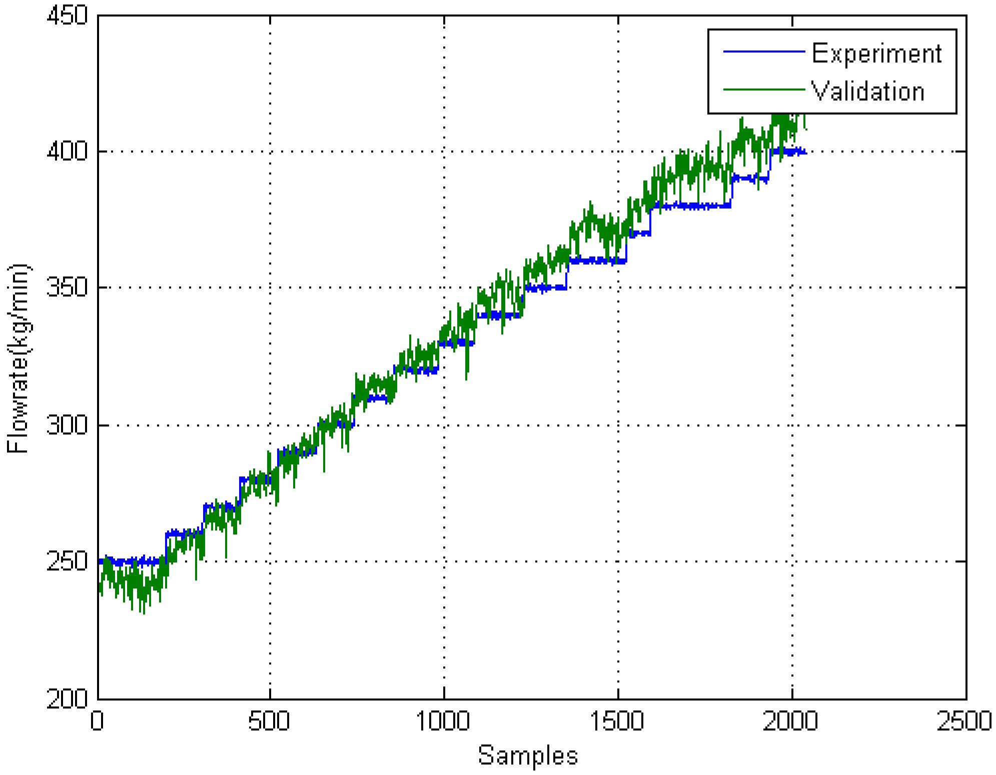

Figure 5.23 Validation results from N4SIDModel (SVD noise reduced data) with Data set 2

Mean error percentage = 2.3047%. RMSE = 9.2318 (kg/min)

With respect to the validation results obtained, Figure 5.22 and Figure 5.23 shows that the N4SID model obtained with noise reduced inputs also provides a fairly good model just like the model obtained by N4SID earlier with the raw data inputs. The mean error percentage of the model validation with respect to Data set 2 is almost 2.3% which is a bit higher that the earlier N4SID model and the root mean square error value (RMSE) is around 9.23 kg/min which is also a bit higher. When considering the dynamics of the model validation, this N4SID model is not that well tracking the dynamics compared to the previous N4SID model. This is vivid when looking at Figure 5.19. However this model can also be taken into consideration for further analysis.

However this model validation also didn't work well with the Data set 1. The same reasons can be mentioned as it was for the 1st N4SID model.

5.5 Neural Network (NN) Model

The following NN model was extracted from the previous work carried out by (Thanushan Abeywickrama, Jeremiah Ejimofor, Minh Hoang, Aderonke OKoro, 2015). The inputs of the NN model are [Density LT15 LT17] and the output is the Flow rate. The number of neurons in the hidden layer of the NN model was 13. The test performance error of the obtained best NN model is 114.24 which is the mean square error Abeywickrama et al., 2015. The RMSE value is 10.61 kg/min. We will now validate this model also with the respect to the two new data sets and compare the results with the obtained subspace models.

5.5.1 NN Model obtained with the Model Calibration dataset

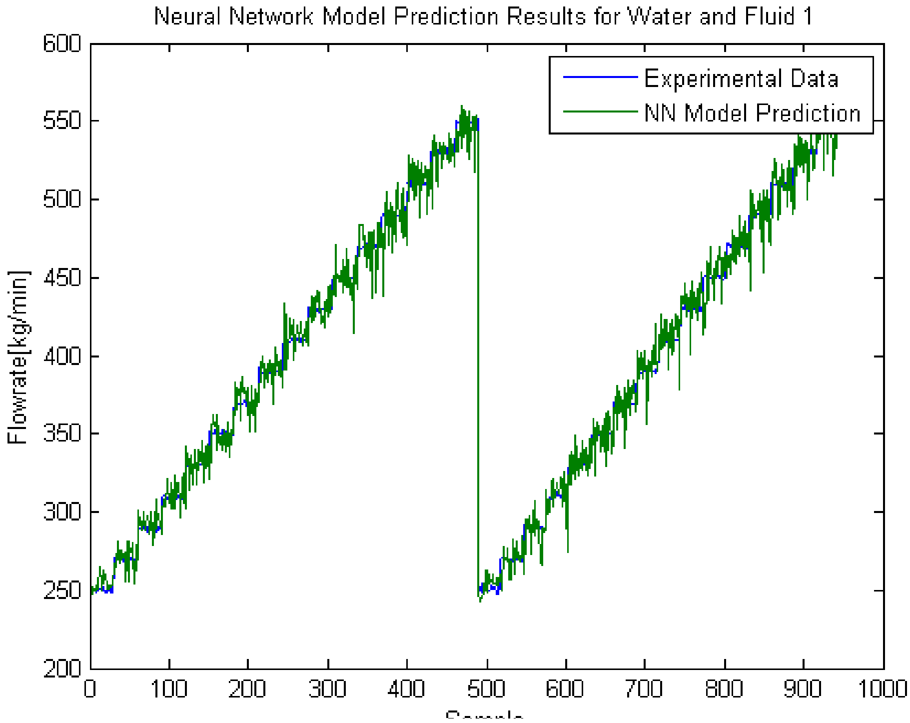

Figure 5.24 NN model prediction results for Water and Fluid 1(Model 1). Number of input variables = 3. Number of hidden layer neurons = 13.

Mean error percentage = 2.0420%. RMSE = 10.6112 (kg/min)

According to (Thanushan Abeywickrama, Jeremiah Ejimofor, Minh Hoang, Aderonke OKoro, 2015) the NN model proves with better estimations for dynamic changes in the Mass flow rate. This is clear with reference to Figure 5.24. The mean error percentage of the calibrated model is around 2%.

5.5.1.1Validation of NN Model with data set 1

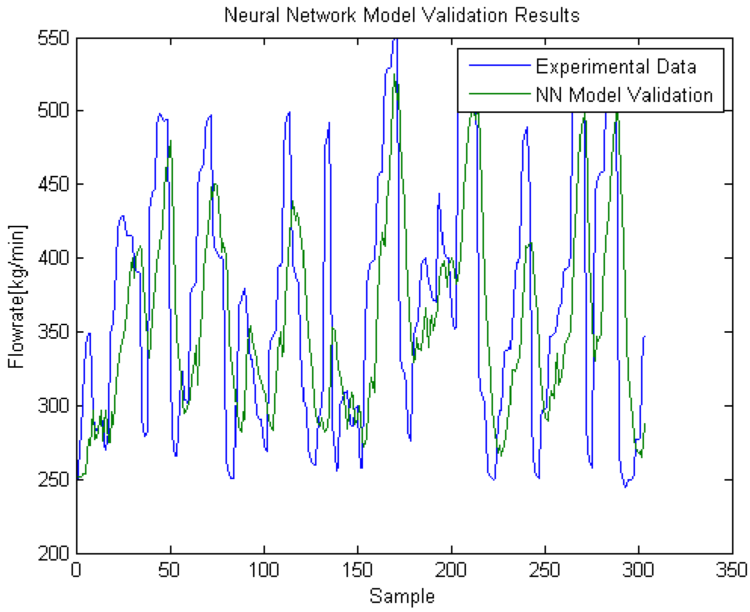

Figure 5.25 Validation results from NN Model (Raw data) with Data set 1

Mean error percentage = 17.1851%. RMSE = 27.3284 (kg/min)

5.5.1.2Validation of NN Model with data set 2

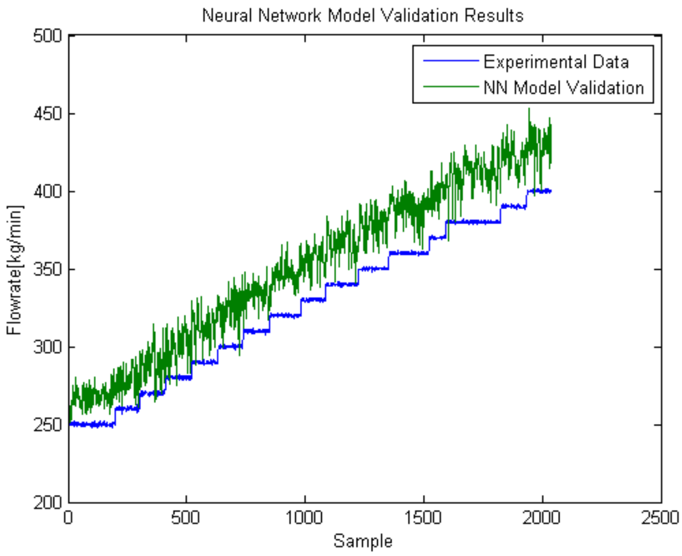

Figure 5.26 Validation results from NN Model (Raw data) with Data set 2

Mean error percentage = 7.8212%. RMSE = 27.3284 (kg/min)

With respect to the validation results obtained, Figure 5.25 and Figure 5.26 shows that the earlier obtained NN model with raw input data doesn't provide a good model (concerning the error values). However according to Figure 5.25 we can see that the dynamic changes in the Mass flow rate are somewhat tracked with respect to Data set1. But the Mean error percentage of this validation is around 17% which is quite high.

5.6 Selection of the best model

Concerning the results obtained with the two validation data sets it seems like Data set 2 has proven some proper results with the calibrated models. Therefore from here onwards the model comparisons will be done with respect to the validation results obtained with Data set 1.

Table 3 Summary of estimation and validation results of the obtained models.

No	Model	Model Calibration Data set		Data set 1 (Validation set)	
		Mean Percentage Error (%)	RMSE (kg/min)	Mean Percentage Error (%)	RMSE (kg/min)
1	DSR Model (Raw Data)	6.6315	30.4191	17.0416	54.6456
2	DSR Model (Noise reduced data)	2.4940	13.1398	9.7714e+13	3.1814e+14
3	SSPEM Model (Raw Data)	5.3968	26.4535	12.4483	41.0635
4	SSPEM Model (Noise reduced data)	2.8508	16.4539	1.7174e+13	5.6342e+13
5	N4SID Model (Raw Data)	2.3464	12.7317	2.0472	8.0770
6	N4SID Model ((Noise reduced data))	2.4171	12.8742	2.3047	9.2318
7	NN Model (Raw data)	2.0420	10.6112	7.8212	27.3284

Table 3 presents the summary of all the models obtained earlier in this chapter. We can clearly see that the subspace models obtained with N4SID algorithm have proven with good Mean Percentage errors as well as the RMSE values.

However from the two N4SID models model number 5 has the best validation results with a 2 % mean error percentage and with an RMSE value of 8 kg/min.

With respect to the validation results obtained, Figure 5.19 shows that the N4SID model obtained with raw data inputs provides the best model (concerning the error values). When considering the dynamics of the model validation this model provides the best dynamics with respect to the changes in the Mass flow rate.

Following is the respective discrete time linear model for the best N4SID model.

$$x_{t+1} = Ax_t + Bu_t$$

$$y_t = Dx_t + Eu_t$$

A = [0.9542 0.0111; 0.0027 0.9577]

B = 0.0000 0.0004 -0.0000; 0.0002 -0.0015 0.0009] D = [1.0e+03 * 1.6648 0.1180]

6 Conclusion

This report mainly explains the formation of models for flow rate estimations in an open channel venturi, with the use of subspace system identification methods. Several approaches were discussed which are Deterministic and Stochastic system identification and Realization (DSR) approach, State Space Prediction Error Methods with combining the N4SID subspace realization method in system identification toolbox in MATLAB. However the models were obtained from the experimental results that were achieved with 2 different fluids on the Venturi channel at University of Southeast Norway. The mass flow rate range was 250kg/min to 550kg/min.

With the use of singular value decomposition technique and the Principal component analysis method it was found out that the density of the fluid has really less correlation with the Mass flow rate, and mainly the level measurements highly correlates with the Mass flow. Therefore the density was not taken as an input variable. However the experiments were carried out with two different fluids which are water with a density of 1000kg/m^3 and Fluid 1 with a density around 1160 kg/m^3.

After obtaining the subspace models, each of the models were validated with two sets of different data which were taken by two separate experiments. More on the Neural Network model which was obtained in Abeywickrama et al., 2015 was also taken into consideration and this model was also validated with the same data sets. With comparison of the mean percentage error values and the root mean square error values from the validation results of all the models it can be concluded that the model obtained with the N4SID subspace algorithm together with the Prediction error methods is the best model. The mean percentage error of this model was around 2% and the RMSE value was around 8kg/min which was in the span of 250kg/min to 550kg/min.

Considering the estimation capability of the selected best model we can clearly see that this model can be used as a flow estimator in a control system used for off shore drilling operations. Another reason is that the model doesn't need the density of the fluid. Therefore there's no need of a Coriolis meter or advanced density measurement techniques to measure the density. Earlier with respect to the NN model obtained in Abeywickrama et al., 2015 the density measurement has played a crucial role in estimating the flow rate. But now it's not a matter. With the N4SID model the flow rate can be measured only with the height measurements with a 2% mean percentage error, irrespective of the properties of the fluids flowing inside the venturi channel.

Moving another advantage of the obtained N4SID subspace model is that it can be used for real time flow rate estimations without any problem. The reason is that it takes around 0.01 seconds to estimate a flow rate value. The sampling time of the ultrasonic flow measurements is 2 seconds. Therefore there are no issues in replacing this model as a flow rate estimator in a control system used for off shore drilling operations.

Appendices

Appendix 1 – Task Description

Telemark University College

Faculty of Technology

FMH606 Master's Thesis

Title: Flow-rate estimation in an open venturi channel

TUC supervisor: Håkon Viumdal (main-supervisor), Saba Mylvaganam (co-supervisor), David Di Ruscio (co-supervisor) and Khim Chhantyal (co-supervisor)

External partner: Geir Elseth, Statoil

Task background:

Drilling operations for oil and gas are becoming more and more advanced and complex due to real time monitoring and control of the processes involved in their executions. The wells to be constructed are becoming more complex leading to new and complex types of drilling procedures with increased need for various specialized tools and equipment.
Thus, there is currently an increased research activity on improving the drilling operation by enhancement of sensor and control systems. As a part of the sensor research, estimations of the flow rates using level measurements in open venturi channels are evaluated in mud flow applications.
The Faculty of Technology at Telemark University College has in collaboration with Statoil designed and assembled a flow loop in the process hall. The rig represent the first step in a potential R&D activity where different advanced sensor applications are to be developed and tested with respect to flow rate and rheological properties in drilling operations. The present project focus on utilizing level measurements in an open venturi channel for determining the flow rate. A multi data sensor fusion concept should also be validated, including readings from additional sensors mounted at the test rig.

Task description:

The present project aim at giving answers to the following questions:
Are level measurements in an open channel with varying cross sections an adequate sensor principle for flow rate measurements in a mud flow loop? If so, what are the advantages compared to standard devices like flow paddle and Coriolis?

The project involves both experimental and theoretical work, and should include the following topics:

Address: Kjølnes ring 56, NO-3918 Porsgrunn, Norway. **Phone:** 35 57 50 00. **Fax:** 35 55 75 47.

(1) Summarize the project reports related to the open channel test-rig at TUC, emphasizing sensors involved in the mud flow loop, with a special focus on flow rate estimations.
(2) Design and run new dedicated experiments with different fluids, flow rates etc. in the open venturi flow loop.
(3) Design a DSR-model to estimate the flow rate of the fluids, based on sampled data from the experiments. Different approaches such as other system identification methods, multi variate data analysis, neural network and fuzzy logic can be used in addition.
(4) Analyze and compare the results of the empirical model(s) designed in (3) with the other models developed by the other group projects summarized in (1).
(5) Discuss the feasibility of including this kind of flow estimators in a control system used for off-shore drilling operation.

Student category:

This project has been especially designed for SCE-students that have accomplished the master group project (SCE4006 Project): *Mud flow measurements in open venturi channel*, autumn 2015.

Practical arrangements:

Telemark University College has an open venturi channel as a part of a flow loop, in the process hall. The open channel flow loop will be used in the project.

Signatures:

Student (date and signature): Thanushan

Supervisor (date and signature):
03.02.2016

Appendix 2 - MATLAB program for centering of Data

```
function [Y1_1,X1_1,X2_2] = my_centering_calc(a,c,n)
%clear all
%clc
load('Water&Fluid1.mat');
%load('Jerry_water.mat')

b=1;
%a = 4700; %302; %152; %2450; numel(LT15);
if n==1||n==2
X1_1 = zeros(a,6);
X2_2 = zeros(a,3);
Y1_1 = zeros(a,1);
end

%%Without Scaling data
if n==0
Y1_1 = Mass_Flow(b:a);
X1_1 = [LT15(b:a) LT17(b:a) LT18(b:a) PDT(b:a) PT(b:a) Density(b:a)];
X2_2 = [LT15(b:a) LT17(b:a) LT18(b:a)];
end

for i=1:a/c
   p = ((i-1)*c+1);
   q = i*c;

   if n==1
   %%With Scaling and centering data
   %X1_1(p:q,:) =  [(LT15(p:q)-mean(LT15(p:q)))/std(LT15(p:q))
(LT17(p:q)-mean(LT17(p:q)))/std(LT17(p:q)) (LT18(p:q)-
mean(LT18(p:q)))/std(LT18(p:q)) (PDT(p:q)-mean(PDT(p:q)))/std(PDT(p:q))
(PT(p:q)-mean(PT(p:q)))/std(PT(p:q)) (Density(p:q)-
mean(Density(p:q)))/std(Density(p:q))];
   X2_2(p:q,:) =  [(LT15(p:q)-mean(LT15(p:q)))/std(LT15(p:q)) (LT17(p:q)-
mean(LT17(p:q)))/std(LT17(p:q)) (LT18(p:q)-
mean(LT18(p:q)))/std(LT18(p:q)) ];
   Y1_1(p:q,:) =  (Mass_Flow(p:q)-
mean(Mass_Flow(p:q)))/std(Mass_Flow(p:q));
   else if n==2
   %%With centering data
   Y1_1(p:q,:) = (Mass_Flow(p:q)-mean(Mass_Flow(p:q)));
%  X1_1(p:q,:) = [(LT15(p:q)-mean(LT15(p:q))) (LT17(p:q)-mean(LT17(p:q)))
(LT18(p:q)-mean(LT18(p:q))) (PDT(p:q)-mean(PDT(p:q))) (PT(p:q)-
mean(PT(p:q))) (Density(p:q)-mean(Density(p:q)))];
   X2_2(p:q,:) = [(LT15(p:q)-mean(LT15(p:q))) (LT17(p:q)-mean(LT17(p:q)))
(LT18(p:q)-mean(LT18(p:q))) ];
       end
   end
end

end
```

Appendix 3 - MATLAB program for DSR method

```
clear all
%clc
load('Water&Fluid1.mat');
%load('Jerry_water.mat')
```

```
b=1;
a = 4712; %302; %152; %2450; numel(LT15);


%%Without Scaling data
Y = Mass_Flow(b:a);

%centering and scaling with c number of blocks, n=1 scaling and centering
[Y1,X1,X2]= my_centering_calc(a,a,0);

%[B1,U1_1,S1_1,V1_1,T1,P1]=mypcr(Y,[Y1 X1],2);
%regression_coefficients_with_density = B1
%Loading_vector_with_density = P1

%[B2,U1_2,S1_2,V1_2,T2,P2]=mypcr(Y,[Y X2],2);
[B2,U1_2,S1_2,V1_2,T2,P2]=mypcr(Y,X2,2);
%regression_coefficients = B2
Loading_vector = P2


Final = U1_2*S1_2*V1_2';
%Theoritical solution
%X2 = [Final(:,2)+mean(LT15) Final(:,3)+mean(LT17)
Final(:,4)+mean(LT18)];
%Y1 = Final(:,1)+mean(Mass_Flow);

%To be realistic
%X2 = [Final(:,1)+mean(LT15) Final(:,2)+mean(LT17)
Final(:,3)+mean(LT18)];
%Y1 = Final(:,1)+mean(Mass_Flow);


%[A,B,D,E,CF,F,x0]=dsr(Y,X2,1);
%[y,x] = dsrsim(A,B,D,E,X2,x0);
%t = 1:1:numel(Y);
%plot(t,Y,t,y)


[A,B,D,E,CF,F,x0]=dsr(Y,X2,1);
%L =5, a = 152, centering c = 2
%L=1 a =4712 centering c = 0
[y,x] = dsrsim(A,B,D,E,X2,x0);
t = 1:1:numel(y);
plot(t,Y,t,y)
xlabel('Samples')
ylabel('Flowrate(kg/min)')
legend('Experiment', 'Estimation')
grid on
my_Err_Perc(Y,y)
```

Appendix 4 – MATLAB program for N4SID and SSPEM method

```
clear all
load('Water&Fluid1.mat');
%load('Data_29_10_2015_2fluids_shuffle.mat')
a = 4712; %302; %152; %2450; numel(LT15);

%%Without Scaling data
Y = Mass_Flow(1:a);

[Y1,X1,X2]= my_centering_calc(a,a,2);

%[B1,U1_1,S1_1,V1_1,T1,P1]=mypcr(Y,[Y1 X1],2);
%regression_coefficients_with_density = B1
%Loading_vector_with_density = P1

%[B2,U1_2,S1_2,V1_2,T2,P2]=mypcr(Y,[Y1 X2],2);
[B2,U1_2,S1_2,V1_2,T2,P2]=mypcr(Y,[X2],2);
%regression_coefficients = B2
Loading_vector = P2

Final = U1_2*S1_2*V1_2';
%X2 = [Final(:,2)+mean(LT15) Final(:,3)+mean(LT17)
Final(:,4)+mean(LT18)];
X2 = [Final(:,1)+mean(LT15) Final(:,2)+mean(LT17) Final(:,3)+mean(LT18)];
Y1 = Final(:,1)+mean(Mass_Flow);

X2 = [LT15 LT17 LT18];
Original = iddata(Y,X2,2);
opt = n4sidOptions('Focus','simulation');
N4SID_Result = n4sid(Original,2,opt);

PEM_Result = pem(Original,N4SID_Result);
compare(Original,PEM_Result,N4SID_Result);

[A,B,D,E,k,x0]=th2ss(PEM_Result);
[y,x] = dsrsim(A,B,D,E,X2,x0);
t = 1:1:numel(y);
plot(t,Y,t,y)
xlabel('Samples')
ylabel('Flowrate(kg/min)')
legend('Experiment', 'Estimation')

grid on
my_Err_Perc(Y,y)

%[A1,B1,D1,E1,K1,x01,V,th_0,th]=sspem(Y,X2,3);
%[y1,x1] = dsrsim(A1,B1,D1,E1,X2,x01);
%t = 1:1:numel(y1);
%plot(t,Y,t,y1)
%xlabel('Samples')
%ylabel('Flowrate(kg/min)')
%legend('Experiment', 'Estimation')
%grid on
%my_Err_Perc(Y,y1)
```

Appendix 5 – MATLAB program for error calculations

```
function [] = my_Err_Perc(Y,y)
error_perc = zeros(numel(Y),1);

for i = 1:numel(Y)
   error_perc(i) = abs((y(i)-Y(i))/Y(i))*100;

end

mean_error_percentage = mean(error_perc)
RMSE = sqrt(mean((Y-y).^2))
end
```

Appendix 6 – DSR and SSPEM algorithm

These two algorithms were developed by David Ruscio (Ruscio, Subspace System Identification, 1995)in his research work (Ruscio, Model Predictive Control and optimization, 2001). The MATLAB toolboxes for these two algorithms can be obtained by referring to the following website.

http://www-pors.hit.no/tf/fag/sce2206/framdriftsplan.html

Appendix 7 – MATLAB program for Neural Network model

This program was obtained from the previous research work carried out by (Thanushan Abeywickrama, Jeremiah Ejimofor, Minh Hoang, Aderonke OKoro, 2015).

```
clear all;
clc;
load('Cal_Val_Test.mat');
x1 = Density';
%x2 = PDT';
x3 = LT15';
x4 = LT17';
%x5 = LT18';
%x6 = PT';
Out = Mass_Flow';

t = 1:1:numel(Out);
Cal_Error = zeros();
Val_Error = zeros();
Test_Error = zeros();

Min_Max = [
           minmax(x1);
           %minmax(x2);
           minmax(x3);
           minmax(x4);
           %minmax(x5);
           %minmax(x6);
           ];
Inputs = [
          x1;
          %x2;
          x3;
          x4;
          %x5;
          %x6;
          ];

[trainInd,valInd,testInd] = divideblock(numel(Density),0.6,0.2,0.2);
mse_init = mean(var(Out(:,trainInd)));

for i=1:1:20
    net =  newff(Min_Max,[i,1],{'tansig' 'purelin'}, 'trainlm');

    net.divideFcn = 'divideblock';
    net.divideParam.trainRatio = 0.6;
    net.divideParam.valRatio = 0.2;
    net.divideParam.testRatio = 0.2;

    net.trainParam.lr = 0.3;
    net.trainParam.goal = 0.01*mse_init;
    net.trainParam.epochs = 1000;
    net.trainParam.show = 25;
    net.trainParam.time = inf;
    net.trainParam.min_grad = 1e-10;
    net.trainParam.max_fail = 1000;

    [net , tr] = train(net,Inputs,Out);
```

```
    T_t(i) = i;
    Cal_Error(i) = sqrt(tr.best_perf);
    Val_Error(i) = sqrt(tr.best_vperf);
    Test_Error(i) = sqrt(tr.best_tperf);

    clear [net , tr];

end

subplot(2,1,1);
plot(T_t,Cal_Error,T_t,Val_Error,T_t,Test_Error);
title('Error vs No. of Neurons')
legend('Cal','Val','Test')

[C,Neurons_C] = min(Cal_Error)
[V,Neurons_V] = min(Val_Error)
[T,Neurons_T] = min(Test_Error)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

net =  newff(Min_Max,[Neurons_T,1],{'tansig' 'purelin'}, 'trainlm');

net.divideFcn = 'divideblock';
net.divideParam.trainRatio = 0.6;
net.divideParam.valRatio = 0.2;
net.divideParam.testRatio = 0.2;

net.trainParam.lr = 0.3;
net.trainParam.goal = 0.01*mse_init;
net.trainParam.epochs = 1000;
net.trainParam.show = 25;
net.trainParam.time = inf;
net.trainParam.min_grad = 1e-10;
net.trainParam.max_fail = 1000;

[net , tr] = train(net,Inputs,Out);

subplot(2,1,2);
Test = net([
            x1(:,tr.testInd);
            %x2(:,tr.testInd);
            x3(:,tr.testInd);
            x4(:,tr.testInd);
            %x5(:,tr.testInd);
            %x6(:,tr.testInd);
            ]);

Test_Err_mse = tr.best_tperf
Test_Err_normal = sum(abs(Out(:,tr.testInd)-Test))/numel(Test)

t1 = 0:1:(numel(Test)-1);
plot(t1,Out(:,tr.testInd),t1,Test);
legend('Experimental Data','NN Model Prediction')
xlabel('Sample')
ylabel('Flowrate[kg/min]')
title('Neural Network Model Prediction Results for Water and Fluid 1')
```

Appendix 8 – MATLAB program for validating the NN Model

```
% a = 305;
% a = 2040;
% Density = zeros(a,1);
% for i = 1:a
% Density(i) = 1016;
% end

Test = net([
            Density';
            %x2(:,tr.testInd);
            LT15';
            LT17';
            %x5(:,tr.testInd);
            %x6(:,tr.testInd);
            ]);


Y = Mass_Flow;
y = Test;
error_perc = zeros(numel(Y),1);

for i = 1:numel(Y)
   error_perc(i) = abs((y(i)-Y(i))/Y(i))*100;
end

mean_error_percentage = mean(error_perc)
RMSE = sqrt(mean((Y-y').^2))

t1 = 0:1:(numel(Test)-1);
plot(t1,Mass_Flow,t1,Test);
legend('Experimental Data','NN Model Validation')
xlabel('Sample')
ylabel('Flowrate[kg/min]')
title('Neural Network Model Validation Results')
```

Appendix 9 – MATLAB program for simulation of discrete-time linear systems

This program was developed by David Ruscio in his research work (Ruscio, Subspace System Identification, 1995).

```
function [y,x] = dsrsim(a,b,d,e,u,x0);
%DSRSIM Simulation of discrete-time linear systems.
%       Y=DSRSIM(A,B,D,E,U)
%       Y=DSRSIM(A,B,D,E,U,x0)
%    PURPOSE:
%    Simulate the time response of the discrete system:
%
%                x_{t+1} = Ax_t + Bu_t
%                y_t     = Dx_t + Eu_t
%
%    ON INPUT:
%    A,B,D,E - Discrete dynamic model matrices.
```

```
%    U          Matrix U must have as many columns as there are inputs, u.
%               Each row of U corresponds to a new time point, a (N x r)
matrix
%               where r is the number of inputs.
%    x0       - Optional initial state vector. x0 is a (n x 1) vector
where
%               n is the number of states.
%    ON OUTPUT:
%    Y        - Matrix Y with system outputs, i.e. a (N x m) matrix where
m is
%               the number of output variables.
%               When invoked with left hand arguments,
%               Y = DSRSIM(A,B,D,E,U) or Y=DSRSIM(A,B,D,E,U,x0)
%    X        - Matrix X with system states, i.e. a (N x n) matrix
%               When invoked with left hand arguments,
%               [Y,X] = DSRSIM(A,B,D,E,U) or [Y,X]=DSRSIM(A,B,D,E,U,x0)
%               returns the output and state time history in the
%               matrices Y and X.
%-----------------------------------------------------------------------
---

% WRITTEN TO BE USED AS SUPPLEMENT TO THE DSR IDENTIFICATION ALGORITHM
% DATE: 26. august 1996

%1.
[N,r]=size(u); [m,n]=size(d);

%2. Define the initial state vector
if nargin == 5
 xm=zeros(n,1);
else
 xm=x0;
end

%3. Initialize output arrays
y=zeros(N,m); x=zeros(N,n);

%4. Simulation loop for evaluation of the states
for i=1:N
% store xm in array x for the states
  x(i,:)=xm';

% update (integrate) the state vector
  xm = a*xm + b*u(i,:)';

end

%5.  Compute the outputs outside the loop for increased speed
y=x*d'+u*e';
%
% END DSRSIM
```

Appendix 10 - MATLAB program for SVD and PCA

```
function [B,U1,S1,V1,T,P]=mypcr(Y,X,im);

[N1,nr]=size(X); [N2,nm]=size(Y); N=max(N1,N2);
X=X(1:N,:);
Y=Y(1:N,:);

if im == 1
 XX = X'*X;
 Y  = X'*Y;
 [U S V]=svd(XX);
else
 [U S V]=svd(X);
end


si=diag(S(1:nr,1:nr))
r=nr;
prompt = 'Number of principal components ?';
r = input(prompt);

end
```

References

Agu, C. (2014). *Model based estimation of drilling mud flow using aVenturi channel.* Porsgrunn: Telemark Univeristy College.

Esbensen, K. H. (2010). *Multivariate Data Analysis - in practice.* Esbjerg: CAMO.

Ljung, L. (2009). *System Identification theory for the User.* Sweden : Prentice Hall PTR.

MathWorks. (2006). *MathWorks*. Retrieved from mathworks.coml: http://se.mathworks.com/help/ident/ref/pem.html

MathWorks. (2016). *idss*. Retrieved from mathworks.com: se.mathworks.com/help/ident/ref/idss.html

MathWorks. (2016). *n4sid*. Retrieved from mathworks.com: http://se.mathworks.com/help/ident/ref/n4sid.html

Overschee, P. V. (1992). *N4SID : Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems*. Belgium: Research Associate of the Belgian National Fund for Scientic.

Ruscio, D. D. (1995). *Subspace System Identification*. Porsgrunn, Norway: Telemark Institute of Technology.

Ruscio, D. D. (2001). *Model Predictive Control and optimization*. Porsgrunn: Telemark University College.

Ruscio, D. D. (2006). *An introduction to MATrix LABoratory.* Porsgrunn: Telemark University College.

Thanushan Abeywickrama, Jeremiah Ejimofor, Minh Hoang, Aderonke OKoro. (2015). *Mud Flow Measurements in Open Venturi Channel*. Porsgrunn: Telemark University College.