University College of Southeast Norway
Faculty of Technology
–

Master's Thesis
Study programme: SCE4206
Spring 2016

Kateryna Baranovska

Model Free Optimal and Predictive Control of the K-Spice Process Simulator

University College of Southeast Norway
Faculty of Technology
http://www.usn.no

This thesis is worth 30 study points

University College of Southeast Norway

MASTER'S THESIS, COURSE CODE FMH606

Student: Kateryna Baranovska

Thesis title: Model Free Optimal and Predictive Control of the K-Spice Process Simulator

Signature: . . . .

Number of pages: <90>

Keywords: System Identification, Linear Quadratic Regulator, PI controller, Slug Flow, well-pipeline-riser, K-Spice, Identification, Validation

Supervisor: David Di Ruscio Sign.: . . . .

2nd supervisor: Sign.: . . . .

Censor: Sign.: . . . .

External partner: Sign.: . . . .

Availability: Open

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

Abstract:

Impact of the slugging can significantly decrease oil production and have damage consequences for equipment. Nowadays, Model free optimal and predictive control is actual topic of discussion and could be not-exhaustive field for research. The process that describes the behaviour of slug flow was running in K-Spice simulator, while control algorithm was designed in MATLAB.

For implementing the model predictive control with integral action, the state space model of the research process should be known. This model can be obtained by using system identification algorithms e.g. DSR, PEM and N4SID, which could achieve high-performed results with accurate state space model by using only input and output data from a real process. Firstly, it were detected all possible control signals and output variables and then, after detailed review, were formulated four control strategies. For all control strategies were done open-loop simulation for identification the bifurcation point and PRBS experiment for collecting input and output data from the real process that was running in K-Spice simulator.

The process model has been developed by implementing three different system identification methods DSR, PEM and N4SID, which used data from PRBS experiment. The basic algorithm of closed loop system with linear quadratic regulator was firstly discussed and then implemented for each control strategy. The LQR was design based on the obtained DSR model and then tested on the model and implemented to the real process running by K-Spice simulator. In contrast to the linear quadratic regulator was implemented PI controller that was tuned by MATLAB application.

Described procedure was performed for all control strategies, which were highlighted in this project. Achieved results from all control strategies were discussed and it was formulate a list of suggestions for the further work.

Contents

Nomenclature

Abbreviation	*Description*
GUI	Graphical user interface
TCP/IP	Transmission Control Protocol/ Internet Protocol
OPC	Open Platform Communications
OPC DA	Open Platform Communications Data Access
PRBS	Pseudo Random Binary Signal
SSM	State Space Model
DSR	Deterministic and Stochastic system identification and Realization
PEM	Prediction error model
N4SID	Subspace state space system identification
LQ	Linear Quadratic
LQR	Linear Quadratic Regulator
PI-controller	Proportional–Integral controller
MAE	Mean absolute error
IAE	Integrated absolute error
eq.	Equation

Foreword

Master Thesis “Model Free Optimal and Predictive control of the K-Spice Process Simulator” is submitted to the University College of Southeast Norway, Porsgrunn, Norway that was the mandatory requirement to fulfil the Master’s degree in System and Control Engineering. Reader should have some knowledge about “Advanced control theory” and “System identification and optimal estimation” courses. The thesis was written with supervision of David Di Ruscio and two external supervisors Roar Nilsen and Christer Dalen.

I want to express my sincere gratitude to David Di Ruscio for his support and direction in the writing of this work and separate gratitude for his patience and willingness to help at any time. Creating a project has been facilitated with huge amount of literature provided by him and the various toolbox for developing the programs scripts.

I would like to thank Roar Nilsen for his technical support and help with solving all problems, which appeared with K-Spice simulator. Special thanks to Christer Dalen for his moral support and for guidance through the project.

Finally I want to thank University College of Southeast Norway for opportunity to be a student and to increase my knowledge in different and exciting classes.

Porsgrunn, 27th May 2012

Kateryna Baranovska

1 Introduction

Oil production - a complex production process that requires significant technical and economic resources. One of the most important parts of the oil production is to stabilize the flow in the system. Nowadays, it has been developed and has been implemented a large number of strategies to control the flow and manage with the “slugging” that could appear in well-pipeline-riser. But this problem is still be relevant and actual. Effect of the slug flow could be destructive for the system, that is the reason of high actuality of this problem.

Kongsberg Group has been developed the K-Spice simulator for establishment process that is close enough to real process.

This software can simulate the behavior of the flow, pressure and temperature in the system over a long time period. Consider functionality of the program, should be mentioned that PI controller connections of different forms are used as basic control strategies. From that statement, the following questions are raised. What is the efficiency of these methods? Is it possible to use other, more pragmatic control techniques? Which one is better?

This software together with the model, which describes the behavior of the flow in the system, has been provided for the work on this project. This model allows discovering of the various control strategies efficiency.

Design controller strategies and implementation of the controllers, which can stabilize the flow, could be challenging task. Model based controller such as LQR can solve this problem. However, it is not fair to underestimate the possibility of well-known PI controllers. To design a controller is necessary to have parameters of mathematical model of the system. This model could be developed by using different system identification methods.

Nowadays there are a significant amount of articles and book, which can explain the nature and behavior of slug flow but the problem of anti-slug flow control is not less actual area for research. The research topic of implementation different control strategies with PI controller were discussed in significant amount of publications. As an example could be represented cascade control.

Sigurd Skogestad (2005) presented an article where it has been researched the cascade control configuration using the topside pressure, in the outer loop together with the mass flow, in the inner loop as anti-slug control strategy. Advance control algorithm was discussed in article Christer Dalen (2015) where he represented the result of implementation LQR and compared them with PI-controller.

1.1. Objectives

The main objectives of this project is obtaining of mathematical model based on collected input and output data from the real process simulator and design and implementation model based controllers into real process simulator together with PI-controllers. Three different identification methods will be presented and compared based on achieved results. The task description is available in *Annex 1*. To summarize all main objectives of this project, it is possible to represent the next key steps:

- Detect possible inputs and outputs of the system
- Collect input and output data from the K-Spice simulator that describes real process
- Develop the state-space model using different identification methods and select the best identification method
- Using the best-obtained model design and implement linear quadratic regulator and compare the simulation result with PI-controller

1.2. Report Structure

The report was drawn up for better discussing and representing all the main objects as well as providing all information related to this project. Excluding formal pages, the main part of report consist of 67 pages and contained figures and tables. The report has been divided into chapters with the next contents:

1.Introduction.

Introduction chapter contains the basic description of the problem and general project overview.

2.System description.

System description chapter describes general process together with simulation description and contains information about using software and communication methods.

3.Theory and methods

This chapter contains basic theoretical definitions and different identification methods, which are related to obtain the main goals of this project.

4.Simulation result

This chapter represents all simulation results from collecting data experiment, implementation of system identification methods, design and implementation of LQR and PI controller. This chapter contains 4 main subchapters that describe 4 implementing control strategy.

5.Discussion

In this part of the report are described what was done during the project together with result of each separated chapter.

6. Conclusion

General conclusions of done work are represent in this chapter together with further work suggestions.

2. System description

2.1. Process description

As already known irregular flow or, as it is called, the slug flow can be a reason of equipment damage and has negative impact on oil production at all. According to the article Espen Storkaas (2006) slug flow in pipeline system can be classified in the next way: hydrodynamic slugging, riser slugging, terrain and transient slugging. The attention will be more focus on riser slugging, which behavior could be described by Figure 2-1.

The oil flow can accumulates in the bottom of riser, which block the gas flow through the riser and after increasing of the pressure riser become full of liquid. But the gas flow still be blocked in the low point of the riser and after some time the amount of the gas will be enough to blow the oil stream out of the riser. After the liquid and gas will start to accumulate again so it is possible to conclude that slugging is a cycle process.

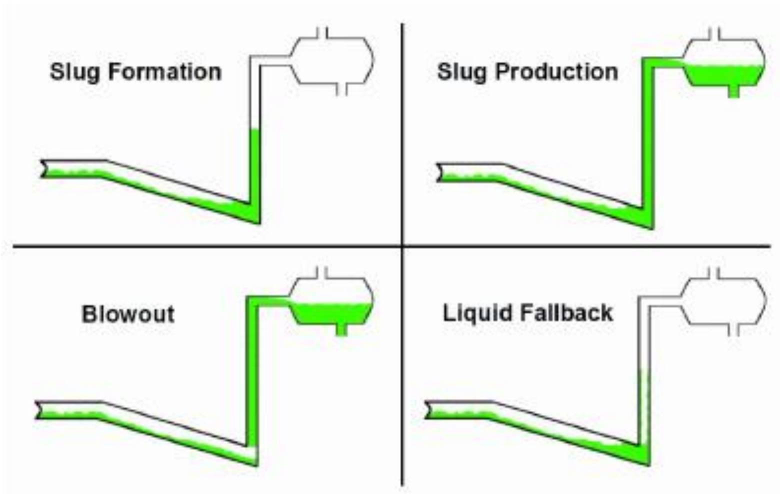


Figure 2-1: General overview of "slugging" (Espen Storkaas (2006)).

Slug flow can destroy separator and has a negative impact on the receiving objects during offshore oil and gas production due to the large fluctuations in flow rates and pressure. All this aspects has a strong influence on the economical aspect of oil and gas production.

Kongsberg oil & gas technologies provided the K-Spice process-model of the well-pipeline-riser, which is shown at Figure 2-2. The general illustration of the riser process can be performed as at Figure 2-2.

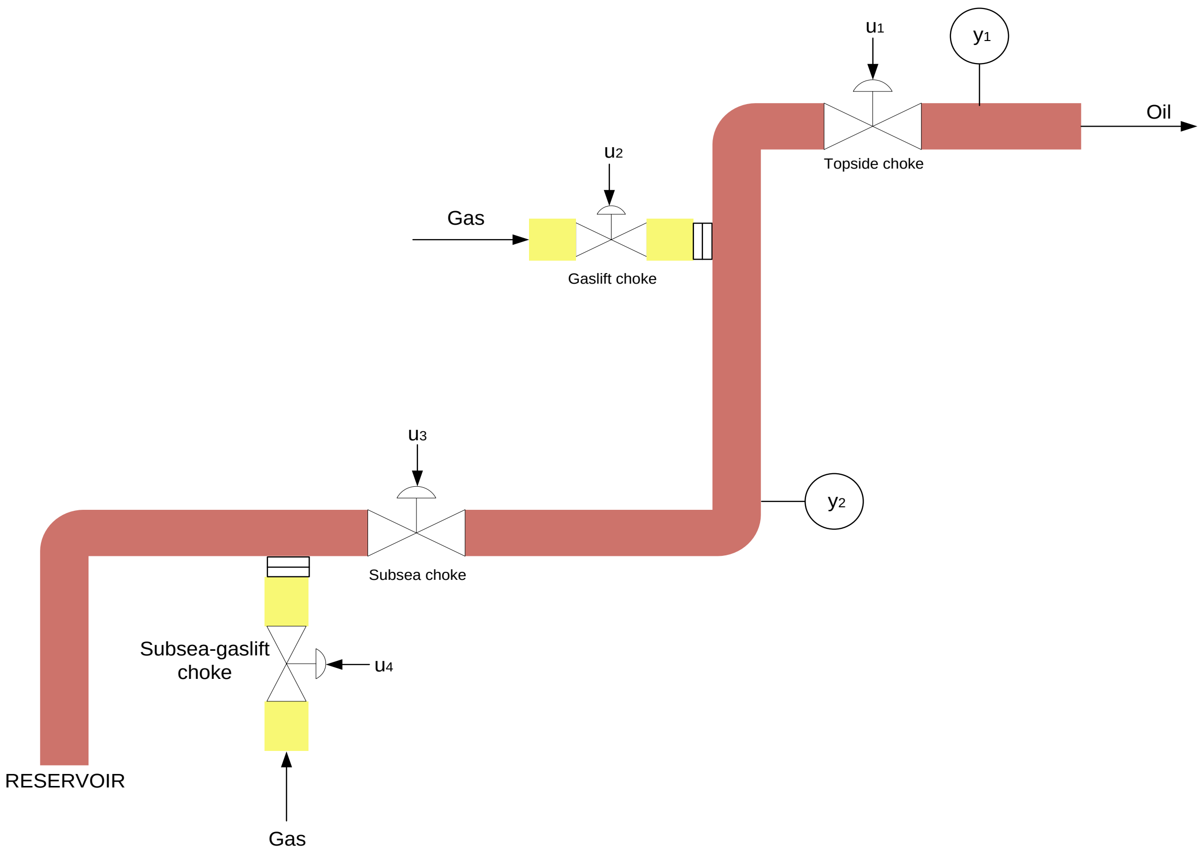


Figure 2-2 Illustration of the well-pipeline-riser according to K-Spice model

According to the block-schema of the model that was presented in Annex 2 and illustration of the riser in Figure 2-2 there are mentioned 4 manipulative inputs for controlling and stabilizing the flow: topside choke, gaslift choke, subsea choke and subsea-gaslift choke. Based on the foregoing controlled inputs, it was possible to identified four control strategies. Each of these strategies involves various combinations of control chokes.

Topside choke, which is mentioned as PIC001 in the model, as input signal u_1 in the illustration, locates above the sea level, and is a main control choke in all strategies as it is located directly before topside separator. The gaslift choke u_2 or *FIC001* is using for stabilizing flow together with the topside choke by reason of reducing density and increasing the flow rare. For comparison, it is also used the subsea-gaslift choke, that is marked as u_4 at Figure 2-1 and as *FIC001GL* on the model. The impact of manipulating of the subsea choke u_3 (*HIC-1* in the model) is considered in separated strategy but at the same moment was used in all mentioned strategies.

As it mention in Eikrem (2014) "slugging" can be removed from the process by stabilizing the outlet flow or stabilizing the pressure in the riser, since these parameters are interdependent. Exactly these parameters are defined in the model as output signals: y_1- outlet flow, which measurements is indicated by flow transmitter FT100, and y_2- pressure in the riser, which values received from pressure transmitter PT006.

Stream equality constrains was given as:

- Gaslift: T=30^0*C*; P=150 *bar*
- Subsea: T=100^0*C*; P=320 *bar*

The inequality constrains can be formulated mostly for input signals as:

$$0 \le u_1 \le 100$$
$$0 \le u_2 \le 100$$
$$0 \le u_3 \le 100$$
$$0 \le u_4 \le 100$$

2.2. Simulation description

Process model that describes the behaviour of oil and gas in the pipeline has been developed in K-Spice simulator. The model has been calculated and developed by using the LedaFlow and implemented in K-Spice program. Unfortunately, some of control strategies is difficult to implement in K-Spice simulator. For example using of MPC algorithm for controlling oil flow and reducing slugging can be complicated for the program. On this basis, it was decided to apply and develop the program script with MPC algorithm and develop an LQ-regulator in MATLAB. All input signal, which regulate opening of valve, will be send from MATLAB to K-Spice simulator using OPC server. At the same time values from transmitters, which describe the behaviour of flow will be send from K-Spice to MATLAB. The general overview of the system are shown at Figure 2-3.

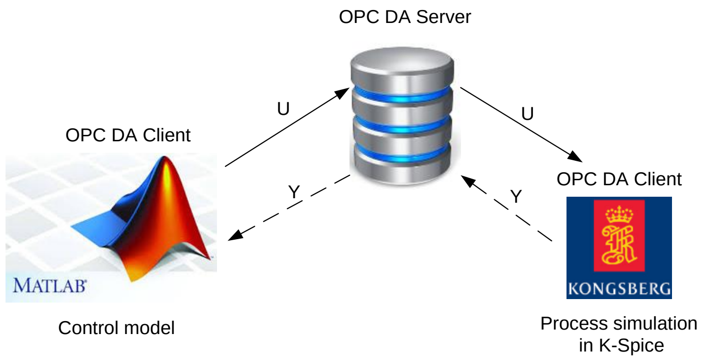


Figure 2-3 General system overview

In the Figure 2-3 U is set of input signals to the control valves and Y is set of output signals. As OPC server is it possible to use MatrikonOPC server. In this master's thesis all of the components were runn in one computer but it is also possible to run application in different machines.

2.3. K-Spice and LedaFlow software

K-Spice is a program software that has been developed by Kongsberg Group since 1989, for simulation different chemical process and especially for oil and gas processes. This software can provide a clear description of the behaviour of the system using the mass and energy balance. A simple user interface helps create a model using different modules and connections. The model can be divided in the separated sections, which connected between each other. K-Spice simulator can be easy linked to other simulators such as LedaFlow or connect to process databases, PLS and SCADA systems. This connection can be implemented by using OPC protocol. When design and development of the model are finished is it possible to test model and implemented control strategy. Information about K-Spice simulator was taken from user guide KONGSBERG (2015).

K-Spice simulator based on separated from each other programs such as:

- K-Spice® SimulationManager
- K-Spice® SimExplorer
- K-Spice® Model Server
- K-Spice® Model Control Language

These programs can communicate through data files and communication protocol based on TCP/IP standards. All project starts by running K-Spice® SimulationManager that is responsible for management all projects files, setups and communication between different applications. Figure 2-4 shows the GUI of the K-Spice® SimulationManager application with “Opening window” and “Running program window”. According to this figure, it is clearly seen that Manager will inform user about error or warring situations, which could happened during the process simulation.

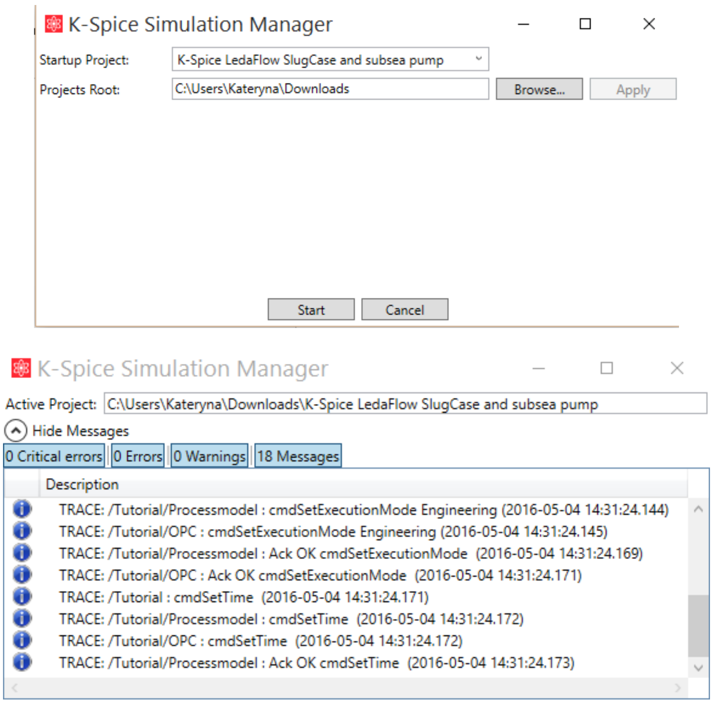


Figure 2-4 K-Spice® SimulationManager

K-Spice® SimulationManager can be connected with several K-Spice® Model Server using a TCP/IP protocol so they can be distributed across different machines. The main tasks of K-Spice® Model Server are execution of the process simulation calculations and supporting of model configuration. Figure 2-2 shows screenshot of running K-Spice® Model Server application.

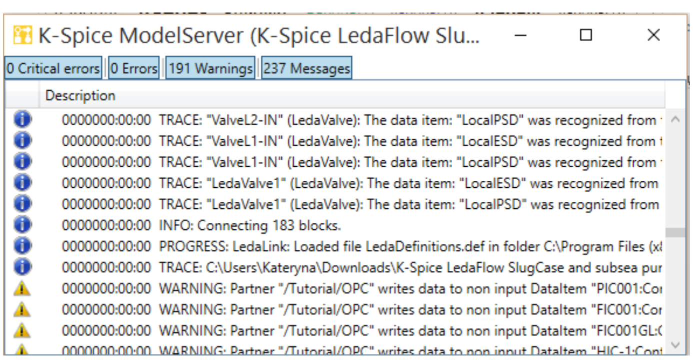


Figure 2-5 K-Spice® Model Server

K-Spice® SimExplorer provides configuration and execution interface for simulation of the process. With this application, it is possible to create or edit the structure of the system, add or remove elements into/from model. The GUI of K-Spice® SimExplorer is shown at Figure 2-5. One of the important moments is that this application can be connected to K-Spice® Model Server via K-Spice® SimulationManager.

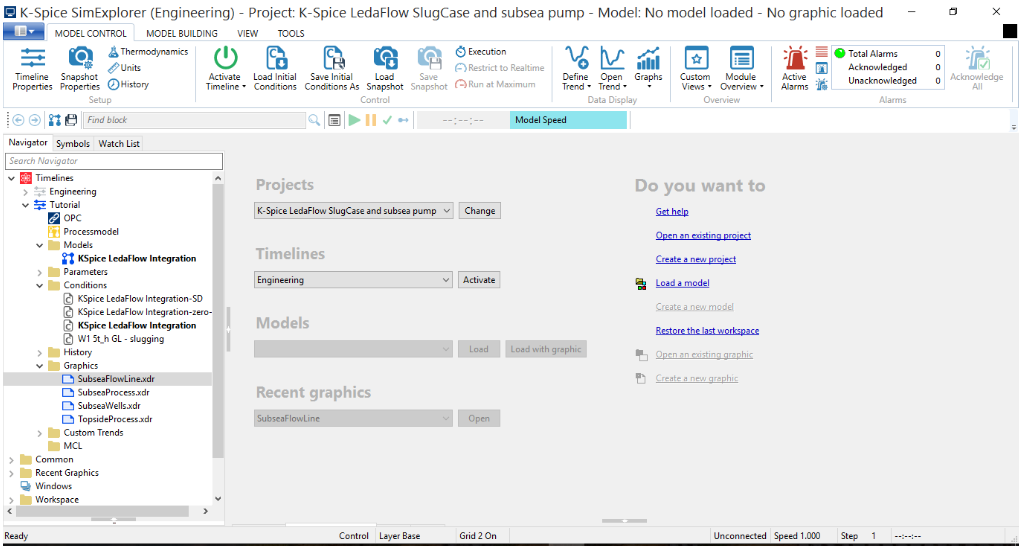


Figure 2-6 K-Spice® SimExplorer

According to Figure 2-6, it is clearly seen that one project can have multiple conditions and have clear hierarchical structure with one time line or multiple timelines, which consist from one or multiple applications. Timeline is responsible for running time of the model and indicates the model speed, which can be selected, general running time and synchronizing load. Other important functions of time line are saving and initializing operations. Timelines header controls partner application, which is responsible for physical calculation of the model. Partner application situates below timeline in the hierarchy and could be run in parallel with other partner application. Application can be also a driver for communication with process database systems, control systems or other eternal systems. It is also possible to load some model parameters from others programs such as LedaFlow.

According to LedaFlow installation guide, LedaFlow® Engineering is an application suite for performing 1D multi-phase flow simulation studies and simulations. It is also the name of the graphical user interface where you can build, configure, run and analyze models, which were created in LedaFlow®. Is it possible to connect LedaFlow model with parameters to K-Spice simulator.

2.4. OPC

OPC is a series of standards and specifications for industrial telecommunication and provides a single interface for managing the automation objects and technological processes (Wikipedia (2016a)). The main objective of the OPC standard is to ensure the possibility of collaboration (interoperability) all automation equipment, which are functioned on different hardware platforms, different industrial networks and produced by different firms.

2.4.1. OPC DA

In this project work, it was used OPC DA server. OPC DA (Data Access) is a group of server-client standards and provides specifications for real time data communication from all devices of data acquisition system. (Wikipedia (2015)). The main functions of OPC DA are reading of current measured values, calculation and value estimation and writing data. According to specification of OPC DA, data transfer can be performed only in real time and it is not possible to read/write any historical data. The sample of value could be composed of next fields: the value, quality, timestamp, access right and properties (that can be description, units etc.) and could be written or read synchronous or asynchronous and it is possible to read as a subscription. OPC DA server contains one tag for each measurement and controller points in the plant, and OPC DA server is responsible for getting or setting the information from controllers. (Skeie (2014))

OPC DA server can have user interface that allows user to perform all support functions to relieve all kinds of operation with the equipment. For example, MatrikonOPC supplies specialized software for working with the OPC-servers. MatrikonOPC Explorer and MatrikonOPC Server for Simulation and Testing were used in this master's thesis for server creation and data communication between K-Spice and MATLAB (both of these programs have their own OPC DA Toolboxes for data communication). The GUI of these two programs are shown at Figure 2-7

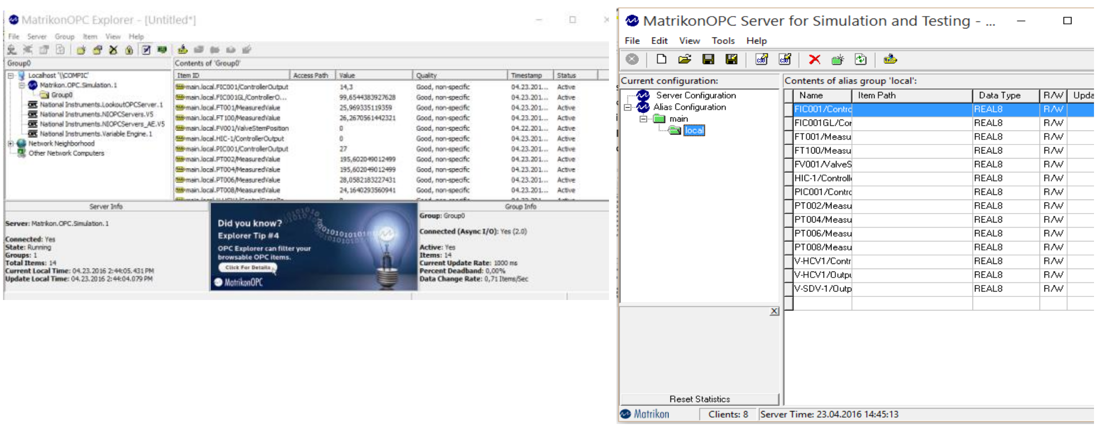


Figure 2-7 GUI of MatrikonOPC Explorer and MatrikonOPC Server for Simulation and Testing

According to the information, which is provided by MatrikonOPC web-site, Matrikon OPC Explorer is a specialized software to work with the OPC-servers that is one of the most popular software packages for OPC platform and it was developed by Matricon Inc, which is currently a division of Honeywell. MatrikonOPC Simulation Server is a free utility used to help test and troubleshoot OPC applications (clients) and connections. (Inc (2016))

2.4.2. OPCDaCom in K-Spice

It is one of the program toolkit, which relates to the K-Spice software package and connects to the K-Spice® SimulationManager and handles all I/O exchanges between active timelines and the remote DCS. (KONGSBERG (2014), KONGSBERG (2015))

In the beginning steps a new link application should be created in K-Space Sim Explorer in a project folder. After project activation, K-Space Sim Explorer will launch the link for data transport. Figure 2-8 shows user interface of SimLink application in K-Spice. Using this application, it is possible to get name and address of the OPC servers and complete the table in MS Access database. All information about measured data and OPC DA setings is collected in these tabels. Using this application it is also possible to control the values that are transported between the project and the server. (KONGSBERG (2014))

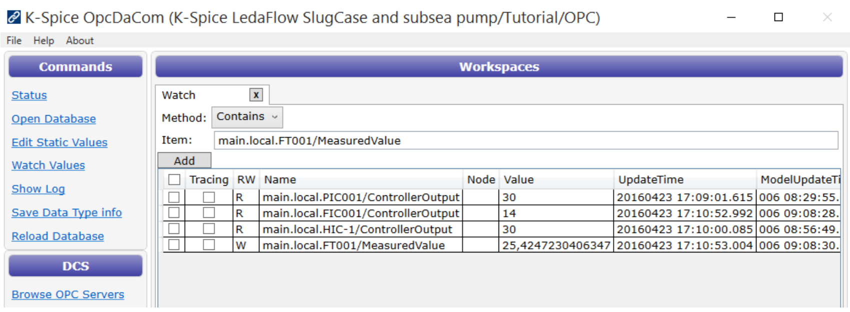


Figure 2-8 SimLink application

For changing OPC setting from read to write it is possible to change the setting in database in “Bidirectional” column: 0 – means that it is only possible to write the data and -1 – means that is possible read and write data from/to database.

3. Theory and Methods

In this part of the report will be presented theoretical definitions related to the definition of the model and types of input signals, a discussion of various methods of identification of the model as well as the different types of regulators such as LQ and PI controllers. Together with the methods will be presented the main MATLAB-function and toolboxes, which were used for developing a program code.

3.1. State Space Model

In a classical form, the state space model (SSM) with signal input and signal output of continuous time system can be formulated as:

$$x_{k+1} = Ax_k + Bu_k \tag{3-1}$$

$$y_k = Dx_k \tag{3-2}$$

Where, $x_k \epsilon \mathbb{R}^n$ is the state vector, $u_k \epsilon \mathbb{R}^r$ is the input vector, $y_k \epsilon \mathbb{R}^m$ is the output vector, *A, B, D* – constant matrices with size $n \times n, n \times r$ and $m \times n$ respectively. This type of model input signal has direct influence to the output.

The well-pipeline-riser process, which was described in Process description can be described by linear discrete time-invariant state space model (LTI) that can be formulated as:

$$\bar{x}_{k+1} = A\bar{x}_k + Bu_k + Ce_k \tag{3-3}$$

$$y_k = D\bar{x}_k + Eu_k + Fe_k \tag{3-4}$$

Where $\bar{x}_k \in \mathbb{R}^n$ – prediction state vector and $e_k \in \mathbb{R}^m$ is the innovation with covariance matrix $E(e_k e_k^T) = I$. A, B, C, D are transition, input, external input and output matrix respectively. Matrix E is direct input to output matrix and F - direct external input to output matrix.

Is it possible to represent *3-3 and 3-4* in other way:

$$\bar{x}_{k+1} = A\bar{x}_k + Bu_k + K\varepsilon_k \tag{3-5}$$

$$y_k = D\bar{x}_k + Eu_k + \varepsilon_k \tag{3-6}$$

Where $\varepsilon_k \in \mathbb{R}^m$ is the innovation with covariance matrix $E(e_k e_k^T) = FF^T$ and $K = CF^{-1}$ is Kalman filter gain.

3.2. Input signals

As input signals will be used two types of signal changes: step and pseudorandom binary sequence (PRBS). According to WIKIPEDIA. 2016b , the step function signal can by given as:

$$u_t = \begin{cases} 0 & t < 0 \\ u_0 & t \geq 0 \end{cases} \quad (3\text{-}7)$$

Using this type of input signal is it possible to choose the amplitude u_0. The time of rising and static gain of step signal will have strong influence on the step response.

The Pseudo Random Binary Sequence (PRBS) is a binary signal that is approximated the spectrum of white noise and generated by a deterministic random generator. The pseudo random bit stream usually has two meanings of value +1 and -1. (Wikipedia (2013)) The PRBS remind a white-noise as far as spectral properties are concerned and it is also possible to generate a signal with prescribed spectral properties, which makes PRBS simple testing signal for most of identification methods. To use PRBS user should choose high and low level of the signal, period and clock period, which can be equal to 1 sampling time.

According to Di Ruscio (2014,) a binary single input experiment signal can be written as MATLAB function:

U=prbs1(N,Tmin,Tmax) (3-8)

Where *U* is input matrices, and $u_k \in \{-1, 1\} \, \forall \, 1 \leq t \leq N$; *N- total number of samples; Tmin-* minimum sample interval*; Tmax*- maximal sample interval. The PRBS signal generate with sample intervals *Ti*, which are random and should fulfil boundaries $T_{min} \leq T_i \leq T_{max}$. The behavior of signal changing will be depend on boundaries parameters, for example, with the low T_i values the input signal will changed slowly and with the high T_i value the signal will changed more rapidly.

Data received from the PRBS experiment is used for identifying model with more occurred high order *n*. (Torsten Soderstrom, 1988)

3.3. System identification

System identification can be explained as possibility to build a model of the system based on observed data from the real process as well as process criterions. Input and output data from the real process will be collected to the file and then by applying different methods of system identification as Deterministic and Stochastic system identification and Realization (DSR), Predictive Error Methods

(PEM) and subspace state space system identification (N4SID). The key point of system identification is to identify the system order and the extended observability matrix from the real process data that is a common feature for all methods. The detailed theory can be found in Ruscio (2009) , Di Ruscio (2014,) and Ljung (1995).

3.3.1. Deterministic and Stochastic system identification and Realization

The key parameters of this method are not only system order (n) but also the number of time moments in the horizon that are necessary for identifying the state at time and the extended observability matrix of the system (L), the number of time instants in the past horizon (J), predictive horizon (M) and structure parameter (g). The clear illustration of horizon was represent by Di Ruscio (2014,) and shown at Figure 3-1

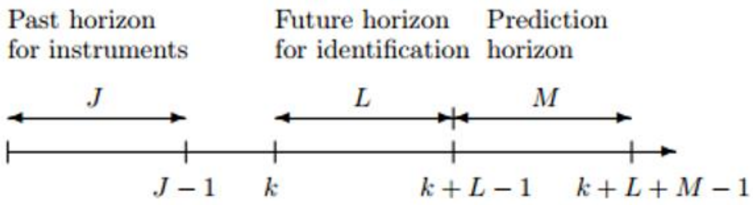


Figure 3-1 Illustration of horizons (Di Ruscio (2014,))

There are certain rules related to the selection of these parameters such as the system order that should be chosen according to the limits, where L is identification horizon and m is the number of outputs, the J parameter should be chosen as small as possible in order to remove the estimate variance and J should be equal or higher then L. For using DSR methods in MATLAB it is possible to run the DSR-Toolbox (Ruscio (2000)). The model matrices can be identifying by function:

$$[A,B,D,E,CF,F,x0]=dsr(Y,U,L,g,J,M,n) \tag{3-9}$$

Where C= CF*inv(F)-Kalman gain, g could be equal to 0 or 1 (if g=0 then E=0) and M is default as M=1. For purpose to calculate the Kalman filter gain, it does not need to solve the Riccati equation. The MATLAB script is available in annex. Full information about DSR methods is available in Di Ruscio (2014,), Ruscio (2007) .

3.3.2. Predictive Error Methods

The basic steps of PEM methods are described in Di Ruscio (2014,) and Ruscio (2001) . In this method it is not possible to choose the different past and future horizons parameters as it was done in DSR.

PEM uses numerical optimization to reduce the prediction error (PE), which is represented as the difference between the measured output and the predicted output of the model.
User can selected only one parameter that is system order n. The MATLAB function of the PEM is:

$$sys = pem(data,init\_sys) \qquad (3\text{-}10)$$

Where *data* is estimation data that contains input and output data from the real process, init_sys is identified model that configures the initial parameterization. The MATLAB script with implementation of PEM is available in *Annex 8*.

3.3.3. Subspace State Space System Identification

In N4SID method as in PEM is not possible to set past and future horizon as well as it does not contain structure parameter g. So if i=L+1, the smallest i which can be chosen in N4SID for the 1st order system will be 2. The computation of extended observability matrix should be done with one oblique projection and orthogonal projection. So the Kalman filter gain can be found by calculation the Riccati equation. N4SID has implementation in MATLAB as function:

$$sys = n4sid(data,nx) \qquad (3\text{-}11)$$

Where data is estimation data from real process and nx is order of estimation model. The implementation of N4SID methods is available in *Annex 8*.
Detailed information about N4SID methods could be found in Peter Van Overschee (1996) and Peter Van Overschee (1994)

3.4. Linear quadratic regulator

LQ-problem could be called the case where dynamic of the system are described by a set of differential equations and the cost, that is described by a quadratic function. Linear-quadratic regulator (LQR) can provide the solution of LQ problem, which is a feedback controller. The illustration of the closed loop control system with LQR is shown at Figure 3-2.

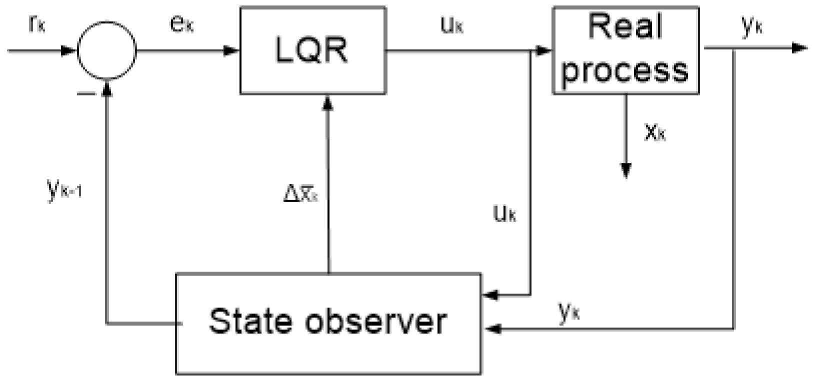


Figure 3-2 Closed loop system with LQR

According to Ruscio (2003) the standard LQ optimal control problem could be preformed as:

$$\Delta u_k = [G_1 \;\; G_2]\begin{bmatrix} \Delta x_k \\ y_{k-1} - r \end{bmatrix} \tag{3-11}$$

The equation can be rewritten as:

$$u_k = u_{k-1} + G_1 \Delta \bar{x}_k + G_2 (y_{k-1} - r_k) \tag{3-12}$$

Where $\Delta \bar{x}_k = \bar{x}_k - \bar{x}_{k-1}$ is state deviation, r is reference G1 and G2 are optimal feedback matrices. This equation describe the system, which is shown at Figure 3-2.

The state observer $\Delta \bar{x}_k$ that is shown at Figure 3-2 could be evolved from (3 5) and formulated as:

$$\Delta \overline{x_{k+1}} = A\Delta \bar{x}_k + B\Delta \bar{u}_k + K(y_k - y_{k-1} - D\Delta \bar{x}_k) \tag{3-13}$$

Where $\Delta \bar{u}_k = u_k - u_{k-1}$, the model matrices A,B,D and E were identified by one of the identification method and should be specified (usually $\Delta \bar{x}_0 = 1$).

The MATLAB function (Ruscio (1995)) for the optimal feedback matrices G1 and G2 was formulated as:

[G1 G2]=dlqdu_pi(A,B,D,Q,R) (3-14)

Where A,B,D - discrete state space model matrices, Q - weighting matrix for the output vector yk and R is weighting matrix for the control deviation vector. The implementation of LQ regulator is shown in Annex 9.

4. Simulation results

This chapter of the master's thesis presents the results from simulations of the four flow control strategies:

1st strategy is to stabilize flow by controlling the topside choke;

2nd strategy is to stabilize flow by controlling the gaslift choke;

3rd strategy is to stabilize flow by controlling the subsea choke;

4th strategy is to stabilize flow by controlling the subsea gaslift choke.

For each strategy, it was performed the results from open-loop simulation, which main task is to obtain the bifurcation point and the PBRS experimental for collecting samples. Each strategy consists of two cases that use different types of output signal. In the first case, the output signal is defined as outlet flow of the system y_1 and in the second case the output signal - the pressure in the riser y_2. Three different identification methods DSR, PEM and N4SID were implemented in each of cases. According to task, it was applied LQ and PI controller for control and stabilize the "slugging". Settings of OPC connection is shown at *Annex 3*. The MATLAB script for open-loop simulation is available in *Annex 4*, for PBRS experiment in *Annex 5*, DSR, PEM and N4SID in *Annex 6*, *Annex 7*, *Annex 8* and LQ and PI controllers in *Annex 9* and *Annex 10* respectively.

4.1. 1st strategy: Control of topside choke

In the first experiment, the flow is controlled only by topside choke (*PIC001* on the model), when subsea choke (*HIC-1* on the model) is constant and equal 50% of opening. Both gas-lift chokes are equal to 0, which means that gas flow does not enter into the system. So it is possible to define inputs and outputs for the 1st case in equation :

$$y \in \Re := \{y_1 \tag{4-1}$$

$$u \in \Re^2 := \begin{cases} u_1 \\ u_3 \end{cases} \tag{4-2}$$

Where output signals were collected into $y \in \Re^3$: y_1 - outlet flow,[*ton/hr*] and inputs $u \in \Re^2$: u_1 and u_3 are topside and subsea chokes respectively,[%] and u_3*=50%.* In the same way, it were formulated the inputs and outputs condition for the 2nd case

$$y \in \Re := \{y_2 \tag{4-3}$$

$$u \in \Re^2 := \begin{cases} u_1 \\ u_3 \end{cases} \tag{4-4}$$

Where y_2 - pressure in the bottom of riser [*bar*]; The duration of each simulation is 3600 sec with the model speed in K-Spice 25-30 sec to the real-time.

4.1.1. Open-loop simulation

Open loop simulation does not have any controllers or system feedback in the system and it has been done for visualizing the slug flow and finding the bifurcation point that is point when process pretend to be stable.

Figure 4-1 shows open-loop simulation with step changing of input signal from 10% to 90% with the step size of valve opening 30%. Figure 4-2 represent the open loop simulation result for outlet flow and pressure in the bottom of riser.

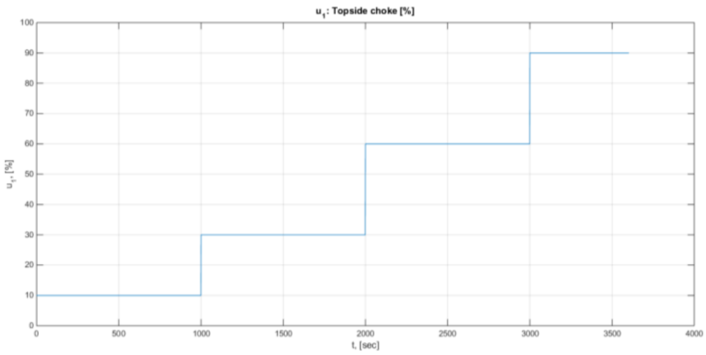


Figure 4-1 Input signal to topside choke, subsea choke kept constant at 50%

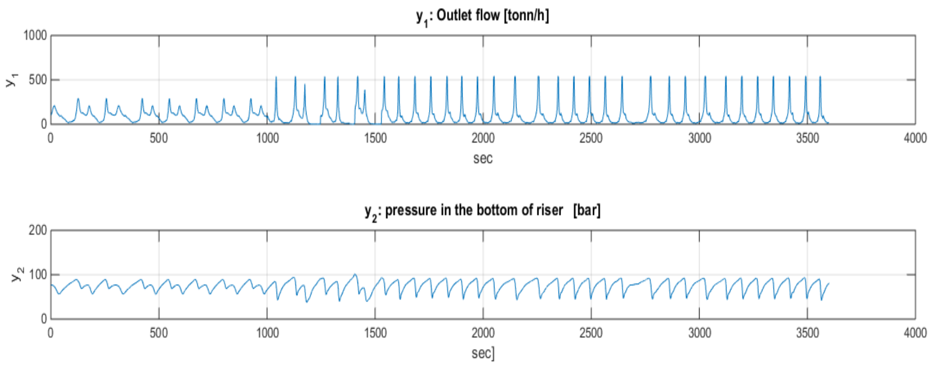


Figure 4-2 Output simulation results of outlet flow and pressure in the bottom of riser

It is clearly seen that the flow is unstable and oscillate with the same behavior, while control signal was equal to 10%. After increasing of the signal up to 90% the irregular flow was changed with the same amplitude from 1.78 ton/hr to 589.4 ton/hr. The same behavior is typical for pressure in the bottom of the riser. Based on the received results it can be assumed that bifurcation point is above or below the monitoring range of input signal changing.

Figure 4-3 shows the simulation results for outlet flow and pressure with input changing from 2% to 10%. As it clearly seen from the figure the flow is stable from 2% until 8% and when input signal is equal 10% then flow has a small oscillations. The key point of stabilizing flow is to keep the pressure as small as possible, which is basically mean to have higher amount of oil production. According to Figure 4-3 it is possible to assume that bifurcation point is u_1=8% because the pressure is the lowest with this control signal and flow does not have any significant oscillation. The output flow oscillate in range of 125 ton/hr.

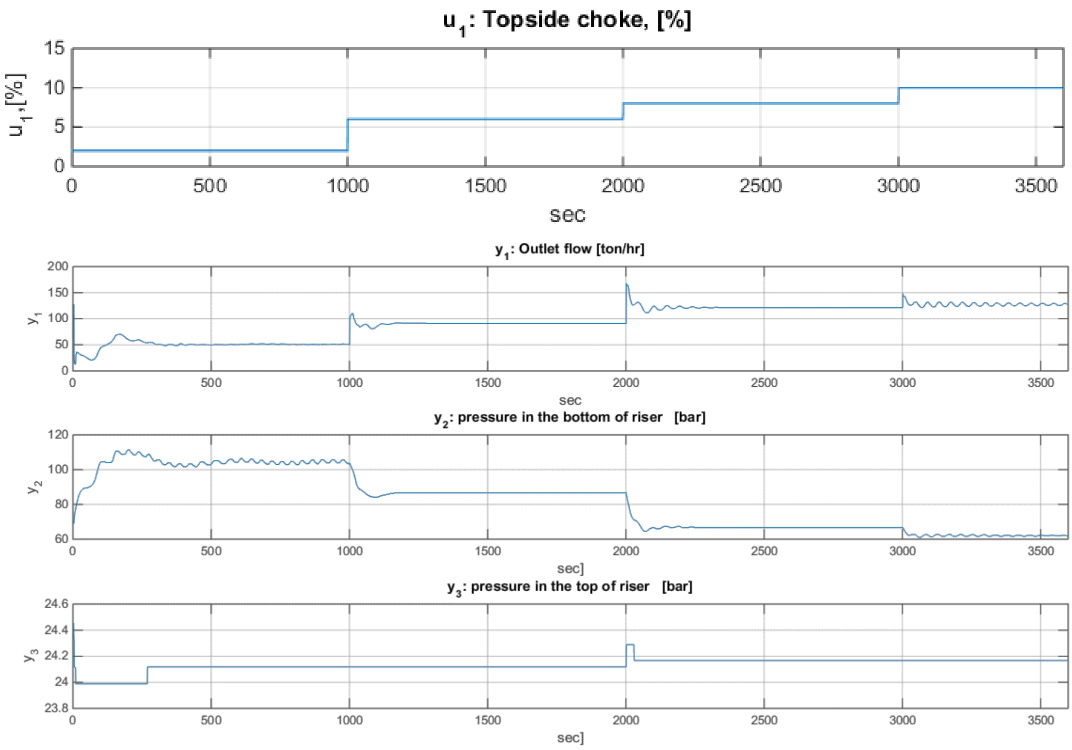


Figure 4-3 Input signal changing and output simulation results of outlet flow and pressure in the bottom of riser

For better overview of the system and collecting samples for future work it was done the PRBS experiment with the low amplitude, which was described in chapter "Input signals"by eq. (3-8). During this experiment the input signal were kept constant for first 100 samples and then changed with amplitude of 3% from bifurcation point and T_{min}= 30 sec and T_{max}=80 sec. PRBS simulation results are shown at Figure 4-4 PRBS simulation results of the 1st strategy Figure 4-4.

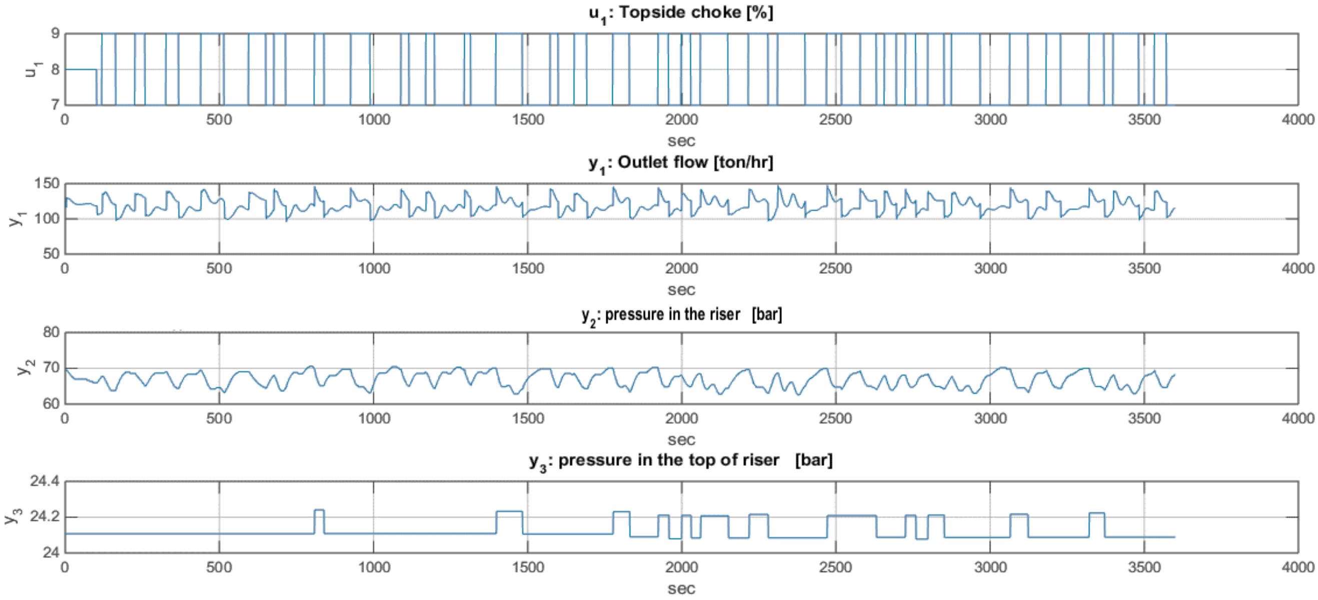


Figure 4-4 PRBS simulation results of the 1st strategy

All collected data were saved in "topsideO.mat" and will be used for model identification and estimation of the parameters.

4.1.2. System identification using DSR, PEM and N4SID

All aforementioned methods of system identification were implemented for both, first and second cases. To obtain a more accurate models, data that was getting during the PRBS experiment were scaled and centered as at shown at Figure 4-5.

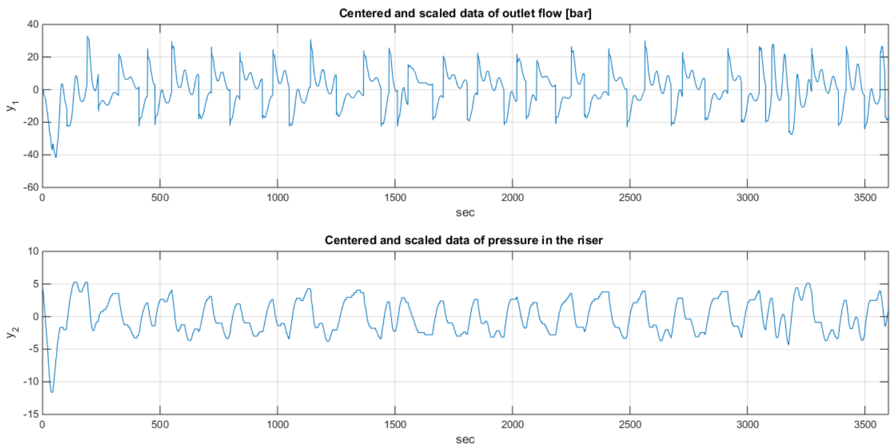


Figure 4-5 Centered and scaled output data of the 1st strategy

DSR methods with parameters *L*=60, *J*=68 and system order *n*=5 provides the best result with the minimal *MSE*=11.7425, for 100% samples were used for identification. During the experiment data set was divided on identification and validation set, which were separated by 75% and 25%, respectively. Mean absolute error of identification set is *MSE*=13.8174 and MSE of validation set is less then for identification set and equal to 6.9006. The model refitted by DSR method with *zero*=0.9848 and *pole*=0.8773 fulfills the conditions of the system stability. Figure 4-6 shows the behavior of the output signal y_1 *(Yid on the figure)* from the actual process as the results of simulation in the K-Spice and the output signal y_1 (*Ym on the figure)* from the received DSR model. The sate matrices of the model *A,B,D* and *E* were identified as :

$$A = \begin{bmatrix} 0.9749 & -0.1072 & 0.0145 & 0.1208 & 0.1424 \\ 0.0660 & 0.9804 & 0.0806 & 0.0552 & 0.1377 \\ -0.0037 & -0.0754 & 0.9982 & -0.0206 & -0.0614 \\ -0.0218 & 0.0192 & 0.0071 & 0.8992 & -0.3391 \\ -0.0138 & 0.0207 & 0.0074 & -0.1207 & 0.5336 \end{bmatrix}$$

$$B = \begin{bmatrix} -2.3275 \\ -3.0556 \\ -0.2929 \\ 4.3602 \\ 5.5751 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.2169 & -0.1663 & 0.0537 & 0.4356 & 0.7113 \end{bmatrix}$$

$$E = 0$$

To obtain the best model by using PEM, the system order *n* was changed in bounds *n*=3:7. The best result was achieved with *n=5* when MSE was minimal and equal to 16.8617. The output result from identification model together with output from the process are shown at Figure 4-6. The system appears to be stable with pole 0.6155 and zero 0.6612. Obtained through PEM model identification, stat matrices *A,B,D,E* appear as follows:

$$A = \begin{bmatrix} 1.0380 & 0.2082 & 0.0423 & -0.0205 & 0.0450 \\ -0.0433 & 0.4806 & -0.4247 & 0.3321 & -0.0984 \\ 0.0431 & -0.1879 & 0.8295 & 0.7821 & 0.4156 \\ 0.0265 & -0.1741 & -0.1483 & 0.1063 & 1.1307 \\ -0.0102 & 0.0371 & 0.0183 & 0.1823 & 0.6231 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0127 \\ -0.0424 \\ -0.0166 \\ -0.0110 \\ 0.0018 \end{bmatrix}$$

$$D = \begin{bmatrix} 309.6901 & -89.6280 & -39.8196 & 7.7341 & -4.8657 \end{bmatrix}$$

$$E = 0$$

A similar principle of finding the system order has been applied for N4SID methods. In accordance with a test in which *n* varies within *n=3:7,* the best result was obtained with *n=5* and MSE=11.2745. The graphical interpretation of the achieved model together with output from the process are shown at Figure 4-6. The stat matrices A,B,D and E :

$$A = \begin{bmatrix} 0.9793 & 0.2083 & 0.04079 & -0.02055 & 0.04516 \\ -0.04496 & 0.4273 & -0.4192 & 0.3257 & -0.09747 \\ 0.04411 & -0.1846 & 0.7774 & 0.7775 & 0.4164 \\ 0.02243 & -0.1518 & -0.1351 & 0.03836 & 1.136 \\ -0.001191 & 0.01019 & -0.01386 & 0.2101 & 0.5583 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.01337 \\ -0.04221 \\ -0.01311 \\ -0.01347 \\ 0.0005935 \end{bmatrix}$$

$$D = \begin{bmatrix} 309.7 & -89.63 & -39.82 & 7.734 & -4.866 \end{bmatrix}$$

$$E = 0$$

For better overview all simulation result are shown at one graph, Figure 4-6.

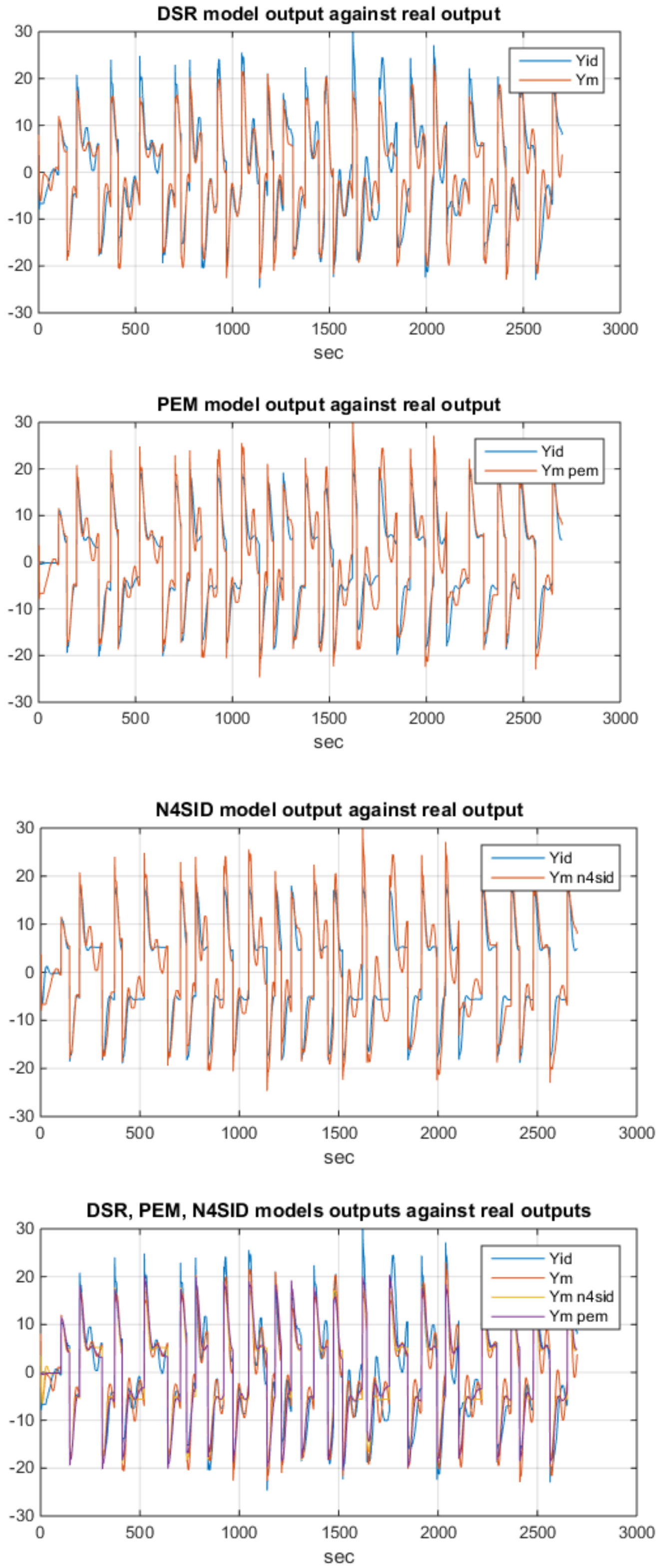


Figure 4-6 DSR, PEM and N4SID models outputs against outputs from the real process for the 1st strategy (1st case)

Table 4-2 shows comparable characteristics of three different methods: DSR,PEM and N4SID.

Table 4-1 DSR, N4SID and PEM identification characteristics for the 1st strategy (1st case)

Method	*n*	*L*	*J*	*MSE*	*Poles*	*Zeros*
DSR	5	60	68	11.7425	0.8773	0.9848
PEM	5	–	–	16.8617	0.6155	0.6662
N4SID	5	–	–	11.2745	0.6911	0.7481

Similar experiments for DSR, PEM and N4SID were conducted for the 2nd case also. According to obtained results, DSR was used with parameters *L=34, J=37* and system order *n=5.* MSE of validation set is 1.0630 and MSE of the identification set is 1.5260.

PEM and N4SID preformed the best results with *n=5.* The simulation results are shown at *Figure 4-7*, where output from different identified models were compared with output from the process. Characteristics of the identified models are described in the Table 4-2. According to this table models, which were obtained by PEM and N4SID algorithms, are unstable.

The stat matrices *A,B,D* and *E* of DSR model were identified as following:

$$A = \begin{bmatrix} 0.9643 & -0.0681 & -0.0693 & 0.1492 & 0.0577 \\ 0.0022 & 1.0004 & -0.1909 & -0.0233 & 0.0553 \\ 0.0065 & 0.0643 & 0.9879 & 0.2855 & 0.0580 \\ -0.0008 & 0.0005 & -0.0015 & 0.9537 & -0.1658 \\ -0.0007 & -0.0004 & 0.0008 & 0.0236 & 0.6900 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.4618 \\ -0.0598 \\ -0.1285 \\ 0.0130 \\ 0.0672 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.2756 & -0.1732 & -0.3158 & 0.4066 & 0.2258 \end{bmatrix}$$

$$E = 0$$

The stat matrices *A,B,D* and *E* of PEM model were identified as following:

$$A = \begin{bmatrix} 1.0644 & -0.0525 & 0.0596 & -0.0076 & -0.0114 \\ 0.0936 & 1.0990 & 0.2830 & -0.1219 & -0.0273 \\ -0.0377 & -0.0987 & 0.8748 & 0.4280 & 0.3491 \\ 0.0042 & -0.0589 & 0.0689 & -0.7538 & 0.4630 \\ 0.0057 & 0.0194 & -0.2368 & 0.2987 & -0.1203 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0005 \\ 0.0021 \\ -0.0063 \\ 0.0108 \\ 0.0050 \end{bmatrix}$$

$$D = \begin{bmatrix} 82.3558 & -1.0101 & -2.6808 & 1.1895 & 0.6099 \end{bmatrix}$$

$$E = 0$$

The stat matrices *A,B,D* and *E* of N4SID model were identified as following:

$$A = \begin{bmatrix} 0.9666 & -0.0396 & 0.0356 & -0.0106 & -0.0084 \\ 0.0468 & 0.9675 & 0.2757 & -0.1204 & -0.0279 \\ -0.0521 & -0.0123 & 0.6530 & 0.3976 & 0.3395 \\ 0.0030 & -0.0391 & 0.0711 & -0.8552 & 0.3524 \\ -0.0051 & 0.0006 & -0.1087 & 0.0785 & -0.5160 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0006 \\ 0.0047 \\ -0.0082 \\ 0.0096 \\ 0.0051 \end{bmatrix}$$

$$D = \begin{bmatrix} 82.3558 & -1.0101 & -2.6808 & 1.1895 & 0.6099 \end{bmatrix}$$

$$E = 0$$

Figure 4-7 shows comparable graphics, where scaled output of the real process is compared with models output.

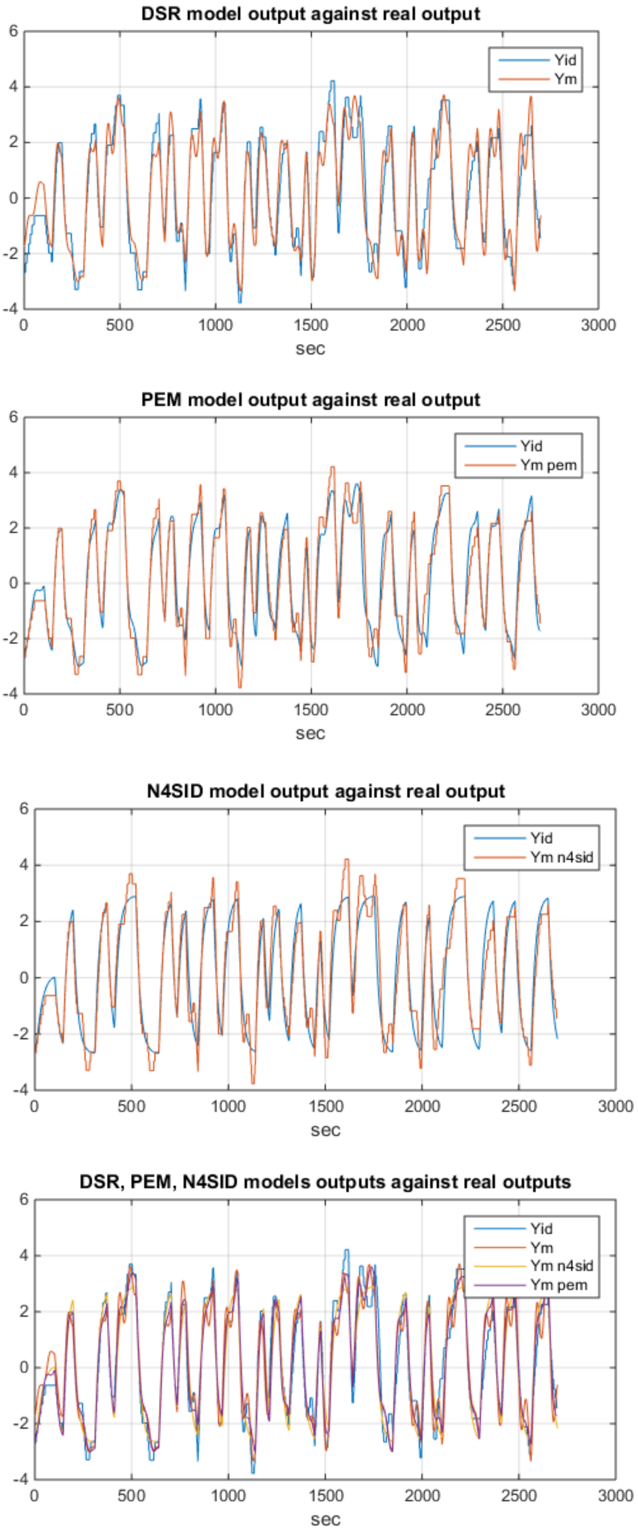


Figure 4-7 DSR, PEM and N4SID models outputs against outputs from the real process for the 1st strategy (2nd case)

Table 4-2 DSR, N4SID and PEM identification characteristics for the 1st strategy (2nd case)

Method	*n*	*L*	*J*	*MSE*	*Poles*	*Zeros*
DSR	5	34	37	0.3042	0.9818	0.9637
PEM	5	–	–	0.3251	0.4328	-1.3633
N4SID	5	–	–	0.3036	0.2432	-0.3118

4.1.3. Implementation of LQ-regulator

Figure 4-8 shows the implementation of LQ controller, which was described in chapter "Theory and Methods".

LQR controls the pressure in the riser according to the reference signal throughout the simulation time T=6000 sec. Considered model is single input and single output so the weighting matrices Q and P were chosen as Q=0.5 and P=500 and it was implemented DSR identified model where output signal was pressure in the riser y_2. First 500 sec of simulation was done without controller, which is easy to notice from the control input graph u_1. According to the *Figure 4-8,* in the begging u_1=7 and after 500 sec control signal was sit by LQR and not by user.

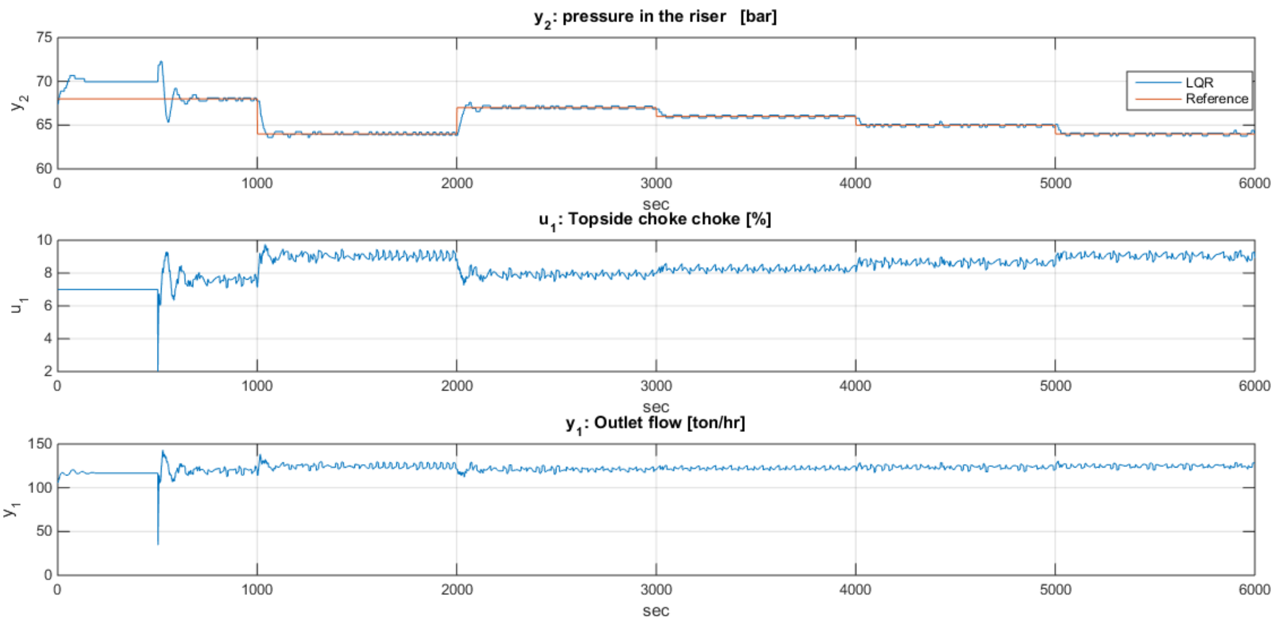


Figure 4-8 LQR of the pressure in the riser for the 1st structure

As it clearly seen from the *Figure 4-8* that output signal y_2, which is pressure in the riser, tracks the reference signal with the good performance. The basis of this conclusion was the integrated absolute error (IAE) that is equal to 374. Also it is clearly seen that by controlling the pressure in the riser the outlet flow is also stabilized and flow oscillations does not exceed 7%. The feedback matrices G_1 and G_2 are :

$$G_1 = [\ -0.1097 \quad 0.0004 \quad -0.0055 \quad -0.0681 \quad -0.0303]$$

$$G_2 = [0.0218]$$

4.1.4. Implementation of PI-controller

MATLAB PID Tuner Toolbox was used to determine PI-controller parameters Kp and Ti. For this purpose, it was used model that was identified by DSR methods. The *Figure 4-9* shows the implementation of PI-controller with the parameters Kp=18 and Ti=108. According to this figure, y_2 tracks the reference signal with the small oscillations and IAE=544. The first 500 sec simulation was done without PI-controller and starting point was lower then bifurcation point u_1=7%.

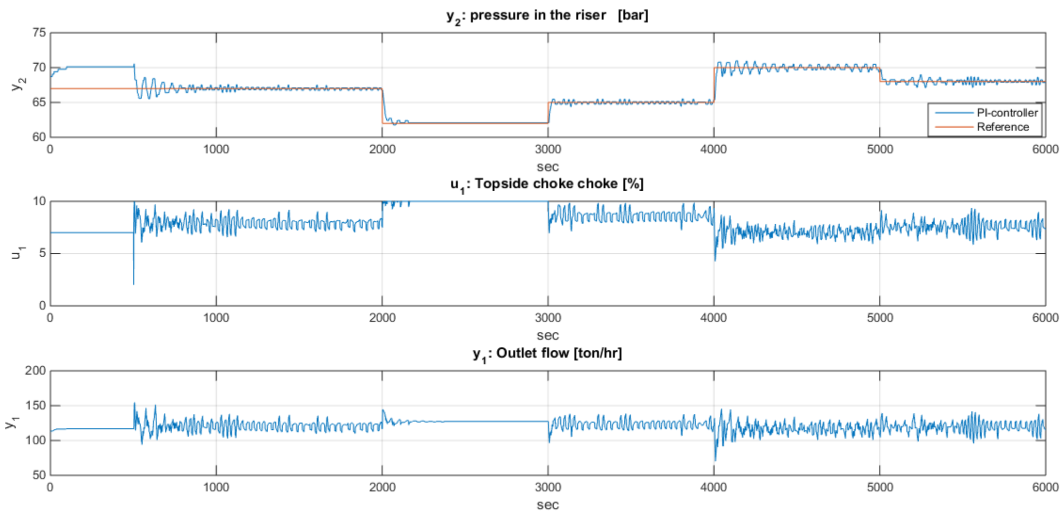


Figure 4-9 PI-controller of the pressure in the riser for the 1st structure

4.2. 2nd strategy: Gas-lift choke

According to Eikrem (2014) one of the solution for stabilizing slug flow is using of gas-lift choke together with topside choke. In this set of experiment all manipulations were applied to gas-lift choke u_2, while topside choke and subsea choke were kept constant u_1=8% and u_3=50%. The subsea gas-lift choke u_4 kept also constant and equal to 2.5%. The 2nd strategy also contains two cases as it was mention in preface of Simulation results. According to this description, it is possible to define inputs and outputs for 1st case of this experiment as :

$$y \in \Re := \{ y_1 \qquad (4\text{-}5)$$

$$u \in \Re^4 := \begin{cases} u_1 \\ u_2 \\ u_3 \\ u_4 \end{cases} \qquad (4\text{-}6)$$

Where output signals were collected into $y \in \Re^3$: y_1 - outlet flow,[*ton/hr*] and inputs $u \in \Re^2$: u_1, u_2, u_3 and u_4 are topside, gas-lift, subsea and subsea gas-lift chokes respectively,[%]. Inputs and outputs for the 2nd case could be identified as:

$$y \in \Re := \{ y_2 \qquad (4\text{-}7)$$

$$u \in \Re^4 := \begin{cases} u_1 \\ u_2 \\ u_3 \\ u_4 \end{cases} \qquad (4\text{-}8)$$

Where y_2 - pressure in the bottom of riser [*bar*]. Samples were collected during 3600 sec with the model speed in K-Spice 25-30 sec to the real-time.

4.2.1. Open-loop simulation

To perform the open-loop simulation for gas-lift choke, the input signal was changed from 7% to 14.5% as it was shown at *Figure 4-10*.

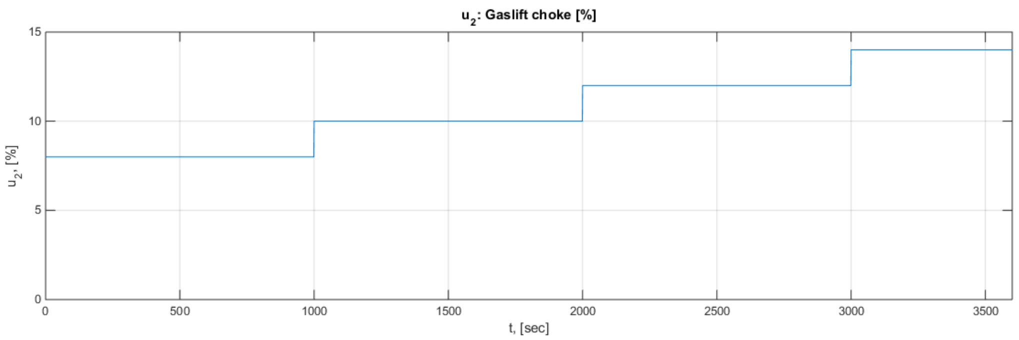


Figure 4-10 Inputs series for gas-lift choke manipulation

Figure 4-11 represents output signals changing in conformity with the changing of the input signal as in *Figure 4-10*.

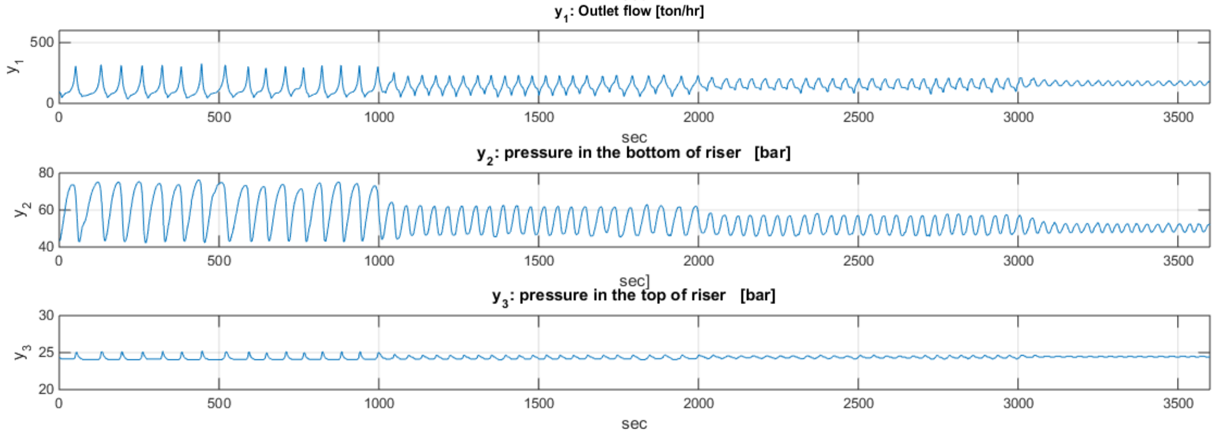


Figure 4-11 Open-loop simulation result of 2nd strategy

According to the obtained results, which are shown at *Figure 4-11* the outlet flow become more stable when control signal u_2=14.5%. At the same moment oscillation of pressure in the riser are significantly decreased compare with u_2=12%. The pressure in the top of the riser has small oscillations during the simulation but when u_2=14.5% pressure become stable. The outlet flow oscillate in range approximately 148 ton/hr.

For collecting a data set from open-loop simulation, the input signal u_2 was established by PBRS with the low amplitudes so the signal changes was from 13.6% to 14.4% and T_{min}=30 sec and T_{max}=120 sec. PRBS simulation results are shown at *Figure 4-12*.

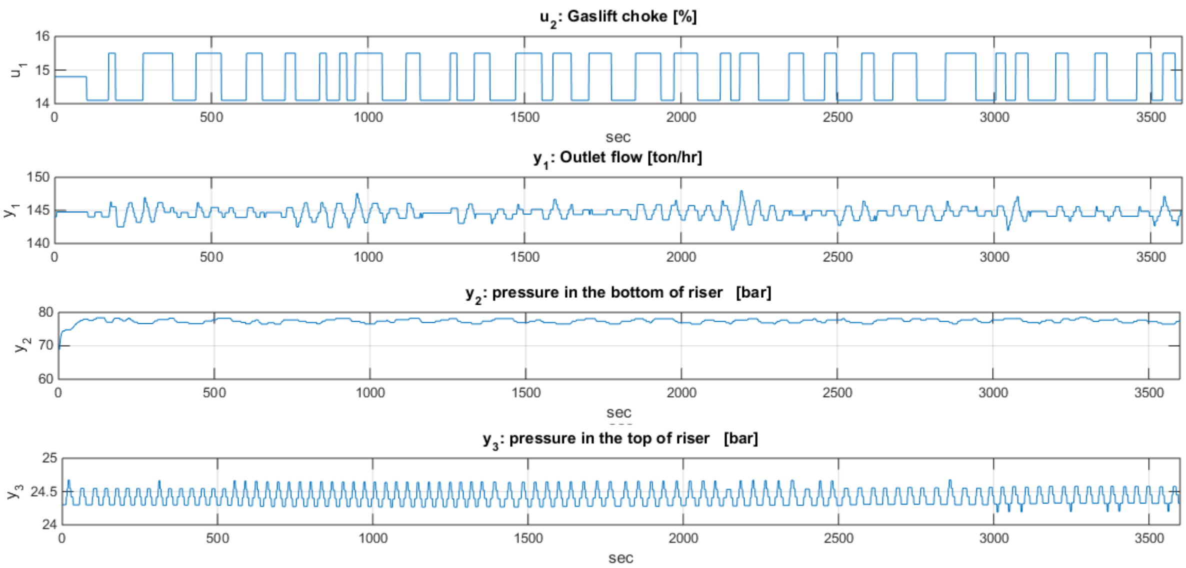


Figure 4-12 PBRS simulation result of 2nd strategy

All data were saved in "gasliftO.mat" file and were used in a system identification.

4.2.2. System identification using DSR, PEM and N4SID

All data received from PBRS experiment where centered and scaled for more occur model identification. It was done by adding the mean value of input and output measurements to the trended data. In the *Figure 4-13* is shown a new PRBS trend with new-scaled data, which are relevant to the results from *Figure 4-12*.

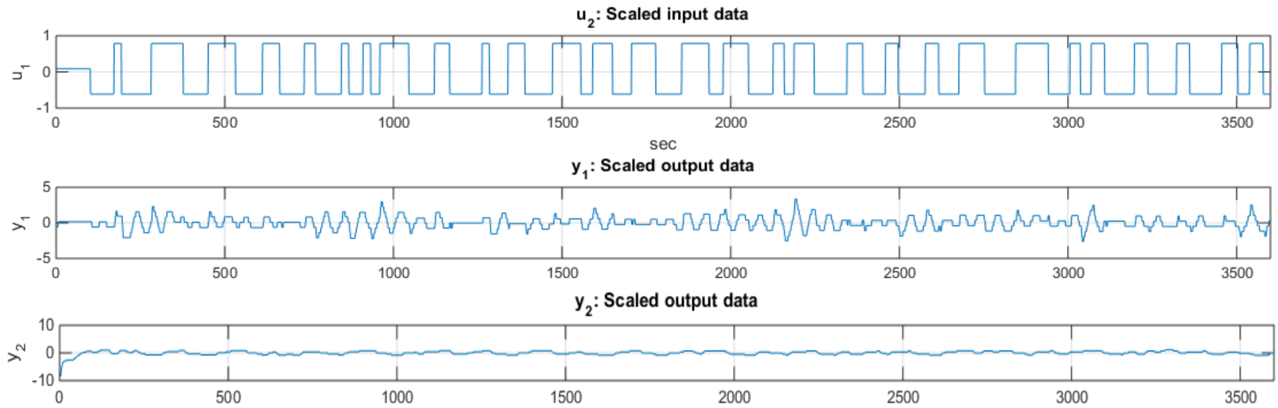


Figure 4-13 Scaled data of PBRS simulation result (2nd strategy)

For the 1st case DSR was used with L=39, J=39 and n=6 and data set was divided into 75% for identification and 25% for validation. It was done experiment when 100% of data set was used for identification. Characteristics of the model are available in Table 4-3. The outlet flow from the model and outlet flow from the process are shown at *Figure 4-14*.

The state matrices of discrete state-space model using DSR algorithm for the minimum phase, which was given in eq. (3-5) and (3-6)

$$A = \begin{bmatrix} 0.9794 & -0.1403 & -0.0991 & 0.0827 & -0.0489 & 0.0731 \\ 0.0957 & 0.9885 & -0.0176 & 0.0962 & 0.0031 & 0.1383 \\ 0.0093 & -0.0097 & 0.9747 & 0.1856 & 0.0074 & 0.2839 \\ -0.0027 & 0.0060 & 0.0182 & 0.8031 & 0.3978 & -0.3979 \\ 0.0022 & -0.0021 & -0.0067 & -0.2442 & 0.9314 & 0.3287 \\ 0.0024 & -0.0032 & -0.0147 & 0.1095 & -0.0384 & 0.5848 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.3111 \\ -0.2062 \\ -0.1231 \\ 0.1490 \\ 0.1987 \\ -0.2318 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.2015 & -0.1481 & -0.3242 & 0.4923 & -0.1295 & 0.5686 \end{bmatrix}$$

$$E = 0$$

PEM identification model preformed the best results with system order n=6. All characteristics of PEM model are given in Table 4-3. *Figure 4-14* represents real process output plot against PEM model output plot. Discrete state- space model using PEM that was described in the chapter State Space Model and state matrices *A,B,D,E* were identified as:

$$A = \begin{bmatrix} 0.9459 & -0.1456 & 0.0871 & -0.0010 & 0.03764 & 0.004337 \\ 0.07853 & 0.9682 & 0.1301 & -0.0537 & 0.01396 & 0.04114 \\ 0.1655 & 0.07122 & 0.3281 & -0.4399 & 0.3573 & -0.1738 \\ 0.04262 & -0.02829 & 0.1231 & -0.5221 & -0.7387 & 0.3776 \\ -0.02828 & -0.01837 & 0.5961 & 0.4804 & -0.4117 & -0.439 \\ 0.03222 & 0.02305 & -0.1528 & -0.3131 & -0.1062 & -0.2683 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0001 \\ -0.0006 \\ 0.0043 \\ -0.00334 \\ 0.0001 \\ 0.01472 \end{bmatrix}$$

$$D = \begin{bmatrix} 256.0086 & -24.7896 & 0.8788 & 12.1003 & 2.1641 & 1.0079 \end{bmatrix}$$

$$E = 0$$

N4SID system identification method showed the best performance with system order n=6. The result of output signal that was obtained by N4SID system identification is shown at *Figure 4-14* together with original process output. PEM model characteristics are shown in the Table 4-3.A,B,D,E matrices of state-space model that was discussed in State Space Model was identified as:

$$A=\begin{bmatrix} 0.9459 & -0.1456 & 0.0872 & -0.0010 & 0.0376 & 0.0043 \\ 0.0785 & 0.9682 & 0.1301 & -0.0537 & 0.0140 & 0.0411 \\ 0.1655 & 0.0712 & 0.3281 & -0.4399 & 0.3573 & -0.1738 \\ 0.0426 & -0.0283 & 0.1230 & -0.5221 & -0.7387 & 0.3776 \\ -0.0283 & -0.0184 & 0.5961 & 0.4804 & -0.4117 & -0.4390 \\ 0.0322 & 0.0231 & -0.1528 & -0.3131 & -0.1062 & -0.2683 \end{bmatrix}$$

$$B=\begin{bmatrix} 0.0001 \\ -0.0006 \\ 0.0043 \\ -0.0033 \\ 0.0001 \\ 0.0147 \end{bmatrix}$$

$$D=\begin{bmatrix} 291.1097 & -23.0191 & 7.2835 & -0.6325 & 5.6436 & -1.7796 \end{bmatrix}$$

$$E=0$$

Table 4-3 contains information about DSR, PEM and N4SID parameters.

Table 4-3 DSR, N4SID and PEM identification characteristics for the 2nd strategy (1st case)

Method	*n*	*L*	*J*	*MSE*	*Poles*	*Zeros*
DSR	6	39	39	0.3042	0.9818	0.9637
PEM	6	–	–	0.3251	0.4328	-1.3633
N4SID	5	–	–	0.3036	0.2432	-0.3118

Figure 4-14 represents all results from three identification methods DSR, PEM and N4SID and shows all models outputs against real process output in one graph.

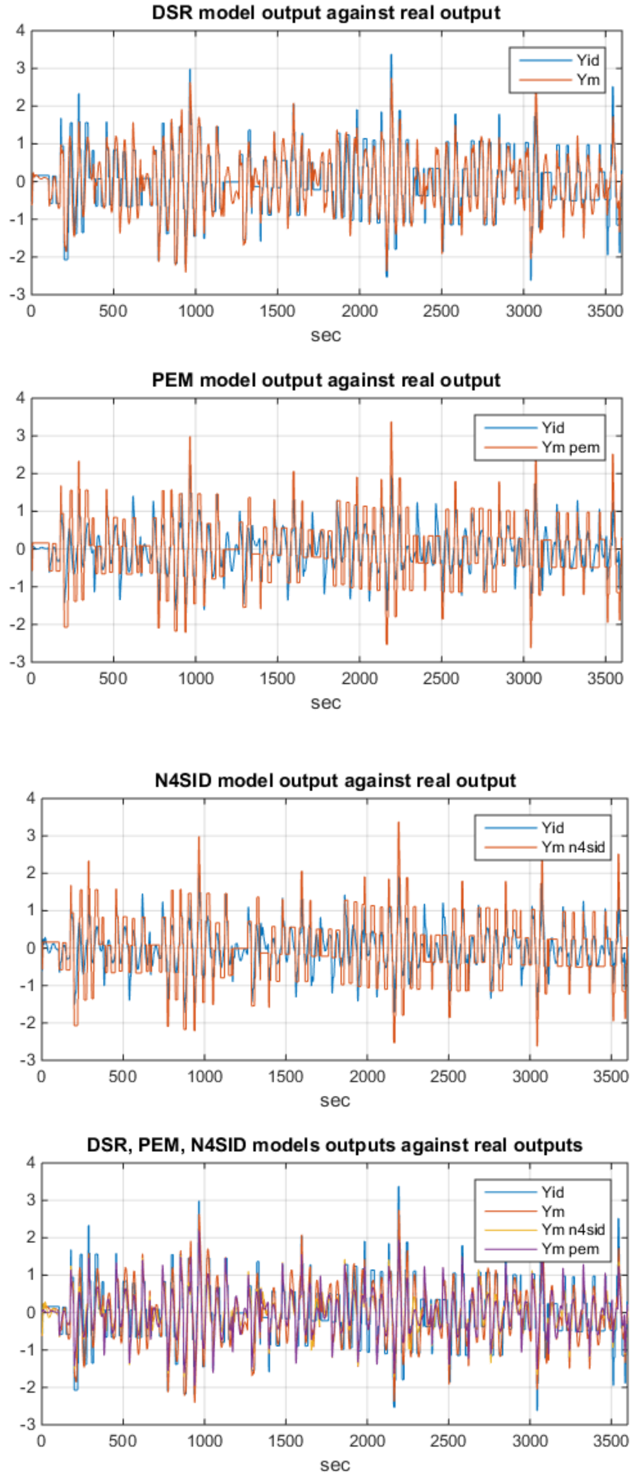


Figure 4-14 DSR, PEM and N4SID models outputs against outputs from the real process for the 2nd strategy (1st case)

The same experiments were done for the 2nd case where the output of the system was pressure in the riser. It was implemented the DSR method with L=29, J=31 and system order n=5. The simulation of the model with these parameter gave the best result with MSE= 0.1260. The model was checked on identification and validation and gave MSE of validation MSE=0.1474 and for identification MSE=

0.1346. Zero and pole of the obtained model are 0.9129 and 0.9438 respectively. The *A,B,D* and *E* matrices are:

$$A = \begin{bmatrix} 0.9874 & -0.1496 & -0.1842 & 0.0759 & -0.0995 \\ 0.1117 & 0.9836 & -0.0499 & 0.0690 & -0.0300 \\ 0.0105 & -0.0074 & 0.9590 & 0.2493 & -0.0435 \\ 0.0002 & 0.0048 & 0.0236 & 0.8033 & 0.4760 \\ 0.0008 & -0.0001 & 0.0079 & -0.2705 & 0.8711 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.3847 \\ -0.1762 \\ -0.2439 \\ 0.1353 \\ 0.2078 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.2329 & -0.1147 & -0.4306 & 0.4579 & -0.2222 \end{bmatrix}$$

$$E = 0$$

PEM method of model identification was implemented with the system order n=5. The model proved good results on identification and validation test and MSE=0.9536. The stability of the system was checked and showed that zero= 0.1495 and pole= 0.5681. A,B,D and E model matrices are:

$$A = \begin{bmatrix} 0.9657 & -0.1353 & 0.0776 & -0.0692 & 0.0171 \\ 0.0951 & 0.9562 & 0.1694 & -0.0350 & -0.0001 \\ -0.0365 & -0.0774 & 0.9404 & 0.7551 & -0.1828 \\ 0.0136 & 0.0674 & -0.2095 & 0.6282 & 0.1741 \\ 0.0004 & 0.0031 & 0.0018 & -0.0810 & -0.6496 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0035 \\ 0.0029 \\ 0.0165 \\ -0.0143 \\ 0.01108 \end{bmatrix}$$

$$D = \begin{bmatrix} 55.1837 & -2.6137 & -2.3561 & 1.6366 & -0.3963 \end{bmatrix}$$

$$E = 0$$

The experiment with N4SID method of identification represented the lowest MSE with system order *n*=5. MSE between process and model output are equal 3.21757. Conclusions about the stability of the system can be done based on zero= 0.3793 and pole= 0.7937. The model matrices A,B,D and E were identified as:

$$A = \begin{bmatrix} 0.9719 & -0.1506 & 0.0863 & -0.0691 & 0.0202 \\ 0.1019 & 0.9553 & 0.1907 & -0.0338 & 0.0003 \\ -0.0209 & -0.0631 & 0.8814 & 0.7716 & -0.1847 \\ 0.0507 & 0.0967 & -0.2353 & 0.5801 & 0.1720 \\ 0.0048 & 0.0054 & -0.0100 & -0.0761 & -0.7771 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0031 \\ 0.0012 \\ 0.0211 \\ -0.0166 \\ -0.0010 \end{bmatrix}$$

$$D = \begin{bmatrix} 55.1837 & -2.6137 & -2.3562 & 1.6366 & -0.3963 \end{bmatrix}$$

$$E = 0$$

Figure 4-15 shows graphs of process output together with model outputs, which were obtained by implementing different system identification methods.

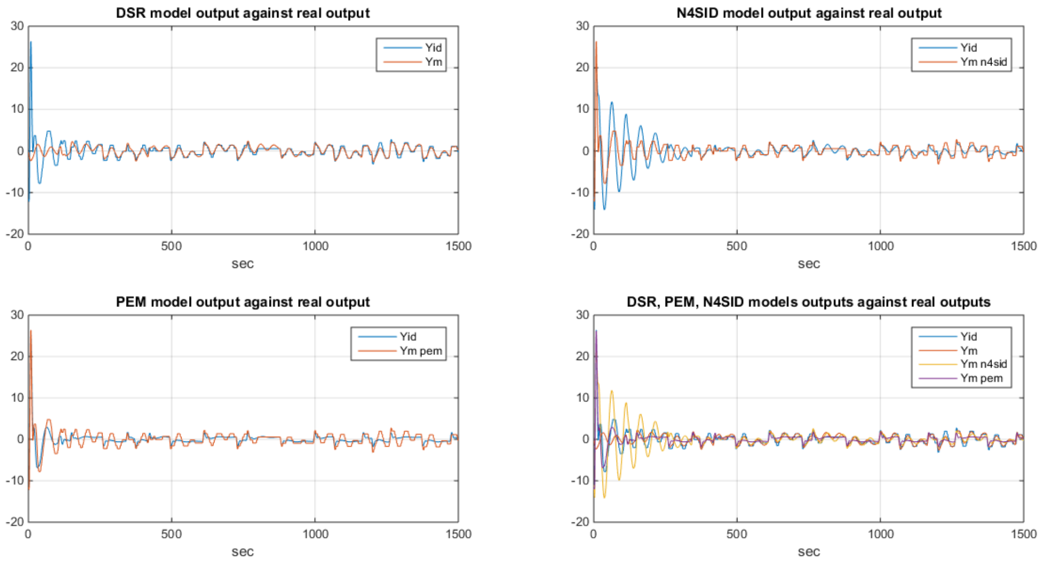


Figure 4-15 DSR, PEM and N4SID models outputs against outputs from the real process for the 2nd strategy (2nd case)

Table 4-4 summarizes all characteristics from DSR, PEM and N4SID algorithms, which was mention before.

Table 4-4 DSR, N4SID and PEM identification characteristics for the 2nd strategy (2nd case)

Method	*n*	*L*	*J*	*MSE*	*Poles*	*Zeros*
DSR	5	29	31	0.1260	0.9438	0.9129
PEM	5	–	–	0.9536	0.5681	0.1495
N4SID	5	–	–	3.21757	0.7937	0.3793

4.2.3. Implementation of LQ-regulator

Linear-quadratic regulator was implemented into the system with Q=1 and P=1000. The simulation was done during 6000 sec. *Figure 4-16* shows the simulation result, where first 500 sec simulation was done without LQR controller and as starting signals were chosen bifurcation point from open-loop siulation. The IAE of the simulation is equal to 251. As it can be clearly seen from the graphics, implementation of LQR remove the oscillation from the outlet flow and tracked the pressure signal. The feedback matrices G_1 and G_2 are:

$$G_1 = \left[0.0233 \quad -0.0115 \quad -0.0087 \quad -0.0749 \quad -0.0205 \right]$$

$$G_2 = \left[-0.0123 \right]$$

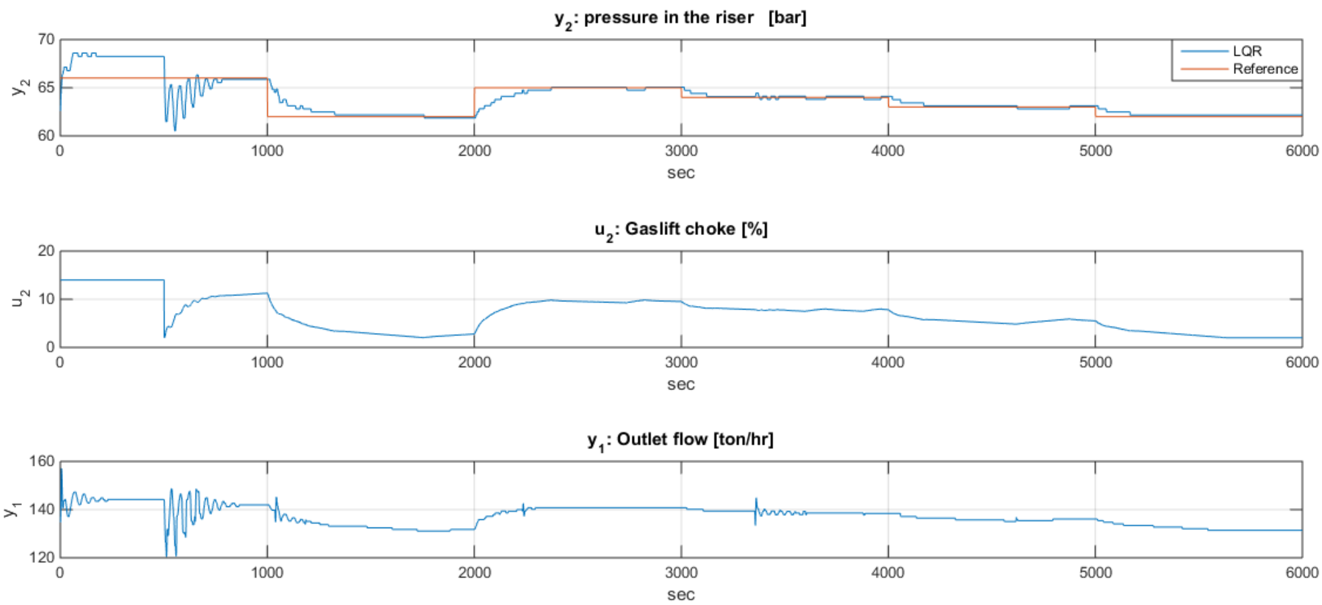


Figure 4-16 LQR implementation into the 2nd strategy

4.2.4. Implementation of PI-controller

PI-controller was tuned by using tuner application in MATLAB. Controller parameters, which were received using this application are Kp=11.398 and Ti=53.037. Controller did not work during first 500 sec. The reference signal was the same as in simulation with LQR controller that was discussed above. The IAE of the simulation is equal to 2.12e+3.

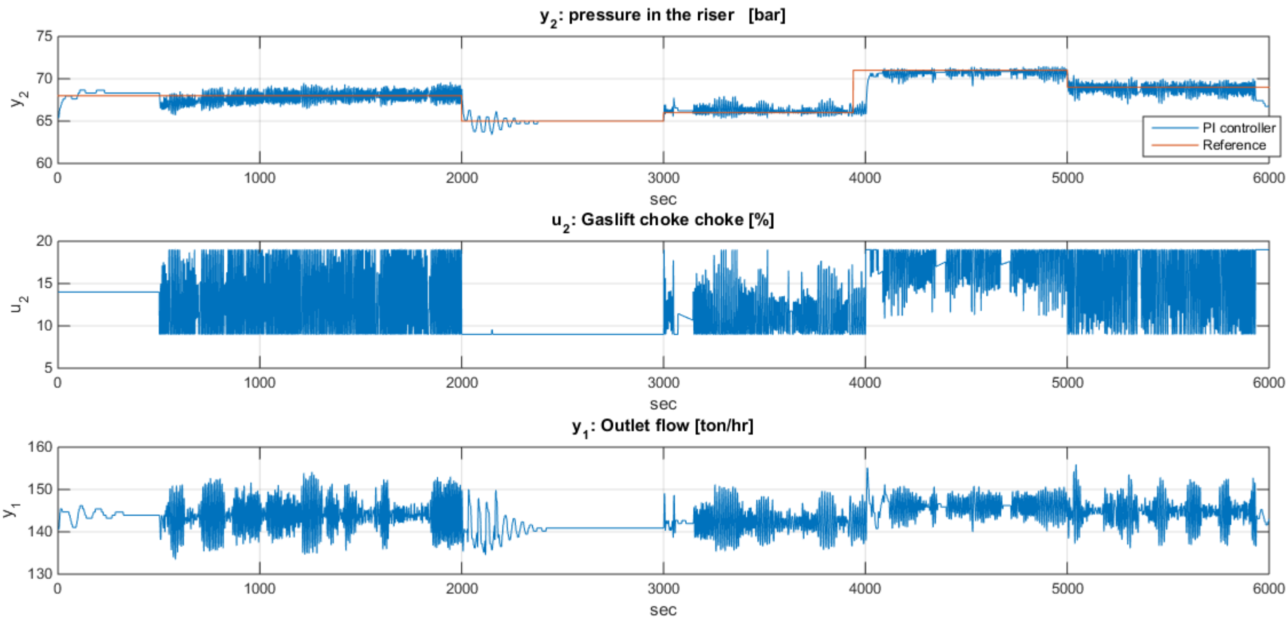


Figure 4-17 PI-controller implementation in the 2nd strategy

4.3. 3rd strategy: Subsea choke

This strategy represents the possibility to stabilize flow by controlling the subsea choke, while topside choke is stable and both gaslift chokes are closed, which mean that u_2 and u_4 equal to 0. According to this strategy, the input signals can be set in the following way:

$$u \in \Re^2 := \begin{cases} u_1 \\ u_3 \end{cases} \tag{4-9}$$

As it was mentioned before, there are two cases to stabilize slug flow. The 1st case is stabilizing flow by control the outlet flow y_1 and 2nd case is stabilizing flow by control the pressure in the bottom of riser y_2:

$$y \in \Re^3 := \begin{cases} y_1 \\ y_2 \end{cases} \tag{4-10}$$

Where output signals were collected into $y \in \Re^3 : y_1$ - outlet flow,[*ton/hr*]; y_2 - pressure in the bottom of riser [*bar*] and inputs $u \in \Re^2 : u_1, and\ u_3$ are topside and subsea chokes respectively,[%].

The duration of each simulation is 3600 sec and simulation speed in K-Spice is 25-30 to the real-time.

4.3.1. Open-loop simulation

Figure 4-18 shows changing of the input signal to the subsea choke. The control signal was changed from 10% to 90%.

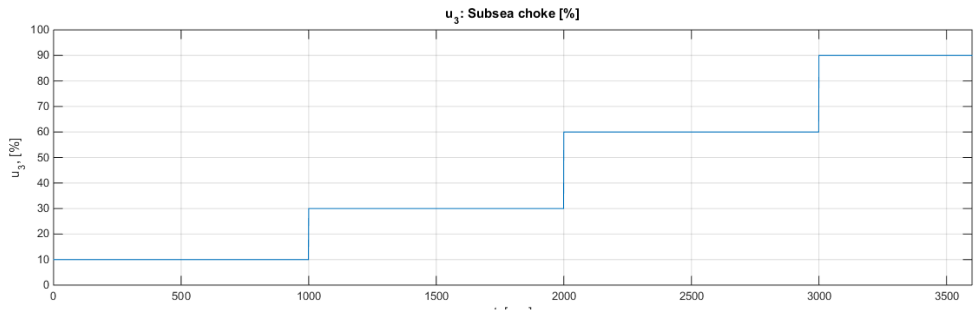


Figure 4-18 Step input changing to the subsea choke

Figure 4-19 describes the behavior of the output signals in the system according to the changes in the input signal. When the valve is closed in less than 30% then the flow does not enter into the riser. However, when the valve is open on 30% or more the flow oscillated from 1.78 ton/hr to 520 ton/hr and pressure in the bottom of riser changed from 48 bar to 100 bar. The pressure in the top of the riser also has significant overshooting.

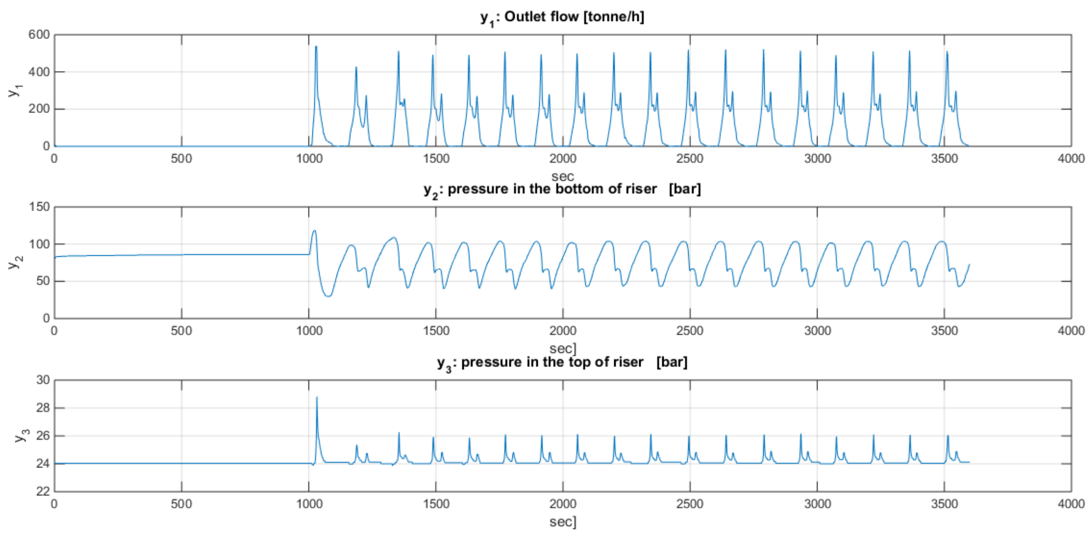


Figure 4-19 Open-loop simulation result of 3rd control strategy

According to the given results the bifurcation point was chosen as 10% of the choke opening. For collecting samples was done PBRS experiment. The input signal varied from 27% to 33%, as it shown at *Figure 4-20*, with the T_{min}=30 sec and T_{max}=120 sec.

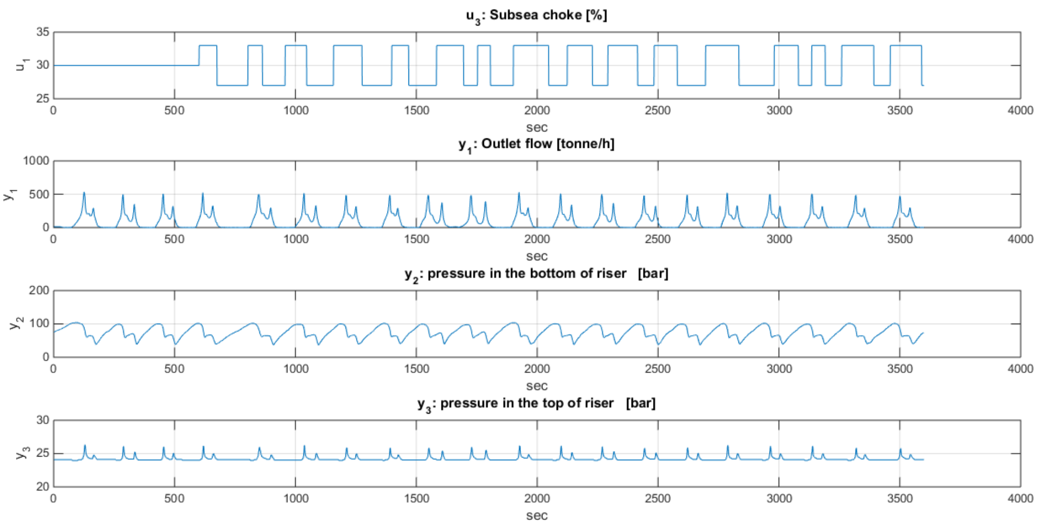


Figure 4-20 PBRS simulation result of 3rd control strategy

During this simulation was collected 3600 samples of input signal u_3 and two output signals y_1 and y_2, which was saved in "subseaO.mat".

4.3.2. System identification using DSR, PEM and N4SID

The data that have been saved in the file "subseaO.mat" were centered and scaled to obtain a more accurate identification model. *Figure 4-21* shows centered and scaled output data of outlet stream and the pressure at the bottom of the riser. The input signal u_3 was also scaled and centered.

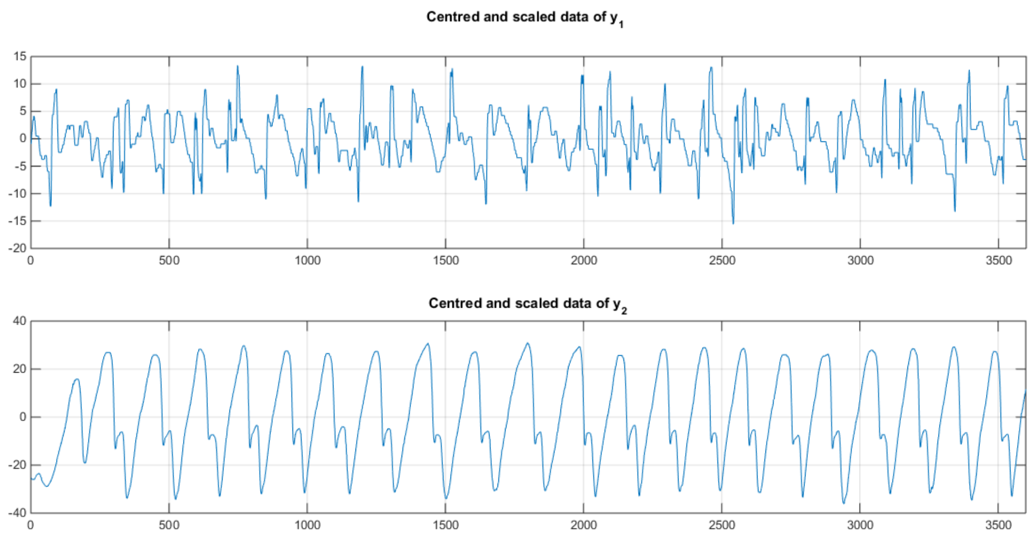


Figure 4-21 Scaled and centered data of PBRS simulation result (3rd strategy)

For the first of the case, where the output signal is outlet flow, it was implemented DSR algorithm with the next parameters *L=59, J=59* and *n=7*. *Figure 4-22* shows output from the real process together with output from obtained identification model. The data set during simulation was divided into two sets validation and identification, each of them holds 25% and 75% of data, respectively. The MSE for validation set is 14.142 and for identification 15.063. Poles and zeros have been identified to determine the stability of the system.

$$A = \begin{bmatrix} 0.9826 & -0.0763 & 0.0237 & 0.0294 & 0.0521 & -0.0336 & 0.0759 \\ 0.0371 & 0.9718 & 0.1014 & 0.0118 & 0.0504 & -0.0654 & 0.0751 \\ 0.0349 & -0.0293 & 0.9504 & 0.0634 & -0.1077 & 0.1499 & -0.0656 \\ -0.0148 & 0.0064 & -0.0512 & 0.9967 & 0.1571 & 0.1636 & 0.0567 \\ -0.0192 & -0.0174 & 0.0709 & -0.1613 & 0.9634 & 0.1884 & -0.0040 \\ -0.0025 & 0.0092 & -0.0261 & -0.0739 & -0.1316 & 0.9526 & 0.2362 \\ -0.0077 & -0.0037 & 0.0033 & -0.0031 & -0.0076 & -0.0609 & 0.9238 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.1353 \\ 0.0753 \\ 0.6899 \\ 0.6788 \\ -0.3469 \\ -0.7310 \\ -0.2597 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.1894 & -0.2228 & 0.2874 & 0.0405 & 0.1297 & -0.3157 & 0.2633 \end{bmatrix}$$

$$E = 0$$

Using N4SID algorithm it was identified model with the system order *n=4*. This system order provide the minimum MSE error, which is equal to 14.9286. According to the *Figure 4-22* the outlet flow from the process y_1, which is mentioned in the figure as *Yid* looks almost the same as outlet flow from the model *Ym*. The systems matrices *A,B,C,D,* which was described in the chapter State space model were identified as:

$$A = \begin{bmatrix} 0.9663 & -0.2479 & -0.0132 & 0.0154 \\ 0.1526 & 0.5900 & 0.3804 & -0.6219 \\ -0.0021 & -0.0278 & 0.9462 & 0.4790 \\ 0.0044 & 0.2097 & -0.1144 & 0.5350 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0004 \\ -0.0018 \\ 0.0008 \\ -0.0011 \end{bmatrix}$$

$$D = \begin{bmatrix} 257.2162 & -24.5912 & -5.7273 & 11.1787 \end{bmatrix}$$

$$E = 0$$

N4SID identification model for the process system is unstable, because zero is 1.0356. Full characteristic of N4SID identification model are given in Table 4-5.

The next method of system identification, which was implemented is PEM. After testing the model, it was found that system order *n=7* provides minimal MSE=14.5198 but, unfortunately, system is unstable with zero=-0.1412. All systems parameter are available in Table 4-5. The state matrices were identified as:

$$A = \begin{bmatrix} 0.9826 & -0.0763 & 0.0237 & 0.0294 & 0.0521 & -0.0336 & 0.0759 \\ 0.0371 & 0.9718 & 0.1014 & 0.0118 & 0.0504 & -0.0654 & 0.0751 \\ 0.0349 & -0.0293 & 0.9504 & 0.0634 & -0.1077 & 0.1499 & -0.0656 \\ -0.0148 & 0.0064 & -0.0512 & 0.9967 & 0.1571 & 0.1636 & 0.0567 \\ -0.0192 & -0.0174 & 0.0709 & -0.1613 & 0.9634 & 0.1884 & -0.0040 \\ -0.0025 & 0.0092 & -0.0261 & -0.0739 & -0.1316 & 0.9526 & 0.2362 \\ -0.0077 & -0.0037 & 0.0033 & -0.0031 & -0.0076 & -0.0609 & 0.9238 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.1353 \\ 0.0753 \\ 0.6899 \\ 0.6788 \\ -0.3469 \\ -0.7310 \\ -0.2597 \end{bmatrix}$$

$$D = [\ -0.1894 \quad -0.2228 \quad 0.2874 \quad 0.0405 \quad 0.1297 \quad -0.3157 \quad 0.2633]$$

$$E = 0$$

Figure 4-22 represents all three models identification methods, which were implemented to this process with outlet flow as output signal and subsea choke as input signal.

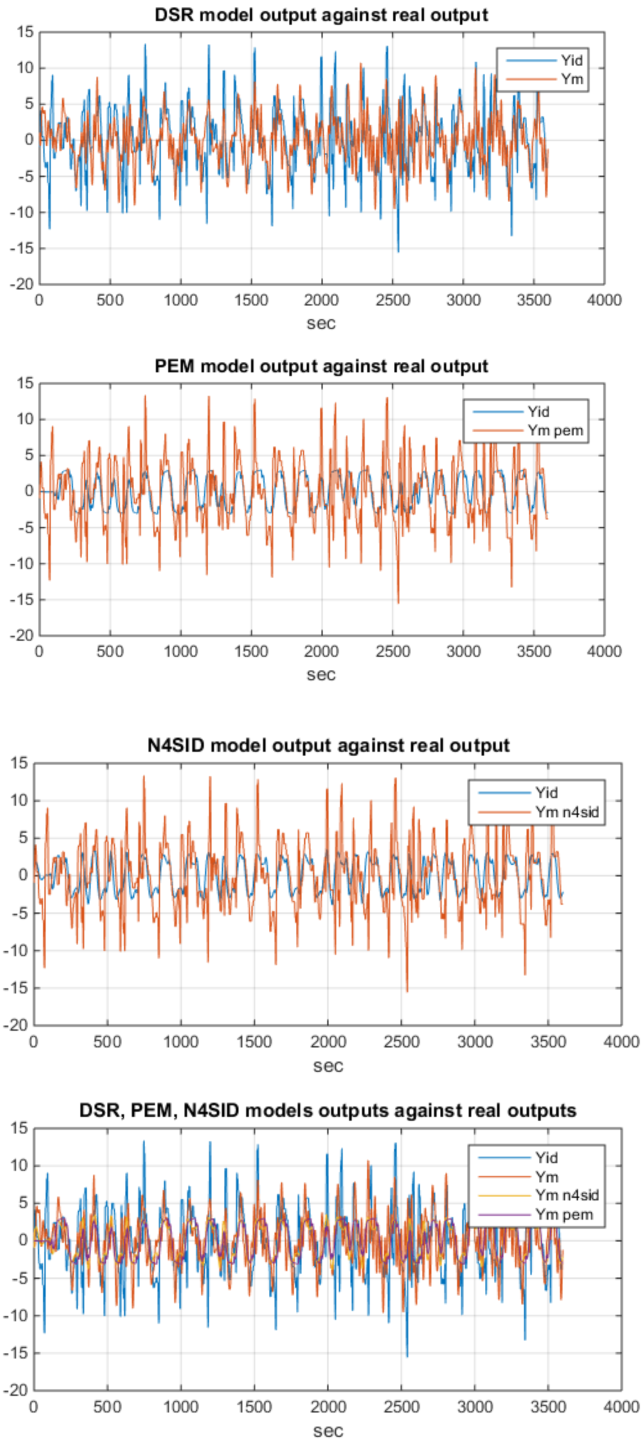


Figure 4-22 DSR, PEM and N4SID models outputs against outputs from the real process for the 3rd strategy (1st case)

Table 4-5 contains characteristics from DSR, N4SID and PEM identification models, which are system order, MSE, poles and zeros.

Table 4-5 DSR, N4SID and PEM identification characteristics for the 3rd strategy (1st case)

Method	*n*	*L*	*J*	*MSE*	*Poles*	*Zeros*
DSR	7	59	59	15.063	0.5659	0.8472
PEM	7	–	–	14.5198	0.3918	-0.142
N4SID	4	–	–	14.9286	0.7594	1.0356

In the second case for model identification was used pressure in the riser as output signal and subsea choke position as input signal. The first identification method which was implement was DSR, with *L=56, J=60* and system order *n=8.* Pole and zero of the model are 0.9580 and 0.9540 respectively, which means that system is stable. Table 4-6 shows all properties of the model. The state matrices A,B,D and E are:

$$A = \begin{bmatrix} 0.9806 & -0.0590 & -0.0555 & -0.0462 & 0.0602 & -0.0419 & 0.0705 & -0.0269 \\ 0.0333 & 0.9939 & -0.0703 & -0.0227 & 0.0501 & -0.0224 & 0.0771 & -0.0131 \\ 0.0138 & 0.0168 & 0.9964 & -0.1645 & 0.0290 & -0.0883 & 0.0114 & -0.0489 \\ -0.0025 & 0.0071 & 0.0560 & 0.9693 & 0.1544 & -0.0388 & 0.1708 & -0.0349 \\ -0.0060 & 0.0010 & 0.0042 & -0.0428 & 0.9369 & 0.2594 & -0.0584 & 0.0640 \\ -0.0017 & 0.0012 & 0.0046 & -0.0085 & -0.1584 & 0.9323 & 0.2459 & -0.0540 \\ -0.0014 & -0.0009 & -0.0001 & -0.0030 & -0.0108 & -0.0860 & 0.9102 & 0.2190 \\ -0.0003 & 0.0001 & -0.0001 & 0.0021 & -0.0095 & 0.0120 & -0.1140 & 0.8726 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0861 \\ 0.1114 \\ -0.0395 \\ -0.0586 \\ -0.0314 \\ -0.0635 \\ -0.0351 \\ 0.0271 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.1941 & -0.1412 & -0.1948 & -0.2924 & 0.2729 & -0.2571 & 0.3282 & -0.1564 \end{bmatrix}$$

$E = 0$

Using PEM algorithm for model identification, the model was executed with system order *n=8* and preform the best result according to the MSE but at the same moment identified model is unstable due to zero= -0.0617870. The graphic of pressure from the process and pressure from the model are shown at *Figure 4-23*. The state matrices *A,B,C* and *D*:

$$A = \begin{bmatrix} 1.0449 & 0.0970 & -0.0319 & 0.0551 & 0.0000 & 0.0115 & -0.0271 & 0.0357 \\ -0.1064 & 1.0310 & 0.1447 & -0.2072 & -0.0503 & -0.0904 & 0.0561 & -0.2191 \\ 0.0137 & -0.1014 & 1.1024 & 0.0669 & 0.2901 & -0.0389 & 0.0858 & 0.4951 \\ 0.0088 & 0.0631 & -0.1219 & -0.6053 & 0.2807 & 0.4526 & -0.1188 & 0.4467 \\ -0.0010 & 0.0096 & 0.0410 & 0.1407 & -0.8285 & 0.5574 & 0.0677 & -0.0103 \\ 0.0090 & -0.0321 & 0.0008 & -0.2844 & -0.5447 & -0.5270 & 0.1071 & 0.4197 \\ 0.0051 & -0.0505 & 0.0353 & -0.4138 & 0.0068 & -0.0725 & -0.8214 & -0.0989 \\ -0.0031 & 0.0746 & -0.1310 & 0.3414 & -0.0575 & -0.3453 & -0.4121 & 0.3363 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0000 \\ 0.0002 \\ -0.0018 \\ 0.0049 \\ 0.0017 \\ -0.0037 \\ 0.0164 \\ 0.0027 \end{bmatrix}$$

$$D = \begin{bmatrix} -58.3131 & -1.4632 & -0.6448 & 0.8144 & 0.2796 & 0.5511 & -0.3099 & 1.0821 \end{bmatrix}$$

$E = 0$

N4SID method of identification preforms the best identification result with system order *n=5,* that provides the minimal error as it shown in the *Figure 4-23*. Full model description is shown in Table 4-6. The state matrices *A,B,D* and *E*:

$$A = \begin{bmatrix} 1.0449 & 0.0970 & -0.0319 & 0.0551 & 0.0000 & 0.0115 & -0.0271 & 0.0357 \\ -0.1064 & 1.0310 & 0.1447 & -0.2072 & -0.0503 & -0.0904 & 0.0561 & -0.2191 \\ 0.0137 & -0.1014 & 1.1024 & 0.0669 & 0.2901 & -0.0389 & 0.0858 & 0.4951 \\ 0.0088 & 0.0631 & -0.1219 & -0.6053 & 0.2807 & 0.4526 & -0.1188 & 0.4467 \\ -0.0010 & 0.0096 & 0.0410 & 0.1407 & -0.8285 & 0.5574 & 0.0677 & -0.0103 \\ 0.0090 & -0.0321 & 0.0008 & -0.2844 & -0.5447 & -0.5270 & 0.1071 & 0.4197 \\ 0.0051 & -0.0505 & 0.0353 & -0.4138 & 0.0068 & -0.0725 & -0.8214 & -0.0989 \\ -0.0031 & 0.0746 & -0.1310 & 0.3414 & -0.0575 & -0.3453 & -0.4121 & 0.3363 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0000 \\ 0.0002 \\ -0.0018 \\ 0.0049 \\ 0.0017 \\ -0.0037 \\ 0.0164 \\ 0.0027 \end{bmatrix}$$

$$D = \begin{bmatrix} -58.3131 & -1.4632 & -0.6448 & 0.8144 & 0.2796 & 0.5511 & -0.3099 & 1.0821 \end{bmatrix}$$

$$E = 0$$

Table 4-6 contains information about parameters of DSR, PEM and N4SID methods.

Table 4-6 DSR, N4SID and PEM identification characteristics for the 3rd strategy (2nd case)

Method	*n*	*L*	*J*	*MSE*	*Poles*	*Zeros*
DSR	8	56	60	0.3867	0.9580	0.9430
PEM	8	–	–	0.4324	0.0915	-0.0617
N4SID	4	–	–	0.5524	0.9395	0.7595

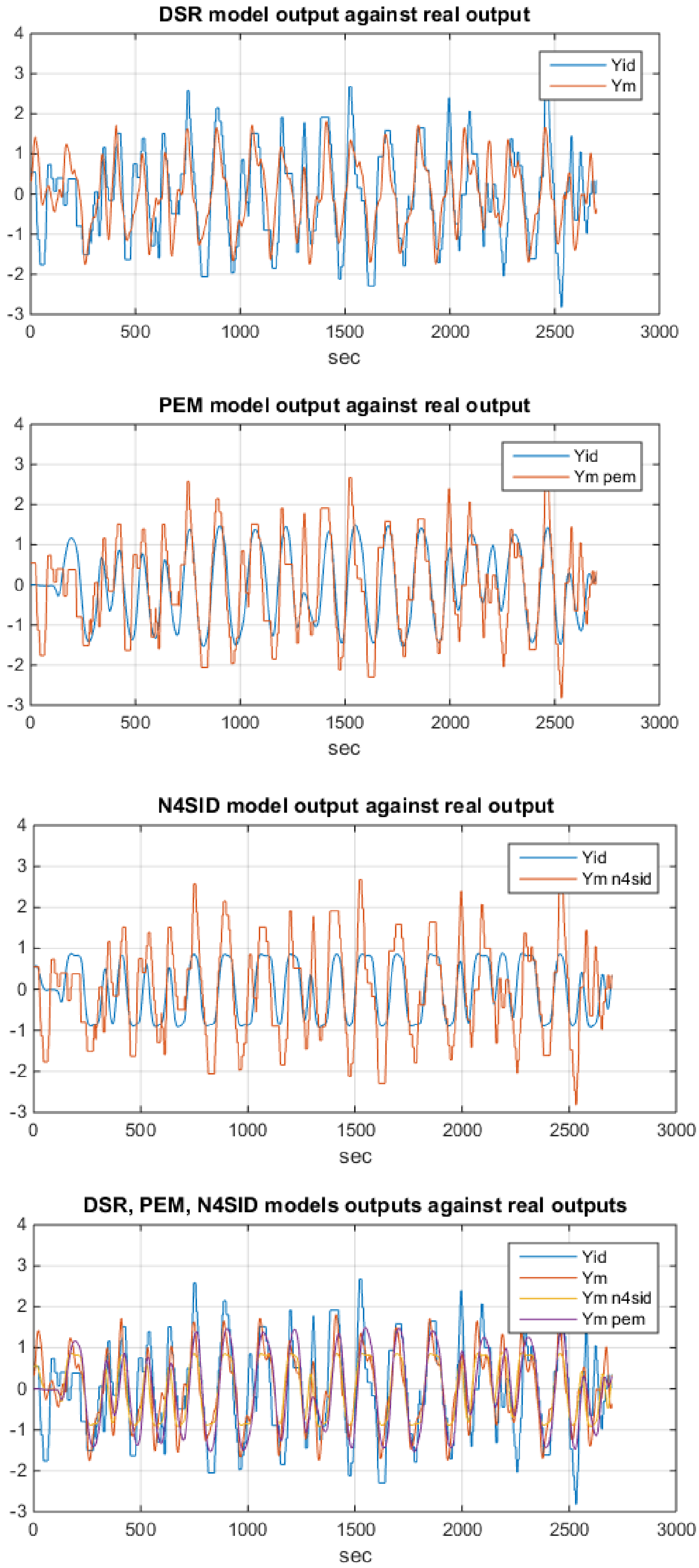


Figure 4-23 DSR, PEM and N4SID models outputs against outputs from the real process for the 3rd strategy (2nd case)

4.3.3. Implementation of LQ-regulator

Figure 4-24 shows the simulation results with LQR. Simulation time is equal to 6000 sec and weighting matrices Q=1 and P=1000. Feedback matrices G_1 and G_2 were obtained by using DSR model from the 2nd case.

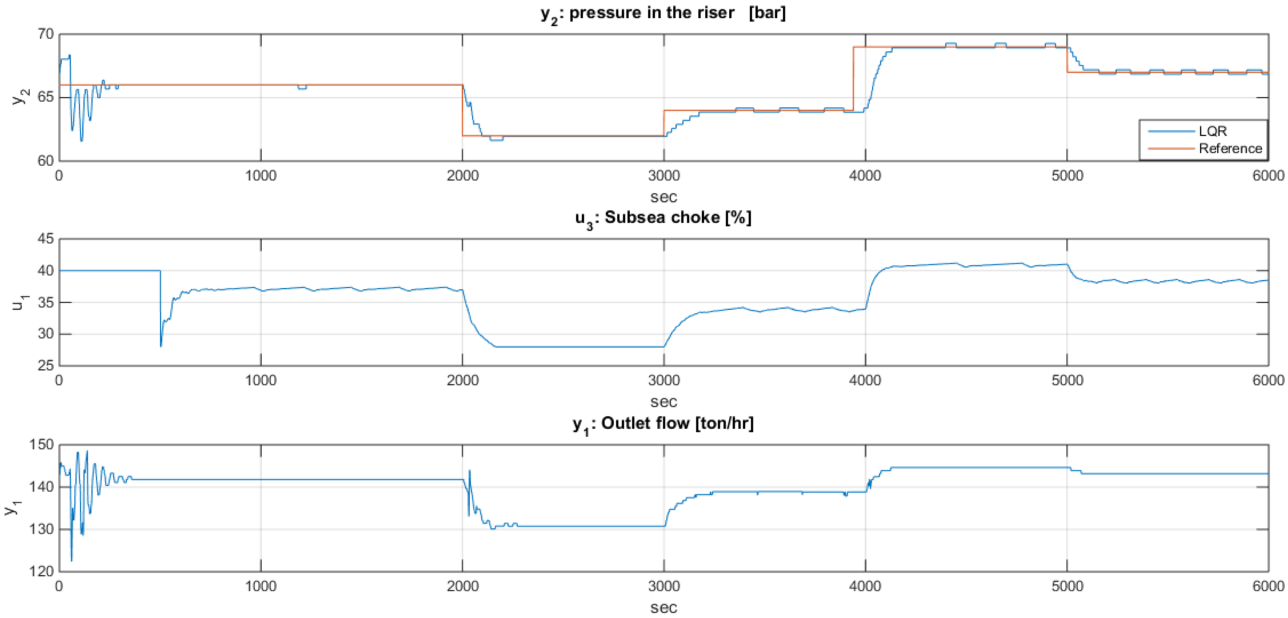


Figure 4-24 Simulation results of the system with LQR, 3rd control strategy

In the beginning of the simulation, control signal u_3 was equal to 40 and y_2=66 bar. After 500 sec LQR controller started to track the signal but, how it is possible to mention, with the small time delay. According to the graph of outlet flow, the flow is stable with not significant oscillations and IAE is equal to 516.The feedback matrices G_1 and G_2 are:

$$G_1 = [\ -0.1156 \quad 0.2346 \quad 0.0196 \quad -0.2578 \quad 0.2029 \quad -0.3221 \quad -0.0018 \quad -0.6468]$$

$$G_2 = [0.0313]$$

4.3.4. Implementation of PI-controller

Figure 4-25 shows simulation result with implementation of PI-controller. The duration of experiment was 6000 sec. During the experiment PI-controller were turn off first 500 sec. According to the received results the outlet flow was almost stabilized, but from 2000 sec to 3000 sec controller did not tracked the reference signal. PI controller was tuned by using MATLAB tuner application with parameters Kp=63 and Ti=141. During the experiment IAE was equal to IAE=2.31e+3.

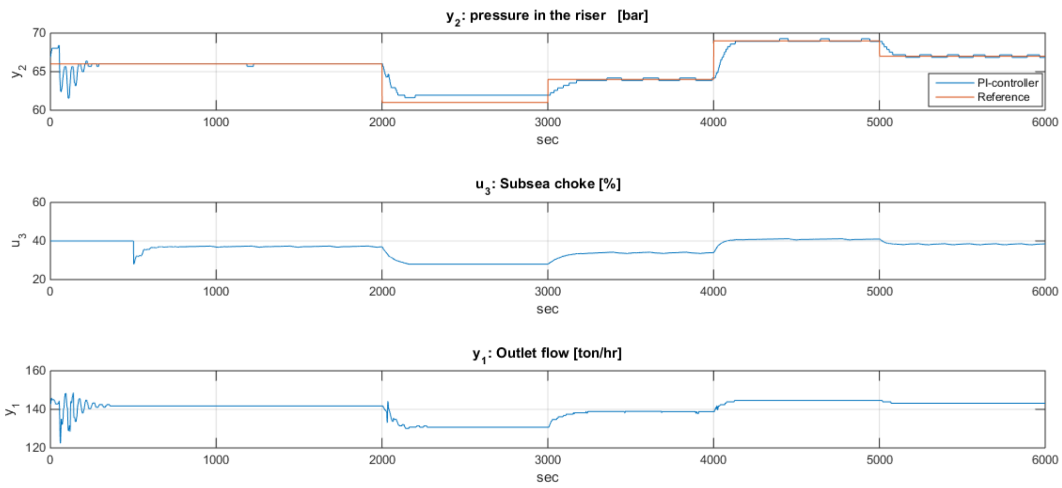


Figure 4-25 Simulation results of the system with PI-controller, 3rd control strategy

4.4. 4th strategy: Subsea gas-lift

The key idea of this strategy is to control only the subsea gaslift choke *FIC001GL* at the same moment topside choke *PIC0001* and subsea choke HIC-1 keep constant u_1=8% and u_3=50% . The gaslift choke *FIC001* is not in use so it is mean that u_2=0%. To stabilize the flow we divided this strategy on two cases. In 1st case, output control signals are measured values of outlet flow from flow transmitter *FT100* and in 2nd case measurement from pressure transmitter *PT006* are taken as output signals. Input and output parameters of the 1st case are:

$$u \in \Re^2 := \begin{cases} u_1 \\ u_3 \\ u_4 \end{cases} \tag{4-11}$$

$$y \in \Re^3 := \{ y_1 \tag{4-12}$$

were output signals were collected into $y \in \Re^3$: y_1 - outlet flow,[*ton/hr*], $u \in \Re^2$: u_1, u_3 and u_4 are topside, subsea and subsea gaslift chokes respectively,[%].

Input and output parameters of the 2st case are:

$$u \in \Re^2 := \begin{cases} u_1 \\ u_3 \\ u_4 \end{cases} \tag{4-13}$$

$$y \in \Re^3 := \{ y_2 \tag{4-14}$$

Where output signals were collected into $y \in \Re^3 : y_2$ - pressure in the bottom of riser *[bar].*

The duration of each simulation is 3600 sec with the model speed in K-Spice 25-30 sec to the real-time.

4.4.1. Open-loop simulation

During the open-loop simulation, input signal u_4 was changed from 2% to 13%, how it is shown at *Figure 4-26* . The open-loop simulation results are shown at *Figure 4-27*. When the control signal reach 13% the outlet flow became less oscillating around 140 ton/hr and pressure in the bottom of riser became more stable in value 61 bar. After this simulation is it possible to suggest bifurcation point is u_4*=13%.* The pressure in the top of the riser is stable and equal to 24.2 bar.

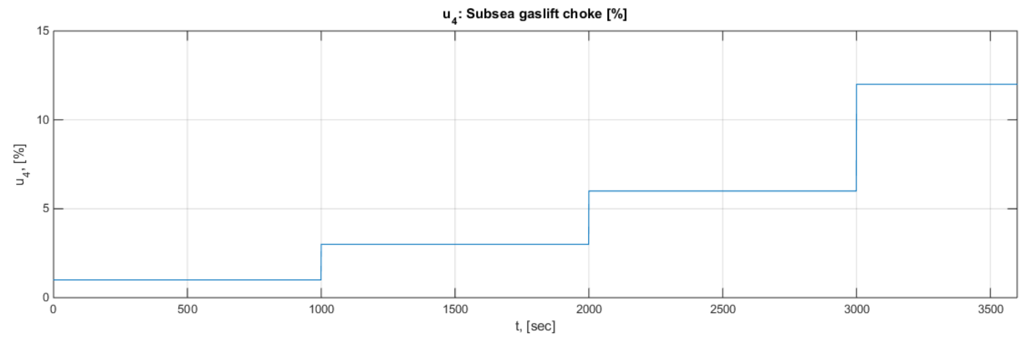


Figure 4-26 Input signal step changing of subsea-gaslift choke

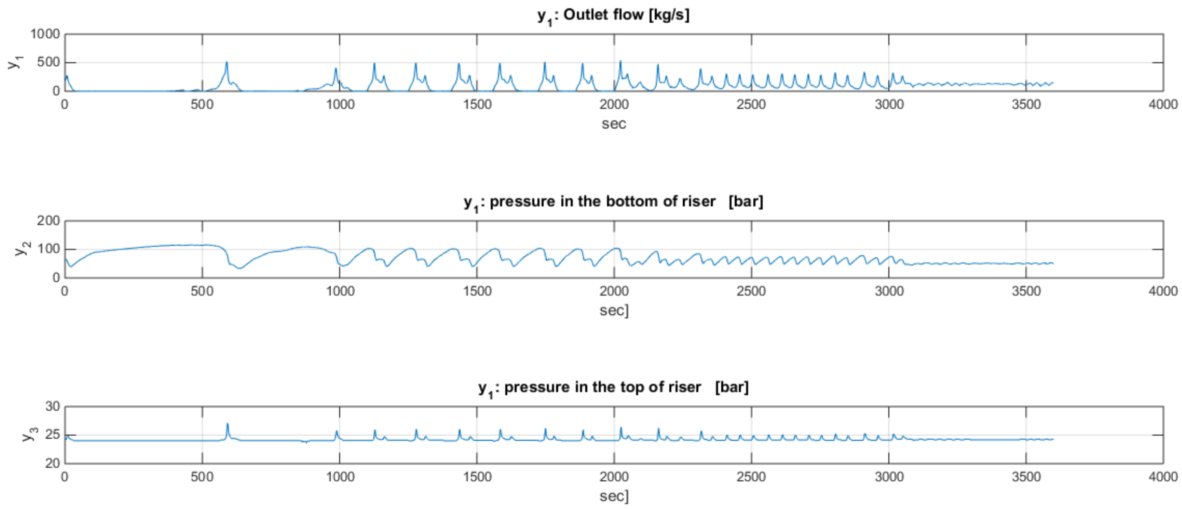


Figure 4-27 Open- loop simulation results of 4th strategy

The results from generating PRBS signal in MATLAB are shown at *Figure 4-28* while u_1 and u_3 are stable. The input signal changed in a range 12-14 %. This signal was generate with the "low frequency", where T_{min}*=20 sec and* T_{min}*=50 sec* and general time of samples collecting was 3600 sec.

Input signal u_4 and measured values of flow and pressure in the bottom of riser were saved in "subsea_gasliftO.mat" file.

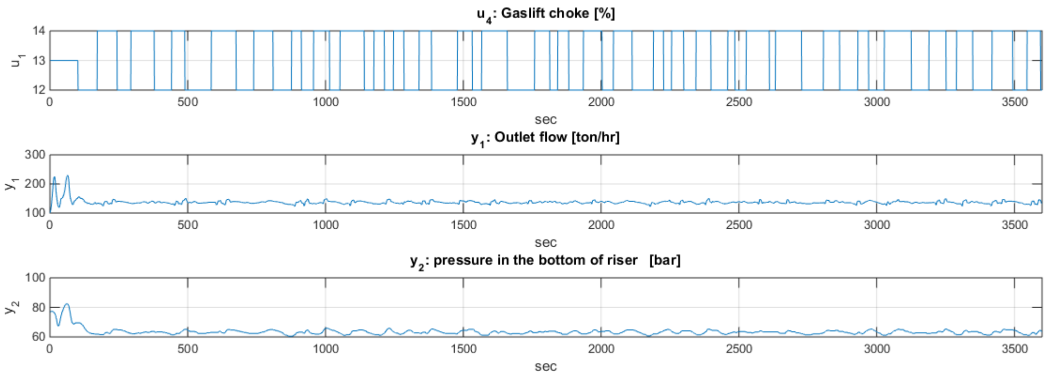
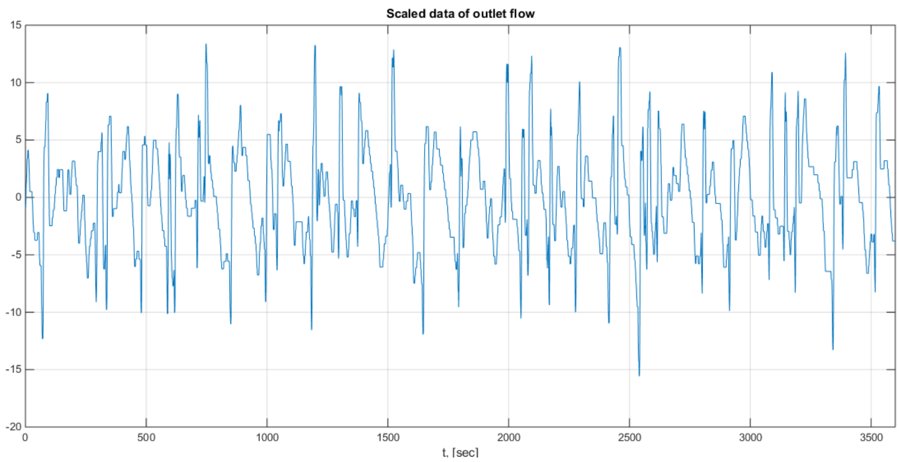


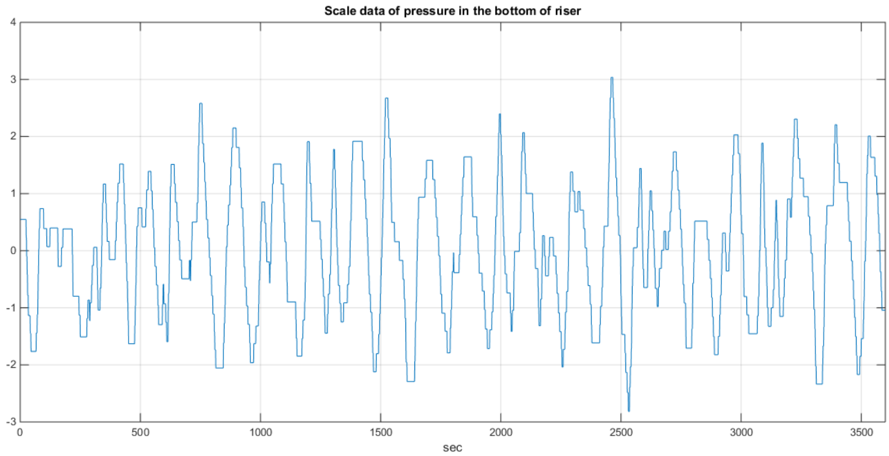
Figure 4-28 PRBS signal simulation results 4th strategy

4.4.2. System identification using DSR, PEM and N4SID

All collected data from the simulation were first centered and scaled for further work with model identification, how it is presented at *Figure 4-29*. The model identification for 1st and 2nd cases was done with using DSR, PEM and N4SID.



a)



b)

Figure 4-29 Centered and scaled output data from PRBS experiment

a) *y_1 - outlet flow; b) y_2 – pressure in the bottom of riser*

To find the best DSR parameters *L,J* and *n* was used MATLAB script that fulfils minimal MSE. For this propose, it was used MATLAB- script from Annex 6. The DSR testing was done, while *n* was changed in a range 3:10, *L* and *J* - in range 10: 60. The best achieved results with minimal error were L=59, J=59, n=7. The identified model preformed good validation and identification results, MSE from validation set is 12.4962 and for identification set MSE=14.1423. The results of model developing are given in the Table 4-7. The identified model matrices *A, B, D, E:*

$$A = \begin{bmatrix} 0.9623 & -0.0781 & 0.0668 & 0.0818 & 0.0955 & 0.0665 & -0.1067 \\ 0.0282 & 0.9810 & 0.0818 & 0.0595 & 0.0671 & 0.0529 & -0.0775 \\ 0.0192 & -0.0183 & 0.9607 & -0.1812 & -0.0349 & -0.1153 & 0.0846 \\ -0.0291 & -0.0259 & 0.0996 & 0.9617 & -0.1487 & -0.0735 & 0.1000 \\ -0.0081 & 0.0049 & 0.0034 & 0.0406 & 0.9822 & -0.2699 & 0.0441 \\ 0.0039 & -0.0064 & 0.0015 & 0.0165 & 0.1132 & 0.9153 & 0.2711 \\ 0.0060 & 0.0024 & -0.0094 & 0.0100 & 0.0048 & -0.0978 & 0.8531 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0572 \\ 0.3532 \\ 0.1991 \\ -0.1328 \\ 0.1850 \\ 0.3948 \\ 0.1823 \end{bmatrix}$$

$$D = [-0.2830 \quad -0.1932 \quad 0.2564 \quad 0.2140 \quad 0.2327 \quad 0.3227 \quad -0.3178]$$

$$E = 0$$

Figure 4-30 shows the real process output measurement together with identified DSR model output. PEM identification model preformed the best results with system order n=7. All characteristics of PEM model are given in Table 4-7. *Figure 4-30* represents real process output plot against PEM model output plot. Discrete state- space model using PEM that was described in the chapter Sate Space Model and state matrices *A,B,D,E* were identified as:

$$A = \begin{bmatrix} 0.8964 & -0.2485 & -0.0218 & 0.0217 & -0.0139 & -0.0094 & 0.0000 \\ 0.1617 & 0.5571 & -0.2557 & -0.6294 & -0.2441 & -0.1207 & -0.0120 \\ 0.0146 & -0.0128 & 0.9122 & -0.3893 & 0.0581 & -0.0685 & -0.1303 \\ 0.0013 & 0.1866 & 0.1022 & 0.5133 & -0.6443 & -0.1929 & -0.1375 \\ -0.0144 & 0.0778 & 0.0137 & 0.0067 & 0.6692 & -0.5093 & -0.3940 \\ 0.0108 & 0.0054 & -0.0199 & 0.1531 & -0.0876 & -0.9380 & 0.5691 \\ -0.0198 & -0.1390 & -0.0832 & -0.2421 & -0.3988 & -0.4242 & -0.3719 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0001 \\ -0.0006 \\ -0.0002 \\ 0.0023 \\ 0.0024 \\ 0.0039 \\ 0.0007 \end{bmatrix}$$

$$D = [256.0086 \quad -24.7896 \quad 0.8788 \quad 12.1003 \quad 2.1641 \quad 1.0079 \quad 0.1842]$$

$E = 0$

N4SID system identification method showed the best performance with system order n=3. The result of output signal that was obtained by N4SID system identification is shown at Figure 4-30 together with original process output. PEM model characteristics are shown in the Table 4-7. A,B,D,E matrices of state-space model that was discussed in State space model was identified as:

$$A = \begin{bmatrix} 0.9664 & -0.249 & -0.02837 \\ 0.1505 & 0.56 & 0.4948 \\ -0.001793 & -0.1368 & 0.8688 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0001593 \\ -0.0001415 \\ -0.0006182 \end{bmatrix}$$

$$D = \begin{bmatrix} 257.6 & -24.87 & -9.264 \end{bmatrix}$$

$E = 0$

Figure 4-30 preforms results from DSR, N4SID and PEM identification methods with output signal from the real process.

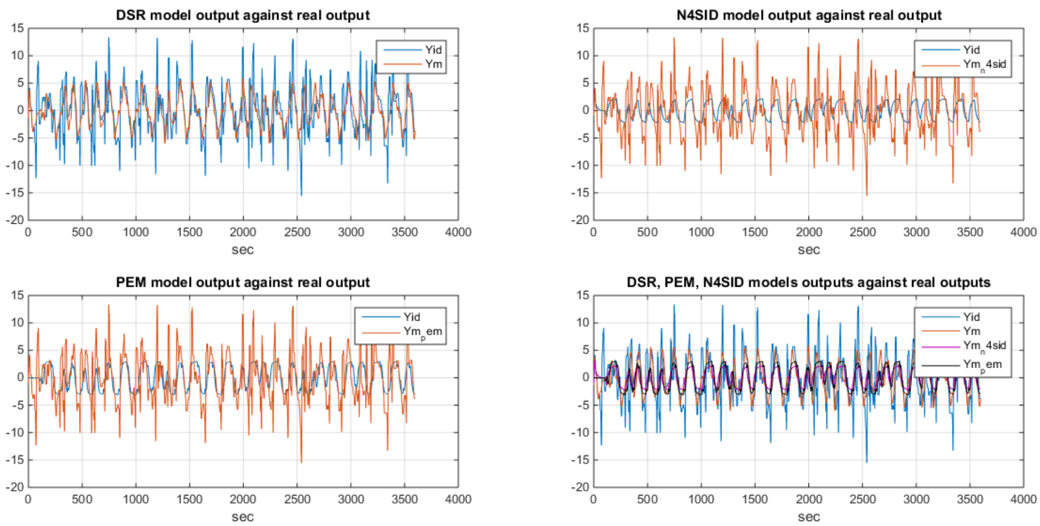


Figure 4-30 DSR, N4SID and PEM identification results of the 4th strategy (1st case)

Table 4-7 summarizes identification parameters from each methods. According to the Table 4-7, models, which was identified with DSR and N4SID, are stable and PEM model is unstable, because zeros value is -0.1412.

Table 4-7 DSR, N4SID and PEM identification characteristics for the 4th strategy (1st case)

Method	*n*	*L*	*J*	*MSE*	*Poles*	*Zeros*
DSR	10	51	51	14.1423	0.8929	0.7638
PEM	7	–	–	14.5198	0.3198	-0.1412
N4SID	3	–	–	15.8412	0.9558	0.8713

For the 2nd case the DSR model was identified with L=66, past horizon J =76 and system order n=10. The data set was divided into 25% for validation and 75% for identification sets. The output signal of the model together with the output signal of the process shown at *Figure 4-31* furthermore, the validation MSE is 9.1053 and identification MSE is 11.0063. Full DSR model characteristics are shown in the Table 4-8. State matrices *A, B, D, E*, which were identified by DSR algorithm are:

$$A=\begin{bmatrix} 0.9728 & -0.0670 & -0.0690 & 0.0291 & 0.0808 & 0.0492 & -0.0091 & 0.0100 & -0.0460 & 0.0456 \\ 0.0358 & 0.9889 & -0.0396 & 0.0426 & 0.0401 & 0.0411 & -0.0066 & 0.0090 & -0.0248 & 0.0333 \\ -0.0078 & -0.0117 & 0.9393 & 0.1490 & 0.1208 & 0.0891 & -0.0264 & 0.0095 & -0.0506 & 0.0928 \\ -0.0009 & -0.0241 & -0.0965 & 0.9786 & -0.1142 & -0.0280 & 0.0067 & -0.0065 & 0.0410 & -0.0270 \\ -0.0206 & -0.004 & -0.0408 & 0.0617 & 0.9461 & -0.1712 & 0.0020 & -0.0223 & 0.0330 & -0.1132 \\ 0.0050 & -0.0055 & -0.0061 & -0.0086 & 0.0861 & 0.9247 & 0.1117 & 0.0498 & 0.2327 & -0.0583 \\ 0.0043 & 0.0039 & 0.0142 & 0.0006 & 0.0069 & -0.1089 & 0.9760 & 0.1814 & -0.0342 & 0.0002 \\ 0.0010 & -0.0014 & -0.0025 & -0.0017 & 0.0081 & -0.0542 & -0.1800 & 0.9829 & 0.0395 & -0.0454 \\ 0.0023 & -0.0021 & -0.0012 & -0.0081 & 0.0056 & -0.1628 & 0.0027 & -0.0411 & 0.9213 & 0.2591 \\ 0.0001 & -0.0028 & -0.0068 & -0.0035 & 0.0216 & 0.0098 & -0.0050 & 0.0326 & -0.1653 & 0.9207 \end{bmatrix}$$

$$B=\begin{bmatrix} -0.0873 \\ 0.2403 \\ -0.0132 \\ 0.0580 \\ -0.1372 \\ 0.6285 \\ 0.3199 \\ 0.0958 \\ 0.1863 \\ 0.3564 \end{bmatrix}$$

$$D=\begin{bmatrix} -0.2325 & -0.1334 & -0.3167 & 0.1217 & 0.2505 & 0.2504 & -0.0334 & 0.0340 & -0.1652 & 0.2467 \end{bmatrix}$$

$$E=0$$

After testing PEM model with different system order *n,* the best result with minimal MSE was obtained with *n=10*. The detailed system characteristic from PEM developed model are given in the

Table 4-8. The graphical representation of output signal from identified model and process in shown at *Figure 4-31*.

The system matrices A,B,D and E were identified with PEM as:

$$A=\begin{bmatrix} 0.8513 & 0.3277 & -0.0829 & 0.0428 & -0.0448 & 0.1291 & -0.1060 & 0.0669 & -0.2336 & 0.0071 \\ -0.1194 & 0.8795 & 0.1168 & -0.0850 & 0.0740 & -0.1766 & 0.1498 & -0.0673 & 0.2506 & 0.0163 \\ 0.0218 & -0.0315 & 0.9687 & -0.2669 & -0.0123 & 0.1206 & -0.0254 & -0.0111 & -0.1234 & -0.0285 \\ 0.0181 & 0.0020 & 0.2591 & 0.9529 & -0.0593 & -0.1734 & 0.0332 & 0.0202 & 0.0664 & 0.0030 \\ 0.0002 & -0.0086 & 0.0189 & 0.0888 & 0.9703 & 0.2305 & 0.0132 & -0.0040 & -0.0356 & -0.0003 \\ -0.0300 & 0.0671 & 0.0013 & 0.1369 & -0.2050 & 0.8884 & -0.0020 & -0.1244 & 0.1553 & -0.0942 \\ 0.0074 & -0.0363 & -0.0313 & -0.0042 & -0.0481 & 0.1214 & 0.9247 & 0.2560 & -0.0629 & 0.0937 \\ -0.0061 & 0.0036 & 0.0151 & -0.0077 & -0.0007 & 0.0489 & -0.2294 & 0.9361 & 0.2603 & 0.0589 \\ 0.0326 & -0.0394 & -0.0100 & -0.0230 & -0.0098 & 0.0899 & -0.0878 & -0.1825 & 0.6783 & 0.4393 \\ -0.0209 & 0.0023 & 0.0142 & 0.0349 & -0.0177 & 0.0284 & -0.0525 & -0.0012 & -0.3419 & 0.8786 \end{bmatrix}$$

$$B=\begin{bmatrix} -1.6750 \\ -1.1955 \\ -1.7885 \\ 0.1863 \\ 1.8155 \\ -0.9229 \\ 1.5676 \\ 0.7415 \\ -0.2086 \\ 0.3664 \end{bmatrix}$$

$$D=\begin{bmatrix} -0.5075 & 0.3337 & -0.1230 & 0.1079 & -0.0855 & 0.2088 & -0.1994 & 0.1201 & -0.4072 & 0.0108 \end{bmatrix}$$

$$E=0$$

N4SID was used with the same system order *n=10* as PEM. System's parameters are available in Table 4-8 and output signal that were gotten from N4SID identification, together with the output signal from process are shown at *Figure 4-31*.

$$A=\begin{bmatrix} 0.9669 & -0.2370 & -0.0742 & 0.0103 & 0.0127 & -0.0082 & 0.0187 & -0.0187 & -0.0071 & 0.0147 \\ 0.1498 & 0.6852 & -0.2978 & -0.2976 & -0.5149 & -0.1184 & 0.0381 & -0.0761 & 0.0686 & -0.1865 \\ 0.0438 & -0.0718 & 0.8912 & -0.3178 & -0.3130 & -0.1155 & -0.0337 & -0.1075 & -0.1228 & -0.2894 \\ 0.0097 & 0.0195 & 0.0697 & 0.8945 & -0.4956 & 0.0086 & 0.1221 & 0.0186 & 0.2443 & 0.1154 \\ 0.0016 & 0.1758 & 0.1002 & 0.1383 & 0.4778 & -0.4657 & 0.2605 & -0.4366 & -0.3858 & -0.4451 \\ -0.0002 & -0.0398 & -0.0398 & 0.0588 & -0.0234 & -0.7279 & -0.6668 & 0.0703 & -0.0382 & 0.0116 \\ -5.62e-06 & 0.0278 & 0.0278 & 0.0429 & -0.0212 & 0.4302 & -0.6233 & -0.4477 & -0.6260 & 0.4422 \\ 0.0017 & 0.0094 & 0.0094 & 0.0226 & 0.1474 & 0.1353 & -0.1914 & -0.4432 & 0.9251 & 0.1801 \\ -0.0048 & -0.0242 & -0.0257 & -0.0387 & 0.1740 & 0.1164 & 0.2259 & -0.6214 & -0.9251 & 0.5523 \\ 0.0023 & 0.0298 & 0.01773 & -0.0195 & -0.0523 & 0.0337 & -0.0998 & 0.1111 & 0.0016 & 0.8102 \end{bmatrix}$$

$$B=\begin{bmatrix} -0.0002 \\ -0.0011 \\ 3.278e-05 \\ -0.0006 \\ 0.0016 \\ 0.0043 \\ 0.0008 \\ -0.0011 \\ 0.0018 \\ -0.0001 \end{bmatrix}$$

$$D=\begin{bmatrix} 254.6 & -24.27 & -5.117 & 6.039 & 9.912 & 1.33 & 1.376 & -0.4551 & -1.509 & 1.893 \end{bmatrix}$$

$$E=0$$

Figure 4-31 preforms comparing result of DSR, N4SID and PEM identification method with output signal which composed from pressure in the bottom of riser.

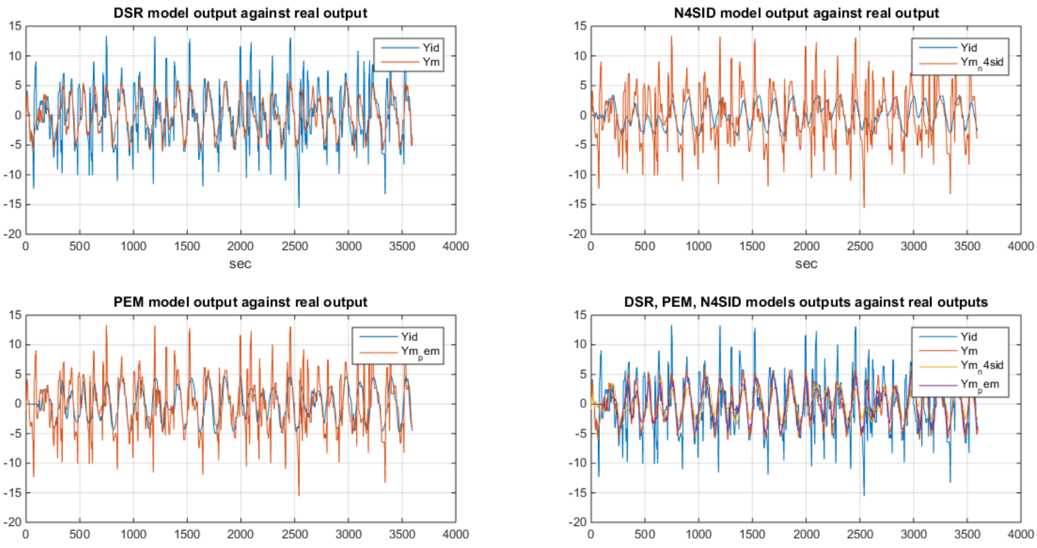


Figure 4-31 DSR, N4SID and PEM identification results of the 4th strategy (2nd case)

Table 4-8 shows DSR, N4SID and PEM identification parameters, which were mentioned above. According to the Table 4-8 it is clearly seen that all systems are stable, because poles and zeros are placed in diapason 0-1.

Table 4-8 DSR, N4SID and PEM identification characteristics for the 3rd strategy (2nd case)

Method	*n*	*L*	*J*	*MSE*	*Poles*	*Zeros*
DSR	10	66	76	11.7952	0.2759	0.3510
PEM	7	–	–	12.1668	0.8929	0.7638
N4SID	3	–	–	15.2626	0.3198	0.5365

4.4.3. Implementation of LQ-regulator

Figure 4-32 shows the simulation result of closed-loop system with LQR. The weighting matrices Q and P were found by trial and error methods and Q=10 and P=2000. For developing LQR were used model that was obtained by DSR identification method.

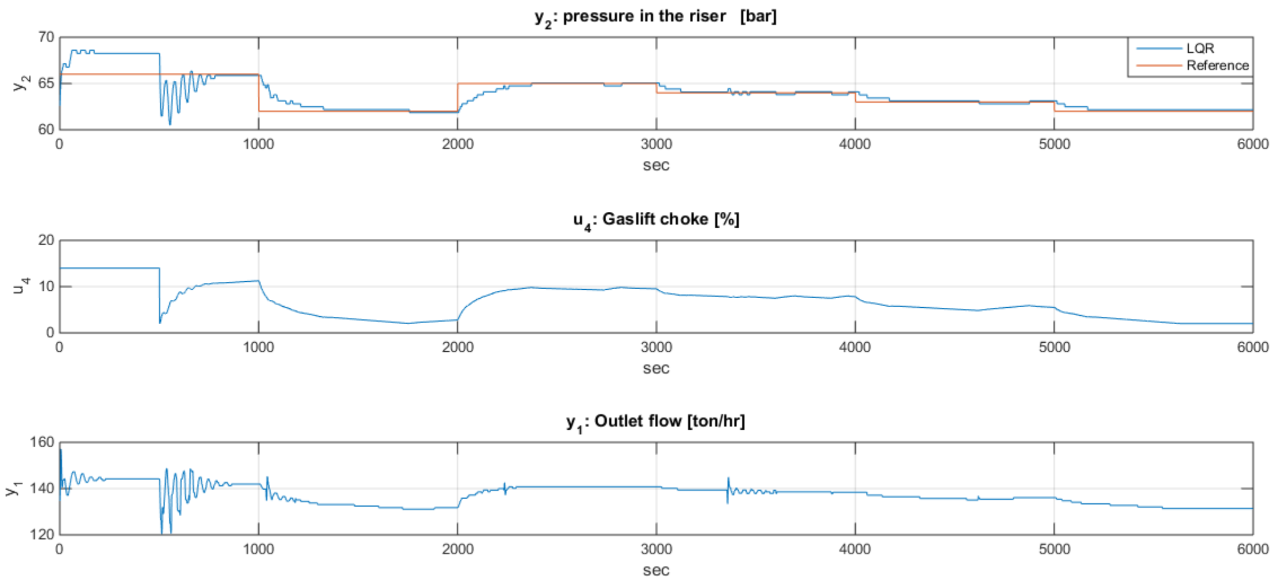


Figure 4-32 Implementation of LQR in 4th control strategy

According to the figure above, outlet flow is stable and it changes together with the pressure. The pressure signal u_2 tracked reference signal with IAE=630. First 500 samples LQR controller was not implemented, which is the reason of significant oscillation of the process in the beginning. The feedback matrices from eq. (3-12) are:

$$G_1 = [\, 0.0802 \quad -0.1180 \quad 0.0233 \quad 0.0105 \quad -0.0073 \quad -0.0115 \quad -0.0087 \quad -0.0749 \quad -0.0205 \quad -0.0577 \,]$$

$$G_2 = [-0.0222]$$

4.4.1. Implementation of PI-controller

Figure 4-33 shows simulation results with implementation of PI-controller. The duration of experiment was 6000 sec. During the experiment PI-controller were turn off and on few times: first 500 sec and then one more time from *t*=3000 sec to *t*=4500 sec. According to the received results the outlet flow almost stabilized during work of PI-controller. PI controller was tuned by using MATLAB tuner application with parameters Kp=35 and Ti=87. During the experiment IAE was equal to IAE=3045.

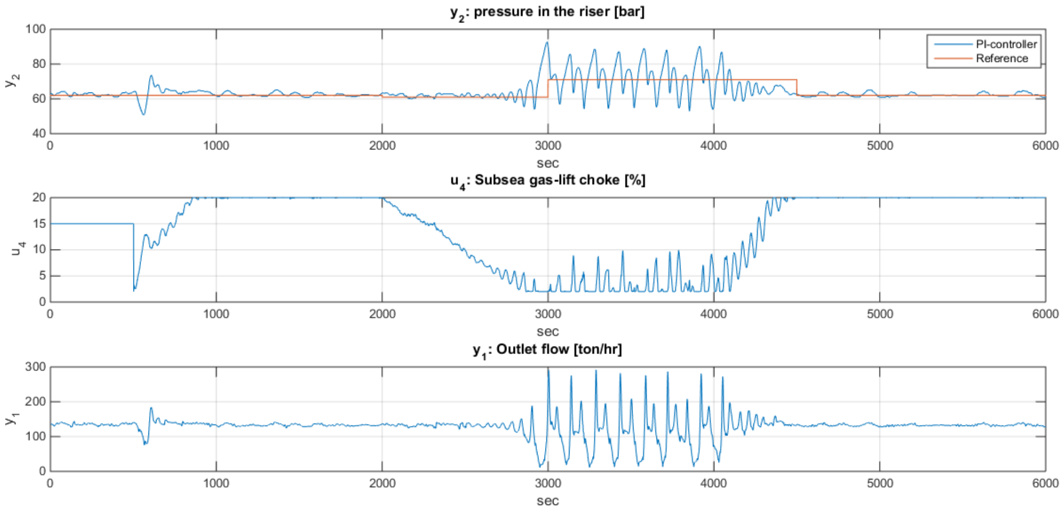


Figure 4-33 Implementation of PI-controller in 4th control strategy

5. Discussion

All obtained results are discussed in this chapter of the master thesis. All chapters are talked over separately and explained choices and results, which were achieved during the work on this master thesis.

5.1. Discussion about system description

In this chapter was presented the K-Spice model and general illustration of the riser and discussed the "nature of slugging". There were mentioned four control inputs, which are topside, gaslift, subsea and subsea-gaslift chokes, as well as two outputs of the system: outlet flow and pressure in the riser. According to the requirements, it was created a system that represents the data exchange process between K-Spice and MATLAB. The data exchange was done by MatrikonOPC server, which was described in subchapter 2.4 OPC. Not less important is description of the K-Spice software, which has been represented together with the main software applications.

5.2. Discussion about theory and methods

Knowledge and understanding of the basic theory used in this study is an important part of this master thesis. This section has been presented three different identification methods DSR, PEM and N4SID. DSR method is different in many aspects from the PEM and N4SID, for example, user can specify not only the system order (*n*) but also different lengths of the past and identified horizons. Input signal was presented into two ways as step function and PRBS, both of them were used in this master thesis in open-loop experiments. As a separate subchapter was described the main principal and functionality of the LQR.

5.3. Discussion about simulation results

According to the model description it was taken to look over four different control strategies. The main difference of these strategies was in control signals, which were sent to different chokes and find the best combination of input signal manipulation that can provide the highest productivity. For each the strategy was find bifurcation point and then were collected and saved 3600 samples by using PRBS experiment. Collected data contained information about input signal changing and both output signals flow and pressure. These data were used for model identification using DSR, PEM and

N4SID. The obtained model were checked with validation and identification sets and compered using MSE and stability analyze. Based on DSR models from each strategy were developed and implemented LQR's, which stabilizes flow better then PI controller. The best results preformed the 2nd control strategy and confirmed that usage of gaslift choke can remove slugging better then other strategies. At the same time topside choke manipulation can stabilize flow with good performance.

5.3.1. 1st control strategy

The 1st strategy represented control of topside choke. The bifurcation point for the input signal of topside choke is u_1=8% of opening choke and outlet flow stabilize with not significant oscillation in range of 125 ton/hr, when pressure oscillated in the range 63 bar. PRBS experiment was done for collecting input and output data. These samples were used during the system identification process. That was done by different methods. Model identification was done for two cases use different output signals. In the 1st case, the best result was obtained by DSR method with MSE=11.7425 and the worst by PEM with MSE=16.8617. In the second case, such a tendency has been seen as at the first but MSE error in DSR method was noticeably lower and equal to 0.3042. The system order was identified as 5th in both cases. Based on obtained result from DSR model in the 2nd case was successfully designed and implemented LQR with Q=0.5 and P=500. Difference between reference signal and output signal that was represent as IAE and equal to 374. For comparison, also used PI controller that showed worse result in stabilizing flow then LQR. In the result IAE was considerably higher equal to 638.

5.3.2. 2nd control strategy

The obtained result from open loop experiment showed that the bifurcation point for the input signal is u_2=14.5% and outlet flow oscillated in the range 148 ton/hr. For collecting samples was done PRBS experiment. The best identification result was achieved by implementing DSR methods, MSE were equal to 0.3042 and 0.1260 for the 1st and 2nd case respectively. The system order was identified as 6th in both cases. PEM and N4SID preformed good identification results according to MSE but both systems were unstable. Model which was obtained in the 2nd case was implemented for developing LQR. The simulation with implementation LQR was done during 6000 sec and weighting matrices Q=1 and P=1000. Model control by LQR perfumed good results and stabilized the outlet flow. IAE of 2nd control strategy is significant lower then in 1st strategy and equal to 251. But completely opposite result was received from implementation of PI controller with Kp=11.398 and Ti=51.037. The reference signal was tracked with significant osculation, which can be explained by anti-windup effect and low performance of tuning application in MATLAB.

5.3.3. 3rd control strategy

3rd control strategy represent results from subsea choke control, while other chokes were stable. Open-loop experiment showed that increasing u_3 higher then 10% leads to an increase in the system overshooting but $u_3<10$ leads to extremely high pressure that can means low outlet flow. For $u_3=40\%$ was done PRBS experiment for collecting samples. Table 4-5 and Table 4-6 showed identification characteristics for 1st and 2nd cases. According to the preformed parameters, N4SID and DSR preformed the lowest MSE in the 1st case and in the 2nd case the best result was obtained by implementation DSR method. The system order was higher in the 2nd case n=8. For design LQR was implemented model received in the 2nd case. Simulation time was equal to 6000 sec and weighting matrices Q=1 and P=1000. The IAE was higher then in 1st and 2nd control strategies and equal to 667. For comparison, it was used PI controller that was tuned by using MATLAB application. From this experiment, IAE was equal to 2.31e+3 because the reference signal did not track in time range from 2000 sec to 3000 sec.

5.3.4. 4th control strategy

The key concept of 4th control strategy was manipulating the subsea-gaslift choke to remove slug flow. This control strategy preformed the worst results. The bifurcation point was found as $u_4=13\%$ and then it was used in PRBS experiment for collecting samples. The best identification result was achieved by implementing DSR method that obtained model with n=10. Implementation of LQR showed the good tracking results with Q=10 and P=2000. Implementation of PI controller showed IAE=1045.

6. Conclusion

The master project is about model free optimal and predictive control of K-Spice process simulator. The master report was started from introduction part and look over all mentioned objectives of this master project according to the task description.

During this master project was given the description of K-Spice simulator and process model, which was provided by Kongsberg Oil & Gas Technologies. Based on this process description were selected possible control inputs, which were topside, gaslift, subsea and subsea-gaslift chokes, and output signals, which were outlet flow and pressure in the riser. By taking into account, the above-mentioned control signals have been allocated four basic control strategies. In each of these strategies for the control signal was selected different chokes. For data exchange between K-Spice and MATLAB was implemented MatrikonOPC server. Design of experiment was done in MATLAB, while process was run by K-Spice simulator.

PRBS experiment was design by using MATLAB and was implemented for collecting the input and output data, which were saved in .mat files. Real data from the process were centred and scaled for better model identification.

During the projected were discussed three different identification methods DSR, PEM and N4SID. For all the aforementioned strategies, all of three methods of system identification have been applied. This provided an opportunity to compare these methods in the best way according to obtained results. All of model identification methods provide a good results but the best results were received by implementation DSR methods. It also should be mentioned that usage measurements of pressure in the riser as output signal gave the best model with minimum MSE in both validation and identification sets.

Models developed using DSR method were implemented for design and implemented LQR's. For each LQR were found weighting matrices Q and P, which could perform the best result of stabilizing the slug flow. PI controller was also implemented to stabilize the slug flow and showed worse result compared with LQR.

The best control strategy after all of the experiments was named 2nd control strategy, where gaslift choke was chosen as control signal. It should also be noting that 1st control strategy with control the topside choke presented the results, which in no way inferior to the 2nd control strategy and can be also applied for removing slugging.

Possible future work are consider are:

1. Considering the possibility of simultaneous control of two or more chokes together and compare results.
2. Time of simulation can be increased for collecting more samples for model identification. PO-MOES and CVA could be considered for system identification.
3. As advance control could be implemented different prediction models and compared this results with presented result in this master thesis.
4. Experiments with implementation of different types of control strategies e.g. cascade control should be also considered in further work of this process.

References/bibliography

CHRISTER DALEN, D. D. R., ROAR NILSEN 2015. Model-free optimal anti-slug control of a well-pipeline-riser in the K-Spice/LedaFlow simulator. *Modeling, Identification and Control,* 36.(3)**,** 179-188.

EIKREM, G. O. 2014. *Statoil slug control*. NTNU: Statoil RDI.

ESPEN STORKAAS, S. S. 2006. *Controllability analysis of two-phase pipeline-riser systems at riser slugging conditions*. *A Journal of IFAC The International Federation of Automatic Control* [Online], 15. Available: http://www.nt.ntnu.no/users/skoge/publications/2007/storkaas_controllability-slug_cep/CONPRA2124.pdf [Accessed 20 December 2006].

INC, H. I. 2016. *MatrikonOPC* [Online]. Available: http://www.matrikonopc.com/index.aspx.

KONGSBERG 2014. *K-Spice DCS Link: OpcDaCom*. Release 1 ed.

KONGSBERG 2015. KONGSBERG K-Spice. Dynamic Process Simulator. Release 3.2 ed.

LJUNG, L. 1995. *System Identification*. Linkoping University.

PETER VAN OVERSCHEE, B. D. M. 1994. *N4SID: Subspace algorithm for identification of combined deterministic-stochastic systems*.

PETER VAN OVERSCHEE, B. D. M. 1996. *SUBSPACE IDENTIFICATION FOR LINEAR SYSTEMS. Theory - Implementation - Applications*, KLUWER ACADEMIC PUBLISHERS.

RUSCIO, D. D. 1995. *Distillation Column Toolbox*. Version 1.0 ed.

RUSCIO, D. D. 2000. *D-SR Toolbox for MATLAB*. Version 1.0 ed.

RUSCIO, D. D. 2001. *MODEL PREDICTIVE CONTROL and optimization*. Telemark University College.

RUSCIO, D. D. 2003. *Discrete LQ optimal control with integral action*. Telemark Institute of technology.

RUSCIO, D. D. 2007. *Subspace system identification of the Kalman filter: open and closed loop systems*. Telemark university college.

RUSCIO, D. D. 2009. *Linear Polynomial Estimator: The State Observer*. Telemark university college.

RUSCIO, D. D. 2014,. Subspace system identification. Theory and applications. Porsgrunn, Norway: Telemark Institute of Technology.

SKEIE, N.-O. 2014. *Course SCE2006 Industrial Information Technology*. Telemark University College.

SIGURD SKOGESTAD, H. S. 2005. *Cascade control experiments of riser slug flow using topside measurements*. *Proceedings of the 16th IFAC World Congress* [Online], 16. Available: http://www.ifac-papersonline.net/Detailed/28878.html.

TORSTEN SODERSTROM, P. S. 1988. *System Identification (Prentice Hall International Series in Systems and Control Engineering)* UK, Prentice Hall

WIKIPEDIA. 2013. *Pseudorandom Binary Sequence* [Online]. Available: https://de.wikipedia.org/wiki/Pseudo-random_bit_stream [Accessed 24 April 2013].

WIKIPEDIA. 2015. *OPC Data Access* [Online]. Available: https://en.wikipedia.org/wiki/OPC_Data_Access.

WIKIPEDIA. 2016a. *Open Platform Communications* [Online]. Available: https://en.wikipedia.org/wiki/Open_Platform_Communications [Accessed 11 April 2016].

WIKIPEDIA. 2016b. *Step function* [Online]. Available: https://en.wikipedia.org/wiki/Step_function [Accessed 22 March 2016].

List of tables and charts

Annexes

Annex 1

Annex 1: Task description

Telemark University College

Faculty of Technology

FMH606 Master's Thesis

Title: Model Free Optimal and Predictive Control of the K-Spice Process Simulator

TUC supervisor: David Di Ruscio

Extrenal supervisor: Roar Nilsen, roar.nilsen@kogt.kongsberg.com, Christer Dalen.

External partner: Roar Nilsen, KOGT. Christer Dalen.

Task description:

1. Give a description of the K-Spice simulator.
2. Perform simulations from K-Spice and describe some cases of particular importance. Detect possible control input variables u, output variables and measurements and states x in the suggested cases of interest. For simplicity the most used strategy may be considered, i.e. using the choke valve as control signal and the bottom riser pressure as the output measurement.
3. Discuss different methods for developing simplified linearized and reduced order models, e.g. model reduction, system identification using DSR etc.
4. Perform an input experiment and collect input and output data matrices U and Y for the use in developing simplified models.
5. Use the DSR method, to develop simplified state space dynamic models.
6. Discuss and suggest different control and optimisation problems of interest, e.g. LQR, MPC and PID control methods.

Task background:

Kongsberg Oil & Gas Technologies has developed K-Spice, a powerful dynamic process simulator for use in the oil and gas industry. Detailed non-linear dynamic models based on thermodynamics and first principles are one of the corner stones of this simulator. These underlying models and the K-Spice simulator may be too complicated to be directly used in Model Predictive (MPC) algorithms. Thus, in order to use simple and robust MPC algorithms, it is of interest to develop simple linearized discrete time state space models.

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

Subspace system identification tools as e.g. the DSR method may be used to develop such simplified dynamic models.

Student category:

SCE students

Practical arrangements: This work is to be done at TUC.

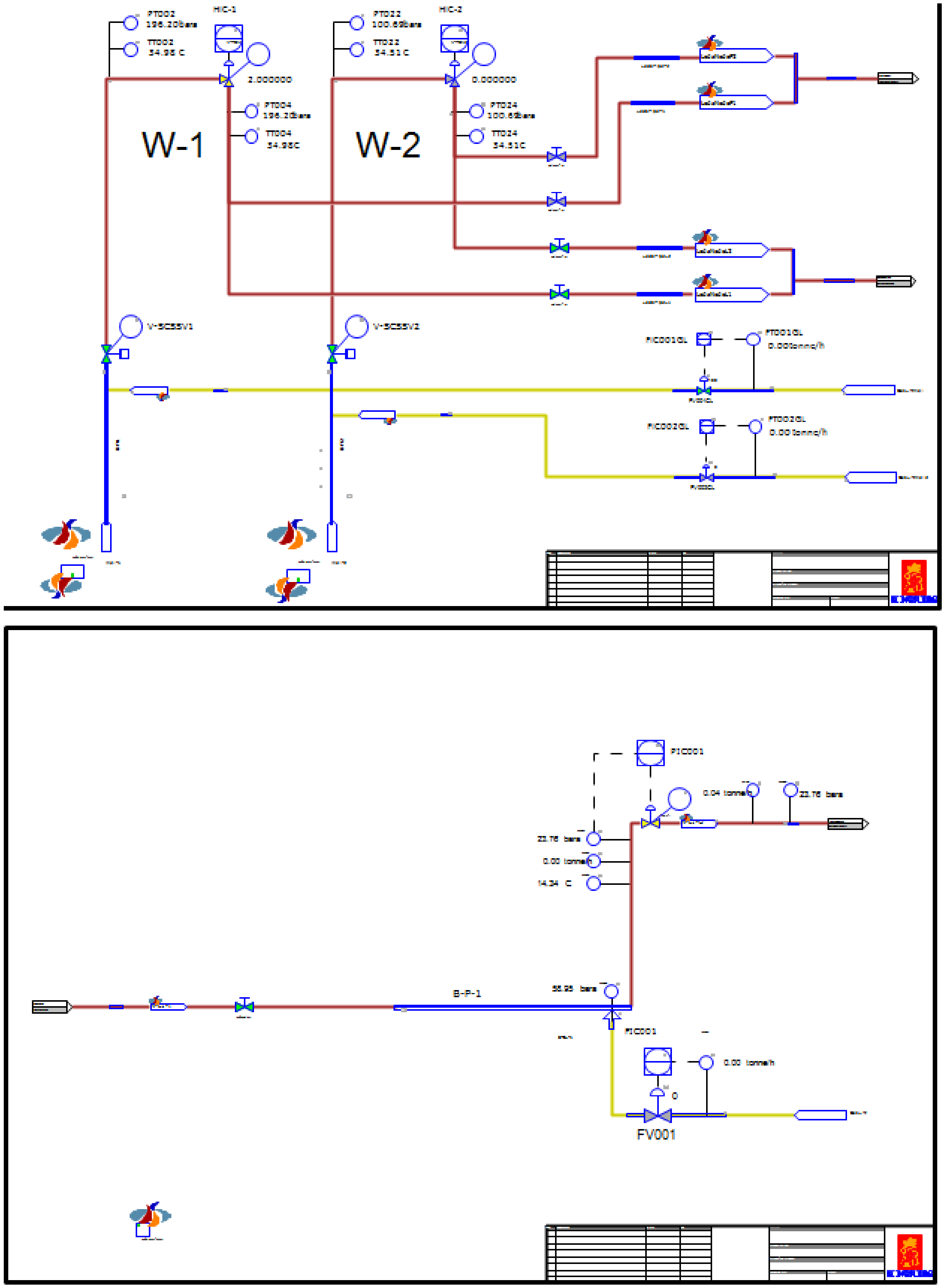
Signatures:

Student (date and signature): 01.09.2016

Supervisor (date and signature):

Annex 2

Annex 2: Subsea wells and pipelines with riser in K-Spice simulator



Annex 3

```
%OPC DA
%Created 2.03.2016, KB

da = opcda('localhost', 'Matrikon.OPC.Simulation.1');
connect(da);
grp = addgroup(da);
itm1 = additem(grp, 'main.local.HIC-1/ControllerOutput');
%itm1 = additem(grp, 'main.local.FIC001/ControllerOutput');
% itm2 = additem(grp, 'main.local.PIC001/ControllerOutput');
%  itm1 = additem(grp, 'main.local.FIC001GL/ControllerOutput');
grp1 = addgroup(da);
itm2 = additem(grp1, 'main.local.FT100/MeasuredValue');
grp2 = addgroup(da);
itm3 = additem(grp2, 'main.local.PT006/MeasuredValue');
grp3 = addgroup(da);
itm4 = additem(grp3, 'main.local.PT008/MeasuredValue');
```

Annex 4

```
opc_data;

Ts=1;% Sampling time [s]
N=3600; % Samples
Y=zeros(N,1);U=zeros(N,1);Z=zeros(N,1);W=zeros(N,1);
u=7.5; u1=8; u2=8.5; u3=9;
write(grp,u);
for i=1:N
    disp('iteration= '),i
    if (i>=100) && (i<200)
         write(grp,u1);
                end
      if(i>=200) && (i<300)
         write(grp,u2);
               end
               if(i>=300)
         write(grp,u3);
                           end
U(i)=u;
      y1=read(grp1);
        y=y1(1).Value;
        Y(i)=y;
        z1=read(grp2);
        z=z1(1).Value;
        Z(i)=z;
        w1=read(grp3);
        w=w1(1).Value;
        W(i)=w;
       pause(Ts);
end

figure(1)
subplot(211),plot(Y),grid,ylabel('y_1'),
title('y_1: Outlet flow [kg/s]'),xlabel('sec');
subplot(212),plot(Z),grid,ylabel('y_2'),
title('y_1: pressure in the bottom of riser   [bar]'),xlabel('sec]');
subplot(313),plot(W),grid,ylabel('y_3'),
title('y_1: pressure in the top of riser   [bar]'),xlabel('sec]');
```

Annex 5:PRBS experiment

```
opc_data;

Ts=1;% Sampling time
N=3600; % n.s
% Input and output arrays
Y=zeros(N,1);U=zeros(N,1);U1=zeros(N,1);W=zeros(N,1);

u=8;Tmin=20; Tmax=120;% topside choke
rnd=prbs1(N,Tmin,Tmax);
expU=rnd*0.5+u; % Input
write(grp,u);
for i=1:N
   disp('iteration= '),i
       y1=read(grp1); % read opc
   y=y1(1).Value;
   Y(i)=y;U(i)=u;
     z1=read(grp2);
      z=z1(1).Value;
      U1(i)=z;
      w1=read(grp3);
      w=w1(1).Value;
      W(i)=w;

   if (i>100) % start

      if u ~= expU(i)
         u = expU(i);
         write(grp,u);
      end

   end
   pause(Ts);
end
figure(1)
subplot(411),plot(U),grid,ylabel('u_1'),
title('u_2: Gaslift choke [%]'),xlabel('sec');
subplot(412),plot(Y),grid,ylabel('y_1'),
title('y_1: Outlet flow [ton/hr]'),xlabel('sec');
subplot(413),plot(U1),grid,ylabel('y_2'),
title('y_2: pressure in the riser   [bar]'),xlabel('sec');
subplot(414),plot(W),grid,ylabel('y_3'),
title('y_3: pressure in the top of riser   [bar]'),xlabel('sec');

% save('subseaO.mat','Y','U','U1');
```

```
%  whos('-file','subseaO.mat')
save('topsideO.mat','Y','U','U1');
  whos('-file','topsideO.mat')
%save('gasliftpressureO.mat','U','U1');
  %whos('-file','topsideO.mat')
% save('gasliftO2.mat','U','U1','Y');
%   whos('-file','gasliftO2.mat')
%save('gasliftO1.mat','Y','U','U1');
%whos('-file','gasliftO1.mat')
```

Annex 6: DSR testing

```
% load 'gasliftO2.mat'
 Uid0=U-mean(U(:));
 Yid0=Y-mean(Y(:));
 Yid=Yid0; Uid=Uid0
N=3600;
addpath('D:\MATLAB\R2014b\d-sr'); % add path to the current folder
P=zeros(2000,1);
LL=zeros(2000,1);
JJ=zeros(2000,1);
NN=zeros(2000,1);
for i=1:6000;
  P(i) = 999999;
  end
i=1;
for  n=6:8;
for L=30:70;
    for J=30:70;
              [A,B,D,E,C,F,x0]=dsr(Yid,Uid,L,0,J,1,n);
Ym=dsrsim(A,B,D,E,Uid,x0);
if L<=J
  P(i)=(Yid-Ym)'*(Yid-Ym)/N;
  LL(i)=L;
JJ(i)=J;
NN(i)=n;
i=i+1;
end
end
end
end
plot([Yid(:,1) Ym(:,1)])

[val,ind] =min(P);
 display(LL(ind));
  display(JJ(ind));
   display(NN(ind));
   display(val);  %MSE
```

Annex 7:Identification and validation

```
addpath('D:\MATLAB\R2014b\d-sr'); % add path to the current folder
N=3600;
Nid=ceil(0.75*N);
 Uid0=U-mean(U(:));
  Yid0=U1-mean(U1(:));
if val==0
  Yid=Yid0; Uid=Uid0;
    Yval=Yid0; Uval=Uid0;

else if val==1;
   Yid=Yid0(1:Nid,:);  Uid=Uid0(1:Nid,:);
       Yval=Yid0(Nid+1:N,:);   Uval=Uid0(Nid+1:N,:)
end

 [A,B,D,E,C,F,x0]=dsr(Yid,Uid,56,0,60,1,8);
 [A,B,D,E,C,F,x0]=dsr(Yval,Uval,56,0,60,1,8);
 Ymval=dsrsim(A,B,D,E,Uval,x0);
Ym=dsrsim(A,B,D,E,Uid,x0);
```

Annex 8

Annex 8:PEM,DSR and N4SID

```
addpath('D:\MATLAB\R2014b\d-sr'); % add path to the current folder
N=length(U);
Nid=ceil(0.75*N);
 Uid0=U-mean(U(:));
 Yid0=Y-mean(Y(:));
disp('Using DSR')
DSR
  Yid=Yid0; Uid=Uid0;
    Yval=Yid0; Uval=Uid0;
     Yid=Yid0(1:Nid,:);  Uid=Uid0(1:Nid,:);
       Yval=Yid0(Nid+1:N,:);   Uval=Uid0(Nid+1:N,:)
%end
 [A,B,D,E,C,F,x0]=dsr(Yid,Uid,29,0,29,1,5);
 [A,B,D,E,C,F,x0]=dsr(Yval,Uval,29,0,29,1,5);
 Ymval=dsrsim(A,B,D,E,Uval,x0);
Ym=dsrsim(A,B,D,E,Uid,x0);
 Pid=(Yid-Ym)'*(Yid-Ym)/1700;
 Pval=(Yval-Ymval)'*(Yval-Ymval)/300;

PEM and N4SID
 dat=iddata(Yid,Uid); % Pack data into an object
 th=pem(dat,5);
 disp('Using N4SID')
 N4SID model
 sys = n4sid(dat,5)
 Ym_n4sid=dsrsim(sys.a,sys.b,sys.c,sys.d,Uid,sys.x0);
 er_n4sid=Yid-Ym_n4sid;
 L_n4sid=er_n4sid'*er_n4sid/N
 disp('Using PEM')
Simulate PEM model
Ym_pem=dsrsim(th.a,th.b,th.c,th.d,Uid,th.x0);
  er_pem=Yid-Ym_pem;
  L_pem=er_pem'*er_pem/N

sys=ss(th.a,th.b,th.c,th.d);
sys=ss(sys.a,sys.b,sys.c,sys.d);
 zeros_pem=tzero(sys);
 pole_pem=eig(th.a)
  zeros_n4sid=tzero(sys);
 pole_n4sid=eig(sys.a)
 disp('zeros,poles')
zero1pem=mean(zeros_pem);
pole1pem=mean(pole_pem);
```

```
zero1n4sid=mean(zeros_n4sid);
pole1n4sid=mean(pole_n4sid);

figure(1)
subplot(221),plot([Yid(:,1) Ym(:,1)]),grid;
title('DSR model output against real output'),xlabel('sec');
legend('Yid','Ym');
subplot(222),plot([Ym_n4sid(:,1) Yid(:,1)]),grid;
title('N4SID model output against real output'),xlabel('sec');
legend('Yid','Ym n4sid');
subplot(223),plot([Ym_pem(:,1) Yid(:,1)]),grid;
title('PEM model output against real output'),xlabel('sec');
legend('Yid','Ym pem');
subplot(224),plot([Yid(:,1) Ym(:,1) Ym_n4sid(:,1) Ym_pem(:,1) ]);grid;
title('DSR, PEM, N4SID models outputs against real outputs') ,xlabel('sec');
legend('Yid','Ym','Ym n4sid','Ym pem')
```

Annex9: *:Design LQR*

```
addpath('D:\MATLAB\R2014b\d-col');

% setup OPC
opc_data;

N=6000; Ts=1;
 ym=66; um=40;% reference
CF=C;
K=CF*inv(F); % calculate kalman gain
n=length(A);
[n,r]=size(B); [m,n]=size(D);
Q=1; P=1000; ;% Q=0,5 P=500-topside; 1;1000-gaslift; gaslift2 - 10 2000
 [G1,G2]=dlqdu_pi(A,B, D,Q,P);
%Initial states
 yold=ym;dx=zeros(n,1);u=um;
 uold=0;eold=0;
IAE=0;TV=0;
Y=zeros(N,1);U=zeros(N,1); V=zeros(N,1);%Input and output arrays
I=ones(N/6,1);
R=[(ym)*I;(ym)*I;(ym-5)*I;(ym-2)*I;(ym+3)*I;(ym+1)*I];% reference signal
write(grp,um)
pause(1)
u_max=20;u_min=2;
start=50;stop=300;
w=0;

 for k=1:N
   disp('iteration= '),k

   y1=read(grp2);
   y=y1(1).Value;
   v1=read(grp1);
   v=v1(1).Value;
   Y(k)=y;U(k)=u;V(k)=v;
   if (k>start&&k<stop)
            r1=R(k);e=y-r1;
        du=G1*dx+G2*(yold-r1);
      u=du+uold;
      uold=u;
        dx=A*dx+B*du+K*(y-yold-D*dx-E*du);% Estimate the states
        if u>(u_max) && (w==1)
         u=u_min;
         w=1;
      elseif u>(u_max)
```

```
        u=u_max;
      end
      if u<u_min
        u=u_min;
      end

      % Integrated Absolute Error (IAE)
      IAE=IAE+abs(e);
        uold=u;eold=e;yold=y;

    end

    write(grp,u)

    pause(Ts)
  end

subplot (311), plot([Y(:,1) R(:,1)]);
grid;ylabel('y_2'),
title('y_2: pressure in the bottom of riser   [bar]'),xlabel('sec');
subplot (312), plot(U);ylabel('u_1'),
grid;title('u_1: Topside choke choke [%]'),xlabel('sec');
subplot (313), plot(V);
grid;ylabel('y_1');
title('y_1: Outlet flow [ton/hr]'),xlabel('sec');
```

Annex 9: PI-controller

```
opc_data
Ts=1.0;
N=300; %N samples

Kp=14; Ti=20%Kp=14580; Ti=1; %-10,60 ok for case_3
KpTi=Kp/Ti;
z=0; %initial controller state
um=8; ym=64;
u=um;uold=0;   %Initial values

IAE=0;TV=0;
u_max=5; u_min=-5;

Y=zeros(N,1);U=zeros(N,1);V=zeros(N,1) %Allocate arrays

I=ones(N/6,1);
R=[(ym)*I;(ym)*I;(ym-5)*I;(ym-2)*I;(ym+3)*I;(ym+1)*I];

write(grp,um)
%pause(5)
u_max=um+5;u_min=um-5;
start=50;stop=300;

for k=1:N

  disp('iteration= '),k

  y=read(grp2);
  y=y.Value;
   v1=read(grp1);
  v=v1(1).Value;
  V(k)=v;Y(k)=y;U(k)=u;% Input and Outputs

  if (k>start&&k<stop)% Start the controller

     r=R(k);e=r-y;

     u=z+Kp*e;

  if u>u_max % Anti-windup
     u=u_max;
     z=z;
  elseif u<u_min;
     u=u_min;
```

```
        z=z;
    else
        z=z+Kp*e/Ti;
    end
        % Integrated Absolute Error (IAE)
    IAE=IAE+abs(e);
     uold=u;
        end
    write(grp,u)
    pause(Ts)
end
subplot (311), plot([Y(:,1) R(:,1)]);
grid;ylabel('y_2'),
title('y_2: pressure in the bottom of riser   [bar]'),xlabel('sec');
subplot (312), plot(U);ylabel('u_1'),
grid;title('u_1: Topside choke choke [%]'),xlabel('sec');
subplot (313), plot(V);
grid;ylabel('y_1');
```