Modeling, Identification and Control, Vol. 36, No. 3, 2015, pp. 179–188, ISSN 1890–1328

Model-free optimal anti-slug control of a well-pipeline-riser in the K-Spice/LedaFlow simulator

Christer Dalen [1] David Di Ruscio [1] Roar Nilsen [2]

[1] *Telemark University College, P.O. Box 203, N-3901 Porsgrunn, Norway.*

[2] *Kongsberg Oil & Gas Technologies, Norway.*

Abstract

Simplified models are developed for a 3-phase well-pipeline-riser and tested together with a high fidelity dynamic model built in K-Spice and LedaFlow. These models are developed from a subspace algorithm, i.e. Deterministic and Stochastic system identification and Realization (DSR), and implemented in a Linear Quadratic optimal Regulator (LQR) for stabilizing the slugging regime. We are comparing LQR with PI controller using different performance measures.

Keywords: optimal controller, integral action, PI controller, Kalman filter, system identification, anti-slug, well-pipeline-riser

1 Introduction

In the offshore industry, multiphase transportation pipelines, which parts may consist of one or several risers, can introduce a set of different flow patterns, in particular; 'Severe slugging'. The signature of 'Severe slugging' phenomenon is large pressure and flow oscillations, and it is of great interest to stabilize this flow regime since it may endanger personnel and equipment, as well to reduce production rate.

A subset of papers proposing different anti-slug control solutions, is bulleted below:

- Introduced gaslift at riser base as control input, controlling riser base pressure, in Alvarez and Al-Malki (2003).
- Feedback PID control strategy of a pipeline-riser, controlling the riser base pressure with the topside choke as control input, in Ogazi AI (2010), Jahanshahi and Skogestad (2015), Storkaas and Skogestad (2007), Storkaas et al. (2001) and Skogestad (2009).
- Cascade control strategies of a well-subsea-riser, controlling riser base pressure with topside- and subsea choke, in Godhavn et al. (2005)

In this paper we will not use any models developed from mechanistic rules, actually, since the controlling results presented in this paper evolve only from a collection of data we may refer to this solution as Model-Free Control (MFC), a concept contained in Di Ruscio (2012). The previous mentioned paper demonstrates MFC on a lab-scale quadruple tank process using an LQR optimal controller. The proposed controller used is optimal in the sense that a standard linear quadratic performance index is minimized. The essential problem in this paper will be to identify system matrices for a linear state space model, using a subspace algorithm, i.e. DSR (Di Ruscio (1996)). The DSR algorithm has shown good performance over other algorithms, compared on an activated sludge process (Sotomayor et al. (2003)).
The main contributions of this paper are itemized as follows:

- System identification approach on the well-

doi:10.4173/mic.2015.3.5

pipeline-riser example, using a subspace algorithm.

- Model-free optimal anti-slug control of 3 different cases, each described in Section 4.

A most valuable tool for investigating such slugging behavior, has been to use the 'state-of-art', modelling tools; LedaFlow multiphase flow simulator (LedaFlow) integrated with a K-Spice dynamic process simulator (K-Spice), developed and used by Kongsberg Oil & Gas Technologies for the last 30 years in the oil and gas industry. K-Spice and LedaFlow are high fidelity simulators and are well suited to investigate the real offshore well, pipeline, riser and topside process integrated in one dynamic model. LedaFlow is an independent and open simulator that is the first to provide slug capturing and the only solution that predicts hydrodynamic slugs.

Enumerated as in sections, the paper is organized as follows:

1. In the introduction we present the anti-slug problem, past solutions and our contributions.

2. In the process description we describe the well-pipeline-riser.

3. In the theory section we define the system model, the problem and the functions which the results of this paper rest upon.

4. In the simulations section we identify models and implement them in a model-free optimal anti slug control for three different cases.

5. Some concluding remarks.

2 Process Description

A 3-phase well-pipeline-riser example integrated in the K-Spice/LedaFlow simulator is studied in this paper. This example has 3 manipulative inputs of interest for controlling flow/pressure; Topside choke, Subsea choke and Gaslift. Together with the sentences itemized below, the pipeline profile; Fig. 1 gives a brief description of the process example.

- Outputs $\begin{cases} y_1\text{: Outlet flow, FT100, [kg/s]} \\ y_2\text{: Riser pressure, PT006, [bara]} \end{cases}$

  where $y_1 \in [0, 100]$ and $y_2 \in [0, 200]$. Note that y_2: Riser pressure is the pressure in the bottom of the riser as illustrated in Fig. 1.

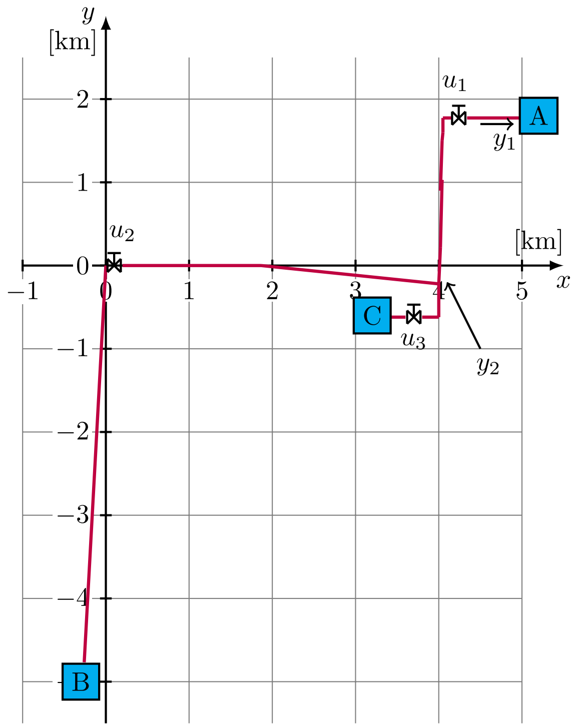

Figure 1: Illustration of the 3-phase well-pipeline-riser process integrated in the K-Spice/LedaFlow simulator.

- Inputs $\begin{cases} u_1\text{: Topside choke, HC001, [\%]} \\ u_2\text{: Subsea choke, V-HCV1, [\%]} \\ u_3\text{: Gaslift choke, FIC001, [\%]} \end{cases}$

  where $u_i \in [0, 100] \;\; \forall \;\; i = 1, 2, 3$.

- Stream-constrains $\begin{cases} A\text{: 25 [bara]} \\ B\text{: 500 [bara], } 100[°C] \\ C\text{: 120 [bara], } 30[°C] \end{cases}$

  Note that bara is the absolute pressure expressed in bar, where 0 bara is associated with total vaccum.

- Gaslift stabilize the production flow rate by decreasing the density and increasing the flow rate.

3 Theory

Definition 3.1 (System model)
We assume that the underlying system can be described by a Linear discrete Time-Invariant (LTI) State Space Model (SSM) of following form

$$\bar{x}_{k+1} = A\bar{x}_k + Bu_k + Ce_k \begin{cases} \text{Initial predicted state} \\ \bar{x}_0 \end{cases}, \quad y_k = D\bar{x}_k + Eu_k + Fe_k, \tag{1}$$

where $k \in \mathbb{N}$ is the discrete time, $\bar{x}_k \in \mathbb{R}^n$ is the predicted state vector, $u_k \in \mathbb{R}^r$ is the input vector, $y_k \in \mathbb{R}^m$ is the output vector and $e_k \in \mathbb{R}^m$ is white noise with unit covariance matrix, i.e. $E(e_k e_k^T) = I$. We may have the model in a traditional way by writing the common Kalman filter on innovations form, i.e.

$$\bar{x}_{k+1} = A\bar{x}_k + Bu_k + K\varepsilon_k \begin{cases} \text{Initial predicted state} \\ \bar{x}_0 \end{cases}, \quad y_k = D\bar{x}_k + Eu_k + \varepsilon_k, \tag{2}$$

where $\varepsilon_k = Fe_k$ is the innovations process, $K = CF^{-1}$ is the Kalman filter gain matrix and $E(\varepsilon_k \varepsilon_k^T) = FF^T$ is the the innovations covariance matrix. Note that in this paper we have forced the feed-through matrix, $E = 0$, by setting $g = 0$ which is shown in Eq. 8.

Definition 3.2 (System Identification Problem) *From known input and output time series, the problem is to identify a state space model, i.e the following system matrices (A, B, C, D, E, F) in Eq. 1 and the initial state $\bar{x}_0$. The time series*

$$\left.\begin{matrix} u_k \\ y_k \end{matrix}\right\} \forall \;\; k = 1, \ldots, N,$$

are organized as output and input matrices, respectively

$$Y = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_N^T \end{bmatrix} \in \mathbb{R}^{N\times m}, U = \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_N^T \end{bmatrix} \in \mathbb{R}^{N\times r}. \tag{3}$$

It is important to note that we are using centered data, i.e. $u_k := u_k - u^0$ and $y_k := y_k - y^0$, where

$$y^0 = \frac{1}{N}\sum_{k=1}^{N} y_k, \tag{4}$$

$$u^0 = \frac{1}{N}\sum_{k=1}^{N} u_k. \tag{5}$$

The removing of trends from the data will often increase the accuracy of the estimated model.

Definition 3.3 (Functions) *A set of functions are itemized below, essentially, the problems considered in this paper are solved, in MATLAB, by combining members from this set, structured inside (nested) for-loop(s). We may associate the MATLAB scripts with the function diagrams/block diagrams shown in Figs. 2, 3 and 4.*

- *Pseudo Random Binary Sequence (PRBS), a MATLAB function designed as*

$$U = prbs1(N, T_{min}, T_{max}), \tag{6}$$

where U is as defined in Eq. 3 and $u_k \in \{-1, 1\} \;\forall\; k = 1, \ldots, N$. The signal u_k is PRBS such as the constant intervals T_i are random in the interval $T_{min} \le T_i \le T_{max}$. See e.g. Fig. 6. The reason for using a PRBS excitation signal is that we want to be able to identify a model with sufficiently high order n. Notice, that a pure step signal only is persistently exciting of order $n = 1$, Söderström and Stoica (1989).

- *Deterministic and Stochastic system identification and Realization, (DSR) Di Ruscio (1996). The model matrices in Eqs.1,2 are identified using the following MATLAB function:*

$$[A, B, D, E, C, F, \bar{x}_0] = dsr(Y, U, L, g, J, M, n) \tag{7}$$

$$\text{where} \begin{cases} L: 1 \le L: \text{Future horizon} \\ g: \text{Structure parameter} \\ \text{Note that } g = 0 \text{ gives } E = 0. \\ J: L \le J: \text{Past horizon} \\ n: 0 < n \le Lm: \text{Number of states} \\ M: M = 1 \text{ is default, a dummy parameter} \end{cases} \tag{8}$$

- *Mean Square Error (MSE):*

$$MSE = \frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_k^d)^2 \;, \tag{9}$$

where $\hat{y}_k^d$ is the output of the deterministic part of the model

$$\bar{x}_{k+1}^d = A\bar{x}_k^d + Bu_k, \quad \hat{y}_k^d = D\bar{x}_k^d, \tag{10}$$

and with initial state $\bar{x}_1^d = \bar{x}_0$.

- *Linear Quadratic Regulator (LQR), Di Ruscio (2012):*

$$u_k = u_{k-1} + G_1\Delta\bar{x}_k + G_2(y_{k-1} - r_k), \tag{11}$$

where the state deviation $\Delta\bar{x}_k = \bar{x}_k - \bar{x}_{k-1}$ *and* $r_k \in \mathbb{R}^m$ *is the reference for the output* y*. A MATLAB script calculates the optimal feedback matrices:* $[G1, G2] = dlqdu\_pi(A, B, D, Q, P)$*, where* Q *and* P *are the weighting matrices for respectively reference tracking and control deviation.*

- *State observer for state deviation, Di Ruscio (2012), evolved from Eq. 2, are*

$$\Delta\bar{x}_{k+1} = A\Delta\bar{x}_k + B\Delta u_k + K(y_k - y_{k-1} - D\Delta\bar{x}_k), \begin{cases} \textit{Initial state} \\ \textit{deviation} \\ \Delta\bar{x}_1 = 0 \end{cases} \tag{12}$$

where $\Delta u_k = u_k - u_{k-1}$*. The model matrices* (A, B, D, K) *are identified from DSR, i.e. from Eq. 7 with* $K = CF^{-1}$*.*

- *Integrated Absolute Error (IAE):*

$$IAE = \int_0^\infty |r - y| dt \tag{13}$$

We may calculate the IAE recursively, as shown in Di Ruscio (2010), in discrete time: $IAE_{k+1} = IAE_k + \Delta t|r_k - y_k|$*, where* Δt *is the sampling time.*

- *Total Value (TV):*

$$TV = \sum_{k=1}^{\infty} |\Delta u_k|, \tag{14}$$

where, $\Delta u_k = u_k - u_{k-1}$*, is the control rate of change.*

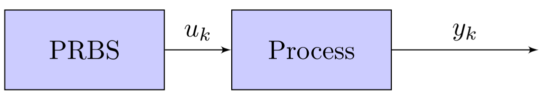

Figure 2: Block diagram of the proposed members working together through iterations of k, bounded as $1 \le k \le N$, to produce the input and output data, u_k and y_k, which is to be organized in matrices, Y and U as in Eq. 3. The PRBS block is as Eq. 6.

Definition 3.4 (Notation)
Because of some untraditional linguistics used through this paper, it is convenient, for not confusing the reader, to give some additional definitions.

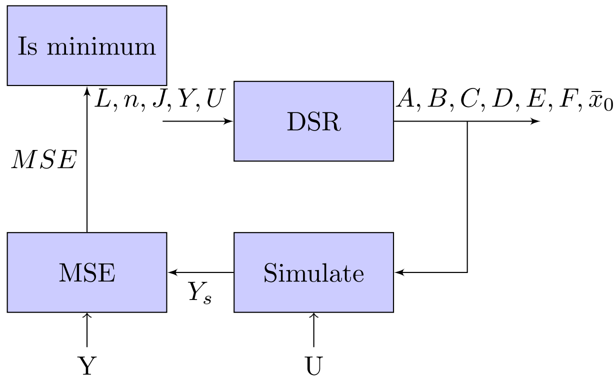

Figure 3: Block diagram of the proposed members working together through iterations of L, n, J, each bounded as described in Eq. 8. The optimal model, meaning the model giving the lowest MSE, is choosen. The 'DSR' block is as Eq. 7 and the 'MSE' block is as Eq. 9.

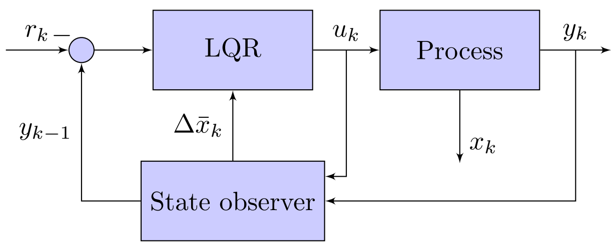

Figure 4: Block diagram of the proposed members working together through iterations of k, bounded as $1 \le k \le N$, to control y_k. The 'LQR' block is as Eq. 11 and the 'State estimator' block is as Eq. 12.

- *Real process := K-Spice/LedaFlow simulator*
- *Model := model identified from the DSR subspace algorithm*
- *@ := around or working point*

4 Simulation Results

4.1 Introduction

We present three cases for which we have applied MFC, where our goal is to control/stabilize the outlet flow. The sampling time is $\Delta t = 1$ sec., however different simulation speeds may be used in the K-Spice/LedaFlow simulator. The steps performed in each case is enumerated below.

1. Identify an interesting operating point, i.e. a point where severe slugging is present.
2. Collect datasets from an input experiment, Fig. 2.
3. Identify model, Fig. 3.
4. Control process, Fig. 4.

4.2 Case A: Topside choke and introduced Gas lift

Introducing gaslift is said to to be the most effective way of stabilizing the slugging regime. Considering the open-loop simulation (Fig. 5), we see that introducing gaslift is stabilizing the flow.
We define the case as

$$y \in \mathbb{R} := \left\{ y_1\text{: Outlet flow } [kg/s] \right. ,$$

$$u \in \mathbb{R}^2 := \begin{cases} u_1\text{: Topside choke @ 25 } [\%] \\ u_3\text{: Gaslift choke @ 1.5 } [\%] \end{cases} .$$

Inputs and outputs were collected into $U \in \mathbb{R}^{N\times 2}$ and $Y \in \mathbb{R}^N$(Fig. 6), where $N = 3600$ samples. The first 125 samples was removed, thereafter the set was divided into 2/3 for identification and 1/3 for validation.

It was observed that using both inputs u_1 and u_3 gave a higher order model, and worse prediction error than if we just used u_3, hence we will assume a single-input and single-output (SISO) model with u_3 as input and set $u_1 = 25.25$. The model is identified with DSR-parameters; $L = 7, J = 12, n = 5$.

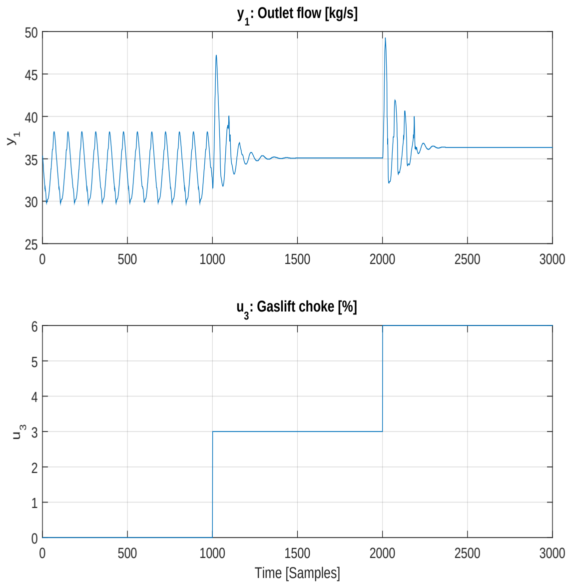

Figure 5: Open-loop simulations in K-Spice. Introducing the Gaslift choke at $Time = 1000$ Samples. Topside choke was kept constant at $u_1 = 25$. **Case A**

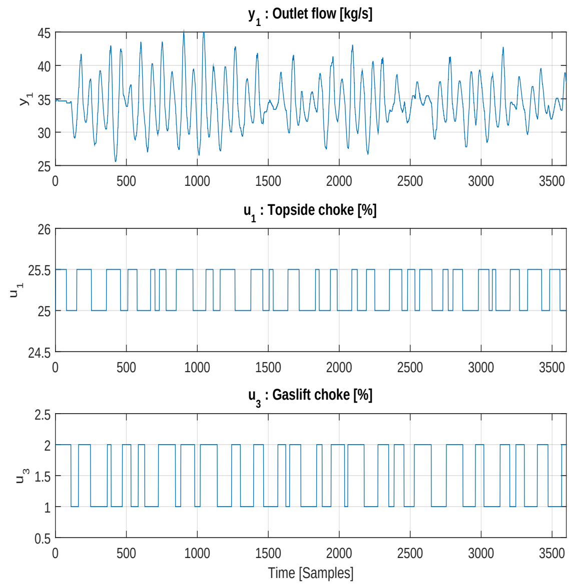

Figure 6: When stepping the topside choke and gaslift valve, the input and output series were collected from the K-Spice model, with a length of $N = 3600$ samples. These inputs are from an experimental design, i.e. PRBS as in Eq. 6 where $T_{min} = 20$ and $T_{max} = 120$. These results are from a MATLAB script associated with the block diagram in Fig. 2. The simulation speed in K-Spice was 30 times real time. **Case A**

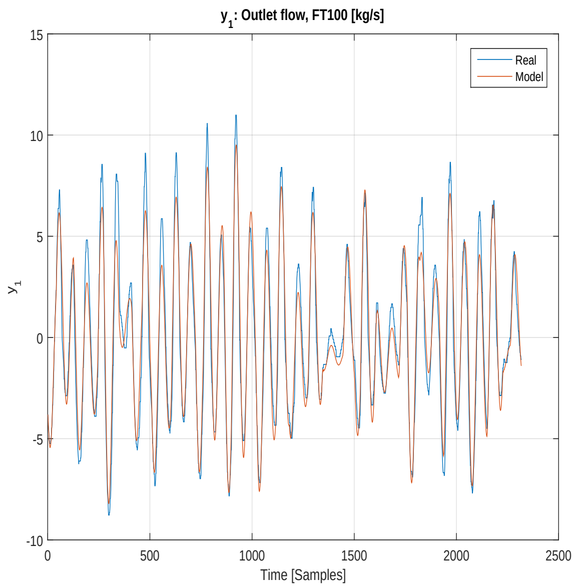

Figure 7: Model ($L = 7, J = 12, n = 5$) simulated and compared to the identification set, giving $MSE = 1.0777$. Results from a MATLAB script associated with Fig. 3. **Case A**

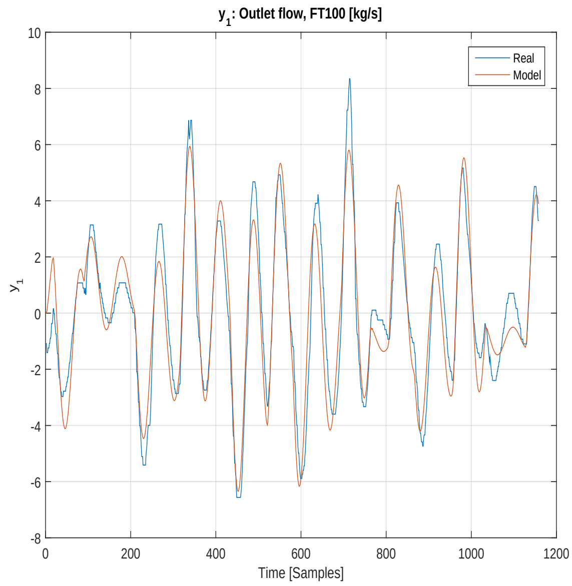

Figure 8: Model ($L = 7, J = 12, n = 5$) simulated and compared to the validation set, giving $MSE = 0.9330$. Results from a MATLAB script associated with Fig. 3. **Case A**

$$
\overbrace{
A = \begin{bmatrix} 0.9900 & -0.4930 & -0.0260 & -0.1033 & -0.0769 \\ 0.0162 & 0.9907 & 0.7521 & -0.2880 & 0.5406 \\ -0.0005 & -0.0006 & 0.6030 & 0.7346 & 0.1834 \\ -0.0002 & 0.0012 & -0.3761 & 0.0323 & 0.9153 \\ -0.0001 & 0.0003 & 0.1408 & -0.5039 & -0.1200 \end{bmatrix}
}^{\textbf{Identified SISO model}}
$$

$$
B = \begin{bmatrix} -0.5783 \\ 0.2534 \\ -0.0918 \\ -0.1306 \\ 0.0198 \end{bmatrix}
$$

$$
D = \begin{bmatrix} -0.3773 & -0.5658 & 0.5351 & -0.3387 & 0.3175 \end{bmatrix}
$$

$$
K = \begin{bmatrix} -3.8226 \\ 0.6193 \\ -0.1150 \\ -0.1107 \\ -0.0488 \end{bmatrix}
$$

The steady state gain is approximately 1.8 and the poles are less than one in magnitude, hence the process is stable.

The model looks to have a good fit to the datasets, see Figs. (Fig. 7) and (Fig. 8), moreover, the model is performing better over the validation set ($MSE = 0.9330$), than the identification set ($MSE = 1.0777$). Fig. 9 shows a successful implementation of the LQR, where the weights are tuned ($Q = 1$ and $P = 1000$) using the identified model. We observe that the control input is moving on towards a constant value after a given time. We are not surprised by the good performance, since the model is proven good in both identification and validation.

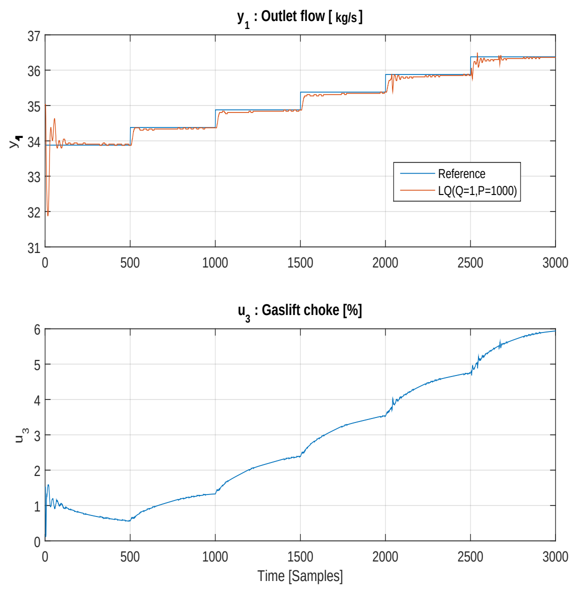

Figure 9: Implementation in K-Spice of the optimal controller, LQR, controlling outlet flow, y_1, with Gaslift choke u_3. The weights are $Q = 1$ and $P = 1000$. Results from a MATLAB script associated with Fig. 4. **Case A**

4.3 Case B: Topside choke and Subsea choke

Two manipulative input variables are chosen; Topside choke, u_1, and Subsea choke, u_2. Considering the open-loop simulations in Fig. 10 and some additional observations, we will assume that the process is marginally stable at $22 < u_1 \leq 100$ and $30 \leq u_2 \leq 45$. Hence, we define following case as

$$y \in \mathbb{R} := \left\{ y_1 \text{: Outlet flow [kg/s]} \right. ,$$

$$u \in \mathbb{R}^2 := \begin{cases} u_1\text{: Topside choke @ 25 [\%]} \\ u_2\text{: Subsea choke @ 40 [\%]} \end{cases} .$$

Input and output time-series were collected from an input experiment, (Fig. 11), and we identified a 5th order model (Fig. 11), from the first 5000 samples, with DSR-parameters; $L = 20, J = 23, n = 5$, which gave minimum $MSE = 2.4207$ (Fig. 12).

$$\overbrace{A = \begin{bmatrix} 0.9967 & -0.1703 & 0.1552 & -0.0712 & -0.0325 \\ 0.0151 & 0.9997 & 0.3479 & 0.0511 & 0.2430 \\ -0.0005 & 0.0001 & 0.3659 & 0.6785 & -0.4122 \\ 0.0000 & -0.0002 & -0.1975 & 0.7618 & 0.4638 \\ 0.0000 & -0.0001 & -0.0277 & 0.0477 & 0.7958 \end{bmatrix}}^{\text{Identified MISO model}}$$

$$B = \begin{bmatrix} -0.0713 & -0.0501 \\ -0.1594 & 0.0326 \\ 0.3942 & -0.0019 \\ -0.0033 & 0.0055 \\ 0.0304 & -0.0087 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.2114 & -0.3619 & 0.8160 & -0.1015 & 0.2399 \end{bmatrix}$$

$$K = \begin{bmatrix} -5.3680 \\ 0.1029 \\ -0.4218 \\ -0.1336 \\ 0.0763 \end{bmatrix}$$

Fig. 13 shows the controlling results of the LQR, tuned from trial-and-error methods. The LQR is introduced at $Time = 1000$ and is in fact able to stabilize the outlet flow in the region which we assumed marginally stable. Note that we have set the controller limits equal to this region. Despite how awful the model fits the identification set (Fig. 12) we are actually achieving seemingly good controlling results with the LQR.

4.4 Case C: Topside choke

We choose to investigate a case with only the topside choke as input variable. Considering the open-loop simulations Fig. 14 and some additional observations, we will assume that the process is marginally stable at $22 < u_1 \leq 100$. Hence, a case was constructed as

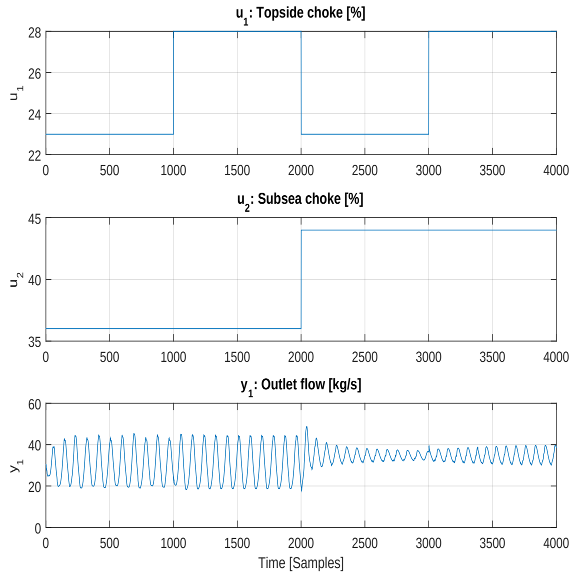

Figure 10: Open-loop simulations in K-Spice. Subsea choke looks to have a much higher steady state gain than the topside choke. **Case B**

$$y \in \mathbb{R}^2 := \begin{cases} y_1\text{: Outlet flow [kg/s]} \\ y_2\text{: Riser pressure [bara]} \end{cases} ,$$

$$u \in \mathbb{R} := \left\{ u_1\text{: Topside choke @ 25 [\%]} \right. .$$

A 4th order SISO model, with only output y_2, was identified from the time-series (Fig. 15) with DSR-parameters; $L = 5, J = 5, n = 4$, with minimum $MSE = 0.560$ (Fig. 16).

$$\overbrace{A = \begin{bmatrix} 0.9984 & -0.7044 & 0.4806 & -0.5121 \\ 0.0039 & 0.9944 & 0.2759 & 0.8613 \\ 0.0000 & -0.0026 & -0.2460 & 1.1076 \\ -0.0001 & 0.0035 & -0.6999 & 0.4946 \end{bmatrix}}^{\text{Identified SISO model}}$$

$$B = \begin{bmatrix} -0.0373 \\ -0.0010 \\ -0.0194 \\ -0.0080 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.4462 & -0.6336 & 0.6047 & 0.0857 \end{bmatrix}$$

$$K = \begin{bmatrix} -2.4289 \\ 0.6259 \\ 0.1498 \\ -0.1096 \end{bmatrix}$$

Fig. 17 shows successful implementations of two different control strategies; LQR and PI. Both controllers are tuned using the identified model. The controllers are introduced at 500 Samples and are both able to stabilize the

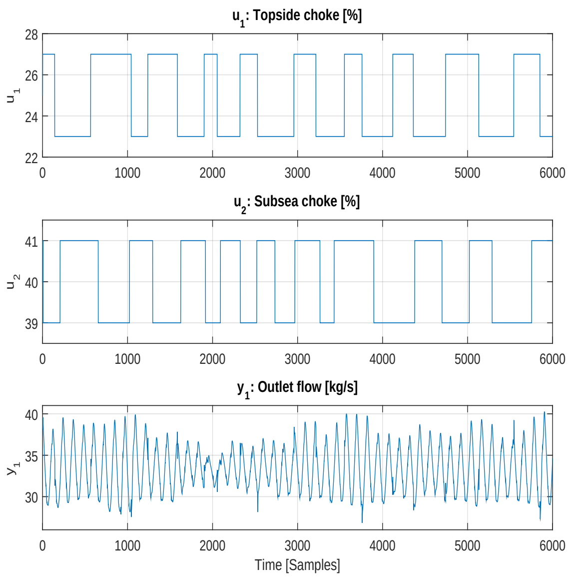

Figure 11: When stepping the topside choke and subsea choke, the input and output series were collected from the K-Spice model, with a length of $N = 6000$ samples. These inputs are from an experimental design, i.e. PRBS as in Eq. 6 where $T_{min} = 150$ and $T_{max} = 500$. These results are from a MATLAB script associated with the block diagram in Fig. 2. The simulation speed in K-Spice was 20 times real time. **Case B**

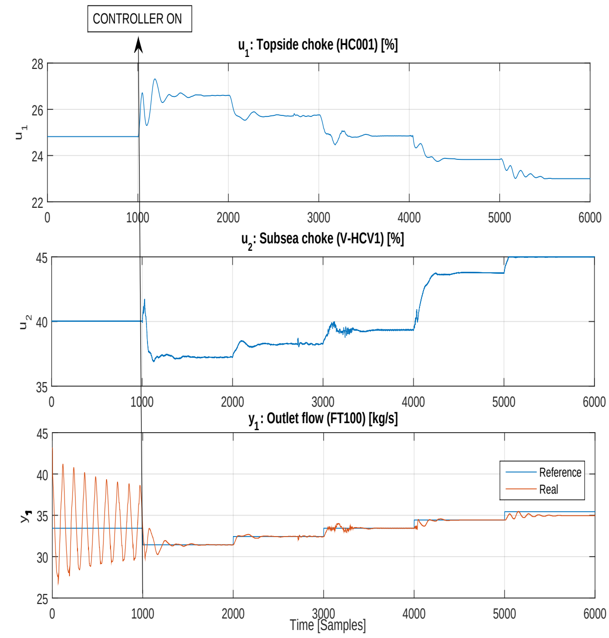

Figure 13: LQR controlling the identified 5th order model with $Q = 1$ and $P = 500 I_{2\times2}$. LQR introduced at $Time = 1000$ Samples. For $Time > 5000$, the Subsea choke, u_2, saturates, because of the bound $22 < u_2 \leq 45$, as specified. **Case B**

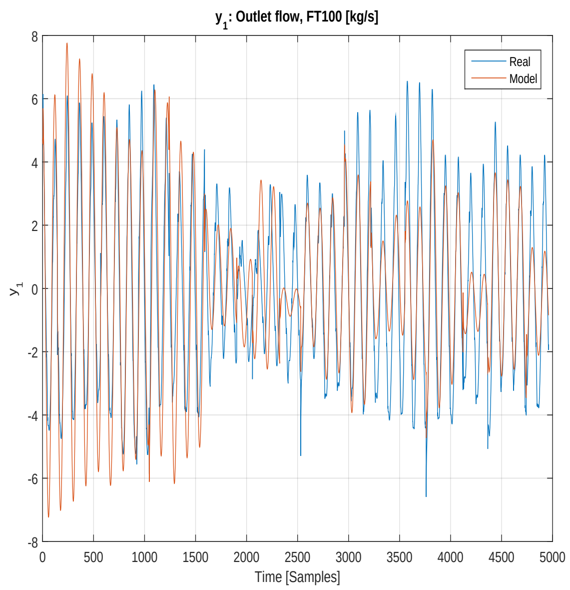

Figure 12: Model $(L = 20, J = 23, n = 5)$ simulated over the identification set. $MSE = 2.4207$. **Case B**

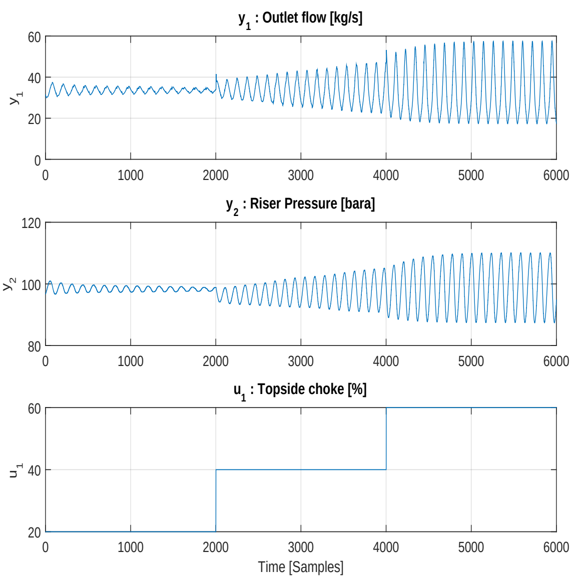

Figure 14: Open-loop simulations in K-Spice. We observe how the amplitudes are increasing as the topside choke is increasing. Note that $2000 \leq Time \leq 4000$ is a marginally stable region. **Case C**

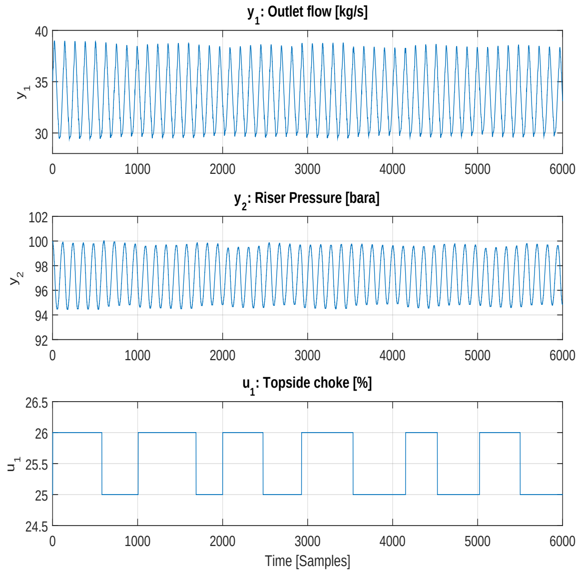

Figure 15: When stepping the topside choke, the input and output series were collected from the K-Spice model, with a length of $N = 6000$ samples. These inputs are from an experimental design, i.e. PRBS as in Eq. 6 where $T_{min} = 300$ and $T_{max} = 700$. These results are from a MATLAB script associated with the block diagram in Fig. 2. The simulation speed in K-Spice was 10 times real time. **Case C**

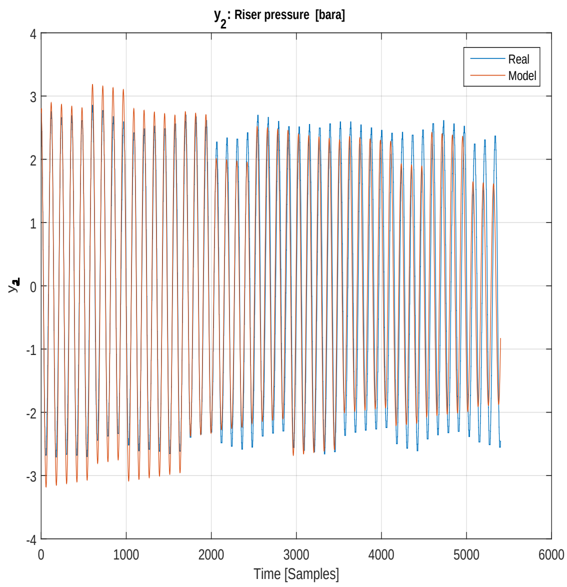

Figure 16: The model $(L = 5, J = 5, n = 4)$ is simulated and compared to the identification set, giving $MSE = 0.5760$. **Case C**

undesired slugging regime in the region assumed to have marginal stability, i.e. $22 < u_1 \leq 100$. The LQR shows better reference tracking ($IAE = 177.5$) than the PI controller ($IAE = 268.0$). Small oscillations are shown to begin after 2500 Samples with the PI controller, however the LQR shows more promising results. It is important to note that the PI controller could probably be tuned better.

Table 1: Comparing PI vs LQR control strategy using measures: Integrated Absolute Error (IAE) and Total Value (TV). See Fig. 17

Controller	Tuning parameters	IAE	TV
PI	$K_p = -10, T_i = 60$	267.961	196.027
LQR	$Q = 1, P = 10$	177.496	306.086

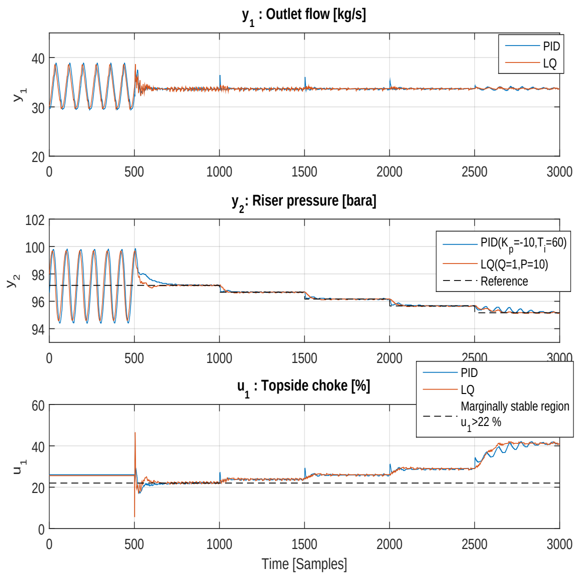

Figure 17: Comparing closed loop controllers: PI $(K_p = -10, T_i = 60, T_d = 0)$ and LQR $(Q = 1, P = 10)$. The controllers are introduced at $Time = 500$ Samples. **Case C**

5 Concluding Remarks

Practical implementations of Model-free optimal anti-slug control was successfully demonstrated on three different cases on the 3-phase well-pipeline-riser example in the K-Spice/LedaFlow simulator. Linearized reduced order SSM was identified from a subspace algorithm, i.e. DSR, based on time-series, collected using an input experiment, i.e. PRBS. In each case we where able to stabilize the outlet flow, using the LQR and PI controllers.

Acknowledgment

The authors acknowledge in bullets

- Kongsberg Oil & Gas Technologies for supporting with license and software for the K-Spice and LedaFlow simulator.
- Telemark University College

MATLAB functions

The MATLAB functions used in this work are available for academic use upon request.

References

Alvarez, C. and Al-Malki, S. Using gas injection for reducing pressure losses in multiphase pipelines. *International Journal of Multiphase Flow*, 2003. doi:10.2118/84503-MS.

Di Ruscio, D. Combined Deterministic and Stochastic System Identification and Realization: DSR - A Subspace Approach Based on Observations. *Modeling, Identification and Control*, 1996. 17(3):193–230. doi:10.4173/mic.1996.3.3.

Di Ruscio, D. On Tuning PI Controllers for Integrating Plus Time Delay Systems. *Modeling, Identification and Control*, 2010. 31(4):145–164. doi:10.4173/mic.2010.4.3.

Di Ruscio, D. Discrete LQ optimal control with integral action: A simple controller on incremental form for MIMO systems. *Modeling, Identification and Control*, 2012. 33(2):35–44. doi:10.4173/mic.2012.2.1.

Godhavn, J.-M., Fard, M. P., and Fuchs, P. H. New slug control strategies, tuning rules and experimental re-

Godhavn, J.-M., Fard, M. P., and Fuchs, P. H. New slug control strategies, tuning rules and experimental results. *Journal of Process Control*, 2005. 15(5):547 – 557. doi:10.1016/j.jprocont.2004.10.003.

Jahanshahi, E. and Skogestad, S. Anti-slug control solutions based on identified model. *Journal of Process Control*, 2015. 30(0):58 – 68. doi:10.1016/j.jprocont.2014.12.007.

K-Spice. K-Spice. 2014. Kongsberg.com/k-spice.

LedaFlow. LedaFlow. 2014. Kongsberg.com/ledaflow.

Ogazi AI, Y. H. . L. L., Cao Y. Slug control with large valve openings to maximize oil production. *SPE Journal*, 2010. 15(3):812–821. doi:10.2118/124883-PA.

Skogestad, S. Feedback: Still the Simplest and Best Solution. *Modeling, Identification and Control*, 2009. 30(3):149–155. doi:10.4173/mic.2009.3.5.

Söderström, T. and Stoica, P. *System Identification*. Prentice Hall, 1989.

Sotomayor, O. A., Park, S. W., and Garcia, C. Multivariable identification of an activated sludge process with subspace-based algorithms. *Control Engineering Practice*, 2003. 11(8):961 – 969. doi:10.1016/S0967-0661(02)00210-1.

Storkaas, E. and Skogestad, S. Controllability analysis of two-phase pipeline-riser systems at riser slugging conditions. *Control Engineering Practice*, 2007. 15(5):567 – 581. doi:10.1016/j.conengprac.2006.10.007.

Storkaas, E., Skogestad, S., and Alstad, V. Stabilisation of desired flow regimes in pipelines. *AIChE Annual meeting, Chicago 1014 Nov., 1996, Paper 45f*, 2001.