NTNU – Trondheim
Norwegian University of
Science and Technology

Disturbance Attenuation in Managed Pressure Drilling

Anders Albert

Master of Science in Engineering Cybernetics
Submission date: June 2013
Supervisor: Ole Morten Aamo, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

DISTURBANCE ATTENUATION IN MANAGED PRESSURE DRILLING

ANDERS ALBERT

NTNU
Norwegian University of
Science and Technology

Master Thesis (TTK4900)

June 2013

Anders Albert: *Disturbance Attenuation in Managed Pressure Drilling,* Master Thesis (TTK4900), © June 2013

SUPERVISOR:
Ole Morten Aamo

ABSTRACT

In reservoirs where multiple wells have been drilled, the window that the pressure has to stay within becomes smaller. This calls for tight control of the pressure profile in the well. An extra challenge is provided when drilling from a floating rig: The Heave motion. During normal operation the heave motion is compensated for. However, when the drillstring needs to be extended it is connected to the rig, which causes it to move with the rig. This creates huge pressure fluctuations in the well. Multiple papers have been written on the subject of suppressing these fluctuations. An experimental lab has been built at NTNU to model this scenario, called the IPT-Heave Lab. The goal of the lab is to have a realistic test environment for the developed controllers and be able to test new control algorithms.

This thesis has concerned itself with making the IPT-Heave Lab operational. The scenario the lab is built for is the use of a technique called constant bottomhole pressure to suppress the heave motion. With this technique a choke is used to control the pressure at the top of the well. In this thesis a controller for the choke is developed, and its performance has been demonstrated in the lab. Furthermore motivated by literature study, a model of the lab has been identified based on a black box modeling principle. The identified model has then been used to design an MPC-controller. The MPC-controller has been demonstrated to suppress disturbances, simplified to a single sine wave, of a period of 3 seconds by approximately 46 %.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Professor Ole-Morten Aamo for giving valuable input and guidance throughout the whole semester. In addition I would like to thank the people involved in the project, who have helped and provided information Jarle Glad and Åge Sivertsen.

Thanks to my fellow students who have been involved with the IPT-Heave lab: Jussi Mikael Aanestad, Martin Standal Gleditsch, Robert Drønnen, Andreas Laupstad Boge and Anish Phade.

Also thanks to Johannes Møgster for providing some valuable input and the implementation code of the Overschee algorithm.

Finally thanks to my friends Line Nordsveen and Morten Olsen for taking the time to read my report and give input on my writing.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

MPD	Managed Pressure Drilling
CBHP	Constant Bottomhole Pressure
PMCD	Pressureized Mud Cap Drilling
DG	Dual Gradient
RFC	Returns Flow Control (also called HSE)
BOP	Blowout Preventer
PID	Proportional-Integral-Derivative (controller)
IPT	Department of Petroleum Engineering & Geophysics (at NTNU)
BHA	Bottomhole Assembly
MPC	Model Predictive Controller

1 INTRODUCTION

This is the introductory chapter for this master thesis. It contains the general motivation for the thesis along with a description of what is desired achieved. A scope for the rest of the thesis will also be given as well as the task description for this master thesis.

1.1 MOTIVATION

Most easy prospects have already been drilled. In many reservoirs only challenging drilling operations remain. What makes the remaining drilling operations challenging is the narrow window the pressure has to remain within. The pressure has to remain within these environmental limits since exceeding them can be catastrophic for the operation (see Chapter 2 for more information). Drilling from a floating rig provides an extra challenge. Namely a floating rig moves with the waves. During normal operations this movement is compensated for. However, when the *drillstring* needs to be extended, an operation called *connection*, the drillstring will be connected to the rig. When the drillstring is connected to the rig it moves with the rig, which causes large pressure fluctuations in the well. Managed pressure drilling (MPD) is a collective term for techniques trying to control the pressure profile of a well. The goal is to use MPD to compensate for the pressure fluctuations experienced during connection, which will make drilling otherwise undrillable prospects possible to drill. In addition MPD can increase the safety and the efficiency of a drilling operation.

1.2 SCOPE AND EMPHASIS

Multiple thesis and articles have been written on the uses of an MPD technique called constant bottomhole pressure (CBHP) to suppress pressure fluctuations caused by heave motions [15], [18], [16], [17], [1] and [2]. All these papers are only theoretical and the algorithms have only been tested in simulators. A lab, called the IPT-Heave Lab, has been built at NTNU to model the connection scenario with CBHP. Much of the work of this master thesis has been to make the IPT-Heave lab operational. With CBHP the pressure at the top of the well is controlled by a choke. During the fall of 2012 a project was done on using this choke to control the pressure at the top of the well. Due to problems with the equipment, that work was not finished. This master thesis is a continuation of that work. Finishing that work has been the first goal of this master thesis. The second goal of this thesis has been to design a controller for the bottomhole pressure in the well and demonstrate its performance in the lab.

1.3 OUTLINE OF THESIS

Following this introductory chapter is a chapter on drilling. This is a background chapter meant to make the unfamiliar reader familiar with the basic terms and concepts of drilling. Chapter 3 is an in depth documentation of the IPT-Heave lab, often referred to as the lab from here on out. The retesting and design of a controller for the choke is described in Chapter 4. A small literature study has been performed on controllers developed for the connection scenario for a floating rig in Chapter 5. This literature study motivated a black box approach for modeling the lab. A linear discrete model was found, with the procedure detailed in Chapter 6. This linear model was used as a basis for the controller developed in Chapter 7. A discussion wraps up the thesis in Chapter 8, before a conclusion is drawn in Chapter 9. Appendix A contains a startup guide for the IPT-Heave lab. This is a really easy introduction with procedures and safety issues of the lab. Software by the company Lenze named Engineer is used to control the motors controlling the disturbance and the choke. Some documentation of this software is given in Appendix B. Following this master thesis is a CD containing the MATLAB files used. Appendix C contains a description of the files on that CD. Finally an article will be attempted to be submitted on the lab, the article is under preparation and a draft is included in Appendix D.

1.4 PROJECT TASK DESCRIPTION

This thesis is a master thesis, which had the following tasks formulations:

1. Verify the newly remodelled choke in the lab. Has backlash, as identified in the project work, been removed?
2. Test reference tracking controllers for pressure developed during the project work in the lab. Develop them further if necessary. Demonstrate their performance in the lab.
3. Review relevant literature on the disturbance rejection problem for heave, in particular the work of Ingar Landet, Hessam Mahdianfar, Aarsnes, Aamo.
4. Develop a control algorithm for the rejection of disturbances (may be based on the references mentioned, or you may derive your own controllers).

5. Demonstrate the performance of your control design in the lab.

6. Write a report.

2 BACKGROUND ON DRILLING

This chapter is taken from Albert [3]. It is repeated here in order to give an introduction to drilling and MPD drilling for the unfamiliar reader.

The first section of this chapter gives an introduction to conventional drilling. It is mainly based on the book "Drilling Technology in Nontechnical Language" by Devereux [4]. In the second section managed pressure drilling (MPD) is presented along with some different MPD techniques and experiences.

2.1 CONVENTIONAL DRILLING

2.1.1 *Reservoir Formation*

Oil and gas (collectively called *hydrocarbons*) are produced from animal and plant remains. In areas like swamps, lakes, coastal regions and shallow seas, animal and plant remains sometimes accumulated millions of years ago. Through time these got buried by sediments, and as the remains got deeper the pressure and temperature would increase. Inside small spaces within the rock, called *pores*, remains would under certain temperature and pressure conditions undergo a chemical transformation into hydrocarbons.

Permeability is a measure of the ability of fluids to flow through rock. For this to be possible the pores must be connected. After a long time the hydrocarbons get squeezed out of their pore by pressure and starts migrating towards the surface as long as the rock is permeable. When the structure of the path of the migration flow for the hydrocarbons has the right conditions, hydrocarbons will be stopped and start accumulating. This is how an oil and gas reservoir is formed.

The reservoir has a complex structure. The hydrocarbons will be contained in pores of the rock matrix of the reservoir. In addition to water there may also be other materials such as clay within the pore spaces. This can cause difficulties when drilling through the reservoir. A reservoir is rarely uniform and the permeability of the reservoir will depend on direction. This will sometimes make it necessary to drill a horizontal well into it instead of a vertical. The whole reservoir will have oil unevenly distributed, and can be thought of as a sponge.

2.1.2 *Discovering and Planning a Well*

A *prospect* is an area where a company believes it to be hydrocarbons present in economical quantities. Geologists will use different techniques to estimate whether or not it is likely to find hydrocarbons in an area. The company will then decide whether to take the economical risk of drilling an exploration well. An *exploration well* is a well drilled to gather information. If hydrocarbons are found, the planning of a production well can start.

It takes a careful plan to drill a well. First the different rock formations that are likely to be encountered must be identified. A *rock formation* is a body of rock that is sufficiently distinctive and continuous to be mapped. A desired depth has to be decided before drilling starts, since the initial diameter of the well will depend on it. Two important parameters also have to be estimated before drilling starts: pore and fracture pressure. *Pore pressure* is the pressure within the rock pores at a certain depth. The *fracture pressure* is the pressure that would fracture the hole. These are important environmental constraints that the pressure of the well must stay within.

2.1.3 *Drilling Equipment*

There is a number of different drilling equipment that are important for drilling a well. This subsection will describe the most important.

DRILLING RIG: This is the machine that is used to drill a wellbore. It consists of everything used to drill a well, except living quarters. [27]

DRILL BIT: This is the tool that is used to crush and cut loose rock from the bottom of the well. There is a number of different types of drill bits that have different advantages and disadvantages. What type of drill bit used depends on the current rock formation. For a drill bit to drill rock, downward force and rotation is required.

DRILL COLLARS: A thick-walled pipe providing force for the drill bit.

BOTTOMHOLE ASSEMBLY (BHA): Drill bit together with drill collars, stabilizers and other equipment used at the bottom of the well are called bottomhole assembly.

DRILLSTRING: A well can be several thousand meters long. To get the drill bit to the bottom of the well, tubes are screwed together on top of the bottomhole assembly. This is called a drillstring.

ANNULUS: The space between the drillstring and the well wall is called annulus.

CUTTINGS: These are small pieces of rock that has gotten lose by the action of the drill bit. [27]

MUD: This is a fluid that gets pumped down through the drillstring and up the annulus. The mud has a number of different functions, but one of the most important is to transport cuttings from the bottom of the well to the surface. At the surface the cuttings get separated from the mud before the mud is circulated back down again. Usually mud is water or oil based with some chemicals added to obtain desired properties.

CASING: A large pipe that is lowered into the well and cemented in place. Its task is to maintain the integrity of the well through the drilling operation and later production. When drilling a well multiple casings are placed through the operations. This is the reason why the diameter of the well depends on the depth of the well, since the number of casings depends on the depth of the well.

DERRICK: This is the large mast on top of the drilling rig. It is used to support the drillstring.

2.1.4 *Drilling Operation*

To drill a well on the seabed, a welded steel structure, called a *template*, is placed there. It will be welded to the seabed at each of its corners. Guide wires are used to keep the rig in place over the template. The template also contains a large pipe with conical guide above, which is used to guide the drill bit down into the well.

The start of drilling the well is called *spudding*. A small diameter drill bit is used since one might encounter a gas pocket, and a small diameter is easier to control than a large one. Later a larger diameter hole is drilled.

About every 30 meters drilling operation is stopped and the drillstring is extended. This operation is called *connection*. A new tube is screwed on top of the old tube before the drilling resumes. On a floating rig the drillstring will be connected to the rig while connection is performed. This will cause the drillstring to move in and out of the well. The movement in and out of the well is called *tipping*. Under normal drilling operations a tool called *heave compensator* prevents tipping from happening. The increased pressure that is caused by the drillstring moving into the well is called *surge pressure*, while the opposite is called *swab pressure* (decrease due to drillstring moving out the well).

There are two components making up the pressure at the bottom of the well during a drilling operation. The first component is the *hydrostatic pressure*, which comes from the weight of the drilling mud. The second component is *frictional pressure* that comes from circulating the mud. To change the pressure it is possible to change the speed of the circulating mud. However this velocity is mainly used to control the *rate of penetration* (how fast the well gets drilled), and should therefore not be used to control pressure. In addition during connections the pumps will be turned off, and only the hydrostatic pressure makes up the pressure in the well. This leaves one option to change the pressure in the well and that is to change the density of the drilling mud. Circulating out old mud and in new mud is both time consuming and expensive, making it far from an ideal way to control the pressure in the well.

There are two concerns regarding pressure during a drilling operation. First, the pressure has to stay below the fracture pressure. If this is exceeded the well might collapse and then it will be lost. It is even more important to stay above the pore pressure during drilling. If the pressure in the well drops below this, one might experience influx into the well from the formation, which is called a *kick*. If the kick is not handled properly it can lead to a *blowout*, which is hydrocarbons reaching the surface and potentially ignite. To deal with a kick the well has to be sealed off using a device called a *blowout preventer* (BOP), and a heavier drilling mud is circulated into the well.

The pore and fracture pressures are estimated before drilling starts. They will also be measured through the drilling process by decreasing the pressure until influx is experienced or increase until the formation starts to fracture. The limits can however suddenly change depending on the rock formation that is encountered. Quickly reacting to such changes can make drilling both more efficient and safer.

2.2 MANAGED PRESSURE DRILLING (MPD)

To maximize the production of a reservoir multiple wells have to be drilled into it. First time a reservoir is drilled the margin between pore and fracture pressure is quite wide, which allows huge pressure fluctuations during the drilling. However for each well drilled it gets more challenging since the previous wells leads to the pressure in the reservoir changing. This leads to narrower margins between pore and fracture pressure. Using conventional drilling techniques it may become impossible to drill because of the large pressure fluctuations from heave motion, and a more tight control of the pressure is needed.

2.2.1 *MPD techniques*

A definition of MPD from International Association of Drilling Contractors (IADC) from Hannegan [11]:

MPD is an adaptive drilling process used to more precisely control the annular pressure profile throughout the wellbore. The objectives are to ascertain the downhole pressure environment limits and managed the annular hydraulic pressure profile accordingly.

There is a number of different MPD techniques. The different techniques being used in marine environments are:

A. Constant Bottomhole Pressure (CBHP)

B. Pressureized Mud Cap Drilling (PMCD)

C. Dual Gradient (DG)

D. Returns Flow Control (RFC also called HSE)

The following descriptions are based on the papers [11, 14, 19, 25].

2.2.1.1 *Constant Bottomhole Pressure*

With CBHP the annulus is sealed off at the surface and the mud flow back to the surface is controlled by a choke. This leads to the pressure in the well now consisting of three components: hydrostatic pressure, frictional pressure drop and the pressure at the choke. Closing the choke will increase the pressure at the top of the well and opening it will decrease it. The pressure at the top of the well will after a small delay transmit itself to the bottom of the well. This makes it possible to use the choke opening to control the pressure at the bottom of the well.

In order to get a pressure drop at the choke there must be circulation of mud in the system. During connection the main pump is shut down. To accommodate for this an extra pump, called backpressure pump, ensures circulation through the choke.

When drilling with CBHP a lighter drilling mud will be used compared to what would be used with a conventional technique.

2.2.1.2 *Pressurized Mud Cap Drilling*

To apply PMCD the operation has to encounter severe losses of drilling mud. This is called *circulation loss* and results from mud being lost into the formation. When this situation occurs, the rig can use seawater with some additives as drilling mud and the mud with cuttings is forced into the zone that caused loss in circulation. To maintain the pressure in the well a backpressure is applied. PMCD can also be used even if severe losses is not encountered by provoking it.

2.2.1.3 *Dual Gradient*

DG is a whole class of drilling techniques where there is a second fluid/gas system. One way to do DG is to use two different drilling muds, and let a pump at the seabed lift the heavy mud with cuttings up to the surface in a separate hose. The layer between the light and the heavy mud lies between the rig and the pump. By adjusting this level one can change the hydrostatic pressure in the well thus controlling the pressure profile. Another way of doing DG is to inject gas into the annulus at some point which makes a different density of the mud below and above the injection point.

2.2.1.4 *Returns Flow Control*

This is not an MPD technique but is considered to be an important tool in an MPD operation. Instead of having the top of the well open to the atmosphere, being at a potential risk of getting mud and hydrocarbons on the rig floor, the flow is controlled. This greatly increases safety of the drilling operation and one can avoid the need for closing the blowout preventer.

2.2.2 *Experience with MPD*

MPD techniques have been successfully applied both to stationary and floating rigs [10, 11, 19]. The Asian Pacific, North Sea, Gulf of Mexico and offshore Brazil are all places where MPD has been used. The motivations for using MPD have been increased safety of drilling operations and a reduction in non-productive time along with being able to drill otherwise undrillable prospects. Reducing the non-productive time is hugely economically beneficial, since drilling is very expensive. Still MPD has not yet been applied to floating rigs in harsh environment like the North sea where heave induced pressure fluctuations can make the bottomhole pressure vary as much as ±60 bars [25].

3 IPT-HEAVE LAB

Together with Appendix A and Appendix B this chapter documents the IPT-Heave Lab. The lab is supposed to model a connection scenario for a floating rig. This chapter contains the basis and simplifications made for that design. In addition a in depth explanation of each component of the lab is given in this chapter. Appendix A is a chapter common for Drønnen [5], Gleditsch [9] and this report. It contains a set of procedures used in the lab, for example startup. Appendix B documents software called Engineer used in the lab.

3.1 BASIS FOR LAB

The IPT-Heave Lab is designed to simulate a connection scenario for a floating rig with the MPD technique: Constant bottomhole pressure (CBHP). The basis for the lab is a rig with a drillstring of 5 inches diameter and a 8.5 inches in diameter hole exposed to a heave of amplitude 1.5 meters and 11 seconds period. The length of the the real well is 4000 meters, and is considered to be vertical. The bottomhole assembly used as a basis is 70 meters long and with a diameter of 6.5 inches. The work on designing the lab was initiated the fall of 2011 by Gjengseth and Svenum [8]. The design data is taken from that report.

3.2 LAB MODEL

3.2.1 *Scaling*

Both scaling and lab simplification argumentation are taken from Gjengseth and Svenum [8]. They are rendered here for the completeness of this chapter.

The main challenge of controlling the pressure with CBHP during a connection scenario is the time for the pressure wave to travel from top to the bottom of the well, called delay. When this delay is large compared to the period of the heave disturbance, maintaining a constant bottomhole pressure becomes difficult. This is the main dynamics that was desired to model.

When scaling down the lab it was decided by Gjengseth and Svenum [8] that the length of the well should be 900 meters. The basis for this choice was the propagation speed of the pressure wave and the desired length of the delay from choke to bottomhole. The heave disturbance was assumed to be a single sine wave. It was also decided that the fastest period the system should handle should be about

5 times the time delay. The pressure wave was assumed to travel with the speed of sound in water at 25 degrees, which is 1498 m/s. The fastest sine wave the disturbance should handle should then be $\frac{900m}{1498m/s}x5 \approx 3$ seconds.

Furthermore the drillstring and the well was scaled down accordingly, with the drillstring being 1 inch and the hole 1.7 inch. Thus maintaining the ratio between them compared to the real case, $\frac{1.7^2}{1^2} = \frac{8.5^2}{5^2}$.

A 11 second period sine wave with an amplitude of 1.5 meters gives a maximum velocity of 0.856 m/s. It was desired to keep this velocity in order to obtain the desired pressure drop over the BHA. To get a 0.856 m/s velocity for a sine wave with 3 second period the amplitude needs to be 41 centimeters.

3.2.2 *Simplifications*

During a connection scenario the main pump, which pumps mud down the drillstring, is slowly ramped down. At the same time the backpressure pump is slowly ramped up to provide sufficient flow through the choke. The BHA is connected to the end of a drillstring and starts moving up as a result of heave motions.

The BHA is a set of different components, as described in Chapter 2, but in the lab it is modeled as a single cylinder with an increased diameter compared to the drillstring.

Three problems were discovered by Gjengseth and Svenum [8] when having the BHA connected to the end of a drillstring. Firstly it is difficult to make the drillstring follow a given path, when only connected at one end. Secondly a diameter of 1 inch for the drillstring will lead to a flow of $26.2l/m$ which will require a pressure difference of 15.7 bar. This will lead the lab to require a working pressure as high as 31.4 bar [8]. The equipment required for handling such high pressures would be very expensive. The force required to move the drillstring would be so high, that it could potentially bend it. A bend on the drillstring would create local pressure variation in the hole. In addition having the BHA at the end of a drillstring would require both a pushing and pulling mechanism.

First it was recognized that the main goal of building the lab is to model surge and swab pressures. Thus making the main pump unnecessary since it is not desired to model a connection scenario with circulation of mud. This makes it unnecessary to have a hollow drillstring and it can be replaced by a solid one, increasing the stability of it. In order to solve some the problems mentioned above, the BHA was also connected to a lower rod making it look more like a piston. The advantage of having a lower rod is that there will not be any significant flow through the well, and thus it is possible to keep the working pressure of the system much lower. Having a lower rod also

Component	Diameter [mm]	Length [m]	Height [m]
Copper Pipe	19 (OD) / 16 (ID)	900	2.3
BHA	40.9	-	0.035
Upper Rod (connected to BHA)	25	-	-
Lower Rod (connected to BHA)	22	-	-
PVC pipe (Hole)	42.53	-	0.8

Table 1: Properties of components in the lab

ID = inner diameter, OD = outer diameter

increases stability, making it necessary only to have a pull mechanism and the drillstring will not bend.

The problem with not having a fluid displacement is that the control system is probably not able to accurately measure the pressure variation downhole by only using topside pressure measurements. To have some fluid displacement the lower rod was decreased in diameter.

For the drillstring it was assumed that it does not affect the pressure downhole by moving up and down, only the BHA. The advantage of doing this assumption is that the drillstring does not have to be extended all the way up to the top of the well. The drillstring is only necessary at the bottom of the well, making it possible to coil the pipe modeling the well. This makes it practical to build the lab.

The part of the well connecting the choke and the BHA was thus made a 900 meter long copper pipe coiled into a barrel. In addition the bottom of the hole was modeled by a transparent PVC pipe able to withstand 16 bar of pressure.

The final measurements used for the components in the lab are given in Table 1.

3.3 COMPONENTS

In this section each component used in the lab will be described. Figure 1 shows a sketch of the lab with measurements marked. A picture of the lab with some of the central measurements and components marked is shown in Figure 2.

3.3.1 *Backpressure Pump*

The backpressure pump provides sufficient flow through the choke. This is necessary in order to get a pressure drop over the choke. Fig-

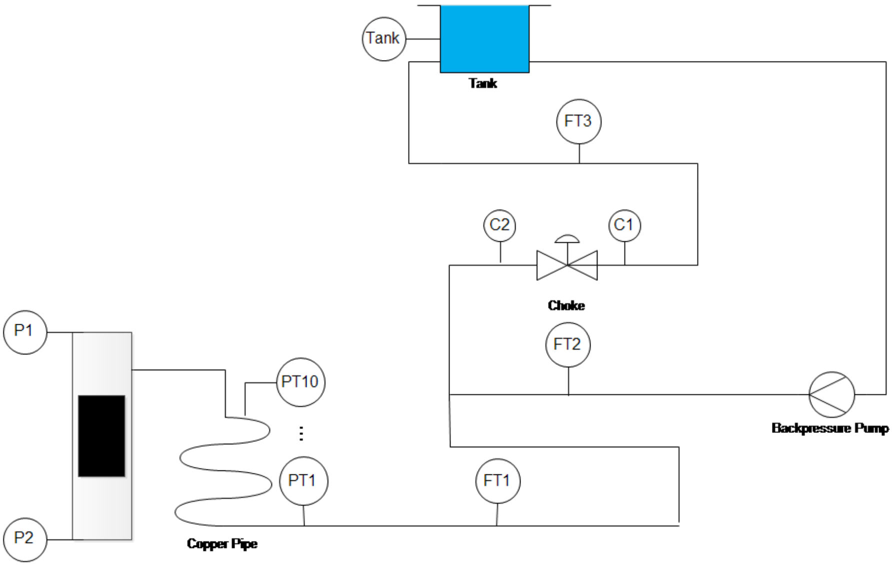


Figure 1: Lab Map

ure 3 shows the locker with the control buttons for the backpressure pump and the feed pump (see Section 3.3.2). The green button turns the power on, while the red turns it off. The switch turns on the feed and the backpressure pump. The feed pump has only the option to be turned on or off. The backpressure pump can adjust power by using the scroll button.

The backpressure pump was supposed to deliver a flow of 40 liter per minute [7], but according to the measurements FT2 and FT3 it does not deliver more than approximately 32 liter per minute. This was originally considered to be problematic when controlling the choke, since it might not provide a fast enough change in pressure drop as a response to a change in choke opening (see Section 4.1.2). However in later stages of designing the controller for the choke it turned out not to be a concern.

On the pump there is a manual pressure measurement, an automatic safety valve and a manual safety valve, named MV7 in Figure 12. Figure 4 shows a picture of the pump with valves and measurements marked. The manual safety valve should be closed during experiments as having it open will make the pump provide submaximal output and thus making it unable to provide a sufficient pressure drop over the piston.

3.3.1.1 *Problem*

During the fall of 2012 pieces of plastic got into the pump. It was discovered when the pump did not deliver the usual flow rate. The plastic pieces were restricting the piston movement and were removed. The only place the plastic pieces could have come into the system is at the tank with the feed pump, where the system is open to air. To

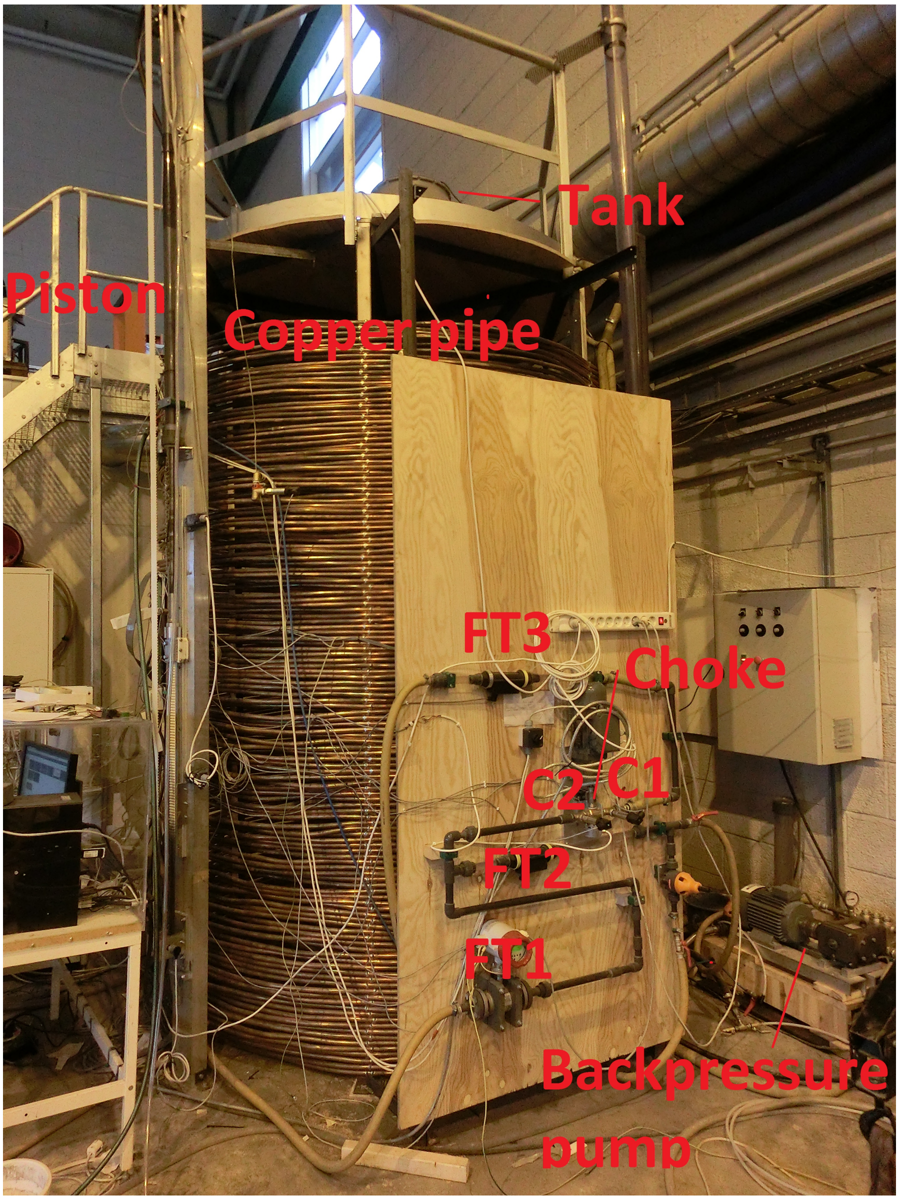


Figure 2: Lab with central measurements and components marked

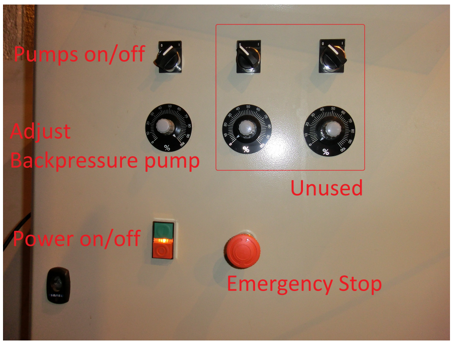


Figure 3: Locker with Control Buttons for the Backpressure Pump
The pump is currently turned on and is running at full capacity, 100 %

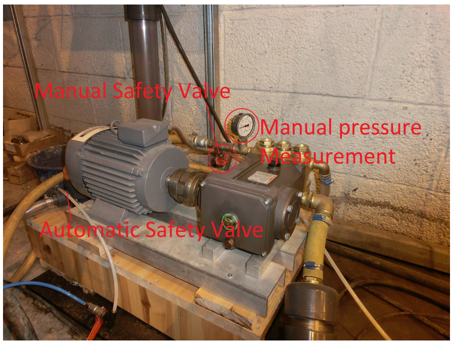


Figure 4: Backpressure Pump

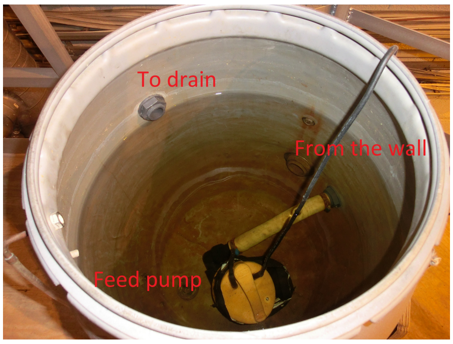


Figure 5: tank

avoid getting more contamination in the system the feed pump was wrapped with wire.

3.3.2 *Tank with Feed Pump*

The tank provides water for the pumps to pump through the system. It should be as full as shown in Figure 5. If it is below this level it should be filled, see Section A.7.

3.3.3 *Choke*

The choke is built with two components: A valve connected to an electrical motor. The valve is made so that it is able to turn completely around, in case the motor starts rotating. The range from 0... 360 degrees corresponds to four ranges from open to closed. Only one of these ranges were chosen. Figure 6 shows the choke in open and closed position. For more information on how the choke is controlled see Section 3.3.8. The choke is also equipped with a limit switch. This switch stops the motor if it starts going outside its operating range. It is also used in the homing procedure (Section A.6). The choke is tailor made by Jarle Glad.

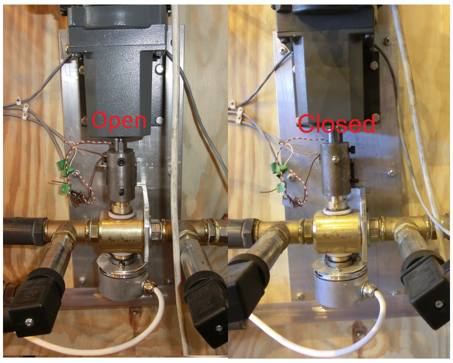


Figure 6: Choke

3.3.3.1 *Potential Problems*

During the spring of 2013 assistance from the producer of the motor and controller for the motor was hired to assist with setting up the software (see Section 4.2.2.2). Two comments were made by the Lenze personnel about potential problems with the current setup. First the electrical motor does not turn completely around. It could easily go warm, since the fan installed in it is designed for the motor turning. His suggestion was to install a fan on top of the motor. In addition the motor was not designed for being in the vertical position in this setup. Gravity could potentially lead the motor oil in the gear to flow away. It could be necessary to unscrew the motor and turn it every once in a while before putting it back in place.

3.3.4 *Pressure Tube*

A extra tube has been added to the lab during the spring of 2013. This tube enables to pressurize the system without the backpressue pump. The reason for this expansion was to test the lab for resonance frequencies. For more on these tests see Gleditsch [9]. Figure 7 shows a picture of this tube. See Section A.9 for a procedure on how to use it.

3.3.5 *Copper Pipe*

The copper pipe models the transport delay from the choke to the bottom of the well. It is equipped with a pressure measurement for every 100 meter of the pipe starting with PT1 closest to the choke and ending with PT10 closest to the bottom of the well.

3.3.6 *Bottom of Well*

The PVC pipe is used to model the bottom of the well. The pipe is transparent making it easy to see the BHA, which is black. It is in-

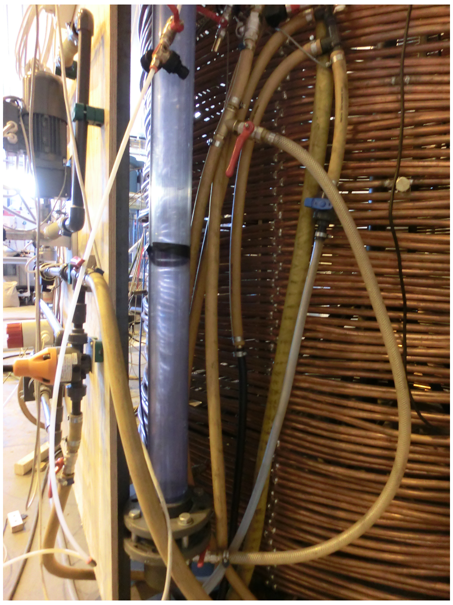


Figure 7: Pressure Tube

stalled a pressure measurement at the bottom (P2) and at the top (P1) of the well. In addition a flowmeter (FT4) was supposed to measure the flow in and out of the well. Unfortunately the producer of the flowmeter has not been able to deliver this as of June 2013.

The BHA, also referred to as the piston, is moved by a sawtooth belt, which is controlled by an electrical motor. At each end of the well the sawtooth belt is equipped with a limit switch. These switches are there for safety. If the motor attempts to run the piston out of the pipe the switch will stop it. In addition the switch for the lowest point in the well is used in the homing procedure (see Section A.6).

The pressure propagation waves that travel through the well are highly depended on the bulk modulus of the water in the well. This bulk modulus will change rapidly if air is introduced to the system. To remove air from the system see Section A.11.

3.3.6.1 *Problem*

If the controller for the piston and choke is enabled (switch in Figure 8), but no control signal is enforced on the piston controller, the piston will start drifting slowly downward due to gravity. Eventually it will flip the lowest limit switch. This will lock the electrical motor and make it necessary to reset the limit switch before being able to run the piston again. Before resetting the limit switch the piston needs to be manually moved away from it, or else it will only trigger the limit switch again. To pull the piston away from the limit switch and resetting it see Section A.4 and Section A.5.

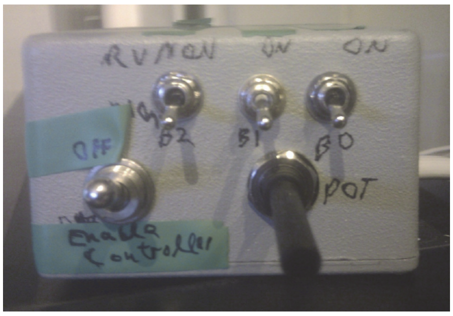


Figure 8: Control box for Piston and Choke

Inputs	-10 V	0 V	10 V
Piston Velocity Reference	Maximum downwards	Rest	Maximum upwards
Choke Position	-	Closed	Open

Outputs	0 V	5 V	10 V
Piston Position	Bottom	Middle	Top
Piston Velocity	Maximum upwards	Rest	Maximum downwards
Choke Position	Closed	Half open	Open

Table 2: Analog Inputs and Outputs for Piston and Choke

3.3.7 *Control Box*

Figure 8 shows the control box used to enable control and perform homing for the piston and the choke motor. To perform the homing procedure see Section A.6.

To shut down the controller, flip the *Enable Controller* switch up to the off position and the three other switches down.

3.3.7.1 *Problem*

The choke has suddenly started not to reach open position after performing the homing procedure. If this happens then first try to restart everything to see if the problem persists. If the problem persists the offset needs to be changed in the software from Lenze. To do this see Section B.4.

3.3.8 *Electrical Motors*

Both the electrical motors controlling the piston (LE2) and the choke (LE1) are controlled by the inverters shown in figure 9. There is a software from Lenze called Engineer used to set up these boxes. See Appendix B for documentation. There are multiple inputs for the two boxes. The most important is the analog inputs and outputs, which are expressed as voltage signals.

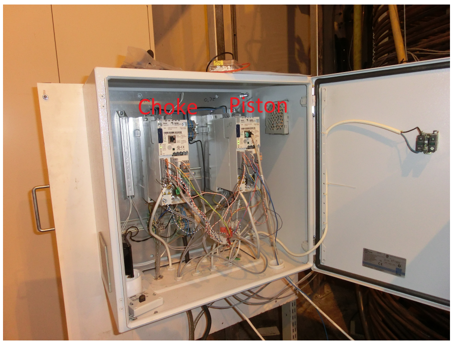


Figure 9: Inverters controlling the electrical motors

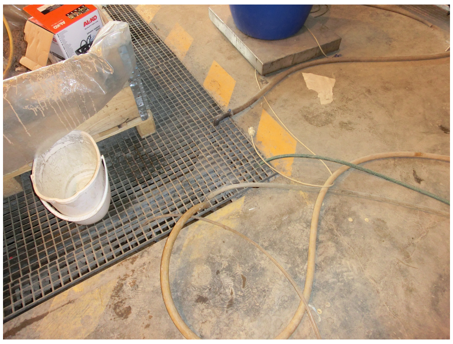


Figure 10: Drain

3.3.8.1 *Problem*

Along with the homing problem described in Section 3.3.7.1 the range of the choke has suddenly changed. If the choke does not home to the correct open position the whole operating range will be shifted. In addition it has been experienced that even after correcting the homing offset the 0..10 Volt does not correspond to the 90..0 degrees. Then the range of the input signal to the choke needs to be changed, see Section B.5 on how to do this.

3.3.9 *Drain*

Figure 10 shows the drain. Down here it is possible to empty out all the water in the system.

3.3.10 *Control Card*

A control card from National Instruments is used to take in all the measurements and set the outputs from the computer. Figure 11 shows the setup. The card has 32 analog input channels and 4 analog output channels. See table 3 for an overview of the input channels for the control card.

Channel	Tag	Measurement	Location	Signal
1	Empty	-	-	-
2	P2	Pressure	Bottom of well	0..10 Volt
3	P1	Pressure	Top of well	0..10 Volt
4	C2	Pressure	Choke inlet	0..10 Volt
5	C1	Pressure	Choke outlet	0..10 Volt
6	PT1	Pressure	0m copper pipe	0..10 Volt
7	Tank	Pressure	Tank	0..10 Volt
8	PT8	Pressure	700m copper pipe	0..10 Volt
9	PT6	Pressure	500m copper pipe	0..10 Volt
10	PT9	Pressure	800m copper pipe	0..10 Volt
11	PT2	Pressure	100m copper pipe	0..10 Volt
12	PT3	Pressure	200m copper pipe	0..10 Volt
13	PT7	Pressure	600m copper pipe	0..10 Volt
14	PT4	Pressure	300m copper pipe	0..10 Volt
15	PT10	Pressure	900m copper pipe	0..10 Volt
16	PT5	Pressure	400m copper pipe	0..10 Volt
17	Empty	-		-
18	FT1	Flow	Copper pipe inlet	0..10 Volt
19	FT3	Flow	Choke outlet	0..10 Volt
20	FT2	Flow	Backpressure pump outlet	0..10 Volt
21	FT4	Flow	Copper pipe outlet*	0..10 Volt
22	Empty	-	-	-
23	Empty	-	-	-
24	Empty	-	-	-
25	Empty	-	-	-
26	Empty	-	-	-
27	FT1	Direction	Copper pipe inlet	0..10 Volt
28	Piston Vel	Velocity	Piston Velcoity	0..10 Volt
29	Encoder	-	-	Digital
30	Empty	-	-	-
31	Piston pos	Position	Piston position	0..10 Volt
32	Choke pos	Pressure	Choke position	0..10 Volt

Table 3: Output Channels Control Card

*FT4 is currently (June 2013) not installed

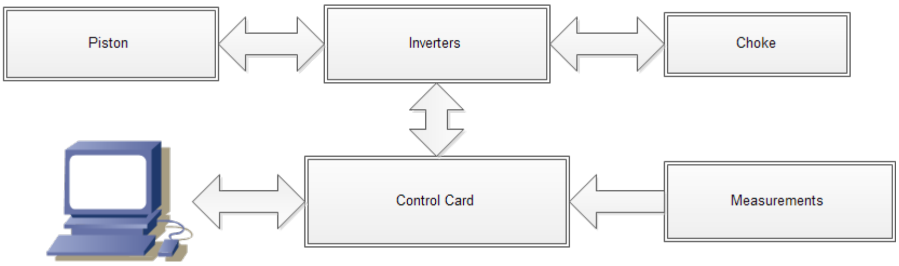


Figure 11: Control Card Setup

3.3.11 *Measurements*

There are basically two types of measurements used in the lab, pressure and flow. In addition there are some position and velocity measurements done by the electrical motors. All the pressure transmitters are delivered by Druck and are the UNIK 5000 models (except the one in the tank). They measure gauge pressure in the range 0..16 bar where 2 Volt = 0 bar and 10 Volt = 16 bar. The transmitter in the tank is a PTX1400 and measures in the range 0-100 mbar. All the transmitters should measure the pressure with an accuracy of 0.15%.

Two types of flowmeters are used in the lab. FT2 and FT3 are turbine flowmeters delivered by Parker measuring in the range 0-100 liter per minute. Here 2 Volt = 0 liter per minute and 10 Volt = 100 liter per minute. FT1 is a magnetic flowmeter able to measure flow in both directions. For FT1 2 Volt = 0 liter per minute and 10 volt = 8 liter per minute. A extra input channel provides the direction of the flow.

3.3.12 *Manual Valves*

There is a number of manual valves installed in the lab. These enables setting up different flow paths, manually close the choke or to drain the system. Figure 12 shows all the valves in the lab and the different flow paths. The backpressure pump pumps water in a closed circulation through the choke, up to the tank and back again. This path is marked by red in the figure. The path from the well to the closed circulation is marked with black. Here the flow moves in both the directions as a result of the piston movement and the choke movement. In addition there is a bypass line, marked with purple, which enables bypass of the copper pipe. The system can by pressurized without the backpressure pump by use of the pressure tube (Section 3.3.4). This path is marked with green. Finally the valves enable the system to be drained. All paths leading to the drain are marked with blue.

Table 4 lists the valves along with their function. All the manual once are controlled manually on the location they are placed except MV11 and MV12. These valves are controlled by an air pressured

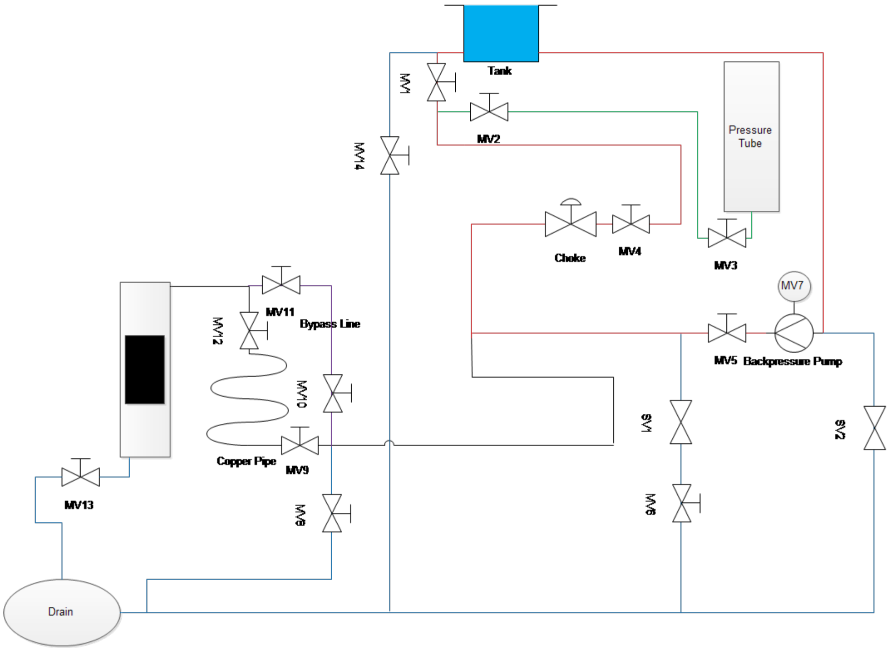


Figure 12: Map of the valves in the lab

switch placed to the left of the electrical motor controlling the choke.

3.3.13 *Safety Valves*

There are two safety valves in the system. One at the backpressure pump and one at the backpressure pump outlet. These will let the water out of the system if the pressure exceeds a threshold in order to protect the system. Through testing it was established that the safety valve at the backpressure outlet opens at approximately 14 bars.

3.3.13.1 *Problems*

When attempting to pressurize the system with the pressure tube (Section A.9) it was discovered that SV1 leaked. To stop this leak an extra valve was installed, MV5. This enables closing the automatic safety valve, which is a potential risk. *Thus MV5 should always be kept open when the backpressure pump is running.*

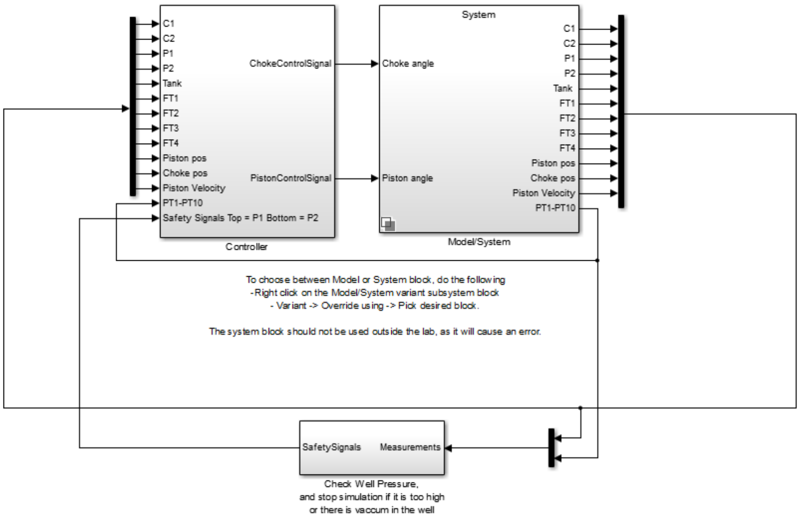
3.4 SIMULINK MODEL

The MATLAB toolbox SIMULINK has been chosen to take in measurements and set output signals in real-time. Figure 13 shows the SIMULINK model used with the lab.

The SIMULINK model has three main boxes. In the system box the input signals are taken in and output signals are set. In addition all

Tag	Type	Function
MV1	Manual	Enable flow to tank
MV2	Manual	Enable flow to pressure tube
MV3	Manual	Enable flow to pressure tube
MV4	Manual	Manual control of choke opening
MV5	Manual	Enable flow from backpressure pump (Used to pressurize system)
MV6	Manual	Enable flow to drain
MV7	Manual screw valve	Manual Safety Valve on backpressure pump
MV8	Manual	Enable flow to drain
MV9	Manual	Enable flow through copper pipe
MV10	Manual	Enable flow through bypass line
MV11	Manual	Enable flow through bypass line
MV12	Manual	Enable flow through copper pipe
MV13	Manual	Enable flow to drain
MV14	Manual	Enable flow to drain
SV1	Automatic	Safety valve
SV2	Automatic	Safety valve

Table 4: Manual and safety valves in the lab



To choose between Model or System block, do the following
-Right click on the Model/System variant subsystem block
- Variant -> Override using -> Pick desired block.

The system block should not be used outside the lab, as it will cause an error.

Check Well Pressure,
and stop simulation if it is too high
or there is vaccum in the well

Figure 13: SIMULINK model

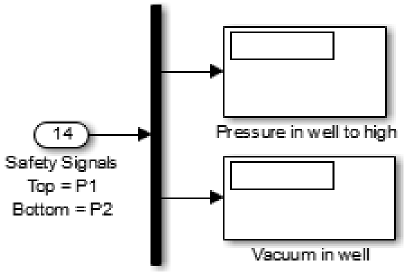


Figure 14: Safety Signals

the input signals are separated from each other and converted from Volt to bar or liter per minute. All the raw measurements are stored by writing them to the workspace, such that the user can use the *save('filename.mat')* command to save all measurements.

The small box at the bottom of the figure is a safety box installed in order to make sure that the pressure in the well stays between -0.5 bar and 10.5 bar in the well. The pressure transmitters measure gauge pressure, which means that measurements of less than -1 barg indicates vacuum in the well. Both vacuum and too high pressure is dangerous for the equipment and should be avoided. Thus the simulation is stopped if either P1 or P2 shows too high or low pressure.

The last box is the controller. Here the control of the piston, the choke and the bottomhole pressure are implemented using all the measurements and output signals. In addition the safety signal from the safety box is taken in. This enables the user to quickly see what has happened when the simulation aborts. Figure 14 shows this box.

The SIMULINK model uses a callback function in order to initialize the model for each run. The file called each time is initModel.m. This file is divided into three parts. The first part is concerned with setting up the controller, see Chapter 7. It can easily be changed if another controller is desired with a different setup.

The measurements have some biases. In addition the choke controller requires some of the measurements to be filtered. The biases were found by having no flow in the system and it ran for approximately two minutes. Then the biases were calculated for selected measurements and stored along the filters for pressure and flow in *biasFilter.mat*. In the second part of *initModel.m* the biases and filters are loaded. In the last part of of the *initModel.m* a dataserie is made in order to identify the choke characteristics, see Section 4.4.2. IMPORTANT: initModel.m clears the workspace of MATLAB, thus the data from a test should be saved in a *.mat* file before a new test is run.

For more information about the choke controller and the bottomhole controller see Chapter 4 and Chapter 7. More information about the choke and piston input signals are found in Section A.13 and Section A.12.

4 CHOKE

During the fall of 2012 it was discovered that the choke installed in the IPT-Heave lab had a backlash Albert [3]. This made the choke perform too poorly for control. As a consequence a new tailored choke was installed between the fall semester of 2012 and the spring semester of 2013. In this chapter the process of testing and tuning a controller for the choke will be addressed.

All pressure signals in this chapter have been filtered with a first order lowpass filter with cutoff frequency of 50 Hz. The choke measured position signal has not been filtered and the flow signals have been filtered with a first order lowpass filter with cutoff frequency of 30 Hz.

4.1 TESTING THE NEW CHOKE

Before installing a controller for the choke two questions needed to be answered. First: Is there still deadband in the choke? Second: How fast is the choke able to move? Tests to answer to these questions were performed and the results are given in this section. In addition the problem with the backpressure pump not delivering enough flow rate is illustrated.

4.1.1 *Choke Properties Tests*

To begin some new properties of the choke had to be established. The new choke is, as the old choke, controlled by an electrical motor delivered by Lenze. The motor takes in a signal between 0-10 volt that through multiple conversions corresponds to 0-360 degrees choke opening. By having a small flow through the choke and visual measurements it was established that 1.75 volt corresponded to fully closed choke, while 4.25 volt corresponded to fully open.

To establish a lower limit for the choke opening the backpressure pump was set to 100%, corresponding to about 32 liter per minute of flow, while the choke opening was steadily decreased. 38 degrees was found to give a pressure of about 10.5 bar at the choke inlet. Even though a lower opening was established no software saturations block was applied in SIMULINK. The safety valve, SV1, was tested and found to be working. The advantage of not having a software saturation block is that it can affect the controller performance. A safety valve is both safer and does not impact the performance of the controller.

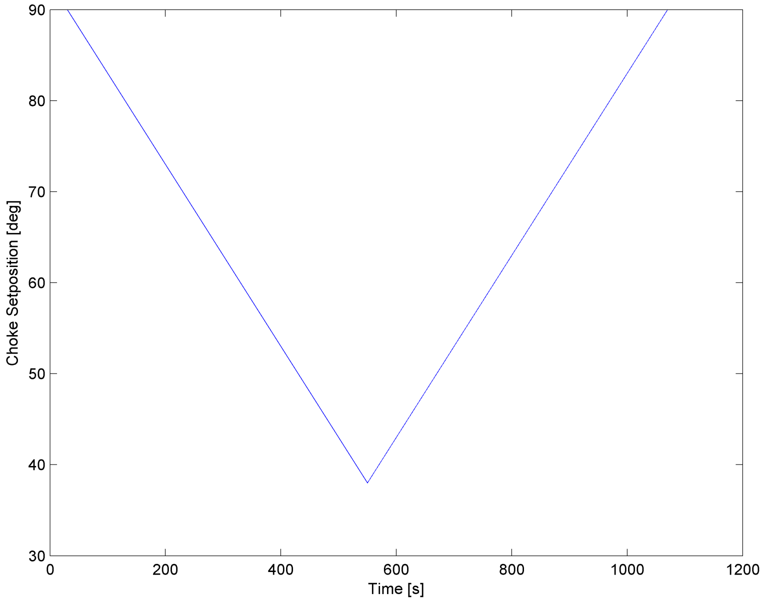


Figure 15: Input Choke Set Position Backlash Test

In the report Albert [3] a step function was used to test for backlash in the choke. The disadvantage of using this approach is that during a step the choke set position will not correspond to the actual choke position at all times. To overcome this disadvantage a slowly varying ramp function was used as illustrated in Figure 15. The result is shown in Figure 16. As can be seen in the figure the backlash is approximately one bar at its worst. Based on the test, the operating range for the choke was considered to be between 38 and 75 degrees.

To find out how fast the choke was able to move a step was applied. It was discovered that a step from 90 to 38 took 0.8 seconds to reach the desired position. To further test the speed of the choke a time varying sine signal was applied to the choke set position. Since the operating range for the choke was found to be from 38 to 75 the sine signal was set to vary between these values. Different frequencies were tested and the result is shown in Figure 17. It can be seen from the figure that increasing frequency makes it difficult for the choke to follow. The disturbance designed for the lab has a period of 3 seconds, which corresponds to 0.33 Hz in frequency. For this frequency a set position of 55 degrees will result in the choke being either about 45 degrees or about 65 degrees depending on whether the choke is closing or opening.

4.1.2 *Pump*

It was discovered during the fall of 2012 that the backpressure pump did not deliver a flow rate of more than about 32 liter per minute

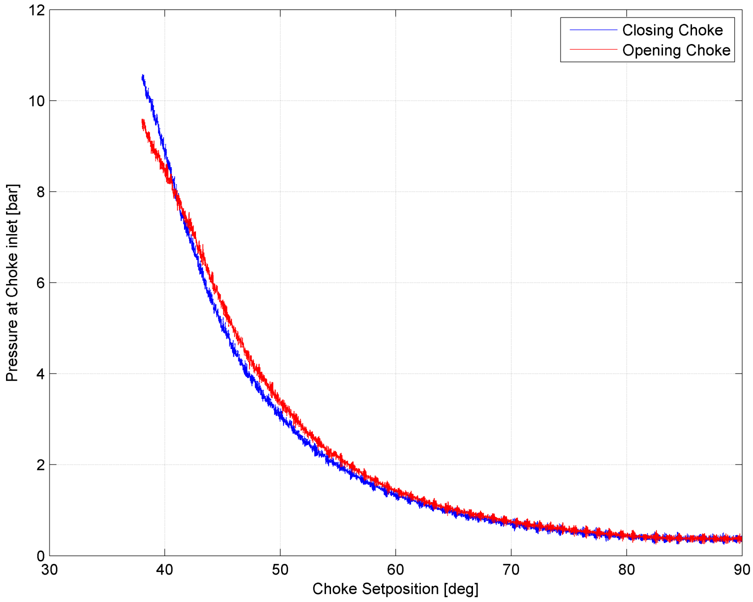


Figure 16: Pressure at Choke Inlet vs Choke Setposition

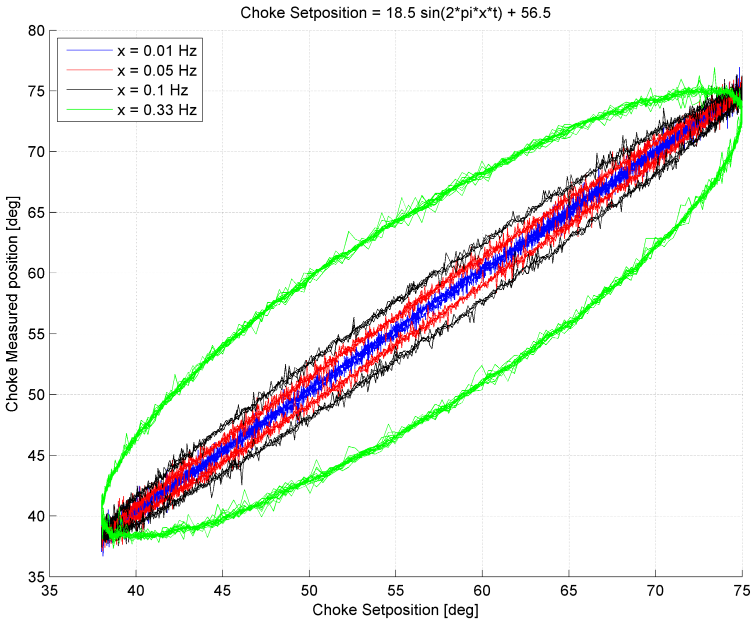


Figure 17: Choke Speed Test

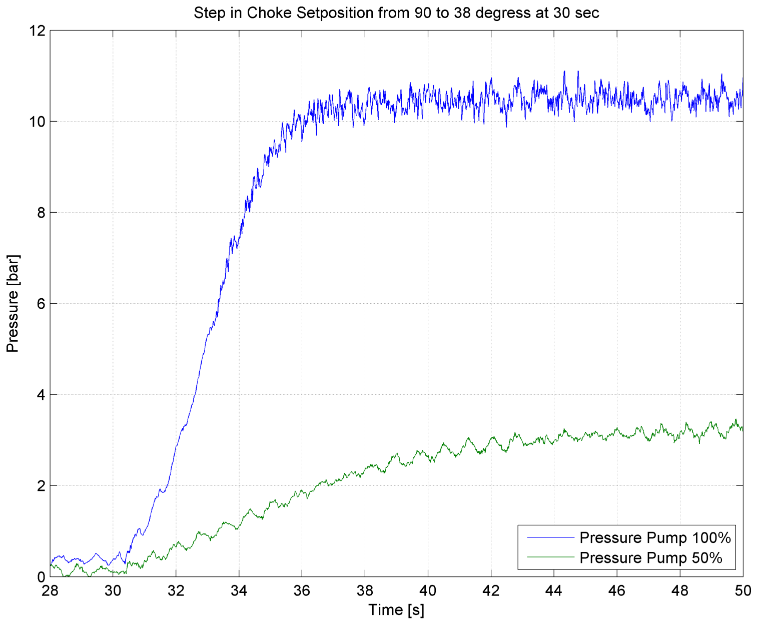


Figure 18: Choke Speed Pressure Test

according to FT2 and FT3 Albert [3]. In the construction of the IPT-Heave lab the pump was expected to deliver about 40 liter per minute [8]. This affects how fast the choke is able to change the pressure. This is illustrated in Figure 18. Here a step has been done from 90 to 38 degrees twice. First the backpressure pump is set at 100% corresponding to about 32 liters per minute and then with 50% corresponding to about 18 liters per minute. As can be seen from the test result the pressure takes more time to increase with a slower flow rate.

4.2 ACTIONS CONSIDERED AFTER CHOKE TEST RESULTS

Without the backpressure pump fulfilling the demands it becomes difficult to address whether the choke was performing satisfactory, meaning moving fast enough with a small enough deadband. However with the choke moving this slowly it was concluded that the choke did not perform well enough for control. A couple of options were looked into to solve this problem.

4.2.1 *Buying a New Choke*

Due to the multiple problems with the choke both having a deadband and moving too slowly, buying a new choke was considered. Both the author of this thesis and Phade [24] looked into finding a new choke. This resulted in two offers. One from Solberg & Andersen As v/Stian Andersen and one from EPC. Both offers included a pneumatic valve.

After reviewing the properties of both offers it was concluded that none of the chokes could be expected to perform better than the choke already installed. None of the chokes were able to have a closing time within one second.

4.2.2 *Improving the Installed Choke*

After some discussing it was concluded that the electrical motor controlling the choke should be able to deliver more power than it already does. By changing the software controlling the motor it should be possible to get the choke to move faster. An encoder was also installed to measure the actual choke position and not just the position of the motor as has been done earlier. This may make it possible to completely remove the deadband.

4.2.2.1 *Encoder*

An incremental decoder was installed in order to measure the choke position. The encoder was connected to the computer using channel 28 and 29. It works as following: A clockwise rotation is encoded by the cycle $00 \rightarrow 01 \rightarrow 11 \rightarrow 10 \rightarrow 00$, a counterclockwise rotation is encoded $00 \rightarrow 10 \rightarrow 11 \rightarrow 01 \rightarrow 00$. In the encoder one full rotation of 0-360 degrees is divided into 5000 pieces, and one such cycle corresponds to one piece or 360/5000 degrees. Thus the encoder has good resolution.

The problem with the encoder is that it requires a fast sampling rate. Too slow sampling rate will lead to samples being lost. Say if an input becomes $00 \rightarrow 11$ there is no way of knowing if the choke has rotated clockwise or counterclockwise.

Through trial and error the fastest sampling time for this SIMULINK real-time application was established to be 1e-4 seconds between each sample corresponding to a frequency of 10 kHz. From the datasheet for the encoder it is listed that the maximum frequency for it is 300 kHz. Thus in order to use the encoder it cannot be implemented with SIMULINK, but must be used with the Lenze software.

Even though a faster sampling rate is needed to capture the movement of the choke when it is moving fast, a slowly varying ramp was applied to the input as shown in Figure 15. The result is shown in Figure 19 and Figure 20. As can be seen in the figure the encoder is able to measure the deadband of the choke, making it possible to remove it.

4.2.2.2 *Software Change*

The motor control was set up using Lenze software called Engineer by Espen Øyebø during the summer of 2012 [3]. Through studying the manual for the motor it was established that the setup used was

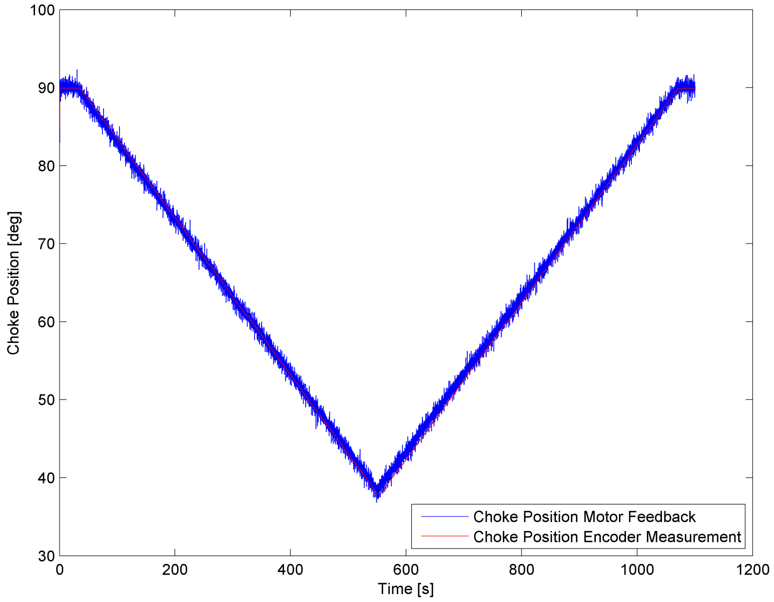


Figure 19: Choke Setposition measured with encoder

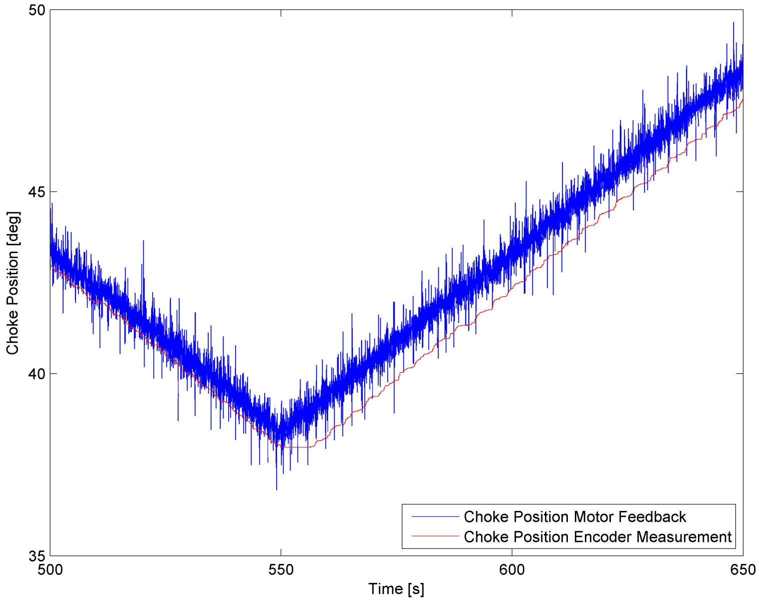


Figure 20: Choke Setposition measured with encoder deadband view enlarged

not ideal and should be improved. In order to save time, personnel from Lenze was hired to help change the setup used in the Lenze software for both the choke and the piston (for more about the piston see Drønnen [5]). The software was changed so that the choke could move faster. As before 0-10 Volt signal was used to control the choke opening, but now 0 Volt corresponds to 90 degree (open) opening and 10 Volt corresponds to 0 degree opening (closed). After homing the controller the choke should go to fully open.

4.2.2.3 *Results of improving the Choke*

After doing the software change for the choke it was tested and the performance was considered sufficient. The encoder was never implemented. The main reason for this is that it needs to be connected with the motor controller and a change in the Lenze software must be made. As of writing this no port on the motor controller with a fast enough sampling rate was available. Should it be desirable to connect the encoder at a later stage Kristian Bakken from Lenze (email: Kristian.Bakken@lenze.com) should be contacted to assist through phone support to help make the necessary changes in the Lenze software.

4.3 RETESTING AFTER IMPROVED CHOKE

After doing the improvement on the choke the test done in Section 4.1.1 was redone. First 41 degrees was found to give about 10 bars at the choke inlet. Then a step was done from 90 to 41 degrees. Earlier about 0.8 seconds were spent in reaching the desired position, after the software change the choke used 0.3 seconds. However the choke overshoots it position with about 2 degrees after the software change and spends 1.2 seconds to settle at the desired value. The other speed test was also performed: The choke was set to a time varying signal of $18.5 \sin 2\pi x t + 56.5$. In Figure 21 the choke measured position is compared to the choke set position before and after the software change. As can be seen in the figure there is a clear improvement. For a period of 3 seconds the choke has gone from 45-65 degrees measured choke opening for a 55 degrees desired opening, to 50-60 degrees measured choke opening.

No improvement has been done on removing the deadband, and thus the deadband test was not done again.

4.4 CHOKE CONTROLLER

After having improved upon the choke an attempt to make a controller was made. In this section the controller derivation of a PID controller with feedforward is done based upon the suggestions from Albert [3]. Gleditsch [9] discovered that the hoses from the end of the

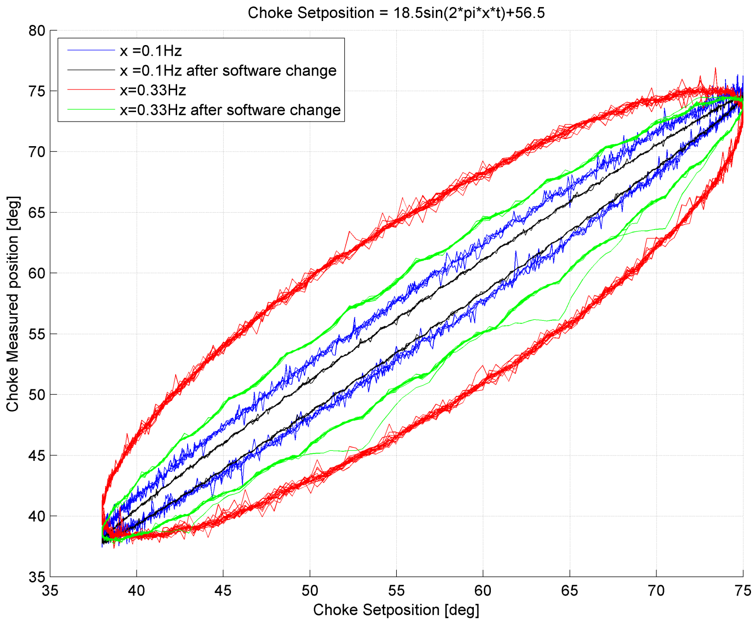


Figure 21: Choke speed compared before and after software change

copper pipe to the choke introduces undesired dynamics into the system. To bypass these dynamics the choke controller was made to have the choice between using the pressure measurement at the choke, C2, and the pressure measurement at the end of the copper pipe, PT1, as reference.

4.4.1 *Filter*

During the design process of the controller it was discovered that the filter used to filter the pressure and flow measurements greatly affected the controller performance. Speed is an essential component to the controller and since a filter introduces a delay it were chosen carefully. Higher order filters introduces more time delay than a lower order filter, and thus a first order filter had to be chosen. The cutoff frequency is related to delay as follows: $\tau = \frac{1}{2\pi f_c}$ where τ is the time delay and f_c is the cutoff frequency [20]. A lower cutoff frequency is desired in order to remove noise, but because of the delay it introduces it cannot be chosen too low. Through trial and error the filter $\frac{50}{s+50}$ was used for pressure measurements and $\frac{30}{s+30}$ was used for flow measurements.

The resulting controller output ended up being very noisy. However the noise could not be removed from the measurements without affecting performance too much. To avoid too much wear and tear on the choke equipment the output was thus filtered with the following filter $\frac{30}{s+30}$.

4.4.2 *Choke Characteristics*

In order to design the controller a choke characteristics was needed. The following equation for a choke was used:

$$G(u) = q_c \sqrt{\frac{\rho}{p_c - p_0}} \tag{1}$$

where u is the choke opening, $G(u)$ is some strictly increasing characteristics function, q_c is the flow through the choke, ρ is the density of the liquid and p_c and p_0 is the pressure at the choke inlet and outlet. The equation can be derived from a Bernoulli equation for steady flow.

A characteristic is needed to be identified for both C2 and PT1 as reference. The following procedure was used to identify characteristics, $G(u)$, for both choices as reference:

1. Run the choke through the slowly varying ramp function, shown in Figure 15.
2. Calculate $G(u)$ using Equation 1 with $\rho = 998.2$.
3. By looking at Equation 1 it can be observed: $|p_c - p_0| \rightarrow 0$, $G(u) \rightarrow \infty$. Removing all points with $u \geqslant 70$ degrees avoids this problem with low pressure drop. Instead in the implementation a maximum angle of 70 degrees was set for the feedforward.
4. The MATLAB function polyfit() was used approximate G(u) as a user set order polynomial. polyfit() uses a least squares technique to approximate the data.

The resulting polynomial identified with C2 as reference:

$$G(u) = 0.01u^3 - 0.80u^2 + 44.29u - 790.82 \tag{2}$$

The resulting polynomial identified with PT1 as reference:

$$G(u) = 21.53u - 0.80 \tag{3}$$

4.4.3 *Controller Implementation*

As suggested by Albert [3] a PID controller with feedforward was implemented. However a modification had to be done: A gain scheduler had to be implemented for the integral gain.

4.4.3.1 *Feedforward*

The feedforward part of the controller was implemented using Equation 1. By changing the inlet pressure, p_c, with the desired inlet pressure, $p_{c,ref}$, and solving the equation with regards to the choke opening leads to:

$$u = G^{-1}(q_c \sqrt{\frac{\rho}{p_{c,ref} - p_0}}) \tag{4}$$

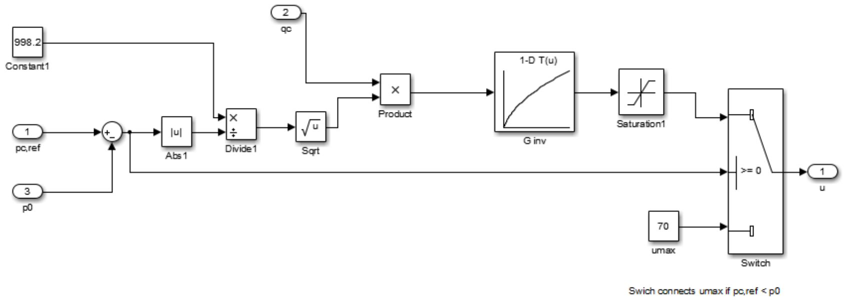


Figure 22: Feedforwad Controller implementation

The inverse of G(u) was implemented using a look-up table block from SIMULINK. To avoid getting an error in the output if $p_{c,ref} \geqslant p_0$ a check was done. If $p_{c,ref} \geqslant p_0$ the controller was set to give out 70 degrees. A saturation block was also used to make sure that the feedforward controller is only operating in the range it has been designed for, having a lower limit of 41 degrees and an upper of 70 degrees.

4.4.3.2 *PID*

During initial tuning discoveries were made. First derivate action only worsened the performance of the controller and thus was removed. This is not surprising since the measurement contains much noise. Second the integral gain turned out to be difficult to tune. A value that worked well for low pressures, did not work well for higher pressures. This motivated the idea from Albert [3] to implement a gain scheduler.

The PI controller was implemented with anti-windup and tracking in a parallel form. The controller implementation can be seen in Figure 23.

For the gain scheduler a different approach was used than suggested by Albert [3]. In Albert [3] a set of PID-controllers implemented in parallel with use of tracking to get bumpless transfer was suggested. Instead the integral gain was implemented as a function given by the current pressure and the pressure deviation. A lookup table was used for each input and the two were multiplied together. At the end of the integral gain function a saturation block was set to make sure the integral stayed within certain limits. The advantage with such a continuous integral gain function is that bumpless transfer does not become an issue. It also enables the user to specify an integral gain for each given pressure. The implementation of the integral gain function is illustrated in Figure 24.

The connection between the choke opening and the inlet is inverse, meaning that increasing choke opening actually means decreasing pressure at the inlet. To compensate for this the output of the con-

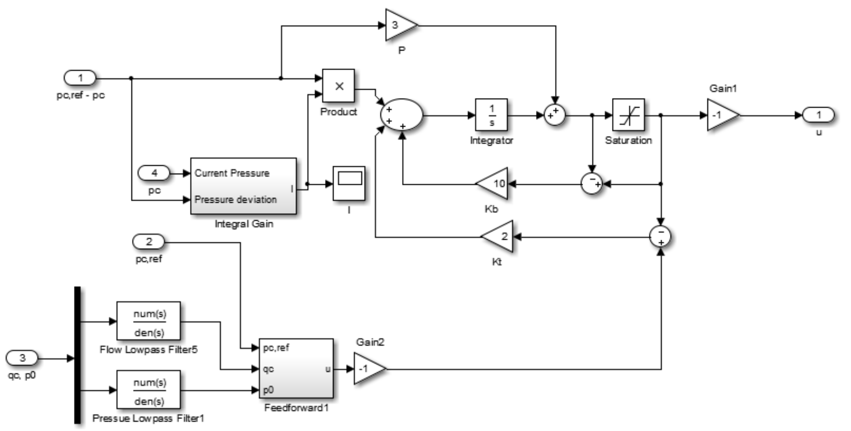


Figure 23: PID with feedforward implementation

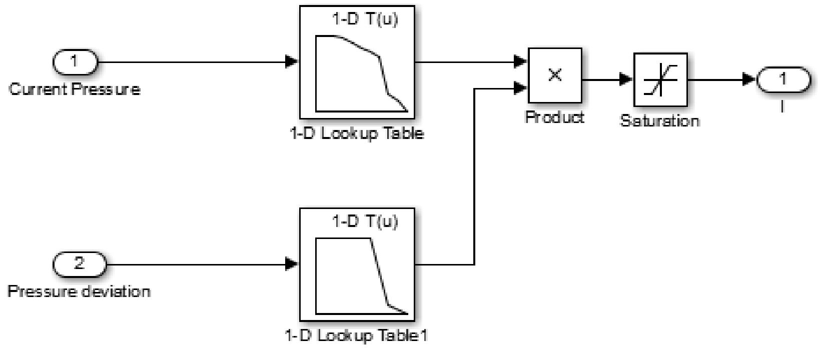


Figure 24: Integral Function Implementation

Reference	P	K_b	K_t	$I_{current\_pressure}$	$I_{pressure\_deviation}$
C2	3	10	2	y = [30 30 30 29 27 25 24 22 8 5 1] x = [0:10]	y = [1 1 1 1 0.2 0.1] x = [0:5]
PT1	1	10	2	y = [30 30 25 20 20 17 15 13 10 5 5] x = [0:10]	y = [1 1 0.7 0.2 0.2 0.1] x = [0:5]

Table 5: Tuning Parameters for Choke Controller

Test	Pressure Reference
Test 1	$2\sin(2\pi\frac{1}{3}t) + 4$
Test 2	$1\sin(1.5\pi\frac{1}{3}t) + 2\sin(2\pi\frac{1}{10}t) + 4.5$
Test 3	$1\sin(2\pi\frac{1}{4}t) + 1.5\sin(2\pi\frac{1}{7}t) + 0.5\sin(2\pi\frac{1}{11}t) + 5$

Table 6: Reference Signals used for Testing Choke Controller

troller had to be multiplied by -1 and thus making it necessary to multiply the feedforward part with -1 before entering it on the tracking part of the PI controller. This lead to the initial for the integral part of the controller being -90 instead of 90 and the upper and lower limit of the anti-wind-up being 0 and -90 instead of 90 and 0.

4.4.4 *Tuning*

The controllers were tuned manually online. The fastest and thus most difficult reference signal for the controllers is expected to be a sine wave with a 3 second period. In addition the sine wave is expected to vary around ± 2 bar. For the online tuning a reference signal of $2\sin(2\pi\frac{1}{3}t) + 4$ was chosen. The resulting tuning parameters can be found in Table 5.

4.4.5 *Controller Performance*

Three different tests were performed in order to assess the controller performance. The reference signal for each test can be found in Table 6. In the first test the controller was suddenly turned on to see how it reacted. Parts of the test have been picked to illustrate performance for C2 as reference in Figure 25, Figure 26 and Figure 27. Similar results can be found for PT1 as reference in Figure 28, Figure 29 and Figure 30.

4.5 CHOKE CONCLUSION

It was concluded from the results obtained in Section 4.4.5 that the controller designed performed satisfactory. It can be seen from all the

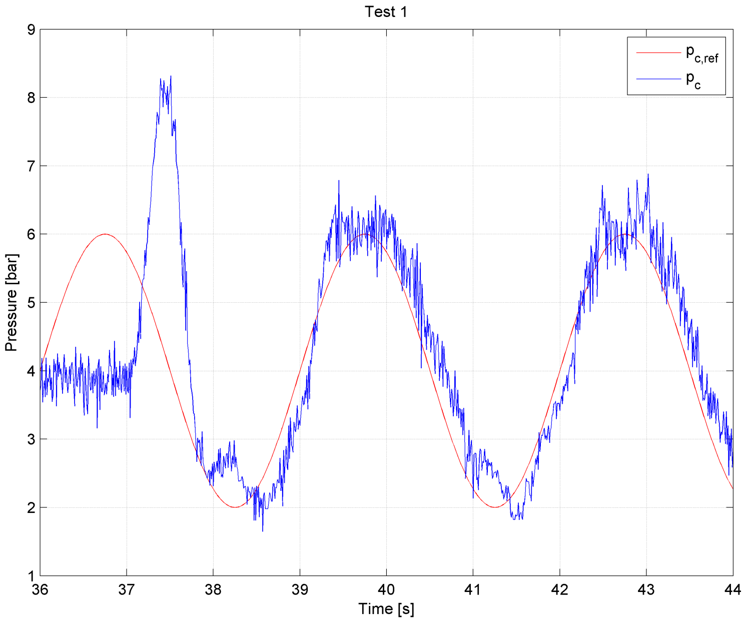


Figure 25: Test 1, C2 as Reference

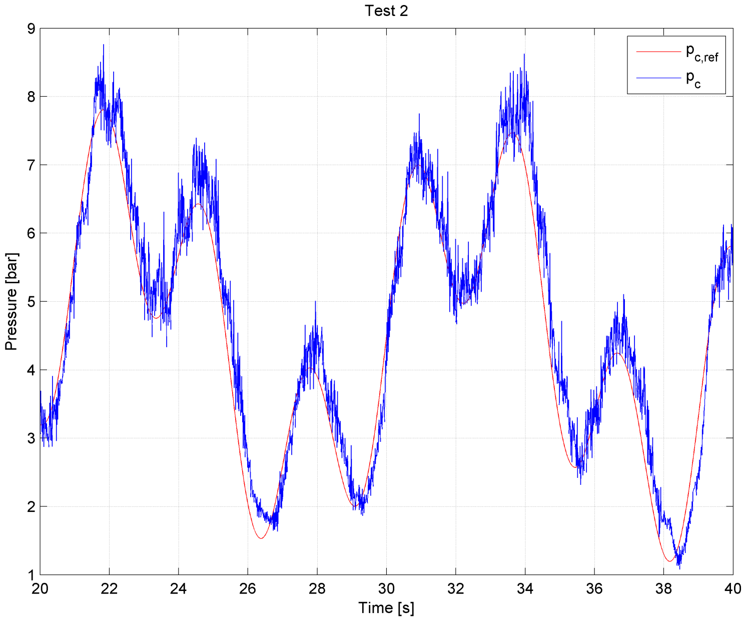


Figure 26: Test 2, C2 as Reference

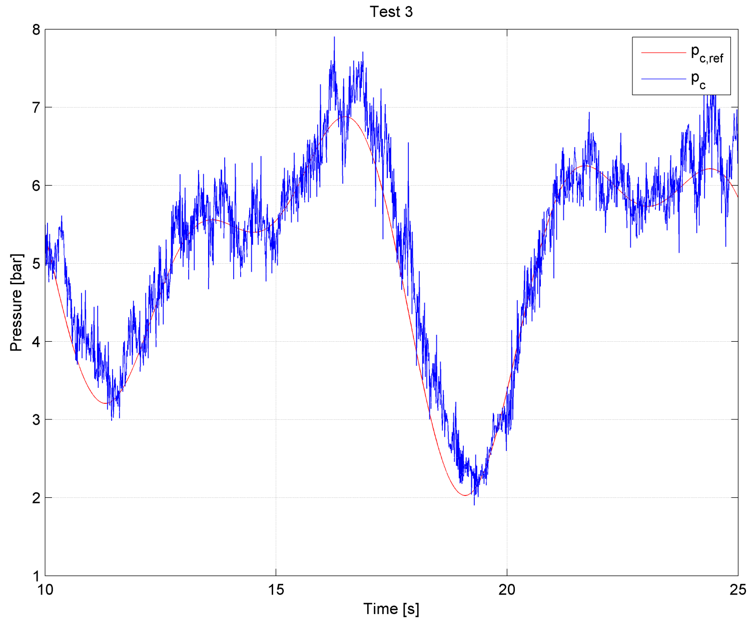


Figure 27: Test 3, C2 as Reference

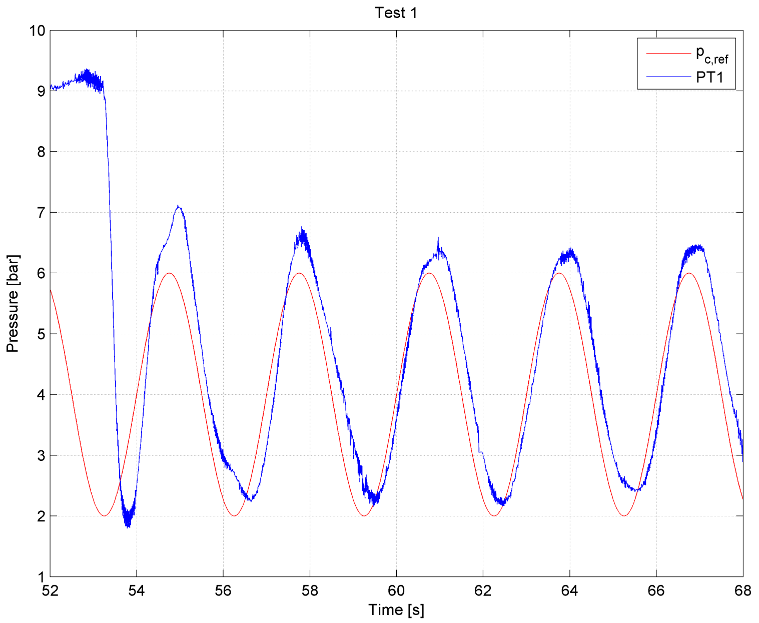


Figure 28: Test 1, PT1 as Reference

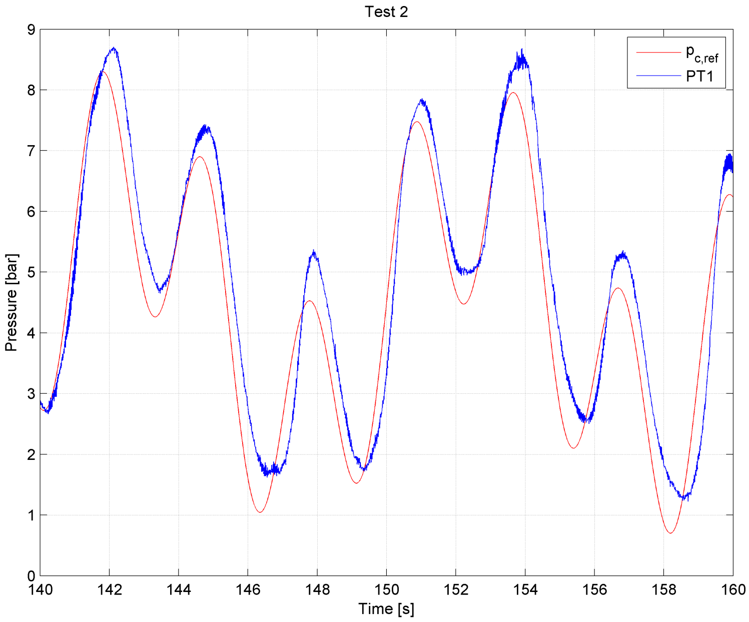


Figure 29: Test 2, PT1 as Reference

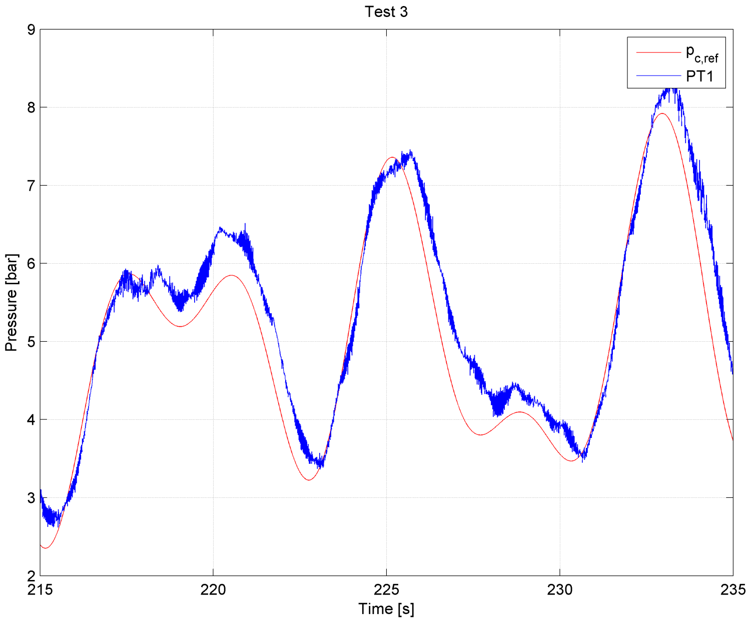


Figure 30: Test 3, PT1 as Reference

tests that the pressure is lagging behind the desired pressure. However, this is to be expected since it takes some time to convert digital signals from the computer to an analog input to the motor. A total lag of approximately 0.2 seconds was therefore considered to be the best performance possible with the current setup. In addition changing the reference to PT1 worsens the performance. This is also expected since the distance between the choke and PT1 is much larger than between the choke and C2, thus introducing an extra delay in the control loop, which deteriorates the performance.

5 REVIEW OF LITERATURE ON THE DISTURBANCE REJECTION PROBLEM

The issue of disturbance rejection, when drilling from a floating rig, has been studied to some extent. Two main models have been developed by Landet et al. [17] and Mahdianfar et al. [18], which will be referred to as the Landet Model and the Mahdianfar Model. The Landet Model has been developed further by Landet et al. [16]. In addition Aamo [1] has used the Landet Model to transform into a generalized model, for which he has developed a control algorithm. Aarsnes [2] has used the same approach as Landet et al. [16] to develop a more complex model. In addition both Landet et al. [17] and Mahdianfar et al. [18] have developed control algorithms for disturbance rejection. In the current chapter the different models with extensions will be presented along with the control algorithms for disturbance rejection.

5.1 SYMBOLS

In order to ease the comparison of the different models and controllers one set of symbols is used. Following is a list of all the symbols used together with an explanation.

p = pressure [Pa]

$\tilde{p}$ = deviated pressure [Pa]

$\bar{p}$ is linearizing point of pressure [Pa]

v = the mud velocity [m/s]

$\tilde{v}$ is deviated mud velocity [m/s]

$\bar{v}$ = the average mud velocity [m/s]

q = mud flow [m^3/s]

A = the cross-sectional area

ρ = the mud density

$\dot{m}$ = mass flow over the cross-section

$d\dot{m}$ = mass flow change over cross-section of a control volume

β is the bulk modulus of the mud

$\bar{\beta}$ = the effective bulk modulus

x = the spatial coordinate

$\hat{F}$ = the frictional pressure loss per unit length

g = the gravitational constant

$\gamma(x)$ = the inclination angle of the pipe

l^j = the length of each control volume

Δh_i = the height of each control volume

$x = 0$ position bottom of the well

$x = l$ position outlet of well, choke

K_c is a choke constant

p_c is the pressure at the choke inlet

p_0 is the pressure at the choke outlet

$G(u)$ is a strictly increasing and invertible function relating to the control signal and the actual choke opening

c = the speed of sound in the mud

ω = frequency

ϕ = phase

a = amplitude

k = coefficient for linear viscous friction

K_1 = Hopp-strain coefficient

N is the number of control volumes

5.1.1 *Subscripts*

The subscripts are used to indicate derivative. It can also be used to indicate what part of the model it is. For example A_a means the cross-sectional area of annulus.

t = the partial derivative with respect to t

x = the partial derivative with respect to x

a = annulus

d = drillstring

b = openhole

c = choke

bpp = backpressure pump

5.1.2 *Superscripts*

In the control volume models and the discretization of models, the superscripts are used to indicate which control volume it is or the part of the discretization.

j = properties of the jth control volume or discretization

N = properties of Nth control volume or discretization

1 = properties of the first control volume or discretization

5.2 MODELING

The two main models developed by Landet et al. [17] and Mahdianfar et al. [18] both use mass and momentum balances. However a different approach is used for the two models. While Landet et al. [17] takes a control volume approach, Mahdianfar et al. [18] utilizes differential analysis. Aarsnes [2] uses a control volume approach as Landet et al. [17] does, however he uses it not only for the annulus as done by Landet et al. [17] and Mahdianfar et al. [18] but also for modeling the drillstring and the openhole in the bottom of the well. This section contains all three models, in addition to the transformation by Aamo.

5.2.1 *Landet Model*

Landet et al. [17] uses a differential control volume with mass and momentum balances to model the annulus. The control volume has area A, density ρ, pressure p and length dx. Reynolds Transport Theorem (White [30, p.143]) applied on a single dimension pipeline for mass and momentum yields:

$$A_a dx \frac{\partial \rho}{\partial t} = \dot{m} - (\dot{m} + d\dot{m}) = -d\dot{m} \tag{5}$$

$$\frac{\partial}{\partial t}(\rho \bar{v}) A_a dx = A_a p - A(p + dp) + \int_A \rho v^2 dA_a - \\ \int_{A_a} [\rho v^2 + d(\rho v^2)] dA_a - \hat{F} dx + A_a \rho g \cos(\gamma(x)) dx \tag{6}$$

In the derivation an assumption of the connection between density and pressure are made: $dp = \frac{\beta}{\rho} d\rho$. In addition the term $\frac{\partial}{\partial x} \int_{A_a} \rho v^2 dA_a$ is considered to be small and is therefore ignored. This result in a set of similar equations as derived in Kaasa et al. [13]:

$$\frac{\partial p}{\partial t} = -\frac{\beta}{A_a} \frac{\partial q}{\partial x} \tag{7}$$

$$\frac{\partial q}{\partial t} = -\frac{A_a}{\rho} \frac{\partial p}{\partial x} - \frac{\hat{F}}{\rho} + A_a g \cos(\gamma(x)) \tag{8}$$

The annulus is then divided into N control volumes and the equations (7) and (8) are integrated over each control volume. Two boundary conditions are needed in order to solve this set of equations numerically. First it is considered that the bottom volume will continuously change with $v_d(t) A_d$. Second the pressure at the top of the well, by the choke, is assumed to be known given by the equation:

$$q_c = K_c \sqrt{p_c - p_0} G(u) \tag{9}$$

This results in the following set of equations describing the flow during connections:

$$\begin{aligned}
\dot{p}^1 &= \frac{\beta^1}{A_a^1 l^1}(-q^1 - \nu_d A_d) \\
\dot{p}^j &= \frac{\beta^j}{A^i l^j}(-q^{j-1} - q^j) \qquad i = 2, 3, \ldots, N-1 \\
\dot{p}^N &= \frac{\beta^N}{A^N l^N}(-q^{(N-1)} - q_c + q_{bpp})
\end{aligned} \tag{10}$$

$$\begin{aligned}
\dot{q}^j &= \frac{A^j}{l^j \rho^j}(p^j - p^{j+1}) - \frac{F^j(q^j)A^j}{\rho^j l^j} - A^j g \frac{\Delta h^j}{l^j} \\
i &= 1, 2, \ldots, N-1
\end{aligned} \tag{11}$$

In order to model the friction Landet et al. [17] looks at experimental data. It turns out that the fluid behaves in accordance to the theory of Newtonian fluids. This means that for low flow rates the flow is laminar with a linear model describing the friction loss. At higher flow rates the flow becomes turbulent and a nonlinear model is needed to describe the friction loss. An additional challenge with modeling the friction loss is that the drilling fluid behaves much like a gel. This is an advantage since it prevents cutting from sinking when the main pumps are turned off. It means that there is statical friction that has to be overcome before any flow can occur, making it difficult to model.

5.2.2 *Aamo Transformation*

Aamo [1] has developed a general theory for rejecting disturbances when actuator equipment and sensors are installed several kilometers away. The system is assumed to be on the form:

$$\begin{aligned}
w_t &= -\epsilon_1(x) w_x + c_1(x) v, \\
v_t &= \epsilon_2(x) v_x + c_2(x) w, \\
w(0,t) &= q w(0,t) + CX(t), \\
v(1,t) &= U(t), \\
\dot{X}(t) &= AX(t)
\end{aligned} \tag{12}$$

Here w and v are states, $q \neq 0$, $x \in [0,1]$, $\epsilon_{1,2}(x) \geqslant 0$, $\epsilon_{1,2}(x) \in C^1([0,1])$, $c_{1,2}(x) \in C([0,1])$, U(t) is the control input and v(1,t) is the measurement. A and C are matrices of appropriate dimensions to the disturbance X(t). The disturbance rejection problem of drilling is a specialized problem of this sort. The following coordinate transfor-

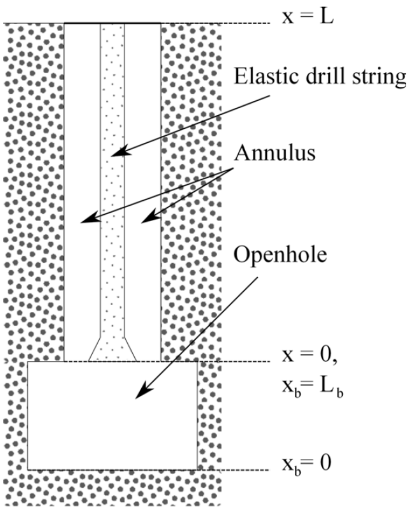


Figure 31: Aarsnes model [2]

mation of the Landet-Model ((7) and (8)) makes it possible to get it on the form in equation (12):

$$\begin{aligned} u(x,t) &= \frac{1}{2}(q(xl,t) + \frac{A}{\sqrt{\beta\rho}}(p(xl,t) - p_{sp} + pglx)) \times e^{\frac{lF}{2\sqrt{\beta\rho}}x} \\ v(x,t) &= \frac{1}{2}(q(xl,t) - \frac{A}{\sqrt{\beta\rho}}(p(xl,t) - p_{sp} + pglx)) \times e^{-\frac{lF}{2\sqrt{\beta\rho}}x} \end{aligned} \tag{13}$$

where

$$\begin{aligned} \epsilon_1(x) &= \epsilon_2(x) = \frac{1}{l}\sqrt{\frac{\beta}{\rho}}, \\ c_1(x) &= -\frac{1}{2}\frac{F}{\rho}e^{\frac{lF}{2\sqrt{\beta\rho}}}, \\ c_2(x) &= -\frac{1}{2}\frac{F}{\rho}e^{-\frac{lF}{2\sqrt{\beta\rho}}}, \\ q &= -1, A = \bar{A}, C = -A\bar{C} \end{aligned} \tag{14}$$

5.2.3 *Aarsnes Model*

Aarsnes [2] divides the modeling problem into three parts: Annulus, drillstring and openhole. This is illustrated in figure 31. Using mass and momentum balances Aarsnes [2] ends up with 6 coupled PDEs. In addition the mud is modeled as a non-Newtonian fluid, which results in a complicated model for the friction loss per control volume. Such a complex model is difficult to simulate and use for control purposes. To make the model available for control purposes Aarsnes [2] makes some simplifications, which leads to a finite dimensional LTI approximation of the model that can be implemented. N control volumes are used to describe the dynamics in the annulus and the drillstring, with N_b control volumes to describe dynamics of the borehole below the drillstring. The resulting system is of order $4N + 2N_b$:

$$\begin{aligned}
\dot{p}_a^j &= \frac{\bar{\beta}_a^j}{l^j}(v_a^{j-1} - v^j), \qquad j = 1, \dots, N \\
\rho \dot{v}_a^j &= \frac{1}{l^j}(p_a^j - p_a^{j+1}) - k_a^j v_a^j + k_d^j v_p^j, \qquad j = 0, \dots, N-1 \\
A_a^N v_N &= q_c, \quad p_a^0 = p_b^{N_b}, \\
\dot{p}_p^j &= \frac{\rho_p^j}{l^j}(v_p^{j-1} - v_b), \\
\rho_p v_p^j &= \frac{1}{l^j}(p_p^j - p_p^{j+1}) - k_p^j(v_p^j - v_a^j) + \frac{K_1}{l^j}(p_a^j - p_a^{j+1}) \qquad j = 0, \dots, N-1 \\
v_p^N &= v_d, \quad p_p^0 = p_b^{N_b} \\
\dot{p}_b^j &= \frac{\rho_b}{l_b^j}(v_b^{j+1} - v_b), \qquad j = 1, \dots, N_b \\
\rho \dot{v}_b^j &= \frac{1}{l_b^j}(p_b^j - p_b^{j+1}) + k_b^j v_b^j, \qquad j = 0, \dots, N_b - 1 \\
v_b^0 &= 0, \quad A_b v_b^{N_b} = A_a v_a^0 + A_p v_p^0
\end{aligned} \tag{15}$$

If there is a change in the annulus cross sectional flow area the following equation should be used instead:

$$\dot{p}_a^j = \frac{1}{A_a^j}(A_a^{j-1} v_a^{j-1} + \Delta A_p v_p^{j-1}) - v_a^j \tag{16}$$

5.2.4 *Mahdianfar Model*

Instead of using a set of control volumes Mahdianfar et al. [18] uses a differential approach. The equation of continuity and the Navier-Stokes equation (White [30, p. 229 and 237]) together with an assumption on the connection between fluid pressure and density ($\rho(t, x) = p_{ref} + \frac{P(t,x) - p_{ref}}{c^2}$) results in the following equations:

$$p_t + v p_x + (k + p) v_x = 0 \tag{17}$$

$$v_t + v v_x + \frac{c^2}{k + P} p_x + g \sin(\gamma(x)) + \hat{F}(x) v^2 = 0 \tag{18}$$

Here k is a constant related to the choice of equation state. The boundary conditions chosen for this set of equations are the flow at the bottom of the well, $x = 0$, and the pressure at the outlet, $x = l$. Resulting in the following two conditions:

$$v(t, 0) = v_0(t) \tag{19}$$

$$p(t, l) = p_l(t) \tag{20}$$

During a connection operation the main pump will be turned off, thus making the movement of the drillstring the only source of flow in the annulus. Therefore a linear model has been developed from these equations around $v(x) \equiv 0$. Zero flow is then used to find a steady pressure profile before a linear model is developed. In addition the assumption of a vertical well ($\sin(\gamma(x)) = 1$) is made. This results in the following equations:

$$\tilde{p}_t + c{v_1}_x = 0 \tag{21}$$

$${v_1}_t + c\tilde{p}_x = 0 \tag{22}$$

Here $u_1 = \frac{k+\bar{P}}{c} v(t, x)$. Riemann invariants are then used to represent the PDEs (21) - (22), using $\tilde{p} = \alpha - \delta$ and $v_1 = \alpha + \delta$:

$$\begin{aligned} \alpha_t + c\alpha_x &= 0 \\ \delta_t - c\delta_x &= 0 \end{aligned} \tag{23}$$

These are a set of the transport equations (Evans [6, p. 18]).

5.2.5 *Disturbance Modeling*

Landet et al. [17], Mahdianfar et al. [18] and Aamo [1] model the disturbance as a set of harmonic waves. In their simulations both make the simplification of using just one single harmonic wave.

$$\begin{aligned} v_d(t) &= a\sin(\omega t + \phi) \\ x_1 &= v_d(t) \\ \dot{x}_1 &= \omega x_2 \\ \dot{x}_2 &= -\omega x_1 \end{aligned} \tag{24}$$

5.3 CONTROL ALGORITHM

Four different control algorithms have been developed. Landet et al. [17] has two control algorithms. First he has a nonlinear output regulation controller in addition to a more simple Linear Internal Model controller. Mahdianfar et al. [18] has developed a controller from simply setting the pressure in the bottom of the well to be at the setpoint. In this paper this will be called the Mahdianfar Controller. By transforming the Landet Model into a general model Aamo [1] uses the control he has developed, called Aamo Controller in this paper.

5.3.1 *Nonlinear Output Regulation Controller*

Landet et al. [17] has used theory from the book *Uniform Output Regulation of Nonlinear Systems* by Pavlov, Wouw and Nijmeijer [23] to develop this control algorithm. The general theory is based upon a system on the form:

$$\begin{aligned} \dot{x} &= f(x,u,\omega) \\ e &= h_r(x,\omega) \\ y &= h_m(x,\omega) \end{aligned} \tag{25}$$

In this equation x is the state vector, u is the control input, e is the regulated input, y is the measured output and ω is the disturbance. Setting $e = p_{bit} - p_{ref}$ while y gets chosen depending on what measured variables are available the resulting controller becomes:

$$u = c(\omega) + K(x - \pi(\omega)) \tag{26}$$

Where K is an appropriate matrix that ensures stability. $c(\omega)$ and $\pi(\omega)$ are solutions to the so-called regulator equations:

$$\begin{aligned} \frac{d}{dt}\pi(\omega) &= f(\pi(\omega), c(\omega), \omega) \\ 0 &= \pi_1(\omega) - p_{ref} \end{aligned} \tag{27}$$

The equations (10) and (11) can be put into the form (25). Then it is possible to solve equation (27). This task grows with the number of harmonic waves used in the disturbance and the number of control volumes used. From Pavlov et al. [23] the boundedness of the solutions and $|p_{bit}(t) - p_{ref}| \to 0$ are guaranteed.

5.3.2 *Linear Internal Model Controller*

A much simpler control strategy is to view the system (11) and (10) as linear. This is true with the exception of the choke equation, (9). The idea is to find a so-called disturbance-generating polynomial and include it in the closed loop error feedback controller. This will make the disturbance vanish asymptotically. A disturbance polynomial for the disturbance is: $\Gamma = s^2 + \omega^2$. Combined with a proportional integral controller:

$$C(s) = Q_0 + \frac{Q_1}{s} + \frac{Q_2 s}{s^2 + \omega^2} \tag{28}$$

Here, Q_i are tuning constants and $C(s)$ is the transfer function from the error signal, e(t), to the signal, $\nu(t)$. The controller then becomes:

$$u(t) = G^{-1}(-\nu(t)) \tag{29}$$

The applicability of this controller is that it does not demand a lot of information and its is simple to implement.

5.3.3 *Mahdianfar Controller*

Mahdianfar et al. [18] simply assumes that all states are known and uses the equations to derive a controller. In the Mahdianfar model $\tilde{p}(t,x)$ is a deviation variable and the control objective is then to get $\tilde{p}(t,0) = 0$ (zero pressure deviation from the setpoint at the bottom of the well). This leads to the following controller:

$$\delta(t,l) = \frac{1}{2}[\cos(\frac{\omega l}{c})x_1(t) + \sin(\frac{\omega l}{c})x_2(t)] \tag{30}$$

Since this controller assumes that all states are known an observer is needed to implement the controller. The top-side measurements can be represented using the notation from (23):

$$Y(t) = \alpha(t,l) + \delta(t - 2\frac{l}{c}, l) \tag{31}$$

The system can then be written on the form:

$$\begin{aligned} \dot{X} &= AX \\ Y(t) &= CX(t-D) \end{aligned} \tag{32}$$

This is an LTI system with sensor delay. A standard infinite-dimensional observer can then be used:

$$\begin{aligned} \dot{\hat{X}} &= A\hat{X} + e^{AD}L(Y - \hat{z}(0)) \\ \hat{z}_t &= \hat{z}_x + Ce^{Ax}L(Y - \hat{z}(0)) \qquad x \in (0,D) \\ \hat{z}(D) &= C\hat{X} \end{aligned} \tag{33}$$

In order to implement the infinite-dimensional observer given in equation (33) it is transformed into a transfer function. This transfer function is then reduced to a first- and a second-order approximation using Laguerre based model reduction. A second-order reduction is found to match the infinite-dimensional transfer function very well.

5.3.4 *Aamo Controller*

Utilizing a backstepping algorithm developed by Vazquez et al. [29] together with an observer, Aamo [1] comes with the following control law:

$$p_l(t) = \frac{\sqrt{\beta\rho}}{A}q_l(t) - 2\frac{\sqrt{\beta\rho}}{A}U(t)e^{\frac{lF}{2\sqrt{\beta\rho}}} + p_{sp} - \rho g l \tag{34}$$

where

$$U(t) = \int_0^1 K^{\nu u}(1,\eta)\hat{u}(\hat{u}(\eta,t)d\eta + \int_0^1 K^{\nu\nu}(1,\eta)\hat{\nu}(\hat{u}(\eta,t)d\eta + K\hat{X}(t) \tag{35}$$

The estimates for $u(x,t), v(x,t)$ and $X(t)$ comes from the following observer:

$$\begin{aligned}
\hat{u}_t &= \epsilon_1(x)\hat{u}_x + c_1(x)\hat{v} + p_1(x)(u(1,t) - \hat{u}(1,t)) \\
\hat{v}_t &= \epsilon_2(x)\hat{v}_x + c_2(x)\hat{u} + p_2(x)(u(1,t) - \hat{u}(1,t)) \\
\hat{u}(0,t) &= q\hat{v}(0,t) + C\hat{X}(t) \\
\hat{v}(1,t) &= U(t) \\
\dot{\hat{X}} &= A\hat{X} + e^{Ad_\alpha}L(u(1,t) - \hat{u}(1,t)) \qquad (36)
\end{aligned}$$

where

$$\begin{aligned}
p_1(x) &= Ce^{Ad_\alpha(x)}L - \epsilon_1(1)P^{uu}(x,1) - \int_x^1 P^{uu}(x,\xi)Ce^{Ah_\alpha(\xi)}Ld\xi \\
p_2(x) &= -\epsilon_1(1)P^{vu}(x,1) - \int_x^1 P^{vu}(x,\xi)Ce^{Ah_\alpha(\xi)}Ld\xi \qquad (37)
\end{aligned}$$

Finally the kernel will be given by the following set of equation:

$$\begin{aligned}
\epsilon_1(x)K_x^{uu}(x,\xi) + \epsilon_1(\eta)K_\xi^{uu}(x,\xi) &= -\epsilon_1^{'}(\xi)K^{uu}(x,\xi) - c_2(\xi)K^{uv}(x,\xi) \\
\epsilon_1(x)K_x^{uv}(x,\xi) - \epsilon_2(\eta)K_\xi^{uv}(x,\xi) &= -\epsilon_2^{'}(\xi)K^{uv}(x,\xi) - c_1(\xi)K^{uu}(x,\xi) \\
\epsilon_2(x)K_x^{vu}(x,\xi) - \epsilon_1(\eta)K_\xi^{vu}(x,\xi) &= -\epsilon_1^{'}(\xi)K^{vu}(x,\xi) + c_2(\xi)K^{vv}(x,\xi) \\
\epsilon_2(x)K_x^{vv}(x,\xi) + \epsilon_2(\eta)K_\xi^{vv}(x,\xi) &= -\epsilon_2^{'}(\xi)K^{vv}(x,\xi) + c_1(\xi)K^{vu}(x,\xi)
\end{aligned} \qquad (38)$$

on $\mathcal{T} = \{(x,\xi) : 0 \leqslant \xi \leqslant x \leqslant 1\}$ with boundary conditions

$$\begin{aligned}
K^{uu}(x,0) &= \frac{\epsilon_2(0)}{q\epsilon_1(0)}K^{uv}(x,0) \\
K^{uv}(x,x) &= \frac{c_1(x)}{\epsilon_1(x) + \epsilon_2(x)} \\
K^{vv}(x,0) &= \frac{q\epsilon_1(0)}{\epsilon_2(0)}K^{vu}(x,0) \\
K^{vu}(x,x) &= \frac{c_2(x)}{\epsilon_1(x) + \epsilon_2(x)} \qquad (39)
\end{aligned}$$

6 MODEL IDENTIFICATION

6.1 MOTIVATION

In Chapter 5 the different models for modeling a well during connections were presented. Similar for all the models were that after doing some assumptions and simplifications they ended up with linear equations. The exception was the choke equation. These results motivates to examine whether a linear model could be identified from the well data. A linear model of the lab has the advantage that it can easily be used to design a controller compared to a more complex nonlinear model.

6.2 NOTATION

The following notation will be used for this and the following chapter:

p_c is the pressure at the choke inlet, in the lab C2

$p_{c,ref}$ is the reference pressure at the choke inlet

p_0 is the pressure at the choke outlet , in the lab C1

p_{dh} pressure downhole, in the lab P2

$p_{dh,ref}$ reference pressure downhole

v_p velocity of piston

$v_{p,ref}$ reference velocity of piston

q_w flow into (and out of) the well, in the lab FT1

q_c flow through the choke, in the lab FT3

6.3 SELECTING INPUT AND OUTPUT VARIABLES

The lab has two inputs: the piston velocity and the choke opening. In a real well it might also be possible to control the backpressure pump, but this is not the case in the lab. The objective of the controller is to control the bottomhole pressure and thus it will be the output of the identification.

The advantage with designing a choke controller (Chapter 4) is that the nonlinearity at the choke can be abstracted away by replacing the choke opening by the choke pressure as input to the model. Then the objective of the controller will be to select a pressure at the choke that will remove a given disturbance. The goal will therefore be to have a model similar to the one shown in Figure 32

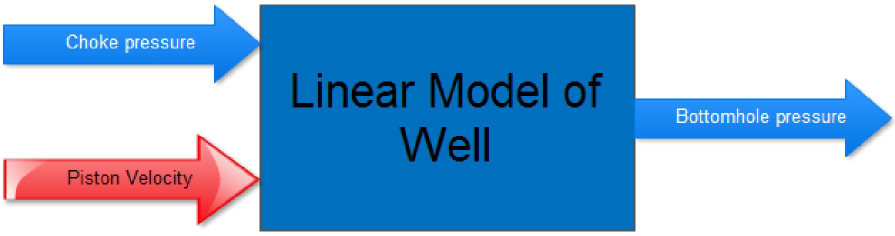


Figure 32: Black box model

Multiple approaches were attempted for identification of an accurate model. It was quickly discovered that the piston reference velocity gave more accurate models than the measurement of the piston velocity, and it was thus used.

6.3.1 *Setup 1*

The simplest approach of input variables is to use the one shown in Figure 32:
Input: $u = [p_c, v_{p,ref}]$
Output: $y = [p_{dh}]$

6.3.2 *Setup 2*

It could prove difficult to identify a model directly as picked in setup number two. From the Mahdianfar model in Section 5.2.4 the inputs are the flow into the well, the pressure at the choke inlet and disturbance. Thus resulting in the following choice of variables:
Input: $u = [p_c, q_w, v_{p,ref}]$
Output: $y = [p_{dh}]$

6.3.3 *Setup 3*

In the third setup it was attempted to exploit some knowledge of the system. The challenge of controlling the bottomhole pressure is the time delay from the top of the well to the bottom. This delay can be estimated before running the identification, and later be approximated by a Padé approximation to get a proper model. It is of course possible to do this indirectly though the identification, but making the assumption may improve the accuracy of the result.

The same assumption could have been done for the flow. However from the measured data a change in pressure at the choke inlet was actually measured at the bottomhole before the flow was affected, see Section 6.4.1.
Input: $u = [p_{c,\text{with delay}}, q_w, v_{p,ref}]$
Output:$y = [\ p_{dh}]$

6.3.4 *Setup 4*

In order to use the setup 2 and 3 (Section 6.3.2 and Section 6.3.3) the flow q_w must be handled. This is not an independent input into the system, but it depends on the choke opening and the piston velocity. The MPC controller designed in Chapter 7 needs to use the model to predict the future in order to decide the current input. In the first attempt to use this controller the flow was predicted in the same way as the disturbance, $v_{p,ref}$, with a harmonic wave (see Section 6.10). The results were disastrous with the prediction being worthless. The reason was that the flow depended very much on the choke opening and as soon as it changed the prediction was no longer valid.

A model of the flow using the available data was needed. From Kaasa et al. [13] there is the following connection between the flow, the bottomhole pressure and the pressure at the choke, equation (25):

$$M\frac{dq_w}{dt} = p_{dh} - p_c - F(q_w, \mu) + G(\rho) \tag{40}$$

Here M is a parameter depending on the cross-sectional area of the pipe and the density function. $F(q_w,\mu)$ is the friction in the well and $G(\rho)$ is the steady state difference between the downhole pressure and the choke pressure. By assuming a linear friction term: $F(q_w,\mu)$ = μq_q, this model was used as an inspiration for a setup to estimate the flow into of the well
Input: u = $[p_{dh} - p_c - G(\rho)]$
Output: y = $[q_w]$

6.3.5 *Setup 5*

A combination of setup 1 and setup 3 was also attempted:
Input: u = $[p_{c,\text{with delay}}, v_{p,ref}]$
Output: y = $[p_{dh}]$

6.3.6 *Setup 6*

In addition to attempting to estimate the flow, q_w, from the equation in setup 4, a simple setup was attempted:
Input: u = $[p_c \; p_c]$
Output: y = $[q_w]$

6.4 CALCULATING ASSUMPTIONS

6.4.1 *Time Delay*

In order to calculate the time delay from the choke inlet pressure, p_c, to the bottomhole pressure, p_{dh}, the following procedure was used:

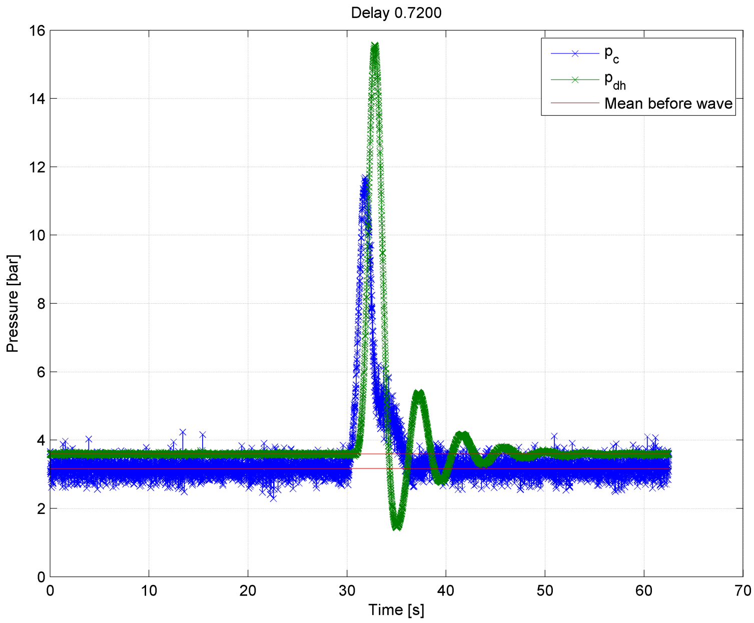


Figure 33: Time delay from choke inlet to bottomhole

1. The choke was kept at constant opening for about 30 seconds.
2. The choke was set to run through a half or full sine wave.
3. A mean value for the pressures was calculated for the first 30 seconds.
4. The pressure was defined to be changed when 50 samples on the same side of the mean value had occurred
5. The time delay was then calculated from taking the time from pressure change at the choke inlet to the pressure change at the bottomhole pressure.

See Appendix C for MATLAB code. Multiple such tests were run and Figure 33 shows a typical result. An average for all the tests were calculated and found to be *0.72 seconds*.

A similar procedure was attempted to find a time delay from the flow into the well to the bottomhole pressure. However it was discovered that a change in the choke opening was discovered quicker downhole than at the pipe flow inlet, as can be seen in Figure 34. It was thus concluded that including a time delay from the flow to the bottomhole pressure will not improve the model.

6.4.2 *Steady state difference between* p_{dh} *and* p_c

For the fourth setup (Section 6.3.4) the steady state difference between the bottomhole and the choke pressure is needed. To find this differ-

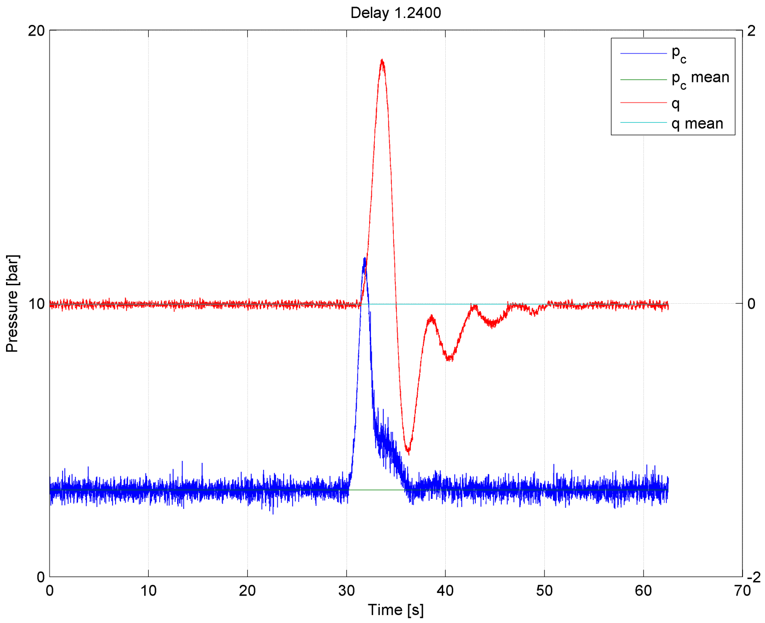


Figure 34: Time delay from choke inlet to copper flow inlet

ence the same dataset as used in Section 4.4.2 shown in Figure 15 was used. In this dataset the choke opening was very slowly ramped up and down. Since the choke is moving so slowly the well was considered to be in steady state through the whole test. The steady state difference was thus found by taking the mean of $p_{dh} - p_c$. Seeing the plot (not included in this report) the difference seemed to be constant through the whole test. The value calculated: $G(\rho) = 0.3771$.

6.5 ALGORITHMS

Two different subspace identification algorithms were used to try to find as accurate model as possible for each of the six setups used. The first algorithm named DSR was developed by David Di Ruscio [26]. The second algorithm, here be referred to as the Overschee algorithm, were developed by Peter Van Overschee and Bart De Moor [22]. To go into detail about the theory and implementation of these algorithms are beyond the necessity of this report, and interested readers are referred to their respective citations. Implementation of the algorithms can be found in Appendix C.

Both algorithms combine deterministic and stochastic properties for identifying a system. The resulting system identified is a discrete system on the form:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + K(y - \hat{y}) \\ \hat{y} &= Cx_k + Du_k \end{aligned} \tag{41}$$

6.6 IDENTIFICATION OF SYSTEM MODEL

To identify a model of the lab several different experiments were used. The experiments consisted of having the piston and the choke controller follow two sets of different sine signals.

In addition to input and output data, the identification algorithms need to specify a maximum order of the system. This maximum order for the system is then used to construct a block Hankel matrix, which is then used to construct something called an extended observability matrix. A single value decomposition of this matrix is done and the singular values are plotted. The user defines the order of the system based upon the number of singular values different from zero (this is not a trivial task). The accuracy of the model will then both depend on the maximum order set for the identification and the order chosen by the user.

Both Ruscio [26] and Overschee and Moor [22] algorithms have the option of predefining the order of the system instead of manually select an order based upon the plot of the singular values for the extended observability matrix.

Summing up there are a lot of different choices that can be done when identifying a system on the form in Equation 41. First six different choices of input and output variables have been given (Section 6.3). Then five different experiments were selected to be suitable for identification. In addition a maximum order and an order for the system had to be chosen. The model identified depends on the maximum order set. For example if the maximum order is set to 10 and the order of the system is set to 4, the result will be different than setting the maximum order to 20 and the order of the system to 4. Finally there are two different algorithms that can be used for the identification.

In order to find the best possible model a script was written to run through all the different possibilities with a maximum order up to 75. The number of identification runs then becomes: Setups x Identifications sets x Number of algorithms x Maximum Order! = 6 x 5 x 2 x 75! = 171 000 runs. It took about 2 days to run the script.

To compare the different models identified a set of twelve different experiments was used, called a test set. Each model was used to estimate the output from the input of each of the experiments in the test set. Since the model is going to be used to predict into the future, the $K(y - \hat{y})$ term from Equation 41 was removed. In other words the output was estimated solely based upon the input with no correction term. The average absolute error was calculated between the test set output and the prediction. After the whole test set was run for a single model, the average absolute error for the whole test set was calculated and stored. Table 7 contains the different experiments used for identification and testing of the identified model. As can be

Name	Piston [cm]				Choke [bar]					id	test
	A1	p1	A2	p2	A1	p1	A2	p2	B		
test6_020513	10	4	2	4.5	2	3	0.25	3.5	4	x	x
test8_2_020513	5	3	2.5	4	0.5	3.5	0.75	4.5	4	x	x
test5_250413	15	5	17	10	1	3	2	12	4	x	x
test6_250413	7.5	6.5	16	4.5	1	3	2.5	20	4	x	x
test8_250413	7.5	6.5	16	4.5	1	3	0.5	17	4	x	x
test1_020513	20	9	10	12	1	10	2	11	5	-	x
test2_020513	30	10.5	5	12	2	8	0.5	11.5	4	-	x
test5_020513	20	4	5	3.25	0.5	3	0.75	3.75	5	-	x
test7_020513	10	3	15	3.25	0.75	4	1	3.5	4	-	x
test4_250413	40	8.5	-	-	1.4	8.5	-	-	3.7	-	x
test7_250413	15	4	-	-	2	3	-	-	4	-	x
test2_250413	15	10	5	20	2	3	1	10	4	-	x

In the table the following abbreviations have been used: A1 = amplitude 1, p1 = period 1, B = bias, id = identification. In addition the piston bias was always = 40 cm. For example for *test6_020513* the piston followed $10\sin(2\pi\frac{1}{10}t) + 2\sin(2\pi\frac{1}{4.5}t) + 40$ and the choke followed $2\sin(2\pi\frac{1}{3}t) + 0.25\sin(2\pi\frac{1}{3.5}t) + 4$.

Table 7: Experiments used to identify lab model

seen in the table all the identification experiments were also used in the test set. When using a set both as identification and to test afterward the result will naturally be very accurate, but since this was the same for all the identification sets used it was not considered to be a problem.

6.7 RESULTS

In order to be able to easily consider the possible trade-off between model order and accuracy, the most accurate model was stored for each order and setup. Then all average errors above 1 bar were considered to be useless and thus removed. This makes it much easier to study the results, which are plotted in Figure 35, Figure 36 and Figure 37. Here for each setup and choice of algorithm the most accurate model of a specific order is plotted.

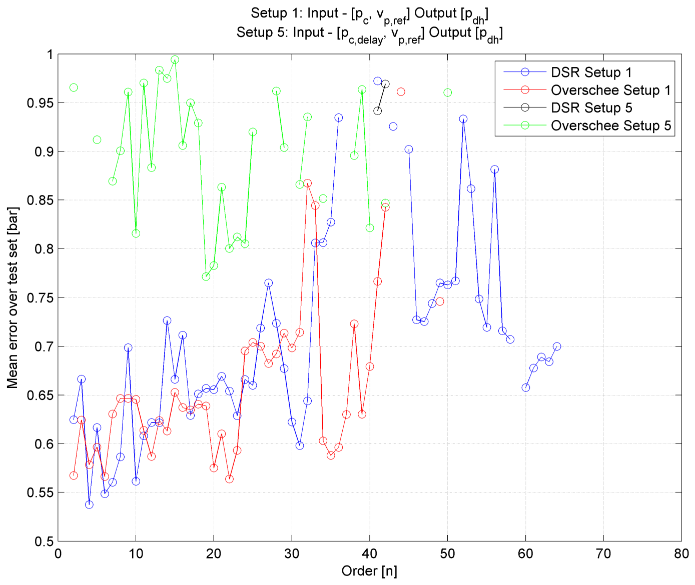


Figure 35: Best Models found for Setup 1 and 5

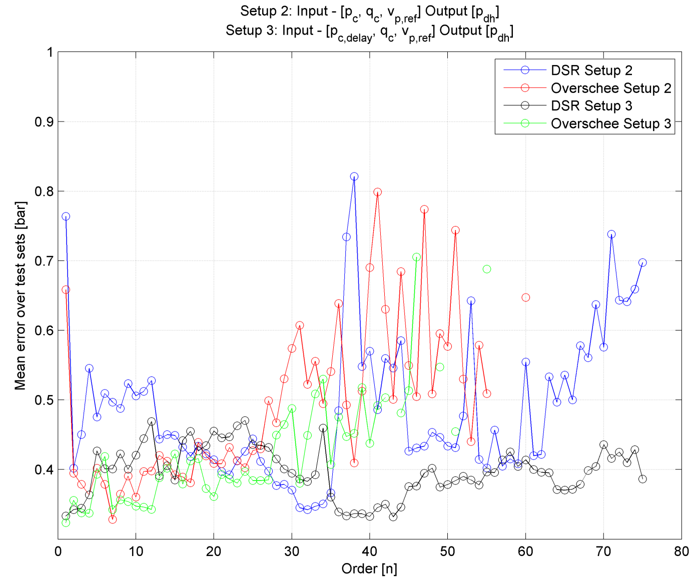


Figure 36: Best Models found for Setup 2 and 3

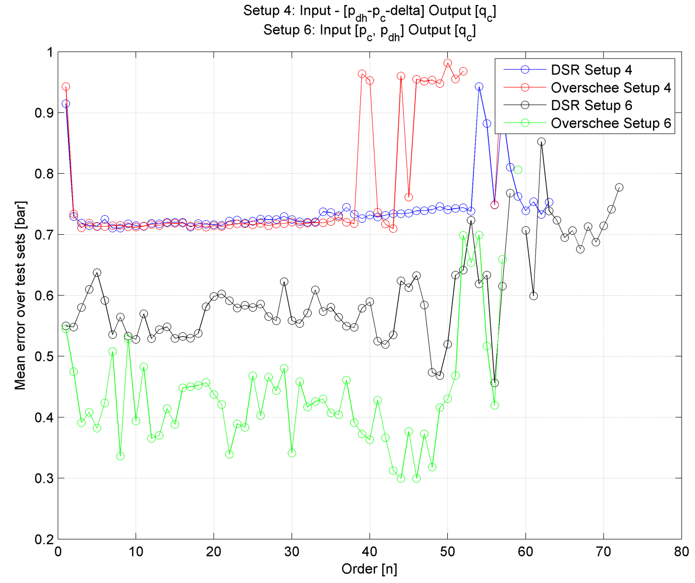


Figure 37: Best Models found for Setup 4 and 6

6.8 DISCUSSION OF RESULTS

During initial testing it was thought too difficult to find an accurate model for setup 1. This is the reason why all the other setups were thought up. It was then a surprise to discover that the simplest setup with no assumption yielded the best result. As can be seen from the graph the DSR algorithm managed to identify a fourth order model that has an average absolute error at just above 0.5 bar for over the whole test set.

As can be seen in the resulting graphs there are more accurate results found for setup 2, 3, 4 and 6. But in order to use one such model, the model from setup 2 or 3 has to be combined with a model from 4 or 6. As can be seen from the graph the Overschee algorithm with setup 3 manages to find a model of first order with an accuracy close to 0.32 in average error over the test set. However, as mentioned, it will have to be combined with for example the eighth order model identified by Overschee for setup 6 with an accuracy at about 0.37. Adding these two together the accuracy will be at approximately 0.7 bar, which is more than the one identified by the DSR algorithm for setup 1.

The fourth order model identified was on the form as Equation 41 with the following matrices:

$$A = \begin{bmatrix} 0.9925 & -0.08565 & -0.01556 & 0.115 \\ 0.002906 & 1.002 & -0.1638 & -0.02512 \\ -4.017e-05 & 0.0001124 & 0.9828 & 0.2725 \\ 3.289e-05 & -1.936e-05 & -0.03967 & 0.9808 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.02297 & 0.2996 \\ 0.02016 & 0.02761 \\ -0.0003411 & 0.02834 \\ 0.0005492 & 0.009049 \end{bmatrix} C = \begin{bmatrix} -0.1772 & -0.2597 & -0.3053 & 0.3316 \end{bmatrix}$$

$$D = \begin{bmatrix} 0.00431 & -0.4688 \end{bmatrix} K = \begin{bmatrix} -6.3912 \\ 3.0082 \\ 0.1989 \\ 0.0455 \end{bmatrix} \tag{42}$$

The algorithm used was DSR with maximum order of 40 and the identification set *test8_2_020513.mat* and the selection of input-output variables from setup 1 (Section 6.3.1). The average error over the whole test set was 0.5375 bar. This model was used when making the MPC-controller Section 7.2. A demonstration of the identified model is given in Figure 38 and Figure 39. In the first figure the model is used without the Kalman gain while in the second it is included.

The LQG-controller designed in Section 7.1.2 was developed before the complete identification script was run. The model used in that

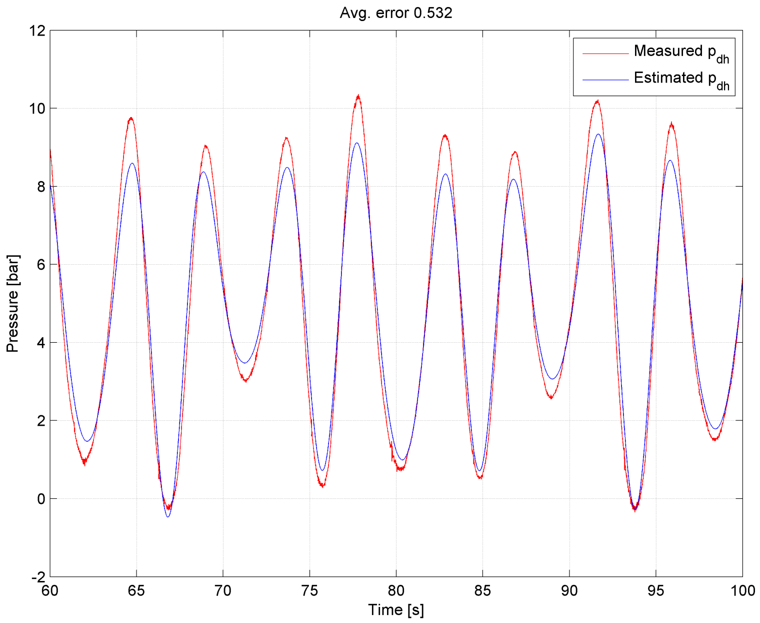


Figure 38: Identified Model Predicting for *test8_250413.mat*

controller was a model identified manually. The setup used was setup 3 with the DSR algorithm. The identification set used was *test8_250413.mat* with a maximum order of 4 and order 4. Resulting in the following identified set:

$$A = \begin{bmatrix} 0.9898 & 0.7031 & -0.2608 & -0.3765 \\ -0.0002 & 0.7061 & 0.6978 & -0.4510 \\ -0.0001 & -0.2033 & -0.3070 & -0.9215 \\ 0.0001 & -0.0885 & 0.3079 & -0.4263 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0140 & -0.0251 & 0.6220 \\ -0.0047 & -0.0052 & -0.7911 \\ -0.0007 & 0.0037 & -0.6116 \\ -0.0007 & -0.0077 & -1.2229 \end{bmatrix} C = \begin{bmatrix} -0.5076 & 0.6848 & -0.5060 & -0.1318 \end{bmatrix}$$

$$D = \begin{bmatrix} 0.0028 & 0.0139 & 0.2103 \end{bmatrix} K = \begin{bmatrix} -0.8576 \\ -0.4267 \\ 0.1163 \\ 0.1404 \end{bmatrix} \tag{43}$$

6.9 PADÉ APPROXIMATION

In order to use the model identified for the LQG controller an approximation had to be done to get a proper linear model. The MATLAB

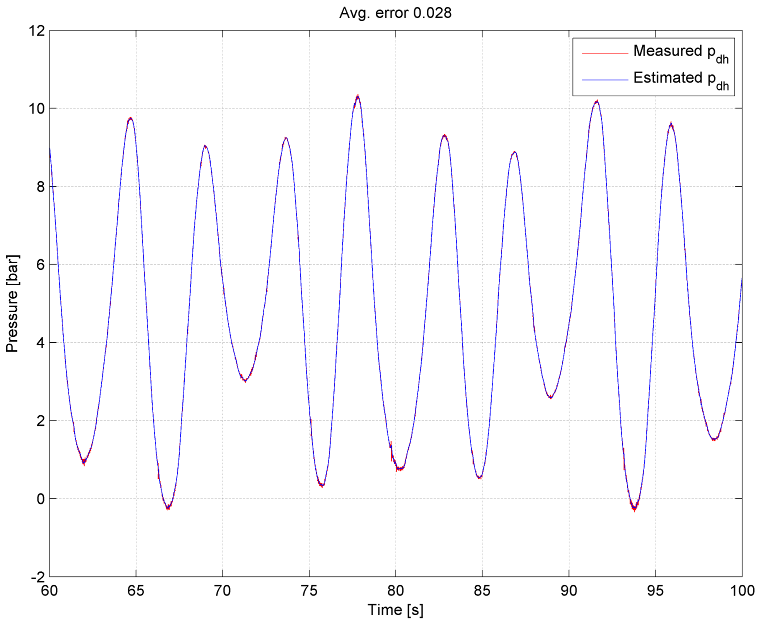


Figure 39: Identified Model Predicting for *test8_250413.mat* with Kalman Gain

functions delayss() and pade() where used to include the Padé approximation in the model, see Appendix C. The Padé approximation has an almost identical response to the model identified with time delay, due to the operating range of the frequencies of the input to the model.

6.10 MODELING DISTURBANCE

The disturbance was modeled as a single harmonic wave. This is the same model described in Section 5.2.5 used by Landet et al. [17], Mahdianfar et al. [18] and Aamo [1]. This requires the frequency of the wave to be known.

$$\dot{x} = A_d x \qquad y = C_d x \tag{44}$$

where

$$A_d = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} C_d = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

The model of the disturbance was implemented with an observer as shown in Figure 40. Pole placement was used to place the poles of $(A_d - LC_d)$ in the left half plane.

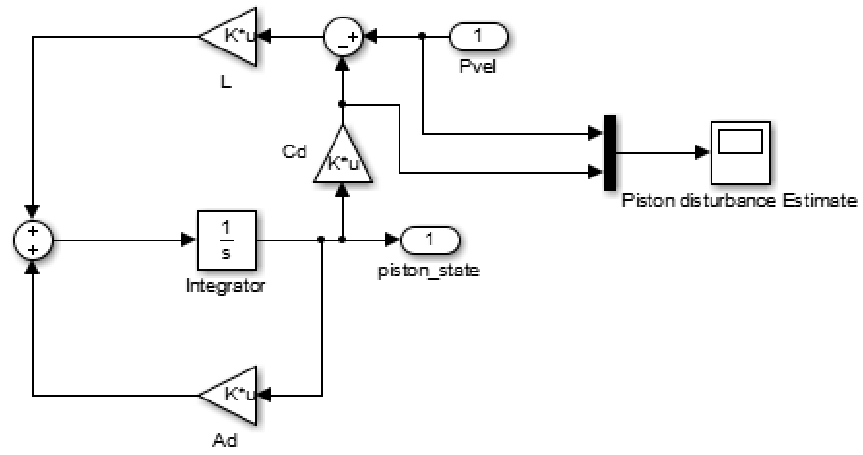


Figure 40: Implementation of Disturbance Estimator

7 BOTTOMHOLE CONTROLLER

First a PI-controller was tuned to try to suppress the heave disturbance (will also be referred to as the piston disturbance or just the disturbance in this chapter). This was done in order to have a starting point. Then based upon a fourth order model with setup 3 was used to design a LQG-controller. The LQG-controller was chosen due to its simplicity to implement, making it very time efficient to test. The experience with the LQG-controller motivated the use of an MPC-controller.

This chapter is built up as follows. First the PI-controller and the LQG-controller implementation and testing is documented. Then the remaining of the chapter is dedicated to the implementation of the MPC and the testing of it.

7.1 INITIAL ATTEMPTS

7.1.1 *PI-Controller*

A PI-controller was attempted first for suppressing disturbance from the piston movement in the lab. For the implementation a PID block from the SIMULINK library *Continuous* was used. It was tuned online with the piston following a sine wave. The input was $p_{dh,ref} - p_{dh}$ and the output p_c. The parameters used in the test can be found in Table 8. Three tests were run on the controller. All consisted of the piston following a single sine wave with a given period and compare the controller to constant choke angle. The target was to follow a constant reference of 4 bars. All the numbers following are deviations from this value. For a period of 10 the controller was able to suppress the disturbance from +1.3 and -1.7 to +0.9 and -0.8 and for 7 second period from +3.4 and -3.1 to ±2. However for a period of

Proportional gain	P = 0.5
Integral gain	I = 0.3
Initial condition integrator	I_0 = 2
Upper saturation limit	= 10
Lower saturation limit	= 0.5
Anti-Windup with backcalculation	K_b = 10

Table 8: Parameters for PI bottomhole controller

5 seconds, the controller worsened the performance compared to a constant choke angle.

The results were as expected. From Section 6.4.1 the delay from p_c to p_{dh} was found to be 0.72 seconds. This means that the controller has to have some kind of prediction in order to suppress disturbances. The PI-controller has no prediction and will only be able to suppress disturbances when the time delay is short compared to the period of the disturbance. It is therefore natural that when the period of the disturbance decreases there will be a point where the controller introduces more pressure fluctuations than it is able to suppress.

7.1.2 *LQG- controller*

The first model of the lab to be identified was one with input = $[p_{c,delayed}\ q_w\ v_{p,ref}]$ and output = $[p_{dh}]$, the setup described in Section 6.3.3. By considering both the flow into the well, q_w, and the piston movement, $v_{p,ref}$, as disturbances an LQG - controller was developed.

The motivation for developing an LQG controller was the simplicity of implementation due to the easy available MATLAB functions. Since the LQG controller does not have any disturbance prediction like the PI-controller, it was not expected to handle disturbances with a small period compared to the time delay from the choke to the bottom of the well.

7.1.2.1 *Implementation*

An LQG controller is the combination of a Kalman filter and an LQR controller. It is assumed to have a linear model of the system of the form:

$$\begin{aligned} \dot{x} &= Ax + Bu + w_d \\ y &= Cx + Du + w_n \end{aligned} \tag{45}$$

with w_d and w_n being the process noise and the measurement noise with covariances $Ew_d(t)w_d(t)^T = W\delta(t-\tau)$ and $Ew_n(t)w_n(t)^T = V\delta(t-\tau)$ see Skogestad and Postlethwaite [28, p. 344-345]. The LQR controller assumes a controller on the form $u = -Kx$ and selects K from minimizing some objective function. In MATLAB this objective function is:

$$J = \int x^T Qx + u^T Ru + 2x^T Nudt \tag{46}$$

The target for the controller is to drive all the states off the system to zero. This is not what is desired for the suppression of the heave motion problem. Thus some modifications are needed before applying the MATLAB functions lqr() and kalman().

First the B and D matrices identified in Chapter 6 were separated into a disturbance term and an actuator term:

$$\begin{aligned}\dot{x} &= Ax + Bu + Ed \\ y &= Cx + Du + Fd\end{aligned} \tag{47}$$

Then the controller was assumed to be on the form $u = -K(x - x^*) + u^*$. Here x^* and u^* are the state and input of the system when it has reached its desired state. They were calculated by assuming that the system reaches a steady state, $\dot{x} = 0$. This is not an entirely valid assumption since the disturbance is expected to vary and thus the desired state and input will also have to vary in order to keep the output constant. The desired reference value is noted as r and by doing some simple algebraic manipulation, u^* and x^* were calculated to be:

$$\begin{bmatrix} x^* \\ u^* \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} -E \\ -F \end{bmatrix} d + \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix} r \tag{48}$$

Setting this choice of x^*, u^* and u into (47) yields the following closed loop system:

$$\dot{x} = (A - BK)x - (A - BK)x^* \tag{49}$$

This system will be stable as long as A-BK has negative eigenvalues. The K matrix was calculated using the MATLAB function lqr() with the weights Q = 11, R = 10 and N = 0 (46).

The Kalman filter was calculated using the system on the form Equation 45 with W = 0.1 and V = 0.05 and the MATLAB function kalman().

See Appendix C for a list of files containing the implementation of the controller.

7.1.2.2 *Simulation*

The controller was implemented with the model (47) and simulated in SIMULINK. Figure 41 shows the simulation result using a slowly varying disturbance. The LQG controller is successful in suppressing the disturbance from the heave motion.

7.1.2.3 *Test in Lab*

The LQG controller was tested in lab. The result turned out to be disastrous with experiencing vacuum in the copper pipe as a result of the controller being unstable.

The huge difference between the lab and the simulation is that in the simulation the choke controller is not taken into account. The LQG controller sets the reference signal for the choke controller. The

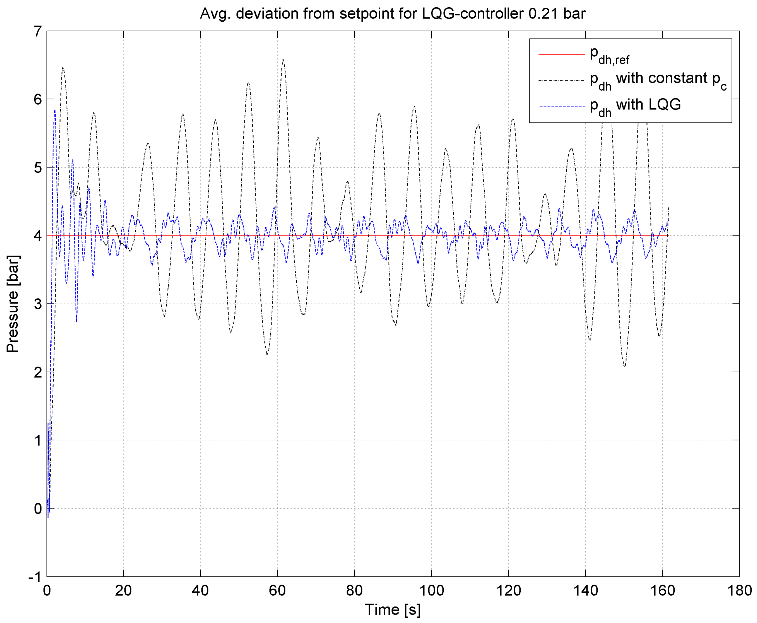


Figure 41: LQG-Controller SIMULINK simulation. In the simulation the choke is not modeled, thus instead of comparing to constant choke angle a comparison is made to constant choke pressure.

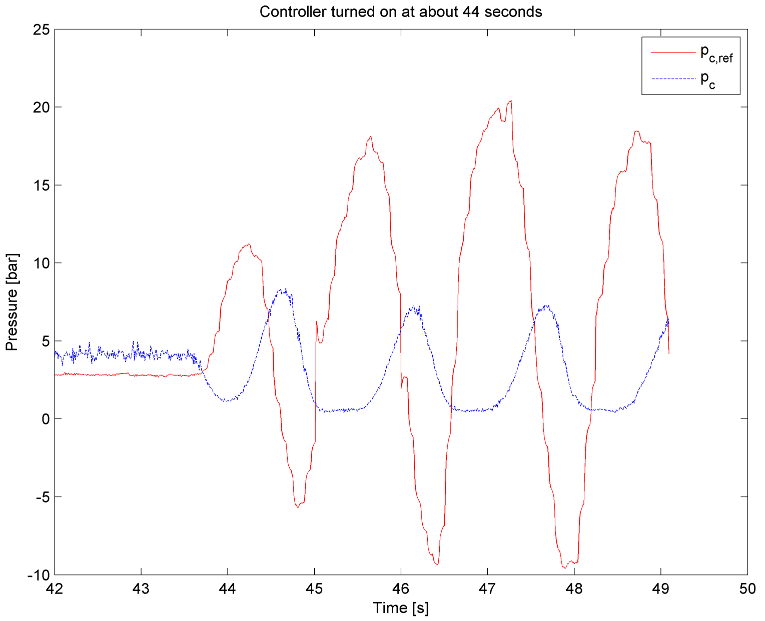


Figure 42

resulting tracking signal for the choke pressure and the pressure can be seen in Figure 42.

When looking at the figure it is clear that the choke controller is unable to follow the reference set by the LQG controller. The LQG controller sets an impossible reference by choosing reference pressure less than zeros in addition to having a period of about 1.5 seconds, which is about twice as fast as what the choke controller is designed for.

One possible solution to the problem might be to put a higher weight on the input usage in the objective function by increasing R. In addition a saturation port needs to be used to avoid getting pressures outside the operating range for the choke controller. However adding a saturation port will make the objective function suboptimal in addition to the fact that the LQG controller does not predict the disturbance. Thus instead of improving the LQG-controller it motivated the design of the controller described in the following section.

7.2 MPC

Motivated by the limitation of the LQG controller in Section 7.1.2 the natural choice seems to be a *Model Predictive Controller* (MPC). An MPC-controller has several advantages over an LQG controller:

1. It can handle constraints on input and outputs.
2. It can utilize knowledge of a disturbance to implement a feedforward mechanism.

However a disadvantage of an MPC controller is that it is computationally demanding to implement. In the IPT-Heave lab the time between samples is 0.01 seconds and the computer will probably not be able to solve a quadratic problem (QP-problem) this fast. This was solved by having a slower sampling rate for the MPC controller.

7.2.1 *QP-Problem Setup*

In the following subsection the setup of the MPC controller is described. It is mainly based upon chapter 6 in Hovd [12] with a few modifications.

The idea behind an MPC controller is to solve a QP-problem for each time sample and use the first output of the solution. A common form for a QP-problem is:

$$\begin{aligned} \min_{\nu} 0.5\nu^T \tilde{H}\nu + c^T\nu \\ \text{subject to} \\ L\nu \leqslant b \end{aligned} \tag{50}$$

The QP-solver used in the MPC implemented in this thesis uses an active-set method for solving the QP-problem, see Nocedal and Wright [21] for more information about the active-set algorithm.

From Section 6.8 the following model was derived:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Ed_k \\ y_k &= Cx + Du_k + Fd_k \end{aligned} \tag{51}$$

In this model y = p_{dh}, $u = p_c$ and $d = v_{p,ref}$. A few modifications of this model were made in order to implement the MPC controller. First D and F were set equal to zero. This increases the error of the model, but it greatly simplifies the calculations that have to be done on-line for the controller, and thus saving a lot of computational power. In addition an extra time delay was introduced from p_c. This was done in order to model the choke controller dynamics described in Section 4.5. The resulting model then becomes:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_{k-l} + Ed_k \\ y_k &= Cx \end{aligned} \tag{52}$$

where $u_{k-l} = 0$ for l < k. The control objective of the MPC-controller is to try to make the bottomhole pressure follow a constant reference. In addition it should be possible to set some constraints, both on the controller input and how fast the controller input can change. In the objective function it would be an advantage to be able to weight between the competing objectives of following the bottomhole reference and the input usage. To accommodate for these demands the following objective function was chosen:

$$\begin{aligned} \min_u f(x,u) = &\sum_{i=0}^{n-1} \{(x_i - x_{ref,i})^T C^T Q C(x_i - x_{ref,i})\} + \\ &\sum_{i=l}^{n-1} \{(\Delta u_i - \Delta u_{ref,i})^T P(\Delta u_i - \Delta u_{ref,i})\} + \\ &(x_n - x_{ref,n})^T C^T S C(x_n - x_{ref,n}) \end{aligned} \tag{53}$$

subjects to the following constraints:

$$\begin{aligned} x_0 &= \text{given} \\ \Delta U_L \leqslant \Delta u_i &\leqslant \Delta U_U \qquad \text{for } 0 \leqslant i \leqslant n \\ Y_L \leqslant Cx_i &\leqslant Y_U \qquad \text{for } 0 \leqslant i \leqslant n \end{aligned} \tag{54}$$

Here n is the prediction horizon and $\Delta u_k = u_k - u_{k-1}$. In order to implement this objective function some changes had to be made in the model in Equation 52:

$$\begin{aligned} \tilde{x}_{k+1} &= \tilde{A}\tilde{x}_k + \tilde{B}\Delta u_{k-l} + \tilde{E}d_k \\ \tilde{y}_k &= \tilde{C}\tilde{x}_k \end{aligned} \tag{55}$$

where

$$\tilde{A} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} B \\ I \end{bmatrix}, \quad \tilde{E} = \begin{bmatrix} E \\ 0 \end{bmatrix}$$

$$\tilde{C} = \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}, \quad \tilde{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ u_{k-l} \end{bmatrix}, \quad \tilde{y}_k = \begin{bmatrix} y_k \\ u_{k-l} \end{bmatrix}$$

These equations can now be combined into the following QP-problem:

$$\min_{\chi,\upsilon} \begin{bmatrix} \chi^T & \upsilon^T \end{bmatrix} \begin{bmatrix} \hat{Q} & 0 \\ 0 & \hat{P} \end{bmatrix} \begin{bmatrix} \chi \\ \upsilon \end{bmatrix} \tag{56}$$

subject to

$$(\chi + x_{ref}) = \hat{A}(\chi + x_{ref}) + A_0 x_0 + \hat{B}(\upsilon + \Delta u_{ref}) + \hat{E}d$$

$$\begin{bmatrix} 0 & -I \\ 0 & I \\ -\hat{H} & 0 \\ \hat{H} & 0 \end{bmatrix} \begin{bmatrix} \chi \\ \upsilon \end{bmatrix} \leqslant \begin{bmatrix} -\Delta\hat{U}_L + \Delta u_{ref} \\ \Delta\hat{U}_U - \Delta u_{ref} \\ -\hat{Y}_L + \hat{H}x_{ref} \\ \hat{Y_U} - \hat{H}x_{ref} \end{bmatrix} \tag{57}$$

where

$$\chi = \begin{bmatrix} x_1 - x_{ref,1} \\ \vdots \\ x_n - x_{ref,n} \end{bmatrix}, \quad \upsilon = \begin{bmatrix} \Delta u_0 - \Delta u_{ref,0} \\ \vdots \\ \Delta u_{n-1} - \Delta u_{ref,n-1} \end{bmatrix},$$

$$\hat{Q} = \begin{bmatrix} C^T Q C & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & C^T Q C & 0 \\ 0 & \cdots & 0 & C^T S C \end{bmatrix}, \quad \hat{P} = \begin{bmatrix} P & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & P \end{bmatrix},$$

$$\hat{A} = \begin{bmatrix} 0 & \cdots & & 0 \\ \tilde{A} & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & \tilde{A} & 0 \end{bmatrix}, \quad A_0 = \begin{bmatrix} \tilde{A} \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$\hat{B} = \begin{bmatrix} 0 & \cdots & & & 0 \\ \vdots & \ddots & & & \vdots \\ 0 & \cdots & & 0 & \\ \tilde{B} & 0 & \cdots & 0 & \\ 0 & \ddots & & \vdots & \\ \vdots & & & 0 & \vdots \\ 0 & \cdots & 0 & \tilde{B} & 0 \end{bmatrix}, \quad \hat{E} = \begin{bmatrix} \tilde{E} & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \tilde{E} \end{bmatrix},$$

$$\hat{H} = \begin{bmatrix} \tilde{C} & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \tilde{C} \end{bmatrix},$$

$$\Delta \hat{U}_L = \begin{bmatrix} \Delta U_L \\ \vdots \\ \Delta U_L \end{bmatrix}, \quad \Delta \hat{U}_U = \begin{bmatrix} \Delta U_U \\ \vdots \\ \Delta U_U \end{bmatrix}, \quad \Delta \hat{Y}_L = \begin{bmatrix} Y_L \\ \vdots \\ Y_L \end{bmatrix}, \quad \Delta \hat{Y}_U = \begin{bmatrix} Y_U \\ \vdots \\ Y_U \end{bmatrix},$$

$$\Delta u_{ref} = \begin{bmatrix} \Delta u_{ref,0} \\ \vdots \\ \Delta u_{ref,n-1} \end{bmatrix}, \quad x_{ref} \begin{bmatrix} x_{ref,1} \\ \vdots \\ x_{ref,n} \end{bmatrix} \tag{58}$$

This QP-problem is an ineffective variant to implement. Looking at the number of optimization variables it is easy to see that it is $n(n_x + 1)$, where n_x is the number of system states. By following a similar procedure as done in Hovd [12, page 131-132], it is possible to reduce the number of optimization variables to n. Start by using Equation 57:

$$
\begin{aligned}
(I-\hat{A})(\chi - x_{ref}) &= A_0 x_0 + \hat{B}(\upsilon + \Delta u_{ref}) + \hat{E}d \\
\chi &= (I-\hat{A})^{-1} A_0 x_0 + (I-\hat{A})^{-1}\hat{B}(\upsilon + \Delta u_{ref}) + (I-\hat{A})^{-1}\hat{E}d - x_{ref} \\
\chi &= \hat{A}_2 x_0 + \hat{B}_2(\upsilon + \Delta u_{ref}) + \hat{E}_2 d - x_{ref}
\end{aligned}
$$

This expression can be separated into two parts by exploiting the superposition principle. One that is independent of the optimization variable, υ, and one that is dependent. They will be named as follows:

$$
\begin{aligned}
\chi_{dev} &= \hat{A}_2 x_0 + \hat{B}\Delta u_{ref} + \hat{E}_2 d - x_{ref} \\
\chi_{\upsilon} &= \hat{B}_2 \upsilon
\end{aligned} \tag{59}
$$

Now it is possible to take Equation 59 and combine with Equation 56, Equation 57 and Equation 58 and get Equation 50 with the following parameters:

$$
\begin{aligned}
\tilde{H} &= \hat{B}_2^T \hat{Q} \hat{B}_2 + \hat{P} \\
c^T &= \chi_{dev}^T \hat{Q} \hat{B}_2 \\
L &= \begin{bmatrix} -I \\ I \\ -\hat{H}\hat{B}_2 \\ \hat{H}\hat{B}_2 \end{bmatrix}, \qquad b = \begin{bmatrix} -\Delta\hat{U}_L + \Delta u_{ref} \\ \Delta\hat{U}_u - \Delta u_{ref} \\ -\hat{Y}_L + \hat{H}(x_{ref} + x_{dev}) \\ \hat{Y_U} - \hat{H}(x_{ref} + x_{dev}) \end{bmatrix}
\end{aligned} \tag{60}
$$

7.2.2 *Implementation*

The MPC-controller was implemented in MATLAB. The advantage of the setup from Section 7.2.1 is that a lot of the calculations can be done off-line.

To implement the off-line part of the problem a *.m-file* was used. The results from the off-line calculation were stored in a *.mat-file*. For the online implementation of the problem a SIMULINK level 2 S-function was used. The S-function loads the *.mat-file* and takes in $p_{dh,ref}$, x_0 and d_0. The outputs are $p_{c,ref}$ and $\Delta p_{c,ref}$. The user must set the period of the disturbance. Inside the S-function the period parameter is used together with d_0 to estimate the future disturbance using the harmonic oscillator described in Section 6.10. The state reference, x_{ref}, is calculated by : x_{ref} = pseudoInverse($\tilde{C}$)$p_{dh,ref}$. The QP-problem is solved in the S-function by using the MATLAB function quadprog().

The inputs to the MPC-controller are found as follows. The reference value for the bottomhole pressure is set by the user. The disturbance state is calculated by using a harmonic oscillator as described in Section 6.10. When doing the model identification, Section 6.8, a kalman parameter gain was also identified. This parameter was then

Listing 1: Tuning part of *setupMPC.m*

```
%%%%%%%%%%%%%%%%% MPC Tuning Parameters %%%%%%%%%%%%%%%%%%%%%%%%%
sampleMPC = 0.2; % Sample time for QP problem
n = 30; % MPC horizon

% Reference point
urefm = 0; % deltaUref

% Weight
yweight = 1000;
finalState = 2000;
uweight = 1;
deltauweight = 800;

% Constraints
deltaUl = -0.5; deltaUu = 0.5;
Yl = [-Inf; 0]; Yu = [Inf; 10]; % Yl = [Yl; Ul] Yu [Yu; Yu]

% Delay for pc
delay = 0.6;
```

used together with the original model from Equation 51 to set up a Kalman filter for estimating the state of the system.

For MATLAB-files implementation of the controller see Appendix C.

7.2.3 *Tuning Parameters*

There are multiple parameters that can be used for tuning the MPC-controller. The advantage of doing the off-line calculation in a *.m-file* and storing them in a *.mat-file* that gets loaded for each sample in the on-line calculation, is that the MPC-controller can be tuned on-line by running the off-line setup file. In addition the MPC-controller gives out the change in the choke pressure even though this is not necessary, but this enables the user to see if the constraints for the change in choke pressure are active or not.

In Listing 1 is the tuning part of the file *setupMPC.m* that is used to do the off-line calculation for the MPC-controller is shown.

The two most critical parameters for calculation time is the sample time for the MPC and the prediction horizon. By multiplying these two parameters the user is able to find out how far into the future the MPC is predicting. As it is tuned in the given listing the MPC is able to predict $0.2x30 = 6$ seconds into the future. The parameter *sampleMPC* should not be much less than 0.05 seconds since calculating the on-line calculation for the MPC takes time. The predicted horizon, n, depends on what the sample time is set to. Shorter times between samples means a shorter horizon must be used.

The parameter *urefm* should always be zero. This is the desired change for $\Delta p_{c,ref}$ which always should be zero.

The weight parameters are used to assign weight to the different objectives in the objective function. The change done in Equation 55 enables the user to easily assign weight to following the reference pressure, change in the output and if desired the output. The weights are used to make the Q, P and S matrix from Equation 53:

$$\begin{aligned} Q &= \begin{bmatrix} Yweight & 0 \\ 0 & uweight \end{bmatrix} \\ S &= \begin{bmatrix} finalState & 0 \\ 0 & uweight \end{bmatrix} \\ P &= \Delta uweight \end{aligned} \tag{61}$$

Constraints are used to make sure the choke controller is able to follow the reference set by the MPC-controller. If the choke controller is having trouble following the reference set from the MPC controller the parameters *deltaUl* and *deltaUu* should be set to values closer to zero. These parameters are very dependent on the sample time of the MPC controller. The setup from Equation 55 makes Y_L and Y_U contain both the limits for the controller variable, p_{dh}, and the output, $p_{c,ref}$. First the $p_{c,ref}$ is set to be between 0 and 10 bars through constraints. This generally ensures that the bottomhole pressure is also within these limits. The limits for the bottomhole pressure are removed since having limits here can easily lead to an infeasible problem (a QP-problem without solution).

Finally a delay from p_c and into the model is set. This delay could easily have been included in the model by using a Padé approximation as described in Section 6.9. However by introducing this time delay into the model with Equation 52 and making it a tunable parameter turned out to be very useful. The *delay* parameters are used together with the sample time of the MPC to set the constant l.

7.3 SIMULATION

Figure 43 shows the MPC-controller suppressing a disturbance with a 3 seconds period applied to the model identified in Section 6.8. The tuning parameters and the implementation can be found in Appendix C.

In theory the MPC-controller should be able to perfectly reject the disturbance if no constraint is active and full knowledge of the disturbance is included. However recalling from Section 7.2.1 the model was simplified from Equation 51 to Equation 52 in order to save online computation time. This simplification should be sufficient to explain the small deviation from perfect tracking in the test shown in Figure 43.

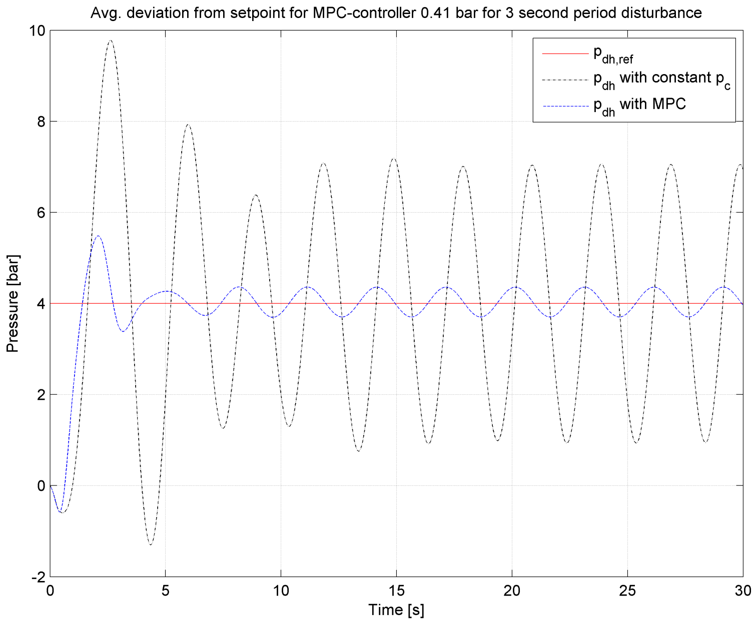


Figure 43: MPC-Controller SIMULINK simulation. In the simulation the choke is not modeled, thus instead of comparing to constant choke angle a comparison is made to constant choke pressure.

Disturbance	Without MPC [bar]		With MPC [bar]		Suppression
Period [s]	Low	High	Low	High	in %
10	2.6	6.2	3.5	4.8	63.9
5	1.5	7.4	2.9	5.4	57.6
3	3	5.6	3.4	4.8	46.2

Table 9: Variation of Bottomhole Pressure with and without MPC-controller

7.4 TEST IN LAB

The MPC-controller was tuned online in the lab. The disturbance was set to follow a 3 seconds period, while adjustments were made in the MPC-setup. The tuning parameters used in the tests can be seen in Listing 1.

The results for attempts at suppressing disturbance with period of 10, 5 and 3 second period with the MPC-controller can be found in Figure 44, Figure 45 and Figure 46. The variation of the pressure for a 4 bar reference with and without the MPC controller for disturbance with different periods can be found in Table 9. The suppression in percentage was calculated by $1 - \frac{\text{pressure variation with MPC}}{\text{pressure variation without MPC}}$.

For a discussion of the results see Chapter 8.

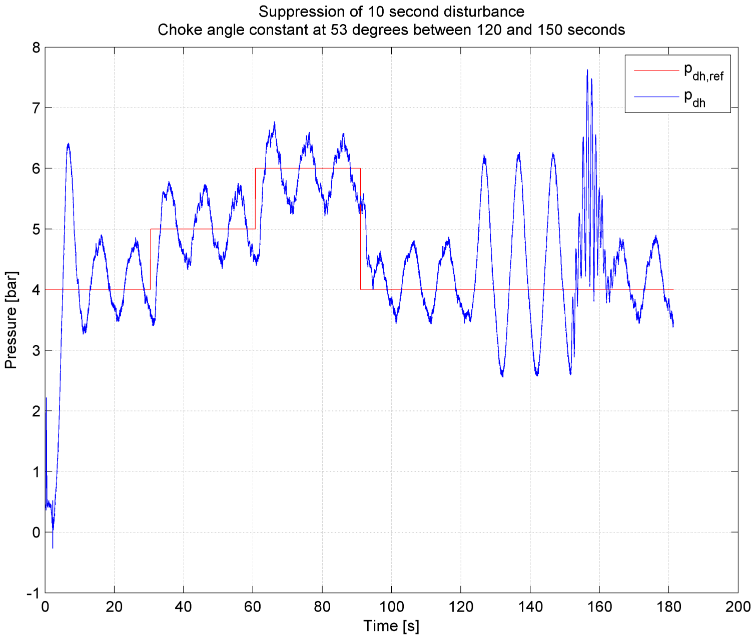


Figure 44: MPC-Controller Suppression of Disturbance with 10 Second Period in Lab

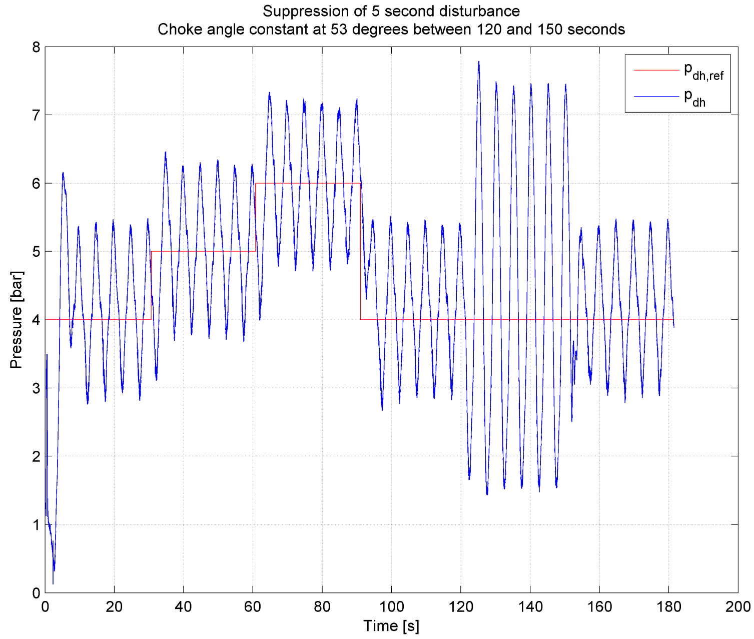


Figure 45: MPC-Controller Suppression of Disturbance with 5 Second Period in Lab

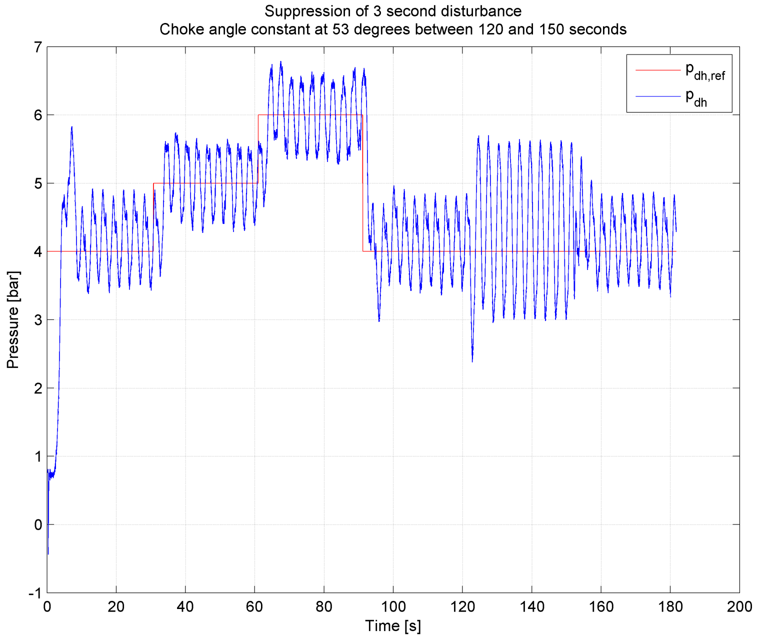


Figure 46: MPC-Controller Suppression of Disturbance with 3 Second Period in Lab

8 DISCUSSION

The MPC-controller in Section 7.2 was only able to suppress the disturbance by 40-60 % depending on the period of the disturbance. As expected it turned out to be easier to suppress disturbances of longer time periods, due to the fact that for shorter time periods the time delay through the well is larger compared to the time period of the disturbance.

There are mainly two reasons why perfect rejection of the disturbance was not achieved. The model used by the MPC is inaccurate. The model identified in Section 6.8 is very accurate when using a Kalman gain to adjust for the measured bottomhole pressure. However in the MPC the model is used to predict into the future, and thus no measured bottomhole pressure is available. In addition in order to save online calculation time the D and F matrix from Equation 51 was set to zero (see Section 7.2.1). Also the model become more inaccurate for shorter time period of the disturbance. In order to demonstrate the inaccuracy of the model Figure 47 shows the prediction used by the MPC-controller for a controller horizon during the suppression of a 3 seconds period disturbance (Figure 46). As can be seen in Figure 47 the MPC-controller uses a prediction that is approximately 2 bars wrong in the worse case.

In addition to the inaccuracy in the model the choke controller is not able to perfectly track the pressure set by the MPC-controller. Figure 48 shows the choke tracking performance for the same period of time as shown in Figure 47. The tracking is at most about 1 bar off the desired pressure.

Besides the two mentioned reasons for the MPC-controllers inability to perfectly reject the disturbance, an inaccurate model of the disturbance might contribute. However, as can be seen in Figure 49, the prediction of the disturbance is very accurate. Thus the prediction of the disturbance does not contribute to worsen the performance of the MPC-controller.

Designing a controller and identifying a model has been a cyclic effort. First a model has been identified. Then the model has been used to design a controller. When testing the controller it has been discovered that more work is needed on identifying a better model. If more time was available a new round of identifying a better model would most likely improve the MPC-controller.

If a new model should be attempted to be identified it should be done as follows. The experiments where the MPC attempts to suppress a 3, 5 and 10 seconds period should be used as identification

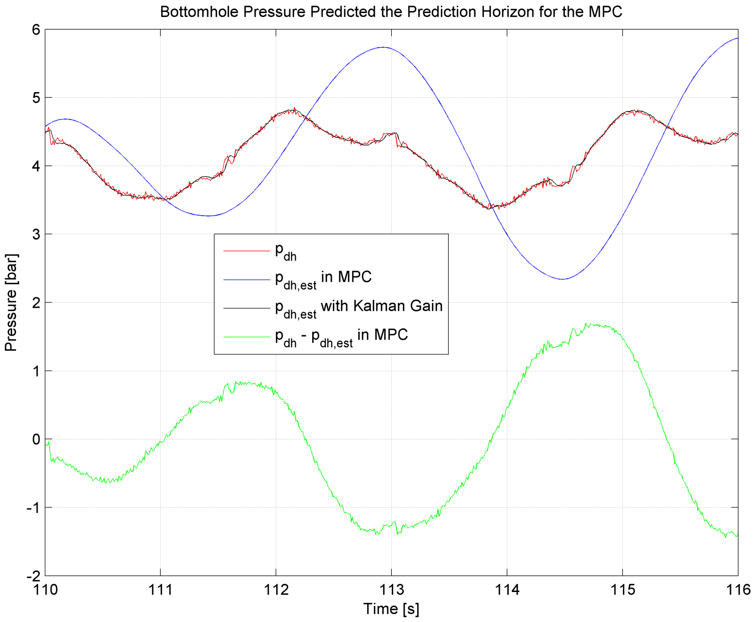


Figure 47: Prediction used by the MPC-controller

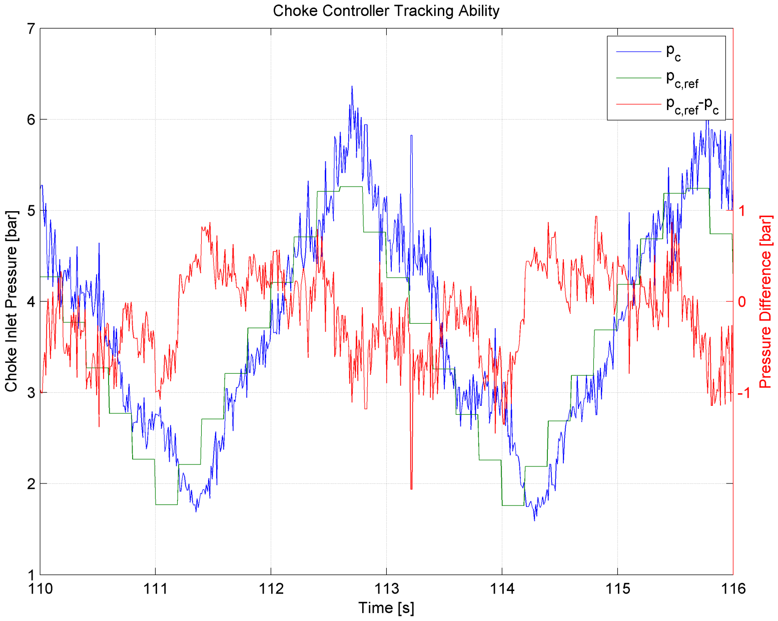


Figure 48: Tracking of the MPC-set trajectory by the Choke controller

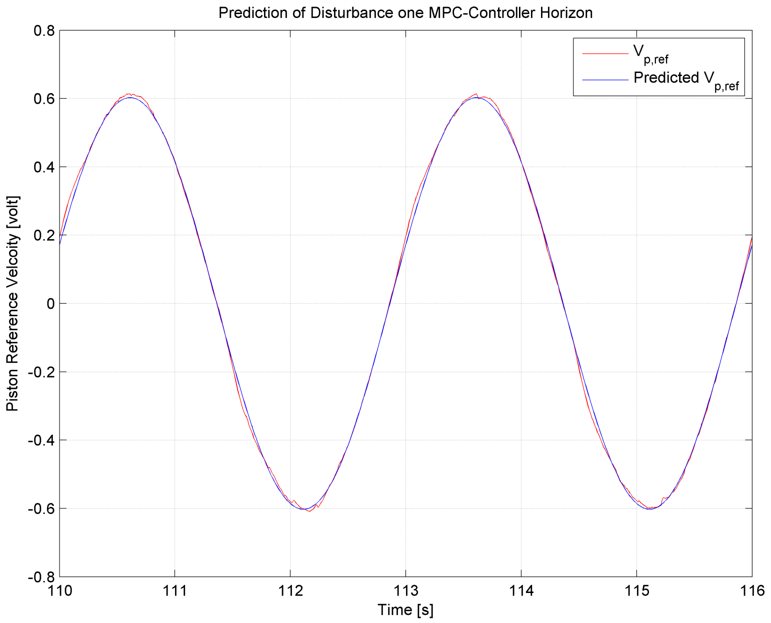


Figure 49: Prediction of Disturbance used by MPC

sets, since these are the cases an accurate model are needed for. It might be considered to expand the online calculation for the MPC-controller by not setting F, and perhaps D, to zero in Equation 51. The model as used by the MPC-controller should then be used when testing the identified model. This was unfortunately not how it was done in Section 6.6. The model used to test the identified model was not with the D and F matrices set to zero as was done for the MPC-controller. This made the model used in the MPC-controller perhaps not be as accurate as possible.

Even if perfect rejection of the disturbance was achieved, the controller would still have made a few important simplifications compared to the real case. First the disturbance is assumed to be a single sine wave with a given frequency. This is not the case at sea. It is easy to expand the model of the disturbance to include multiple sine waves with given frequencies. However the model will still assume some periodicity in the waves. This may not be a valid assumption. It might be necessary to have a more complex model with measurements of the sea around the rig in order to predict the heave disturbance.

Another simplification of the problem is the bottomhole measurement. This will probably not be available in the real case. The controller will have to control the pressure based upon an estimate. Also not having the bottomhole pressure available will make identifying a model impossible as done in Section 6.6.

The goal of this thesis has been to show that it is possible to suppress heave disturbance in an experimental lab setup. Making the above simplifications have made this easier to achieve. The goal has been reached since the MPC-controller introduced in this thesis was able to suppress a 3 seconds period heave disturbance by approximately 46 % compared to constant choke angle. This is a clear improvement over the PI-controller discussed in Section 7.1.1. The PI-controller was only able to improve performance over a constant choke angle for disturbances with period of 7 seconds and longer.

9 CONCLUSION AND FURTHER WORK

9.1 CONCLUSION

Much of the work of this master thesis has been to make the IPT-Heave Lab operational. As discussed in detail in Chapter 4 getting the choke controller to perform satisfactory proved more difficult than expected. In addition getting the flowmeter, FT1, from the producer was delayed multiple times. It was not installed in the lab until Easter. However the lab is now (June 2013) fully operational with the exception of the planned flow measurement FT4, which will be placed at the copper pipe outlet.

After getting the electrical motor controlling the choke to perform satisfactory, a PI-Controller with feedforward and gain scheduling for the integral part has been designed and demonstrated to perform satisfactory in the lab (Chapter 4).

A small literature review has been performed in Chapter 5. This literature study motivated the attempt at identifying a linear model of the system using a black box approach. A script has been created as a framework for identifying models of the lab. This has been used to identify a reasonably accurate model of the lab.

The linear model identified in Chapter 6 has been used to design an MPC-controller. The MPC-controller has been demonstrate to suppress disturbances with a period of 3 seconds by approximately 46 %. This was a clear improvement compared to the PI-controller tuned for the problem in Section 7.1.1, which was not able to improve heave suppression compared to a constant choke angle for disturbances with a period of 5 seconds or faster.

9.2 FURTHER WORK

The results in this thesis motivate several future areas to focus research on when it comes to the suppression of heave disturbance for a floating rig during a connection scenario.

First it is possible to continue the work of this thesis with suppressing the heave disturbance taking the simplification of a disturbance of a single sine wave and use the measurement of the bottomhole pressure. As discussed in Chapter 8 it should be possible to improve the suppression of the disturbance compared to what was achieved in this thesis. However, even though not a perfect rejection of a 3 seconds disturbance have been achieved, a suppression has been accomplished and it might be time to move on.

After having successfully identified a linear model based upon a black box approach the natural next step would be to develop a model based upon physical principles. However even a model based upon physical principles must be expected to have some kind of estimation. For example, the friction and the effective bulk modulus of water in the pipe must be expected to be estimated. An online identification algorithm will probably have to be developed for estimating these parameters.

The model used in the MPC-controller does not take advantage of the flow measurement, FT1, topside. The measurement should be included, and doing so will most likely improve the estimation of the bottomhole pressure.

When a physical model is developed the MPC-controller from this thesis or one of the controllers from Chapter 5 can be implemented in the lab.

When suppression of a disturbance consisting of a single sine wave has been done successfully, a more realistic disturbance model should be attempted. Ideally no assumed knowledge of the disturbance should be expected, unlike in this thesis where the frequency of the wave is assumed to be known a priori.

BIBLIOGRAPHY

[1] Aamo, O. M. (2012). Distrubance Rejection in 2 x 2 Linear Hyperbolic Systems. *IEEE Transaction on Automatic Control*, *PP*(99):1. DOI: 10.1109/TAC.2012.2228035.

[2] Aarsnes, U. J. F. (2012). *Reduced Order Observer Design for Managed Pressure Drilling*. Master thesis, Norwegian University of Science and Technology. Trondheim, Norway.

[3] Albert, A. (2012). *Disturbance Attenuation in Managed Pressure Drilling: Tracking Control of the Choke Pressure*. Project thesis, Norwegian University of Science and Technology. Trondheim, Norway.

[4] Devereux, S. (2012). *Drilling Technology in Nontechnical Language*. PennWell, Tulsa, Oklahoma, USA, 2th edition.

[5] Drønnen, R. (2013). *Disturbance Generation in an Experimental Lab Setup for Managed Pressure Drilling*. Master thesis, Norwegian University of Science and Technology. Trondheim, Norway.

[6] Evans, L. C. (2010). *Partial Differential Equations*. The American Mathematical Society, Rhode Island, USA, 2th edition.

[7] Gjengseth, C. S. (2012). *Lab for Heave Motion During Managed Pressure Drilling*. Master thesis, Norwegian University of Science and Technology. Trondheim, Norway.

[8] Gjengseth, C. S. and Svenum, T.-I. (2011). *Heave Compensated Managed Pressure Drilling: A Lab Scaled Rig Design*. Project thesis, Norwegian University of Science and Technology. Trondheim, Norway.

[9] Gleditsch, M. S. (2013). *Disturbance Attenuation in Managed Pressure Drilling Using Impedance Matching and an Experimental Lab Setup*. Master thesis, Norwegian University of Science and Technology. Trondheim, Norway.

[10] Godhavn, J.-M. and Knudsen, K. A. (2010). *High Performance and Reliability for MPD Control System Ensured by Extensive Testing*. Paper published at: IADC/SPE Drilling Conference and Exhibition, New Orleans, Louisiana, USA.

[11] Hannegan, D. (2006). *Case Studies - Offshore Managed Pressure Drilling*. Paper published at: SPE Annual Technical Conference and Exhibition, San Antonio, Texas, USA.

[12] Hovd, M. (2012). *Lecture notes for the course Advanced Control of Industrial Process*. Institutt for Teknisk Kybernetikk, NTNU, Norway.

[13] Kaasa, G., Stamnes, O., Imsland, L., and Aamo, O. (2011). *Intelligent Estimation of Downhole Pressure Using a Simple Hydraulic Model*. Paper published at: IADC/SPE Managed Pressure Drilling and Underbalanced Operations Conference and Exhibition, Denver, Colorado, USA.

[14] Kozicz, J. (2006). *Managed Pressure Drilling - Recent Experience, Potential Efficiency Gains, and Future Opporunities*. Paper published at: IADC/SPE Asia Pacific Drilling Technology Conference and Exhibition, Bangkok, Thailand.

[15] Landet, I. S. (2011). *Modeling and Control for Managed Pressure Drilling from Floaters*. Master thesis, Norwegian University of Science and Technology. Trondheim, Norway.

[16] Landet, I. S., Mahdianfar, H., Aarsnes, U. J. F., Pavlov, A., and Aamo, O. (2012a). *Modeling for MPD Operations with Experimental Validation*. Paper published at: IADC/SPE Drilling Conference and Exhibition, San Diego, California, USA.

[17] Landet, I. S., Pavlov, A., and Aamo, O. (2012b). Modeling and Control of Heave-Induced Pressure Fluctuation in Managed Pressure Drilling. *IEEE Transactions on Control Systems Technology, PP*(99):1.

[18] Mahdianfar, H., Aamo, O., and Pavlov, A. (2012). *Attenuation of Heave-Induced Pressure Oscillations in Offshore Drilling Systems*. Paper published at: American Control Conference, Fairmont Queen Elizabeth, Montreal, Canada.

[19] Nas, S. and Toralde, J. S. (2009). *Offshore Managed Pressure Drilling Experiences in Asia Pacific*. Paper published at: IADC/SPE Drilling Conference and Exhibition, New Orleans, Louisiana, USA.

[20] Nilsson, J. W. and Riedel, S. A. (2008). *Electric Circuits*. Pearson, Singapore, 8th edition.

[21] Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, USA, 2th edition.

[22] Overschee, P. V. and Moor, B. D. (2011). *Subspace Identification for Linear Systems: Theory - Implementation - Applications*. Springer London, Katholieke Universiteit Leuven, Belgium, 1th edition.

[23] Pavlov, A., Wouw, N. V. D., and Nijmeijer, H. (2005). *Uniform Output Regulation of Nonlinear Systems: A Convergent Dynamics Approach*. Birkhauser, Switzerland, 1th edition.

[24] Phade, A. (2013). *Heave Compensated Managed Pressure Drilling* - Lab Experiments. Master thesis, Norwegian University of Science and Technology. Trondheim, Norway.

[25] Rasmussen, O. S. and Sangesland, S. (2007). *Evaluation of MPD Methods for Compensation of Surge-and-Swab Pressures in Floating Drilling Operations*. Paper published at: IADC/SPE Managed Pressure Drilling and Underbalanced Operations Conference and Exhibition, Galveston, Texas, USA.

[26] Ruscio, D. D. (1995). *Subspace System Identification Theory and Applications*. Telemark Institute of Technology, Porsgrunn, Norway, 1th edition.

[27] Schlumberger (2012). *Oilfield Glossary*. http://www.glossary.oilfield.slb.com/.

[28] Skogestad, S. and Postlethwaite, I. (2007). *Multivariable Feedback Control*. John Wiley & Sons Ltd, Chichester, West Sussex, England, 2th edition.

[29] Vazquez, R., Krstic, M., and Coron, J.-M. (2011). *Backstepping Boundary Stabilization and state Estimation of a 2 x 2 Linear Hyperbolic System*. Paper published at: European control conferene (CDC-ECC), Orlando, FL, USA.

[30] White, F. M. (2008). *Fluid Mechanics*. McGraw-Hill, New York, USA, 6th edition.

A IPT-HEAVE LAB START GUIDE

This chapter contains a set of procedures for the lab. It is meant as an absolute basic required for using the IPT-Heave lab. The same appendix is also given in the reports by Drønnen [5] and Gleditsch [9]. Some of the content from Chapter 3 may be repeated for the sake of both this chapter being independent and identical to Drønnen [5] and Gleditsch [9].

A.1 STARTUP PROCEDURE

1. Turn on computer. Password: *Espen*.
2. Make sure the BHA does not trigger the limit switches and that there is water in the tank, see Section A.7. If necessary pull the BHA away from the limit switch, see Section A.4.
3. Make sure the manual valves are set in desired positions, Figure 50.
4. Turn on the Lenze controller with the switch on the electrical cabinet (placed to the left of the computer). Make sure the socket to the NI cards is plugged in (placed below the computer), Figure 51.
5. Home the BHA by using the switches on the control box, procedure described in Section A.6.
6. Turn on the feed and backpressure pumps, Figure 52. Pressurize the system if desired by the procedure described in Section A.8 or Section A.9.
7. Choose desired reference for the BHA. It is set in the *Reference Choice* block in Simulink (Figure 61) and choose the parameters in the chosen reference category. If a custom BHA reference is made, be sure it is correct, Section A.12.
8. Choose desired input for choke, see Section A.13.
9. Turn controllers for BHA and choke on.
10. Press the start button in SIMULINK.

A.2 SHUTDOWN PROCEDURE

1. Stop the SIMULINK model

2. Store the results from experiment using MATLAB command *save('filename.mat')*.
3. Turn off BHA and choke controller by flipping all the switches down on the control box, except the enable controller switch which shall be flipped up. Shown in Figure 56.
4. Turn off feed and backpressure pumps, Figure 52.
5. Turn off the Lenze controller with the switch on the electrical cabinet
6. Turn off computer

A.3 WARNINGS

- Make sure the manual valves *MV5* and *MV6* are open when the pumps are running. Closing them could lead to rapid pressure increase and damage to equipment and personnel. Fig. 50
- Do not change the reference choice for the BHA while running
- If wave generator or rig data is selected as reference signal for the BHA, do not change the parameters while running
- Pay attention to the pressure while running, the choke may cause big pressure rises in the system.
- Pressure in the system should not exceed 10.5 bars or be below -0.5 bars. The pressure transmitters are measuring gauge pressure and a measurement below -1 bar will indicate vacuum. Both a pressure exceeding 10.5 bars and vacuum may be damaging to the system. In the well these limits are ensured by a control block in the SIMULINK model that will stop the simulation if the limits are exceeded.

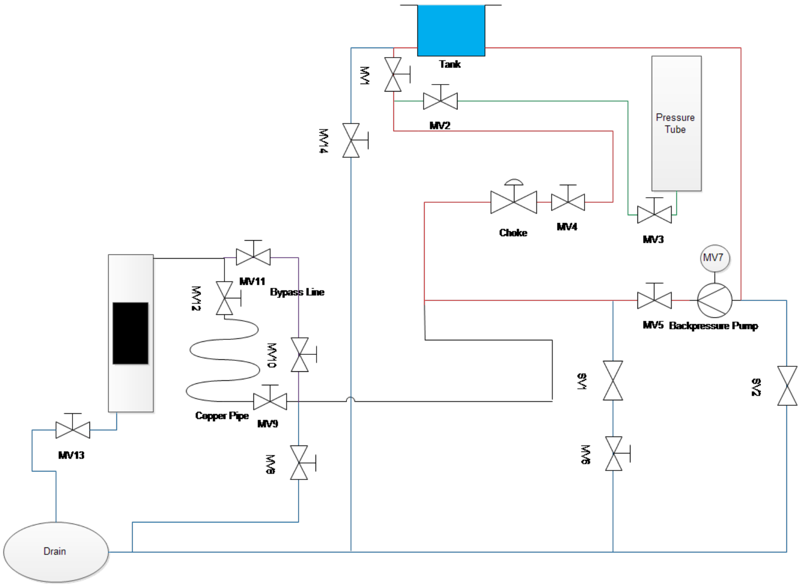


Figure 50: The manual valves enables multiple flow paths for the system. In addition MV4 enables to manually close the choke. MV6 should always be open when the backpressure pump is on.

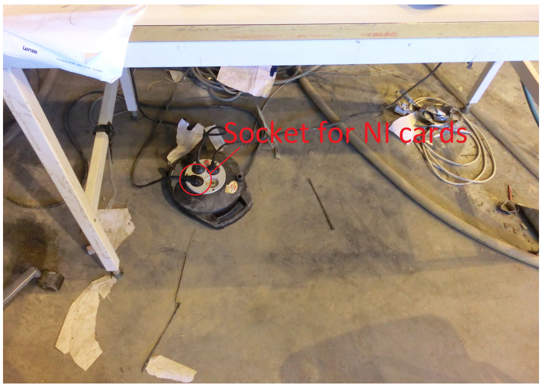


Figure 51: Socket for NI cards

A.4 PULL BHA AWAY FROM LIMIT SWITCH

The BHA is connected to a lower and an upper rod. These rods are connected to a sawtooth belt, which is controlled by an electrical motor. To ensure that the electrical motor does not attempt to run the BHA out of the PVC tube, two limits switches have been installed. Triggering one of these will lead to the motor being locked. To reset a limit switch see Section A.5. The setup is shown in Figure 53. If the end of the lower rod connected to the BHA is triggering the lower limit switch the BHA needs to be pulled away. To do this first make sure that the controller for the BHA is turned off and that the power of the electrical cabinet is switched to off. Then grab the sawtooth belt and physically pull until the end of the lower rod is clear of the lower limit switch.

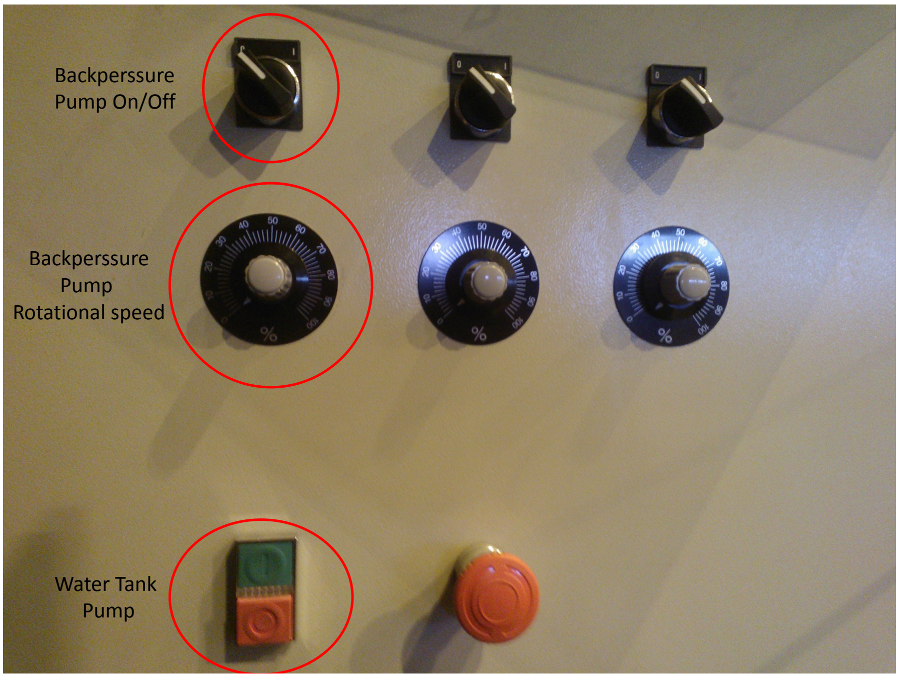


Figure 52: Electrical cabinet to the feed and the backpressure pump. To turn them on first hit green button to turn on the feed pump. Then flip the switch on top, to turn on the backpressure pump. The scroll wheel enables adjusting the rotational speed for the backpressure pump. The pumps are turned off by first turn the scroll wheel to 0 %, flip the top switch to zero and hitting the red button (not the emergency stop).

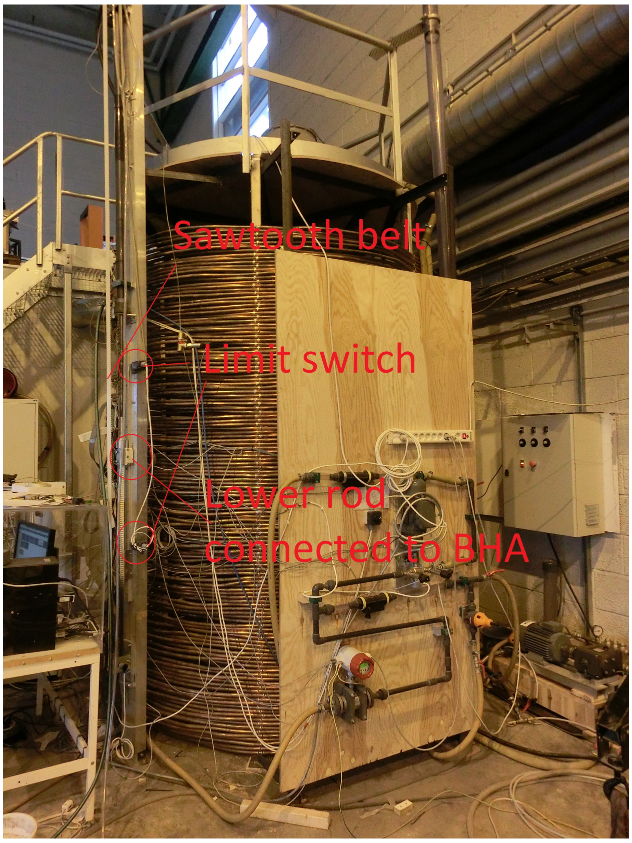


Figure 53: BHA with limit switches marked.

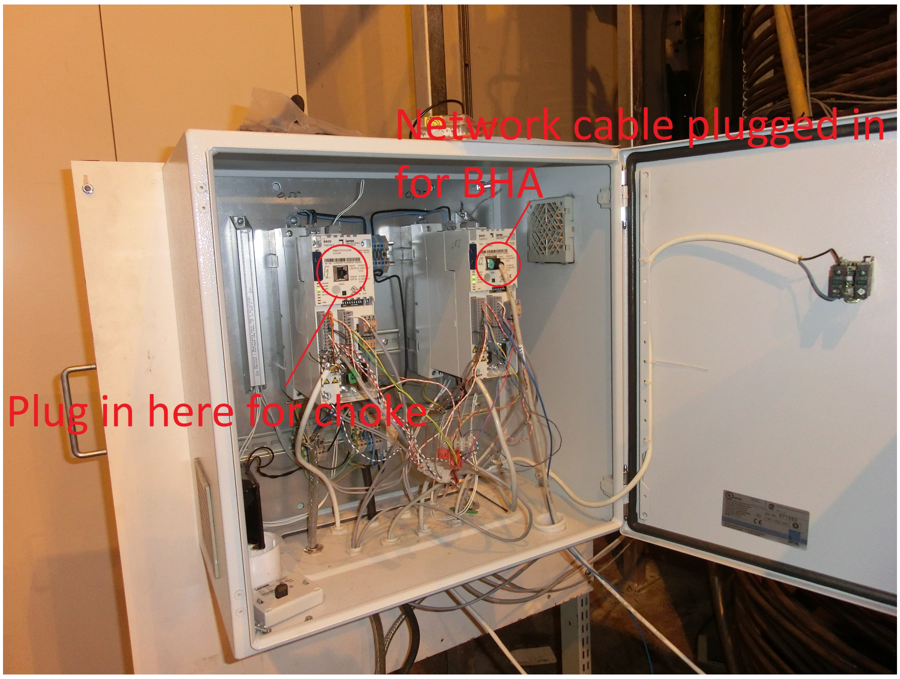


Figure 54: How to plug in network cable

A.5 RESET LIMIT SWITCH IN ENGINEER

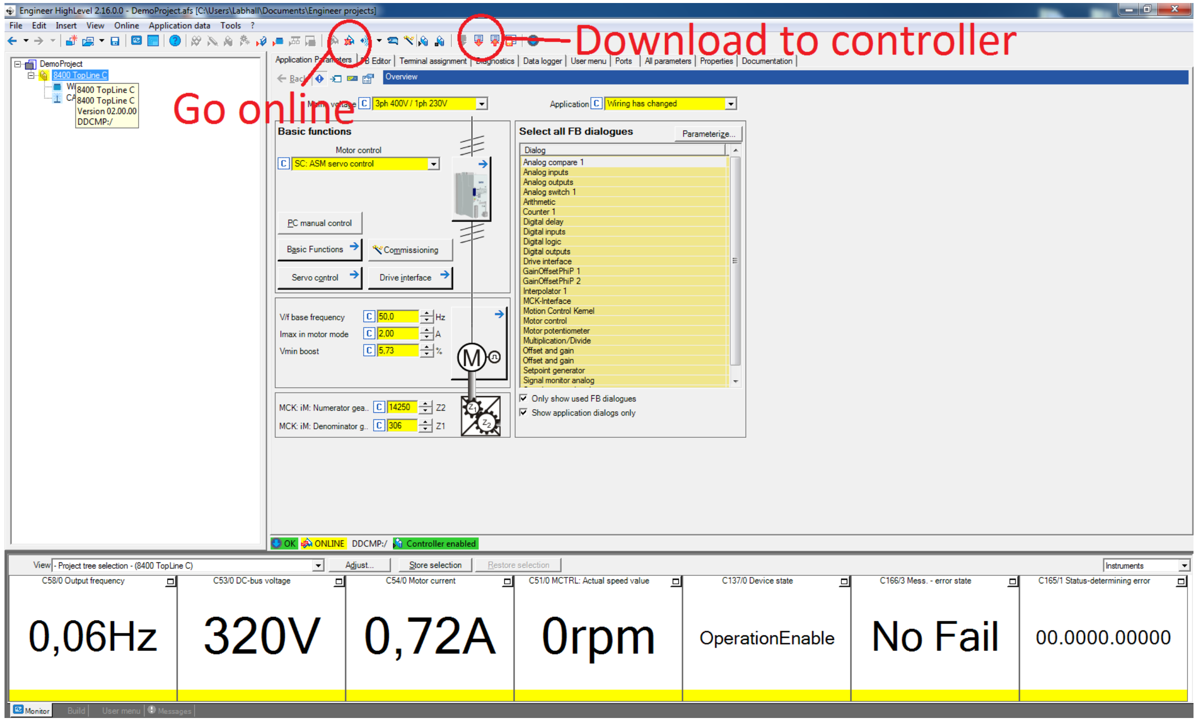
There are two ways to reset the limit switch. It can be done in the software Engineer or the Lenze controllers must be shut down and restarted.

To reset the errors in Engineer, follow this procedure:

1. Plug the network cable into the controller you want to access the setting for (BHA/choke). See Figure 63.
2. Startup Engineer by double clicking on the *L-Force Engineer High-Level* icon on the desktop
3. In the startup screen you get the choice between opening an existing Engineer project or make a new project. Select open an existing project.
4. For the choke, choose *LE1_Anders* and for the BHA choose *LE2_Anders*.
5. Once the project is opened up click the *Go online* button (marked in Figure 64). Your screen should now look like Figure 64.
6. You are now ready to make changes in the software for the controller.

Once Engineer is online, the limit switch can be reset:

1. Turn off the controller by flipping the *Enable controller* switch on the control box.


Go online
Download to controller
File Edit Insert View Online Application data Tools ?
Application Parameters | FB Editor | Terminal assignment | Diagnostics | Data logger | User menu | Ports | All parameters | Properties | Documentation
Overview
Basic functions
PC manual control
Basic Functions
Commissioning
Servo control
Drive interface
Select all FB dialogues
Parameterize...
Dialog
Monitor | Build | User menu | Messages

Engineer HighLevel 2.16.0.0 - DemoProject.afs [C:\Users\Labhall\Documents\Engineer projects]

DemoProject
8400 TopLine C
8400 TopLine C
8400 TopLine C
Version 02.00.00
DDCMP:/

Mains voltage: C 3ph 400V / 1ph 230V

Application: C Wiring has changed

Motor control: C SC: ASM servo control

V/f base frequency	C 50.0	Hz
Imax in motor mode	C 2.00	A
Vmin boost	C 5.73	%
MCK: iM: Numerator gea..	C 14250	Z2
MCK: iM: Denominator g..	C 306	Z1

Analog compare 1
Analog inputs
Analog outputs
Analog switch 1
Arithmetic
Counter 1
Digital delay
Digital inputs
Digital logic
Digital outputs
Drive interface
GainOffsetPhiP 1
GainOffsetPhiP 2
Interpolator 1
MCK-Interface
Motion Control Kernel
Motor control
Motor potentiometer
Multiplication/Divide
Offset and gain
Offset and gain
Setpoint generator
Signal monitor analog

☑ Only show used FB dialogues
☑ Show application dialogs only

OK ONLINE DDCMP:/ Controller enabled

View - Project tree selection - (8400 TopLine C) | Adjust... | Store selection | Restore selection | Instruments

C58/0 Output frequency	C53/0 DC-bus voltage	C54/0 Motor current	C51/0 MCTRL: Actual speed value	C137/0 Device state	C166/3 Mess. - error state	C165/1 Status-determining error
0,06Hz	320V	0,72A	0rpm	OperationEnable	No Fail	00.0000.00000

Figure 55: Startup screen loaded project

Enable the Lenze Controllers	UP=OFF, DOWN=ON
Homing/Run (B2)	UP=Run, DOWN=Home
BHA (B1)	UP=ON, DOWN=OFF
Choke (B0)	UP=ON, DOWN=OFF

Table 10: Switches on controller box

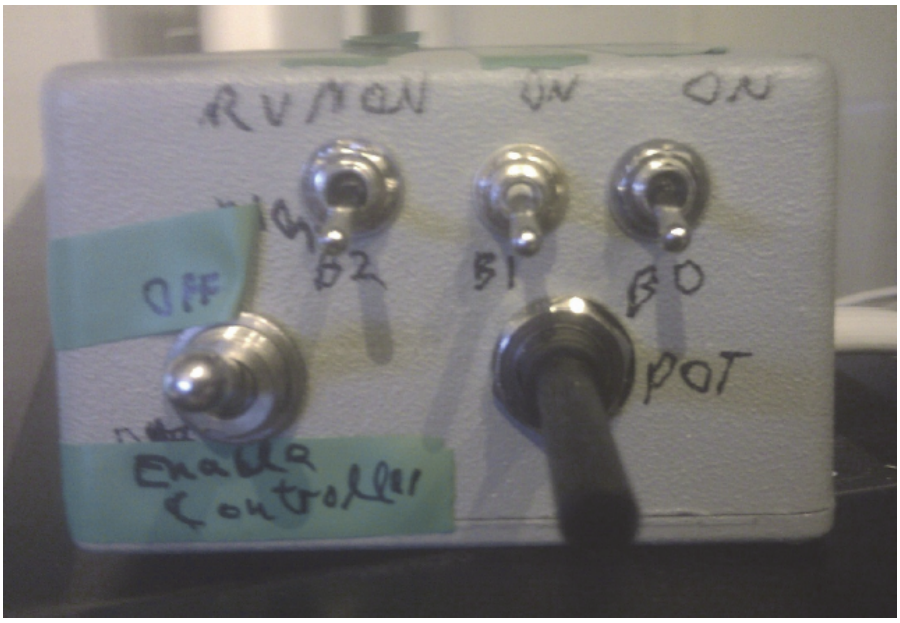


Figure 56: Manual switches for control of the BHA and choke.

2. Pull the BHA away from the limit switch, see Section A.4.
3. In Engineer click on the *Diagnostics* tab.
4. In the lower right corner there should be a button called *Reset Error*. Click this button
5. The status display at the bottom called *C166/3 Mess. - error state* should now change from *Ck02: Neg. HW-LimitSwitch* to *No Fail*.

A.6 HOMING PROCEDURE

To use the BHA and the choke, a homing procedure must be performed. This enables the controllers to find their starting location. Figure 56 shows the control box for the BHA and the choke. The different options for the controller are described in Table 10. Homing procedure:

1. Enable the controllers by flip the lower left switch down.
2. Make sure *B2* is flipped down. This keeps the controllers in homing mode.

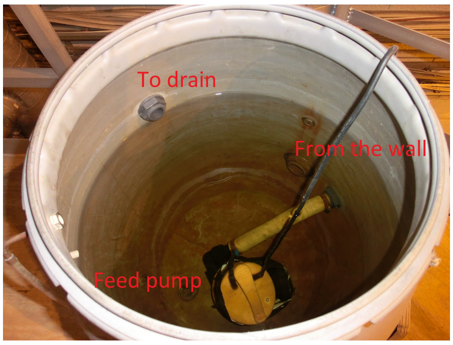


Figure 57: Tank

3. Flip *B1* and *B0* up to perform the homing procedure for the BHA and the choke.
4. Wait until the BHA and the choke has reached their positions and stopped moving.
5. Flip *B1* and *B0* back down again
6. Flip *B2* up to enable run mode and flip *B1* and *B0* to set BHA and choke in ready. The BHA and the choke are now ready to take input from SIMULINK.

The last step should not be performed before the experiment is ready to run. When the controllers are in homing mode the motors are locked. However if the BHA is in run mode, it will start to move downwards due to gravity. After homing the BHA is close to the lower limit switch and it will not take much time before it will be trigged. If the limit switch gets triggered the BHA needs to be pulled away (Section A.4) and the limit switch needs to be reset (Section A.5). Waiting to put the controllers into run mode will prevent this.

The motors can also be locked in between experiments to avoid the BHA moving downwards towards the lower limit switch. This done by flipping the *enable controller* switch up to off. As long as the electrical cabinet power is turned on, it is unnecessary to perform a new homing procedure before running experiments. When a new experiment is ready just remember to flip the *enable controller* back down again.

A.7 FILL WATER IN THE TANK

The tank is located at the top of the copper pipe. The water level should be as shown in Figure 57. If it is below this level, the crane should be turned on. Figure 58 shows where the crane is and how to turn it on and off. After the crane has been turned on and water starts to flow out through the drain, the crane should be turned off.

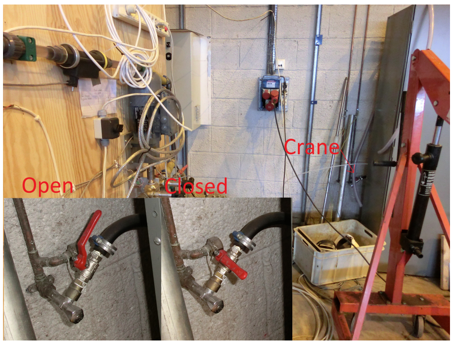


Figure 58: Crane

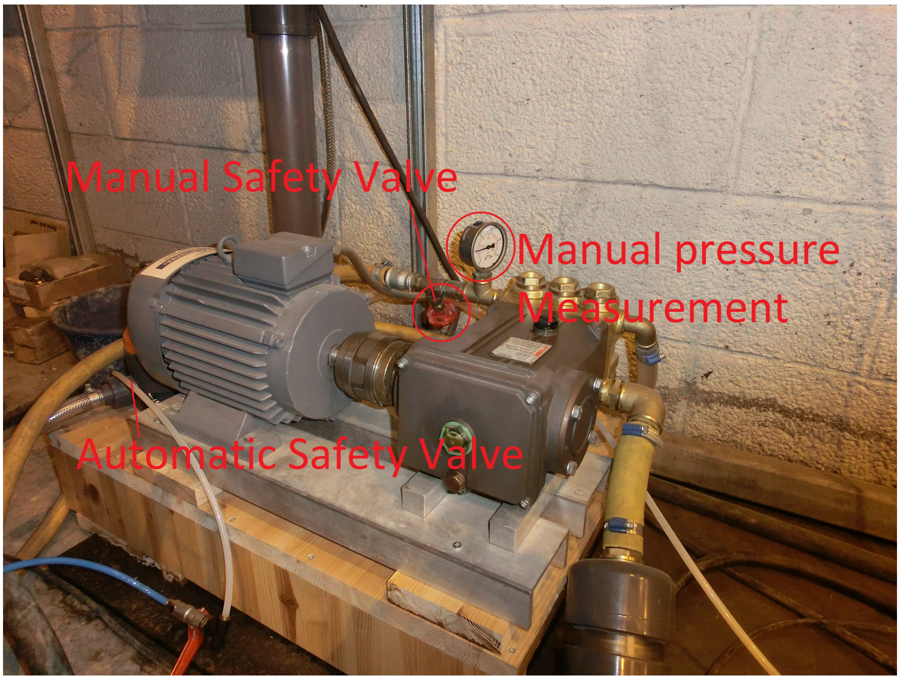


Figure 59: Backpressure pump with the pressurizing valve.

A.8 PRESSURIZE SYSTEM WITH BACKPRESSURE PUMP

In order to perform certain experiments, it might be desirable to pressurize the system. There are multiple ways of doing this, but the simplest way is as follows:

1. Open the choke and make sure MV4 and MV5 are open. MV7 should be closed

2. Set the Backpressure pump to desired velocity.

3. Set the choke opening in SIMULINK that corresponds to the desired pressure in the system.

It is also possible to use the MV4 or MV7 to pressurize the system.

A.9 PRESSURIZE SYSTEM WITH PRESSURE TUBE

If one desires to perform experiments on the pressurized system without the backpressure pump running, then the pressure tube should be used. This tube is positioned upright and is approximately 4m long with an inner radius of around 4cm. It is filled partially with water and partially with air. The tube is pressurized with an air compressor which is connected to the top end of the tube. The pressure setpoint is set to 5bar (gauge) as default. In order to use the pressure tube instead of the backpressure pump to pressurize the system, the following steps should be performed:

1. Make sure there is at least some water in the pressure tube. If the water level is too low, we risk that the air leaks from the tube and into the rest of the system. This may alter the system behavior and skew our results. If there is no water in the system, perform the steps in Section A.10 before performing the following steps.

2. If there is enough water in the tube, it is important that the following manual valves are closed in order to prevent leaks from the system: MV5, MV6, MV1.

3. Once these valves are closed, one may open manual valves MV2 and MV3.

4. In addition, the pressurized air inlet to the pressure tank should be opened.

Once these steps are performed, the system should be set up to a pressure setpoint of around 5bar. Once the testing is completed, perform the following steps in order to switch back to the original setup. For safety reasons, this should always be done.

1. Shut the manual valves MV2 and MV3

2. Open manual valves MV1, MV5, MV5.

3. Refilling the Pressure Tube

A.10 REFILLING PRESSURE TUBE WITH WATER

If the water level in the pressure tube is not sufficiently high, we need to refill it with water. This is most easily done using the backpressure pump. In order to do this, the following steps should be performed:

1. For safety reasons, make sure the following manual valves are open: MV6, MV5, MV1.

2. Make sure the following manual valves are closed: MV2, MV3.

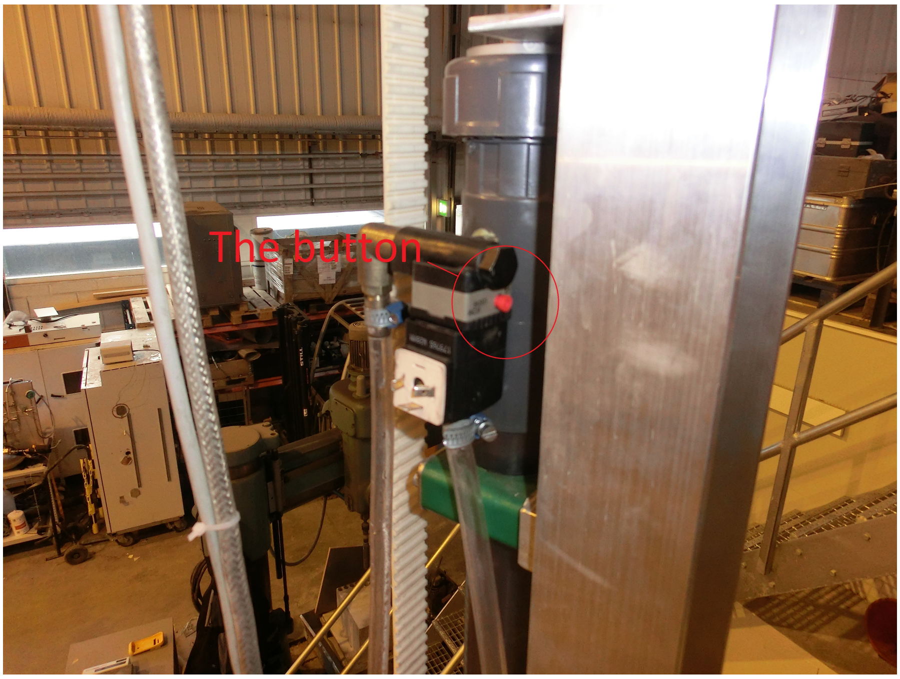


Figure 60: Button for Releasing Air from the System

3. Turn on the backpressure pump to around 30% power.
4. Gradually close MV1 until the manometer on the backpressure pump shows approximately 7bar. Leave MV1 in this position. Now water should flow up into the water tank above MV1, whilst the pressure in the system is higher than that in the pressure tube.
5. Open MV2.
6. Gradually open MV3 until the desired water level is reached. A water level of 30cm should be sufficient provided that there is no significant leak in the system.
7. Close MV3.
8. Open MV1 and turn off the backpressure pump.

A.11 REMOVING AIR FROM SYSTEM

Inevitably some air will find its way into the system. To remove this air the system should be pressurized (Section A.8 or Section A.9) and the red button close to the piston, marked in figure 60, should be pushed.

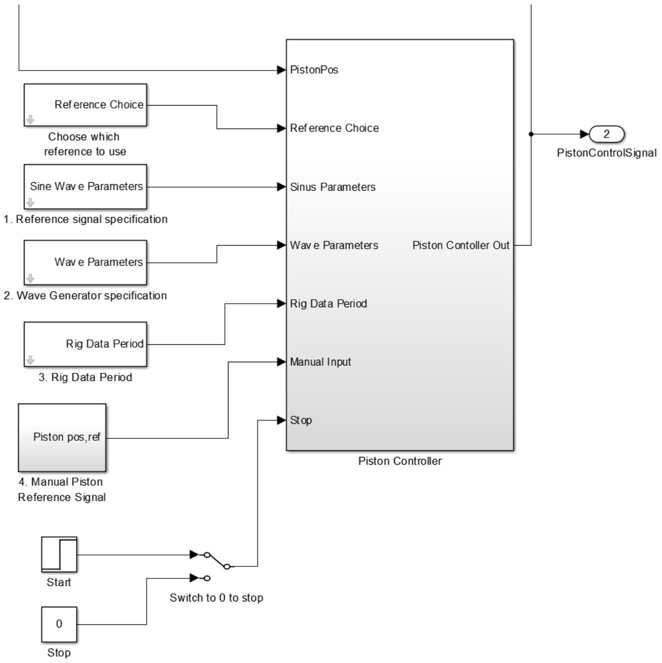


Figure 61: BHA SIMULINK with three prefabricated reference choices.

A.12 BHA SIMULINK

A controller for the BHA is made in SIMULINK. This controller has four choices for reference signal (the choice is specified by number in the *Reference Choice* block shown in Figure 61):

1. *Sine Wave Parameters*: This enables the user to set a reference signal as a combination of 1-3 sine waves.

2. *Wave Parameters*: This set the reference signal to follow a generated series of waves based upon a JONSWAP spectrum with user defined parameters. See Table 11 for proposed values. The filter is found in the subsystems for the Piston control.

3. *Rig Data Period*: This sets the dominant reference signal to follow a scaled down real rig series. The user specifies the dominate wave period of the wave series.

4. *Manual Piston Reference Signal*: This enables the user to set a manual reference signal for the choke. The signal should be between 0 and 80 cm, where 0 cm = bottom of well, 80 cm = top of well.

5. *Stop*: The BHA can be set to rest in case it is desired to only run the choke.

A.13 CHOKE SIMULINK

The choke opening is set by a signal in SIMULINK between 0 and 90 degrees. 90 degrees correspond to open choke and 0 degrees to closed. Figure 62 shows the setup for in SIMULINK for the choke with controllers. The manual switches enable the user to choose between a constant choke opening, specify a reference signal for the choke to track, or to use the choke pressure controller. The choke pressure controller can both take in a user specified reference signal or reference signal from the bottomhole controller.

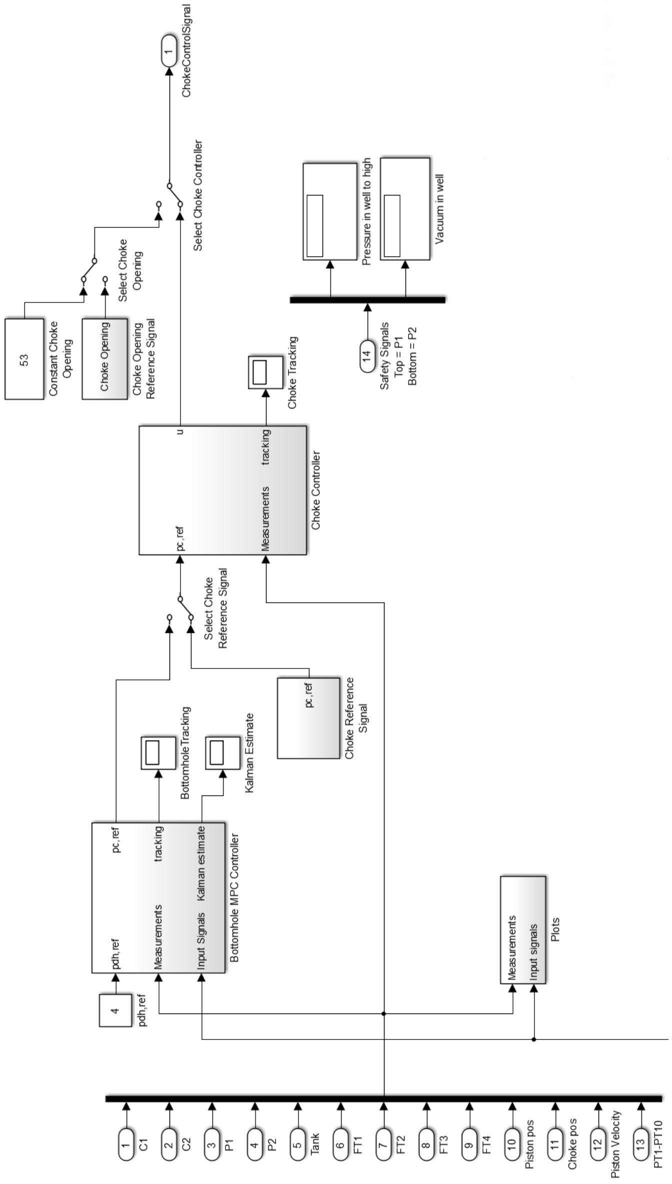


Figure 62: Choke SIMULINK

Filter ω_0 [rad/s]	Period [s]	Wave intensity σ
0.1	3 6 15	1100 1000 600
0.5	3 6 15	200 200 200
1	3 6 15	100 125 150
2	3 6 15	75 90 150
5	3 6 15	50 60 150

Table 11: List with suggestions to σ with chosen filter and period. The seed in the uniform random number block is set to 0. If the Seed or the C-compiler is change these values are not valid.

B LENZE SOFTWARE (ENGINEER) DOCUMENTATION

The software from Lenze used to control the electrical motors controlling the piston and the choke is called Engineer. This software has numerous possibilities for setup for making the motors perform a whole range of tasks. It follows that the software has a quite high level of complexity. None of the students working on the lab have a extensive knowledge of how the software works, and that was the reason assistance from Lenze was hired to help set up the controllers (see Section 4.2.2.2). However some knowledge have been gained from hiring the assistance, and a few simple tasks can be done. The procedures are contained in this appendix .The part from Appendix A about Engineer will be repeated here for completeness.

B.1 STARTUP OF ENGINEER

1. Plug the network cable into the controller you want to access the setting for (piston/choke). See Figure 63.

2. Startup Engineer by double clicking on the *L-Force Engineer High-Level* icon on the desktop

3. In the startup screen you get the choice between opening an existing Engineer project or make a new project. Select open an existing project.

4. For the choke you chose *LE1_Anders* and for the piston chose *LE2_Anders*.

5. Once the project is opened up click the *Go online* button (marked in Figure 64. Your screen should now look like Figure 64.

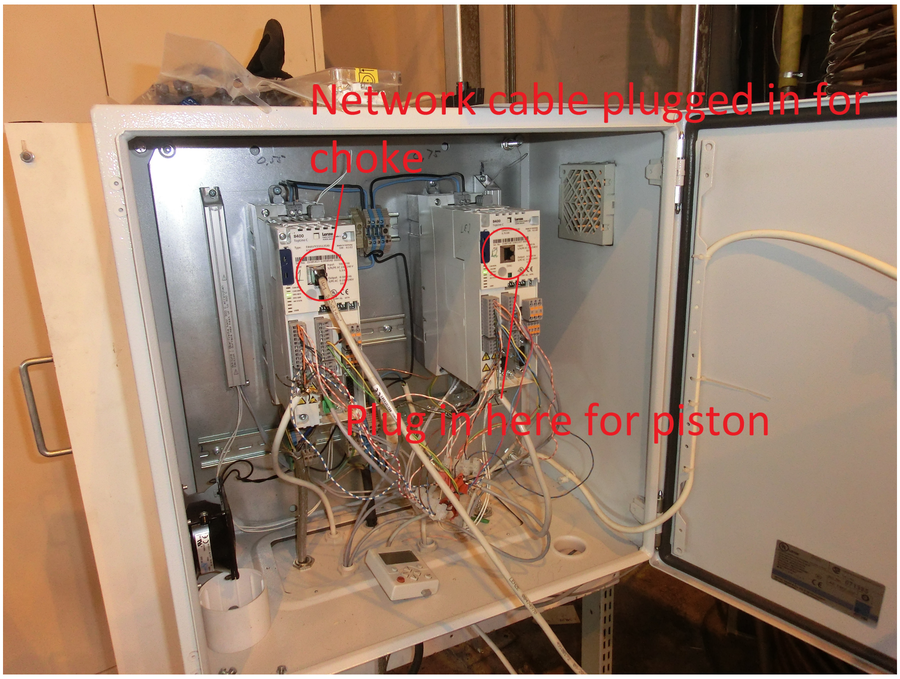
6. You are now ready to do changes in the software for the controller.

B.2 RESET LIMIT SWITCH

When the piston hits the lowest limit switch the motor will be locked. To reset this error do the following procedure:

1. Turn off the controller by flipping the *Enable controller* switch on the control box. (see Figure 8).

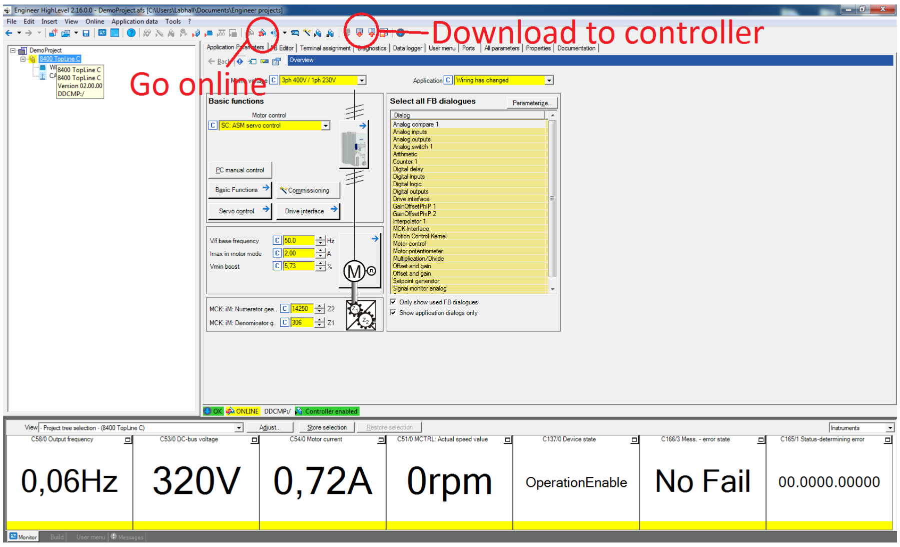
2. Grab the belt connected to the piston and physically pull the trolley away from the limit switch.



Network cable plugged in for choke

Plug in here for piston

Figure 63: How to plug in network cable



Download to controller

Go online

Engineer HighLevel 2.16.0.0 - DemoProject.afs [C:\Users\Labhall\Documents\Engineer projects]

Mains voltage: C 3ph 400V / 1ph 230V

Application: C Wiring has changed

Motor control: C SC: ASM servo control

V/f base frequency: C 50,0 Hz

Imax in motor mode: C 2,00 A

Vmin boost: C 5,73 %

MCK: iM: Numerator gea..: C 14250 Z2

MCK: iM: Denominator g..: C 306 Z1

☑ Only show used FB dialogues

☑ Show application dialogs only

OK ONLINE DDCMP:/ Controller enabled

View: - Project tree selection - (8400 TopLine C)

C58/0 Output frequency	C53/0 DC-bus voltage	C54/0 Motor current	C51/0 MCTRL: Actual speed value	C137/0 Device state	C166/3 Mess. - error state	C165/1 Status-determining error
0,06Hz	320V	0,72A	0rpm	OperationEnable	No Fail	00.0000.00000

Figure 64: Startup screen loaded project

3. In Engineer click on the *Diagnostics* tab.

4. In the lower right corner there should be a button called *Reset Error*. Click this button

5. The status display at the bottom called *C166/3 Mess. - error state* should now change from *Ck02: Neg. HW-LimitSwitch* to *No Fail*.

B.3 SAVE CHANGED SETTINGS TO CONTROLLER

After doing some changes in the controller of the piston or choke the new settings needs to be saved to the controller before any changes are made.

1. Click on the *Download parameters to device* button marked in Figure 64.

B.4 CHANGE CHOKE HOMING POSITION

After the homing procedure is completed the choke should go to open position and the piston should be at the bottom position. If this is not the case it can be changed following:

1. Click on the tab *Application Parameters* (this may already be selected since it is the default choice).

2. Warning: Do not change the Motor control scroll down menu or the Application choice at the upper right corner. Doing either of these will reset the setting for the controller back to default without giving a warning.

3. Click on the *Basic Functions* button within the *Basic Functions* frame.

4. In the center of the frame click on the box *Homing*.

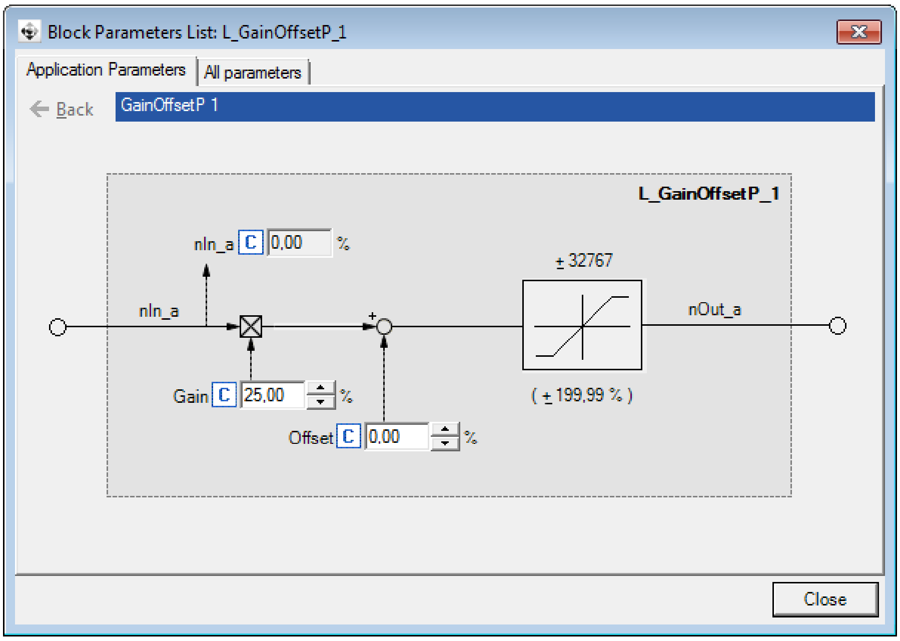
5. The homing position can now be changed by changing *Ref. reference offset*. If you have the motor in homing mode it will change as this value changes.

B.5 CHANGE CHOKE OPERATING RANGE

It has been experience that the choke suddenly changes its operating range.

1. Click on the *FB Editor* tab

2. In the FB Editor the complete setup for the controller is represented as different boxes with connections in between. Similar to the SIMULINK toolbox in MATLAB.



nIn_a C 0,00 %

Gain C 25,00 %

Offset C 0,00 %

Figure 65: L_GainOffsetP_1

3. Scroll down to you find the box named *LS_AnalogInput*. It should be located to the left part of the model.
4. The box is connected to another box named *L_GainOffsetP_1*. By clicking on the parameter list at the upper right corner of this box you will get a screen like shown in Figure 65.
5. By adjusting the gain and the offset the operating range of the choke can be modified.

C MATLAB FILES OVERVIEW

No MATLAB code is included in this appendix. However, all the MATLAB files and SIMULINK models used in this thesis are on a CD attached to the report. In this chapter a overview of the files on the CD is given. The chapter is divided into one section for each folder on the CD.

C.1 HELP FUNCTIONS

This folder contains a set of support functions used by multiple files. It is suggested to add it to the MATLAB path when working with the other files on the CD.

The most important functions (the remaining functions are self-explanatory):

- *signalNoFilter* This function is used to take in a dataset saved from a lab experiment and return the measurements in the following structure: [t,choke,piston,flow,tpressure,bpressure,pipe].
- *plotPcTracking* Plots the reference signal and the choke pressure for a lab experiment.
- *plotPdhTracking* Plots the reference signal and the bottom pressure for a lab experiment.

C.2 CHOKE

The folder contains all the functions used in the chapter about the choke.

- *chokechar_020513.mat* A choke characteristic test. The choke has been slowly ramped up and down.
- *findChar* Takes in a choke characteristic test (as for example *chokechar_020513.mat*) and calculates the choke characteristic using polyfit() using a polynomial order set by the user. The result is written to a *.txt* file for easy copy-past into a SIMULINK look-up table.
- *plotRes* Takes in a couple of test for the choke controller and plots the results
- *PID_impl* This is a SIMULINK model used to test out the choke controller for some user-set input signals.

- *initPID_impl* The initialize function called by *PID_impl* at startup. It task is to load the filters into workspace for *PID_impl*.

C.3 IDENTIFICATION

The folder used in the identification chapter.

- *dsr* The folder containing the DSR-algorithm implementation. It is recommended to add this folder to the MATLAB path.
- *overschee* The folder containing the Overschee-algorithm implementation. It is recommended to add this folder to the MATLAB path.
- *runIdentification* This is a script containing a framework for identifying a model based up on the procedure described in Chapter 6. Running the script with maxOrder 75 will take approximately two days.
- *postProcessingIdentification* This *.m-file* takes in the output from the *runIdentification* file and plots the best models identified for each setup and for each order. It enables the user to easily chose between the potential trade-off between model order and accuracy.
- *runSingle* When a model has been decided upon from *postProcessingIdentification*, the input can be put into this file and it will find the specific model, plot a demo of it and store the result in a *.mat-file*.
- *viewTimeseries* Takes in a set of lab experiments and plot the bottomhole pressure for the whole set.

C.4 LQG

This folder contains the implementation of the LQG-controller described in Section 7.1.2.

- *model4_exclDelay.mat* This is the model identified, which the LQG-controller is based upon.
- *LQG_model.slx* This is a SIMULINK implementation of the LQG-controller with the model identified.
- *initandRunLQG* This file uses the model (*model4_exclDelay.mat*) and implements LQG-controller. After implementing the controller the SIMULINK implementation is run through twice: Once with the controller and once without. A plot comparing the two results are made and stored in a *.png-file*.

C.5 MPC

This folder contains both the MPC-controller implementation in SIMULINK and the experiments from the lab. The experiments from the lab as well as the processing files are stored in the folder *LabResults* and in the following list the files in that folder will be marked accordingly

- *model_28-May-2013.mat* Model used by the MPC-controller.
- *mpc_online* The MATLAB level 2 S-function implementation of the MPC-controller.
- *setupMPC* This file calculates the offline parameters for the MPC-controller and stores them in *mpcSetup.mat* and *mpcSamleTime.mat*.
- *mpc_impl.slx* This is the SIMULINK implementation of the MPC-controller with the model identified.
- *initandRunMPC* This file sets up the harmonic oscillator for disturbance estimation and the kalman filter for state estimation. Then the file runs through the SIMULINK implementation, *mpc_impl.slx*, twice: Once with MPC-controller and once with constant pressure. The results are plotted together and stored in a *.png-file*.
- *LabResults\demoRes* This file takes in three different lab experiments with disturbance periods of 3, 5 and 10 seconds. The results are plotted and stored.
- *LabResults\pcTrackingAbility* This file takes in the lab experiment with a disturbance of 3 seconds and illustrates the suppression ability of the MPC-controller.
- *LabResults\testEstimators.slx* This SIMULINK model is used to illustrate the predictive ability of the kalman-filter and the harmonic oscillator.
- *LabResults\testModelPrediction* This file uses the SIMULINK model *testEstimator* to illustrate the predictive ability of the kalman-filter.
- *LabResults\testModelPrediction* This file uses the SIMULINK model *testEstimator* to illustrate the predictive ability of the harmonic oscillator.

C.6 SIMULINK MODEL LAB

This folder contains the SIMULINK model used with the lab and all the help functions.

- *interfaceIPT_lab_8may2013.slx* This is the SIMULINK model used with the lab to obtain all the lab experiments used in this report. For more information on how it works see Section 3.4, Section A.12 and Section A.13.

- *setupMPC* Is the same file as in the MPC folder. Tuned for the lab.

- *mpc_online* Is the same file as in the MPC folder.

- *inirModel* This file gets called by *interfaceIPT_lab_8may2013.slx* each time it is run. It sets up the harmonic oscillator and kalman-filter used in the experiments. It also loads *biasFilter.mat* which contains the filters and biases used in the experiments. Finally it sets up the ramp function used to identified the choke characteristics.

D

ARTICLE UNDER PREPARATION FOR SUBMISSION

Suppression of Heave Disturbance in Lab for Offshore Drilling with Model Predictive Controller

Anders Albert
Department of Engineering Cybernetics,
Faculty of Information Technology, Mathematics
and Electrical Engineering, Norwegian University of
Science and Technology (NTNU),
Trondheim, Norway

***Abstract*—Fewer and fewer new drilling prospects are discovered on the Norwegian Continental Shelf. The hydrocarbon production is declining as a result. A way to continue producing is to drill more wells in fields and environment that were deemed to be undrillable originally. The fields deemed undrillable may be due to scenarios such as narrow pressure window, depleted fields, extreme loss circulations etc. Managed Pressure Drilling is a set of techniques that allows drilling in such difficult scenarios. Drilling in the North Sea also provides extra challenges due to harsh environment. During a connection scenario when drilling from a floating rig, the heave motion can induce large surge and swab pressures in the well. This will be referred to as the heave problem.**

Multiple papers have been written on using a technique called constant bottomhole pressure to suppress the heave problem. The technique consists of sealing off the annulus and using a choke to control the pressure of the well. In order to have a realistic environment to test out controllers in, an experimental lab has been built to model the connection scenario. In this paper the experimental setup of the lab will be explained. The assumptions and simplifications are discussed. A system identification has been performed to identify a linear model of the lab and an MPC-controller has been designed based on this model. The MPC-controller has been demonstrated to suppress a disturbance, simplified to a single sine wave of period 3 seconds, approximately 46 %.

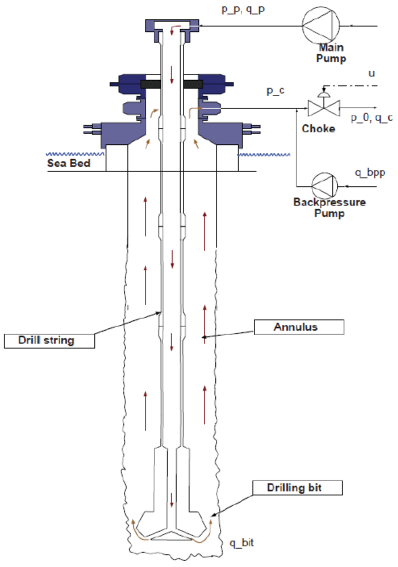


Fig. 1. CBHP setup [5]

I. Introduction

Managed Pressure Drilling (MPD) has become an increasingly popular set of techniques for drilling wells [1], [2]. During drilling operations a fluid, called mud, is pumped down through the hollow drillstring. The muds main purpose is to transport rocks cut lose by the drillbit, called cuttings, up to the surface through annulus. The target for applying MPD is to control the pressure profile of a well, thus controlling the pressure of the mud. During normal drilling operations the pressure has to stay within certain environmental limits. First it has to stay above the pore pressure. If the pressure drops below the pore, pressure influx can be experience that can lead to a kick, which in worse case become blowout. Second the pressure must be lower than the fracture pressure. If this pressure is exceeded, it may fracture the well which can result in the whole well becoming lost.

One such MPD technique is called constant bottomhole pressure (CBHP). In conventional drilling the pressure of the well depends on two factors: The density of the mud and the friction loss due to circulation of the mud. With CBHP the annulus is sealed off and the flow back from the surface is controlled by the choke. This enables the use of a mud with a lower density since the pressure of the well now also depends on the pressure drop over the choke. Figure 1 illustrates the CBHP setup.

An additional challenge for controlling the pressure of the well is provided when drilling from a floating rig. A floating rig will move up and down with the waves, a motion called the heave. During normal operation this movement is compensated for by a heave compensator. However, when the drillstring is in slips for making connection (extending the drillstring), the drillstring is connected to the rig. This leads the drillstring to move in and out of the hole. This creates pressure fluctuations called surge (moving into hole) and swab (moving out of the hole) pressures. These pressure fluctuations can be as large as ± 24 bars [3]. This problem will be referred to as the heave problem in this paper.

Multiple papers have been written on using CBHP to suppress the pressure fluctuations from the heave problem [4]–[6]. In order to have a realistic environment to test out these algorithms a lab has been built to model the connection scenario for a floating rig with CBHP.

This paper presents the setup together with the simplifications and assumptions for the lab. The lab is then used to identify a linear model, which is used to design an MPC-controller. The MPC-controller is then demonstrated to suppress disturbances by approximately 46%.

II. Design of Lab

A. Design and Simplification

The lab has been built in order to model a connection scenario. During a connection scenario the main pump will normally be turned off. To ensure circulation through the choke when using CBHP an extra pump, called backpressure pump, is added. When making a connection the main pump will be slowly ramped down, while the backpressure pump will slowly be ramped up. In addition the heave problem will occur only when the drillstring is in slips.

Three main difficulties with creating the lab were encountered. First it is difficult to make the drillstring follow a given path. Second, even a scaled down bottomhole assembly (BHA) and drillstring will cause huge pressure fluctuations when moving up and down the well, requiring very expensive equipment to withstand the pressure. Finally high pressure in the well could easily lead to bending of the scaled down drillstring.

The ramping down the main pump and up the backpressure pump scenario was not included in the lab due to several advantages that this result in. First the main pump does not need to be included. Also no circulation of mud occurs, thus the drillstring can be solid instead of hollow, making it more stable.

To model the BHA a single cylinder with increased diameter compared to the drillstring was used. In addition to being connected to an upper rod, modeling the drillstring, it was also connected to a lower rod. The result is that the bottom of the well looks like a piston. This solves several of the difficulties mention above. First having both and upper and lower rod make it easy to follow a given path. In addition the pressure in the well will be much smaller and can be scaled down by varying the diameter of the BHA cylinder. Also having the BHA connected to two points, only make it necessary to have a pulling mechanism instead of both a pulling and pushing mechanism to move it.

The piston modeling of the BHA provides a new problem. The control system is likely to need some fluid displacement in order to detect the pressure fluctuations from the surface of the well. To accommodate the lower rod was given a smaller diameter than the upper rod.

The drillstring itself has been assumed to not contribute much to the pressure fluctuations by just moving up and down though the well. This makes it unnecessary to include in the lab model, which is essential to building the lab in a space restricted area. The well itself has been built as a pipe that has been coiled in order to save space.

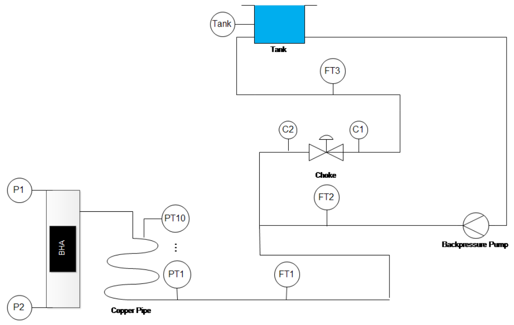


Fig. 2. Schematics of Lab

The well was thus separated into two components: A long copper pipe, used for modeling the time delay from the surface to the bottom of the well, and the bottom of the well modeled by a PVC pipe. Using a transparent PVC pipe has the advantage of easily being able to see the movement of the BHA cylinder that is black.

In order to model the disturbance the BHA with rods were connected to a sawtooth belt. The belt was then controlled by an electrical motor. Using the electrical motor makes it possible to move the BHA straight up and down in the PVC pipe.

The choke was tailor made for the lab. It consists of an electrical motor controlling the movement of a choke valve.

The heave disturbance was simplified to be a single sine wave. This was done to estimate pressure drops before the lab was built and in the initial controller. The lab has the ability to create more complex disturbances if desired later on.

Water has been used for mud. A closed circulation system has been setup with a tank and a backpressure pump. The resulting lab sketch can be seen in figure 2. Flow measurement have been marked with FT. The pressure measurement around the choke are marked with C2 and C2, in the bottom of the well P1 and P2, while the pressure measurements in the copper pipe are marked PT1-10.

Two control cards have been used for receiving and setting actuator signals. These control cards have been connected to a computer and the MATLAB toolbox SIMULINK has been used to process measurements and implement controllers for the lab in real-time.

B. Scaling

The lab designed is based upon a real vertical well. The scaled down measurements of the lab and the measurements from the real case the lab is based upon can be found in table I. In the table the lab component corresponding to the real well component is given in parenthesis. In addition some of the components which are only in the lab are also given in the table.

When scaling down the lab the ratio between the cross-sectional area of the drillstring and the well was kept constant,

TABLE I
SCALING OF THE LAB

Component	Real well	Lab
Well length (copper pipe)	4000m	900m
Copper pipe diameter	-	16 mm
Well diameter (PVC)	8.5”	42.53 mm
PVC pipe length	-	80 cm
Drillstring (upper rod)	5”	25 mm
Lower rod	-	22 mm
BHA diameter	6.5”	40.9 mm
BHA length	70m	35 mm
Disturbance period	11 s	3 s
Disturbance amplitude	1.5 m	40 cm

$\frac{8.5^2}{5^2} = \frac{42.53^2}{25^2}$. The BHA was down scaled in order to obtain a desired pressure drop around ± 2 bar in the well.

The length of the well was set to 900 meters. It was desired to have the lab able to create disturbances of period about 5 times the time delay for the pressure wave to propagate from the top of the well to the bottom. The delay was calculated by assuming that the pressure would travel with the speed of sound in water at 25 degrees, which is 1498 m/s. The fastest sine period the lab should handle then becomes $\frac{900m}{1498m/s} \approx 3$ seconds.

In addition it was desired to keep the velocity of the drillstring from the real case. A sine wave of 11 second period and 1.5 meter amplitude has a maximum velocity of 0.856 m/s. To obtain this velocity with a 3 second period the amplitude was set to 40 cm. Thus making the length of the PVC pipe 80 cm.

Finally the lab was designed for an operating pressure between 0- 10 bar.

III. SYSTEM IDENTIFICATION

In the approaches used by [4]–[6] the modeling equations of the connection scenario simplify to a set of linear equations with the exception of the choke valve. To avoid the nonlinearity in the choke, the control problem was separated into two parts. First a choke controller was developed for obtaining a desired pressure at the choke. Second a bottomhole controller was designed, which use the pressure at the choke in order to suppress the heave disturbance.

In order develop controller some models for the choke and the lab had to be developed.

A. Choke modeling

Based upon the Bernoulli equation for steady flow [7] the following equation for pressure drop over the choke can be obtained:

$$q_c = G(u)\sqrt{\frac{p_c - p_0}{\rho}} \tag{1}$$

where q_c is the flow through the choke. p_c and p_0 is the choke inlet and outlet pressure. ρ is the density of the fluid and G(u) is a strictly increasing characteristic function individual to the choke valve, which depends on the choke opening, u.

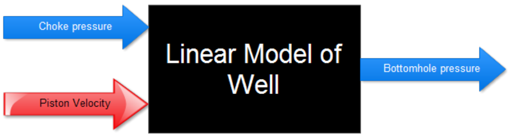


Fig. 3. Black Box Model of Lab

The characteristic function was found by slowly closing the choke opening from open until a maximum pressure was obtained in the system. Then the choke opening was slowly opened again. The MATLAB function polyfit() was used for identifying a characteristic equation for the choke:

$$G(u) = 0.01u^3 - 0.80u^2 + 44.29u - 790.82 \tag{2}$$

B. Lab Model

For identifying a model of the system a black box approach was used. In the lab there are two inputs into the system: the BHA velocity and the choke position. The choke controller is supposed to replace the choke position with the choke pressure. Thus the resulting model becomes as shown in figure 3.

A linear model was assumed for the lab. Then a subspace identification algorithm called DSR [8] was used to identify a discrete system on the form:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Ed_k \\ y_k &= Cx_k + Du_k + Fd_k \end{aligned} \tag{3}$$

y is the bottomhole pressure, u is the choke pressure and d is the velocity reference for the BHA.

To identify a the model described in equation (3) the BHA and the choke inputs were set to follow two different sets of sine waves. In the DSR algorithm the user selects the order of the system based on a singular value decomposition. The singular value decomposition depends on a Hankel matrix. The order of the Hankel matrix is set by the user. Multiple combinations were attempted for Hankel matrix order and system order. A 4th order model was identified and considered to be sufficiently accurate.

C. Disturbance Modeling

In order to suppress the disturbance an estimate was needed. In both [4], [5] the disturbance is modeled as a set of harmonic waves. In simulations both papers simplify the disturbance to a single sine wave. This is also the simplification done when calculating the pressure drop done when designing the lab. Thus the disturbance was modeled as a harmonic wave:

$$\dot{x} = A_d x \qquad y = C_d x \tag{4}$$

where

$$A_d = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} C_d = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

This models assumes a priori knowledge of the frequency of the disturbance.

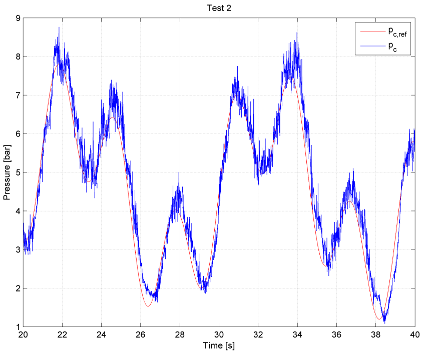


Fig. 4. Choke Controller Tracking Ability

IV. Controller

A. Choke Controller

In order to control the pressure at the choke a PI-controller with feedforward and gain scheduling for the integral constant was designed. The feedforward part of the controller was found by solving equation (1) for the choke opening , u. The gain scheduler was made by a function having two inputs. First a lower gain was used for higher pressures, since a small change in the choke opening at high pressures results relatively large pressure change. In addition the gain was made smaller if the pressure deviates much from desired pressure.

Figure 4 shows the choke controllers tracking ability for a reference signal consisting of three sine waves.

B. Bottomhole Controller

When designing a bottomhole controller a Model Predictive Controller (MPC) was chosen. This controller has several advantages when it comes to suppressing a heave disturbance:

- It can handle constraints
- It can easily utilize knowledge of disturbance

Unfortunately an MPC-controller demands much computational power. This was solved by increasing the times between samples for the MPC-controller.

In implementing an MPC-controller a quadratic problem gets solved on the form:

$$\min_{v} 0.5v^T\tilde{H}v + c^Tv \qquad (5)$$
$$\text{subject to}$$
$$Lv \leq b$$

To solve the quadratic problem the MATLAB function quadprog() was used with an active-set algorithm [9]. The objective function for the MPC-controller was chosen to be:

$$\begin{aligned}\min_{u} f(x,u) = &\sum_{i=0}^{n-1}\{(x_i - x_{ref,i})^T C^T Q C(x_i - x_{ref,i})\}+ \\ &\hspace{20em}(6)\\ &\sum_{i=l}^{n-1}\{(\Delta u_i - \Delta u_{ref,i})^T P(\Delta u_i - \Delta u_{ref,i})\}+ \\ &(x_n - x_{ref,n})^T C^T S C(x_n - x_{ref,n})\end{aligned}$$

where Q, P and S are weight matrices. Instead of using the output, y_k, as optimization variable, the system state was used. This eases the implementation. The weight matrices, Q and S, can easily be changed to weigh the output variable by multiplying them by C from equation (3). This can be done since, as seen later, D and F from equation (3) are set equal to zero. When using the state as optimization variable, the reference output cannot be used directly and a reference state has to be calculated. The reference state was calculated from the reference output: x_{ref} = pseudo inverse(y_{ref}). Finally n is the controller horizon for the MPC-controller.

This objective function will attempt to drive the reference to a desired state and try to avoid changing the input. These are competing objectives and using this objective function enables the use of weighting the different objectives. The constraints used along the objective function:

$$\begin{aligned} x_0 &= \text{given} && && (7)\\ \Delta U_L \leq \Delta u_i &\leq \Delta U_U && \text{for } 0 \leq i \leq n \\ Y_L \leq C x_i &\leq Y_U && \text{for } 0 \leq i \leq n \end{aligned}$$

In addition the system equation were used as constraint with some modification:

$$\begin{aligned}\tilde{x}_{k+1} &= \tilde{A}\tilde{x}_k + \tilde{B}\Delta u_{k-l} + \tilde{E}d_k \qquad (8)\\ \tilde{y}_k &= \tilde{C}\tilde{x}_k\end{aligned}$$

where

$$\tilde{A} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} B \\ I \end{bmatrix}, \quad \tilde{E} = \begin{bmatrix} E \\ 0 \end{bmatrix}$$
$$\tilde{C} = \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}, \quad \tilde{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ u_{k-l} \end{bmatrix}, \quad \tilde{y}_k = \begin{bmatrix} y_k \\ u_{k-l} \end{bmatrix}$$

There are multiple reasons for changing the system equations. First D and F from equation (3) are set equal to zero. This introduces some inaccuracy to the model, but it saves online computation time. Second the input was changed to change in input. This is necessary to implement the objective function, but it also enables putting constraints on the input change. Third, with the last change, the ability to put constrains on the input is lost. By expanding the output states to also include the input it is still possible to put constraints on it. Finally the choke controller introduces an extra time delay into the system in addition to the time delay from the surface to the bottom of the well. This time delay will always have to be estimated, thus it is included into the model as a tuning parameter, l. When the l parameter is set to $l > 0$ the following is given: $l \leq k \Rightarrow \Delta u = 0$.

TABLE II
MPC TUNING PARAMETERS

Tuning parameters	
sampleMPC	0.2
n	30
y_{weight}	1000
$y_{finalstate}$	2000
u_{weight}	1
Δu_{weight}	800
Delay from $p_{c,ref}$ to p_c	0.6 s

TABLE III
MPC CONSTRAINTS

lower	parameter	upper
-0.6	Δu	0.6
-Inf	Y	Inf
0	U	10

These equations can, by using some calculation, be included into a problem on the form in equation (5). The resulting MPC-controller have the tuning parameters and constraints that can be found in table II and III. The sample time for the MPC controller is selected as a tuning parameter in order to have the possibility to adjust it accordingly to the controller horizon. The table also contains the values chosen for the MPC-controller used in the lab. The weight matrices Q, P and S are calculated as follows:

$$Q = \begin{bmatrix} y_{weight} & 0 \\ 0 & u_{weight} \end{bmatrix}$$
$$S = \begin{bmatrix} y_{finalstate} & 0 \\ 0 & u_{weight} \end{bmatrix}$$
$$P = \Delta u_{weight} \tag{9}$$

No constrains were used for the output, the bottomhole pressure, as can be seen in table III. Having a constraint on this state can easily lead to an infeasible quadratic problem. Also having constraints on the input, the choke pressure, ensures the bottomhole pressure.

The MPC-controller was implemented using a MATLAB level-2 S-function with three inputs: The desired bottomhole pressure, the disturbance state and system state. The disturbance state was used to predict the disturbance for the controller horizon. The equation state was calculated from a kalman filter based upon equation (3).

V. RESULTS AND DISCUSSION

The MPC-controller was used to suppress sine waves of period 3, 5 and 10 seconds. Table IV contain the approximated suppression of the different disturbances while figure 5, 6 and 7 contain the suppression of 3, 5 and 10 seconds disturbance.

As can be seen in table IV it is more difficult to suppress faster varying disturbances than slower. This comes as no

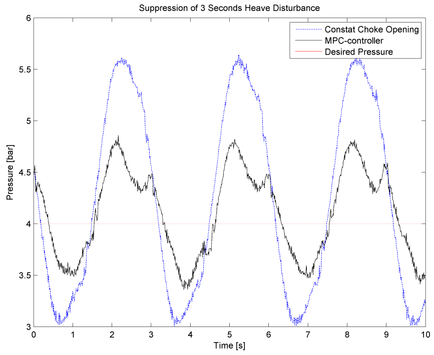


Fig. 5. Suppression of Heave Disturbance with 3 Second Period

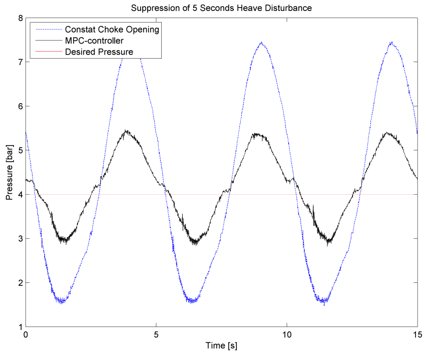


Fig. 6. Suppression of Heave Disturbance with 5 Seconds Period

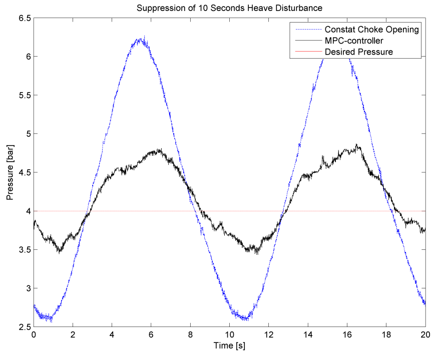


Fig. 7. Suppression of Heave Disturbance with 10 Seconds Period

TABLE IV
SUPPRESSION OF SINE WAVE WITH REFERENCE PRESSURE EQUAL TO 4 BAR

Disturbance	Without MPC [bar]		With MPC [bar]		Suppression
Period [s]	Low	High	Low	High	in %
10	2.6	6.2	3.5	4.8	63.9
5	1.5	7.4	2.9	5.4	57.6
3	3	5.6	3.4	4.8	46.2

surprise since for the faster varying disturbance the time delay from the surface to the bottom of the well will make up a larger part compared to the period of the disturbance.

Two main reasons have been identified explaining why not perfect tracking has been obtained. First the fourth order model identified was found to not be perfectly accurate. The inaccuracy of the model increased with faster varying disturbance and could in worst case be almost 2 bars off. A more accurate model would clearly increase the performance of the MPC-controller. In addition the choke controller does not manage to track the desired choke pressure perfectly. It has been observed, in worse case, to be 1 bar off.

The harmonic oscillator could potentially introduce an error when estimating the BHA velocity. However it was observed that it estimated the disturbance very accurately for a single sine wave.

Even if the MPC-controller had obtained perfect tracking of the desired bottomhole pressure, the lab would still be a simplification. In the real case the heave movement cannot be modeled by a single sine wave. As the MPC-controller is sensitive to model error, the estimation of the heave movement is essential to accurately suppress it.

In a real well the bottomhole pressure will probably have to be estimated. Thus the model used cannot use the bottomhole measurement to find the state of equation (3).

VI. Conclusion and Further Work

The heave problem has been presented. A lab designed to model the heave problem realistically has been discussed. A linear model has been identified of the lab, which has been used to design an MPC-controller. The MPC-controller has been demonstrated to suppress a heave disturbance of 3 second period by approximately 46 %.

There are multiple areas to focus on for further research. First a model based upon a physical principles should be attempted to be identified. Still, even a model based upon physical principles must expect to use some kind of estimation for estimating parameters like frictional loss in the pipe and the effective bulk modulus of the water in the pipe.

Once a physical model has been established a controller can be developed. The MPC-controller from this thesis could be implemented or some of the other controllers [4]–[6].

When a controller has managed to suppress a heave disturbance of a single sine wave using only measurements at the surface, a more complex disturbance should be attempted. Ideally no assumed knowledge of the disturbance should be used, unlike this paper where the frequency of the wave is assumed to be known a priori.

After successfully testing the suppressing a heave disturbance in the lab model steps can be taken to implement the solution for a floating rig in the North Sea.

References

[1] S. Nas and J. S. Toralde (2009), Offshore Managed Pressure Drilling Experiences in Asia Pacific, *IADC/SPE Drilling Conference and Exhibition, New Orleans, Louisiana, USA*

[2] D. Hannegan (2006), Case Studies - Offshore Managed Pressure Drilling, *SPE Annual Technical Conference and Exhibition, San Antonio, Texas, USA*

[3] O. S. Rasmussen and S. Sangesland (2007), Evaluation of MPD Methods for Compensation of Surge-and-Swab Preessures in Floating Drilling Operations, *IADC/SPE Managed Pressure Drilling and Underbalanced Operations Conference and Exhibition, Galveston, Texas, USA*

[4] H. Hessam, O.M. Aamo and A. Pavlov (2012), Attenuation of Heave-Induced Pressure Oscillations in Offshore Drilling Systems, *American Control Conference, Fairmont Queen Elizabeth, Montreal, Canada*

[5] I.S. Landet, A. Pavlov and O.M. Aamo (2012), Modeling and Control of Heave- Induced Pressure Fluctuation in Managed Pressure Drilling, *IEEE Transactions on Control Systems Technology PP* (99), 1 doi: 10.1109/TCST.2012.2204751

[6] O.M. Aamo (2012), Disturbance Rejection in 2 x 2 Linear Hyperbolic Systems, *IEEE Transaction on Automatic Control 58* (5), 1095 - 1106 doi: 10.1109/TAC.2012.2228035

[7] F.M. White (2008) *Fluid Mechanics* (6th edition) New York: McGraw-Hill

[8] D.D. Ruscio (1995) *Subspcae System Identification Theory and Applications* (1th edition) Porsgrunn: Telemark Institute of Technology

[9] J. Nocedal and S.J. Wright (2006) *Numerical Optimization* (2th edition) New York: Springer