



University of  
Stavanger

**FACULTY OF SCIENCE AND TECHNOLOGY**

## **MASTER'S THESIS**

Study programme/specialisation:

Petroleum Engineering / Natural Gas  
Engineering

Spring semester, 2020

Open

Author: Kent Arne Ask

Programme coordinator:

Supervisor(s): Kjell Kåre Fjelde

Title of master's thesis:

PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

Credits: 30

Keywords:

PI Regulation  
CBHP-MPD  
Simulation  
Connection  
Kick

Number of pages: 52

+ supplemental material/other: 27

Stavanger, 15.06 / 2020  
date/year

## Summary

An introduction to control engineering theory is given. The thesis further explores the implementation of a PI regulator in a transient flow model based on the AUSMV scheme. Optimal regulator parameters are important in CBHP-MPD when a control system is used to regulate an automatic choke. an overview of various MPD techniques is also given.

Simulations of connection and kick has been performed when a PI regulator is implemented in the transient flow model. The purpose of the cases was to see if the implemented PI regulator was able to maintain the bottom hole constant when disturbances such as pressure pulses, during a connection due to friction loss in the well. A second scenario looked at a slower process where kick is taken at the bottom. and due to expansion if it was possible to circulate different kick volumes through the MPD system.

The results show that for the pressure pulses disturbance the regulator manages to keep the bottom hole constant. Other findings are related to the change in viscosity and sampling time where there was need of tuning the PI parameters. The regulator also manages to keep the BHP close to the setpoint when different kick volumes was taken when water-based mud and free gas was simulated. Although in an MPD operation only small influxes can be circulated through the MPD system due to limitations on the RCD and mud gas separator.

## Acknowledgment

I would like to thank my supervisor Kjell Kåre Fjelde, your help and guidance have been extraordinary throughout the period of work. Thank you very much

I would also like to thank my family and friends for encouraging words and support over the years. A special thanks to my father who have been a good discussion partner.

Also, a special thanks to my girlfriend for all the support, patience, and love along the way.

## Table of content

Summary .....	2
Acknowledgment.....	3
Table of content.....	4
1. Introduction .....	6
2. Control engineering theory.....	7
2.1 Introduction to control engineering .....	7
2.2 Control engineering .....	7
2.3 Process disturbances.....	7
2.4 PID control theory .....	8
2.5 PID controller on discrete form.....	10
2.5.1 Discretization of each term.....	11
3. Integration of PI regulator .....	12
4. Ziegler Nichols' closed loop tuning method .....	13
4.1 Examples of how the Ziegler-Nichols' method was used .....	14
5. Introduction to Managed pressure drilling .....	15
5.1 MPD barriers .....	17
5.2 MPD techniques .....	18
5.2.1 Pressurized mud cap drilling (PMCD) .....	18
5.2.2 Dual gradient drilling (DGD) .....	19
5.2.3 Continuous circulation system (CCS) .....	19
6. Constant bottom hole drilling (CBHP).....	20
6.1 How to control pressure in the CBHP- MPD system.....	21
6.1.1 Choke valve.....	22
6.1.2 Automation during connections .....	23
6.1.3 Kick handling for the CBHP-MPD system .....	24
6.1.4 Influx management envelope (IME).....	25
7. Transient two-phase model based on AUSMV scheme .....	26
8. Simulation cases .....	32
8.1 Simulation of connection.....	32
8.1.1 No regulation.....	32
8.1.2 Simulation with PI regulator implemented and different viscosities.....	33
8.1.3 Simulation of different sampling interval.....	39
8.1.4 PI regulation with different pump start/close intervals.....	40

8.2 Simulation of Kick .....	42
8.2.1 Simulation with initial kick volumes 0,234m <sup>3</sup> , 0,948m <sup>3</sup> and 3,674m <sup>3</sup> .....	42
9. Discussion .....	48
10. Conclusion.....	50
11. References .....	51
12. Appendix .....	53

## 1. Introduction

When drilling at a small drilling window where the pressure difference between the pore pressure and fracture pressure is small, managed pressure drilling can be used. Different MPD techniques can be used where the CBHP-MPD technique is on off them. The constant bottom hole is obtained by closing off annulus with (RCD) rotating control device, making it a closed system. This closed system circulates the return flow through a choke valve creating a backpressure. During connections, a backpressure pump is used to maintain the flow across the choke such that the choke pressure can be increased to compensate for the lost well friction. (Saeed, Lovorn, & Arne Knudsen, 2012). Varying the flow through the choke and the choke opening creates a backpressure in annulus which will keep the BHP at desired pressure.

When an automatic choke is used, there is a need for a control system. The control system consists of a controller that controls the automatic components (i.e choke valve) with a control algorithm (Saeed et al., 2012). The algorithm is based on three terms (PID) which calculates the appropriate output signal. The controller needs to be tuned properly to make the controller stay in the scope of the desired set point.

This thesis looks upon how a PI regulator/controller can be implemented in a transient flow model solved by the AUSMV scheme and how the regulator parameters can be tuned to maintain constant bottom hole pressure when changes to the measured value occur. An introduction to different MPD techniques and control theory will also be presented.

Two simulations were performed. One connection case where the main pumps are turned off during a 10-minute connection to see if the regulation of the differential pressure of the choke managed to keep the desired BHP constant. Here there are acceleration and friction effects. In the second case, different kick volumes were taken and circulated through the MPD system. Here the hydrostatic pressure will change as the kick is circulated upwards and expands. The purpose for this test was to see if a pre-set of regulation parameters would be able to keep the bottom hole pressure constant when the gas was circulated out of the well.

PI regulator/controller will be used for the simulations. Water and ideal gas.

## 2. Control engineering theory

### 2.1 Introduction to control engineering

Control engineering or technology has been around for centuries. In fact, 2000 years b.c. the Babylonians constructed an automatic watering system based on level control (Haugen, 2012, p. 459). In 1868 James C. Maxwell performed mathematical analysis of speed control on steam engines. This was when the development of control engineering theoretical methods started (Haugen, 2012, p. 459).

In the 1930s, the automatic control technology developed further, and the P controller was developed for the process industry which later developed to PI and PID (proportional, integral, and derivative) controller. The PID controller is the most abundant type of controller used in the process industry to this day (Haugen, 2012, p. 460).

At this point there was no sufficient way to tune the controller, but then Ziegler and Nichols came along and developed a way to tune the PID controller (Haugen, 2012, p. 460). This is the method used in this thesis. A more elaborated description will be addressed in chapter 4.

### 2.2 Control engineering

In order to maintain control of a process one can use manual control where the operator physically operates an actuator (valve, pump) or automatic control where a controller, or advanced computer is programmed to keep the set point constant, or in other words the error close to zero. Error is defined by:  $e = \text{set point} - \text{measured value}$ . ( $e = SP - Mv$ ). Examples of process variables are level, temperature, flow, pressure, concentration, position, and speed. (Haugen, 2012, p. 15). For the simulations in this thesis, the process variable will be pressure when studying the CBHP-MPD system. The set point is the constant pressure at the bottom. ( $e = \text{constant bottom hole pressure} - \text{measured bottom hole pressure}$ )

### 2.3 Process disturbances

There are different process (pressure) disturbances that will make the bottom hole pressure deviate from its intended set point. During connections, the main pump and flow through drill string and annulus is stopped, and the annulus friction disappear. This causes a drop in bottom hole pressure. To avoid this, the backpressure pump is started inducing flow across the choke such that an increase in the choke pressure will compensate for the lost friction. The choke pressure will work as the actuator manipulating the simulated BHP.

Another example of a pressure disturbance is when a free gas kick enters the well and causes changes in the hydrostatic pressure in the well when the kick is circulated upwards and

expands. This reduction in hydrostatic pressure must be compensated by an increase in choke pressure. The effect of this will be shown in the kick simulation in chapter 8.2.

A third example of a pressure disturbance is a change in the flow rate from the main pumps. Changes in annulus friction, will have to be compensated by a change in choke pressure to maintain the bottom hole pressure at the set point.

Another pressure disturbance could be changes in the effective downhole mudweight due to pressure and temperature effects which would cause changes in the hydrostatic pressure. However, temperature changes are slow processes taking hours and the pressure variations in the well will not cause excessive changes in the mud density. (Syltoy et al., 2008)

In drilling operations, it is possible to measure the process variable. A simulated drilling operation will calculate the measured parameters and the simulated BHP will represent the process measurements.

Simulators can function as a redundancy to the operation, if the simulator is calibrated correctly it can take over for the physical measurements until the failing equipment is fixed. This is describe more closely in (Syltoy et al., 2008) where a flow model is used in an MPD operation.

### 2.4 PID control theory

This part is written with reference to (Haugen, 2012, pp. 280-289) . The theory is mostly taken directly from there. With attempt to illustrate the theory behind the PID controller function.

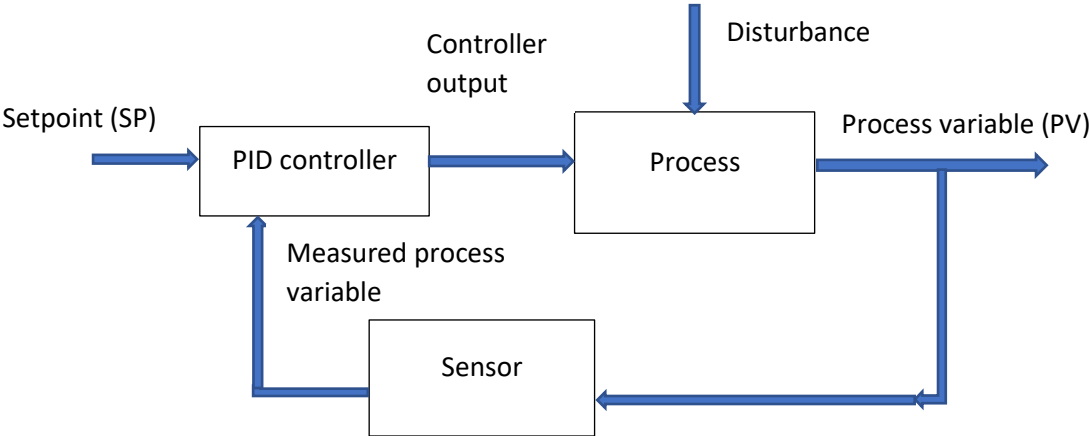


Figure 1: represent block diagram of the PID controller and feedback control system. reworked from (Haugen, 2012, p. 279).



The figure shows a block diagram of how the signal flow will eventually reach the PID controller for calculation of the output value. The calculation takes place in the controller and an output value is sent to the process where the automatic control is taking place. Within the process a sensor is placed to pick up the measured value before it is sent back to the controller where the controller calculates the error signal ( $e=SP-MV$ ). If the control signal has changes somehow by a disturbance for instance, the controller makes the essential calculations to keep the error signal as close to zero as possible.

The PID controller is the most abundant when it comes to automatic control engineering. The letters represent the different mathematical functions needed to maintain adequate regulation of a process. It is made up of 3 functions which work together to get the desired output value. The general equation is time continuous and takes the form (Haugen, 2012, p. 280):

$$\begin{aligned}u(t) &= u_{\text{man}} + K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau + K_p T_d \frac{de(t)}{dt} \\ &= u_{\text{man}} + u_p(t) + u_i(t) + u_d(t) \\ &= M + P + I + D\end{aligned}$$

$u(t)$ : output value

$u_{\text{man}}$  manual set gain

$e$ : error signal (set point – measured value)

$K_p$ : proportional gain

$T_i$ : integral time

$T_d$ : derivative time

M term:  $u_{\text{man}}$ , represent the value the operator set when the controller is in manual mode.  $u_{\text{man}}$  cannot be set when the controller is in automatic mode. Normally a reasonable value is preferred to maintain a “bumpless transfer” between man/auto and auto/man.

An example of a practical “bumpless transfer” is based on own experience from a flow control loop at the automation laboratory at Stavanger offshore technical school. The students/field operators open a bypass valve, so the process value is 50% flow through the bypass line. Then the operator/ students set the controller in auto and set the setpoint value to 50% flow. Now he/she will carefully close the bypass valve and make sure the automatic valve is taking over. It is important to keep the process value as close to the set point as

possible. When the bypass valve is closed, and the flow passes through the automatic valve and main process line, the controller loop is in auto and all other changes done to the system is done by the controller. “Bumpless transfer” is done when starting and shutting down a process. The reason why is it done is to make sure equipment is not damaged and make sure the process stream does not alter much when reaching other processes downstream.

P term represents the proportional gain denoted  $K_p$ . The P term calculates the gain with the error ( $K_p e$ ). it is proportional with the error signal ( $e$ ). according to (Haugen, 2012, p. 288), the P controller is not sufficient to give zero error during static condition. The P term gives a fast control signal change when the error signal changes. The P term will therefore help making the controller fast.

I term represent the integral time,  $\frac{K_p}{T_i} \int_0^t e(\tau) d\tau$  makes sure that the control signal changes until the error signal is as close to 0 as possible. (Haugen, 2012, p. 281). Because of this the I term is the most important term in the controller. It makes sure that the error signal approaches 0 in static conditions.  $T_i$  is also called reset time related to how fast the I term changes its value (Haugen, 2012, p. 285).

The integral gain can also be expressed:  $K_i = \frac{K_p}{T_i}$ , which is what is used in the matlab code in the appendix.

D term represent derivative function of the PID controller  $K_p T_d \frac{de(t)}{dt}$ . The D term only work on error signal changes. It gives a positive or negative contribution to the gain to keep the error signal close to the set point. This can help in avoiding an overshoot. Because of this the regulation can be more controlled or stable when the D term is involved. However, there will be noise present in the measurements. The D term will tend to amplify the noise. Hence in many practical applications it is neglected. (Haugen, 2012, p. 289). Due to this almost all PID controllers run as PI controller, and the derivative part is set as  $T_d=0$ .

### 2.5 PID controller on discrete form

The equation above is on a continuous form, and computer-based controllers makes the calculations on discrete form. Therefore, the continuous form needs to be converted to discrete form for the computer program (i.e. MATLAB) to perform the calculations.

(Haugen, 2012, p. 286):

$$u(t_k) = u_{\text{man}} + u_p(t_k) + u_i(t_k) + u_d(t_k)$$

$u(t_k)$  = output value, controller gain at present time

$u_{\text{man}}$  = constant, set by operator in manual mode

$$u_p(t_k) = K_p e(t_k)$$

$$u_i(t_k) = u_i(t_{k-1}) + \frac{K_p T_s}{T_i} e(t_k)$$

$$u_d(t_k) = K_p T_d \frac{e(t_k) - e(t_{k-1})}{T_s}$$

by discretise each term the Total equation becomes(Haugen, 2012, p. 286):

$$u(t_k) = u_{\text{man}} + K_p e(t_k) + u_i(t_{k-1}) + \frac{K_p T_s}{T_i} e(t_k) + K_p T_d \frac{e(t_k) - e(t_{k-1})}{T_s}$$

Time interval  $T_s$  (seconds) between the samples and controller gain calculations are called sampling interval or time step. A typical value for  $T_s$  in an industrial controller is 0,1s.  $k$  is a counting variable which indicates the time level.  $k$  represents the present time, and  $k-1$  represent the previous time level, and these are separated by one-time step  $T_s$ .

### 2.5.1 Discretization of each term

The result should be an equation or algorithm for the controller gain at present time  $u(t_k)$

M term: constant

P term: becomes  $u_p(t_k) = K_p e(t_k)$

I term: from mathematics integration is defined as summation of rectangles under a curve.

The time discrete I term is a numerical calculation of this integral. In PID controllers the Euler's backward method is used, which means rectangular approximation backward in time.

The integral I approximately equal to the sum of the time step  $T_s$  multiplied with each backward rectangle.

$$\begin{aligned} \int_0^{t_k} e \, d\tau &\approx T_s e(t_1) + T_s e(t_2) + \dots + T_s e(t_{k-1}) + T_s e(t_k) \\ &= T_s [e(t_1) + e(t_2) + \dots + e(t_{k-1}) + e(t_k)] \end{aligned}$$

The factor  $\frac{K_p}{T_i}$  is multiplied in to get the approximation to  $u_i(t)$  at time  $t_k$ :

$$u_i(t_k) = \frac{K_p}{T_i} \int_0^{t_k} e \, d\tau \approx \frac{K_p T_s}{T_i} [e(t_1) + e(t_2) + \dots + e(t_{k-1}) + e(t_k)]$$

The I term is defined as proportional with the sum of all the error signal, or proportional with accumulated error signal from  $t=0$  which is the point where the controller started to present time.

If the controller has a lot of stored error signals, a more practical implementation of the I term can be found by splitting up the summation term:

$$\begin{aligned} u_i(t_k) &= \frac{K_p T_s}{T_i} [e(t_1) + e(t_2) + \dots + e(t_{k-1}) + e(t_k)] \\ &= \frac{K_p T_s}{T_i} [e(t_1) + e(t_2) + \dots + e(t_{k-1})] + \frac{K_p T_s}{T_i} e(t_k) \quad (u_i(t_{k-1}) = \frac{K_p T_s}{T_i} [e(t_1) + e(t_2) + \dots + e(t_{k-1})]) \\ &= u_i(t_{k-1}) + \frac{K_p T_s}{T_i} e(t_k) \end{aligned}$$

Seen from the equation above one only need the present error signal  $e(t_k)$  and the previous time step value of the I term  $u_i(t_{k-1})$  to calculate  $u_i(t_k)$ .

The integral time  $T_i$  is called “reset time”, this is since  $T_i$  is an expression of how fast the I term change its value and therefore the total gain value in its work on achieving zero error signal. The I term is continuously “resetting” (changing) the control signal (Haugen, 2012, p. 288)

D term: in the continuous controller is  $u_d(t) = K_p T_d \frac{de(t)}{dt}$ . It is normal to approximate the derivative of  $e$  with a backward difference i.e as difference between present error and the previous time step, divided by the time step length:

$u_d(t_k) = K_p T_d \frac{e(t_k) - e(t_{k-1})}{T_s}$ . The D term gives a control signal contribution which is proportional with error signals rate of change. Rate of change is  $\frac{e(t_k) - e(t_{k-1})}{T_s}$ . The reason for removing the D term or put the D term value to zero in the PID controller calculations is to avoid abrupt control signal changes due to noise in the process.

### 3. Integration of PI regulator

The PI regulator implemented in the matlab code is derived from continuous form to discrete form from (Sui, 2019):

$$u(t) = u_{man} + K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau$$

Taking the derivative on both sides and using Euler's method the formula takes the form:

$$u(t) = u(t - 1) + k_p(e(t) - e(t - 1)) + k_i e(t) * dt$$

Here,  $\frac{k_p}{T_i} = k_i$ ,  $dt=ts$  (how often the regulator is updated)

The discrete form in the equation above is how the PI regulator is implemented.  $u_{man}$  term is not present due to the constant term disappear when taking the derivative on both sides.

Another advantage by applying this method is no need for storage of summation while simulating.

Regulation is performed on the choke differential pressure directly. Simulated bottom hole values are the measured value. Set point is set to the desired constant bottom hole pressure. No noise is implemented on the simulated values.

#### 4. Ziegler Nichols' closed loop tuning method

According to (Haugen, 2012, p. 320), Ziegler and Nichol's found a way to determine the critical gain (ultimate gain) for the P, I and D controller in an experimental way. by first making the PID controller a clean P controller by turning the I value to a large number, and D value to 0. Then they made sure the process they were controlling was stable with only the P controller being operative. As soon as the process was stable, they put a stress on the system until sustained oscillation occur and called this value for the critical gain ( $K_{p_{crit}}$ ) also denoted  $K_{pu}$ , ultimate gain. Critical period can be found, and the PI parameters is calculated according to figure 2 below.

A description of the closed loop method follows based on (Haugen, 2012, p. 320) follows:

1. Put the controller in manual, the process is as close to the nominal value as possible.
2. Put the process to the specific working point. This is done by altering the gain manually until the process variable is close to the set point.
3. Make sure the PID controller is a clean P controller.  $K_p = 0$ ,  $T_i =$  large number,  $T_d = 0$ .
4. Put the controller in automatic.

5. Increase  $K_p$  until oscillations occur after a leap in the set point. The controller system will now be in its stability limit.
6. Note the  $K_p$  that gives the oscillations. This is the critical gain or ultimate gain  $K_{pu}$ . Also write down the period  $P_u$  for the oscillation. This is the critical period.
7. Decide which controller you wish to tune. (P, PI, PID) for this thesis a PI controller will be used for the simulation. Calculate the controller parameters according to the table below.

	$K_p$	$T_i$	$T_d$
P-regulator	$0,5K_{pk}$	$\infty$	0
PI-regulator	$0,45K_{pk}$	$\frac{P_u}{1,2}$	0
PID-regulator	$0,6K_{pk}$	$\frac{P_u}{2}$	$\frac{P_u}{8} = \frac{T_d}{4}$

Figure 2: show Ziegler Nichols' formulas for control parameters, closed loop method (Haugen, 2012, p. 321)  $K_{pk}$  =critical gain.  $P_u$  = critical period.

The controller used for the choke differential pressure in the matlab code is tuned according to a PI controller. From the figure 2 above the  $K_p$  value is found by multiplying the critical gain with 0,45. The  $T_i$  value is found by dividing the critical period by 1,2.

#### 4.1 Examples of how the Ziegler-Nichols' method was used

The following description of the Ziegler Nichols's method was followed when tuning the PI parameters in the connection case when trying to optimize the parameters  $k_p = 1,00$ ,  $T_i = 10,00s$ . With viscosity 0,05 Pa S:

1.  $T_i$  was set to 10,00s
2.  $K_p$  was set to 1,00 at first. (gives  $K_i=0,1$ )
3.  $K_p$  was increased in steps by 0,02 until sustained oscillations was present
4.  $K_{p_{crit}}$  found and calculated according to Ziegler-Nichols formulas table.
5.  $T_p$  critical period found from the plot and calculated according the Ziegler-Nichols formula table.

The tuning was not satisfactory, so further manipulation with trial and error of  $T_i$  and  $k_p$  was performed:

1. first  $k_p$  was halved and simulation was run. Not satisfactory tuning
2.  $T_i$  was halved and simulation run. Not satisfactory tuning
3.  $T_i$  was halved one more time and simulation was run. Not satisfactory tuning.
4.  $K_p$  was increased by 0,1 increments. Not satisfactory tuning.
5.  $K_p$  increased by 0,1 increments. Not satisfactory tuning.
6.  $T_i$  was halved. Satisfactory tuning.

This gave  $k_p = 0,434$  and  $t_i = 3,646s$ . Seen in the figures 14 and 15 in simulation of connection.

Another example of how the Ziegler Nichols' method was used:

The following description of the Ziegler Nichols's method was followed when tuning the PI parameters in the connection case. Chang of viscosity from 0,05 Pa S to 0,001 Pa S,  $k_p = 1,00$ ,  $T_i = 10,00s$ .

7.  $T_i$  was set to 10,00s
8.  $K_p$  was set to 1,00 at first. (gives  $K_i=0,1$ )
9.  $K_p$  was decreased in steps by 0,02 until sustained oscillations was present
10.  $K_{p_{crit}}$  found and calculated according to Ziegler-Nichols formulas table.
11.  $T_p$  critical period found from the plot and calculated according the Ziegler-Nichols formula table.

This gave  $k_p = 0,369$  and  $t_i = 16,67s$ . seen in figure 17 in simulation of connection

## 5. Introduction to Managed pressure drilling

As stated in (Breyholtz, Nygaard, & Nikolaou, 2010) the SPE/IADC official definition of MPD is:

*An adaptive drilling process used to more precisely control the annular pressure profile throughout the well bore. The objectives are to ascertain the downhole pressure environment limits and to manage the annular hydraulic pressure profile accordingly.*

According to the NORSOK D10, managed pressure drilling is systems manipulating annulus pressure from the surface to control the bottom hole pressure. One of the systems includes equipment such as choke valve, choke manifold, backpressure pump(s), separator (mud gas

separator) and rotating control device (RCD). This equipment is set in a closed loop where the backpressure pumps starts when the main pumps are shut down during a connection to maintain the pressure at the bottom of the well. This is controlled by the choke valve, flow, and pressure measurement where the choke valve will open or close to keep the BHP constant. Closed loop system means that the well is not open to the atmosphere, but are sealed off using a rotating controller device (RCD)(Abimbola, Khan, Khakzad, & Butt, 2015).

Automation of drilling procedures have been more and more used due to the technology development which has made it possible to drill and reach reservoirs that back in the days would be to “risky” to drill fore (Saeed et al., 2012). The method is managed pressure drilling (MPD) where it is possible to reach the “deserted” reservoirs or reservoirs with a narrow drilling window (pressure difference between fracture pressure and pore pressure is small) because of the back-pressure system which upheld the pressure in the well.



5.1 MPD barriers

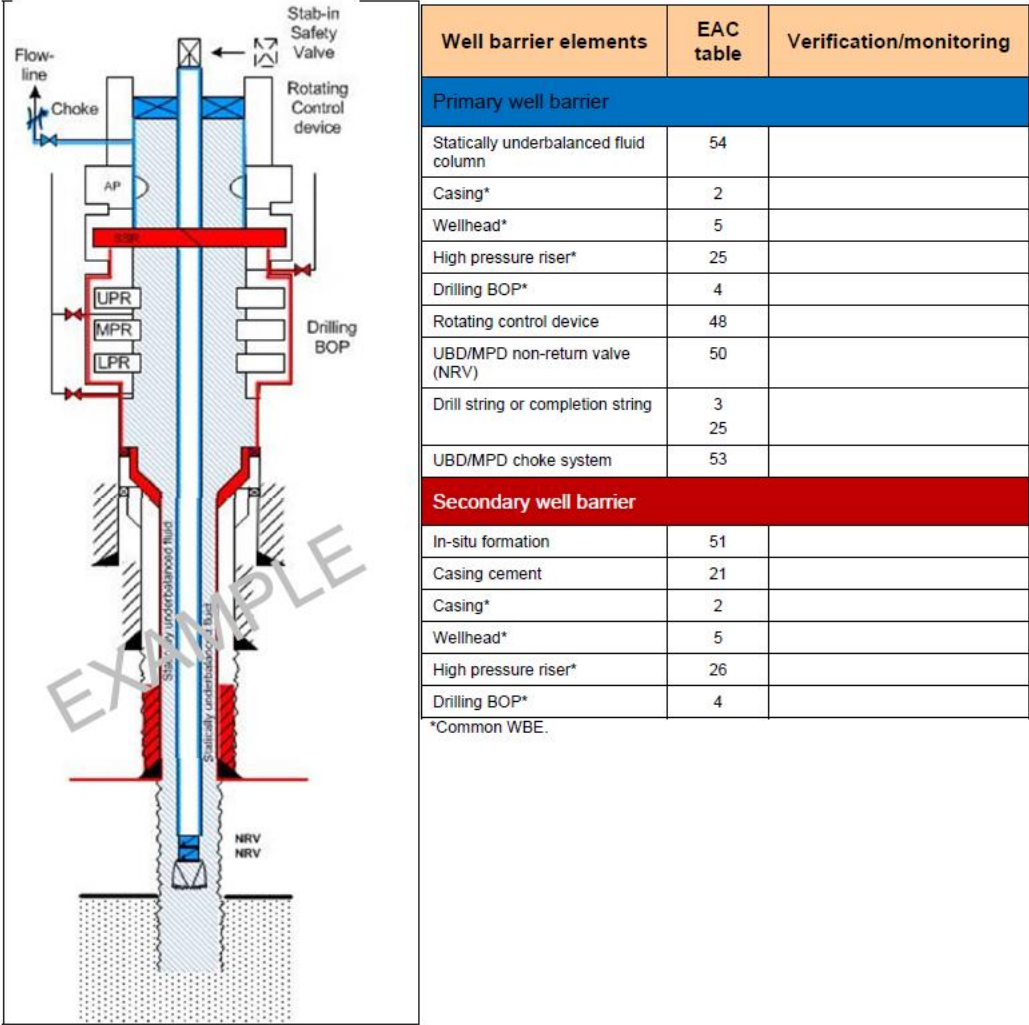


Figure 3: show drilling and tripping of string in UB fluid according to NORSOK D10. Primary and secondary barriers, blue colour: primary, red colour: secondary.

From figure 6 that illustrates the primary and secondary well barriers, the primary barriers include the MPD choke system. The choke system has to be automated and include the equipment and control system for the surface circulation system to be able to provide backpressure.

Some of the most abundant MPD techniques will be briefly described further on.

## 5.2 MPD techniques

As stated in (Abimbola et al., 2015) with reference to (Haghshenas, Paknejad, Rehm, & Schubert, 2008) different techniques are developed and the most used is: constant bottom hole pressure (CBHP), pressurized mud cap drilling (PMCD) and dual gradient drilling (DGD).

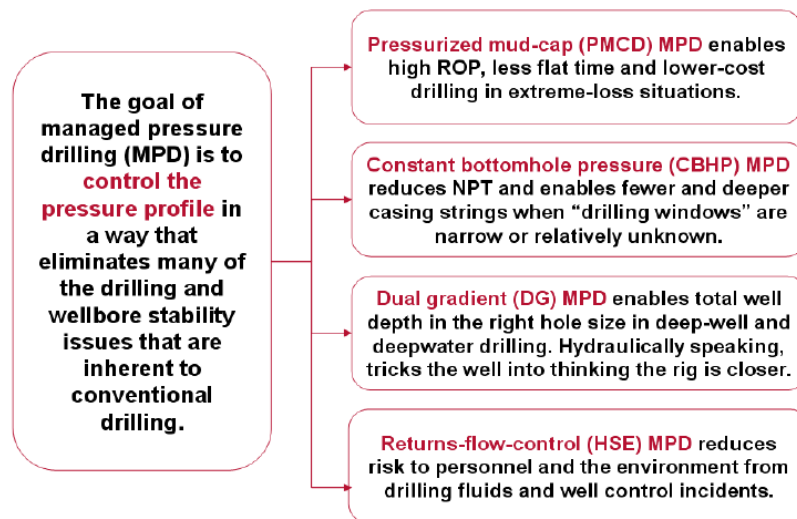


Figure 4: illustrates the goal for MPD and overview of different techniques. (Gedge, Dalgit Singh, Bandico Refugio, Ta Quoc, & Bot, 2013)

### 5.2.1 Pressurized mud cap drilling (PMCD)

(Abimbola et al., 2015) with reference from (Moore, 2008) states that PMCD is used where fractured formations are present and where there is a huge potential for massive mud losses. It is also stated that it is an improved technique from mud cap drilling or floating mud cap drilling, which is an open system drilling. In an open system, heavy mud is placed in the annulus above the fractured formation to balance the fracture pressure. The IADC defines PMCD as: “drilling with no returns to surface, where an annulus fluid column, assisted by surface pressure, is maintained above a formation that is capable of accepting fluid and cuttings”. (Benny, Hidayat, Karnugroho, Sosa, & Toralde, 2013). The annulus above the fracture is filled with a lightweight mud and the annulus is sealed with an RCD on top such

that there will be a positive backpressure on top of the well. Monitoring of this pressure can give an indication of a kick migrating upwards in the annulus. When this happens, the kick will be bullheaded back into the fractures. Often a “cheaper” mud or fluid is used and injected in the fracture together with drilled cuttings. The reason for this is if there is an influx of formation fluid, the influx is bull-headed back into the formation, so the mud expenses is not as costly and a safely drilling method is upheld. (Abimbola et al., 2015) cited from (Moore, 2008) and (Benny et al., 2013).

### 5.2.2 Dual gradient drilling (DGD)

According to (Abimbola et al., 2015) with reference to (Cohen, Stave, Schubert, & Elieff, 2008), dual gradient drilling is a method where two different drilling fluids with different densities are used to drill a well. The lighter fluid is on top and above the seafloor, and the heavier fluid is below the mud line. This will make the drilling window wider and increased production is achieved, which leads to more economical well. One technique is “pump and dump” where the return is dumped at the sea floor. Another technique is the riser less method where the return flow is going through a smaller diameter return line with help from mud lift pumps. Because of the lower dimension of the return line, the DGD method uses less space at the rig floor, compared to the risers in conventional drilling. Therefore, smaller rigs can be used which is beneficial in deep-water applications. (Abimbola et al., 2015) cited from (Cohen et al., 2008)

There is a drilling system that has some resemblance with the dual gradient system and that is the controlled mud cap drilling system described in (Bysveen, Fossli, Stenshorne, Skärgård, & Hollman, 2017). This is a drilling system that operates with a reduced mud level in the riser and this creates a dual gradient effect. In addition, this system is also designed for handling mud losses in fractured carbonates.

### 5.2.3 Continuous circulation system (CCS)

Continuous circulation system (CCS) are sometimes characterized as CBHP method. CCS is used during a connection, where it makes sure the equivalent circulation density (ECD) remain fixed so the BHP does not change. (Abimbola et al., 2015) cited from (Vogel & Brugman, 2008). The CCS makes it possible to circulate through the drill string during connections. An example of application is presented in (Syltoy et al., 2008) where it is used in combination with the CBHP method.

## 6. Constant bottom hole drilling (CBHP)

Constant bottom hole pressure is a technique where the pressure is held constant even though the main pumps are turned off and kept inside the drilling window. The pressure limits for the window are pore pressure (lower) and fracture pressure (upper) (Abimbola et al., 2015) cited from (Fredericks, 2008). This is done with help from the RCD, choke, and backpressure pump to provide the closed circulation system. During conventional drilling the static and dynamic BHP is defined as (Abimbola et al., 2015):

$$\text{BHP}_{\text{static}} = P_h \quad (P_h = \text{hydrostatic pressure of the mud column})$$

$$\text{BHP}_{\text{dynamic}} = P_h + P_{\text{ann}} \quad (P_{\text{ann}} = \text{annulus friction pressure})$$

Due to the closed circulation system the bottom hole pressure remains constant and the static and dynamic BHP takes the form (Abimbola et al., 2015):

$$\text{BHP}_{\text{static}} = P_h + P_{\text{bp}} \quad (P_{\text{bp}} = \text{backpressure})$$

$$\text{BHP}_{\text{dynamic}} = P_h + P_{\text{ann}} + P_{\text{bp}}$$

Here the  $P_{\text{bp}}$  is the backpressure created by the flow through the choke. When the main pumps are off due to a connection the BHP is up held with the backpressure, and during circulation the BHP will be calculated with the dynamic BHP formula (Abimbola et al., 2015). It is the flow from the backpressure pump that makes it possible to exert a backpressure on the well when the main pumps are off during connections.

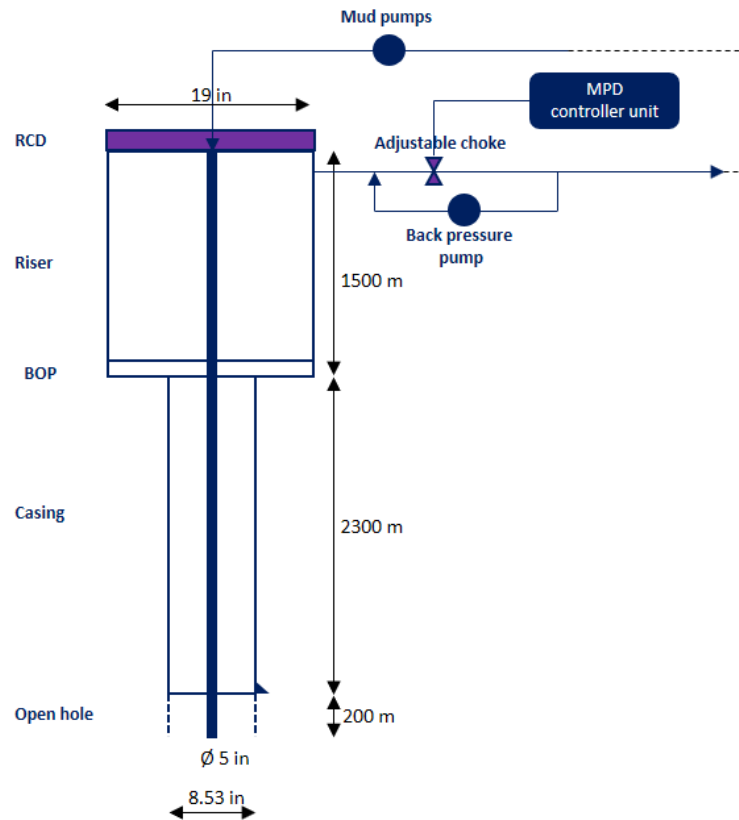


Figure 5: show the set up for CBHP-MPD system (Gomes, Frøyen, Fjelde, & Bjørkevoll, 2018)

Figure 4 illustrates the set up for CBHP-MPD from (D Gomes et al., 2018) with the RCD, choke, and back pressure pump present which is a part of the closed loop system when adjusting the backpressure to keep the BHP constant.

### 6.1 How to control pressure in the CBHP- MPD system

According to NORSOK D10 (p. 138):

*A dedicated MPD choke manifold shall be used to control the wellbore pressure and reduce the pressure at surface to acceptable levels before entering the separation equipment or the shakers. A manual MPD choke system is not accepted as a part of the primary well barrier.*

To meet the requirements, (Godhavn & Knudsen, 2010) describes that the choke is automatically controlled to acquire the back pressure needed. For the choke to be automatically controlled a control system is used. The control system consists of a hydraulic model that estimates the parameters received from the measurement while drilling (MWD) tool needed for the back pressure to keep the BHP constant. (Godhavn & Knudsen, 2010).

The hydraulic model based on the well geometry, calculates the parameters in real time from the MWD (pressures, temperature, flow rate) based on inputs ( mud rate, mud rheology, mud density, temperature) (Godhavn & Knudsen, 2010). Basically, to compensate for the parameters altering the BHP the hydraulic model is needed to be able to calculate the adjustments the choke has to do to keep the BHP constant (Syltoy et al., 2008).

As stated in (Godhavn & Knudsen, 2010) The control system/choke controller needs to be stable and robust. Stability is connected to the ability to remain in a scope of an initial set equilibrium of the system. Robustness is connected to the controller's ability to deal with process noise and measurement noise without getting unstable. (Godhavn & Knudsen, 2010).

### 6.1.1 Choke valve

What control signals the choke valve gets from the controller is one important parameter.

Another essential parameter is how fast the valve opens or closes (Syltoy et al., 2008). Both parameters need to react fast to make the choke valve be able to maintain the desired BHP. A simple equation of flow through a valve is shown below (Haugen, 2012, p. 108):

$$q = k_v * z * \sqrt{\frac{\Delta p}{\rho}}$$

q = flow,  $k_v$  = valve constant, z = valve opening,  $\Delta p$  = choke differential pressure,  $\rho$  = density

Manipulating the formula with regard of the valve opening:

$$z = \sqrt{\frac{q^2 \rho}{k_v^2 \Delta p}}$$

This equation was implanted in the matlab code in the appendix. However, it was decided to regulate on the choke differential pressure. Reworking the equation an expression for the differential pressure is occurred.

### 6.1.2 Automation during connections

Automatic choke system is used during a connection. This is done by turning off the circulation pumps and the choke pressure will keep the BHP constant. An example of a test is described in (Godhavn & Knudsen, 2010). The test was performed at the Gullfax C. the test was closely monitored by the choke operator communicating with the driller before automatic mode. First the controller was tuned to make sure the stability and robustness was in order. The circulation was ramped down from 1600lpm to 0 in 120 seconds. BPP ramped up to 950lpm automatically. The backpressure setpoint was increased from 22 to 37 bar by the hydraulic model. The BHP set point was set to 98 bar. (Godhavn & Knudsen, 2010).

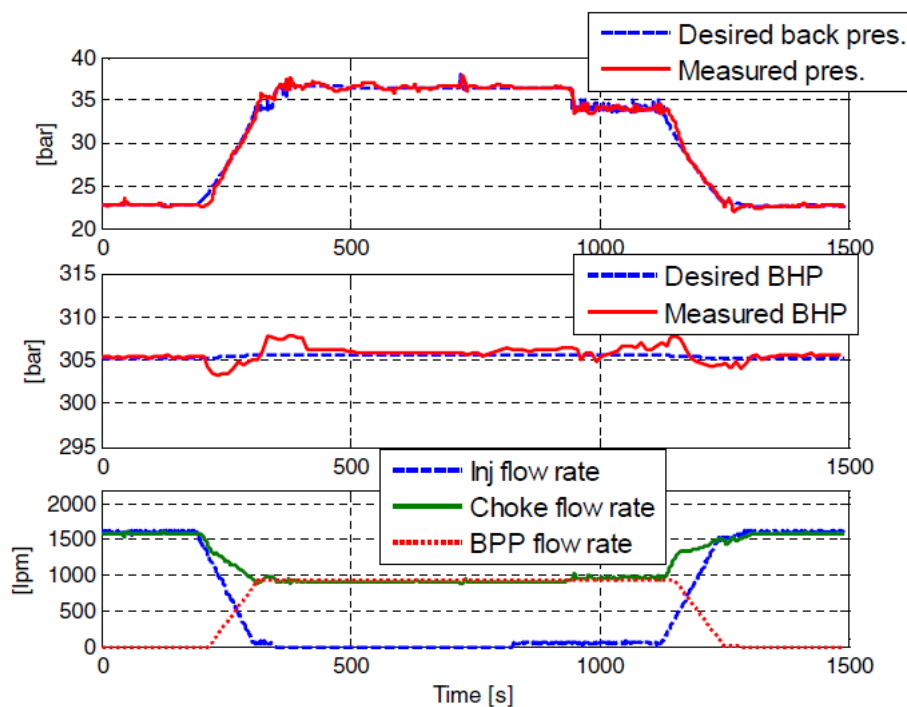


Figure 6: illustrates the results from the connection test.(Godhavn & Knudsen, 2010)

The results from the connection test at Gullfaks C is illustrated in figure 5 from the middle plot where the BHP is plotted with time one can see that the control system manages to keep the BHP constant within a plus minus 5 bar pressure window.

6.1.3 Kick handling for the CBHP-MPD system

Managed pressure drilling works in a narrower drilling window, and the CBHP - MPD system can handle small influxes. It means that small kicks can be circulated directly through the MPD choke and to the mud gas separator. However, for larger kick volumes, the BOP will be activated, and the well will be controlled by conventional well control methods using the rig choke.

However, how large kicks that can be handled directly by the MPD equipment depends on the formation integrity but also surface equipment capabilities. As stated in NORSOK D10, (p.138): *“The surface system shall be selected and dimensioned to handle the anticipated fluid/solids, including formation fluids if potential exists for influx removal with MPD.”*

When circulating a kick with the MPD system, the pressure at the bottom will intentionally be kept fixed at target by adjusting the choke pressure. If a free gas kick is circulated upwards, it will expand and reduce the hydrostatic pressure in the well. This will have to be compensated by increasing the backpressure. A larger kick will lead to a higher backpressure. However, there is a limit for how large the pressure can be at the top of the well before the equipment capabilities are exceeded. Hence, there will be a limitation for how large kicks that can be handled.

In NORSOK D10 (p. 142), the following well control matrix illustrates this. It is a requirement to work out this.

		Surface Pressure Indicator			
		At Planned Drilling Back Pressure	At Planned Connection Back Pressure	> Planned Back Pressure & < Back Pressure Limit	≥ Back Pressure Limit
Influx Indicator (See Chart 1 Below)	No Influx	Continue Drilling	Continue Operation	Continue operation, adjust system to decrease WHP	Secure well, evaluate next planned action
	Operating Limit	Continue drilling, adjust system to increase BHP	Continue operation, adjust system to increase BHP	Continue operation, adjust system to decrease WHP and increase BHP	Secure well, evaluate next planned action
	< Planned Limit	Cease drilling, adjust system to increase BHP	Adjust system to increase BHP	Secure well, evaluate next planned action	Secure well, evaluate next planned action
	≥ Planned Limit	Secure well, evaluate next planned action	Secure well, evaluate next planned action	Secure well, evaluate next planned action	Secure well, evaluate next planned action

Figure 7: illustrates well control matrix for MPD according to NORSOK D10.

*Operating limit: A well specific limit below which drilling can continue.*

*≥Planned limit: A well specific limit where MPD ceases and transitions to well control operations,*



As shown in figure 7 according to NORSOK D10 when an influx exceeds the planned limit the kick must be handled the conventional way.

#### 6.1.4 Influx management envelope (IME)

How to use transient flow models for evaluating how large kicks a CHBP- MPD system can handle was discussed in (D Gomes et al., 2018).

A concept that was explained in (Gabaldon et al., 2017) was the influx management envelope (IME concept). The concept is based on kick tolerance, where volume and intensity of influx is plotted. Regions of when it is safe to circulate a kick through the MPD equipment, or if the BOP has to be activated and the kick is handled in a conventional matter is established (Gabaldon et al., 2017).

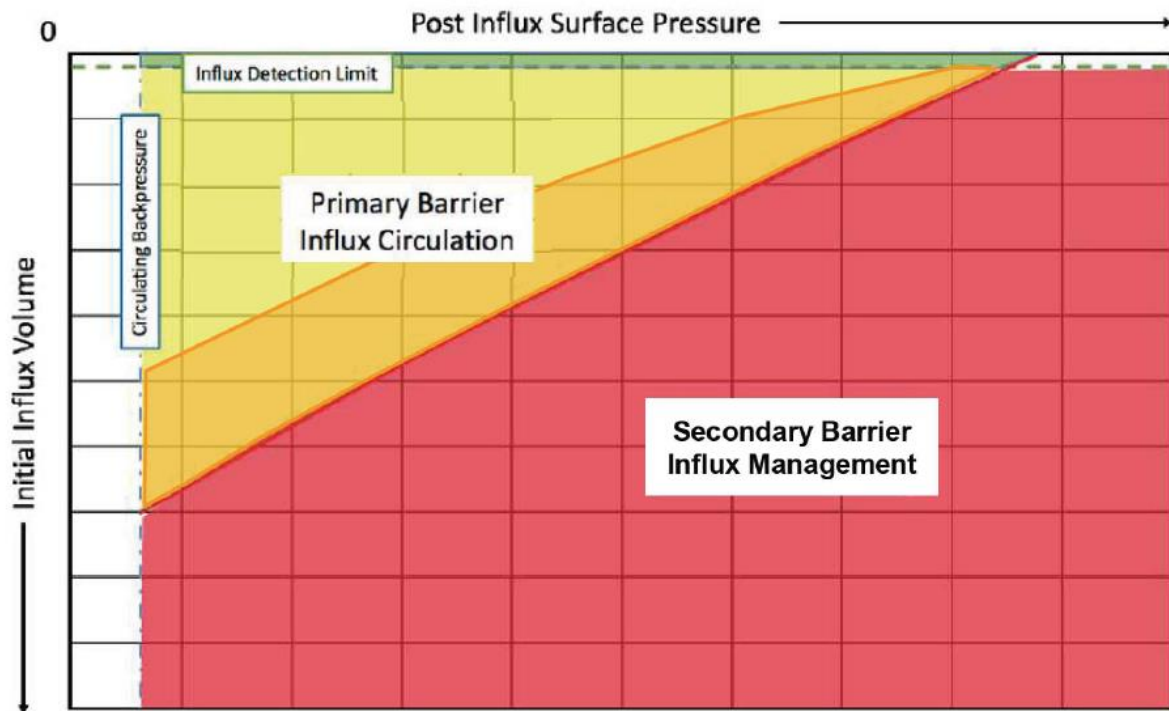


Figure 8: show the influx management envelope concept (Gabaldon et al., 2017).

In figure 8 above, the colours represents the areas where an influx can be removed with primary barrier or if the influx is to large, so the secondary barrier has to be used and the well is shut in (Gabaldon et al., 2017).

Green area show that no influx is noticed and the MPD operation goes as planned.

Yellow area is where an influx is spotted, the influx is small enough for the primary barrier to handle.

Red area indicates that some of the primary barriers will be violated and the well should be secured with secondary barrier.

Orange area was included in the work of (Gabaldon et al., 2017) and it represent where an influx can be removed with primary barrier, but other parameters must be altered in order to not exceed the limits.

The IME is a good tool for how to handle the influxes during a MPD operation (Gabaldon et al., 2017).

Both (D Gomes et al., 2018) and (Gabaldon et al., 2017) rely om using transient flow models to evaluate the maximum flowrates and the maximum surface pressures that the MPD will experience when circulating a certain kick volume through the system. These has then to be compared with the surface equipment limitations (pressure limitation of RCD and riser if used and what rates the mud gas separator can handle)

## 7. Transient two-phase model based on AUSMV scheme

The simulations performed in the thesis is based on transient drift flux model combined with AUSMV (advection upstream splitting model) scheme. Described in (Fjelde & Evje, 2010, p. 2), two-phase models are used to evaluate the behaviour of oil and gas flowing in a pipeline. In (Evje & Fjelde, 2002), the AUSM scheme was combined with flux vector splitting to make it more suitable for two-phase systems.

The matlab code was provided by supervisor and a PI regulator was implemented. A short description of the two-phase model and AUSMV scheme is presented below.

The two-phase model as stated in (Udegbunam, Fjelde, Evje, & Nygaard, 2015, p. 98) involve 3 different equations. It represents 1 dimensional flow along a pipe/well. Two equations represent mass conservation of liquid and gas. The third equation is the mixture momentum equation. The energy equation is neglected since temperatures are considered fixed in time. Assuming no mass exchange between the phases, the equations takes the form (D Gomes et al., 2018, p. 3):

Liquid phase:

$$\frac{\partial}{\partial t} (\alpha_l \rho_l) + \frac{\partial}{\partial z} (\alpha_l \rho_l v_l) = 0$$

Gas phase:

$$\frac{\partial}{\partial t} (\alpha_g \rho_g) + \frac{\partial}{\partial z} (\alpha_g \rho_g v_g) = 0$$

Mixture momentum:

$$\frac{\partial}{\partial t} (\alpha_l \rho_l v_l + \alpha_g \rho_g v_g) + \frac{\partial}{\partial z} (\alpha_l \rho_l v_l^2 + \alpha_g \rho_g v_g^2 + p) = -(F_w + \rho_{\text{mix}} g \cos \theta)$$

Where:

$\alpha_l$  and  $\alpha_g$  represent the volume fractions of liquid and gas, the sum equals 1 ( $\alpha_l + \alpha_g = 1$ ).

$\rho_l$  and  $\rho_g$  are the densities of liquid and gas.

$v_l$  and  $v_g$  represent the velocities of liquid and gas.

$p$  is pressure.

$F_w$  is friction force (expressed as a gradient).

$\rho_{\text{mix}}$  is the mixture density ( $\rho_{\text{mix}} = \alpha_l \rho_l + \alpha_g \rho_g$ ).

$g$  is the gravity acceleration constant.

$\theta$  represent the inclination ( 0 degrees means vertical).

Combining  $\rho_{\text{mix}}$ ,  $g$  and  $\theta$  which is the gravity elements,  $F_g$ . the source term takes the form  $q = F_g + F_w$ .

Described in (Udegbum et al., 2015, p. 99) the source term i.e. the right hand side of the mixture momentum equation can be denoted  $-q$ . The conservation equations can be displayed on the following form:

$$\partial_t \begin{bmatrix} \alpha_l \rho_l \\ \alpha_g \rho_g \\ \alpha_l \rho_l v_l + \alpha_g \rho_g v_g \end{bmatrix} + \partial_x \begin{bmatrix} \alpha_l \rho_l v_l \\ \alpha_g \rho_g v_g \\ \alpha_l \rho_l v_l^2 + \alpha_g \rho_g v_g^2 + p \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -q \end{bmatrix}$$

When there is a certain number of unknown variables, closure laws are needed to make the drift flux model complete. Some simple closure laws consist of (D Gomes et al., 2018, p. 3):

Gas slip model:

$$v_g = K v_{mix} + S$$

The  $v_{mix}$  is the mixture of the velocity. ( $v_{mix} = \alpha_l v_l + \alpha_g v_g$ ),  $K$  is a constant and  $S$  represent the liquid velocity relative to the gas velocity- It represents how fast the gas rises in a static well (D Gomes et al., 2018). For the simulations in this thesis, the parameters were set to  $K=1,2$  and  $S= 0,55$  m/s.

Liquid density model:

$$\rho_l = \rho_{l0} + \frac{p-p_0}{a_l^2}$$

$a_l$  represent the speed of sound in liquid and was set to  $a_l = 1500$  m/s in the code provided. The density of liquid at surface conditions was set to  $\rho_{l0} = 1000$  kg/m<sup>3</sup>. The pressure at the surface conditions set to  $p_0 = 1$  bar.

Gas density model:

$$\rho_g = \frac{p}{a_g^2}$$

$a_g^2$  represent the speed of sound in gas set to  $a_g = 316$  m/s.

Friction model (unit Pascal per meter):

$$F_w = \frac{2f \rho_{mix} v_{mix} |v_{mix}|}{(d_{out} - d_{in})}$$

The friction factor  $f$  represents if the flow is laminar or turbulent decided by the Reynolds number. Reynolds number smaller than 2000 indicates laminar flow and Reynolds number larger than 3000 indicates turbulent flow. The friction factor used  $f = \frac{24}{Re}$  for laminar and  $f = 0,052Re^{-0,19}$  for turbulent flow.

Before using a numerical scheme, one must use discretization where the well is divided into several boxes. The number of boxes used will also reflect the accuracy of the calculations. The conservation and closure laws will be calculated for each box to find the unknown conservative variables (Udegbumam et al., 2015, p. 101). The conservative variables are products of different physical variables ( $u_1 = A\rho_l\alpha_l$ ,  $u_2 = A\rho_g\alpha_g$ ,  $u_3 = A\rho_l\alpha_l v_l + A\rho_g\alpha_g v_g$ ) defined

from the conservation laws. After finding the updated conservative variables, the physical variables like pressure, phase densities, phase volume fraction and phase velocities can be found. The closure laws are needed for this.

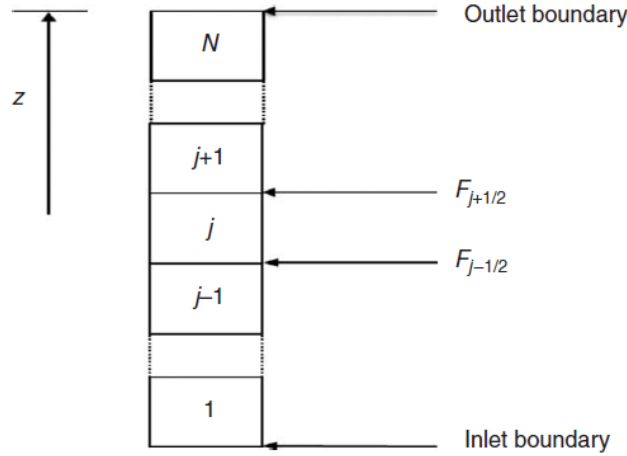


Figure 9: illustrates the discretization (Udegbunam et al., 2015, p. 101).

The figure 9 above, show how the well is discretized into number of boxes. The inlet is at the bottom at box one and the outlet is at the top box. In simulations for the connection case, 25 boxes were used.  $\Delta z$  represent the length of each box.

Variables are calculated for the next time level by using variables from the step before. The general up dating formulas used in (D Gomes et al., 2018, p. 5) below shows this process:

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta z} \left( F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n \right) = \Delta T Q_j^n$$

In this formula it shows how the conservative variables in the cell at the new time level is calculated using information about the conservative variables in the cell at the old time level and the fluxes which are also based on information from the neighbor cells. The timestep length ( $\Delta t$ ) is divided by the length of the cell ( $\Delta z$ ) multiplied with the flux at the boundary to the next cell ( $F_{j+1/2}^n$ ) minus the flux at the boundary toward the previous cell ( $F_{j-1/2}^n$ ). For the momentum equation, also the source term including hydrostatic pressure and friction must be accounted for. In the represent the cell number and n represent the time level (D Gomes et al., 2018, p. 5). The values at the middle of the cell will be calculated.

One can note that the fluxes defined in the general updating formula shown above are using information from the old time level, this makes the scheme explicit in time and the CFL condition has to be satisfied to make the scheme stable. The flux at a boundary will always depend on the information in the two neighbouring cells. How the fluxes are calculated will depend on which numerical scheme that is being used. Here the AUSMV scheme has been used and the formulas for the fluxes in this case can be found e.g. in (Udegbonam et al., 2015).

To make the scheme more accurate and reduce numerical diffusion, the slope limiter concept was used described in (D Gomes et al., 2018). Numerical diffusion is when the numerical scheme tends to smooth out transition zones that physically should be sharp.

The CFL conditions makes a constraint on the timestep length which depends on the CFL number, length of boxes and the propagation speed of waves in the two-phase flow system. The equation is found below (Udegbonam et al., 2015, p. 101) and (D Gomes et al., 2018):

$$\Delta t \leq CFL \frac{\Delta z}{\max(|\lambda_1|, |\lambda_2|, |\lambda_3|)}$$

Here CFL is a number that must be between 0 and 1 and it depends on which explicit scheme that is being used. The lamdas represents the waves that are propagating in the two-phase flow system.

According to (D Gomes et al., 2018, p. 4), Two of the  $\lambda$ 's shown in the equation above are related to the sonic waves propagating upstream and downstream in the system. They will be present when changes to the flow rate occur and when the choke is being altered (D Gomes et al., 2018, p. 4).

Assuming the liquid is incompressible an approximation of the sound velocity in the two-phase flow, a thorough mathematical analysis can be carried out to find an expression for it. (Udegbonam et al., 2015) cited from (Benzoni-Gavage, 1991).

$$\omega = \sqrt{\frac{p}{\alpha_g \rho_l (1 - K \alpha_g)}}$$

This gives  $\lambda_1 = v_1 - \omega$ ,  $\lambda_2 = v_g$  and  $\lambda_3 = v_1 + \omega$ . Showing the pressure pulses traveling upward and downward. A more elaborated description can be found in (Evje & Fjelde, 2002, p. 679) and (Udegbunam et al., 2015, p. 100). It should be noted that the advanced mathematical analysis of this system was performed by (Benzoni-Gavage, 1991) cited in (Udegbunam et al., 2015) and the expressions represented above have their background from this work.

The pressure is one of the unknown physical variables. At the new time level pressure is determined from the equation (Udegbunam et al., 2015):

$$p(\alpha_l \rho_l, \alpha_g \rho_g) = \frac{a_l^2 (-b + \sqrt{b^2 - 4ac})}{2}$$

Where,  $a = \frac{1}{a_l^2}$ ,  $b = k_0 - \alpha_l \rho_l - \frac{a_g^2}{a_l^2} \alpha_g \rho_g$ ,  $c = -k_0 a_g^2 \alpha_g \rho_g$ ,  $k_0 = \rho_{l,0} - \frac{p_0}{a_l^2}$

After the pressure at the new box is calculated, liquid and gas densities, which are unknown physical variables can be calculated from the liquid density model ( $\rho_l = \rho_{l,0} + \frac{p-p_0}{a_l^2}$ ) and gas density model ( $\rho_g = \frac{p}{a_g^2}$ ) (Udegbunam et al., 2015, p. 101). The phase volume fraction can then be found using the updated conservative variables in combination with the updated phase densities.

Further the phase velocities are calculated based on the updated conservative variable from the mixture momentum equation ( $\alpha_l \rho_l v_1 + \alpha_g \rho_g v_g$ ) and by using the gas slip relation. (Udegbunam et al., 2015, p. 101).

The fluxes at the beginning or bottom of the well and on the outlet on top cannot be calculated using the AUSMV scheme. Boundary fluxes calculation can be found in (D Gomes et al., 2018). For the simulations in this thesis an open well configuration will be assumed. The pressure at outlet is known to be set to the choke pressure exerted by the back pressure MPD system.

## 8. Simulation cases

### 8.1 Simulation of connection

The connection case is simulated with the well data: 4000m vertical well, outer diameter ( $d_o$ ) 8,5 inches (0,2159 m), inner diameter ( $d_i$ ) of 5,0 inches (0,127 m), and a flow rate of 40 kg/s ( 2400 lpm). The flow rate is adjusted at the bottom of the well.

The choke pressure ( $p_{choke}$ ) is set to be 10 bar under circulation, and after 30 seconds the controller is activated. The controller regulates directly on the differential pressure of the choke to maintain constant bottom hole pressure. The setpoint is set to 430 bar.

After 200 seconds a 10-minute-long connection is taking place. The main pumps are ramped up and down in 60 seconds. The circulation pumps are shut down and the choke pressure is increased to compensate for the friction loss. (for an MPD situation this is done by starting the backpressure pump(s) which controls a flow through the choke keeping the bottom hole pressure constant. The backpressure flow is only circulating through the managed pressure drilling (MPD) equipment). However, for this case only main pumps are ramped up/down.

#### 8.1.1 No regulation

Connection case without automatic control with viscosity 0,001 Pa S and 0.05 Pa S to see how the bottom hole pressure varies during a connection of 10 minutes.

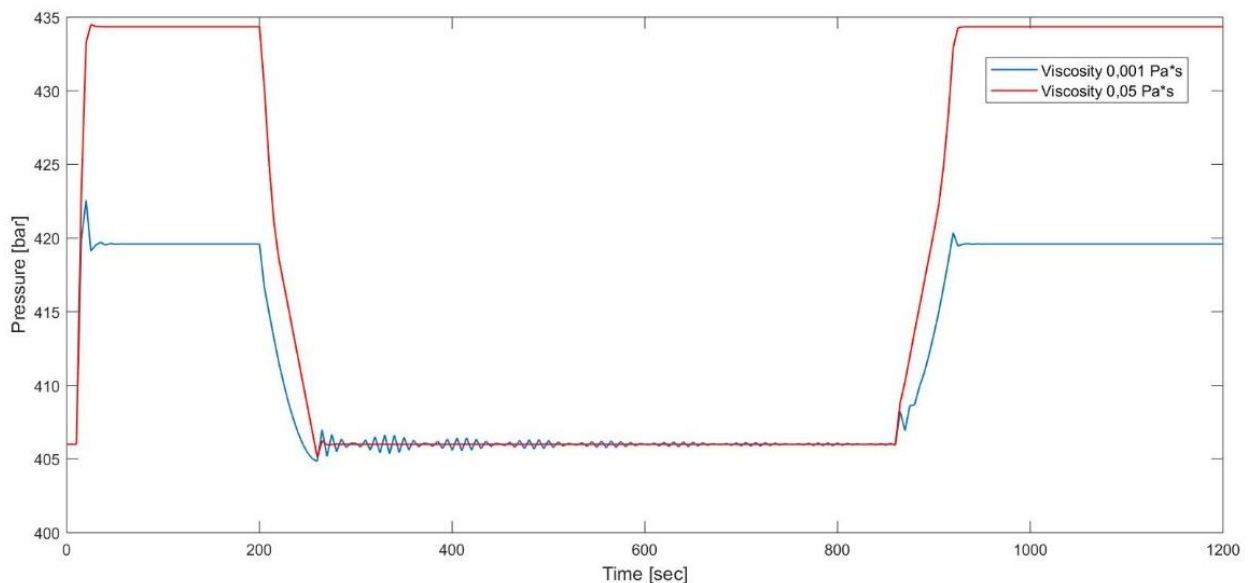


Figure 10: show the variations of the BHP during connection with no automatic control and different viscosities.



Looking at figure 10, that illustrates the BHP during connection and without automatic control. one can see that with viscosity 0,001Pa S around 250 seconds, the BHP decreases to about 405 bars. Some oscillation is present from 250 to 600 seconds ranging from about 405 to 407 bars before the oscillations gets less and stabilize around 406 bars. The simulation with viscosity 0,05 Pa S indicated by the red line show how the BHP decreases from 434 to 406 Bar during connection and no oscillation is present. The drop in pressure corresponds to the loss in friction when the pump is turned off. The frictional pressure loss is larger for a fluid with large viscosity. Note that a friction model for Newtonian fluid is used here.

The oscillations seen, can be explained by pressure pulses that are created when the pump is turned off and is an effect caused by acceleration. The oscillations are more pronounced just after the pump is stopped and then they seem to be dampened. They are mainly present for the fluid with low viscosity. The way the pulses are dampened is depending on the viscosity of the fluid.

This also illustrates how the viscosity can potentially alter the regulation since the oscillations may have impact on that. The low viscosity corresponds to water but in a real drilling situation, a Non-Newtonian drilling fluid will be used which is expected to cause more frictional resistance. Hence, it is chosen to set the viscosity to 0.05 Pa S in the rest of the simulation to reduce the effect of pressure pulses. This is the only way to dampen the effect of the pressure pulses. A future study could involve testing Non-Newtonian drilling fluids to see how the pulses dampen in this case.

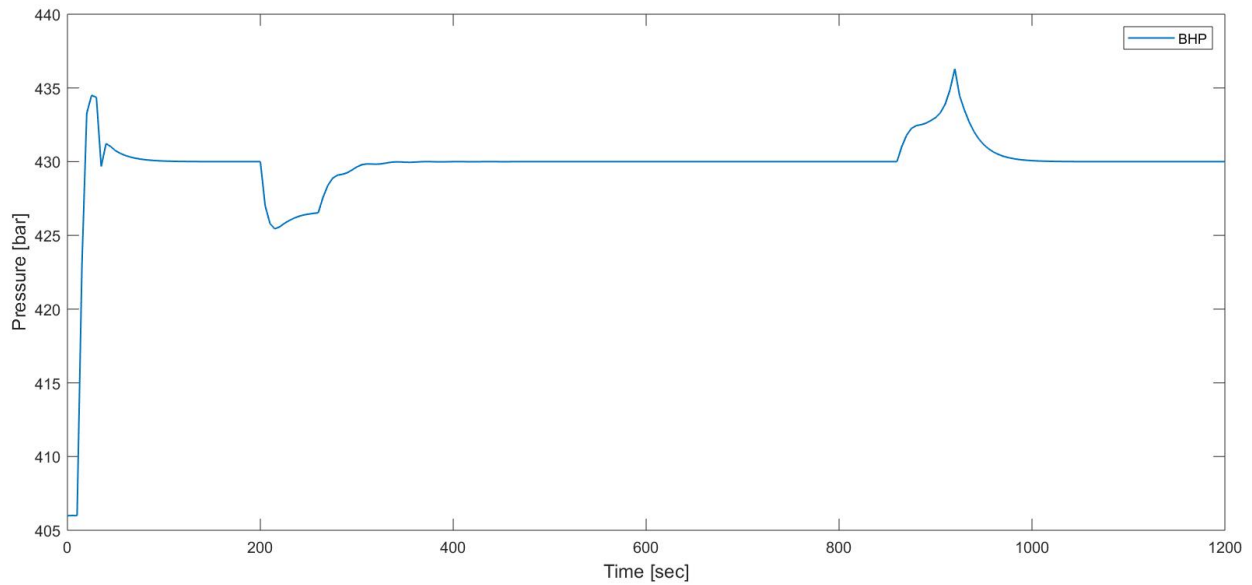
### 8.1.2 Simulation with PI regulator implemented and different viscosities.

In this chapter the intent was to tune the regulator (alter  $k_p$  and  $t_i$ ) and change the viscosity between 0,001 Pa S and 0,05 Pa S to see how the pressure pulses affected the regulator parameters. The main pumps are shut off and started during a 60 second period.

#### 8.1.2.1 Viscosity 0,05 Pa S:

A simulation was performed, and viscosity 0,05 Pa S was used. The sampling time equal to the timestep of the scheme ( $t_s=dt(0,02s)$ ), and control parameters:  $k_p = 1,0$ ,  $t_i = 10,0s$  gives  $k_i = 0,1$ .

## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model



*Figure 11: show BHP with automatic control and viscosity 0,05Pa S.*

Figure 11 show the bottom hole pressure during connection and one can see that the regulator manages to hold the set pressure at 430 bar. At 200 seconds when the main pumps are shut down the pressure decreases to 425,4 bar and increases again as the back-pressure pumps starts and the choke differential pressure increases to hold the pressure constant at 430 bar. 430 bars are reached at 335 seconds.

When the 10 minutes connection is completed after 800 seconds, the main pumps starts, and the backpressure pump is stopped. Its seen when the BHP increases to 436,3 bar due to the circulation of the main pumps starts. The BHP increases before it decreases and stabilizes at 430 bar around 1000 seconds.

PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

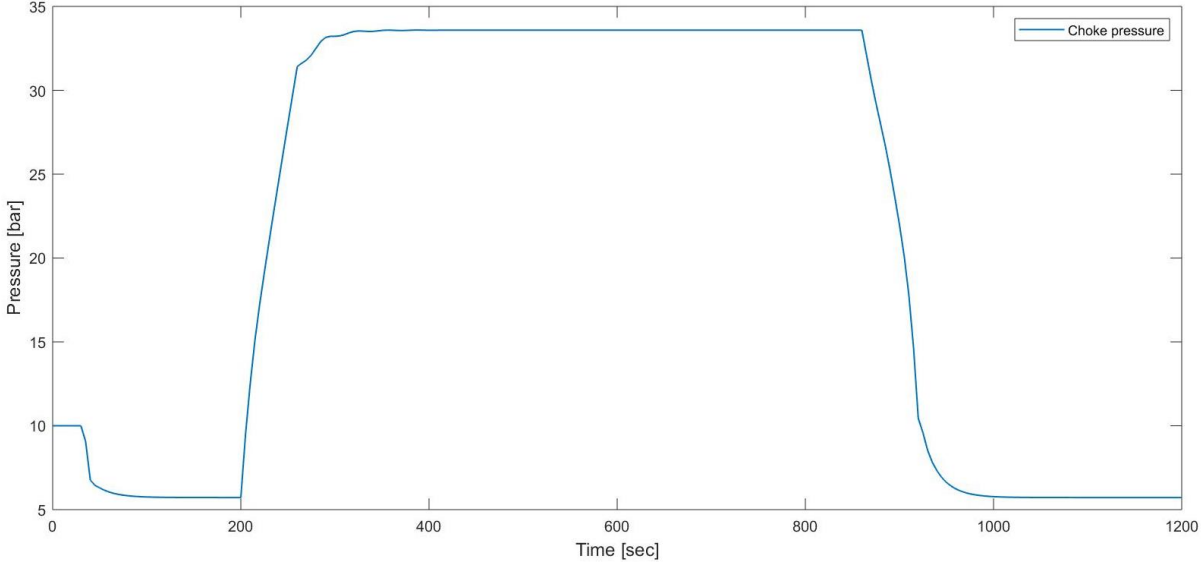


Figure 12: show choke pressure vs time with  $k_p=1,0$  and  $t_i=10s$  viscosity  $0,05\text{ pa S}$ .

From figure 12 above one can see the choke pressure increases when the pump is turned off at 200 seconds. The choke pressure remains constant during the 10 minutes connection and decreases when the pump is started.

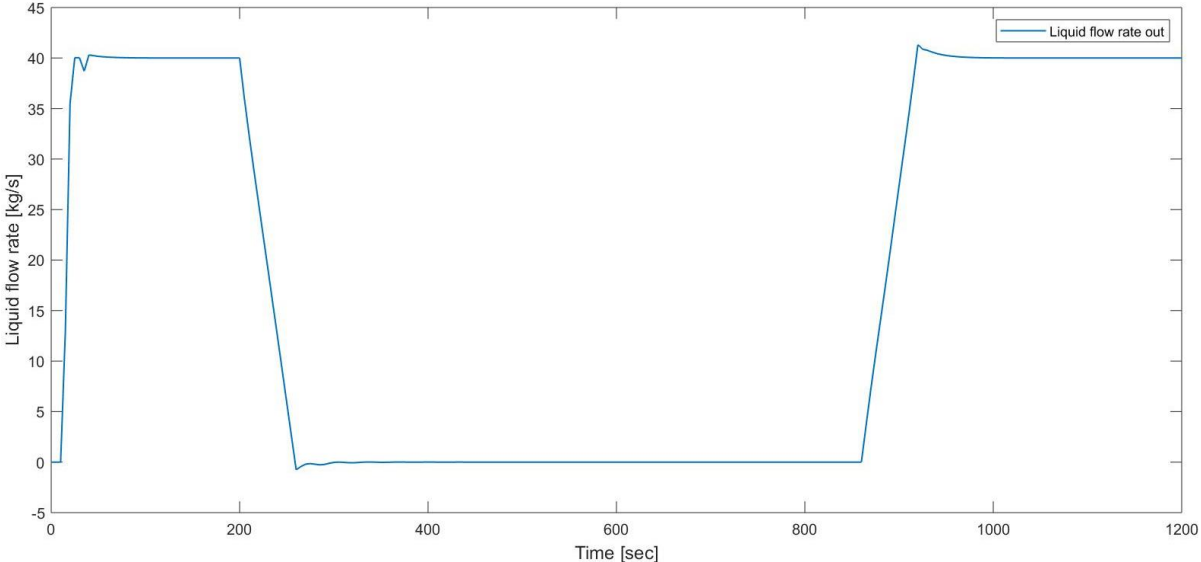
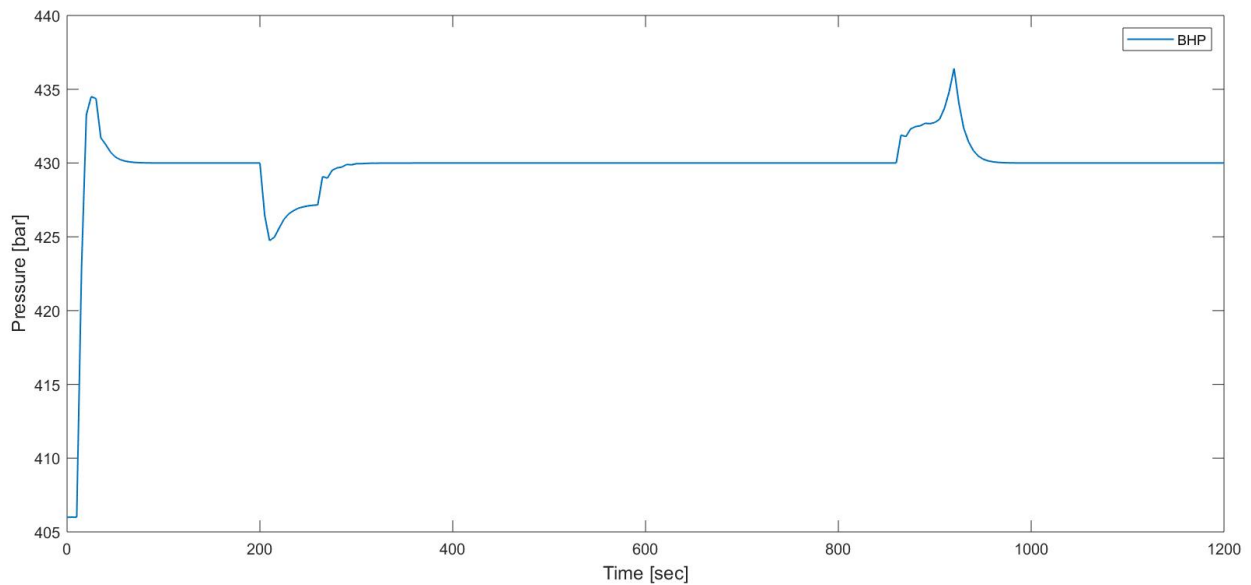


Figure 13: show liquid flowrate vs time during connection.

Illustrated in figure 13 above the liquid flow rate decreases toward 0 when the pump is turned off. No flow during the connection and increases to 40 kg/s as the pumps starts.

An attempt to alter the regulator parameters was attempted to see if it was possible to make the regulation reach closer to 430 bars when the main pumps are stopped and starts.

New  $k_p$ ,  $t_i$  and  $k_i$  was found by the Ziegler Nichols closed loop method and further manipulated with trial and error to get a satisfied regulation was tried. The new regulation parameters found was:  $k_p = 0,434$ ,  $t_i = 3,646$  gives  $k_i = 0,1190$ .



*Figure 14: show the BHP with viscosity 0,05 Pa S and new  $k_p$ ,  $t_i$  and  $k_i$ .*

The bottom hole pressure with the new regulation parameters is illustrated by figure 14. The difference between using the new regulation parameters  $k_p = 0,434$ ,  $t_i = 3,646$  this gives  $k_i = 0,1190$  with viscosity 0,05 Pa S is small, but when the pumps are shut of the bottom hole pressure decreases to 424,7 bar. When the pumps are turned back on the BHP increases to 436,4 bar. 430 bar is reached after 300 seconds.

PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

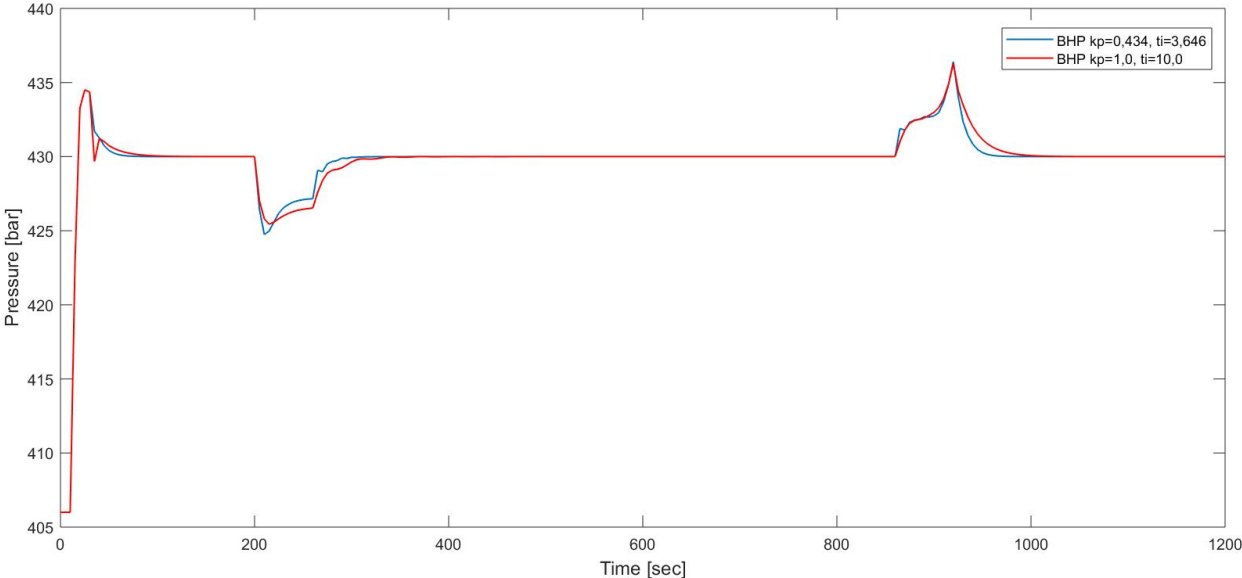


Figure15: show the BHP with viscosity 0,05 Pa s and two different regulator parameters

Figure 15 above illustrates the difference between the two regulator parameters described above. A slightly more satisfactory regulation was reached when using  $k_p = 1,0$ ,  $t_i = 10,0s$  gives  $k_i = 0,1$ .

8.1.1.2 Viscosity 0,001 Pa S:

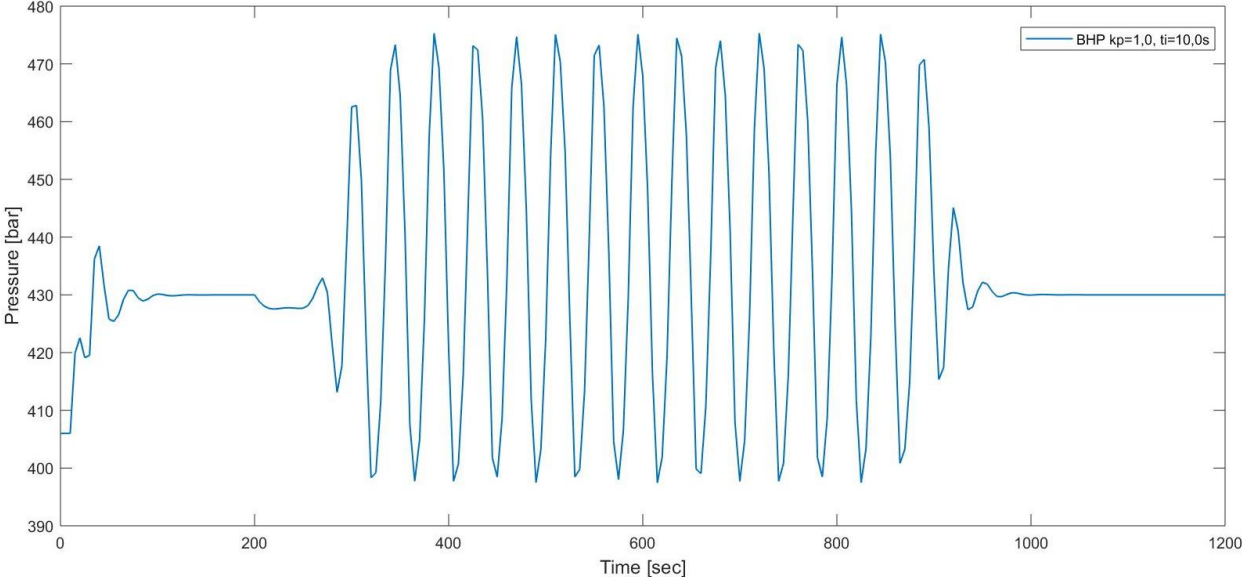
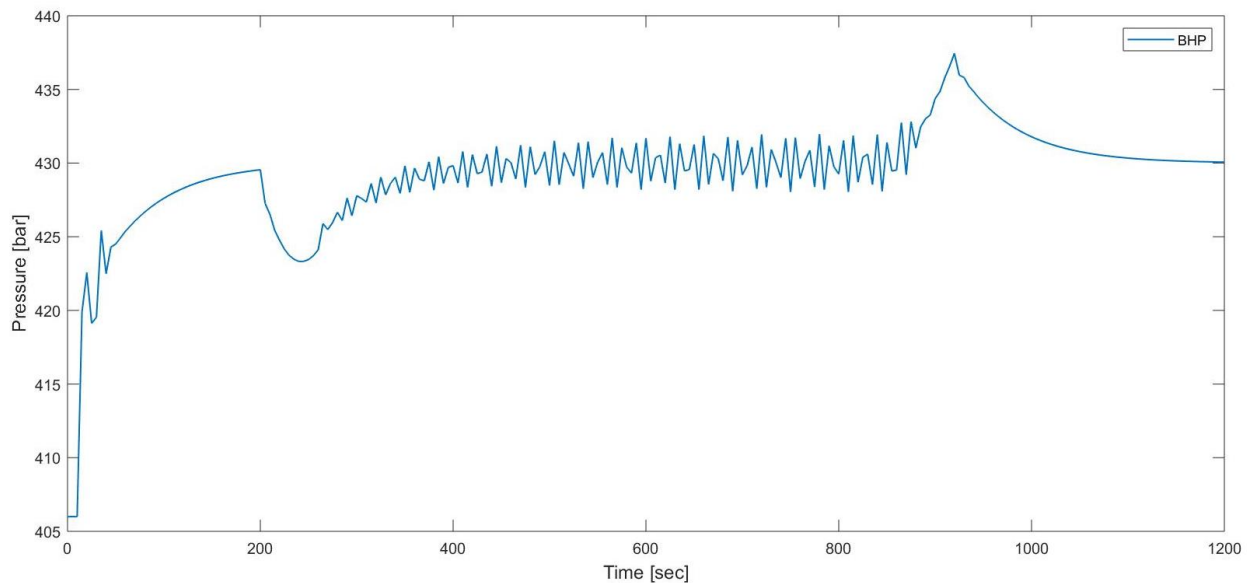


Figure 16: illustrates the BHP with viscosity 0,001Pa S while keeping the regulator parameter fixed ( $k_p = 1,0$ ,  $t_i = 10,0s$ )

The regulator parameters  $k_p$  and  $t_i$  was kept fixed from the previous run with viscosity 0,05 Pa S, when changing to viscosity 0,001Pa S. The figure 16 above show that the choke regulation parameters does not manage to keep the bottom hole pressure constant. Large pressure oscillations are seen during the static period in this case.

A new set of regulation parameters was found by the Ziegler Nichols' method, the critical gain was found to be 0,82 and the critical period was found to be 20 seconds. According to Ziegler Nichols method, the new  $k_p = 0,369$  and  $t_i = 16,67s$ .



*Figure 17: show the BHP with tuned PI parameters. Viscosity 0,001Pa s.*

Seen from figure17 the regulator shows oscillations during the 10 minutes connection. An explanation for this might be the pressure pulses creates disturbance for the tuning/regulation.

It was interesting to note that the chosen viscosity had impact on how the regulator parameters must be chosen. Different parameter sets had to be used in each case to stabilise the pressures. Hence, it is natural to think that this can have some relation to the generation of pressure pulses when we shut down the well. For the fluid with low viscosity, it takes time for these pulses to be dampened out when we have static conditions.

### 8.1.3 Simulation of different sampling interval

A test to see the effect of changing the sample time for the connection case was performed with viscosity 0,05Pa S.

Simulation was run with  $k_p = 1,0$ ,  $t_i = 10,0s$  gives  $k_i = 0,1$   $dt=0,02s$  ( $ts=dt$ ).

A new simulation by keeping the same regulator parameters changing sampling interval to  $ts=1,0s$ .

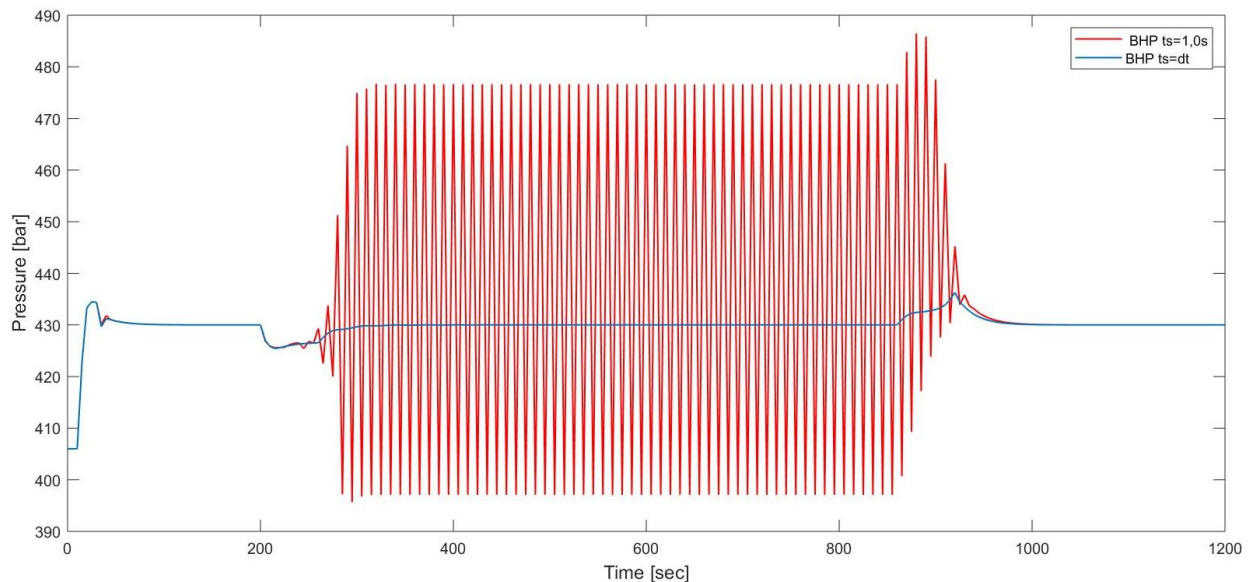


Figure 18: show the BHP with different sample time.  $ts=0,02s$  and  $ts=1,0s$ .

The effect of changing the sample time is shown in figure 18 above. When the sample time is increased large oscillations is present. A new set of regulator parameters is needed.

The new set of parameters where found to be  $k_p=0,45$  and  $t_i = 8,33$ .

To achieve  $k_p=0,45$  and  $t_i = 8,33$  it was tested to apply Ziegler-Nichols' directly from figure 2. Using 1,0 as critical gain and 10,0s as critical period. The result is shown in figure 19.

Below. When the sample time is increased to 1,0s. More error signal is calculated.  $K_i$  factor is 0,05 compared to 0,1 when sampling every 0,02 seconds. Hence the  $k_i$  factor must decrease to stabilize the pressure.

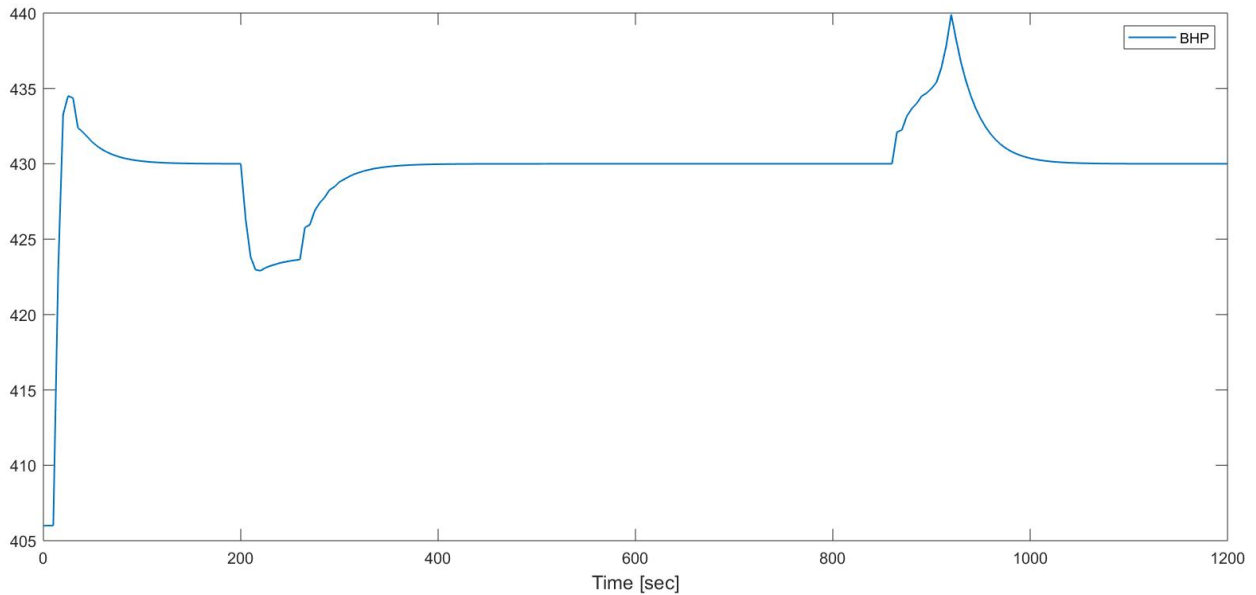


Figure 19: show BHP vs time,  $k_p= 0,45$  and  $t_i = 8,33.ts=1,0s$

In figure 19 the new set off regulation parameters are illustrated when  $ts=1,0s$ . it seemed easier to tune when sampling time was larger.  $K_p$  and  $t_i$  was reduced to maintain the regulation.

An observation is the BHP when the pumps are turned off, the BHP decreases to about 423 bar and increases to about 440 bar when turned back on. This might be related to the sampling interval calculates a larger error signal when the pumps are turned off/on.

#### 8.1.4 PI regulation with different pump start/close intervals

It was decided to alter the time interval for pump stop and start to check how that impacted the regulation parameters. One can have in mind that the reduction in friction per time unit when starting to shut down the pump will be lower. In addition, one can expect that the pressure pulses caused by acceleration is lower since the change in flow takes place over a longer time interval.

The simulation was between 60 seconds pump ramp up/down vs 20 seconds.

The cases illustrated above all had a pump ramp down/up of 60 seconds. The purpose of this test was to see what changes occurred when the pump ramp down/up time was decreased.



The regulation parameters where  $k_p = 1,0$ ,  $t_i = 10,0$  s. which gives  $k_i = 0,1$ . Viscosity  $0,05$  Pa S.

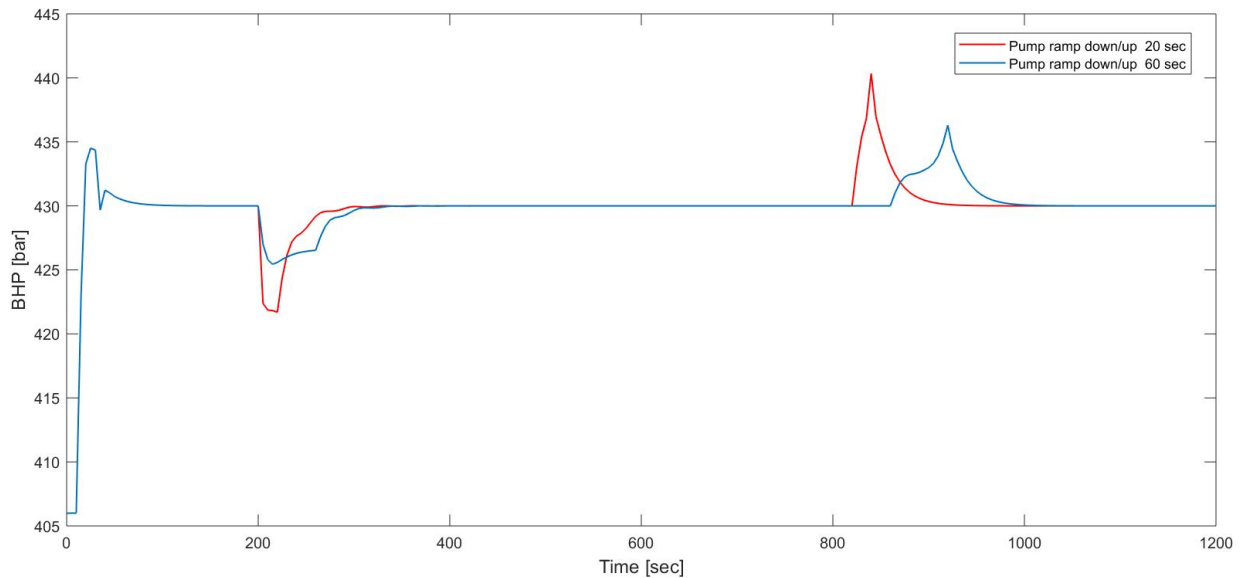


Figure 20: show the BHP when the pump off/on is altered with fixed regulation parameters.

The figure 20 above illustrates the BHP when pumps are turned off/on at different time. During the 20 seconds interval the bottom hole pressure decreases to about 422 bar and increases to about 440 when the pumps are turned back on.

During the 20 seconds it takes for the pumps to close, a larger pressure drop is seen. An explanation for this can be related to more rapid drop in pressure due to the loss of friction pressure. But there can also be an effect caused by increased acceleration effects (pressure pulses)

For the 60 seconds interval the BHP decreases to about 425 bar and show a smoother transaction from when the main pumps are turned off until the choke pressure takes over to maintain the bottom hole at the set pressure. As the main pumps are closed with a slower rate the control parameters, choke pressure has more time to compensate for the change.

## 8.2 Simulation of Kick

The kick simulations are run with the same well data as the connection case. 4000m vertical well, outer diameter ( $d_o$ ) = 8,5 inches (0,2159 m), inner diameter ( $d_i$ ) of 5,0 inches (0,127 m), and a liquid mass rate of 40 kg/s (2400) lpm, and fluid density 1000kg/m<sup>3</sup>. Set point 450 bar.

The scenario is that liquid circulation starts at 10 seconds and is kept constant at 40 kg/s until the end of simulation. At 30 seconds the setpoint is set to 450 bar. Gas is injected into the well at 200 seconds. the gas rate is increased linearly until 210 seconds. Then gas is injected with a constant rate until 300 seconds. From 300 to 310 seconds the gas rate is turned off. The liquid circulation of 40 kg/s continues until the kick is circulated out.

Different kick sizes were simulated to see if the regulation parameters ( $k_p=1,0$ ,  $t_i=10,0s$  gives  $k_i=0,1$ ) could be used for all three volumes.

### 8.2.1 Simulation with initial kick volumes 0,234m<sup>3</sup>, 0,948m<sup>3</sup> and 3,674m<sup>3</sup>

#### 8.2.1.1 Gas volume

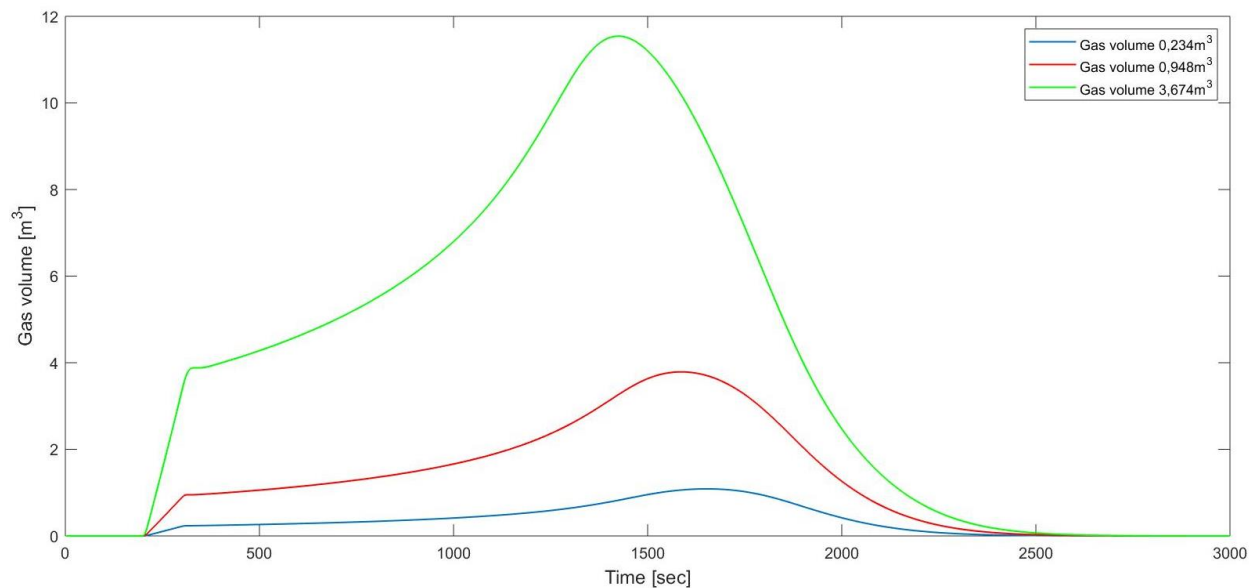


Figure 21: show the gas volume vs time for simulated kick sizes.

From figure 21 above it is illustrated at 310 seconds after the gas injection, the volumes are 0,234m<sup>3</sup>, 0,948m<sup>3</sup> and 3,674m<sup>3</sup> for the simulated cases.

Due to gas expansion the gas volumes increase with time as the gas travels toward the surface. The peaks indicate when gas is exiting the well. For the smallest kick (blue line), the peak is at 1665 seconds with volume 1,087 m<sup>3</sup>. The middle kick (red line) peaks at 1585 seconds with volume 3,788m<sup>3</sup>. After 1425 seconds the largest kick (green line) peaks with volume 11,54m<sup>3</sup>. Currently the gas exits the well.

Larger kick will decrease the hydrostatic pressure due to more gas expansion.

### 8.2.1.2 Liquid mass rate out

The circulation of the liquid is held constant at 40kg/s. Gas injection rate is 1,0kg/s for the smallest volume kick, 4,0kg/s for the middle, and 16,0kg/s for the largest volume kick.

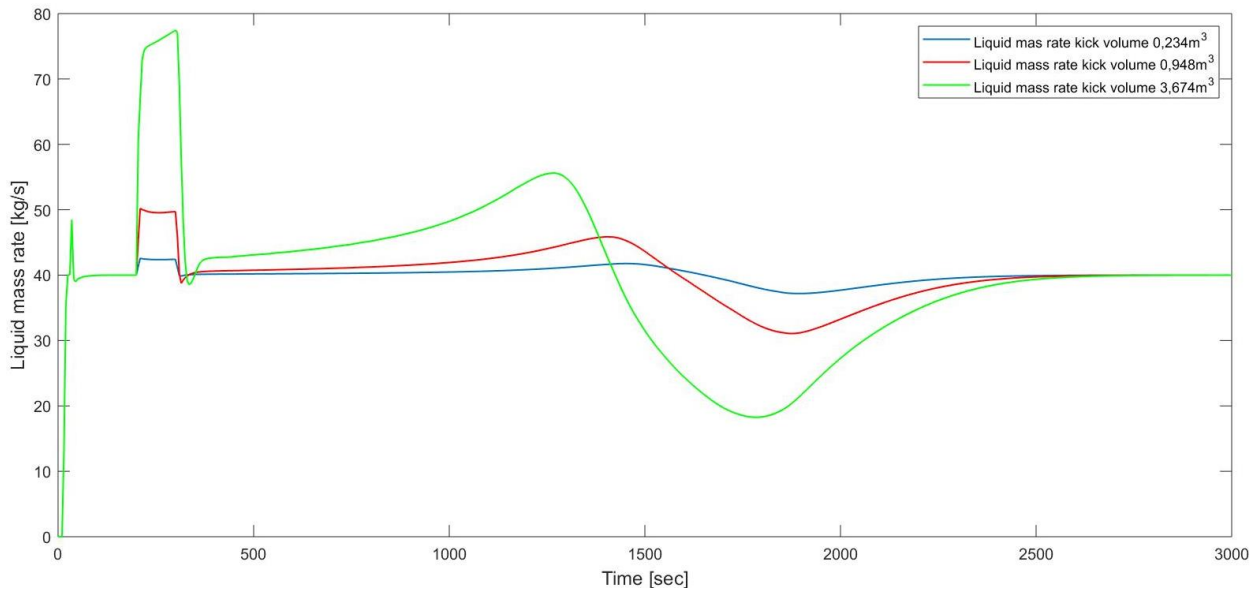


Figure 22: illustrates liquid mass rate vs time for the different simulation cases.

Figure 22 above, show the liquid mass rate out 40 kg/s where (2400lpm) is the circulation rate. Some increase is seen when the kick enters at 200 seconds to 310 second. This is because the gas displaces liquid resulting in more outflow than the inlet rate. After 310 second it shows liquid mass rate of 40 kg/s is steady, for the smallest kick. The kick reaches the surface around 1500 seconds where a small increase followed by a decrease is seen for the smallest kick volume.

For the larger kick volumes, it shows that the liquid mass rate deviates from the 40 kg/s due to the more expanded gas at it reaches the surface. This can be a challenge for the mud gas separator. Hence there is a limitation on how large kicks that can be circulated directly through the MPD equipment. Where the influx matrix described in the kick handling in CBHP-MPD system chapter is violated.

8.2.1.3 Gas mass rate out

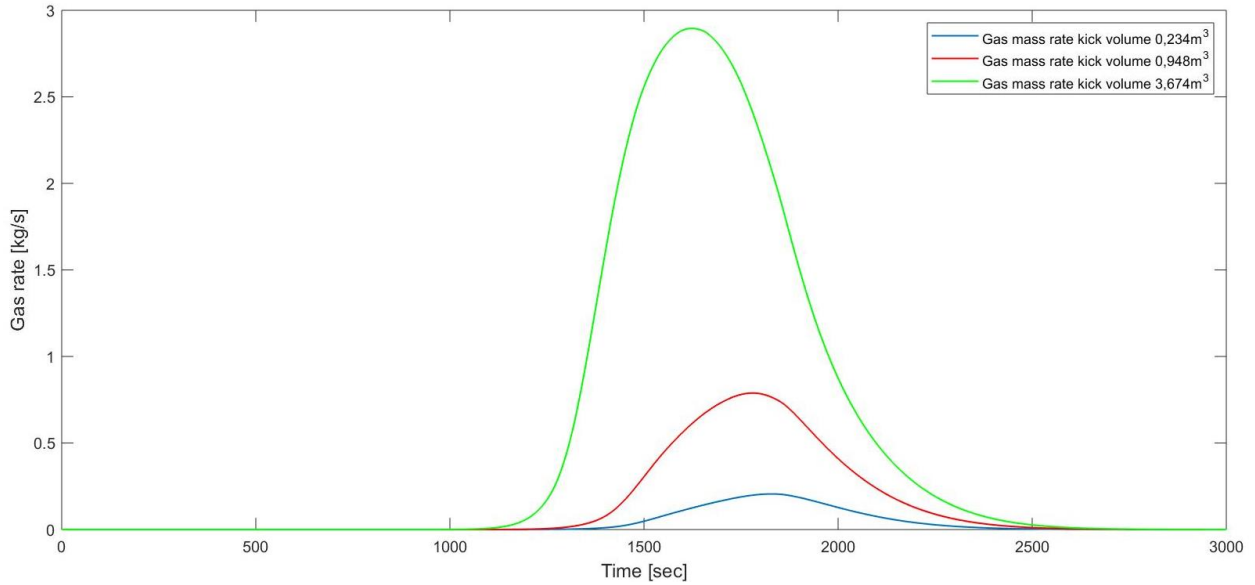


Figure 23: show gas rate vs time for 3 different kick volumes.

Figure 23 above show what time the gas will reach the surface. And with what rate it arrives.

The smallest kick reaches the surface at 1865 seconds with a mass rate of 0,2 kg/s. The gas rate increases from 0 to 0,2 kg/s in 575 seconds. Here we are only using 25 boxes and if a larger number of boxes had been used, a sharper transition would have been seen (D Gomes et al., 2018). This is because the numerical diffusion would have been reduced.

Also illustrated in the plot is that larger kick volumes have a much larger gas mass rate at surface. This can also be a challenge for the mud gas separator and puts a constraint on how large kicks that can be circulated.

Another observation is that larger kicks reaches the surface a bit faster. This can be related to the term  $K_{vmix}$  in the gas slip relation since more gas will lead to more expansion forcing the liquid faster out of the well. Hence, the mixture velocity will increase. The mixture velocity represents the average velocity. (D Gomes et al., 2018).

### 8.2.1.4 BHP

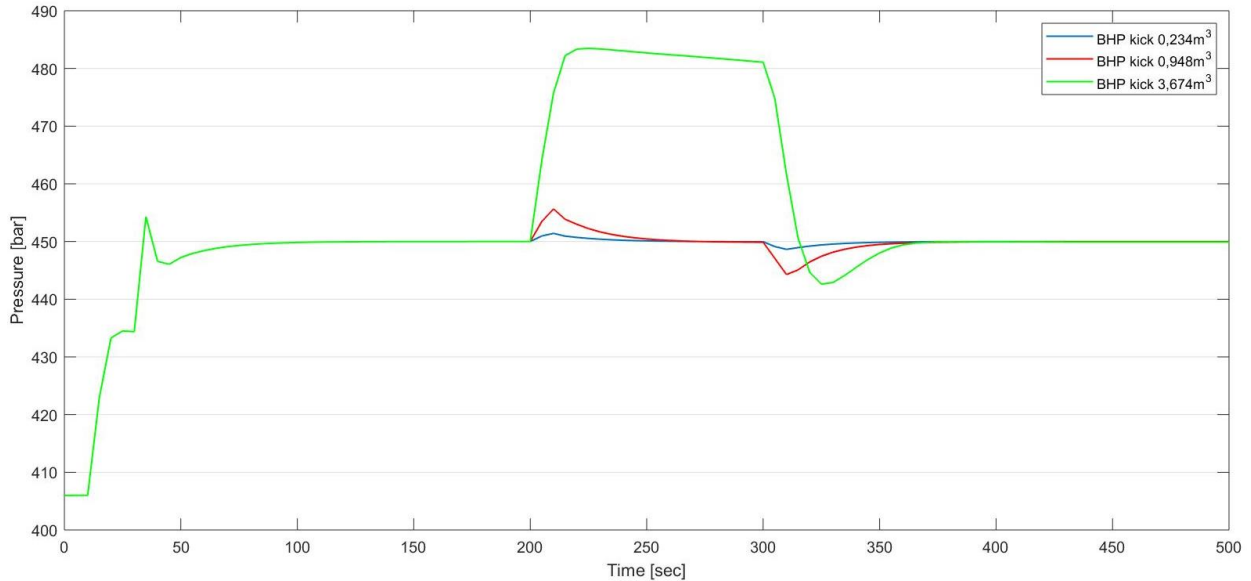


Figure 24: show the BHP vs time with 3 different kick volumes. (x axis set from 0 to 500)

The bottom hole pressure in the beginning of the simulation is illustrated in figure 24 above. From the plot one can see that the bottom hole is initially about 406 bar. When the regulation starts after 30 seconds the bottom hole pressure increases and settles at the set point 450 bar. At 200 seconds gas starts to enter the well and it is seen by the increase in BHP creating some disturbance in the bottom hole pressure readings.

Gas rate injection is kept steady until 300 seconds and is turned off at 310 seconds. During the injection period, the friction in the well will become larger because of increased rates. This is the way the kick is introduced in the simulator, but it may not represent how kicks are taken. They can for example be taken by swabbing or they can enter the well over a longer period.

For the smaller kicks, the regulator can keep the bottom hole pressure close to the setpoint, but some disturbances are seen, especially at start and end of the gas injection. The regulator manages this by reducing the choke pressure seen in (Fig 26).

but for the largest kick, it is seen that the choke pressure cannot be reduced sufficiently since it goes down to 1 bar.

Hence, the bottom hole pressure increases during the injection period due to the increased friction that cannot be compensated satisfactorily since the choke pressure cannot be reduced more. The lower limitation for choke pressure is set to 1 bar.

## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

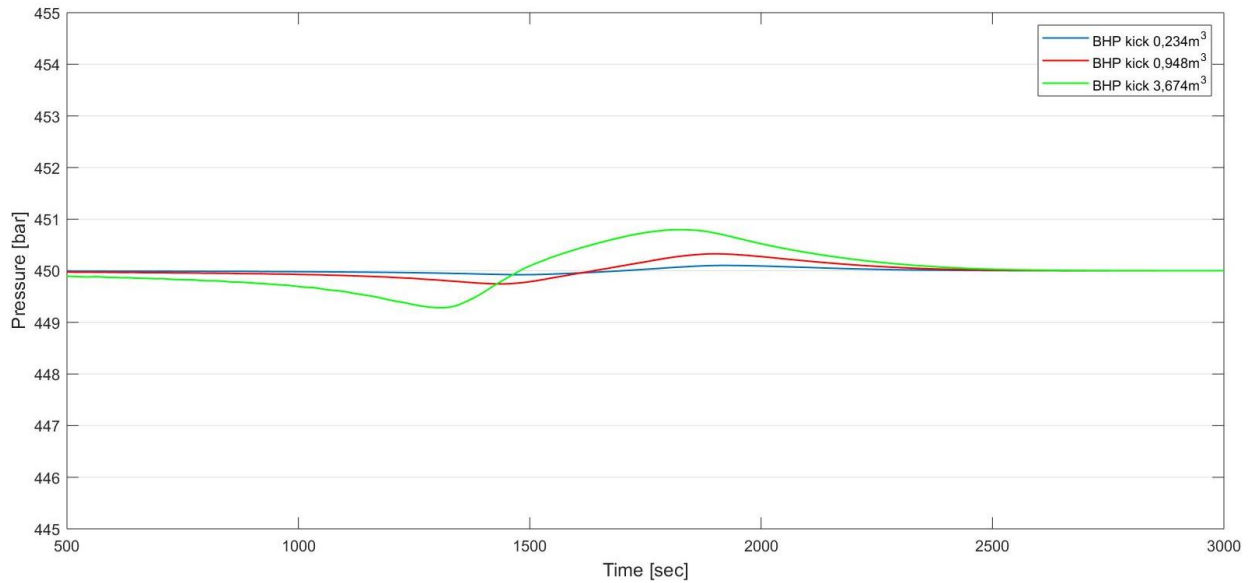


Figure 25: show the BHP with different kick vs time. (x axis from 500 to 3000 second, y axis 445 to 455 bar)

The focus of the simulation was to see how the PI regulator can control the BHP pressure when the kick is circulated towards surface. The gas expansion reduced the hydrostatic pressure which must be compensated by increasing the choke pressure. This is a slower process compared to e.g what is seen when friction in the well is altered by a change in flowrates. Friction acts fast while changes in hydrostatic pressure are usually slower.

In figure 25 above, the bottom hole pressure is plotted with time. The y axis is zoomed to see the small decrease at about 1500 seconds and increase at about 2000 seconds. These oscillations indicate when the gas reaches the surface and is circulated out. The bottom hole pressure oscillates 0,1 bar around the set point for the smallest kick volume. This show that the regulation parameters  $k_p = 1,0$  and  $t_i = 10,0s$  can keep the bottom hole pressure close to the setpoint 450 bar.

For kick volume  $0,948m^3$  the bottom hole pressure varies from 449,7 to 450,3 bar.

The largest kick volume,  $3,674m^3$ , the bottom hole pressure oscillates from 449,3 to 450,8 bar.

For all the three kick volume simulations, the PI regulator parameters  $k_p = 1,0$  and  $t_i = 10,0s$  managed to keep the bottom hole pressure close to the setpoint of 450 bar.

### 8.2.1.5 Choke pressure

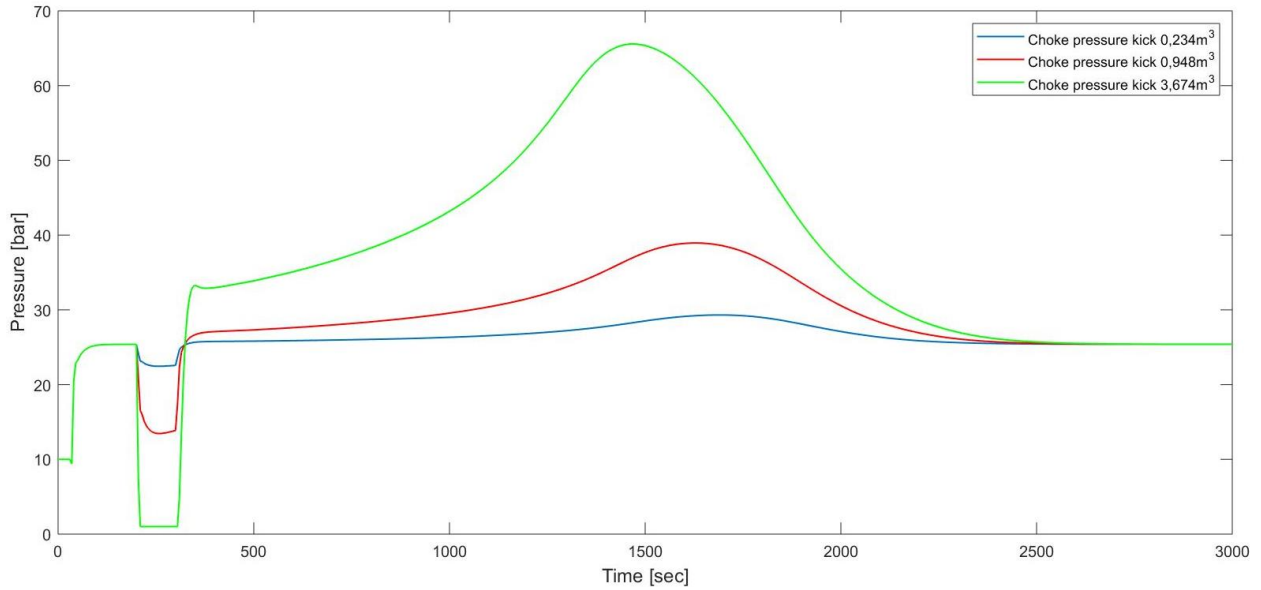


Figure 26: show choke pressure vs time for different kick volumes.

From figure 26 above the choke pressure will increase with time as the expanded gas reaches the exit point. This is because of the gas expansion decreases the hydrostatic pressure in the well. When the gas exits the well, the choke pressure reduces, and the hydrostatic pressure increases when only liquid remains in the well.

A larger kick volume will require a larger choke pressure to keep the bottom hole pressure at the setpoint. One can note that also the choke pressure can be subject to a pressure limitation. For instance, the rotating control device on top of the well, can only hold around 1500 psi (D. Gomes et al., 2018). This is responsible for sealing the well such that the flow can be led through the choke and hydrocarbons can be treated safely.

## 9. Discussion

Connection cases:

Implementation of a PI regulator in the AUSMV scheme was performed and it worked well when regulating on the choke differential pressure

For the regulator it showed that with PI parameters of  $k_p=1,0$  and  $t_i=10,0s$  worked very well. The reason for this is not fully understood and should be further investigated. These parameters barely exceeded a  $\pm 5$  bar pressure window for the BHP when pumps ramp up/down, viscosity  $0,05Pa\ S$ . The tuning trials did not show any improvements, hence more time optimizing the parameters could have been invested.

Comparing the Automation during connection example performed at Gulfax C in chapter 6.1.2 and the results from simulation of connection case. The results show similarities, when keeping the bottom hole constant. however, this might not be a realistic comparison due to the difference in density. The results in this thesis exceeds  $\pm 5$  bar pressure window.

When tuning the regulator with viscosity  $0,05 Pa\ S$ , the tuning might not have been done correctly. Manipulation with trial and error had to be done to get a similar regulation.

for the simulation with different sampling time, it showed that when the sampling time is altered a new set of regulation parameters had to be found. It seemed like it was easier to tune when the sampling time was larger. This might be related to the fact that the scheme runs with very small timestep compared to the regulation algorithm. It was also stated in (Godhavn, 2009) where the control system, hydraulic model used sampling time every 2 seconds.

This might also be related to when Ziegler-Nichol's performed their experiments in 1940 they used regulator types that deviates from the PID regulators used today.

Other tuning techniques could have been tested.

Trial and error method might be sufficient, although it might be time consuming.

Kick case:

The results indicate that the regulation parameters  $k_p=1$  and  $t_i=10,0s$  managed to keep the bottom hole pressure close to the set point, when all kick volumes were circulated out.



In MPD operations, the main constraints on how large kicks that can be circulated are related to what the maximum surface flow rates will be and what the max choke pressure will be. Hence, only small kick sizes can be circulated out. For larger kicks, the BOP is closed, and the kick is circulated out using the conventional method.

Hence, only small kicks are allowed to be circulated and we have demonstrated that for water-based mud and free gas, it is possible to regulate the bottom hole pressure quite well during kick circulation. But, for oil-based systems where gas solubility will come in to play, the kick behaviour will be different where it will transfer from dissolved to free gas form. Sharp changes in well geometry where a kick passes from a narrow well section to a wider riser will also cause changes in the hydrostatic pressure due to shortening of the kick height that has to be compensated for (D. Gomes et al., 2018). Hence, further investigations are recommended.

## 10. Conclusion

Implementation of a PI regulator in the AUSMV scheme was performed and it worked well when regulating on the choke differential pressure.

Investigation of acceleration and friction effects show that change in viscosity, sampling interval and pump ramp down/up time affects the regulation parameters new PI parameters needed to be found.

Pressure pulses created when shutting down the well impacted the regulation. When the viscosity is low this is more prominent.

Simulation with different sampling time show that new regulation parameters are needed. Due larger calculated error signal, the  $k_i$  factor needs to be reduced to maintain constant BHP.

Altering the pump ramp up/down from 60 sec to 20 sec, show a larger pressure drop for the lowest rate due to loss of friction pressure. For the 60 sec, the choke pressure has more time to compensate for the loss of friction.

Simulation of three different kick volumes, the regulation parameters ( $k_p = 1,0$  and  $t_i = 10,0s$  give  $k_i = 0,1$ ) manages to keep the BHP close to the set point when water based mud and free gas is simulated.

In MPD operation only small kick can be taken through the system due to surface equipment limitations.

Larger volumes due to expansion decreases the hydrostatic pressure.

The connection case without regulation, where the effect of the lower viscosity showed the BHP oscillating. A future study with a non-Newtonian drilling fluid to see how it impacted the pulses is recommended.

## 11. References

- Abimbola, M., Khan, F., Khakzad, N., & Butt, S. (2015). Safety and risk analysis of managed pressure drilling operation using Bayesian network. *Safety Science*, 76, 133-144. doi:10.1016/j.ssci.2015.01.010
- Benny, B., Hidayat, A. M. M., Karnugroho, A., Sosa, J., & Toralde, J. S. S. (2013). *Combining Pressurised Mud Cap Drilling (PMCD) and Early Kick Detection (EKD) Techniques for Fractured Formations Overlying a High Pressure Reservoir in Offshore Kalimantan*. Paper presented at the SPE Asia Pacific Oil and Gas Conference and Exhibition, Jakarta, Indonesia. <https://doi.org/10.2118/165893-MS>
- Benzoni-Gavage, S. (1991). *Analyse Numerique des Modeles Hydrodynamiques d'Ecoulements Diphasique Instationnaires dans les Reseaux de Production Petroliere*. (Doctorial), Claud Bernard University, Lyon France.
- Breyholtz, Ø., Nygaard, G., & Nikolaou, M. (2010). *Automatic control of managed pressure drilling*. Paper presented at the Proceedings of the 2010 American Control Conference.
- Bysveen, J., Fossli, B., Stenshorne, P. C., Skärgård, G., & Hollman, L. (2017). *Planning of an MPD and Controlled Mud Cap Drilling CMCD Operation in the Barents Sea Using the CML Technology*. Paper presented at the IADC/SPE Managed Pressure Drilling & Underbalanced Operations Conference & Exhibition, Rio de Janeiro, Brazil. <https://doi.org/10.2118/185286-MS>
- Cohen, J., Stave, R., Schubert, J., & Elieff, B. (2008). Dual-Gradient Drilling. In B. Rehm, J. Schubert, A. Haghshenas, A. S. Paknejad, & J. Hughes (Eds.), *Managed Pressure Drilling* (pp. 181-226): Gulf Publishing Company.
- Evje, S., & Fjelde, K. K. (2002). Hybrid flux-splitting schemes for a two-phase flow model. *Journal of Computational Physics*, 175(2), 674-701. doi:10.1006/jcph.2001.6962
- Fjelde, K. K., & Evje, S. (2010). *The AUSMV scheme - A simple but robust model for analyzing two-phase flow*. University of Stavanger.
- Fredericks, P. (2008). Constant Bottom-Hole Pressure with Pressure as a Primary Control. *Managed Pressure Drilling*. Editors: Rehm B., Schubert J., Haghshenas A., Paknejad AS, and Hughe J. In: Gulf Publishing Company, Houston, Texas, USA.
- Gabalton, O., Brand, P., Culen, M., Haq, I., Gonzalez-Luis, R., Silva, T., . . . Bacon, W. (2017). *Case study: first experience of developing an influx management envelope IME for a deepwater MPD operation*. Paper presented at the IADC/SPE Managed Pressure Drilling & Underbalanced Operations Conference & Exhibition.
- Gedge, B., Dalgit Singh, H. K., Bandico Refugio, E., Ta Quoc, B., & Bot, N. V. (2013). *Managed Pressure Drilling - A Solution for Drilling the Challenging and Un-Drillable Wells in Vietnam and South East Asia*. Paper presented at the SPE Asia Pacific Oil and Gas Conference and Exhibition, Jakarta, Indonesia. <https://doi.org/10.2118/165775-MS>
- Godhavn, J. M. (2009). *Control requirements for high-end automatic MPD operations*. Paper presented at the SPE/IADC Drilling Conference, Proceedings. Conference Paper retrieved from <https://www.scopus.com/inward/record.uri?eid=2-s2.0-70450042488&partnerID=40&md5=f67c3c55908497ddfa765f5e8a629ed2>
- Godhavn, J. M., & Knudsen, K. A. (2010). *High Performance and Reliability for MPD Control System Ensured by Extensive Testing*. Paper presented at the IADC/SPE Drilling Conference and Exhibition, New Orleans, Louisiana, USA. <https://doi.org/10.2118/128222-MS>
- Gomes, D., Frøyen, J., Fjelde, K., & Bjørkevoll, K. (2018). *A Numerical Comparison and Uncertainty Analysis of Two Transient Models for Kick Management in a Backpressure MPD System*. Paper presented at the SPE Norway One Day Seminar.
- Gomes, D., Nilsen, M. S., Frøyen, J., Bjørkevoll, K. S., Lage, A. C. V. M., Fjelde, K. K., & Sui, D. (2018). *A transient flow model for investigating parameters affecting kick behavior in OBM for HPHT wells and backpressure MPD systems*. Paper presented at the Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE.

- Haghshenas, A., Paknejad, A. S., Rehm, B., & Schubert, J. (2008). The why and basic principles of managed well-bore pressure. In *Managed pressure drilling* (pp. 1-38): Elsevier.
- Haugen, F. (2012). *Reguleringsteknikk* (2 ed.). Bergen: Fagbokforlaget.
- Moore, D. (2008). Mud cap drilling. In *Managed Pressure Drilling* (pp. 155-180): Elsevier.
- Norway, S. (2013). *Well integrity in drilling and well operations, NORSOK Standard D-010: , Rev. 4.* Standards Norway: Lysaker
- Saeed, S., Lovorn, R., & Arne Knudsen, K. (2012). *Automated drilling systems for MPD C-the reality.* Paper presented at the IADC/SPE Drilling Conference and Exhibition.
- Sui, D. (2019). [Lecture Notes ING 130].
- Syltoy, S., Eide, S. E., Berg, P. C., Torvund, S., Larsen, T., Fjeldberg, H., . . . Low, E. (2008). *Highly Advanced Multitechnical MPD Concept Extends Achievable HP/HT Targets in the North Sea.* Paper presented at the SPE/IADC Managed Pressure Drilling and Underbalanced Operations Conference and Exhibition, Abu Dhabi, UAE. <https://doi.org/10.2118/114484-MS>
- Udegbumam, J. E., Fjelde, K. K., Evje, S., & Nygaard, G. (2015). *On the Advection-Upstream-Splitting-Method Hybrid Scheme: A Simple Transient-Flow Model for Managed-Pressure-Drilling and Underbalanced-Drilling Applications.* Paper presented at the SPE Drilling & Completion. <https://doi.org/10.2118/168960-PA>
- Vogel, R., & Brugman, J. (2008). Continuous circulation system. In *Managed pressure drilling* (pp. 127-142): Elsevier.

## 12. Appendix

Matlab code

Color **red** indicates changes made

```
% Transient two-phase code based on AUSMV scheme: Gas and Water
% The code assumes uniform geometry

% time - Seconds

% p - pressure at new time level (Pa)
% dl - density of liquid at new time level (kg/m3)
% dg - density of gas at new time level (kg/m3)
% eg - phase volume fraction of liquid at new time level (0-1)
% ev - phase volume fraction of gas at new time level (0-1)
% vg - phase velocity of gas at new time level (m/s)
% vl - phase velocity of liquid at new time level (m/s)
% qv - conservative variables at new time level ( 3 in each cell)
% temp - temperature in well (K)

% po - pressure at old time level (Pa)
% dlo - density of liquid at old time level (kg/m3)
% dgo - density of gas at new old level (kg/m3)
% ego - phase volume fraction of liquid at old time level (0-1)
% evo - phase volume fraction of gas at old time level (0-1)
% vgo - phase velocity of gas at old time level (m/s)
% vlo - phase velocity of liquid at old time level (m/s)
% qvo - conservative variables at old time level ( 3 in each cell)
% temp - temperature in well (K)

clear;
t = cputime
tic,

% Geometry data/ Must be specified
wellddepth = 4000;
nobox = 25; %Number of boxes in the well

% Note that one can use more refined grid, 50, 100 boxes.
% When doing this, remember to reduce time step to keep the CFL number
% fixed below 0.25.. dt < cfl x dx/ speed of sound in water. If boxes are
% doubled, then half the time step.

nofluxes = noibox+1; % Number of cell boundaries
dx = wellddepth/nobox; % Boxlength
%dt = 0.005;
```

```
% Welldepth. Cell 1 start at bottom
x(1)= -1.0*welldepth+0.5*dx;
for i=1:nobox-1
  x(i+1)=x(i)+ dx;
end

% VERY IMPORTANT: BELOW THE TIMESTEP IS SET. MAKE SURE THAT THE
% CFL CONDITON IS FULFILLED. IF NUMBER OF BOXES IS CHANGED. DX
% WILL
% CHANGE AND DT HAS TO BE ADJUSTED TO KEEP THE CFL NUMBER FIXED.

dt= 0.02; % Timestep (seconds)
dtdx = dt/dx;
time = 0.0; % initial time.
endtime = 3000; % Time for ending simulation (seconds)
nosteps = endtime/dt; %Number of total timesteps. Used in for loop.
timebetweensavingtimedata = 5; % How often in s we save data vs time for plotting.
nostepsbeforesavingtimedata = timebetweensavingtimedata/dt;

% Additional parameters needed for control engineering below:
% Sampling intervall Ts- how often we update the control parameter.
% A variable that finds how many timesteps has to be performed with the
% flowmodel before updating the control parameters. (The scheme runs with
% very small timesteps compared to the PI algorith)

% We also initialize old and new error. Note that error
% is defined as setpoint minus measured value.

ts = dt; % Sampling intervall. Note minimum sampling time is defined by ``
% timestep of the scheme.
%ts = 1.0;

numberoftimestepsbefore_reg=ts/dt

%setp = 43000000; % 430 bars (but code works in Pascal)
setp = 45000000;
% Here on can play with the control parameters.
% Have divided Kp by ten for simplicity.
%
kp = 1.0;
ti = 10.00;
ki =kp/ti;

eold = 0;
enew = 0;
```

## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

```
%regtrue = 0.0 % 1.0 if regulate, zero if not.  
regtrue = 1.0
```

```
% Initial Boundary condition at outlet  
% HER SETTES DENNE TIL 10 BAR FOR Å ETTERLIGNE EN MPD OPERASJON  
%pbondout=100000; % Pascal (1 bar)  
pbondout=1000000; % Pascal (10 bar)  
pressureoutlet = pbondout;
```

```
% Slip parameters used in the gas slip relation.  $v_g = K v_{mix} + S$   
k = 1.2;  
s = 0.55;
```

```
% Initial temperature distribution. (Kelvin)  
% Note that this is only used if we use density models that depend on  
% temperature
```

```
tempbot = 110+273;  
temptop = 50+273;  
tempgrad= (tempbot-temptop)/welldepth;  
tempo(1)=tempbot-dx/2*tempgrad;  
for i = 1:nobox-2  
    tempo(i+1)=tempo(i)-dx*tempgrad;  
end  
tempo(nobox)=tempo(nobox-1)-dx*tempgrad;
```

```
temp = tempo;
```

```
% Different fluid density parameters  
% Note how we switch between different models later.  
% These parameters are used when finding the  
% primitive variables pressure, densities in an analytical manner.  
% Changing parameters here, you must also change parameters inside the  
% density routines roliq and rogas.
```

```
% Simple Water density model & Ideal Gas. See worknote Extension of AUSMV  
% scheme.
```

```
rho0=1000; % Water density at STC (Standard Condition) kg/m3  
Bbeta=2.2*10^9; % Parameter that depend on the compressibility of water  
Alpha=0.000207; % Parameter related to thermal expansion/compression  
R = 286.9; % Ideal gas parameter  
P0=100000; % Pressure at STC (Pa)  
T0=15+273.15; % Temperature at STC (K)
```

## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

% Very simple models (PET510 compendium)

al = 1500; % Speed of sound in water.

rt= 100000; % Ideal gas parameter in model  $\rho_{og} = p/rt$  ( $rt = ag^2$ )

rho0=1000; % Water density at STC (Standard Condition) kg/m<sup>3</sup>

P0=100000; % Pressure at STC (Pa)

T0=15+273.15; % Temperature at STC (K)

% Viscosities (Pa\*s)/Used in the frictional pressure loss model (dpfric).

% viscl = 0.001; % Liquid phase

viscl = 0.05

viscg = 0.0000182; % Gas phase

% Gravity constant

g = 9.81; % Gravitational constant m/s<sup>2</sup>

% Well opening. opening = 1, fully open well, opening = 0 (<0.01), the well  
% is fully closed. This variable will control what boundary conditions that  
% will apply at the outlet (both physical and numerical): We must change  
% this further below in the code if we want to change status on this.

wellopening = 1.0; % This variable determines if  
% the well is closed or not, wellopening = 1.0 -> open. wellloping = 0  
%-> Well is closed. This variable affects the boundary treatment.

bullheading = 0.0; % This variable can be set to 1.0 if we want to simulate  
% a bullheading operation. But the normal is to set this to zero.

% Specify if the primitive variables shall be found either by  
% a numerical or analytical approach. If analytical = 1, analytical  
% solution is used. If analytical = 0. The numerical approach is used.  
% using the itsolver subroutine where the bisection numerical method  
% is used. We use analytical.

analytical = 1;



## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

```
% Initialization of rest of geometry.  
% Here we specify the outer and inner diameter and the flow area  
% We assume 12.25 x 5 inch annulus. But this can be modified.  
  
for i = 1:nobox  
  
% do(i)=0.331;  
  
do(i)= 0.2159  
di(i) = 0.127;  
  
area(i) = 3.14/4*(do(i)*do(i)- di(i)*di(i));  
  
end  
  
% Initialization of slope limiters. These are used for  
% reducing numerical diffusion and will be calculated for each timestep.  
% They make the numerical scheme second order.  
for i = 1:nobox  
s1(i)=0;  
s2(i)=0;  
s3(i)=0;  
s4(i)=0;  
s5(i)=0;  
s6(i)=0;  
end  
  
% Now comes the intialization of the physical variables in the well.  
% First primitive variables, then the conservative ones.  
  
% Below we intialize pressure and fluid densities. We start from top of  
% the well and calculated downwards. The calculation is done twice with  
% updated values to get better approximation. Only hydrostatic  
% considerations since we start with a static well.  
  
for i = 1:nobox  
eg(i)=0.0; % Gas volume fraction
```

## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

```
ev(i)=1-eg(i); % Liquid volume fraction  
end
```

```
p(nobox)= pbondout+0.5*9.81*dx*...  
(ev(nobox)*rholiq(P0,T0)+eg(nobox)*rogas(P0,T0)); % Pressure (Pa)  
dl(nobox)=rholiq(p(nobox),tempo(nobox)); % Liquid density kg/m3  
dg(nobox)=rogas(p(nobox),tempo(nobox)); % Gas density kg/m3
```

```
for i=nobox-1:-1:1  
p(i)=p(i+1)+dx*9.81*(ev(i+1)*dl(i+1)+eg(i+1)*dg(i+1));  
dl(i)=rholiq(p(i),tempo(i));  
dg(i)=rogas(p(i),tempo(i));  
end
```

```
for i=nobox-1:-1:1  
rhoavg1=(ev(i+1)*dl(i+1)+eg(i+1)*dg(i+1));  
rhoavg2=(ev(i)*dl(i)+eg(i)*dg(i));  
p(i)=p(i+1)+dx*9.81*(rhoavg1+rhoavg2)*0.5;  
dl(i)=rholiq(p(i),tempo(i));  
dg(i)=rogas(p(i),tempo(i));
```

```
end
```

```
% Initialize phase velocities, volume fractions, conservative variables  
% and friction and hydrostatic gradients.  
% The basic assumption is static fluid, one phase liquid.
```

```
for i = 1:nobox  
vl(i)=0; % Liquid velocity new time level.  
vg(i)=0; % Gas velocity at new time level  
eg(i)=0.0; % Gas volume fraction  
ev(i)=1-eg(i); % Liquid volume fraction  
qv(i,1)=dl(i)*ev(i)*area(i); % Conservative variable for liquid mass (kg/m)  
qv(i,2)=dg(i)*eg(i)*area(i); % Conservative variable for gas mass (kg/m)  
qv(i,3)=(dl(i)*ev(i)*vl(i)+dg(i)*eg(i)*vg(i))*area(i); % Conservative variable for mixture  
moementum  
fricgrad(i)=0; % Pa/m  
hydgrad(i)=g*(dl(i)*ev(i)+eg(i)*dg(i)); % Pa/m  
end
```

```
% Section where we also initialize values at old time level
```

```
for i=1:nobox  
dlo(i)=dl(i);  
dgo(i)=dg(i);
```

## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

```
po(i)=p(i);
ego(i)=eg(i);
evo(i)=ev(i);
vlo(i)=vl(i);
vgo(i)=vg(i);
qvo(i,1)=qv(i,1);
qvo(i,2)=qv(i,2);
qvo(i,3)=qv(i,3);
end

% Intialize fluxes between the cells/boxes

for i = 1:nofluxes
    for j =1:3
        flc(i,j)=0.0; % Flux of liquid over box boundary
        fgc(i,j)=0.0; % Flux of gas over box boundary
        fp(i,j)= 0.0; % Pressure flux over box boundary
    end
end

% Main program. Here we will progress in time. First som intializations
% and definitions to take out results. The for loop below runs until the
% simulation is finished.

countsteps = 0;
counter=0;
countreg=0; % Intialization of the counter that shall be
            % used to find out when it is needed to update the
            % control parameters

printcounter = 1;
pin(printcounter) = (p(1)+dx*0.5*hydgrad(1))/100000; % Pressure in bar at bottom for time
storage
pout(printcounter)= pbondout/100000; % Pressure at outlet of uppermost cell
pnobox(printcounter)= p(nobox)/100000; % Pressure in middle of uppermost cell
liquidmassrateout(printcounter) = 0; % liquid mass rate at outlet kg/s
gasmassrateout(printcounter)=0; % gass mass rate at outlet kg/s
timeplot(printcounter)=time; % Array for time and plotting of variables vs time
pitvolume=0;
pitrates =0;
pitgain(printcounter)=0;

pchoke(printcounter)=pbondout/100000;

% zopning = 0
% chokeopening(printcounter)=zopning;
% setpoint(printcounter)=setp/100000;
```

```
kickvolume=0;  
bullvolume=0;
```

```
% The temperature is not updated but kept fixed according to the  
% initialization.
```

```
% Now comes the for loop that runs forward in time. This is repeated for  
% every timestep.
```

```
for i = 1:nosteps  
    countsteps=countsteps+1;  
    counter=counter+1;  
    % Update the countreg variable to keep count on  
    % when it is time to update the control parameters.
```

```
    countreg = countreg+1;  
    time = time+dt; % Step one timestep and update time.
```

```
% Then a section where specify the boundary conditions.  
% Here we specify the inlet rates of the different phases at the  
% bottom of the pipe in kg/s. We interpolate to make things smooth.  
% It is also possible to change the outlet boundary status of the well  
% here. First we specify rates at the bottom and the pressure at the outlet  
% in case we have an open well. This is a place where we can change the  
% code to control simulations. If the well shall be close, wellopening must  
% be set to 0. It is also possible to reverse the flow (bullheading).
```

```
% In the example below, we take a gas kick and then circulate this  
% out of the well without closing the well. (how you not should perform  
% well control)
```

```
% Note there are two variables wellopening and bullheading that can be  
% changed in the control structure below to close the well or start  
% reversing the flow i.e. pumping downwards.
```

```
% Note that if we will change to bullheading throughout the control stucture,  
% the variable inletligmassrate  
% has to be defined as negative since pumping downwards at outlet will be  
% in negative direction (postive direction of flow has been chosen to be  
% upwards)
```

```
% NB, NOTE THAT THIS IS ONE OF THE MAIN PLACES WHERE YOU HAVE TO  
ADJUST THE  
% CODE TO CONTROL THE SIMULATION SCENARIO.
```

XX = 1.0; % Gasrate in kg/s  
%XX = 0,0;

YY= 40.0; % Liquidrate in kg/s

if (time < 10)

inletligmassrate=0.0;  
inletgasmassrate=0.0;

elseif ((time>=10) & (time < 20))  
inletligmassrate = YY\*(time-10)/10; % Interpolate the rate from 0 to value wanted.  
inletgasmassrate = 0

elseif ((time >=20) && (time<200))  
inletligmassrate = YY;  
inletgasmassrate = 0;

elseif ((time >=200) && (time<210))  
inletligmassrate = YY;  
inletgasmassrate = XX\*(time-200)/10; % gas influx

elseif ((time >=210) && (time<300))  
inletligmassrate = YY;  
inletgasmassrate = XX;

elseif ((time >=300) && (time<310))  
inletligmassrate = YY;  
inletgasmassrate = XX-XX\*(time-300)/10;

elseif (time > 310)  
inletligmassrate=YY;  
inletgasmassrate=0;

end

% elseif ((time >=200) & (time<260))  
% % inletligmassrate = YY-YY\*(time-200)/10;  
% inletligmassrate = YY-YY\*(time-200)/60;  
% inletgasmassrate = XX-XX\*(time-200)/60;  
% elseif (time > 260 & time < 860) % 10 minutt nedstengning connection  
% inletligmassrate=0;  
% inletgasmassrate=0;  
%  
% elseif (time > 860 & time < 920)  
% inletligmassrate=YY\*(time-860)/60;  
% inletgasmassrate=0;

## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

```
% The commented code below are from some previous runs. It shows. e.g. how
% we can close the well.
%elseif((time>=500)&(time<510))
% inletligmassrate = YY-YY*(time-500)/10;
% inletgasmassrate = XX-XX*(time-500)/10;
% elseif(time>=510)
% inletligmassrate=0;
% inletgasmassrate=0;
% wellopening=0.0;
% end

%XX = 4;
% XX (kg/s) is a variable for introducing a kick in the well.
%YY = 15; % Liquid flowrate (kg/s) (1 kg/s = 1 l/s approx)
% if (time < 10)
%
% inletligmassrate=0.0;
% inletgasmassrate=0.0;
%
% elseif ((time>=10) & (time < 20))
% inletligmassrate = 0*(time-10)/10;
% inletgasmassrate = XX*(time-10)/10;
%
% elseif ((time >=20) & (time<110))
% inletligmassrate = 0;
% inletgasmassrate = XX;
%
% elseif ((time>=110)& (time<120))
% inletligmassrate = 0;
% inletgasmassrate = XX-XX*(time-110)/10;
% elseif ((time>=120&time<130))
% inletligmassrate =0;
% inletgasmassrate =0;
% elseif ((time>=130)&(time<300))
% inletligmassrate =0;
% inletgasmassrate =0;
% elseif ((time>=300)&(time<310))
% inletligmassrate= YY*(time-300)/10;
% inletgasmassrate =0;
% elseif((time>=310))
% inletligmassrate= YY;
% inletgasmassrate =0;
% end

kickvolume = kickvolume+inletgasmassrate/dgo(1)*dt; % Here we find the kickvolume
% initially induced in the well.
```

```
% Here we specify the physical outlet pressure. Here we have given the pressure as  
% constant. It would be possible to adjust it during openwell conditions  
% either by giving the wanted pressure directly (in the command lines  
% above) or by finding it indirectly through a chokemodel where the chokeopening  
% would have had to be an input parameter. The chokeopening variable would equally had  
% to be adjusted inside the control structure given above.
```

```
% HER KAN VI GÅ INN OG SETTE CHOKETRYKKET DIREKTE SAMT  
% LEGGE INN PI REGULERING. VI LEGGER DETTE FØR BEREGNING AV  
% VERDIER PÅ DET  
% NYE TIDSTEGET. DVS VI OPPDATERTARER RANDKRAVET FØR VI  
% OPPDATERER ALLE  
% CELLENE MED AUSMV SKJEMAET.
```

```
if regtrue == 1.0
```

```
    % If this is true, we can regulate.
```

```
if time < 30
```

```
    % We dont have any regulation before 30 seconds. It has just been chosen
```

```
    % lik this.
```

```
    pressureoutlet = pbondout; % set equal to the initial outlet pressure.
```

```
    eold = 0;
```

```
    enew = 0;
```

```
    countreg = 0;
```

```
elseif time >= 30
```

```
    if countreg >= numberoftimestepsbefore_reg
```

```
        % Here it is time to update the control parameter
```

```
        measuredbhp = p(1)+hydgrad(1)*0.5*dx+fricgrad(1)*0.5*dx;
```

```
        % Note that to get the exact position of the BHP at the inlet
```

```
        % of the lowermost box, we subtract "half a cell" since
```

```
        % p(1) is defined in the center of the cell. Then one omits
```

```
        % problems with having different positions if the grid is refined
```

```
        % etc.
```

```
        enew = setp-measuredbhp; % e defined as setpoint minus measurement
```

```
        pressureoutlet = pressureoutlet+kp*(enew-eold)+ki*enew*ts;
```

```
        % note that pressureoutlet at the left side of = will contain the
```

```
        % old choke pressure.
```

```
        eold = enew;
```

```
        countreg = 0;
```

```
        % Legger inn en nedre begrensning på choketrykk.
```

```
        % En kan sjekke med teori i bøker om det er fornuftig å legge inn
```

## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

```
% noe sånn. En kan feks ikke opererer med negative trykk og
% det å sette atmosfærisk trykk som nedre begrensning kan kanskje
% være fornuftig.

if pressureoutlet < 100000
    pressureoutlet = 100000.0;
end

else
    % here we dont update since we have not reached the sampling time
    % yet.
    % i.e. we dont update enew, eold and eold.
    % they will first be updated the next time sampling timeintervall is
    % exceeded.
end

% Here we are finished with the updating of pressureoutlet.
% This variable will now be used in the boundary condition treatment
% below.

end

kv = 0.021;

%kv = 0.00021;
deltaPV = pressureoutlet-100000; % for å gage pressure

% Q =A*v
% Q =A*vmix % Tofase

% Qrate = area(nobox)*(vlo(nobox)*evo(nobox)+vgo(nobox)*ego(nobox)); % m3/s
% mixturetetthet = dlo(nobox)*evo(nobox)+dgo(nobox)*ego(nobox); % kg/m3
% zopning = (Qrate*Qrate*mixturetetthet/(kv^2*deltaPV))^0.5; % fraksjon 0-1
%

% Q = Kv*z*(pv/rho)^0.5
% Anta at choketrykkfallet skal være 1 bar, anta og full ventilåpning.
% 2400 l/min = 2400/1000/60 = 0.04 m3/s

% 0,04 = kv*(100000/1000)^0.5 =kv*10
% kv = 0,04/10 = 0,004

% 0,04 = kv*(1000)^0.5, antatt 10 bar trykkfall ved chokeåpning lik 1
% kv = 0,00129

end
```



## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

```
% Based on these given physical boundary values combined with use
% of extrapolations techniques
% for the remaining unknowns at the boundaries, we will define the mass and
% momentum fluxes at the boundaries (inlet and outlet of pipe).

% inlet/bottom fluxes first.
if (bullheading<=0)
% Here we pump from bottom
flc(1,1)= inletligmassrate/area(1);
flc(1,2)= 0.0;
flc(1,3)= flc(1,1)*vlo(1);

fgc(1,1)= 0.0;
fgc(1,2)= inletgasmassrate/area(1);
fgc(1,3)= fgc(1,2)*vgo(1);

fp(1,1)= 0.0;
fp(1,2)= 0.0;

% Old way of treating the boundary
% fp(1,3)= po(1)+0.5*(po(1)-po(2)); % Interpolation used to find the
% pressure at the inlet/bottom of the well.

% New way of treating the boundary
fp(1,3)= po(1)...
+0.5*dx*(dlo(1)*evo(1)+dgo(1)*ego(1))*g...
+0.5*dx*fricgrad(1);

else
% Here we pump from the top. All masses are assumed to flow out of the
% well into the formation. We use first order extrapolation.
flc(1,1)=dlo(1)*evo(1)*vlo(1);
flc(1,2)=0.0;
flc(1,3)=flc(1,1)*vlo(1);

fgc(1,1)=0.0;
fgc(1,2)=dgo(1)*ego(1)*vgo(1);
fgc(1,3)=fgc(1,2)*vgo(1);

fp(1,1)=0.0;
fp(1,2)=0.0;
fp(1,3)=20000000; (Pa) % This was a fixed pressure set at bottom when bullheading
end
```

% Outlet fluxes (open & closed conditions)

if (wellopening>0.01)

% Here open end conditions are given. We distinguish between bullheading  
% & normal circulation.

if (bullheading<=0) % Here we dont bullhead, i.e we circulate from bottom

% Here the is normal ciruclation and open well)

flc(nofluxes,1)= dlo(nobox)\*evo(nobox)\*vlo(nobox);

flc(nofluxes,2)= 0.0;

flc(nofluxes,3)= flc(nofluxes,1)\*vlo(nobox);

fgc(nofluxes,1)= 0.0;

fgc(nofluxes,2)= dgo(nobox)\*ego(nobox)\*vgo(nobox);

% fgc(nofluxes,2)=0; Activate if gas is sucked in!?

fgc(nofluxes,3)= fgc(nofluxes,2)\*vgo(nobox);

fp(nofluxes,1)= 0.0;

fp(nofluxes,2)= 0.0;

fp(nofluxes,3)= pressureoutlet;

else

% Here we are bullheading.

flc(nofluxes,1)= inletligmassrate/area(nobox);

flc(nofluxes,2)= 0.0;

flc(nofluxes,3)= flc(nofluxes,1)\*vlo(nobox);

fgc(nofluxes,1)=0.0;

fgc(nofluxes,2)=0.0;

fgc(nofluxes,3)=0.0;

fp(nofluxes,1)=0.0;

fp(nofluxes,2)=0.0;

fp(nofluxes,3)= po(nobox)...

-0.5\*dx\*(dlo(nobox)\*evo(nobox)+dgo(nobox)\*ego(nobox))\*g...

+0.5\*dx\*fricgrad(nobox); %check sign here on friction

% Physically, the friction should be added when going from

% mid point in upper cell to outlet. But if fricgrad(nobox) is

% negative there should be a minus in front of the term to have

% + in the end.

end

else

% Here closed end conditions are given

flc(nofluxes,1)= 0.0;

flc(nofluxes,2)= 0.0;

flc(nofluxes,3)= 0.0;

```

fgc(nofluxes,1)= 0.0;
fgc(nofluxes,2)= 0.0;
fgc(nofluxes,3)= 0.0;

fp(nofluxes,1)=0.0;
fp(nofluxes,2)=0.0;

% Old way of treating the boundary
% fp(nofluxes,3)= po(nobox)-0.5*(po(nobox-1)-po(nobox));

% New way of treating the boundary
fp(nofluxes,3)= po(nobox)...
-0.5*dx*(dlo(nobox)*evo(nobox)+dgo(nobox)*ego(nobox))*g;
% -0.5*dx*fricgrad(nobox); % Neglect friction since well is closed.
end

% Implementation of slopelimiters. They are applied on the physical
% variables like phase densities, phase velocities and pressure.

% It was found that if the slopelimiters were set to zero in
% the boundary cells, the pressure in these became wrong. E.g. the upper
% cell get an interior pressure that is higher than it should be e.g. when
% being static (hydrostatic pressure was too high). The problem was reduced
% by copying the slopelimiters from the interior cells. However, both
% approaches seems to give the same BHP pressure vs time but the latter
% approach give a more correct pressure vs depth profile. It is also better
% to use when simulating pressure build up where the upper cell pressure
% must be monitored. It should be checked more in detail before concluding.
% BUT; there has been mass conservation problems with the scheme for the
% case where the slopelimiters were copied (see master thesis of Keino)
% A possible fix has been included below where the slopelimiter related to
% the gas volume fraction is set to zero in the first cell.

for i=2:nobox-1
sl1(i)=minmod(dlo(i-1),dlo(i),dlo(i+1),dx);
sl2(i)=minmod(po(i-1),po(i),po(i+1),dx);
sl3(i)=minmod(vlo(i-1),vlo(i),vlo(i+1),dx);
sl4(i)=minmod(vgo(i-1),vgo(i),vgo(i+1),dx);
sl5(i)=minmod(ego(i-1),ego(i),ego(i+1),dx);
sl6(i)=minmod(dgo(i-1),dgo(i),dgo(i+1),dx);
end

% Slopelimiters in outlet boundary cell are set to zero!
% sl1(nobox)=0;
% sl2(nobox)=0;
% sl3(nobox)=0;
% sl4(nobox)=0;
% sl5(nobox)=0;

```

## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

```
% sl6(nobox)=0;

% Slopelimiters in outlet boundary cell are copied from neighbour cell!
sl1(nobox)=sl1(nobox-1);
sl2(nobox)=sl2(nobox-1);
sl3(nobox)=sl3(nobox-1);
sl4(nobox)=sl4(nobox-1);
sl5(nobox)=sl5(nobox-1);
sl6(nobox)=sl6(nobox-1);

% Slopelimiters in inlet boundary cell are set to zero!
% sl1(1)=0;
% sl2(1)=0;
% sl3(1)=0;
% sl4(1)=0;
% sl5(1)=0;
% sl6(1)=0;

% Slopelimiters in inlet boundary cell are copied from neighbour cell!
sl1(1)=sl1(2);
sl2(1)=sl2(2);
sl3(1)=sl3(2);
sl4(1)=sl4(2);
sl5(1)=sl5(2);
sl6(1)=sl6(2);

% FIX FOR OMITTING THE GAS MASS CONSERVATION PROBLEM
sl5(1)=0;

% Now we will find the fluxes between the different cells.
% NB - IMPORTANT - Note that if we change the compressibilities/sound velocities of
% the fluids involved, we may need to do changes inside the csound function.
% But the effect of this is unclear.

for j = 2:nofluxes-1

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

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% First order method is from here: If you want to test this, activate this
% and comment the second order code below.
% cl = csound(ego(j-1),po(j-1),dlo(j-1),k);
% cr = csound(ego(j),po(j),dlo(j),k);
% c = max(cl,cr);
% pll = psip(vlo(j-1),c,evo(j));
% plr = psim(vlo(j),c,evo(j-1));
```

## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

```

%   pgl = psip(vgo(j-1),c,ego(j));
%   pgr = psim(vgo(j),c,ego(j-1));
%   vmixr = vlo(j)*evo(j)+vgo(j)*ego(j);
%   vmixl = vlo(j-1)*evo(j-1)+vgo(j-1)*ego(j-1);
%
%   pl = pp(vmixl,c);
%   pr = pm(vmixr,c);
%   mll= evo(j-1)*dlo(j-1);
%   mlr= evo(j)*dlo(j);
%   mgl= ego(j-1)*dgo(j-1);
%   mgr= ego(j)*dgo(j);
%
%   flc(j,1)= mll*pll+mlr*plr;
%   flc(j,2)= 0.0;
%   flc(j,3)= mll*pll*vlo(j-1)+mlr*plr*vlo(j);
%
%   fgc(j,1)=0.0;
%   fgc(j,2)= mgl*pgl+mgr*pgr;
%   fgc(j,3)= mgl*pgl*vgo(j-1)+mgr*pgr*vgo(j);
%
%   fp(j,1)= 0.0;
%   fp(j,2)= 0.0;
%   fp(j,3)= pl*po(j-1)+pr*po(j);

% First order methods ends here

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

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

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

% Second order method starts here:
% Here slopelimiter is used on all variables except phase velocities

psll = po(j-1)+dx/2*s12(j-1);
pslr = po(j)-dx/2*s12(j);
dsll = dlo(j-1)+dx/2*s11(j-1);
dslr = dlo(j)-dx/2*s11(j);
dgl1 = dgo(j-1)+dx/2*s16(j-1);
dglr = dgo(j)-dx/2*s16(j);

vlv = vlo(j-1)+dx/2*s13(j-1);

```

```

vlh = vlo(j)-dx/2*s13(j);
vgv = vgo(j-1)+dx/2*s14(j-1);
vgh = vgo(j)-dx/2*s14(j);

gvv = ego(j-1)+dx/2*s15(j-1);
gvh = ego(j)-dx/2*s15(j);
lvv = 1-gvv;
lvh = 1-gvh;

cl = csound(gvv,psll,dsl1,k);
cr = csound(gvh,pslr,dslr,k);
c = max(cl,cr);

pll = psip(vlo(j-1),c,lvh);
plr = psim(vlo(j),c,lvv);
pgl = psip(vgo(j-1),c,gvh);
pgr = psim(vgo(j),c,gvv);
vmixr = vlo(j)*lvh+vgo(j)*gvh;
vmixl = vlo(j-1)*lvv+vgo(j-1)*gvv;

pl = pp(vmixl,c);
pr = pm(vmixr,c);

mll= lvv*dsl1;
mlr= lvh*dslr;
mgl= gvv*dgl1;
mgr= gvh*dglr;

flc(j,1)= mll*pll+mlr*plr;
flc(j,2)= 0.0;
flc(j,3)= mll*pll*vlo(j-1)+mlr*plr*vlo(j);

fgc(j,1)=0.0;
fgc(j,2)= mgl*pgl+mgr*pgr;
fgc(j,3)= mgl*pgl*vgo(j-1)+mgr*pgr*vgo(j);

fp(j,1)= 0.0;
fp(j,2)= 0.0;
fp(j,3)= pl*psll+pr*pslr;

```

%%% Second order method ends here

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

% Here sloplimiters is used on all variables. This  
 % has not worked so well yet. Therefore it is commented away.

```

%   psll = po(j-1)+dx/2*s12(j-1);
%   pslr = po(j)-dx/2*s12(j);
%   dsll = dlo(j-1)+dx/2*s11(j-1);
%   dslr = dlo(j)-dx/2*s11(j);
%   dgll = dgo(j-1)+dx/2*s16(j-1);
%   dglr = dgo(j)-dx/2*s16(j);
%
%   vlv = vlo(j-1)+dx/2*s13(j-1);
%   vlh = vlo(j)-dx/2*s13(j);
%   vgv = vgo(j-1)+dx/2*s14(j-1);
%   vgh = vgo(j)-dx/2*s14(j);
%
%   gvv = ego(j-1)+dx/2*s15(j-1);
%   gvh = ego(j)-dx/2*s15(j);
%   lvv = 1-gvv;
%   lvh = 1-gvh;
%
%   cl = csound(gvv,psll,dsll,k);
%   cr = csound(gvh,pslr,dslr,k);
%   c = max(cl,cr);
%
%   pll = psip(vlv,c,lvh);
%   plr = psim(vlh,c,lvv);
%   pgl = psip(vgv,c,gvh);
%   pgr = psim(vgh,c,gvv);
%   vmixr = vlh*lvh+vgh*gvh;
%   vmixl = vlv*lvv+vgv*gvv;
%
%   pl = pp(vmixl,c);
%   pr = pm(vmixr,c);
%   mll= lvv*dsll;
%   mlr= lvh*dslr;
%   mgl= gvv*dgll;
%   mgr= gvh*dglr;
%
%   flc(j,1)= mll*pll+mlr*plr;
%   flc(j,2)= 0.0;
%   flc(j,3)= mll*pll*vlv+mlr*plr*vlh;
%
%
%   fgc(j,1)=0.0;
%   fgc(j,2)= mgl*pgl+mgr*pgr;
%   fgc(j,3)= mgl*pgl*vgv+mgr*pgr*vgh;
%
%   fp(j,1)= 0.0;
%   fp(j,2)= 0.0;
  
```

## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

```
% fp(j,3)= pl*psll+pr*pslr;

end

% Fluxes have now been calculated. We will now update the conservative
% variables in each of the numerical cells.

% The source terms can be calculated by using a
% for loop.
% Note that the model is sensitive to how we treat the model
% for low Reynolds numbers (possible discontinuity in the model)
for j=1:nobox
    fricgrad(j)=dpfric(vlo(j),vgo(j),evo(j),ego(j),dlo(j),dgo(j), ...
        po(j),do(j),di(j),viscl,viscg); % Pa/m
    hydgrad(j)=g*(dlo(j)*evo(j)+dgo(j)*ego(j)); % Pa/m
end

sumfric = 0;
sumhyd= 0;

for j=1:nobox

% Here we solve the three conservation laws for each cell and update
% the conservative variables qv

    ar = area(j);

% Liquid mass conservation
    qv(j,1)=qvo(j,1)-dtdx*((ar*flc(j+1,1)-ar*flc(j,1))...
        +(ar*fgc(j+1,1)-ar*fgc(j,1))...
        +(ar*fp(j+1,1)-ar*fp(j,1)));

% Gas mass conservation:

    qv(j,2)=qvo(j,2)-dtdx*((ar*flc(j+1,2)-ar*flc(j,2))...
        +(ar*fgc(j+1,2)-ar*fgc(j,2))...
        +(ar*fp(j+1,2)-ar*fp(j,2)));

% Mixture momentum conservation:

    qv(j,3)=qvo(j,3)-dtdx*((ar*flc(j+1,3)-ar*flc(j,3))...
        +(ar*fgc(j+1,3)-ar*fgc(j,3))...
        +(ar*fp(j+1,3)-ar*fp(j,3)))...
        -dt*ar*(fricgrad(j)+hydgrad(j));

% Add up the hydrostatic pressure and friction in the whole well.
    sumfric=sumfric+fricgrad(j)*dx;
```



```
sumhyd=sumhyd+hydgrad(j)*dx;
```

```
end
```

```
% Section where we find the physical variables (pressures, densities etc)
% from the conservative variables. Some trickes to ensure stability. These
% are induced to avoid negative masses.
```

```
gasmass=0;
liqmass=0;
```

```
for j=1:nobox
```

```
% Remove the area from the conservative variables to find the
% the primitive variables from the conservative ones.
```

```
qv(j,1)= qv(j,1)/area(j);
qv(j,2)= qv(j,2)/area(j);
```

```
if (qv(j,1)<0.00000001) % Trick to avoid negative masses.
    qv(j,1)=0.00000001;
end
```

```
if (qv(j,2)< 0.00000001) % Trick to avoid negative masses.
    qv(j,2)=0.00000001;
end
```

```
% Here we summarize the mass of gas and liquid in the well respectively.
% These variables are important to show that the scheme is conserving
% mass. (if e.g. gas leaks in our out of the well unintentionally in the simulation
% without being specified in the code,something fundamental is wrong.
```

```
gasmass = gasmass+qv(j,2)*area(j)*dx;
liqmass = liqmass+qv(j,1)*area(j)*dx;
```

```
% Below, we find the primitive variables pressure and densities based on
% the conservative variables q1,q2. One can choose between getting them by
% analytical or numerical solution approach specified in the beginning of
% the program. Ps. For more advanced density models, this must be changed.
```

```
if (analytical == 1)
%     % Analytical solution:
```

## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

```
% here the simple density models used in PET 510 Wellflow modelling
% compendium is used.

t1=rho0-P0/al^2;

% Coefficients:
a = 1/(al*al);
b = t1-qv(j,1)-rt*qv(j,2)/(al*al);
c = -1.0*t1*rt*qv(j,2);
%

% Note here we use the very simple models from the PET510 course
p(j)=(-b+sqrt(b*b-4*a*c))/(2*a); % Pressure
dl(j)=rholiq(p(j),temp(j)); % Density of liquid
dg(j)=rogas(p(j),temp(j)); % Density of gas

% The code below can be activated if we want to switch to the other set
% of density models. Also then remember to do the changes inside
% functions rogas og rholiq if we change density models.

% x1=rho0-P0*rho0/Bheta-rho0*Alpha*(temp(j)-T0);
% x2=rho0/Bheta;
% x3=-qv(j,2)*R*temp(j);

% a = x2;
% b = x1+x2*x3-qv(j,1);
% c = x1*x3;

% p(j)=(-b+sqrt(b*b-4*a*c))/(2*a); % Pressure
% dl(j)=rholiq(p(j),temp(j));
% dg(j)=rogas(p(j),temp(j));
else

% Numerical Solution: This might be used if we use more complex
% density models. Has not been used for years.

[p(j),error]=itsolver(po(j),qv(j,1),qv(j,2)); % Pressure
dl(j)=rholiq(p(j),temp(j)); % Density of liquid
dg(j)=rogas(p(j)); % Density of gas

% Incase a numerical solution is not found, the program will write out "error":
if error > 0
    error
end
end

% Find phase volume fractions
eg(j)= qv(j,2)/dg(j);
ev(j)=1-eg(j);
```

```
% Reset average conservative variables in cells with area included in the variables.
```

```
qv(j,1)=qv(j,1)*area(j);
qv(j,2)=qv(j,2)*area(j);
```

```
end % end of loop
```

```
% Below we find the phase velocities by combining the
% conservative variable defined by the mixture momentum equation
% with the gas slip relation.
% At the same time we try to summarize the gas volume in the well. This
% also measure the size of the kick.
```

```
gasvol=0;
```

```
for j=1:nobox
```

```
% The interpolations introduced below are included
% to omit a singularity in the slip relation when the gas volume
% fraction becomes equal to 1/K. In addition, S is interpolated to
% zero when approaching one phase gas flow. In the transition to
% one phase gas flow, we have no slip conditions (K=1, S=0)
% We will let the k0,s0,k1,s1 be arrays to make it easier to incorporate
% different flow regimes later. In that case, the slip parameters will
% vary from cell to cell and we must have slip parameter values for each
% cell.
```

```
ktemp=k;
stemp=s;
```

```
k0(j) = ktemp;
s0(j) = stemp;
```

```
% Interpolation to handle that (1-Kxgasvolumefraction) does not become zero
```

```
if ((eg(j)>=0.7) & (eg(j)<=0.8))
    xint = (eg(j)-0.7)/0.1;
    k0(j) = 1.0*xint+k*(1-xint);
elseif(eg(j)>0.8)
    k0(j)=1.0;
end
```

```

% Interpolate S to zero in transition to pure gas phase
if ((eg(j)>=0.9) & (eg(j)<=1.0))
    xint = (eg(j)-0.9)/0.1;
    s0(j) = 0.0*xint+s*(1-xint);
end

% Note that the interpolations above and below can be changed
% if numerical stability problems
% are encountered.

%
if (eg(j)>=0.999999)
    % Pure gas
    k1(j) = 1.0;
    s1(j) = 0.0;
else
    % Two phase flow
    k1(j) = (1-k0(j)*eg(j))/(1-eg(j));
    s1(j) = -1.0*s0(j)*eg(j)/(1-eg(j));
end

help1 = dl(j)*ev(j)*k1+dg(j)*eg(j)*k0;
help2 = dl(j)*ev(j)*s1+dg(j)*eg(j)*s0;

vmixhelp1 = (qv(j,3)/area(j)-help2)/help1;
vg(j)=k0(j)*vmixhelp1+s0(j);
vl(j)=k1(j)*vmixhelp1+s1(j);

% Variable for summarizing the gas volume content in the well.
gasvol=gasvol+eg(j)*area(j)*dx;

end

% Old values are now set equal to new values in order to prepare
% computation of next time level.

po=p;
dlo=dl;
dgo=dg;
vlo=vl;
vgo=vg;

```

```
ego=eg;
evo=ev;
qvo=qv;
```

```
% Section where we save some timedependent variables in arrays.
% e.g. the bottomhole pressure. They will be saved for certain
% timeintervalls defined in the start of the program in order to ensure
% that the arrays do not get to long!
```

```
if (counter>=nostepsbeforesavingtimedata)
    printcounter=printcounter+1;
    time % Write time to screen.
```

```
% Outlet massrates (kg/s) vs time
```

```
liquidmassrateout(printcounter)=dl(nobox)*ev(nobox)*vl(nobox)*area(nobox);
gasmassrateout(printcounter)=dg(nobox)*eg(nobox)*vg(nobox)*area(nobox);
```

```
% Outlet flowrates (lpm) vs time
```

```
liquidflowrateout(printcounter)=liquidmassrateout(printcounter)/...
    rho(liq(P0,T0))*1000*60;
gasflowrateout(printcounter)=gasmassrateout(printcounter)/...
    rho(gas(P0,T0))*1000*60;
```

```
% Hydrostatic and friction pressure (bar) in well vs time
```

```
hyd(printcounter)=sumhyd/100000;
fric(printcounter)=sumfric/100000;
```

```
% Volume of gas in well vs time (m3). Also used for indicating kick
% size in well.
```

```
volgas(printcounter)=gasvol;
```

```
% Total phase masses (kg) in the well vs time
```

```
% Used for checking mass conservation.
```

```
massgas(printcounter)=gasmass;
massliq(printcounter)=liqmass;
```

```
% pout calculates the pressure at the outletboundary. I.e. upper edge
% of uppermost cell. Corresponds where the well ends at surface. The
% reason we do this is the fact than in AUSMV is all variables defined
% in the mid point of the numerical cells.
```

```
pout(printcounter)=(p(nobox)-0.5*dx*...
(dlo(nobox)*evo(nobox)+dgo(nobox)*ego(nobox))*g-dx*0.5*fricgrad(nobox))/100000;
```

```
% pin (bar) defines the pressure at the inlet boundary, I.e lower edge
```

## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

```
% of the lowermost cell. Corresponds to TD of well.
pin(printcounter)=
(p(1)+0.5*dx*(dlo(1)*evo(1)+dgo(1)*ego(1))*g+0.5*dx*fricgrad(1))/100000;

% Pressure in the middle of top box (bar).
pnobox(printcounter)=p(nobox)/100000; %

% Store the control parameter, choke pressure that has to be applied.
% Divide by 100000 to get it in bar.
pchoke(printcounter)=pressureoutlet/100000;

%   chokeopening(printcounter)= zopning;

% Time variable
timeplot(printcounter)=time;

counter = 0;

end
end

% end of stepping forward in time.

% Printing of resultssection

countsteps % Marks number of simulation steps.

% Plot commands for variables vs time. The commands can also
% be copied to command screen where program is run for plotting other
% variables.

toc,
e = cputime-t

% Plot bottomhole pressure
plot(timeplot,pin)

% Show cfl number used.
disp('cfl')
cfl = al*dt/dx

plot(timeplot,pin)
%plot(timeplot,hyd)
%plot(timeplot,fric)
```

## PI Regulation in CBHP-MPD Simulation using a Transient Flow Model

```
%plot(timeplot,liquidmassrateout)
%plot(timeplot,gasmassrateout)
%plot(timeplot,volgas)
%plot(timeplot,liquidflowrateout)
%plot(timeplot,gasflowrateout)
%plot(timeplot,massgas)
%plot(timeplot,massliq)
%plot(timeplot,pout)
%plot(timeplot,pnobox)
%plot(timeplot,pchoke)
%plot(timeplot,chokeopening)

%Plot commands for variables vs depth/Only the last simulated
% values at endtime is visualised

%plot(vl,x);
%plot(vg,x);
%plot(eg,x);
%plot(p,x);
%plot(dl,x);
%plot(dg,x);
```