



**NTNU – Trondheim**  
Norwegian University of  
Science and Technology

# Adaptive Control for Severe Slugging Mitigation

**Adrian Støbakk Finvold**

Chemical Engineering and Biotechnology

Submission date: June 2015

Supervisor: Sigurd Skogestad, IKP

Co-supervisor: Vinicius de Oliveira, IKP

Norwegian University of Science and Technology  
Department of Chemical Engineering



---

# Abstract

---

Offshore production facilities using pipeline-riser systems are experiencing terrain-induced cyclic flow instability due to its pipeline configuration and multiphased flow. The cyclic flow instability has been referred to as severe slugging. Anti-slug control using the top-side choke valve is time-variant and open-loop unstable in multiphased flow. The control systems available today are not robust and tend to become unstable over time, as a result of inflow disturbances or plant dynamic changes. This study investigates adaptive controller aiming to increase the robustness and restore the closed-loop performance while operating under unknown uncertainties, and thus drive the system towards a higher production rate.

Three types of adaptive systems was evaluated in order to increase the robustness of already existing optimal anti-slug control system. One self-tuning approach, and two Model Reference Adaptive Controllers (MRAC) - with increasing complexity was investigated. A comparison of an optimal tuned Proportional-integral-derivative-filter (PIDF) controller to Linear-Quadratic-Regulator (LQR) followed by Loop-Transfer-Recovery (LTR) without adaptation was also investigated.

There was a clear difference between self-tuning and the adaptive systems, in which a scalar reference model improved the robustness and outperformed the other adaptive systems. A self-tuning system was based upon updating the controller using the slope after system excitation, along with a desired closed-loop constant. The main drawback of self-tuning was that a miscalculation would result in instability with no point of return. A more complex observer-like reference model adaptive system was not efficient enough using only one measurement. This may be due to the significant measurement noise, which alters the error between the observing model and the actual process. The robustness, performance and stability margins had minor differences in comparison to an optimal tuned PIDF and Linear-Quadratic-Gaussian with Loop-Transfer-Recovery (LQG/LTR).



---

# Sammendrag

---

Offshore produksjonsanlegg som bruker stigerør systemer opplever syklisk ustabil strømning basert på rørdningskonfigurasjonen og flerfasig massetransport. Denne sykliske ustabiliteten blir referert til som severe slugging. Anti-slug kontrollsystemer som bruker en strupeventil på toppen av stigerøret er tidsvariante og ustabil ved åpen-sløyfe. Eksisterende kontrollsystemer tilgjengelige i dag er ikke robuste, og vil omsider ustabil som en konsekvens av usikkerhet i innstrømningen eller dynamiske endringer i prosessanlegget. Denne studien undersøker adaptive kontrollsystemer med mål i å øke robustheten og gjenvinne ytelsen ved lukket sløyfe, og dermed drive systemet mot en økt produksjonsmengde.

Tre ulike typer adaptive systemer har blitt undersøkt for å øke robustheten av allerede eksisterende anti-slug kontrollsystemer. En av systemene innebar et selvjusterende system, og to modell referanse baserte systemer. En sammenligning av en optimalt tunet PIDF kontroller mot LQR, etterfulgt av LTR uten adaptiv prosedyre ble også undersøkt.

Det var en klar forskjell mellom selvjusterende og de adaptive systemene, der en skalar referansemodell utklasserte de andre modellene. Det selvjusterende systemet var basert på en oppdatering av kontrolleren ved å bruke stigningskurven etter en eksitasjon av prosessen, i tillegg til en ideell tidskonstant. Den største utfordringen med selvjusteringen var at en feilberegning kunne resultere i ustabilitet uten mulighet for gjenvinning av stabilitet. Et mer komplekst adaptive system ved bruk av observator referansemodell var ikke effektiv nok med kun én måling. Dette kan skyldes den betydelige målestøyen som opptrer i måledataene. Robusthet, ytelse og stabilitetsmarginer viste mindre forskjeller i sammenligning av en optimalt tunet PIDF og med LQG/LTR.



---

# Acknowledgements

---

*“We can only see a short distance ahead, but we can see plenty there that needs to be done.”*

— Alan Turing, *Computing machinery and intelligence*

Supervisory is described according to Wikipedia as a process that accepts an instruction, carries it out autonomously, reports the results and awaits further commands. This is, nevertheless, describing an extra control layer to adaptive system within control theory. With that being said, it also describes the relationship between a student and its supervisors.

I would like to express great appreciation to my supervisor *Sigurd Skogestad* for his brilliance within the field of control theory. As Sigurd is always available for a theoretical discussion, a communication delay in this supervisory control is negligible. As a student with a highly dynamic learning curve, Sigurd’s level of expertise is regulating it to a more somewhat linear curve.

I wish to express my sincere thanks to the adaptive control guru, soon-to-be PhD, *Vinicius De Oliveira*. His expertise within robust adaptive systems is inspiring, and his valuable guidance was encouraging to a student with its poles located far into the RHP.

A warm and special thanks to my girlfriend, *Ida*, for being understanding during days with excess overshoots with respect to other priorities. The integrated error was, in periods, increasing during excessive work, which at some events would saturate followed by integral windup.

Thanks to the group at process-system engineering for good motivation during days filled with stress and lack of progression. The group has been a good moral support during days with unstable motivation and in complex situations.



---

# Declaration of compliance

---

I hereby declare this thesis as an independent work in agreement with the exam rules, and regulations of the Norwegian University of Science and Technology.

June 10, 2015  
*Trondheim, Norway*



---

Adrian Finvold

---

# Contents

---

<b>Abstract</b>	<b>i</b>
<b>Sammendrag</b>	<b>iii</b>
<b>Contents</b>	<b>viii</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Severe slugging in flow line-riser systems . . . . .	1
1.2 Previous work . . . . .	3
1.2.1 Controllability . . . . .	4
1.3 Optimal controller . . . . .	5
1.3.1 Proportional-Integral-Derivative controller . . . . .	5
1.3.2 Linear-Quadratic-Regulator . . . . .	6
1.4 Adaptive control . . . . .	7
1.4.1 Self-tuning . . . . .	8
1.4.2 Reference model based adaption (MRAC) . . . . .	8
1.5 Digital signal filtering . . . . .	9
<b>2 Aims and Objectives</b>	<b>11</b>
2.1 Specific objectives . . . . .	11
<b>3 Methods</b>	<b>13</b>
3.1 Experimental slugging setup . . . . .	13

---

3.2	Simulation model . . . . .	15
3.2.1	Theoretical background . . . . .	15
3.3	Self-tuning based on response slope . . . . .	15
3.3.1	Theoretical design . . . . .	15
3.3.2	Implementation . . . . .	19
3.4	Scalar reference model adaptive system . . . . .	20
3.4.1	Theoretical design . . . . .	20
3.4.2	Implementation . . . . .	24
3.5	Augmented state observer-like reference model . . . . .	25
3.5.1	Theoretical design . . . . .	25
3.5.2	Implementation . . . . .	33
3.6	Supervisory adaptation . . . . .	36
3.7	Evaluation of stability and robustness limitations . . . . .	37
<b>4</b>	<b>Results</b>	<b>39</b>
4.1	Controllability and System dynamics with the four state model . . . . .	40
4.2	Self-tuning system based on response slope . . . . .	42
4.3	Adaptive system based on scalar reference model . . . . .	45
4.3.1	Simulation . . . . .	45
4.3.2	Experimental . . . . .	47
4.4	Adaptive system based on augmented state observer reference model . . . . .	49
4.4.1	Simulations . . . . .	54
4.4.2	Experimental . . . . .	57
<b>5</b>	<b>Discussion</b>	<b>61</b>
5.1	Simplified four-state model simulation . . . . .	62
5.2	Self-tuning system based on response slope . . . . .	62
5.3	Adaptive system based on scalar reference model . . . . .	63
5.4	Adaptive system based on observer-like reference model . . . . .	64
<b>6</b>	<b>Conclusion</b>	<b>67</b>
6.1	Future work . . . . .	69
	<b>Bibliography</b>	<b>71</b>
	<b>Appendices</b>	<b>74</b>
	<b>A Additional results for self-tuning system</b>	<b>75</b>
	<b>B Matlab scripts</b>	<b>79</b>

B.1	Simplified four state model . . . . .	79
B.2	Tune PID with use of set point response . . . . .	79
B.3	Tune PID with use of LQR-LQG/LTR . . . . .	83
B.4	Projection Operator . . . . .	87
B.5	Check for instability with cross-correlation . . . . .	88
B.6	Check for instability using Integrated Absolute Error . . . . .	91
B.7	Estimate frequency of peak . . . . .	92

---

# List of Figures

---

1.1	Steps involved in severe slugging for a pipeline-riser configuration . . . .	2
1.2	Illustrations of flexible riser configurations . . . . .	3
1.3	Illustrative example presenting the states used in the four-state model .	4
1.4	Correlation between topside choke valve and subsea pressure . . . . .	5
1.5	Illustrative example presenting the Lyapunov stability theorem . . . . .	9
3.1	Experimental slug model setup at NTNU . . . . .	14
3.2	Control scheme illustrating the SISO system . . . . .	14
3.3	Comparison of a approximated fourth order system and the actual system	16
3.4	Example of high-frequent noise present in measurement data . . . . .	17
3.5	Block scheme presenting the self-tuning system . . . . .	19
3.6	Location of poles and zeros in the applied discrete Butterworth filter . .	20
3.7	Block scheme presenting the scalar reference adaptive system . . . . .	25
3.8	Block scheme of LQG adaptive system with Kalman filter state estimator	29
3.9	Dead-zone modification illustrated . . . . .	32
3.10	Block scheme of observer-based adaptive system . . . . .	33
3.11	Cross-correlation for a given signal to determine stability . . . . .	37
4.1	Correlation between process gain and valve opening . . . . .	40
4.2	Location of system Poles and Zeros with respect to valve opening . . . .	41
4.3	Simulative result: Bode plot of open-loop system with optimal controller (PIDF) . . . . .	41
4.4	Experimental result: Fixed controller using optimal PIDF . . . . .	42
4.5	Experimental result: Pressure measurement with and without filtering .	43
4.6	Experimental result: Self-tuning system updating $K_p$ and $\tau_i$ optimal PIDF controller in standard form . . . . .	44

4.7	Experimental result: Change of adaptive parameters, $K_p$ and $\tau_i$ using self-tuning system in standard form . . . . .	44
4.8	Simulative result: Fixed controller with optimal PIDF . . . . .	45
4.9	Sumulative result: Adaptive system based on scalar reference model . . . . .	46
4.10	Simulative result: Change of adaptive parameters, $K_u$ , $K_r$ and $K_u$ . . . . .	46
4.11	Experimental result: Fixed controller using optimal PIDF controller . . . . .	47
4.12	Result: Experimental adaptive system using scalar reference model . . . . .	48
4.13	Experimental result: Change of adaptive parameters, $K_x$ , $K_r$ and $K_u$ . . . . .	48
4.14	Simulative result: Relationship between delay margin, settling time and maximum controller output with respect to tuning parameter, $q$ , for optimal controller, LQR . . . . .	49
4.15	Simulative result: Bode plot of open-loop dynamics with optimal (LQR) controller with different tuning values . . . . .	50
4.16	Simulative result: Bode plot of optimal LQG/LTR controller with different tuning values . . . . .	51
4.17	Simulative result: Comparison of Bode plot with optimal LQR - and LQG/LTR controller . . . . .	52
4.18	Simulative result: Comparison of Sensitivity and Complementary Sensitivity for optimal PIDF, LQR and LQG/LTR controller . . . . .	53
4.19	Simulative result: Fixed controller using LQG/LTR controller . . . . .	54
4.20	Simulative result: Simulations of observer-like reference model with LQG/LTR adaptive system . . . . .	55
4.21	Simulative result: Change of adaptive gains using observer-like reference model LQG/LTR . . . . .	55
4.22	Simulative result: Presentation of absolute error between observing model and process model . . . . .	56
4.23	Experimental result: Fixed controller using LQR controller . . . . .	57
4.24	Experimental result: Presentation of absolute error between observing model and process model . . . . .	58
4.25	Experimental result: Adaptation using LQG/LTR . . . . .	58
4.26	Experimental result: Change of adaptive gains using observer-like reference model LQG/LTR . . . . .	59
A.1	Experimental result: Additional result of self-tuning system updating $K_p$ in standard form . . . . .	75
A.2	Experimental result: Additional result of self-tuning system updating $K_p$ in parallel form . . . . .	76
A.3	Experimental result: Additional result of self-tuning system updating $K_p$ and $K_i$ in parallel form . . . . .	77

---

A.4	Experimental result: Change of adaptive parameter $K_p$ using self-tuning system in standard form . . . . .	77
A.5	Experimental result: Change of adaptive parameter $K_p$ using self-tuning system in parallel form . . . . .	78
A.6	Experimental result: Change of adaptive parameter $K_p$ and $K_i$ using self-tuning system in parallel form . . . . .	78

---

# List of Tables

---

1.1	Optimal PIDF controller with respect to all parameters . . . . .	6
4.1	Experimental set point pressures at given time . . . . .	39
4.2	Stability margins and robustness analysis with respect to tuning parameter $q$ and optimal quadratic controller LQR . . . . .	50
4.3	Stability margins and robustness analysis with respect to tuning parameter $v$ and LQG/LTR . . . . .	51
6.1	Conclusive result: Robustness, Performance and Stability analysis with respect to control designs . . . . .	68
6.2	Conclusive result: Comparison of self-tuning - and adaptive systems . .	68

---

# Acronyms

---

**ARE** *Algebraic Riccati Equation*

**IAE** *Integrated Absolute Error*

**LQG** *Linear-Quadratic-Gaussian*

**LQG/LTR** *Linear-Quadratic-Gaussian with Loop-Transfer-Recovery*

**LQR** *Linear-Quadratic-Regulator*

**LTFM** *Loop Transfer Function Matrix*

**LTI** *Linear Time-invariant*

**LTR** *Loop-Transfer-Recovery*

**MIMO** *Multiple Inputs Multiple Outputs*

**MPC** *Model Predictive Control*

**MRAC** *Model Reference Adaptive Controller*

**PID** *Proportional-integral-derivative*

**PIDF** *Proportional-integral-derivative-filter*

**RHP** *Right-half plane*

**RPS** *Robustness, Performance and Stability*

**SISO** *Single Input Single Output*

**SPR** *Strictly Positive Real*

---

# List of symbols

---

Symbol	Description	Unit
$\hat{x}$	<i>Estimated system states</i>	-
$\hat{y}$	<i>Estimated pressure output</i>	kPa
$r_{cmd}$ or $r$	<i>Set point</i>	kPa
$\dot{x}$	<i>State with respect to time</i>	-
$\dot{y}$	<i>Output with respect to time</i>	kPa/s
$\mathcal{C}$	State Controllability Matrix	-
$K_p$	<i>Adaptive gain w.r.t. state</i>	-
$K_u$	<i>Adaptive gain w.r.t. controller usage</i>	-
$K_r$	<i>Adaptive gain w.r.t. set point</i>	-
$K_x$	<i>Adaptive gain w.r.t. pressure</i>	-
$L_v$	<i>Observer gain</i>	-
$q$	Penalty parameter (LQR)	-
$v$	Tuning parameter (LQG/LTR)	-
$\tilde{\bullet}$	Augmented matrix	-
$u_{bl}$	<i>Baseline controller output</i>	-
$u_{ad}$	<i>Adaptive controller output</i>	-
$Z$	<i>Valve opening</i>	%

---

---

## Greek letters

---

Symbol	Description	Unit
$k$	System gain	-
$\tau$	Time constant	sec
$\theta$	System delay	sec
$\kappa$	Adaptive gain (Slope adaptive system)	-
$\gamma$	Tuning parameter (Rate of adaption)	-
$\eta$	PIDF filter coefficient	-
$\omega$	Temporal Frequency	Hz
$\omega_c$	Cutoff Frequency	rad/s
$\Phi$	Lipschitz continuous vector regressor	-
$\epsilon$	Boundary limit	-
$\Theta_{max}$	Projecting boundary	-
$\mu$	Dead-zone modification	-



# Introduction

---

### 1.1 Severe slugging in flow line-riser systems

Floating facilities are able to move laterally with the wind and waves. The pipeline risers are, however, fixed to the seafloor, which implies that a certain displacement may occur between the riser pipeline and the connection point to the subsea reservoir. Various mechanisms and configurations are used to expand and contract the pipeline riser movements in reference to the floating facility. Some mechanisms used to act as motion compensator are e.g. buoyancy cans attached to the flexible pipeline. These buoyancy cans contribute to ensure that the pipeline may withstand the dynamic tension and stress level. Some of the configurations applied today are exemplified in Figure 1.2.

Slugging are most frequently caused by the flow line geometry and uneven fluid distribution among phases within the equipment. These properties are inducing multiphase flow velocity differences and instability in gas lift flow. This multi phased flow may include oil, water, gas, sand, sediments, and other solid compounds such as wax, asphaltenes and naphthenates among others.<sup>1</sup> The effect of these phases combined with the modified pipeline configuration induces terrain-induced slug flow cycles with large pressure and flow variations.<sup>2</sup> A slug is a mixture of liquid and solid components which are forced out of the pipeline due to a high pressurized gas pocket. Most oil reservoirs produces gas, which will accumulate at the highest point, while the liquid will form a liquid blockage at the lowest point of the pipeline.<sup>3</sup> The accumulated gas will eventually build up and violently burst out through the pipeline. A step wise schematic presentation of severe slugging is presented in Figure 1.1. In this figure there are four stages summarized, in which can be explained as following:

- **Step a)** - Gas accumulates at the highest point, and consequently liquid at the lowest point due to density differences.
- **Step b)** - The gas pressure is lower than the hydrostatic pressure. The liquid is blocking the riser entrance, and prevents the gas from escaping. Gas pressure will continue to increase.
- **Step c)** - When the gas pressure exceeds the hydrostatic pressure, the gas will push the liquid out of the riser - causing a violent blowout.
- **Step d)** - All the available liquid in riser has been pushed out, and will eventually fall back into the lowest point and the process will repeat.

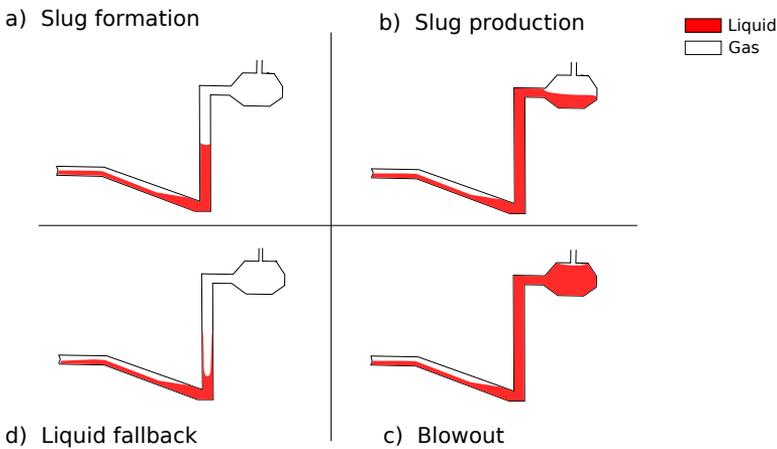


Figure 1.1: Steps involved in terrain-induced slug flow cycle for a pipeline-riser configuration with a separator at topside of the riser pipe. Liquid presented in red, and gas in white.

Unstable flow may, consequential, lead to poor separation followed by heat exchanger damage, while unstable pressure may lead to high level trips in compressor or separators. The economic losses due to maintenance cost and equipment replacement are substantial. Some production facilities are using a "slug catcher" to cope with the oscillating mass flow. One of the disadvantages of using a slug catcher at the production facility is the large size and buffer volume necessary to handle both the violent flow volumes and - velocities.<sup>2</sup> The main goal in production industries is to maximize the production rate, while still maintaining system stability.

An automatic control system is necessary to ensure a stable and steady flow - also known as a Flow Assurance system. The main concept of flow assurance is to ensure a successful - and economical flow from reservoir to point of interest. The financial losses due to a failure in the flow assurance system may be enormous.

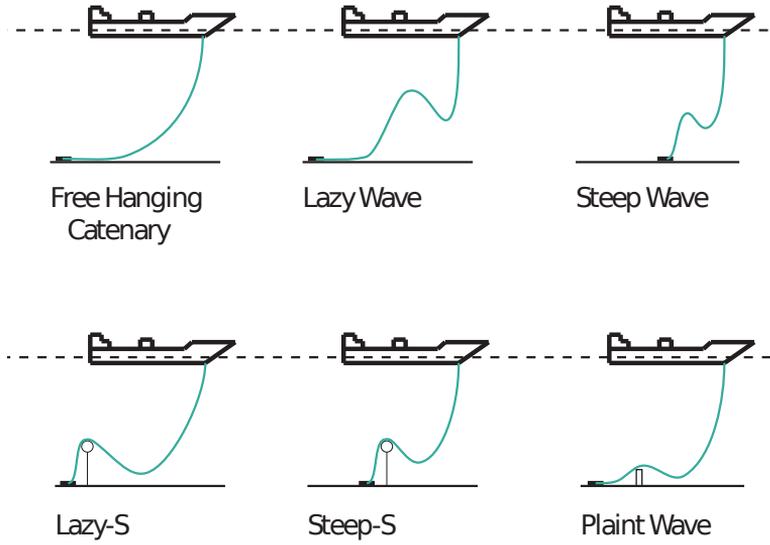


Figure 1.2: Exemplified common used flexible riser configurations.<sup>4</sup>

There are several approaches to cope with inflow disturbance and plant changes, but they are not very robust.<sup>5</sup>

## 1.2 Previous work

A simplified four-state model describing the cyclic slug flow was previously developed by Jahanshahi and Skogestad<sup>6</sup>. This model describes the system with a relative good match with few tuning parameters. The four-state equations are presented in Equations (1.1a)-(1.1d). The model is simplified in terms of only two phases; namely liquid, and gas. The liquid phase is calculated with the properties of water.

$$\frac{dm_{g,p}}{dt} = w_{g,in} - w_{g,r} \quad (1.1a)$$

$$\frac{dm_{l,p}}{dt} = w_{l,in} - w_{l,r} \quad (1.1b)$$

$$\frac{dm_{g,r}}{dt} = w_{g,r} - \alpha w \quad (1.1c)$$

$$\frac{dm_{l,r}}{dt} = w_{l,r} - (1 - \alpha)w \quad (1.1d)$$

Were:

- $m_{g,p}$ : mass of gas in pipeline
- $m_{l,p}$ : mass of liquid in pipeline
- $m_{g,r}$ : mass of gas in riser
- $m_{l,r}$ : mass of liquid in riser
- $\alpha$ : gas mass fraction

The flow rates of gas,  $w_{g,r}$ , and liquid,  $w_{l,r}$ , from the pipeline to the riser are determined by a pressure drop across the riser-base where they are described by virtual valve equations. The outlet mixture flow rate,  $w$ , is determined by the percentage opening of the top-side choke valve,  $Z$ . The flow rates and actual gas mass fraction,  $\alpha$  are previously presented by Jahanshahi and Skogestad<sup>6</sup>. A linearized form of this model will be the basis for the control design throughout this thesis.

An illustrative presentation of the states with reference to experimental configuration set up is presented in Figure 1.3.

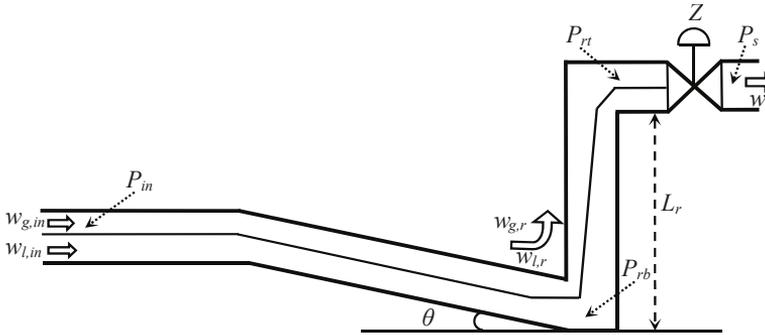


Figure 1.3: A schematic presentation of the notation of states as described in Equation (1.1). Illustrative graphics from previous work of Jahanshahi et al.<sup>5</sup>

### 1.2.1 Controllability

Automatic control using the top-side choke valve,  $Z$ , can eliminate severe slugging, but increases the back pressure and decreases the production rate. Control of the choke valve in combinations with pressure measurements is the recommended solution to prevent severe slugging flow conditions at offshore oilfields.<sup>5</sup>

With the inlet pressure,  $P_{in}$  as measurement variable, Jahanshahi et al.<sup>5</sup> found the critical opening at the top-side choke valve to be  $Z^* = 15\%$ . This implies that a valve opening  $Z > 15\%$  results in unstable, oscillating system without closed-loop control. A experimental bifurcating plot with data from Jahanshahi and

Skogestad<sup>7</sup> is presented in Figure 1.4. This plot describes the oscillating max/min pressure (solid line), and the static gain of the system (dashed line) with respect to given valve opening,  $Z$ . The gain of the system is described as  $k = \frac{dy}{du} = \frac{dP_{in}}{dZ}$ . This implies an inverse response in which a reduced set point pressure,  $P_{in}$ , leads to a larger topside valve opening,  $Z$ , to maintain constant system gain.

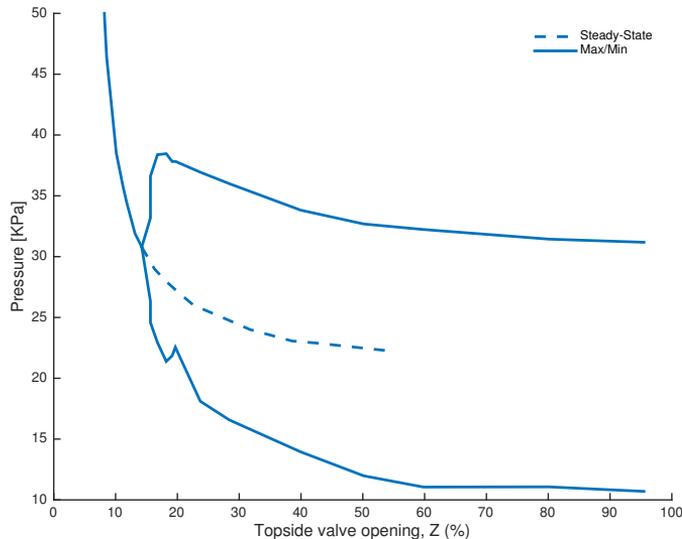


Figure 1.4: Bifurcating plot with experimental data obtained from Jahanshahi and Skogestad<sup>7</sup>. A valve opening,  $Z$ , larger than 15% causes the inlet pressure,  $P_{in}$  (see Figure 1.3) to oscillate between the solid lines describing the maximum and minimum pressure. The dashed line is the static system gain at steady-state.

## 1.3 Optimal controller

### 1.3.1 Proportional-Integral-Derivative controller

The Proportional-integral-derivative (PID) is generally based on minimizing a certain error between desired operating point, and the actual process output.

The PI and PID controllers are the most widely used controller in the industries. The PID controllers are superior in combination of performance and robustness, with respect to its relative simplicity. PID controller is often represented as the fundamental component, integrated into more sophisticated control schemes - such as adaptive systems and Model Predictive Control (MPC).<sup>8</sup> A filter part may be added to the controller to prevent rapid randomized noise from entering to the controller and alter the output. This is known as a Proportional-integral-derivative-filter (PIDF) controller.<sup>9</sup> The optimal PIDF controller with a valve opening of  $Z =$

30% as nominal operating condition, was found by Jahanshahi and Skogestad<sup>7</sup>, and is presented in Table 1.1. A description of a PIDF controller is defined in Equation (1.2a) and (1.2b) in two different structures.

$$u(t) = K_c \left( e(t) + \frac{1}{\tau_i} \int_0^t e(s) ds + \frac{\tau_d \dot{e}(t)}{\frac{\tau_d}{\eta} \dot{e}(t) + 1} \right), \quad \text{Standard form} \quad (1.2a)$$

$$u(t) = K_c e(t) + K_i \int_0^t e(s) ds + K_d \frac{\dot{e}(t)}{\tau_f \dot{e}(t) + 1}, \quad \text{Parallel form} \quad (1.2b)$$

were:  $e = y_s - y = r - y$ , and  $\tau_f = \frac{\tau_d}{\eta}$ . The standard form of the PIDF controller defines a filter coefficient,  $\eta = \tau_d / \tau_f$ , which defines the location of the pole in the derivative filter.<sup>10</sup>

$$K_{c_{std}} = K_{c_{par}}, \quad \tau_{i_{std}} = \frac{K_{c_{par}}}{K_i}, \quad \tau_{d_{std}} = \frac{K_d}{K_{c_{par}}}, \quad \tau_{f_{par}} = \frac{K_d / K_{c_{par}}}{\eta} \quad (1.3)$$

Transformation of parameters from Ideal - (1.2a) to parallel PID form (1.2b) are described in Equation (1.3).

Table 1.1: Optimal PIDF controller with respect to all parameters found by Jahanshahi and Skogestad<sup>7</sup>.

Standard form	$K_c$	$\tau_i$	$\tau_d$	$\eta$
	-3.09	1.9	60.4	15.1
Parallel form	$K_c$	$K_i$	$K_d$	$\tau_f$
	-3.09	-1.62	-187	4

### 1.3.2 Linear-Quadratic-Regulator

The LQR design problem is to design a controller such that a cost function,  $J$  is minimized in terms of weight factors in reference to controller output, and speed of system response.<sup>11</sup> A state space representation is used upon designing the LQR controller. The Linear-Quadratic-Regulator (LQR) is one of the most commonly used control design within aerospace.<sup>12</sup> Previous work has shown that the LQR controller may outperform the general PID with certain system dynamics.<sup>13,14</sup>

With all the states available for measure, the LQR design generally gives good performance with good stability margins. One of the main reasons for the Linear-Quadratic-Regulator attractiveness within the industrial application, is the guaranteed frequency properties with a minimum gain margin of  $[-6, \infty]$ , and phase margin of at least  $60^\circ$ .<sup>15</sup> The general cost function of LQR design is presented in Equation (1.4).<sup>11</sup>

$$J_{LQR} = \int_0^{\infty} x^T Q x + u^T R u dt \quad (1.4)$$

were the penalty factor  $Q \in \mathbb{R}^{n \times n}$  and  $R \in \mathbb{R}^{m \times m}$  are symmetric positive-definite matrices. The purpose of the cost function is to minimize the "energy" of both the controlled output and the control signal. The tuning parameters;  $Q$  and  $R$ , weights the cost function in terms of state or control output, respectively.

In cases where the states are not available for feedback, a dynamic compensator is required to predict the states - also known as observing design. The physical states are then estimated with use of the actual measured variable, and the states are indirectly calculated in combination with an adequate model representation. When an optimal controller (LQR) is combined with an optimal state estimator (Kalman filter), the control design is referred to as a Linear-Quadratic-Gaussian (LQG) problem.

In order to prevent degradation of frequency domain properties, such as gain -, phase and delay margins of the design with state estimation, a recovery design is required. The recovery is known as Loop-Transfer-Recovery. A combination of both state estimation and a frequency domain recovery is referred to as Linear-Quadratic-Gaussian with Loop-Transfer-Recovery (LQG/LTR). The Kalman filter is then no longer recognized as an optimal state estimator, but a dynamic compensator/observer tuned for optimal robustness and performance.<sup>15</sup> The observer gain is, in other words, not optimized for noisy measurements, but with respect to optimal Loop-Transfer-Recovery (LTR).

## 1.4 Adaptive control

An optimal controller is designed and linearized around a certain operation conditions. The performance of this controller is expected to be robust and stable in presence of disturbance within these conditions. If the operating conditions are changed, the uncertainties will change, which implies that the system baseline controller will deteriorate from the optimal closed-loop performance. The main objective of the adaptive controller is to maintain consistent performance of the closed-loop system in presence of uncertainties and disturbances.

### 1.4.1 Self-tuning

Self-tuning differs from adaptive on the basis of that an adaptive is on-line at all time. Self-tuning is more an one-time event adjustment after a certain plant stimulus. A set point change followed by a controller retune is considered a one-time event - unless the system is continuously exciting the process plant in order to retune. This distinction is, however, debated, but the proposed definition above is consistent throughout this thesis.

### 1.4.2 Reference model based adaption (MRAC)

The main idea of Model Reference Adaptive Controller is to specify the desired performance of a command-tracking system. This would eventually define the ideal response of the system output due to external commands. The ideal reference model design should represent the actual system sufficient enough to be used as a reference to measure and control the physical states. In other words an "explicit model following"-system. In essence, a Model Reference Adaptive Controller (MRAC) system exert a command or a set point, and drives both the plant and reference model towards desired trajectories. The difference between the actual plant and the reference model defines the tracking error, which is actively used in the adaptive laws for correction. The MRAC design requires the adaptive laws to preserve closed-loop stability. This could be described by Lyapunov stability theorem.

Lyapunov stability theorem defines how an energy-like function, a so-called Lyapunov function, changes over time. The energy-like function contributes to a conclusion whether the system is stable without solving the differential equation. Stability of a dynamic system can be addressed as the trajectories of the system under small perturbations of the initial conditions over an infinite time interval. The notion of stability requires that the continuity property hold infinitely in time. Lyapunov stability is illustrated with Figure 1.5. In this figure the solution of an arbitrary differential equation is defined as  $x(t; x_0)$ , with the initial condition  $x(t_0) = x_0$ . This solution is unique and exists on a finite open-ended interval  $[t_0, T)$ . The changed system property due to initial values can be described as follows: Given any positive constant  $\epsilon > 0$  there exists a sufficiently small positive constant  $\delta > 0$ . For all perturbed initial conditions  $(x_0 + \Delta x_0)$  were  $|\Delta x_0| \leq \delta$ . The corresponding perturbed solution  $x(t; x_0 + \Delta x_0)$  deviates from the original system by no more than  $\epsilon$ . In other words,  $\|x(t; x_0 + \Delta x_0) - x(t; x_0)\| \leq \epsilon \quad \forall (t_0 \leq t < T)$ . This is further demonstrated in Figure 1.5, were the red line illustrates the perturbed initial conditions and are within the  $2\epsilon$ -strip. Lyapunov stability requires the system to orbit within some given boundary, marked as a dashed line in Figure 1.5.<sup>16</sup>

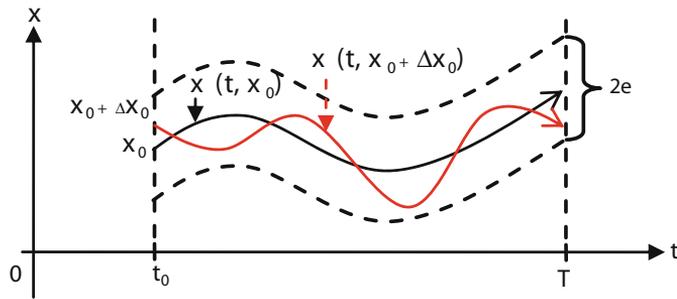


Figure 1.5: Lyapunov Stability illustrated: Continuity of system solutions with respect to initial conditions. The red line illustrates the system trajectories with perturbed initial conditions were it is, per definition, stable as long as the trajectory stays within the  $\epsilon$  boundary marked in dashed lines.<sup>15</sup>

## 1.5 Digital signal filtering

A digital filter is a system which performs mathematical operations on continuous or sampled, discrete-time signals to reduce or amplify certain conditional frequency range. The mathematical representation of the filter is determining the behavior of the filter. In control theory the signal obtained from the system may be influenced by high frequent noise in which a low frequency-pass filter is required to present a smoother form of the signal. The low-pass filter passes signals with a frequency lower than a certain frequency cutoff, which eliminates the high frequency. A low-pass filter is often used with discrete data containing high frequent noise, in combination with PID/PD-controller. The derivative part in the controller may greatly amplify the short-term signal fluctuations without the filter.<sup>17</sup> The low-pass filter enables the interpretation of the signal to represent the signal fluctuations more accurate and improves the subsequent calculations.<sup>18</sup>



## Chapter 2

---

# Aims and Objectives

---

Severe slugging has a negative impact on the operation of offshore production facilities which may, in the worst cases, cause a plant shutdown. Control systems available today are not robust and tend to become unstable over time, as a result of inflow disturbances or plant dynamic changes. Automatic control using the top-side choke valve can eliminate severe slugging, but increases the back pressure and decreases the production rate. The economical benefit of a high production rate combined with an open-loop unstable system at large choke valve opening reveals an area, which need to be optimized and driven to its maximum. An optimal controller is designed and linearized around a certain operation conditions. The performance of this controller is expected to be robust and stable in presence of disturbance within these conditions. If the operating conditions are altered, the uncertainties will change, which implies that the system baseline controller will deteriorate from the optimal closed-loop performance. The main objective with an adaptive controller is to increase the robustness and restore the closed-loop performance while operating under unknown uncertainties, and thus drive the system toward a higher production rate. This was the motivational fundament behind this thesis.

### 2.1 Specific objectives

The aim is to find a simple and robust adaptive system to mitigate the cyclic slug flow. One self-tuning method, and two different model reference adaptive controllers will be investigated. This work is a continuation of previous research by Jahanshahi et al.<sup>5</sup> on the slugging rig installed at NTNU, with the simplified

four-state model. This model is used to evaluate both the system behavior and the estimated adaptive controller. The control structure throughout this thesis will be pressure measurement at pipeline inlet to control the topside choke valve. This control structure has by previous work shown to be controllable.<sup>19</sup> To the authors knowledge, there has not been conducted any previous adaptive experiments using a Single Input Single Output in comparison between optimal PID and LQG/LTR on the slugging rig.

Adaptive systems will be step-wise implemented with increasing complexity in both simulation and experimentally. First step is a self-tuning system that will be presented and implemented experimentally, followed by a scalar reference model implemented simulative and experimentally. Both methods will be in use of an optimal tuned PIDF as a baseline controller. Further on, a more complicated adaptive system will be presented and implemented both in simulations and experimental. The baseline controller in this adaptive system will include an optimal LQR-controller in combination with Loop-Transfer-Recovery. Common to all adaptive approaches is that only one measurement will be available to regulate the system. Stability margins of an optimal tuned PIDF controller to LQR followed by LTR without adaptation will also be investigated.

## Chapter 3

---

# Methods

---

Implementation and comparison of one self-tuning method, and two types of adaptive systems - with increasing complexity was evaluated. The adaptive systems was simulated before they were implemented into the actual process. The adaptive concepts based on reference models in this thesis are inspired by Lavretsky and Wise<sup>15</sup> and their extensive work within the field of aerospace. The MATLAB models are modified from De Oliveira et al.<sup>20</sup> previous work with Multiple Inputs Multiple Outputs (MIMO) on the severe slugging problem.

### 3.1 Experimental slugging setup

The experiments where performed on a laboratory for anti-slug control at the Chemical Engineering Department of NTNU. A schematic presentation of the experimental setup of the slug rig is presented in Figure 3.1. The pipeline and the riser are made from flexible pipes with  $2\text{cm}$  inner diameter. The length of the pipeline is  $4\text{m}$  with  $15^\circ$  incline angle. The height of riser is  $3\text{m}$ . A buffer tank is installed to simulate the effect of a long pipe with the same volume, such that the total pipe length would be about  $70\text{m}$ .

In Figure 3.1 the buffer pressure ( $P_1$ ) is used as measured variable with the top side choke valve as input for control. A controller system in form of a Proportional-integral-derivative(PID) - or a Linear-Quadratic-Regulator(LQR) controller is coupled with pressure data. This implies a Single Input Single Output (SISO) system, leaving one degree of freedom for control. An illustration of the control scheme is presented in Figure 3.2. In this figure the pressure control,  $PC$ , sends the signal to a pressure transmitter,  $PT$ , which is used as direct input to the controller. In this

case noted as a PID. The measurement frequency is limited to,  $f_s = 10Hz$ . The pressure within the separator after the topside choke valve is nominally constant at atmospheric pressure. The feed into the pipeline is assumed to be at a constant flow rate of  $4 \frac{L}{min}$  of water, and  $4.5 \frac{L}{min}$  of air.

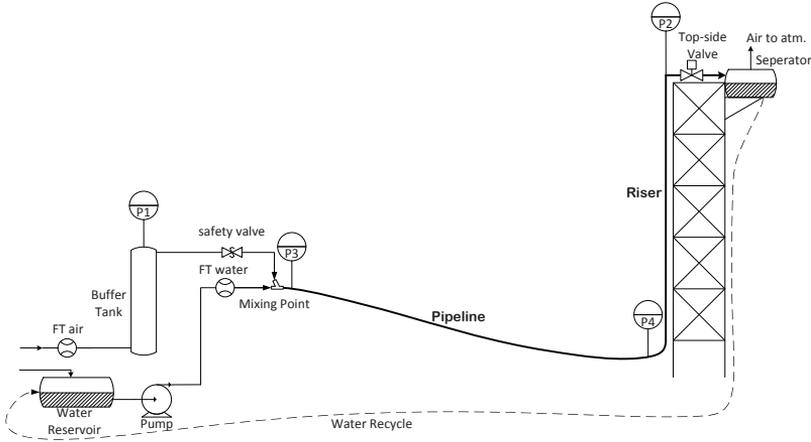


Figure 3.1: Experimental slug model setup at NTNU. Measured variable is in this figure  $P_1$ , and the controlled variable is the topside choke valve at top of riser. Buffer tank is installed to simulate the effect of a long pipe, such that the total pipe length would be about 70m. Figure obtained by previous work conducted by Jahanshahi and Skogestad<sup>7</sup>

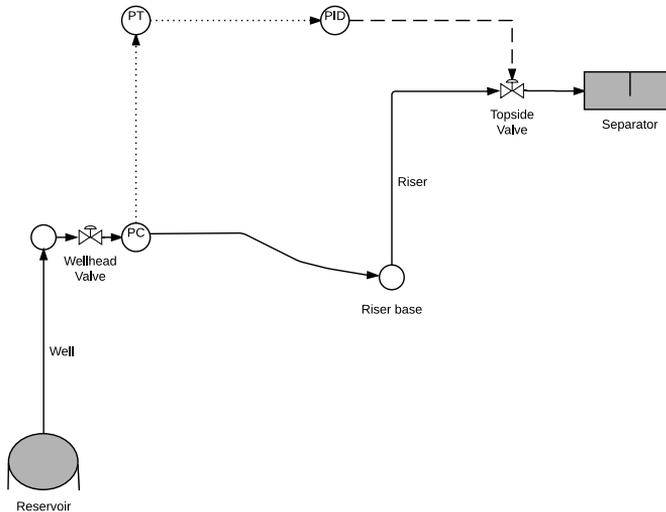


Figure 3.2: Illustration of a SISO system with pressure control at well top, and topside choke valve as controlled variable.

## 3.2 Simulation model

### 3.2.1 Theoretical background

The four state model used for simulation were developed by Jahanshahi et al.<sup>5</sup>. This model was used to map the relationship between system gain, poles and zeros, with respect to valve opening. A state space linearized at a valve opening of  $Z = 30\%$  was used to determine the state controllability and through design of MRACs, based on previously optimal control design at these operation conditions.

A system is controllable if the controllability matrix  $\mathcal{C}$  has full row rank. That is the largest possible number of linearly independent rows in a matrix.<sup>21</sup>

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \quad (3.1)$$

where  $n$  is the number of states.

The simulation was performed using MATLAB and Simulink.

## 3.3 Self-tuning based on response slope

### 3.3.1 Theoretical design

The system response after a set point change may be used to calculate the time constant. This self-tuning system is based on using the estimated closed-loop time constant,  $\tau_c$ , to update the controller. In order to determine the closed-loop time constant, a simple general method was implemented. The closed-loop time constant was calculated with an induced excitation of the system. The maximum slope in a step response of the system was used to estimate the time constant. This is a well known procedure, but are known to only be valid for a first order plus time delay systems.<sup>22</sup> This evaluation is thus a first order approximation of the slugging system.

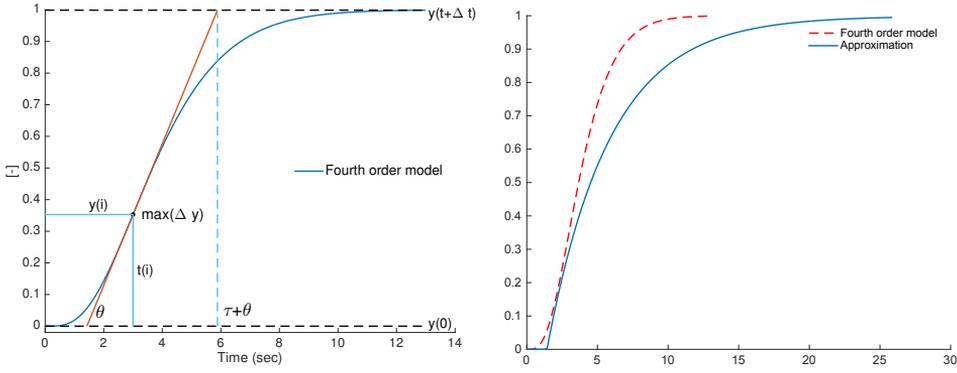
An example of this technique can easily be explained with an arbitrary fourth-order system described as  $G(s) = \frac{1}{(s+1)^4}$ . A step response to this system is presented in Figure 3.3(a).

The time constant,  $\tau$ , gain,  $k$ , and delay,  $\theta$  can be estimated with use of Equation (3.2)-(3.5).

$$k = \frac{y(t + \Delta t) - y(0)}{\Delta u} \quad (3.2)$$

$$\tau = \frac{|y(t + \Delta t) - y(0)|}{\max(\Delta y)} \quad (3.3)$$

$$\theta = t(i) - \frac{|y(i) - y(0)|}{\max(\Delta y)} \quad (3.4)$$



(a) Step response of  $u = 1$ , of an arbitrary fourth order system described as  $G(s) = \frac{1}{(s+1)^4}$  marked in blue. The red line marks the tangent to  $\max(\Delta y)$ . Blue dashed line marks the time constant,  $\tau$ .

(b) Comparison of a fourth order model (dashed red line) described as  $G(s) = \frac{1}{(s+1)^4}$ , and a approximated first order model plus time delay (solid blue line) with gain,  $k = 0.9989$ , time constant,  $\tau = 4.4588s$ , and delay,  $\theta = 1.4254$ .

Figure 3.3: Induced set point change to a fourth order model. Open loop time constant,  $\tau$ , gain,  $k$  and delay,  $\theta$  are estimated using the slope of the response. The fourth order system is compared with the estimated parameters with actual system response,  $G(s)$ .

were the  $u$  is the set point command,  $i$  is the index of maximum slope and is calculated as expressed in Equation (3.5).

$$i = \left\{ j : \max \frac{\Delta y(j)}{\Delta t(j)} \right\} \quad (3.5)$$

From Figure 3.3(a), the system gain was calculated to be,  $k = 0.9989$ , time constant,  $\tau = 4.4588s$ , and delay,  $\theta = 1.4254$ . A comparison of the actual fourth order system, and the approximated system equation is presented in Figure 3.3(b).

The goal is to drive the system toward the desired closed-loop time constant. A desired time constant is chosen as a reference to the self-tuning system. The difference in the desired - and estimated closed-loop time constant  $\tau_c$ , is used to update the controller gain, as described in Equation (3.6). A combination of Equation (3.6) and (3.7) is referred to as the self-tuning law for this system with respect to gain.

$$K_{c_{new}} = \kappa K_{c_{old}} \quad (3.6)$$

Were  $\kappa$  is a case criterion and is defined in Equation (3.7).  $K_{c_{old}}$  is the optimal PIDF controller gain as derived by Jahanshahi et al.<sup>23</sup>. The process is iterative, thus the next tuning procedure the  $K_{c_{old}}$  represents the previous gain.

$$\kappa = \begin{cases} 1, & \text{if } |\tau_c - \tau_{des}| \leq 1. \\ \gamma(\tau_c - \tau_{des}), & \text{otherwise.} \end{cases} \quad (3.7)$$

Where  $\gamma$  is a tuning parameter. The case criterion is defined to reduce the aggressiveness of the self-tuning system.

A simple approach to update the integral part was inspired by Jahanshahi<sup>19</sup>. A new integral time or integral gain are estimated as described in Equation (3.8).

$$\tau_I = \tau_{I_{nom}} \left( \frac{\bar{Z}}{Z_{nom}} \right), \quad K_I = K_{I_{nom}} \left( \frac{\bar{Z}}{Z_{nom}} \right) \quad (3.8)$$

where  $\tau_I$  is the integral time,  $K_i$  is the integral gain.  $\tau_{I_{nom}}$  and  $K_{I_{nom}}$  is the integral time and integral gain at nominal operating point (Table 1.1), respectively.  $\bar{Z}$  is the mean valve opening and  $Z_{nom}$  is the valve opening at nominal operating point.

This proposed method uses a factor, based on the mean valve opening,  $\bar{Z}$  and valve opening at nominal operating point,  $Z_{nom}$ , to update the controller. The integral time or - gain is recalculated as the mean valve opening is changing. A recalculation of the integral part in either forms impacts the controller characteristics differently. The transformation from Ideal - (1.2a) to parallel PID form (1.2b) is described in Equation (1.3).

The measurement output from the slugging system consists of high frequent noise. The noise poses complications with respect to calculations of the new closed-loop time constant without proper filtering. An example of the high frequent noise is presented in Figure 3.4. In this figure a certain area is further zoomed in to magnify the high frequent noise. Unstable and irrelevant regions in this example, at start and end, is excluded with purpose.

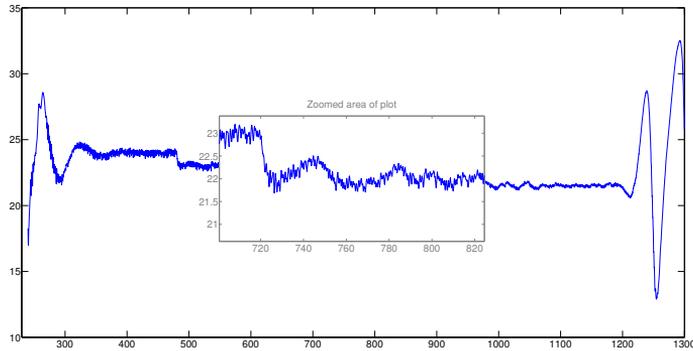


Figure 3.4: Illustrative example of high frequent noise in measurement output. Unstable region ( $200 < t < 1300$ ) is excluded with purpose. Meta figure is the given zoomed callout.

### Digital signal filtering

A Butterworth filter is commonly used in digital signal processing. The filter is designed to maximize the flatness of the pass band in the frequency response.<sup>24</sup> A normalized representation of an analog Butterworth filter is presented in Equation (3.9).

$$H(j\omega)H(-j\omega) = |H(j\omega)| = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}} \quad (3.9)$$

where  $n$  is the number of filter poles,  $\omega$  is the angular frequency [ $rad/s$ ],  $\omega = 2\pi f$ , where  $f$  is the temporal frequency [ $Hz$ ].

A second order,  $n = 2$ , Butterworth filter with a cut off frequency,  $\omega_c = 1^{rad/s}$  becomes:

$$|H(j\omega)| = \frac{1}{\sqrt{1 + \omega^4}} = \frac{1}{\sqrt{(1 - \omega^2)^2 + (\sqrt{2}\omega)^2}} \quad (3.10)$$

The complex conjugate of Equation (3.10) is presented in Equation (3.11).

$$H(j\omega) = \frac{1}{(1 - \omega^2) + (\sqrt{2}\omega)j} = \frac{1}{-\omega^2 + \sqrt{2}\omega j + 1} \quad (3.11)$$

A transformation of the filter represented in the frequency domain into the s-domain is performed by replacing  $\omega$  with  $s/j$ . A second order Butterworth filter represented in the s-domain is presented in Equation (3.12).

$$H(s) = \frac{1}{s^2 + \sqrt{2}s + 1} \quad (3.12)$$

In Equation (3.10), the cut off frequency of the filter is presented with a normalized cut off frequency at  $1^{rad/s}$ . The filter can be designed for any cut-off frequencies with a substitution of the Laplace operator,  $s$ , replaced with  $s = \frac{s}{\omega_c}$ .<sup>25</sup> The general form of the Butterworth filter in s-domain for a given cut off frequency is presented in Equation (3.13).

$$H(s) = \frac{\omega_c^2}{s^2 + s\sqrt{2}\omega_c + \omega_c^2} \quad (3.13)$$

The transformation of the filter from continuously s-domain to the discrete z-domain requires *bilinear transformation*.<sup>26</sup> This transformation involves digitalizing an analogue filter which maps the s-plane poles and zeros into the z-plane, and are designed to preserve the frequency characteristics. A mathematical description of bilinear transformation is presented in Equation (3.14). During mapping the entire  $j\omega$  axis is mapped onto the unit circle in the z-plane. The left-half s-plane is mapped inside the unit circle, and the right-half s-plane is mapped outside the

unit circle. This implies that the stable poles from s-planes are located inside the unit circle in the z-domain.

$$s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (3.14)$$

where  $T$  is the sampling period

A combination of Equation (3.12) and (3.14) gives a second order digital Butterworth filter.

### 3.3.2 Implementation

As the new controller parameters,  $K_c$ ,  $K_i$  and  $\tau_i$ , are obtained, they were used, both separately and in combination to update the optimal PIDF controller in both standard, and parallel form to observe their different influence on the system. A schematic illustration of the self-tuning system is presented in Figure 3.5.

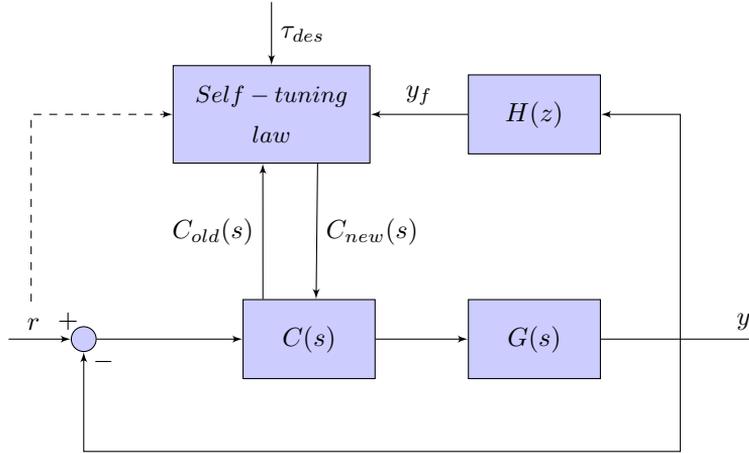


Figure 3.5: Block scheme of the self-tuning system. The controller is presented as  $C(s)$ , system dynamics as  $G(s)$  and a discrete low-pass filter is presented as  $H(z)$ . Desired closed-loop time constant is annotated as  $\tau_{des}$  and filtered signal is annotated as  $y_f$ .

MATLAB was used to calculate the proper poles and zeros for a second order Butterworth filter with a normalized cut off frequency of 0.02. The poles and zeros are describing in the filter as presented in Equation (3.15). This representation of the filter is also known as a finite impulse response filter in discrete form.<sup>27</sup>

$$H(z) = (9.447 \times 10^{-4}) \frac{(z + 1)^2}{z^2 - 1.911z + 0.915} \quad (3.15)$$

A pole-zero plot with the corresponding poles and zeros describing the filter is presented in Figure 3.6.

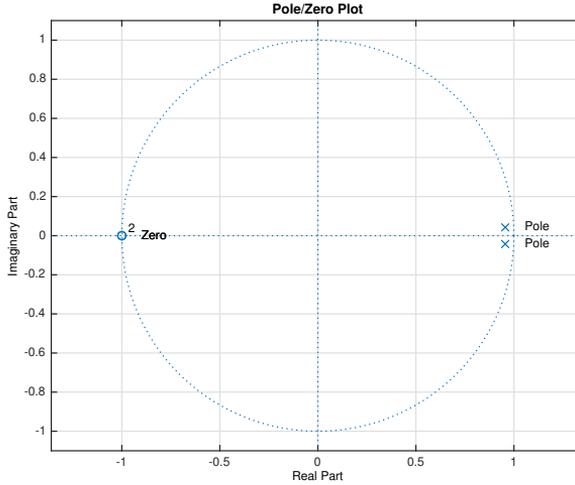


Figure 3.6: Location of poles and zeros in the applied discrete Butterworth filter. Zeros =  $[-1, -1]$ , and Poles =  $[0.9556 + 0.0425i, 0.9556 - 0.0425i]$ .

## Experimental

At nominal operating point, the closed-loop time constant,  $\tau_c$  was estimated to 3 seconds. This was used as the reference time constant,  $\tau_{des}$ , through further calculations. The valve opening at nominal operating point with optimal PIDF controller (Table 1.1), was in previous studies found to be  $Z_{nom} = 30\%$ <sup>7</sup>. The tuning parameter was defined as  $\gamma = 1$ , during experimental runs.

## 3.4 Scalar reference model adaptive system

### 3.4.1 Theoretical design

The intention of this model is to design an adaptive controller which includes a feedback - and a feed forward term based a first order reference model, in addition to the baseline controller. The concept and derivation of the adaptive laws in this section are adapted from Lavretsky and Wise<sup>15</sup>.

Consider a Linear Time-invariant (LTI) system described by

$$\dot{x} = Ax + B(u - \Theta^T \Phi_u(x)) \quad (3.16a)$$

$$y = Cx \quad (3.16b)$$

where  $\Theta$  is an unknown constant, and  $\Phi_u$  is a regressor related to the baseline controller  $u_{bl}$ . The regressor vector has to satisfy the Lipschitz continuous condition.<sup>15</sup>

An assumption of  $C = I_{n_x \times n_x}$  implies that the system state is the measurable output  $y = x$ .

and, similarly, a reference system described by

$$\dot{x}_{ref} = A_{ref}x_{ref} + B_{ref}r_{cmd} \quad (3.17a)$$

$$y_{ref} = C_{ref}x_{ref} \quad (3.17b)$$

Combining Equation (3.16a) and (3.17a), a control solution can be formulated as:

$$u = \left( \frac{A_{ref} - A}{B} \right) x + \left( \frac{B_{ref}}{B} \right) r_{cmd} + \Theta^T \Phi_u(x) \quad (3.18)$$

Were a feedback term, feed forward term and baseline controller term can be described as presented in Equation (3.19), respectively.

$$K_x = \left( \frac{A_{ref} - A}{B} \right), \quad K_r = \left( \frac{B_{ref}}{B} \right), \quad K_u = \Theta^T \quad (3.19)$$

Since the system parameters is unknown, the ideal controller gains,  $K_x$ ,  $K_r$  and  $K_u$ , has to estimated instead of computed directly. Which implies that Equation (3.18) boils down to:

$$u = \hat{K}_x x + \hat{K}_r r_{cmd} + \hat{K}_u \Phi_u(x) \quad (3.20)$$

were  $K_x, K_r$  and  $K_u$  in this case, represents the estimated feedback gain, feed forward gain and baseline controller gain, respectively.

With Equation (3.18) describing the controller output, the reference model in Equation (3.17a), can be rewritten with unknowns as:

$$\dot{x}_{ref} = \underbrace{(A + B \cdot K_x)}_{A_{ref}} x_{ref} + \underbrace{(B \cdot K_r)}_{B_{ref}} r_{cmd} \quad (3.21)$$

with gain estimation error described as:

$$\Delta K_x = \hat{K}_x - K_x \quad (3.22a)$$

$$\Delta K_r = \hat{K}_r - K_r \quad (3.22b)$$

$$\Delta K_u = \hat{K}_u - K_u \quad (3.22c)$$

With Equation (3.22) and Equation (3.21), the closed-loop system can be rewritten into the following form:

$$\begin{aligned} \dot{x}_{ref} = & \underbrace{(A + B \cdot K_x)}_{A_{ref}} x + \underbrace{(B \cdot K_r)}_{B_{ref}} r \\ & + B(\Delta K_x x + \Delta K_r r_{cmd} + \Delta K_u \Phi_u(x)) \end{aligned} \quad (3.23)$$

Error dynamics can be described by a subtraction of Equation (3.21) from Equation (3.23)

$$\dot{e} = A_{ref} e + B(\Delta K_x x + \Delta K_r r_{cmd} + \Delta K_u \Phi_u(x)) \quad (3.24)$$

In Equation (3.24), there are four error signals in which (1) is the process error  $e$ , (2) is the feedback gain estimation error  $K_x$ , (3) is the feed forward gain estimation error,  $K_r$  and (4) is the baseline controller estimation error,  $K_u$ . The adaptive goal is to use the gains ( $\hat{K}_x, \hat{K}_r, \hat{K}_u$ ) as tuning parameters, such that all these four errors converge to zero, both globally and asymptotically. To solve the problem, a Lyapunov candidate function (also known as energy function), must be defined. This function represents the total "kinetic energy" of all accumulated error in the system.

$$V(e, \Delta K_x, \Delta K_r, \Delta K_u) = \frac{e^2}{2} + \frac{|B|}{2\gamma_x} \Delta K_x^2 + \frac{|B|}{2\gamma_r} \Delta K_r^2 + \frac{|B|}{2\gamma_u} \Delta K_u^2 \quad (3.25)$$

The Lyapunov candidate function described in Equation (3.25) represents a weighted sum of squares of all the errors in the system. The positive scalar weights ( $\gamma_p, \gamma_r, \gamma_u$ ) will eventually become adaptation rate constants. A differentiated Lyapunov function with respect to time can be calculated,

$$\begin{aligned} \dot{V}(e, \Delta K_x, \Delta K_r, \Delta K_u) = & e\dot{e} + \frac{|B|}{\gamma_x} \Delta K_x \dot{K}_x + \frac{|B|}{\gamma_r} \Delta K_r \dot{K}_r \\ & + \frac{|B|}{\gamma_u} \Delta K_u \dot{K}_u \end{aligned} \quad (3.26)$$

The differentiated function described in Equation (3.26), is the system "power" with respect to changes. A substitution of Equation (3.24) into (3.26), describes the trajectories of the error dynamics with respect to time.

$$\begin{aligned} \dot{V}(e, \Delta K_x, \Delta K_r) = & A_{ref} e^2 + e \cdot B(\Delta K_x x + \Delta K_r r_{cmd}) \\ & + \frac{|B|}{\gamma_p} \Delta K_x \dot{K}_x + \frac{|B|}{\gamma_r} \Delta K_r \dot{K}_r + \frac{|B|}{\gamma_u} \Delta K_u \dot{K}_u \end{aligned} \quad (3.27)$$

A rearrangement of Equation (3.27) yields:

$$\begin{aligned} \dot{V}(e, \Delta K_x, \Delta K_r) = & A_{ref} e^2 + \Delta K_x |B| \left( \text{sgn}(B) \cdot x \cdot e + \frac{\dot{\hat{K}}_x}{\gamma_x} \right) \\ & + \Delta K_r |B| \left( \text{sgn}(b) \cdot r_{cmd} \cdot e + \frac{\dot{\hat{K}}_r}{\gamma_r} \right) \\ & + \Delta K_u |B| \left( \text{sgn}(B) \cdot \Phi_u(x) \cdot e + \frac{\dot{\hat{K}}_u}{\gamma_u} \right) \end{aligned} \quad (3.28)$$

As the goal is to achieve Lyapunov stability, it is desirable that the energy function described in Equation (3.28) dissipates in time. In other words,  $\dot{V} \leq 0$ . It requires that  $\dot{V}$  is non-positive, when evaluated along the system trajectories. This can be achieved if the adaptive laws are selected as following:

$$\dot{\hat{K}}_x = -\gamma_x \cdot x \cdot e \cdot \text{sgn}(B) \quad (3.29a)$$

$$\dot{\hat{K}}_r = -\gamma_r \cdot r_{cmd} \cdot e \cdot \text{sgn}(B) \quad (3.29b)$$

$$\dot{\hat{K}}_u = -\gamma_u \cdot \Phi_u(x) \cdot e \cdot \text{sgn}(B) \quad (3.29c)$$

As a result of the Equation (3.29), the second and the third term in Equation (3.28) disappears. Which results in

$$\dot{V}(e, \Delta K_x, \Delta K_r) = A_{ref} e^2 \leq 0 \quad (3.30)$$

A consequence of Equation (3.30) the system "kinetic energy",  $V$ , is decreasing function of time, thus Lyapunov stable.

By assumption of satisfaction of Lipschitz continuous condition with respect to the baseline controller,  $\Phi_u(x) = u_{bl}$ , the final adaptive laws is rewritten as presented in Equation (3.31). The baseline controller is the optimal PIDF controller designed by Jahanshahi and Skogestad<sup>7</sup> with its respective parameters presented in Table 1.1.

$$\frac{\partial \hat{K}_x}{\partial t} = \dot{\hat{K}}_x = -\gamma_x x (y - y_{ref}) \cdot \text{sgn}(B) \quad (3.31a)$$

$$\frac{\partial \hat{K}_r}{\partial t} = \dot{\hat{K}}_r = -\gamma_r r (y - y_{ref}) \cdot \text{sgn}(B) \quad (3.31b)$$

$$\frac{\partial \hat{K}_u}{\partial t} = \dot{\hat{K}}_u = -\gamma_u u_{bl}(y - y_{ref}) \cdot \text{sgn}(B) \quad (3.31c)$$

were  $\gamma_x, \gamma_r$  and  $\gamma_u$  are tuning parameters in form of rate of adaption. The integration of  $K_x, K_r$  and  $K_u$  are put on hold, if the controlled variable, the valve opening, is saturated.

With the integrated adaptive laws from Equation (3.31), the adaptive controller output is described in Equation (3.32).

$$u_{ad} = \gamma_s \left( \hat{K}_x(y - y_0) + \hat{K}_r(r - r_0) + \hat{K}_u(u - u_0) \right) \quad (3.32)$$

were  $y_0, r_0$  and  $u_0$  are the pressure, set point and controller output at nominal operating point, respectively. A scaling factor,  $\gamma_s$ , is introduced to prevent saturation of the system. This was considered as an additional tuning parameter. A linear valve equation was implemented in the design of the model developed by Jahanshahi and Skogestad<sup>6</sup>. With the controller linearized around  $Z = 30\%$ , it is obvious that  $u_0 = 30$ . This leads to  $(u - u_0) = u_{bl}$ , where  $u_{bl}$  represents the baseline controller output. The final controller output is presented in Equation (3.33).

$$u = u_{bl} + u_{ad} \quad (3.33)$$

were  $u_{bl}$  is defined in Equation (1.2) with optimal PIDF controller from Table 1.1.

### 3.4.2 Implementation

A reference model was implemented as a open-loop first order model as presented in Equation (3.34).

$$G_{ref}(s) = \frac{1}{\tau_{des}s + 1} \quad (3.34)$$

At nominal operating point, the desired time constant,  $\tau_c$  was estimated to 3 seconds. This was used as the reference time constant,  $\tau_{des}$ , through further calculations. The tuning parameters  $\gamma_x, \gamma_r, \gamma_u$  and  $\gamma_s$  was determined using a trial and error approach. The result of the adaptive system was compared against the optimal controller, without the adaptive part activated.

The adaptive system based on scalar reference model was implemented as presented in Figure 3.7.

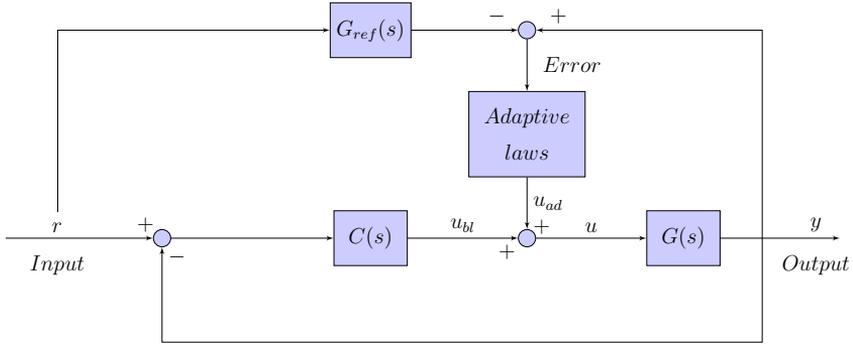


Figure 3.7: Block scheme presentation of the scalar reference adaptive system. In this figure the  $u_{bl}$  represents the optimal PIDF controller,  $C(s)$ , output, and  $u_{ad}$  is the adaptive controller output as described in Equation (3.32). The slug process is labeled,  $G(s)$  and reference model is labeled  $G_{ref}(s)$ .

### Simulation

The tuning parameters,  $\gamma_x, \gamma_r$  and  $\gamma_u$ , was in simulation observed to have large impact with respect to the stability. Minor changes led to large oscillations. The most optimal tuning parameters were found to be as presented in Equation (3.35).

$$\begin{aligned} \gamma_x &= 1, & \gamma_r &= 1 \\ \gamma_u &= 1, & \gamma_s &= 1 \end{aligned} \quad (3.35)$$

### Experimental

As the experimental measurement data was highly influenced by randomized noise, the tuning parameter  $\gamma_s$  was observed to be critical. The optimal tuning parameters were found to be as presented in Equation (3.36).

$$\begin{aligned} \gamma_x &= 0.5, & \gamma_r &= 1.7 \\ \gamma_u &= 0.07, & \gamma_s &= 0.07 \end{aligned} \quad (3.36)$$

The adaptive system was compared against experiments with a fixed controller. That is with the adaptive part disabled.

## 3.5 Augmented state observer-like reference model

### 3.5.1 Theoretical design

As the pressure is not actually explicitly describing neither of the states, the states will be estimated using an observing design. The measured output is extended to

include the integrated error between the physical pressure and observing pressure, in addition to the physical pressure. The state space is hence manipulated to include the integrated error, with respect to set point, as an extra input variable.

Consider a LTI system described by

$$\dot{x}_p = A_p x_p + B_p u \quad (3.37a)$$

$$y = C_p x_p + D_p u \quad (3.37b)$$

were  $D_p$  is zero due to absence of feed forward in the system.

with error dynamics described by Equation (3.38).

$$e = r - y = r - C_p x_p \quad (3.38)$$

and feedback control described by Equation (3.39).

$$u = K \cdot x \quad (3.39)$$

were  $u$  can be differentiated with respect to time.

$$\dot{u} = K \cdot \dot{x} = K(A_p x_p + B_p u) \quad (3.40)$$

By augmenting the system dynamics to include output tracking error, the extended system can be described as presented in Equation (3.41).

$$\underbrace{\begin{bmatrix} \dot{e}_I \\ \dot{x}_p \end{bmatrix}}_{\tilde{x}} = \underbrace{\begin{bmatrix} 0_{m \times m} & -C_p \\ 0_{n_p \times m} & A_p \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} e_I \\ x_p \end{bmatrix}}_{\tilde{x}} + \underbrace{\begin{bmatrix} 0_{m \times m} \\ B_p \end{bmatrix}}_{\tilde{B}} u + \underbrace{\begin{bmatrix} I_{m \times m} \\ 0_{n_p \times m} \end{bmatrix}}_{B_{cmd}} r \quad (3.41)$$

were  $e_I$  is the integrated error dynamics and consequently  $\dot{e}_I = e$ . The corresponding output vector as presented in Equation (3.42).

$$\tilde{y} = \underbrace{\begin{bmatrix} 1 & 0_{m \times n_p} \\ 0_{m \times m} & C_p \end{bmatrix}}_{\tilde{C}} \underbrace{\begin{bmatrix} e_I \\ x_p \end{bmatrix}}_{\tilde{x}} \quad (3.42)$$

The four-state slug model is consistent of a system state vector of  $x_p \in \mathbb{R}^4$ , and control input vector  $u \in \mathbb{R}^1$  it follows that  $y = C_p x_p \in \mathbb{R}^1$ . The augmented system size as presented in Equation (3.43).

$$\tilde{A} \in \mathbb{R}^{n \times n}, \quad \tilde{B} \in \mathbb{R}^{n \times m}, \quad \tilde{B}_{cmd} \in \mathbb{R}^{n \times m}, \quad \tilde{C} \in \mathbb{R}^{(m+1) \times n} \quad (3.43)$$

Were the number of states,  $n_p = 4$ , number of inputs,  $m = 1$  and the extended system dimension as,  $n = n_p + m = 5$ .

### Linear Quadratic Regulator

The general cost function of LQR design is presented in Equation (3.44).<sup>11</sup>

$$J_{LQR} = \int_0^{\infty} x^T Q x + u^T R u dt \quad (3.44)$$

were the penalty factor  $Q \in \mathbb{R}^{n \times n}$  and  $R \in \mathbb{R}^{m \times m}$  are symmetric positive-definite matrices, as defined in Equation (3.45)

$$Q = \begin{bmatrix} q & 0_{m \times n_p} \\ 0_{n_p \times n_p} & 0_{n_p \times m} \end{bmatrix}, \quad R = \begin{bmatrix} I_{m \times m} \end{bmatrix} \quad (3.45)$$

The goal is to design the gain matrix to follow set point commands without use of large gains. With  $R = 1$ , the selected  $Q$  is penalizing the error state,  $e$ , in Equation (3.41). The element position  $q$  in  $Q$  is in other words the penalty scalar factor with respect to the error.

A minimization of,  $J_{LQR}$ , Equation (1.4) can be rewritten as:<sup>28</sup>

$$\tilde{A}^T X + X \tilde{A} - X \tilde{B} R^{-1} \tilde{B}^T X + Q = 0 \quad (3.46)$$

were  $X$  is an unknown symmetric matrix

The minimized cost function in form of Equation (1.4) rewritten into Equation (3.46) is also known as the Algebraic Riccati Equation (ARE).

From this equation, the optimal LQR control policy is described in Equation (3.47), with the Loop Transfer Function Matrix (LTFM) described in Equation (3.48)

$$u_{LQR} = - \underbrace{R^{-1} \tilde{B}^T X}_{K_{LQR}} \tilde{x} = -K_{LQR} \tilde{x} \quad (3.47)$$

$$L_{LQR}(s) = K_{LQR}(sI - \tilde{A})^{-1} \tilde{B} \quad (3.48)$$

The reference model is defined in Equation (3.49).

$$\begin{aligned} A_{ref} &= \tilde{A} - \tilde{B} K_{LQR}, & B_{ref} &= \begin{bmatrix} 0_{m \times n_p} & 1 \end{bmatrix}^T, \\ C_{ref} &= \tilde{C}, & D_{ref} &= 0 \end{aligned} \quad (3.49)$$

### Linear Quadratic Gaussian with Loop Transfer Recovery

As the physical pressure was available for measurement during state estimation, a Loop-Transfer-Recovery was implemented to recover the desired performance as predicted by LQR.

Consider the following LTI augmented system model as:

$$\dot{x} = \tilde{A}x + \tilde{B}u + w \quad (3.50a)$$

$$y = \tilde{C}x + v \quad (3.50b)$$

were  $w$  and  $v$  are zero by mean, uncorrelated random Gaussian noise.

The estimated state is designed by using Kalman filter state estimator:

$$\dot{\hat{x}} = \tilde{A}\hat{x} + \tilde{B}u + L_v(y - \hat{y}) \quad (3.51)$$

were  $\hat{y}$  is the output estimate, and  $L_v \in \mathbb{R}^{n \times m}$  is the Kalman gain, and is defined by

$$L_v = P_f \tilde{C}^T R_0^{-1} \quad (3.52)$$

With the estimated state described in Equation (3.51) and the Kalman gain described in Equation (3.52), the steady-state error covariance,  $P_f = E\{xx^T\}$ , can be solved by using the ARE in form of Equation (3.53).

$$\tilde{A}^T P_f + P_f \tilde{A} - P_f \tilde{C} R_0^{-1} \tilde{C}^T P_f + Q_0 = 0 \quad (3.53)$$

were  $Q_0$  and  $R_0 \in \mathbb{R}$  are the process - and measurement noise covariance matrix from Equation (3.52), respectively. Using the estimated state  $\hat{x}$  and the solution of the LQR minimization,  $K_{LQR}$ , the optimal controller with observer gain,  $L_v$ , is formed. The estimation of the optimal control output is described in Equation (3.54),

$$u_{LQG/LTR} = -K_{LQR}\hat{x} \quad (3.54)$$

with the LTFM described in Equation (3.55).

$$L_{LQG}(s) = K_{LQR}(sI - \tilde{A} + \tilde{B}K_c + L_v\tilde{C})^{-1}L_v\tilde{C}(sI - \tilde{A})^{-1}\tilde{B} \quad (3.55)$$

A block scheme describing the combination of LQR controller and a Kalman filter state estimator is presented in Figure 3.8.

By observing the Loop Transfer Function Matrix in Equation (3.55), it is obvious that the dynamic compensator,  $L_v$ , alters the frequency domain of the LQG system. The recovery design is performed using the Lavretsky method.<sup>15</sup>

New covariance matrices was defined to tune the process - and measurement noise, using a positive scalar  $v$ .

$$Q_v = Q_0 + \left(\frac{v+1}{v}\right)\tilde{B}\tilde{B}^T \quad (3.56)$$

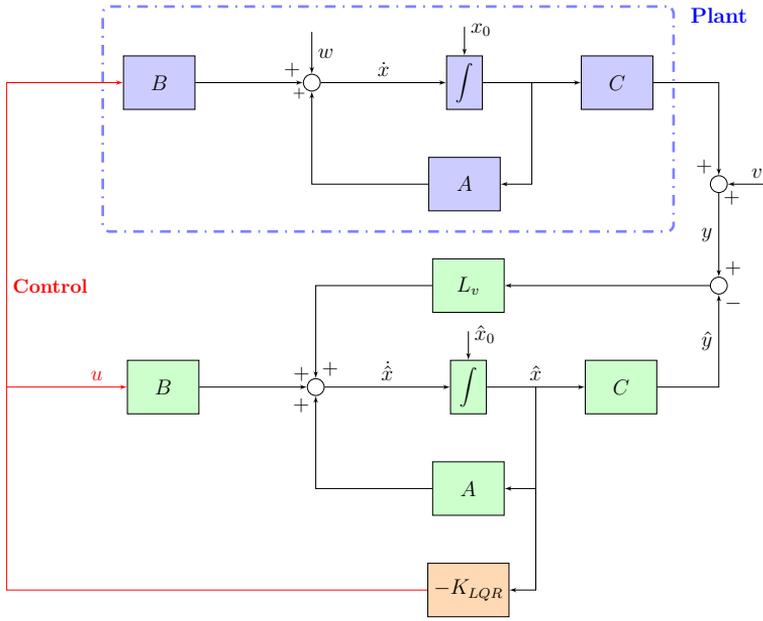


Figure 3.8: Block scheme in state space representation of LQG adaptive system with Kalman filter state estimator as integral control estimated feedback as described in Equation (3.51).  $w$  and  $v$  are zero by mean, uncorrelated random Gaussian noise.

$$R_v = \left( \frac{v}{v+1} \right) R_0 \quad (3.57)$$

Were  $v > 0$  is a common scalar tuning parameter chosen during design, and  $\bar{B} = \begin{bmatrix} \tilde{B} & X \end{bmatrix}$ . The extended "squared-up" matrix,  $\bar{B}$ , is formed to have its column rank equal to the row rank of  $\tilde{C}$ , such that  $\tilde{C}\bar{B}$  becomes invertible. That is  $\bar{B} \in \mathbb{R}^{n \times (m+1)}$ . The corresponding extended system  $\tilde{C}(sI - \tilde{A})^{-1}\bar{B}$  has a minimum phase, in other words no transmission zeros in the Right-half plane (RHP). This known as the "squaring-up" part of the Lavretsky method. The added column is fictitious in form of pseudo-inputs of the system and do not represent any physical inputs of the system.<sup>29</sup> A substitution of Equation (3.56) and (3.57) into the generic ARE (Equation 3.58) the extended ARE emerges and is presented in Equation (3.59).

$$\tilde{A}^T P_f + P_f \tilde{A} - P_f \tilde{C} R_v^{-1} \tilde{C}^T P_f + Q_v = 0 \quad (3.58)$$

$$\tilde{A}^T P_v + P_v \tilde{A} - \left(1 + \frac{1}{v}\right) P_v \tilde{C}^T R_0^{-1} \tilde{C} P_v + Q_0 + \left(1 + \frac{1}{v}\right) \bar{B} \bar{B}^T = 0 \quad (3.59)$$

The observer gain is calculated as presented in Equation (3.60).

$$L_v = P_v \tilde{C}^T R_v^{-1} \quad (3.60)$$

The observer system model designed from reference model (Equation 3.49) and optimal observer gain is defined in Equation (3.61).

$$\begin{aligned} A_{ob} &= A_{ref} - L_v \tilde{C}, & B_{ob} &= \begin{bmatrix} B_{ref} & L_v \end{bmatrix}, \\ C_{ob} &= \tilde{C}, & D_{ob} &= 0 \end{aligned} \quad (3.61)$$

Singular value decomposition is used to describe the "gain" and direction through the system, and is calculated as described in Equation (3.62).

$$\bar{B}^T \tilde{C}^T R_0^{-1/2} = U \Sigma V \quad (3.62)$$

The factorized vectors,  $U$  and  $V$  are used to compose a orthonormal matrix,  $W$ , as described in Equation (3.63)

$$W = (U \cdot V)^T \quad (3.63)$$

In this equation the singular value decomposition in Equation (3.62) gives a orthonormal matrix  $W \in \mathbb{R}^{m \times m}$ , in such that the error dynamics almost becomes Strictly Positive Real (SPR), as long as the tuning parameter,  $v$ , is sufficiently small.<sup>15</sup>

A scaling factor based on  $B$  and controller dynamics is presented in Equation (3.64). This factor was used to scale the observer error.

$$F = R_0^{-1/2} W S \quad (3.64)$$

We assume the numbers of inputs are less than number of outputs.

Where  $S$  is chosen as:<sup>15</sup>

$$S = \begin{bmatrix} I_{m \times m} & 0_{(p-m) \times m} \end{bmatrix} \quad (3.65)$$

where  $p = 2$  represents the number of outputs as pressure and integrated error.

The adaptive laws are chosen as presented in Equation (3.66).

$$\dot{K}_p = \gamma_p \Phi_x(\hat{x}) (y - y_{ref}) \cdot F \quad (3.66a)$$

$$\dot{K}_u = \gamma_u \Phi_u(u_{bl}) (y - y_{ref}) \cdot F \quad (3.66b)$$

where  $\gamma_p$  and  $\gamma_u$  are tuning parameters in form of rate of adaption. The regressor vectors,  $\Phi_x$  and  $\Phi_u$ , has to satisfy the Lipschitz continuous condition.<sup>15</sup>

To ensure that the adaptive laws do not exceed the limitation of the controllers, certain boundaries have to be defined.

$$\dot{K}_{p_{proj}} = Proj(\dot{K}_p, \Theta_{max}) \quad (3.67a)$$

$$\dot{K}_{u_{proj}} = Proj(\dot{K}_u, \Theta_{max}) \quad (3.67b)$$

The Projection operator presented in Equation (3.67), is designed to make sure  $\dot{K}_p$  and  $\dot{K}_u$  stays within a certain boundary  $\Theta_{max}$ . This imposes that the adaptive laws stays smoothly uniformly bounded in time. This operator is further described by Pomet and Praly<sup>30</sup>, and accompanied by MATLAB code presented in Appendix B.4.

$$K_p = max(-\Theta_{max}, min(\Theta_{max}, K_p)) \quad (3.68a)$$

$$K_u = max(-\Theta_{max}, min(\Theta_{max}, K_u)) \quad (3.68b)$$

If the Projection operator fails, the adaptive laws,  $\dot{K}_p$  and  $\dot{K}_u$ , could get out of its boundary,  $\Theta_{max}$ . To prevent a projection failure a limiting criterion is defined, and are presented in Equation (3.68). This is implemented in order to make sure the controller gains stays within the boundaries,  $-\Theta_{max} < K_{p/u} < \Theta_{max}$  in a more static matter after integration.

$$u_{ad} = -K_p^T \Phi_x(\hat{x}) - K_u^T \Phi_u(u_{bl}) \quad (3.69)$$

The final controller output is presented in Equation (3.70).

$$u = u_{bl} + u_{ad} \quad (3.70)$$

were  $u_{bl}$  is defined in Equation (3.54).

### Dead-Zone Modification

To enforce the robustness of adaptive laws in presence of process noise the adaptation should be stopped when the norm of the tracking error becomes smaller than some value,  $\epsilon$ .

$$\mu = \begin{cases} 1, & \text{if } \|e\| > \epsilon_{min}. \\ 0, & \text{if } \|e\| \leq \epsilon_{min}. \end{cases} \quad (3.71)$$

were  $\epsilon_{min}$  represents a signal-to-noise boundary, obtained by observation of the absolute scaled error between reference model and actual process,  $|(y - y_{ref})| \cdot |F|$ .

The disadvantage of using Equation (3.71) is the immediate transition between adaption laws being active or non active. A smoother modulation of the dead-zone boundary is presented in Equation (3.72).<sup>31</sup> An illustrative example of the dead-zone modulation is presented in Figure 3.9.

$$\mu(\|e\|) = \max\left(0, \min\left(1, \frac{\|e\| - \delta\epsilon_{min}}{(1 - \delta)\epsilon_{min}}\right)\right) \quad (3.72)$$

were  $\delta$  is the transition range factor from the  $\epsilon_{min}$  boundary.

The adaptive laws presented in Equation (3.66) are thus be modified to

$$\dot{K}_p = \gamma_p \mu \Phi_x(\hat{x}) (y - y_{ref}) \cdot F \quad (3.73a)$$

$$\dot{K}_u = \gamma_u \mu \Phi_u(u_{bl}) (y - y_{ref}) \cdot F \quad (3.73b)$$

The regressor vectors,  $\Phi_x$  and  $\Phi_u$ , are chosen as defined in Equation (3.74).

$$\Phi_x(\hat{x}) = C_{obs}\hat{x}, \quad \Phi_u(u_{bl}) = u_{bl} \quad (3.74)$$

The adaptive laws with dead-zone modification are further projected as previously described.

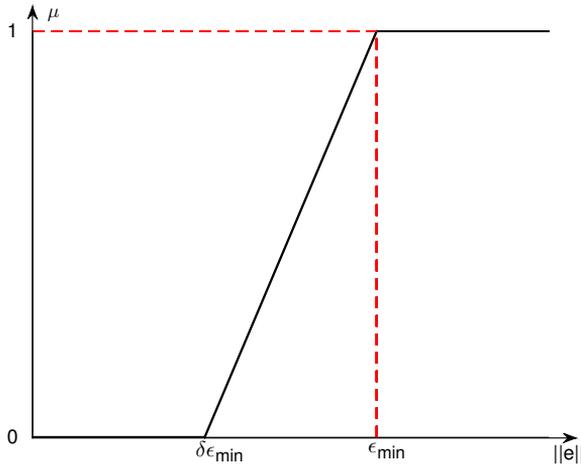


Figure 3.9: Illustrative example of the dead-zone modulation as described in Equation (3.72).

### 3.5.2 Implementation

A Luenberger-observer form was implemented as described in Equation (3.75).

$$\dot{\hat{x}} = A\hat{x} + Bu + L_v(y_m - \hat{y}) \quad (3.75)$$

where  $\hat{x}$  is the estimated state by observation,  $L_v$  is the correction term as a function of the estimation error, also known as the observer gain.

The augmented matrices linearized at nominal operating condition with  $Z = 30\%$  are presented in Equation (3.76)-(3.77).

$$\tilde{A} = 1.0 \times 10^3 \begin{bmatrix} 0 & -3.9200 & -0.0060 & 0 & 0 \\ 0 & -0.0001 & 0.0000 & 0.0036 & 0.0000 \\ 0 & -0.0451 & -0.0095 & 2.5640 & 0.0036 \\ 0 & 0.0001 & -0.0001 & -0.0086 & -0.0000 \\ 0 & 0.0451 & 0.0171 & -6.1178 & -0.0082 \end{bmatrix} \quad (3.76)$$

$$\tilde{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -3.01 \times 10^{-6} \\ -2.22 \times 10^{-3} \end{bmatrix}, \quad \tilde{C} = 1.0 \times 10^3 \begin{bmatrix} 0.001 & 0 \\ 0 & 3.920 \\ 0 & 0.006 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}^T \quad (3.77)$$

The observer-based system was implemented as presented in Figure 3.10.

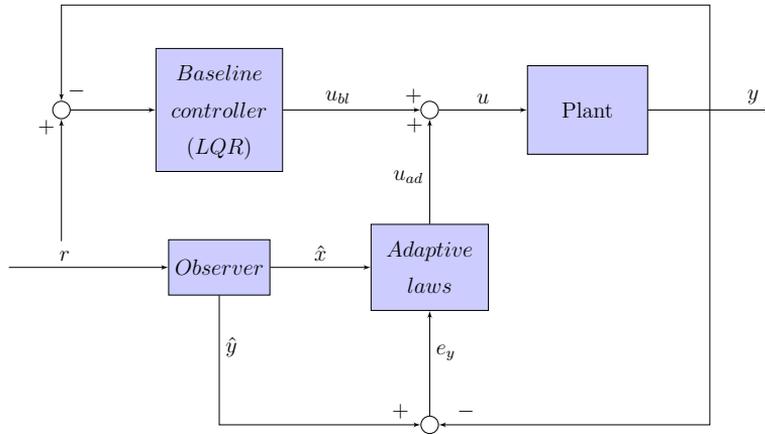


Figure 3.10: Simplified block scheme of observer-based adaptive system.

The optimized LQR controller was designed as described in Equation (3.78).

$$Q = \begin{bmatrix} q & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad R = 1 \quad (3.78)$$

where  $q$  is the penalty scalar parameter with respect to state error. The performance and robustness of the controller with respect to  $q$  was characterized by delay margin, settling time, controller usage and frequency roll-off for the system.

### LQG/LTR

In the design of the Kalman filter the measurement noise covariance,  $R_0$ , were chosen as

$$R_0 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad (3.79)$$

The right and left singular matrix,  $U$  and  $V$  were defined as

$$U = 1, \quad V = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.80)$$

Which resulted in the orthonormal matrix,

$$W = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.81)$$

With  $S$  chosen as

$$S = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (3.82)$$

The scaling factor  $F$ , is presented in Equation (3.83).

$$F = \begin{bmatrix} -0.7071 & 0 \end{bmatrix}^T \quad (3.83)$$

The squared up matrix were chosen using uniformly distributed random numbers with a variance of  $10^{-4}$  and a mean of  $5 \times 10^{-3}$ . This matrix were kept constant through all tuning optimization procedures, in order to keep each simulation comparable.

$$\bar{B} = \begin{bmatrix} 0.0110 & 0.0084 & 0.0080 & 0.0095 & 0.0092 \\ 0.0086 & 0.0106 & 0.0124 & 0.0092 & 0.0093 \end{bmatrix}^T \quad (3.84)$$

The performance and robustness the Loop-Transfer-Recovery with respect to  $v$  was characterized by Robustness, Performance and Stability (RPS) analysis.

The optimal Linear-Quadratic-Gaussian with Loop-Transfer-Recovery was designed with penalty factor  $q_{LQR} = 10$  and tuning scalar  $v_{LTR} = 100$ .

### Simulation

The apparent optimal tuning parameters representing the rate of adaption are presented in Equation (3.85). The simulated adaptive system was compared to a fixed LQG/LTR controller.

$$\begin{aligned}\gamma_p &= 1000, & \gamma_u &= 100 \\ \Theta_{max} &= 10\end{aligned}\tag{3.85}$$

### Experimental

As the experimental measurements are influenced by randomized noise, the dead-zone,  $\epsilon$ , was by observation of the scaled error ( $|F| \cdot |\hat{y} - y|$ ) found to be approximately  $\epsilon = 0.0016$ . The following tuning parameters representing the rate of adaption are presented in Equation (3.86). The experimental adaptive system was compared to a fixed LQG/LTR controller.

$$\begin{aligned}\gamma_p &= 0.05, & \gamma_u &= 0.005 \\ \Theta_{max} &= 10, & \epsilon_{min} &= 0.0016\end{aligned}\tag{3.86}$$

The adaptive system was compared against experiments with a fixed controller using LQG/LTR. That is with the adaptive part disabled.

### 3.6 Supervisory adaptation

A supervisor control is the administrative part of the adaptive control system. The supervisory system does all the decision-making while detecting any potential malfunctioning of the adaptive controller, and if any necessary action can be performed to maintain stability. The supervisory control can be used to quickly adjust the operation conditions, such as set point. If the controlled variables are saturated, the adaptive controller is inoperable, hence the system stability would be difficult to regain. A change of controller gain may not be applicable to the current operating condition, while oscillating.<sup>32</sup>

The supervisory system determines the stability in two separate design forms. One in which measures the Integrated Absolute Error (IAE) between set point and actual system state, and one in which uses the period of oscillation to determine whether the system is stable or not. Previous work conducted by De Oliveira et al.<sup>20</sup> uses a supervisor system as described, and will be, in this thesis, only used to determine the current stability condition.

#### (1) - INTEGRATED ABSOLUTE ERROR:

Integrated Absolute Error is a performance test of the system. The performance is calculated by integrating the absolute error between set point, and the physical state over time. If the IAE exceeds a certain boundary,  $\epsilon$ , the system is defined as unstable. The IAE is calculated as presented in Equation (3.87), with the boundary concept defined in Equation (3.88).

$$IAE = \int_0^t |r - y| dt \quad (3.87)$$

$$\text{unstable} = \begin{cases} 1, & \text{if } IAE \geq \epsilon. \\ 0, & \text{otherwise.} \end{cases} \quad (3.88)$$

Where  $\epsilon$  is some value obtained by trial and error.

#### (2) - SIGNAL FREQUENCY: PERIOD OF OSCILLATION:

Cross-correlation may be used to measure the frequency of the signal. This distance/displacement is defined as the period of oscillation.

The distance between the peaks are used in comparison with another segment window. If the frequency is decreasing the system are moving towards instability. Set point must be changed upwards in order to regain stability. If the frequency is increasing the system is controlled with a too aggressive controller gain. Set

point must be changed downwards to counteract. The adaptive system should be temporary turned off if the system is flagged as unstable, or if the control gain is too high. An illustration of the concept of cross-correlation is presented in Figure 3.11. In this figure the ordinary signal (blue) is shifted (red) and matched with an equal length signal from previous segment window (dashed line). The frequency is increasing, which implies that the system is moving towards unstable conditions.

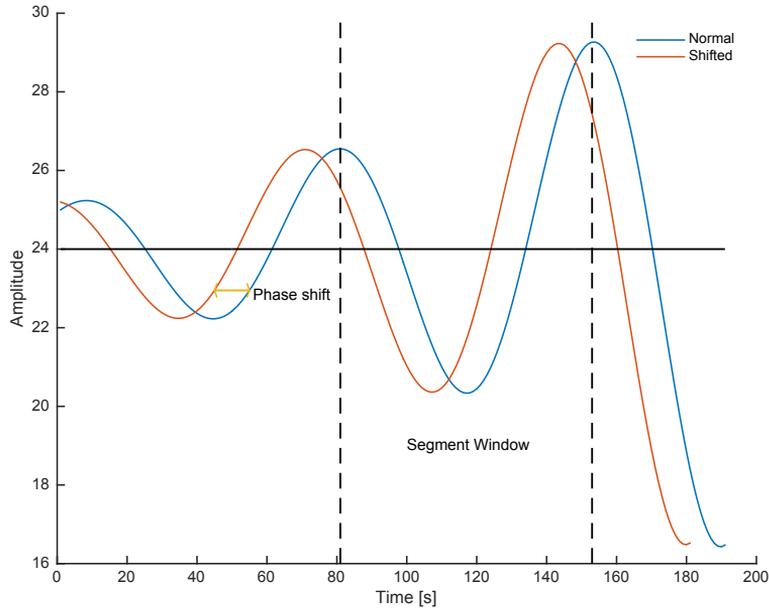


Figure 3.11: Cross-correlation for an arbitrary signal. The normal signal (blue) is shifted (red) and matched with an equal length signal from previous segment window (dashed line) to determine stability.

### 3.7 Evaluation of stability and robustness limitations

There are several ways to determine the RPS of a closed-loop system. The most common approaches to determine the stability is to calculate the Gain margin (GM), phase margin (PM), delay margin (DM). The stability margins measures how close a stable-closed loop system is to instability before actually closing the

loop. The margins are calculated as described in Equation (3.89).

$$GM_L = \frac{1}{|L(j\omega_{L180})|} \quad \text{Lower Gain Margin} \quad (3.89a)$$

$$PM = \angle L(j\omega_c) + 180^\circ \quad \text{Phase Margin} \quad (3.89b)$$

$$\theta_{max} = \frac{PM}{\omega_c} \quad \text{Delay Margin} \quad (3.89c)$$

Where  $\omega_{L180}$  is the smallest phase crossover frequency at  $180^\circ$  and  $\omega_c$  is the gain crossover frequency at 1 magnitude.

Lower Gain margin describes the gain factor the loop gain,  $|L(j\omega)|$  may be decreased before the closed-loop system becomes unstable. As there is more than one phase crossover frequency for unstable systems, the lower gain margin will be the critical gain margin. Phase margin describes the maximum phase lag the closed-loop system may be shifted before the system becomes unstable. Delay margin is the maximum allowed time delay before the closed-loop is at the borderline to instability.

Sensitivity and robustness, on the other hand, may be characterized by the sensitivity peak ( $M_t$ ) and complementary sensitivity peak ( $M_s$ ). The peak represents the maximum value of the frequency response or sensitivity norm,  $\|\bullet\|_\infty$ . A small  $M_s$  benefits feedback stability or set-point tracking, while a small  $M_t$  benefits sensitivity to process noise.<sup>21</sup> These benefits, however, are not possible simultaneously as for a SISO:  $S(s) + T(s) = 1$ . The maximum peaks of the sensitivity and complementary sensitivity are defined as presented in Equation (3.90).

$$M_s = \|S\|_\infty = \max_\omega |S| \quad , \quad M_t = \|T\|_\infty = \max_\omega |T| \quad (3.90)$$

Where  $S$  and  $T$  are described in Equation (3.91).

$$S = \frac{1}{1 + GC}, \quad T = \frac{GC}{1 + GC} \quad (3.91)$$

Where  $G$  is the process dynamics, and  $C$  is the controller.

The sensitivity peak,  $M_t$  are often smaller than the complimentary sensitivity peak,  $M_s$  for stable processes, and oppositely for unstable processes.<sup>21</sup> The slugging system is an unstable system, which implies that the expected sensitivity will be shifted towards  $M_t > M_s$ .

The system RPS analysis will be relative to a linearized operating point with valve opening,  $Z = 30\%$ .

## Chapter 4

---

# Results

---

All experiments were conducted with a run time of either 45 or 60 minutes, with set point intervals as presented in Table 4.1. Run time and set points in simulations and experiments are not equal, and can not be directly compared. The optimal PIDF controller derived by Jahanshahi et al.<sup>23</sup> was used as baseline controller for the self-tuning system and the adaptive system based on scalar reference model. The adaptive system using an observer-like model was implemented with Linear-Quadratic-Gaussian with Loop-Transfer-Recovery as the baseline controller. All experiments were conducted under equal operating conditions, except for the self-tuning system results.

*Table 4.1: Experimental set point changes numbered with its respective value at given time.*

Set point (N)	Set point [kPa]	Time [sec] (45 min run)	Time [sec] (60 min run)
1	24.0	0	0
2	23.0	675	900
3	22.0	1013	1350
4	21.5	1350	1800
5	21.0	1688	2250
7	20.5	2025	2700
8	20.0	2362	3150

## 4.1 Controllability and System dynamics with the four state model

The four state MATLAB model previously developed by Jahanshahi and Skogestad<sup>6</sup>. Only one pressure measurement,  $P_1$  (Figure 3.1) was available throughout the simulations. The model was also used to create a state space linearization with a valve opening of 30%. This linearized state space was used to design and evaluate the state controllability, stability and performance of PIDF and LQG/LTR.

The relationship between system gain and to valve opening,  $Z$ , is plotted in Figure 4.1. This figure illustrates the rapid change of the system gain with respect to valve opening. A pole-zero plot with respect to the valve opening is presented in Figure 4.2.

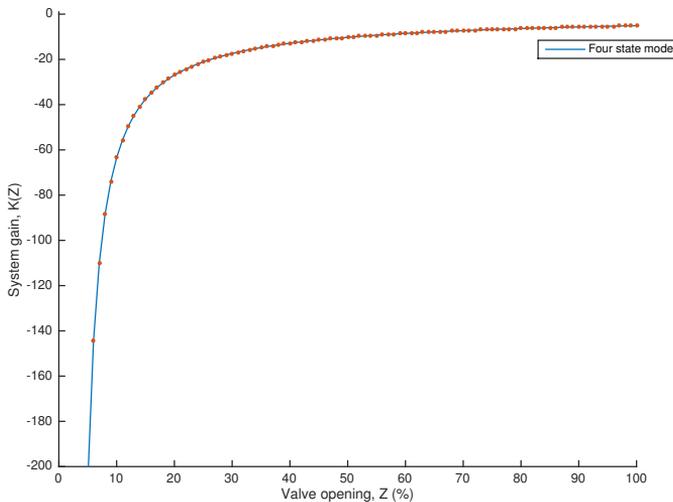


Figure 4.1: Plot of simulated system gain,  $K$ , with respect to valve opening,  $Z$ , and buffer pressure,  $P_1$  as noted in Figure 3.1.

The state controllability matrix is presented in Equation (4.1).

$$\mathcal{C} = \begin{bmatrix} 0 & 0 & -0.0003 & 0.0155 \\ 0 & -0.0156 & 0.4088 & -6.1177 \\ 0 & 0 & 0.0003 & -0.0241 \\ -0.0022 & 0.0366 & -0.8723 & 11.9907 \end{bmatrix} \quad (4.1)$$

The state controllability matrix,  $\mathcal{C}$ , has a full row rank of 4, thus equal to the number of states.

The open-loop dynamics with optimal PIDF controller was plotted using Bode. The result is presented in Figure 4.3. The Robustness, Performance and Stability

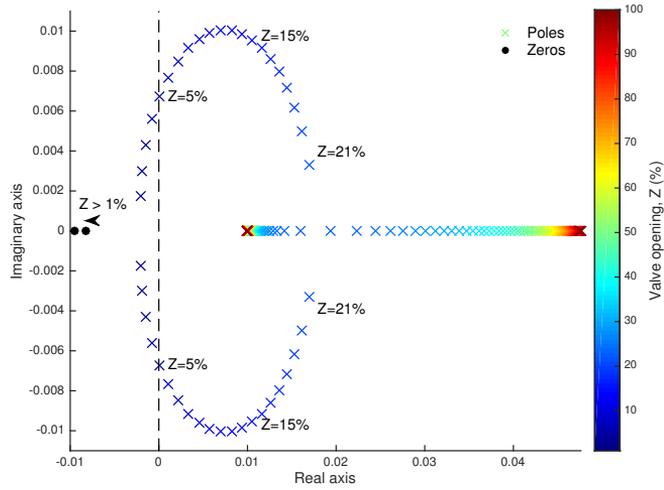


Figure 4.2: Location of RHP-Poles ( $\times$ ) and Zeros ( $\bullet$ ) of the open-loop system with buffer pressure,  $P_1$ , measurement with respect to valve openings,  $Z$ , selected by colorbar.

(RPS) analysis for this system has a Lower Gain margin of 0.36, Phase margin of  $65.34^\circ$ , Delay margin of 1.90s,  $\|S\|_\infty : 1.03$  and  $\|T\|_\infty : 1.57$ .

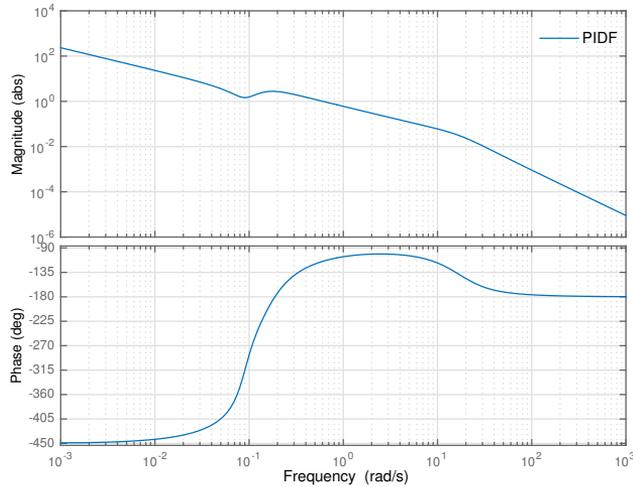


Figure 4.3: Bode plot of open-loop system linearized with a valve opening of 30% with an optimal PIDF controller. Lower Gain Margin: 0.36, Phase Margin:  $65.34^\circ$ , Delay Margin: 1.90,  $\|S\|_\infty : 1.03$  and  $\|T\|_\infty : 1.57$ .

The results generated with the four state model illustrated the challenges regarding controlling the closed-loop. Poles are located in the RHP and system gain increases asymptotically towards zero.

## 4.2 Self-tuning system based on response slope

All self-tuning experiments had a run time of 60 minutes in total with the optimal PIDF as baseline controller. The set point changes are identical for all experiments, and are presented in Table 4.1. A reference run with constant controller parameters, hence no adaptation, is presented in Figure 4.4. This figure covers both the pressure in upper figure, and the valve opening in the lower figure. This is the reference for comparison for the results with adaption enabled. The controller was updated using following approaches:

1. Standard form: Update  $K_p$
2. Standard form: Update  $K_p$  and  $\tau_i$
3. Parallel form: Update  $K_p$
4. Parallel form: Update  $K_p$  and  $K_i$

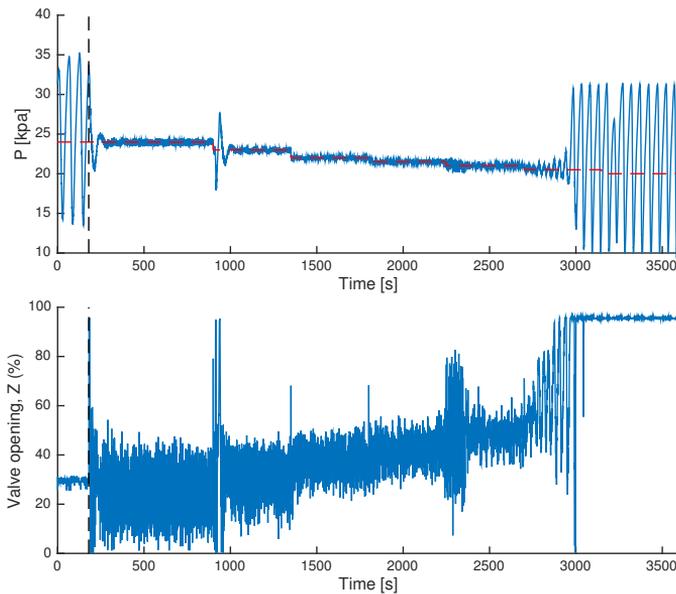


Figure 4.4: Experimental data with fixed gain with optimal PIDF. Black dashed line marks the controller activation. In upper figure: subsea pressure with set point pressure (red dashed line) from Table 4.1. In lower figure: Valve opening,  $0 \leq Z \leq 100$ , with respect to set point changes. Oscillatory slugging at set point,  $r = 21.0\text{kPa}$ .

The implementation of the self-tuning system based on slope response, required filtering to estimate new controller parameters.

### Butterworth filter

A second order,  $n = 2$ , Butterworth filter with a normalized cut off frequency at,  $\omega = 0.02$ , was applied to the pressure measurement in order to simplify the estimate of the closed loop time constant,  $\tau_c$ . The effect of the digital filter is presented in Figure 4.5. In this figure the blue line is the actual measured signal, with the red line demonstrating the effect of the applied digital filter. Meta figure is a zoomed callout of an area containing a given set point change.

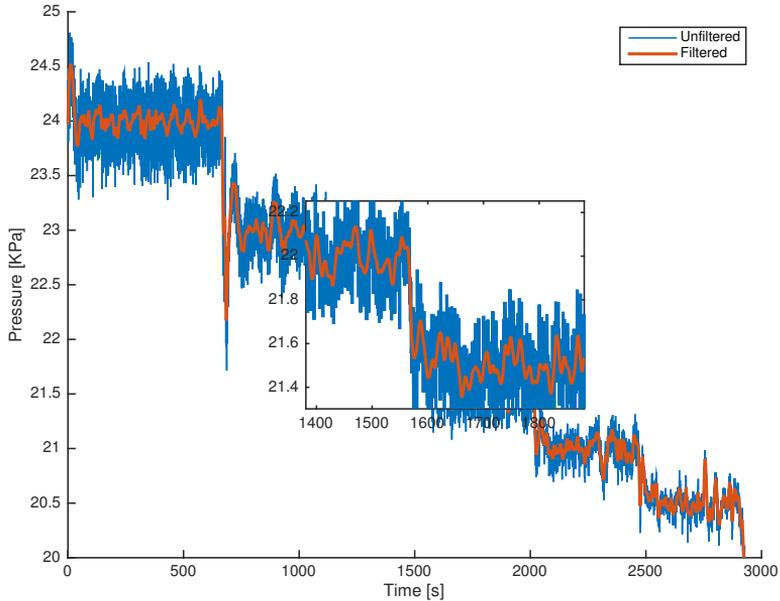


Figure 4.5: Experimental data: Pressure measurement with filtering (red solid line), and without filtering (blue solid line).

The most effective tuning system were found to be the approach updating the controller with new  $K_p$  and  $\tau_i$  in standard form. This extended the stable valve opening from 42% to around 60%. The results of this experiment is presented in Figure 4.6. There were, however, minor differences between updating  $\tau_i$  or not, but nevertheless better. Additional results are presented in Appendix A. The desired time constant  $\tau_{des}$  were estimated through experiments to be 3 seconds. The tuning parameter,  $\gamma$ , were in all experiments set to  $\gamma = 1$ .

The new controller updated in standard form with its respective parameters are presented in Figure 4.7.

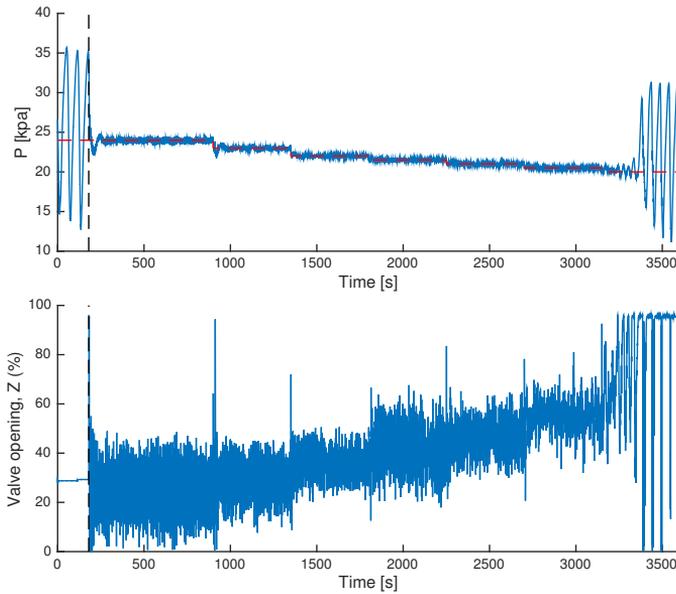


Figure 4.6: Experimental data with self-tuning system updating  $K_p$  and  $\tau_i$  using optimal PIDF controller in standard form (Equation 1.2a). Black dashed line marks the controller activation. In upper figure: subsea pressure with set point pressure (red dashed line) from Table 4.1. In lower figure: Valve opening,  $0 \leq Z \leq 100$ , with respect to set point changes. Oscillatory slugging at set point,  $r = 20.0\text{kPa}$ . The updated gain and integral time is presented in Figure 4.7.

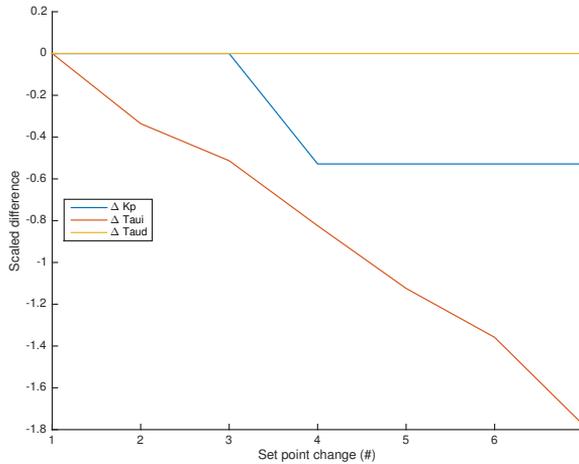


Figure 4.7: Change of controller parameters,  $K_p, \tau_i, \tau_d$  during adaptation with tuning parameter  $\gamma = 1$  and  $\tau_{des} = 3$ . The controller were updated in standard form (equation 1.2a). Table 4.1 presents the actual set points with respect to actual set point change number. Note that the parameters are scaled, that is  $\Delta K_p = \frac{K_{p,0} - K_{p,\#}}{K_{p,0}}$ , were  $K_{p,0}$  is the optimal controller gain at nominal operating point (Table 1.1).

### 4.3 Adaptive system based on scalar reference model

#### 4.3.1 Simulation

The adaptive system based on scalar reference model was simulated using the optimal PIDF controller as baseline. Simulative run with fixed controller, hence no adaptation was conducted for proper comparison. This result is presented in Figure 4.8.

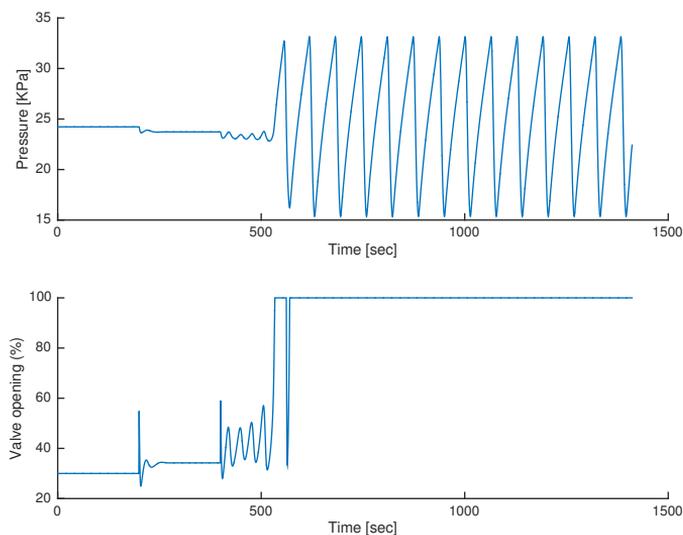


Figure 4.8: Simulated data with fixed PIDF controller, hence no adaptation.

With use of trail and error, the proper tuning parameters during simulations were found to be  $\gamma_x = \gamma_r = \gamma_u = \gamma_s = 1$ . These parameters extended the stable valve opening from 37% to around 65%. The simulated result is presented in Figure 4.9.

The calculated adaptive gains during simulation used in the adaptive laws are presented in Figure 4.10.

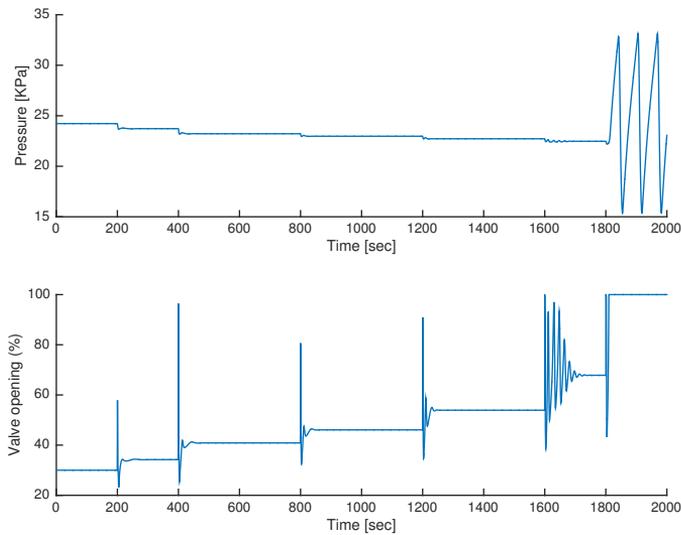


Figure 4.9: Simulations of the adaptive system based on scalar reference model. Tuning parameters used during simulation were  $\gamma_x = 1$ ,  $\gamma_r = 1$ ,  $\gamma_u = 1$ , and  $\gamma_s = 1$ .

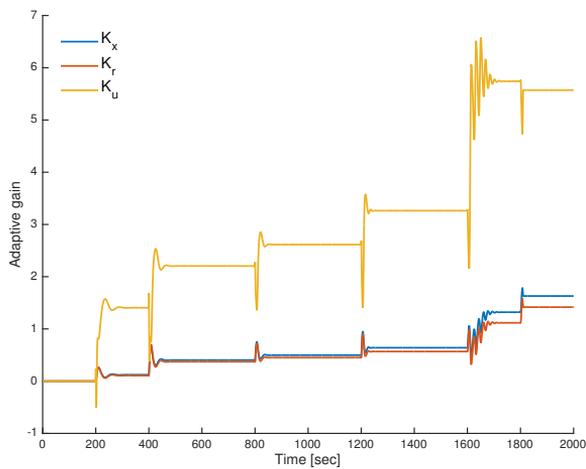


Figure 4.10: Adaptive system based on simulated scalar reference model. Tuning parameters used during simulation were  $\gamma_x = 1$ ,  $\gamma_r = 1$ ,  $\gamma_u = 1$ , and  $\gamma_s = 1$ .

### 4.3.2 Experimental

All the experimental results are presented in a run time of 45 minutes in total. The set point changes are identical for all experiments, and are presented in Table 4.1. Experimental run with fixed PIDF controller, hence no adaptation was conducted for proper comparison. This result is presented in Figure 4.11.

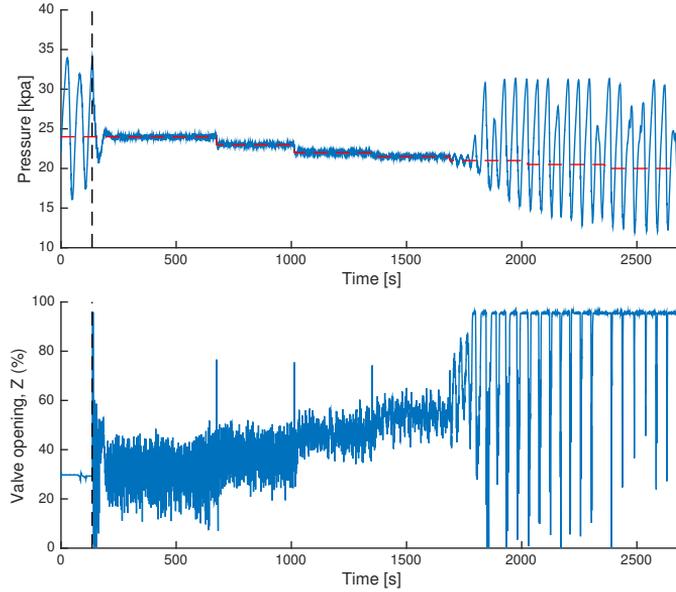


Figure 4.11: Experimental data with fixed PIDF controller, hence no adaptation. Black dashed line marks the controller activation. Oscillatory slugging at set point,  $r = 21.0\text{kPa}$ .

The adaptive tuning parameters were found to be  $\gamma_x = 0.5$ ,  $\gamma_r = 1.7$ ,  $\gamma_u = 0.07$ , and  $\gamma_s = 0.07$  by trial and error. These parameters extended the stable experimental valve opening from 58% to around 72%. The experimental result of the adaptive system based on scalar reference model is presented in Figure 4.12.

The calculated adaptive gains used in the adaptive laws are presented in Figure 4.13.

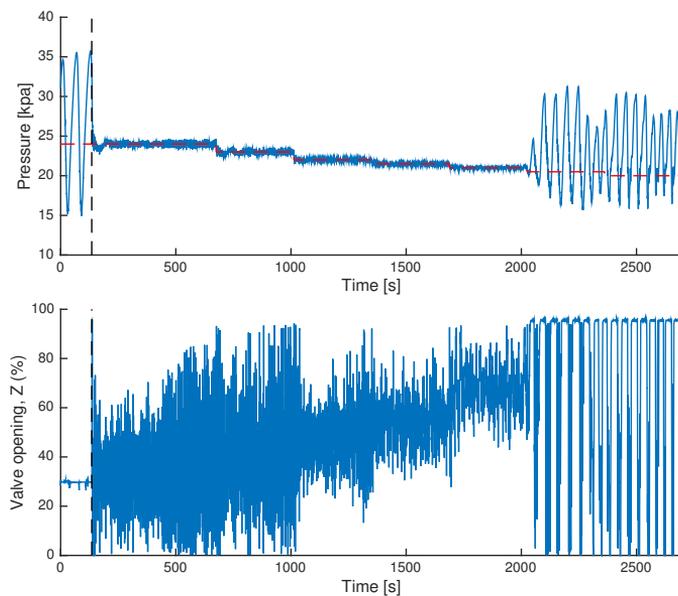


Figure 4.12: Experimental results with adaptive system based on scalar reference model. Black dashed line marks the controller activation. Oscillatory slugging at set point,  $r = 20.5\text{kPa}$ . Tuning parameters used during experiment were  $\gamma_x = 0.5$ ,  $\gamma_r = 1.7$ ,  $\gamma_u = 0.07$ , and  $\gamma_s = 0.07$ .

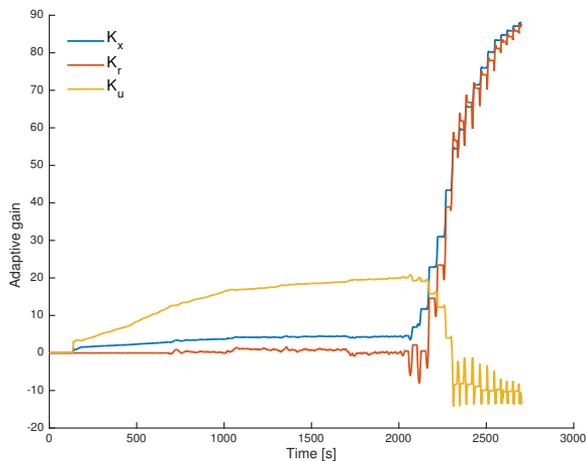


Figure 4.13: Experimental results with adaptive gains based on scalar reference model. Tuning parameters used during the experiment run were  $\gamma_x = 0.5$ ,  $\gamma_r = 1.7$ ,  $\gamma_u = 0.07$ , and  $\gamma_s = 0.07$ .

#### 4.4 Adaptive system based on augmented state observer reference model

The state space was extended to include the integrated error as a fictitious physical state. The LQR controller was designed, followed by Loop-Transfer-Recovery using a dynamic observing model. Delay margin, settling time and maximum controller output was calculated in order to determine the robustness and performance of the optimal LQR controller with respect to a scalar penalty parameter  $q$ . The results of the effect of  $q$  to the design of LQR is presented in Figure 4.14.

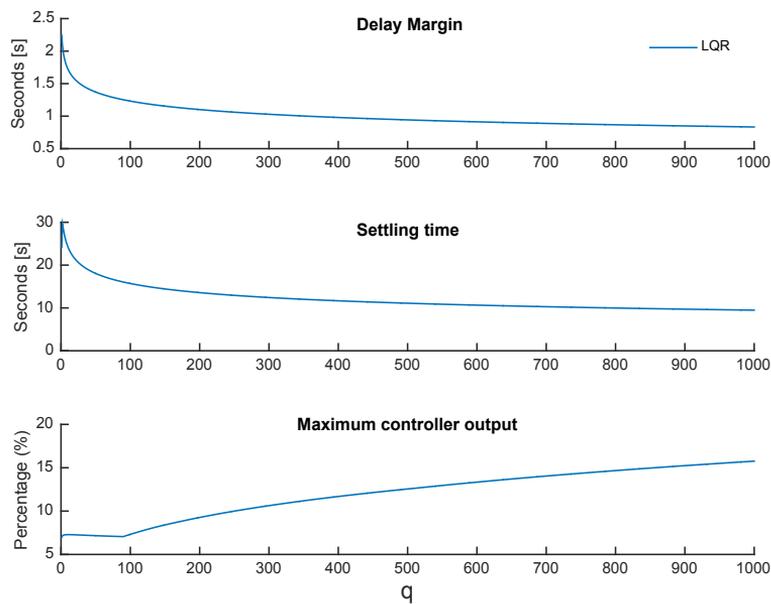


Figure 4.14: Relationship between Delay Margin, Settling Time and Maximum controller output with respect to tuning parameter,  $q$ , for optimal controller, LQR.

The stability of the LQR-controller corresponding to the effect of tuning parameter,  $q$ , was identified using Bode plot. The results are presented in Figure 4.15, with the corresponding stability margins and robustness analysis presented in Table 4.2.

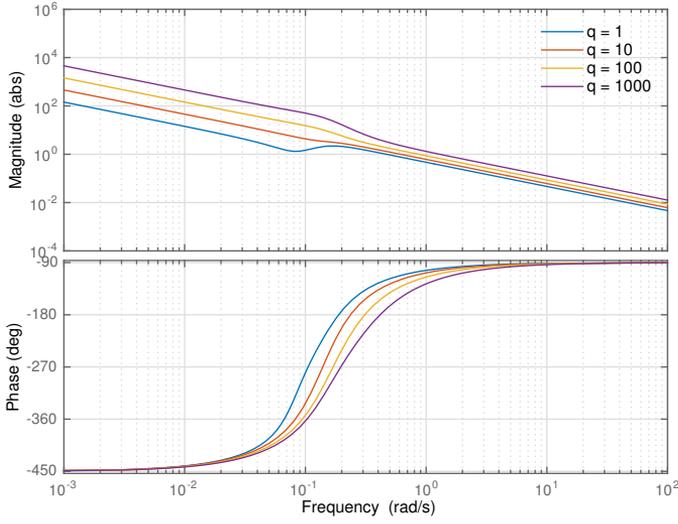


Figure 4.15: Bode plot of open-loop dynamics with optimal LQR controller, with tuning factor set to  $q = 1$ ,  $q = 10$ ,  $q = 100$  and  $q = 1000$ .

Table 4.2: Robustness, Performance and Stability (RPS) analysis with respect to tuning scalar  $q$  for the optimal quadratic controller LQR.

$q$	1	10	100	1000
Lower Gain Margin	0.46	0.40	0.33	0.26
Phase Margin ( $^{\circ}$ )	60.14	60.42	60.95	61.77
Delay Margin (s)	2.24	1.72	1.23	0.83
$\ S\ _{\infty}$	1.00	1.00	1.00	1.00
$\ T\ _{\infty}$	1.87	1.74	1.63	1.52

The optimal quadratic controller, LQR, were calculated to be as presented in Equation (4.2), using the scalar penalty factors  $q = 10$ , and  $R = 1$ .

$$K_{LQR} = 10^4 \begin{bmatrix} -5.2409 & -0.0583 & -1.3856 & -0.0257 & 0.0003 \end{bmatrix} \quad (4.2)$$

With the calculated gain matrix the corresponding open-loop transfer function,  $L_{LQR}$ , was obtained and is described in Equation (4.3).

$$L_{LQR}(s) = \frac{0.45484(1 + 1.036(7.714s) + (7.714s)^2)}{s(1 - 1.143(6.645s) + (6.645s)^2)} \quad (4.3)$$

The stability of the LQG/LTR-controller with respect to tuning parameter,  $v$ , and constant  $q_{LQR} = 10$ , was observed using Bode plot. The result is presented in Figure 4.16, with the corresponding stability margins presented in Table 4.3.

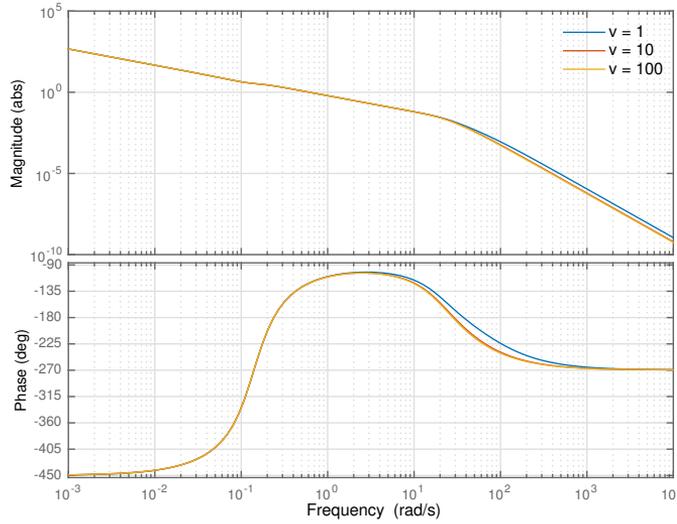


Figure 4.16: Bode plot of optimal LQG/LTR controller, with varying tuning factor  $v = 1$ ,  $v = 10$ , and  $v = 100$ . Penalty factor  $q_{LQR} = 10$  is constant for all  $v$ .

Table 4.3: Robustness, Performance and Stability (RPS) analysis with respect to tuning scalar  $v$  for the optimal controller LQG/LTR with constant  $q_{LQR} = 10$ .

$v$	1	10	100
Lower Gain Margin	0.403	0.404	0.405
Phase Margin ( $^{\circ}$ )	59.36	59.01	58.95
Delay Margin (s)	1.68	1.675	1.67
$\ S\ _{\infty}$	1.03	1.03	1.03
$\ T\ _{\infty}$	1.75	1.75	1.75

The observer gain was calculated to be as presented in Equation (4.4), with a penalty parameter,  $q_{LQR} = 10$ , tuning parameter  $v = 100$ . Process - and measurement covariance defined as  $Q_0 = Q$  and  $R_0 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ , respectively.

$$L_v = \begin{bmatrix} 0.0109 & 0.4678 & -0.0054 & 9.1545 & -0.9589 \\ -0.0002 & -0.0041 & 0.0002 & -0.1701 & 2.4593 \end{bmatrix}^T \quad (4.4)$$

A comparison of the increased robustness between LQR and LQG/LTR is presented in Figure 4.17. In this figure the Linear-Quadratic-Regulator controller is plotted in comparison to the effect of Loop-Transfer-Recovery. These controllers were designed with a penalty parameter,  $q_{LQR} = 10$ , and tuning parameter  $v_{LTR} = 100$ .

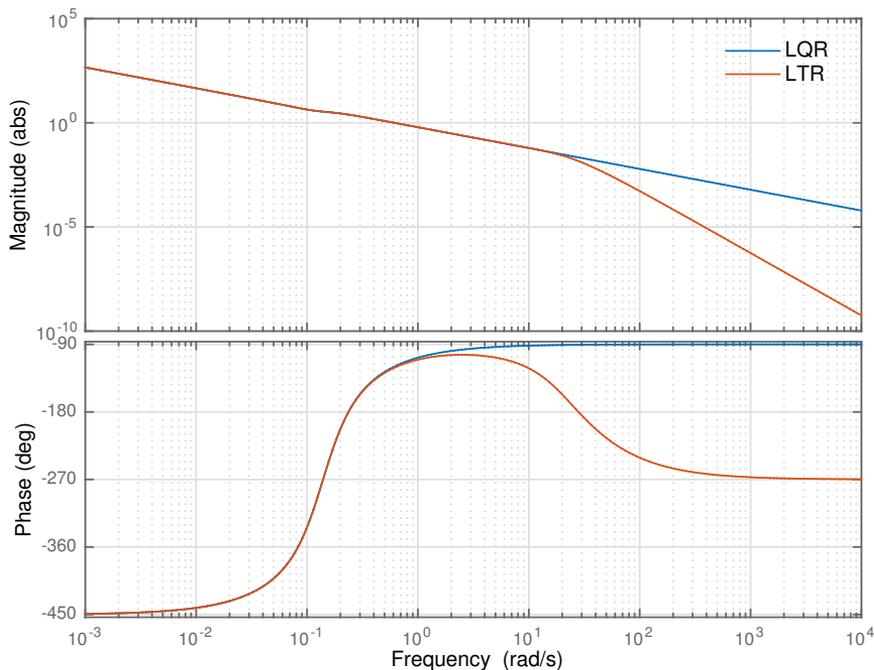


Figure 4.17: Comparison of Bode plot with optimal controller LQR and LQG/LTR, with tuning parameters  $q_{LQR} = 10$  and  $v_{LTR} = 100$ .

A comparison of the robustness and performance of PIDF, LQR and LQG/LTR in terms of sensitivity and complementary sensitivity is presented in Figure 4.18.

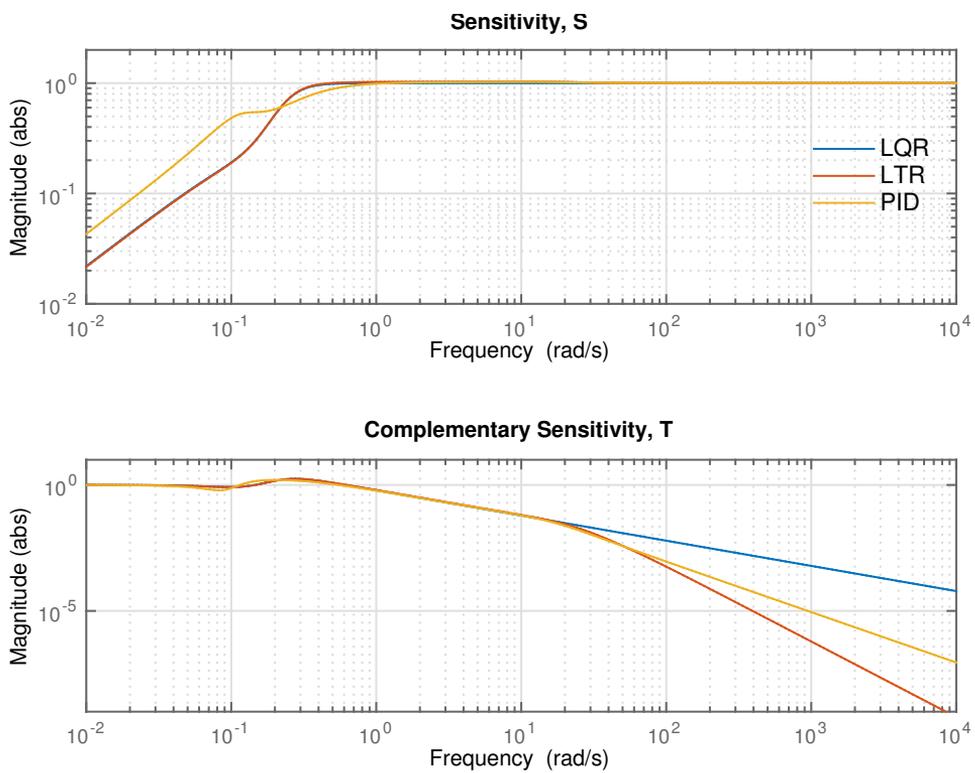


Figure 4.18: Bode magnitude of optimal PIDF, LQR and LQG/LTR controller. In upper figure: Sensitivity,  $S$ . In lower figure: Complementary Sensitivity,  $T$ . Tuning parameters are  $q_{LQR} = 10$  and  $v_{LTR} = 100$ .

#### 4.4.1 Simulations

The adaptive system based on observer-like reference model was simulated using LQG/LTR controller as baseline. Simulative run with fixed controller, hence no adaptation was conducted for proper comparison. This result is presented in Figure 4.19.

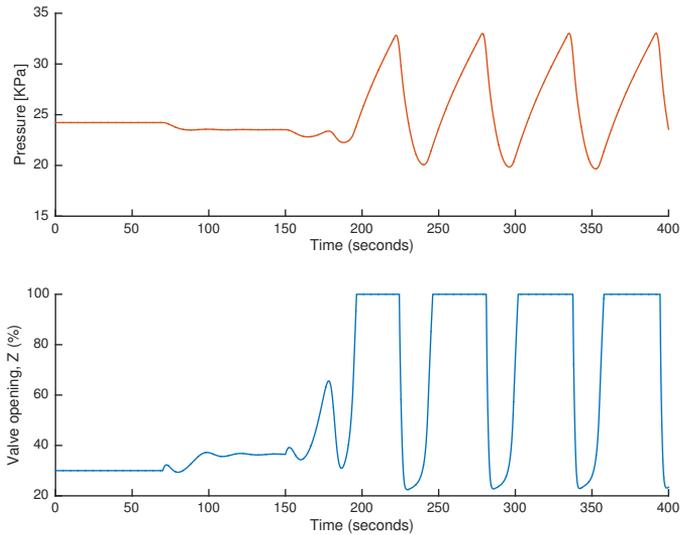


Figure 4.19: Simulated data with fixed Linear-Quadratic-Gaussian with Loop-Transfer-Recovery controller, hence no adaptation.

With use of trail and error, the proper tuning parameters were found to be  $\gamma_p = 1000$ ,  $\gamma_u = 100$  and  $\Theta_{max} = 10$ . These parameters extended the stable valve opening from 47% up to 90%, during simulations. The result is presented in Figure 4.20.

The calculated adaptive gains during simulation used in the adaptive law are presented in Figure 4.21.

The absolute error between observing model and process model is presented in Figure 4.22.

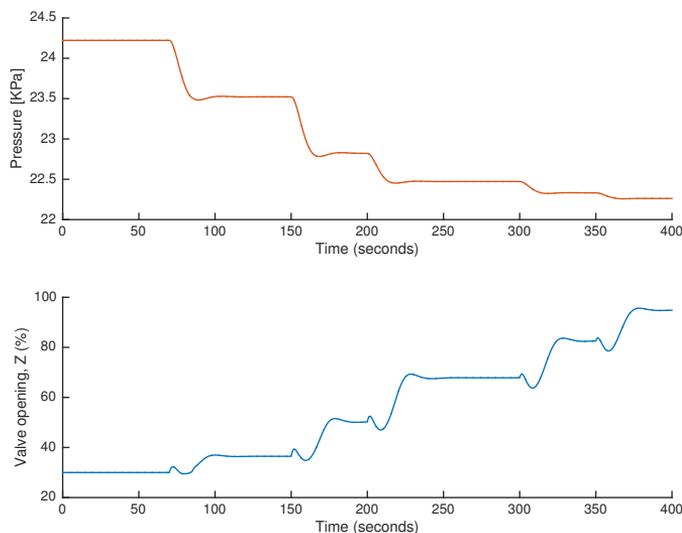


Figure 4.20: Simulations of observer-like reference model with use of LQG/LTR adaptive system. Tuning parameters used during simulations were  $\gamma_p = 1000$ ,  $\gamma_u = 100$  and  $\Theta_{max} = 10$ .

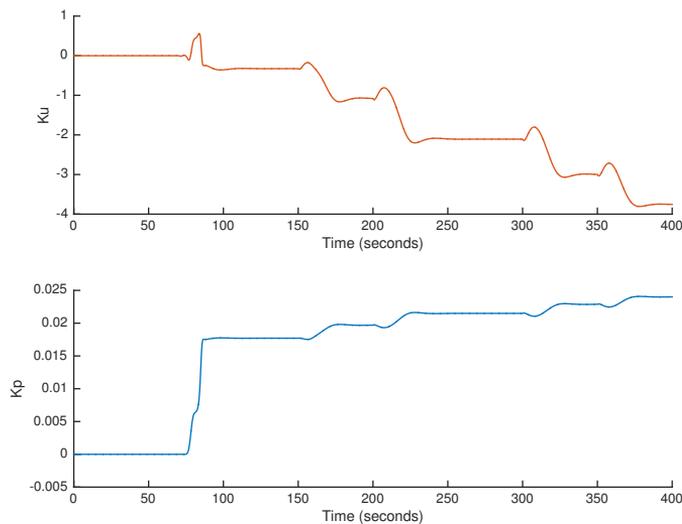


Figure 4.21: Change of adaptive gains using observer-like reference model with LQG/LTR adaptive system, during simulation. The presented adaptive gains is in reference to Figure 4.20. In upper figure: Adaptive gain with respect to baseline controller usage,  $K_u$ . In lower figure: Adaptive gain with respect to observed system states,  $K_p$ . Tuning parameters used during the simulation were  $\gamma_p = 1000$ ,  $\gamma_u = 100$  and  $\Theta_{max} = 10$ .

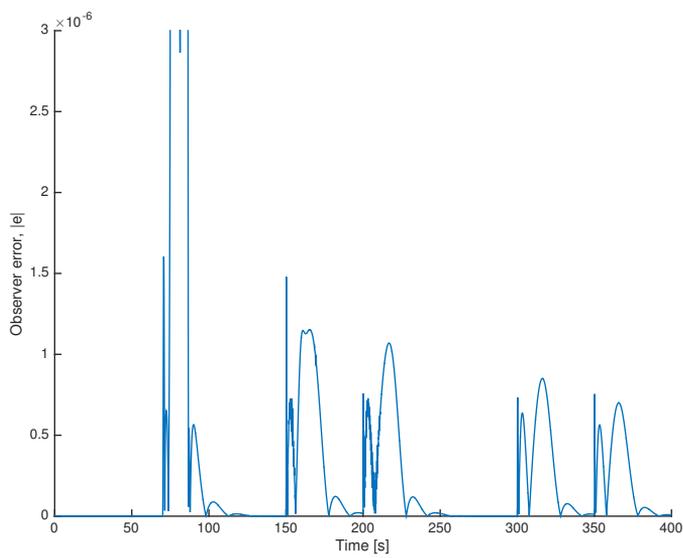


Figure 4.22: Simulated data showing the scaled absolute error between observing model and process model. Each "bumps" represents the error between the actual system and the observing model at a given set point change. Each peak represents a set point change in which the adaptive laws are adapting to.

### 4.4.2 Experimental

All the experimental results are presented in a run time of 45 minutes in total. The set point changes are identical for all experiments, and are presented in Table 4.1. Experimental run with fixed LQG/LTR controller, hence no adaptation was conducted for proper comparison. This result is presented in Figure 4.23. The dead-zone boundary was determined by comparison of the scaled absolute output error between observing model and system output.

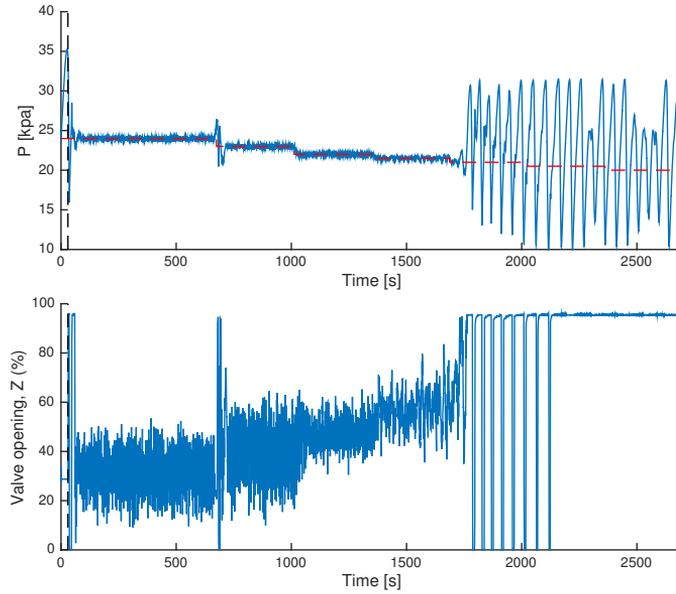


Figure 4.23: Experimental data with fixed LQG/LTR controller, hence no adaptation. Black dashed line marks the controller activation. Oscillatory slugging at set point,  $r = 21.0\text{kPa}$ .

The dead-zone were found to be  $\epsilon_{min} = 0.0016$ , by observation of the scaled absolute error between observing model and process model. This value was chosen to prevent the adaptive system to adapt to the large amount of noise differences between the actual process, and the observer model. The result illustrating the dead-zone is presented in Figure 4.24.

The experimental adaptive tuning parameters were found to be  $\gamma_p = 0.05$ ,  $\gamma_u = 0.005$  and  $\Theta_{max} = 10$  by trial and error. These parameters curtailed the stable experimental valve opening from 57% to around 56%. The experimental result is presented in Figure 4.25.

The calculated adaptive gains during the experiment used in the adaptive laws are presented in Figure 4.26.

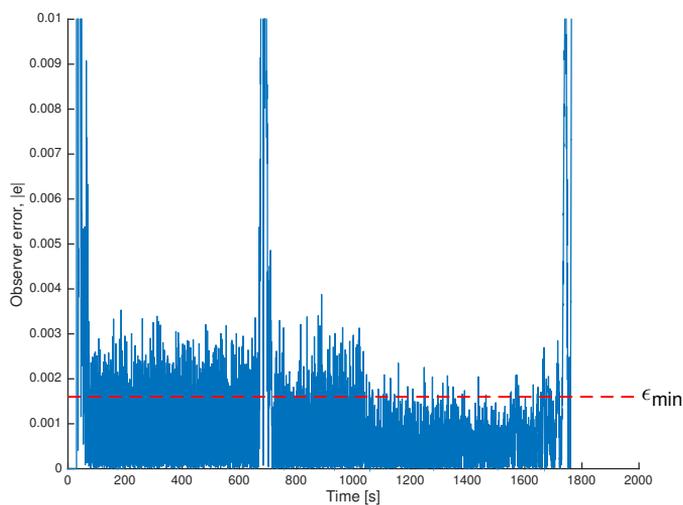


Figure 4.24: Experimental data for scaled absolute error between Observing model and actual model. Note the lack of obvious "bumps" compared to the simulated data presented in Figure (4.22). The presented error is in reference to Figure 4.23. At  $t \approx 1700s$  the system goes completely unstable and both adaptive gains and the controller saturates.

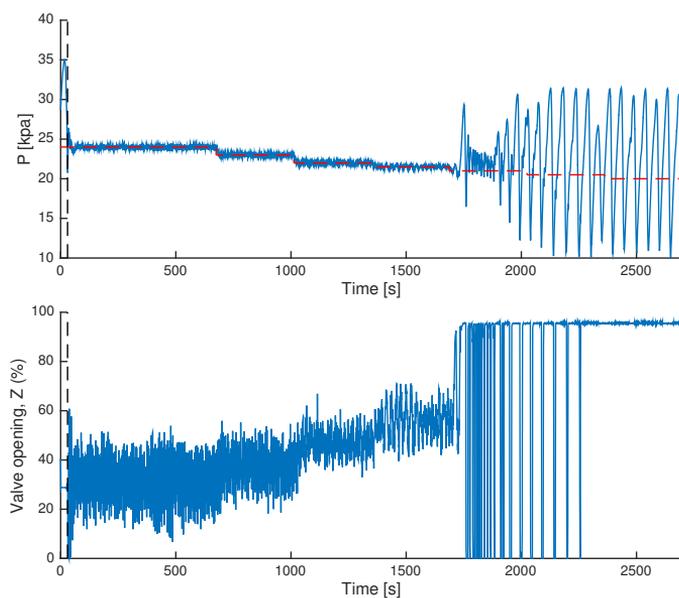


Figure 4.25: Experimental data with LQG/LTR adaptation enabled. Black dashed line marks the controller activation. Oscillatory slugging at set point,  $r = 21.kPa$ . Tuning parameters used during experiment were  $\gamma_p = 0.05$ ,  $\gamma_u = 0.005$  and  $\Theta_{max} = 10$ .

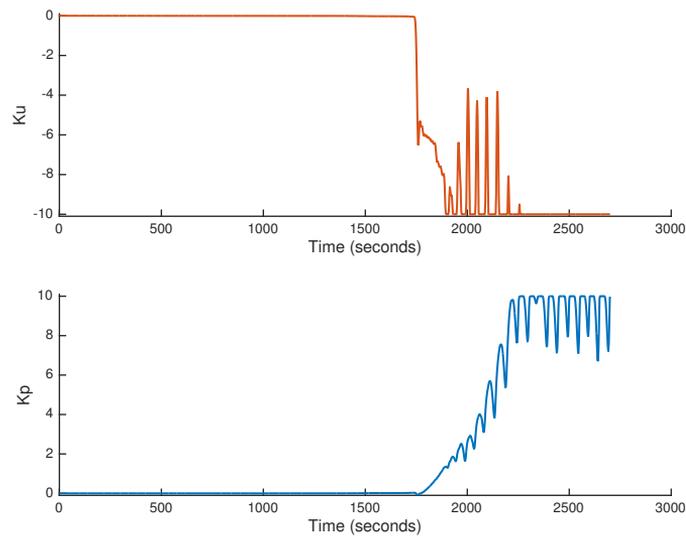


Figure 4.26: Change of adaptive gains using observer-like reference model with LQG/LTR adaptive system, during experiment. The presented adaptive gains is in reference to Figure 4.25. In upper figure: Adaptive gain with respect to baseline controller usage,  $K_u$ . In lower figure: Adaptive gain with respect to observed system states,  $K_p$ . Tuning parameters used during the experiment run were  $\gamma_p = 0.05$ ,  $\gamma_u = 0.005$  and  $\Theta_{max} = 10$ .



## Chapter 5

---

# Discussion

---

Anti-slug control using the top-side choke valve in combination with buffer pressure measurement is an unstable system, but nonetheless controllable. The system gain increases asymptotically towards zero, with increased valve opening. This implies that the preservation of system stability will be difficult and in a nonlinear relationship with the valve opening. Previous work conducted by De Oliveira et al.<sup>20</sup> has shown that an adaptive anti-slug controller using the top-side choke valve can be driven to toward its operational limit using observer-like Model Reference Adaptive Controller (MRAC). This adaptive system is, however, based on measurements of both the buffer pressure, and pressure at top of the riser pipeline - along with a supervisor system.

The aim in this thesis was to investigate simple, yet robust adaptive system be integrated into the severe slugging system based on only one physical measurement, namely the buffer pressure. A large part of this thesis is based on previous work by Lavretsky and Wise<sup>15</sup>, who uses their adaptive systems within aerospace. This thesis is thus partly demonstrating how this may be applied within chemical engineering as well.

The economical perspective of a voluminous mass transportation emphasizes the importance of a stable large valve opening. The stable valve opening is a better representation of the controller performance, compared to the evaluation of a low set point operation. This is based on an unknown ideal set point, which are dependent on current operating conditions. This illustrates the necessity of a supervising system as previously described. The supervisory control can be used to quickly adjust the operation conditions, such as set point. If the controlled variables are saturated, the adaptive controller will be inoperable, hence the system stability will be difficult to regain. A supervising system will in addition reduce the required

human interaction.

All experiments had to be ran on the same day to eliminate operating condition differences. This is valid for all experiments, except the self-tuning system. This implies that a direct comparison of the self-tuning design in reference to the adaptive designs are not completely valid. Results presented from simulations should not be directly compared to the experimental results either. The experimental system is highly influenced by randomized noise, and the rather unknown persistent accuracy of the four state model. Measurement noise in the system appears due to turbulence flow in rather thin pipelines. This alters the measurement. The simulative results were only used as "proof-of-concept" and to indicate how the experiment should be tuned and optimized. The goal was to design a robust adaptive system on the experimental rig. Fine-tuning in simulations were therefore given a low priority.

All experiments ended in instability were the magnitude of the controller gain eventually was too low based on the span of the valve opening compared to a fixed controller at nominal operating conditions. With increasing valve opening, the controller gain must decrease to counteract the nonlinearity. The outcome of a controller with a too low gain was evidently instability. This implies that the adaptive systems are not compensating enough to keep the system stable at its critical operating point. The Lyapunov stability criterion requires the system to orbit within some given boundary, and thus close to system equilibrium. This seems to be a too strong requirement in the experimental system with its increasing degree of unknown disturbances at decreasing operating pressure.

## 5.1 Simplified four-state model simulation

The mapped relationship between system gain and valve opening shows that the system gain increases asymptotically towards zero, with increased valve opening. This implies that the preservation of system stability will be difficult and in a nonlinear relationship with the valve opening.

The pole-zero plot reveals the poles in the Right-half plane. As the valve opening increases, the poles move further into the complex plane, which implies that the system stability are more challenging to maintain.

## 5.2 Self-tuning system based on response slope

As the self-tuning system is considered a one-time event in terms of a set point change, the system was dependent on flawless calculations. A consequence of a

miscalculation would result in instability, with no point of return without a supervising system or human interaction. The performance of this system basically depends on the performance of the filter. As the filter induces a nonlinear phase shift, the signal is altered, thus the response slope. This could have been improved by use of a Savitzky-Golay filter. This filter is known to increase the signal-to-noise ratio with limited distortion of the signal.<sup>33</sup> The self-tuning system has to assume that any response it observes is a result of the induced excitation, and the result may easily be confused with high frequent oscillations that interacts with the measurements. As the noise was a critical factor, the system was implemented directly into the experimental rig without simulations.

The optimal PIDF controller was updated in both standard - and parallel form. A combination of  $K_p$  and  $\tau_i$  updated in standard form was shown to be the most effective approach. There were, however, minor differences between updating  $\tau_i$  or not, but nevertheless better.

One advantage with the self-tuning system is that an operator may more easily supervise the re-tuned controller before its applied. The necessary security check is simpler in periodic tuning, compared to continuously updated adaptive gains. One major disadvantage, on the other hand, is that the adaptive part can not be disabled, since the PIDF controller is being updated rather than used as a baseline.

The application of this system was shown to be rather impractical. The calculated controller was often miscalculated, which resulted in instability. This may be due to the filtering or the approximation of a first order system.<sup>22</sup>

### 5.3 Adaptive system based on scalar reference model

The adaptive system based on a scalar reference model is to some degree easy to implement. This is practical in situations where the system state space matrices,  $A$  and  $B$ , are unknown. The only necessary information is the knowledge of control effectiveness, *sign* of  $B$ . The output vector  $y$  is, however, assumed to be the system state upon design. This is generally the case with PID-controller, where it is driven by the error between set point and actual process output. The adaptive parameters  $\hat{K}_x$ ,  $\hat{K}_r$  and  $\hat{K}_u$  are not guaranteed to converge to their true unknown values,  $K_x$ ,  $K_r$  and  $K_u$ . These parameters are only guaranteed to remain uniformly bounded in time. Sufficient conditions for parameter convergence are known as persistency of excitation.<sup>15,34</sup>

The first order open-loop reference model with a desired time delay,  $\tau_{des} = 3$ , were proven to be capable of driving the adaptive system towards a larger valve opening compared to the fixed controller during experiments. Both simulative re-

sults and experimental results presented a stable valve opening up to 65% and 72% respectively. In comparison of the adaptive gains in simulation and experimental, it was observed that all gains were following the same pattern, until the system got unstable and saturated.

The optimal PIDF controller was used as baseline during adaptation. It would be interesting to observe the effect of an LQR as the baseline controller, in combinations with scalar reference model.

## 5.4 Adaptive system based on observer-like reference model

The popularity of these methods is due to the guaranteed properties, such as closed-loop stability and robustness to uncertainties. This is, however, based on if and how accurate the physical states are actually measured. The observer-like model is highly dependent on a accurate model representation of the physical system.

The integrated error was included into the system state vector by augmentation of the state space. The integrated error was thus interpreted as an additional state. This was proven to be effective in terms of following tracking commands. The adaptive system with observer-like reference model is the most complex adaptive system evaluated in this thesis. The optimal LQR controller with Loop-Transfer-Recovery involved solving a complex cost function along with a state observing design. This is not as intuitive as the common known PID.

As a result of Loop-Transfer-Recovery the frequency roll-off was indubitable improved, thus improving the robustness of the controller. The LTR was able to recover the gain -, phase - and delay margin as designed by the LQR. With increased penalty parameter  $q_{LQR}$  the settling time improved, while delay margin receded and maximum controller output increased. This illustrates that a certain trade-off has to be taken into account while designing the optimal control system.

Simulative results presents outstanding adaptive performance with a stable valve opening up top 90%, while experimental results gave a stable valve opening of mediocre 56%. The absolute error from observing model and actual output ( $|\hat{y} - y|$ ), is clearly illustrating the necessary transients for adaptation. The noise from the experimental system conceals the necessary transients. The absolute output error is mostly influenced with randomized noise error. The dead-zone is not improving the adaptive process enough. In the scalar system, the output is directly compared to the reference model - hence not using a observing state. The experimental adaptive controller output were close to zero with the given tuning parameters. If the tuning parameters were to be increased, the adaptive contribu-

tion was too overwhelming and thereby resulted in instability. Previous research conducted by Jahanshahi<sup>19</sup> found that a Luenberger observer-like reference model was not robust using the buffer pressure,  $P_1$ . This observer was designed with optimal state estimator, instead of optimal Loop-Transfer-Recovery as evaluated in this thesis. The results are, nevertheless, the same. The conclusion to instability problems was believed to be due to the chain of integrating sections from the buffer pressure to top pressure,  $P_2$  as noted in Figure 3.1. The adaptive laws for the observer model requires more measurable variables than output variables.<sup>15</sup> The observing model was designed on a "fabricated" measurable variable (the integrated error). This may also be a reason for the nonviable observing reference model. This is supported by previous work by De Oliveira et al.<sup>20</sup>, which are successfully adapting the same rig using two physical measurement, in addition to the integrated error. These physical measurements involves the buffer pressure,  $P_1$  in addition to top pressure,  $P_2$  and drives the maximum stable valve opening to around 78%.

The experimental results comparing fixed control PIDF and LQG/LTR, i.e. adaptation disabled, showed minor performance differences. This is supported by their respective Bode and sensitivity plots. There were small differences in terms of stability margins, robustness and performance. The PIDF was, nevertheless, better in terms of stability and performance compared to LQG/LTR. The complementary sensitivity was greater than the sensitivity for all control systems. That is  $\|T\|_\infty > \|S\|_\infty$ . This is often observed in unstable processes, and implies that the system will be more fragile to process noise on behalf of better set-point tracking.<sup>21</sup>



# Conclusion

---

Offshore production using pipeline-riser system are experiencing cyclic flow instability due to its pipeline configuration where a liquid blockage occurs at the lowest point, followed by accumulation of gas at its highest point. The accumulation of gas in the pipeline will eventually reach its maximum and violently burst out. This cyclic flow instability has been referred to as severe slugging. The aim in this thesis was to investigate simple, yet robust adaptive systems to be integrated into the slugging rig. Three types of adaptive systems were evaluated in order to increase the robustness of an already existing optimal anti-slug control system. One self-tuning approach, and two Model Reference Adaptive Controllers (MRAC) - with increasing complexity.

The three types of control systems investigated was an optimal PIDF previously presented by Jahanshahi and Skogestad<sup>7</sup>, and a Linear-Quadratic-Regulator followed by Loop-Transfer-Recovery. The stability margins with respect to each control system are presented in Table 6.1. The complementary sensitivity was greater than the sensitivity for all control systems. That is  $\|T\|_{\infty} > \|S\|_{\infty}$ . The experimental results comparing fixed PIDF and LQG/LTR, i.e. adaptation disabled, showed minor performance differences. The PIDF was, nevertheless, better in terms of stability and performance compared to LQG/LTR.

The adaptive results in this thesis were evident. There was a clear difference between self-tuning and the adaptive systems, in which the scalar reference model outperformed the others. The self-tuning system was based on updating the controller using the slope after system excitation, along with a desired closed-loop constant. Its main drawback was that a miscalculation would result in instability with no point of return. This could be a result of improper filtering or the assumption of a first-order system. The performance of the self-tuning was more based on

Table 6.1: Robustness, Performance and Stability analysis with respect to investigated controllers.

Controller	GM <sub>L</sub>	PM (°)	DM (s)	$\ S\ _{\infty}$	$\ T\ _{\infty}$
PIDF	0.36	65.34	1.90	1.03	1.57
LQR	0.40	60.42	1.72	1.00	1.74
LQG/LTR	0.405	58.95	1.67	1.03	1.75

luck, rather than consistent stability. The observer-like reference model adaptive system was not efficient enough using only one measurement. This may be due to the large noise, which alters the error between the observing model and the actual process. This is supported by the results in simulation, and previous work conducted by De Oliveira et al.<sup>20</sup> where multiple measurements were used. They achieved a maximum stable valve opening up to around 78%. Detailed comparison of each system evaluated in this thesis are presented in Table 6.2.

Table 6.2: Comparison of investigated adaptive systems and fixed controllers with Integrated Absolute Error, maximum stable valve opening and lowest stable operating pressure.

Adaptive law	IAE	Max stable valve opening (%)	Lowest stable pressure [KPa]
Slope based*	3156	60	20.5
Fixed PIDF*	6035	42	21.5
Scalar MRAC	3887	72	21.0
Observer MRAC	5130	56	21.5
Fixed PIDF	5525	58	21.5
Fixed LQG/LTR	5586	57	21.5

\* Slope based self-tuning system was updated using standard form with both  $K_p$  and  $\tau_I$ . The asterisked entries were not ran under the same conditions as the scalar and LQG/LTR, and are therefore not fully representative in comparison.

This thesis have illustrated both the necessity of optimal adaptive control, and the challenges involved by maintaining consistent closed loop-performance in presence of uncertainties and disturbances.

## 6.1 Future work

In the future it would be interesting to investigate the effect of the scalar reference model with a optimal LQR controller as baseline. A supervising system should also be implemented into the scalar reference model adaptive system. A reasonable presentation of the results using a supervising system would be to use a mean valve opening throughout the whole experiment.

Pressure reading at other locations should also be investigated with both scalar reference model and Linear-Quadratic-Gaussian with Loop-Transfer-Recovery. Previous research shows contrasting controllability issues.<sup>19</sup> For the sake of completeness, all experiments should be run with equal operating conditions.

The flow rate feed into the pipeline was kept constant during all simulations and experiments. Low pressurized oil reservoirs and more dense liquid fractions are more prone to instabilities, and should be investigated with the scalar reference model adaptive system to map robustness differences.



---

# Bibliography

---

- [1] M.G. Gerritsen and L.J. Durlofsky. Modeling fluid flow in oil reservoirs. *ANNUAL REVIEW OF FLUID MECHANICS*, 37:211–238, 2005. ISSN 0066-4189. doi: 10.1146/annurev.fluid.37.061903.175748.
- [2] K. Havre, K. O. Stornes, and H. Stray. Taming slug flow in pipelines. *ABB review*, 4:55–63, 2000.
- [3] Schlumberger. Understanding gas-condensate reservoirs. *Oilfield Review*, 17(4), 2005.
- [4] P.J. O’Brien and J.F. McNamara. Significant characteristics of three-dimensional flexible riser analysis. *Engineering structures*, 11(4):223–233, 1989.
- [5] E. Jahanshahi, S. Skogestad, and A. Helgesen. Controllability analysis of severe slugging in well-pipeline-riser systems. In *IFAC Workshop-Automatic Control in Offshore Oil and Gas Production*, pages 101–108, 2012.
- [6] E. Jahanshahi and S. Skogestad. Simplified dynamical models for control of severe slugging in multiphase risers. In *World Congress*, volume 18, pages 1634–1639, 2011.
- [7] E. Jahanshahi and S. Skogestad. Closed-loop model identification and pid/pi tuning for robust anti-slug control. In *10th IFAC International Symposium on Dynamics and Control of Process Systems*, 2013.
- [8] A. Visioli. *Practical PID control*. Springer Science & Business Media, 2006.
- [9] T. Häggglund. Signal filtering in pid control. In *IFAC conference on Advances in PID Control*, 2012.

- 
- [10] Mathworks. Documentation: Simulate continuous- or discrete-time pid controllers, 2015. URL <http://se.mathworks.com/help/simulink/slref/pidcontroller.html>. Visited: April 2015.
- [11] J. Hespanha. Lecture notes on lqr/lqg controller design. *University of California, Santa Barbara*, 2005.
- [12] E. Lavretsky and K. Wise. Adaptive output feedback design using asymptotic properties of lqg/ltr controllers. *Automatic Control, IEEE Transactions on*, 57(6):1587–1591, 2012.
- [13] A. N. Nasir, M. A. Ahmad, and M. F. Rahmat. Performance comparison between lqr and pid controllers for an inverted pendulum system. In *INTERNATIONAL CONFERENCE ON POWER CONTROL AND OPTIMIZATION: Innovation in Power Control for Optimal Industry*, volume 1052, pages 124–128. AIP Publishing, 2008.
- [14] A. N. Nasir, M. Ahmad, and N. Hambali. Performance comparison between lqr and pid controller for a speed and path trajectory control of a hovercraft system. *International Conference on 2nd International Conference on Underwater System Technology: Theory and Applications*, 2008.
- [15] E. Lavretsky and K. Wise. *Robust and Adaptive Control: With Aerospace Applications*. Springer Science & Business Media, 2012.
- [16] R. M. Murray, Z. Li, S. S. Sastry, and S. Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.
- [17] G. J. Silva, A. Datta, and S. P. Bhattacharyya. *PID controllers for time-delay systems*. Springer Science & Business Media, 2007.
- [18] J.F. Kaiser and W.A. Reed. Data smoothing using low-pass digital filters. *Review of Scientific Instruments*, 48(11):1447–1457, 1977.
- [19] E. Jahanshahi. *Control Solutions for Multiphase Flow*. PhD thesis, Norwegian Universit of Science and Technology, 2013.
- [20] V. De Oliveira, J. Jaschke, and S. Skogestad. An autonomous approach for driving systems towards their limit: an intelligent adaptive anti-slug control system for production maximization. In *IFAC Workshop on Automatic Control in Offshore Oil and Gas Production, Brazil*, 2015.
- [21] S. Skogestad and I. Postlethwaite. *Multivariable feedback control: Analysis and Design*, volume 2. Wiley New York, 2007.

- [22] D. Seborg, T. F. Edgar, and D. Mellichamp. *Process dynamics & control*. John Wiley & Sons, 2006.
- [23] E. Jahanshahi, V. De Oliveira, C. Grimholt, and S. Skogestad. A comparison between internal model control, optimal pidf and robust controllers for unstable flow in risers. In *The International Federation of Automatic Control, Cape Town, South Africa*, 2014.
- [24] W. Storr. Butterworth filter design and low pass butterworth filters, 2013. URL [http://www.electronics-tutorials.ws/filter/filter\\_8.html](http://www.electronics-tutorials.ws/filter/filter_8.html). Visited: March 2015.
- [25] R.E. Challis and R.I. Kitney. The design of digital filters for biomedical signal processing part 3: The design of butterworth and chebychev filters. *Journal of biomedical engineering*, 5(2):91–102, 1983.
- [26] Prof. S. Roberts. *Design of Digital Filters*. university of oxford, 2003. URL <http://www.robots.ox.ac.uk/~sjrob/>. Visited: March 2015.
- [27] Mathworks. Documentation: Butterworth filter design, 2015. URL <http://se.mathworks.com/help/signal/ref/butter.html>. Visited: March 2015.
- [28] J. C. Willems. Least squares stationary optimal control and the algebraic riccati equation. *Automatic Control, IEEE Transactions on*, 16(6):621–634, 1971.
- [29] P. Misra. Numerical algorithms for squaring-up non-square systems, part ii: General case. *Proceedings of American Control Conference, San Francisco, CA*, 1998.
- [30] J-B. Pomet and L. Praly. Adaptive nonlinear regulation: estimation from the lyapunov equation. *Automatic Control, IEEE Transactions on*, 37(6):729–740, 1992.
- [31] J-JE. Slotine and J.A. Coetsee. Adaptive sliding controller synthesis for nonlinear systems. *International Journal of Control*, 43(6):1631–1651, 1986.
- [32] R. A. Hilhorst, J. Van Amerongen, P. Löhnberg, and H. JAF. Tulleken. A supervisor for control of mode-switch processes. *Automatica*, 30(8):1319–1331, 1994.
- [33] A. Savitzky and M. JE. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

- [34] IMY. Mareels and M. Gevers. Persistency of excitation criteria for linear, multivariable, time-varying systems. *Mathematics of Control, Signals and Systems*, 1(3):203–226, 1988.

## Appendix A

---

# Additional results for self-tuning system

---

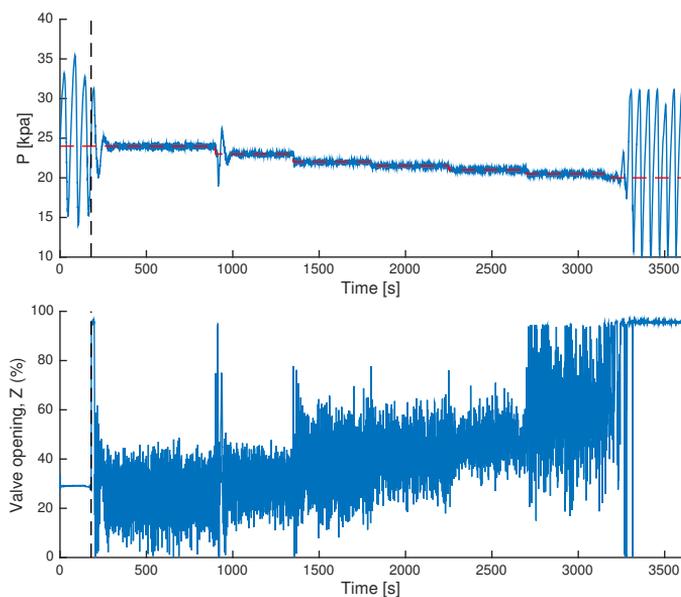


Figure A.1: Adaptive gain with optimal PIDF in standard form (Equation 1.2a) controller linearized around  $Z^* = 30\%$ . Black dashed line marks the controller activation. In upper figure: subsea pressure with set point pressure (red dashed line) from Table 4.1. In lower figure: Valve opening,  $0 \leq Z \leq 100$ , with respect to set point changes. Oscillatory slugging at set point,  $r = 20.0\text{kPa}$ . The updated gain is presented in Figure A.4

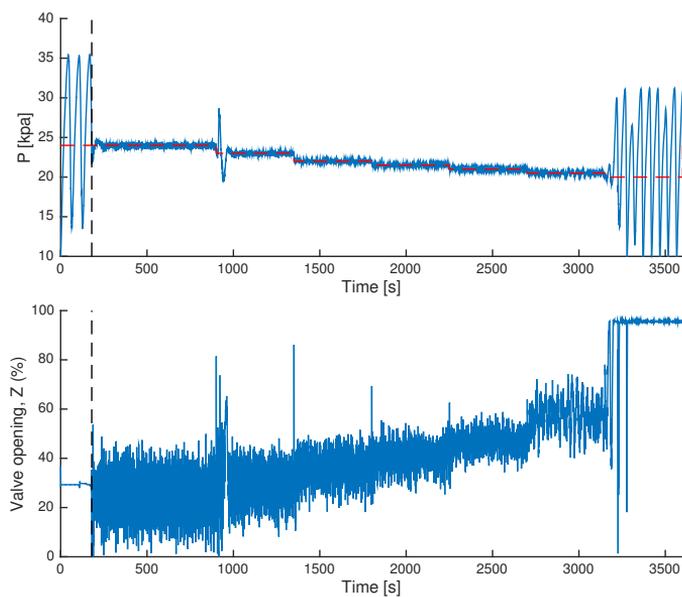


Figure A.2: Adaptive gain with optimal PIDF in parallel form (Equation 1.2b) controller linearized around  $Z^* = 30\%$ . Black dashed line marks the controller activation. In upper figure: subsea pressure with set point pressure (red dashed line) from Table 4.1. In lower figure: Valve opening,  $0 \leq Z \leq 100$ , with respect to set point changes. Oscillatory slugging at set point,  $r = 20.0\text{kPa}$ . The updated gain is presented in Figure A.5

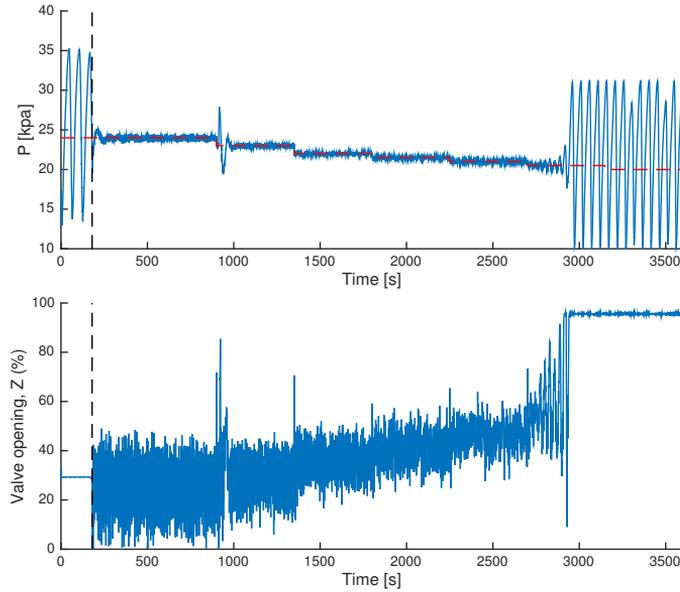


Figure A.3: Adaptive gain and  $K_i$  with optimal PIDF in parallel form (Equation 1.2b) controller linearized around  $Z^* = 30\%$ . Black dashed line marks the controller activation. In upper figure: subsea pressure with set point pressure (red dashed line) from Table 4.1. In lower figure: Valve opening,  $0 \leq Z \leq 100$ , with respect to set point changes. Oscillatory slugging at set point,  $r = 20.0\text{kPa}$ . The updated gain and  $K_i$  is presented in Figure A.6

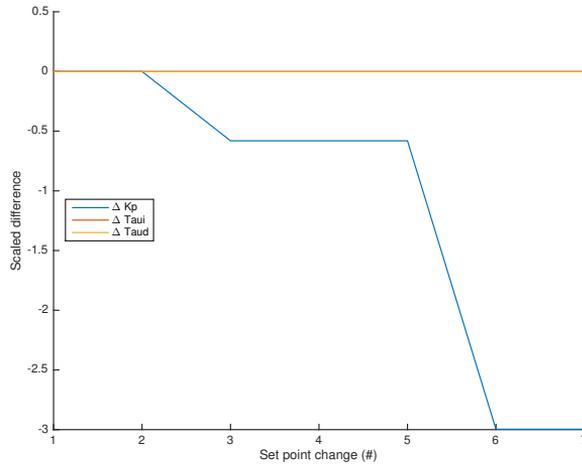


Figure A.4: Change of controller parameters,  $K_p$ ,  $\tau_i$  and  $\tau_d$  during adaptation with tuning parameter  $\gamma = 1$  and  $\tau_{des} = 3$ . The controller was updated in standard form. Table 4.1 presents the actual set points with respect to actual set point change number. Note that the parameters are scaled, that is  $\Delta K_p = \frac{K_{p,0} - K_{p,\#}}{K_{p,0}}$ , where  $K_{p,0}$  is the baseline controller gain.

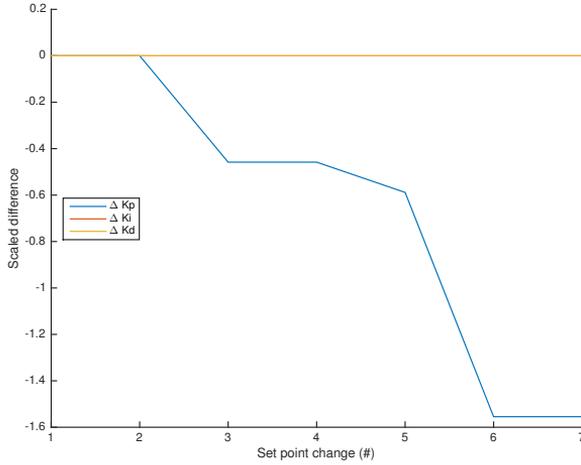


Figure A.5: Change of controller parameters,  $K_p, K_i$  and  $K_d$  during adaptation with tuning parameter  $\gamma = 1$  and  $\tau_{des} = 3$ . The controller was updated in parallel form. Table 4.1 presents the actual set points with respect to actual set point change number. Note that the parameters are scaled, that is  $\Delta K_p = \frac{K_{p,0} - K_{p,\#}}{K_{p,0}}$ , where  $K_{p,0}$  is the baseline controller gain.

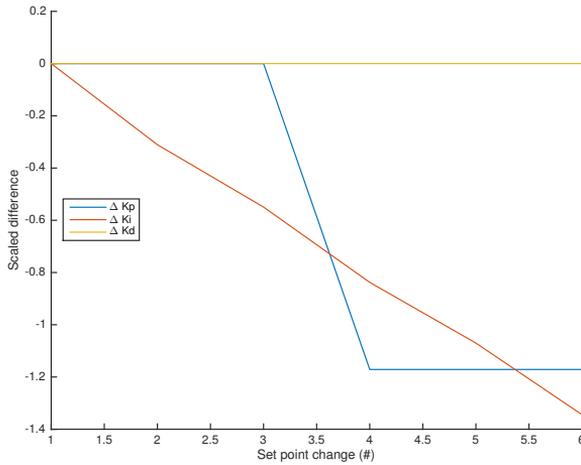


Figure A.6: Change of controller parameters,  $K_p, K_i$  and  $K_d$  during adaptation with tuning parameter  $\gamma = 1$  and  $\tau_{des} = 3$ . The controller was updated in parallel form. Table 4.1 presents the actual set points with respect to actual set point change number. Note that the parameters are scaled, that is  $\Delta K_p = \frac{K_{p,0} - K_{p,\#}}{K_{p,0}}$ , where  $K_{p,0}$  is the baseline controller gain.

## Appendix B

---

# Matlab scripts

---

### B.1 Simplified four state model

This model can be found online from previous research published by Jahanshahi<sup>19</sup>.

### B.2 Tune PID with use of set point response

*Listing B.1: tunePID.m*

```
1 function [ controller, log ] = getTuningParam(data2, SYS, stepinfo, ...
    tk, log)
2 % Calculate new PIDF controller based on step response
3 % Input: Measurement data, Previous controller, Step data, run ...
    time, logging
4 % Output: New controller, and log data
5 %
6 % Author: Adrian Finvold
7 %
8
9
10 Ts = data2.Ts; %Sampling time
11 ystart = stepinfo.ystart;
12 stepindx = stepinfo.index;
13 z_crit = 30;
14 piddata = pidstd(SYS);
15 piddata_par = pid(SYS);
16
17 Kp_prev = piddata.Kp;
18 TauI_prev = piddata.Ti;
```

```
19
20 Ki_prev = piddata_par.Ki;
21 Kd_prev = piddata_par.Kd;
22 Tf_prev = piddata_par.Tf;
23
24
25 gamma = 1; %Tuning factor
26 tau_des = 3; % Desired closed loop constant
27 t_back = 2; % seconds back after step
28 t_forw = 8; % seconds forward after step
29
30 %% First agenda: Find new controller gain
31 y = data2.yf(1,ystart:tk*10);
32 % get the recommended order and cutoff for high pass
33
34 [b,a] = butter(2,0.02);
35 y_f2 = filter(b,a,y);
36
37 % Crop signal to desired step area
38 y_fitted = y_f2(stepindx-ystart-(t_back*10):(stepindx-ystart+(t_forw...
    *10)));
39 t_new = data2.t(stepindx-ystart-(t_back*10):(stepindx-ystart+(t_forw...
    *10)));
40
41 % Calculate step response
42 du=stepinfo.size; % Stepsize
43 %gain=(y_fitted(end)-y_fitted(1))/du;
44 dy=diff(y_fitted);
45 dt=diff(t_new);
46 [maxdy,I]=max(abs(dy)./dt);
47 time_constant=abs(y_fitted(end)-y_fitted(1))/maxdy;
48 %time_delay=t_new(I)-abs(y_fitted(I)-y_fitted(1))/maxdy;
49
50
51 % Calculate Tuning parameters
52 tau_diff = abs(tau_des-time_constant);
53
54 if (tau_diff ≤ 1) % If time constant is lower than desired, do not ...
    change gain
55     alpha = 1;
56 else alpha = gamma *(time_constant-tau_des);
57 end
58
59 tempgain = alpha * Kp_prev; % Temporary gain
60 if tempgain > Kp_prev % If new calculated gain is lower than old ...
    gain
61     Kp_new = Kp_prev; % Use same gain as previous
62 else
```

```
63     Kp_new = alpha * Kp_prev;
64 end
65
66
67 ydata.ystart = stepinfo.ystart;
68 ydata.tk = tk;
69 ydata.y = y;
70 ydata.yfiltered = y_f2;
71 ydata.ycropped = y_fitted;
72 ydata.tcropped = t_new;
73 ydata.order = n;
74 ydata.cutoff = Wc;
75 ydata.dy = dy;
76 ydata.tau_des = tau_des;
77 ydata.tau_diff = tau_diff;
78 ydata.timeconst = time_constant;
79 ydata.k_0 = stepinfo.Kp_0;
80 ydata.prevgain = Kp_prev;
81 ydata.tempgain = tempgain;
82 ydata.newgain = Kp_new;
83
84
85 %% Second agenda: Find new integral time based on
86 % Tau_i = Tau_i,old * mean(Z)/Z,old
87
88 % Skip first setpoint change
89
90 z = data2.yf(8,ystart:tk*10);
91
92 % Crop signal to desired step area
93 steps = find(diff(data2.r));
94 stepinfo.index;
95 thisstep = find(steps==stepinfo.index);
96
97 if stepinfo.index == steps(1)
98     z_cropped = z(1:stepindx-ystart);
99     t_cropped = data2.t(1:stepindx-ystart);
100 else
101     z_cropped = z(steps(thisstep-1)-ystart:steps(thisstep)-ystart);
102     t_cropped = data2.t(steps(thisstep-1)-ystart:steps(thisstep)-ystart)...
103         ;
104 end
105 [b,a] = butter(1,0.01);
106 zfiltered = filter(b,a,z_cropped);
107
108 % Get mean valve opening
109
```

```

110 zmean = mean(zfiltered(200:end)); % Skip 200steps first based on ...
      filterslope at startup
111
112 % Get old integral time
113
114 tau_i_0 = stepinfo.tau_i_0;
115 Ki_0 = stepinfo.Ki_0;
116 tau_i_new = tau_i_0 * zmean/z_crit; % Tau_i = Tau_i,old * mean(Z)/Z,...
      old
117 Ki_new = Ki_0 * zmean/z_crit;
118
119 if tau_i_new < Tau_i_prev
120     tau_i_new = Tau_i_prev;
121 end
122
123 zdata.z = z;
124 zdata.zcropped = z_cropped;
125 zdata.tcropped = t_cropped;
126 zdata.zfiltered = zfiltered;
127 zdata.mean = zmean;
128 zdata.tau_i_0 = tau_i_0;
129 zdata.tau_i_prev = Tau_i_prev;
130 zdata.tau_inew = tau_i_new;
131
132 %% Export new PID-controller
133 pid_new = pidstd(Kp_new,piddata.Ti,piddata.Td,piddata.N) % Serial ...
      form ny Kp
134 %pid_new = pidstd(Kp_new,tau_i_new,piddata.Td,piddata.N) % Serial ...
      form ny tau_i
135 %pid_new = pid(Kp_new, Ki_prev, Kd_prev, Tf_prev) % Paralell form ...
      fixed tau_i
136 %pid_new = pid(Kp_new, Ki_new, Kd_prev, Tf_prev) % Paralell form ...
      ny Ki
137 %pid_new = pid(Kp_new, (piddata_par.Ki/tau_i_new), Kd_prev, Tf_prev) ...
      %Feilen: Ki = (piddata.Ki/tau_i_new) = Kp,old / tau_i^2
138 %pid_new = pidstd(piddata.Kp,piddata.Ti,piddata.Td,piddata.N) % ...
      Fixed controller
139 controller = ss(pid_new);
140
141 log.Kp(end+1) = pidstd(controller).Kp;
142 log.Tau_i(end+1) = pidstd(controller).Ti;
143 log.Tau_d(end+1) = pidstd(controller).Td;
144 log.N(end+1) = pidstd(controller).N;
145
146 log.Ki(end+1) = pid(controller).Ki;
147 log.Kd(end+1) = pid(controller).Kd;
148 log.Tf(end+1) = pid(controller).Tf;
149

```

```
150
151 %% Finish up by print and save
152
153 % Save tuning info
154 filename = strcat('partialresult-', num2str(tk), '.mat');
155 save(filename, 'stepinfo', 'controller', 'SYS', 'data2', 'zdata', 'ydata', ...
        'log');
156
157 end
```

### B.3 Tune PID with use of LQR-LQG/LTR

*Listing B.2: DesignAdaptiveControl.m*

```
1 % Calculate the LQR - LQG/LTR controller based on system state space
2 %
3 % Author: Vinicius de Oliveira & Adrian Finvold
4 %
5
6 clc
7 clear all
8 close all
9
10 s=tf('s');
11
12 aux=open('state-space.mat'); %30% opening
13 sys_ss=aux.sys;
14
15 %Open loop plant dynamics
16 Ap=sys_ss.a;
17 Bp=sys_ss.b(:,1);
18 Cm=sys_ss.c(1,:);
19 Cr=sys_ss.c(1,:);
20 Dp=sys_ss.d(1:1);
21 I=eye(size(Ap));
22 % G=minreal(sys_ss.c(1,:)*inv(s*I-sys_ss.a)*sys_ss.b);
23
24
25 nx=length(Ap);
26 ny=size(Cm,1);
27 nu=size(Bp,2);
28
29 %augmenting the system to get integral action
30 Aaug=[zeros(1,1) -Cr; zeros(nx,1) Ap];
31 Baug=[Dp;Bp];
```

```

32 Bcmd=[1 zeros(1,nx)]';
33 %Assuming that P_bottom and P_top are measured or estimated ...
    elsewhere
34 Caug=[1 zeros(1,nx); zeros(ny,1) Cm];
35 Daug=[Dp;0];
36
37
38
39 %% step 1: design a LQR controller to obtain desired robustness ...
    properties
40
41     %Tuning constant for the reference model: higher is faster
42     q_i=10;
43
44     Q=diag([1 0 0 0 0]);
45     Q(1)=q_i*Q(1);
46     R=1*eye(1,1);
47
48     Klqr=lqr(Aaug,Baug,Q,R); % Quadratic cost controller: Feedback ...
        gain
49     I=eye(size(Aaug));
50
51     Llqr=minreal(Klqr*inv(s*I-Aaug)*Baug); % G_ref: Reference model
52
53     %reference system
54     Aref=Aaug-Baug*Klqr;
55     Bref=[1 zeros(1,nx)]';
56     Cref=Caug; % ?
57
58     Ccl=[-Klqr; %ym
59         -Cref; %ucontrol
60         -Klqr*Aref %udot
61         ];
62
63     Mcl=ss(Aref,Bref,Ccl,0);
64     [Gm,Pm,Wgm,Wpm] = margin(Llqr);
65     auxmargin=allmargin(Llqr);
66
67     [yall,t]=step(Mcl);
68     y=yall(:,1);
69     u=yall(:,end-1);
70     udot=yall(:,end);
71
72
73     figure(1)
74     plot(t,y);
75     ylabel('dP, Kpa')
76     xlabel('Time, s')

```

```

77
78     figure(2)
79     plot(t,u);
80     ylabel('Input, %')
81     xlabel('Time, s')
82
83     %%
84
85
86     %reference system
87     Aref=Aaug-Baug*Klqr;
88     Bref=[zeros(1,nx) 1]';
89     Cref=Caug;
90
91     %reference model
92     Mcl=ss(Aref,Bref,Cref,0);
93
94
95
96     %% observer design -squaring up stage
97     % Bsqr=10+20*rand(5,2);
98     Bsqr=5e-4*[21.9498  17.1921
99                16.7062  21.1664
100               15.9845  24.8509
101               19.0519  18.4867
102               18.4529  18.5871
103 ];
104
105     sys_aug=ss(Aaug,Bsqr,Caug,0);
106     display('-----zeros original coordinates-----')
107     zerverc = tzero(minreal(sys_aug))
108     if zerverc>0
109         error('transmission zeros must be stable')
110     end
111
112     %%Compute Kalman gain
113     Q0=Q(1:end,1:end);
114     R0=2*eye(size(Caug,1));
115
116     % vvec=logspace(-2,1.5,10);
117     v=100; % Tuning parameter LQG/LTR
118
119     Qv=Q0+(v+1)/v*Bsqr*(Bsqr');
120
121     Rv=v/(v+1)*R0;
122
123     %====using ricatti equation====
124     [Lv,Pv] = lqe(Aaug,eye(size(Aaug)),Caug,Qv,Rv);

```

```
125
126 % LTR Controller
127 Ac = Aref-Lv*Caug;
128 Bc1 = Lv;
129 Bc2=Bref;
130 Cc = -Klqr;
131 Dc1 = [0. 0. ];
132 Dc2 = 0.;
133
134 %SS model of loop gain at the plant input
135 Ain = [ Aaug 0.*Baug*Cc; Bc1*Caug Ac];
136 Bin = [ Baug; Bc1*Daug];
137 Cin = -[ Dc1*Caug Cc];%change sign for loop gain
138 Din = -[ Dc1*Daug];
139
140 Lltr = minreal(ss(Ain,Bin,Cin,Din));
141 wrange=logspace(-5,2,200);
142
143 % write closed loop observer LTI model
144 Aob=Aref-Lv*Caug;
145 Bob=[Bref Lv];
146 %outputs estimation error: ey=Cxest-ym
147 Cob=[Caug];
148 Dob=[zeros(ny+1,1) -eye(ny+1)];
149 sysobserver=balred(ss(Aob,Bob,Cob,Dob),5);
150 sysobserver.InputName={'ref','P1','Integral error'};
151 sysobserver.OutputName={'e1','e2'};
152
153 syscontrol = ss(Aob, Bob, -Klqr, 0);
154
155 auxmarginLTR=allmargin(Lltr);
156 [yallLTR,tLTR,xLTR]=step(sysobserver);
157 yLTR=yallLTR(:,1);
158 uLTR=yallLTR(:,end-1);
159 udot=yallLTR(:,end);
160
161 figure
162 bode(Llqr,Lltr,wrange)
163 drawnow
164 grid on
165 hold on
166 legend('LQR','LTR')
167
168 %Closed loop reference model eigenvalues
169 eig(Aob)
170
171
172 %constants for the adaptive law
```

```

173 S=[1 0];
174 auxM=Baug'*Caug'*R0^(-0.5);
175 [U,Sig,V]=svd(auxM);
176 W=(U*V)';
177 F=R0^(-0.5)*W*S';
178
179
180 %save('adaptive_system4States', 'Klqr', 'sysobserver','F','...
      syscontrol','Aref','Bref','Cref');
181 %% Simulate controller against Esmaeils NonLinear model
182 AdaptOn=1;
183 callNonLinearSim

```

## B.4 Projection Operator

*Listing B.3: Projector.m*

```

1  % Projector Operator x_dot=Proj(x,y). Ensures that x remain inside ...
      some
2  % bounds in a smooth way.
3  %
4  %   x: n-by-m matrix of parameters
5  %   y: n-by-m matrix of time derivatives of the parameters
6  %   xmax: bound for parameter x.
7  %
8  % Author: Vinicius de Oliveira
9  %
10 function projected=Proj(x,y,xmax)
11
12
13 [n,m]=size(x);
14 if n~=size(y,1) || m~=size(y,2)
15     error('par and par_deriv must have the same dimensions')
16 end
17
18
19 %convex function that define the convex sets for the projection ...
      operator
20 %domain: f(x)≤0 -> omega_0; f(x)≤1 -> omega_1
21 f=@(theta,epsp,thetamax) ( (1+epsp)*norm(theta)^2-thetamax^2)/(epsp*...
      thetamax^2));
22
23 %grandient of f wrt par
24 fgrad=@(theta,epsp,thetamax) ( 2*(1+epsp)/(epsp*thetamax)*theta);
25

```

```

26 Gamma=eye(n);
27
28 %projection tolerance
29 epsproj=0.1;
30
31 %do the projection columnwise
32 for i=1:m
33
34     fgradj=fgrad(x(:,i),epsproj,xmax);
35     fj=f(x(:,i),epsproj,xmax);
36
37     if fj>0 && y(:,i)'*fgradj>0
38         projected(:,i)=y(:,i)-Gamma*fgradj*(fgradj)'/(fgradj)'*...
39             Gamma*fgradj)*y(:,i)*fj;
40     else
41         projected(:,i)=y(:,i);
42     end
43 end

```

## B.5 Check for instability with cross-correlation

*Listing B.4: check\_instability.m*

```

1  %% This function checks in real-time if the system is getting ...
   unstable.
2  % Uses cross-correlation function and FFT
3  %
4  % INPUTS:
5  % OUTPUT: flag=1 -> instability detected; flag=0 -> normal operation
6  % flag=0.5 -> unsure, wait longer, dont change setpoints
7  % flag=2 -> too high gain
8  %
9  % Author: Vinicius de Oliveira
10 %
11 function [flag] = check_instability_v2(y,ysp,Ts,k)
12
13 % instability checked every dT [s]
14 dT=10;
15 kcheck=round(dT/Ts);
16
17 %size of the window [s]
18 h=1.0*60;
19 kd = round(h/Ts);
20

```

```
21 flag=0;
22
23 powerhist=[];
24 %checks every kcheck samples
25 if(mod(k,kcheck)==0)
26 %     if true
27
28     for j=1:3
29
30         if(k<=kd*j)
31             datawindow = y(:,1:k);
32             errorvec=ysp(:,1:k)-datawindow;
33
34         else
35             datawindow = y(:,k-(j-1)*kcheck-kd:k-(j-1)*kcheck);
36             errorvec=ysp(:,k-(j-1)*kcheck-kd:k-(j-1)*kcheck)-...
                 datawindow;
37         end
38
39         %removes DC componente so that we don't mistakenly detect a ...
40         % 0 freq peak
41         datawindow=datawindow-mean(datawindow);
42
43         [testcross,lags]=xcorr(datawindow);
44         [pks,locs] = findpeaks(testcross,'SORTSTR','descend');
45
46         if length(pks)<2
47             pks=pks(1);
48             locs=locs(1);
49             distancepeaks=abs(lags(locs(1)));
50         else
51             pks=pks(1:2);
52             locs=locs(1:2);
53             distancepeaks=abs(lags(locs(2)));
54         end
55
56         [ powerpeak,freqpeak ] = getPowerPeak( datawindow );
57
58         if j==1
59             %
60             t=linspace(0,dT,length(errorvec));
61             IAE=trapz(t,abs(errorvec));
62             meanerror=abs(mean(errorvec));
63             distancepeaksLast=distancepeaks/100;
64             freqpeakLast=freqpeak;
65             %
66         end
```

```
67
68     powerhist=[powerpeak powerhist];
69
70     end
71
72     %if both numbers are positive, we have two subsequent increase ...
73     in
74     %amplitude
75     increase= diff(powerhist);
76     auxincr=sign(increase);
77     numincrease=sum(auxincr(auxincr>0)) ;
78
79     %second order derivative
80     secondincrease=sign(diff(increase));
81
82     % meanerror
83     % IAE
84     % distancepeaksLast
85     % numincrease
86     % secondincrease
87
88     %checks if the distance between peaks of the cross-correlation ...
89     is
90     %inside some bound -> this distance is the period of oscillation
91     % major problem/disturbance
92     % if IAE>=30
93     % flag=100;
94
95     %if frequency of peak is too high, gain is too big for given
96     %conditions
97     if freqpeakLast>=0.2 && powerhist(end)>=3
98         flag=2;
99
100     elseif (distancepeaksLast>=2.0 && distancepeaksLast<=6)&& ...
101         numincrease==2 && secondincrease>=0 || IAE>=10
102         flag=1;
103
104     % wait longer to see what happens
105     elseif (distancepeaksLast>=1.7 && distancepeaksLast<=6) || (...
106         meanerror>=0.05)
107         flag=0.5;
108
109     %stable
110     else
111         flag=0;
112     end
```

```
111
112
113
114 else
115     %not checked
116     flag=-1;
117     %     majorproblem=-1;
118
119 end
```

## B.6 Check for instability using Integrated Absolute Error

*Listing B.5: double\_check\_instability.m*

```
1  %% This function checks in real-time if it is safe to change ...
   setpoint
2  % Uses ISE
3  %
4  % INPUTS:
5  % OUTPUT: flag=100 -> problem detected; flag=0 -> normal operation
6  %
7  % Author: Vinicius de Oliveira
8  %
9
10 function [flag] = double_check_instability(y, sp, Ts, k)
11
12
13 %size of the window [s]
14 h=30;
15 kd = round(h/Ts);
16 %checks every dT seconds
17 dT=5;
18 kcheck=round(dT/Ts);
19
20 if(mod(k,kcheck)==0)
21 % if true
22
23     if(k<=kd)
24         errorwindow = y(:,1:k)-sp(:,1:k);
25     else
26 %         size(y(:,k-kd:k))
27 %         size(sp(:,k-kd:k))
28         errorwindow = y(:,k-kd:k)-sp(:,k-kd:k);
29     end
30     t=linspace(0,h,length(errorwindow));
```

```

31
32     IAE=trapz(t,abs(errorwindow))
33     % meanerr=abs(mean(errorwindow));
34
35     if IAE>=40
36         flag=100;
37     else
38         flag=0;
39     end
40
41 else
42     flag=-1;
43
44 end
45
46
47
48
49 end

```

## B.7 Estimate frequency of peak

*Listing B.6: get\_Power\_peak.m*

```

1  % Get frequency of peaks
2  % Input: vector
3  % Output: Peak magnitude and frequency
4  %
5  % Author: Vinicius de Oliveira
6  %
7  function [ powerpeak,freqpeak ] = getPowerPeak( yvec )
8  L=length(yvec);
9  Ts=0.1;
10 Fs=1/Ts;
11 NFFT = 2^nextpow2(L);
12 Y = fft(yvec,NFFT)/L;
13 Y=Y(1:NFFT/2+1);
14 psdx=100*((1/(Fs)).*abs(Y));
15 psdx(2:end-1) = 2*psdx(2:end-1);
16 freq=Fs/2*linspace(0,1,NFFT/2+1);
17
18
19 [powerpeak,indexmax]=max(psdx);
20 freqpeak=freq(indexmax);
21

```

22 `end`

---