
Compact Sub Sea Separation:
Implementation and comparison of two different control structures

by Christian Ellingsen

Supervisor: Prof. Sigurd Skogestad
Co-supervisor: Gisle Otto Eikrem, StatoilHydro

Department of Chemical Engineering
Norwegian University of Science and Technology
2007

Preface

This report is the product of many hours of hard work. Sometimes it has been frustrating, e.g. when the computer or the OPC-server doesn't want to co-operate. But everything turned out just perfect at the end, and I must say that it has been very interesting and a good learning experience.

The purpose of the project has changed a little from the start, and ended up with a comparison of two different control structures on a Matlab model of a compact sub sea separation module. The first control structure is based on PI-controllers, while the second structure is based on a MPC-controller.

I would like to thank my supervisor Professor Sigurd Skogestad and my co-supervisor Gisle Otto Eikrem for all their help through the project. I also have to thank almost the entire Control Group at StatoilHydro, Trondheim for their help with SEPTIC and the OPC-server.

Christian Ellingsen, Trondheim 25.11.2007

Abstract

The oil and gas industry is continuously developing new technology. This is crucial to reach the more and more difficult oil and gas reserves. In connection to this, StatoilHydro is doing research on new technology for sub sea separation of liquid and gas. This technology is based on in-line separation units, and is thought to be as compact as possible. The challenge of this compact system is that it needs fast and robust control.

Two different control applications have been tested out on a pre-developed Matlab model of the system. The first control structure is based on PI-controllers, while the second application is based on a MPC-controller. The PI-controllers were tuned using the SIMC tuning rules (Skogestad 2002). For the second application, the MPC-tool SEPTIC was used. This is a MPC-controller which StatoilHydro has developed and are using in several of their own applications.

The simulations showed a clear difference between the two applications. The MPC-controller performed much better in most of the cases. It gave smoother use of the manipulated variables, and at the same time faster and more robust control of the controlled variables. There was also possible to lay closer to the constraints when using the MPC-controller. The PI-controllers gave some oscillations in some of the cases. This was caused by different gain depending of which direction the system was going in. A solution to this could be to implement a split range controller for the PI-application. In addition there is possible to use cascade control to break some of the most coupled PI-controller loops.

Multivariable control as for example the MPC-controller should be considered for this system. The process is very interactive and has several constraints which should be held. This is two problems which a MPC-controller is designed to handle. There should be thought about implement PI-controllers at the bottom layer just to get a stable system. This would also secure the system if the MPC-controller on top crashes.

Contents

1	INTRODUCTION.....	1
1.1	THE MOTIVATION FOR COMPACT SUB SEA SEPARATION.....	1
1.2	A BRIEF DESCRIPTION OF THE SEPARATOR UNIT STUDIED.....	1
1.3	THE MAIN TASKS FOR THE CONTROLLER SYSTEM.....	2
1.4	THE PURPOSE OF THE PROJECT.....	3
2	THE MATHEMATICAL MODEL.....	4
2.1	THE ORIGIN OF THE MODEL AND ITS WEAKNESSES.....	4
2.2	MODEL IMPROVEMENTS PERFORMED.....	4
2.2.1	<i>Introduction of degasser dynamics.....</i>	<i>4</i>
2.2.2	<i>Implementation of the return streams.....</i>	<i>4</i>
2.2.3	<i>Introduction of pump and compressor dynamics.....</i>	<i>5</i>
3	BASIC CONTROLLER THEORY.....	6
3.1	A REVIEW ON THE SIMC-TUNING RULES.....	6
3.1.1	<i>SIMC-tuning of a first-order process.....</i>	<i>6</i>
3.1.2	<i>SIMC-tuning of an integrating process.....</i>	<i>7</i>
3.2	BASIC THEORY ON MPC.....	8
4	IMPLEMENTATION AND TUNING OF THE CONTROLLERS.....	10
4.1	CHOICE OF INPUT/OUTPUT PAIRINGS.....	10
4.2	TUNING OF THE PI-CONTROLLERS.....	10
4.2.1	<i>Tuning of control loop 1.....</i>	<i>10</i>
4.2.2	<i>Tuning of control loop 2.....</i>	<i>11</i>
4.2.3	<i>Tuning of control loop 3.....</i>	<i>12</i>
4.2.4	<i>Tuning of control loop 4.....</i>	<i>13</i>
4.2.5	<i>Tuning of control loop 5.....</i>	<i>15</i>
4.2.6	<i>Tuning of control loop 6.....</i>	<i>16</i>
4.3	IMPLEMENTATION OF THE MPC.....	17
4.3.1	<i>The control structure.....</i>	<i>17</i>
4.3.2	<i>The MPC software, SEPTIC.....</i>	<i>17</i>
4.3.3	<i>An example on the step-response models.....</i>	<i>18</i>
4.4	TUNING OF THE MPC-CONTROLLER.....	18
4.4.1	<i>Choice of priorities.....</i>	<i>19</i>
4.4.2	<i>Choice of tuning parameters.....</i>	<i>19</i>
5	SIMULATION RESULTS.....	20
5.1	PRESSURE IN SEPARATOR TANK.....	20
5.2	LIQUID LEVEL IN SEPARATOR TANK.....	21
5.3	LIQUID LEVEL IN THE BOOT SECTION OF THE DELIQUIDIZER.....	22
5.4	SET-POINT CHANGE LEVEL, SEPARATOR TANK.....	22
5.5	REDUCED INFLOW TO THE SYSTEM.....	24
6	DISCUSSION.....	25
6.1	PRESSURE IN SEPARATOR TANK.....	25
6.2	LIQUID LEVEL IN SEPARATOR TANK.....	25
6.3	LIQUID LEVEL IN THE BOOT SECTION OF THE DELIQUIDIZER.....	26

6.4	SET-POINT CHANGE LEVEL, SEPARATOR TANK.....	26
6.5	REDUCED INFLOW TO THE SYSTEM	26
6.6	THE USE OF MANIPULATED VARIABLES	27
7	CONCLUSION.....	28
	SUGGESTIONS TO FURTHER WORK.....	28
	REFERENCES	30
	APPENDIX A – THE MATHEMATICAL MODEL	I
A.1	THE SEPARATOR TANK	I
A.2	THE DEGASSER	II
A.3	THE DELIQUIDIZER	V
A.4	THE RETURN STREAMS	VIII
A.5	COMMENTS TO THE EQUATIONS.....	VIII
	APPENDIX B – THE DIFFERENT CASES	IX
	APPENDIX C – THE MAIN MATLAB SCRIPT.....	X

1 Introduction

1.1 *The motivation for compact sub sea separation*

The oil and gas industry has to continuously develop new technology to reach the enormous oil and gas resources which are inaccessible today. This oil and gas could be difficult to reach because of for example low reservoir pressure or large water depths. There are also a lot of small reservoirs which are too expensive to develop with the technology of today.

There are different solutions to each of the mentioned problems, but a compact separation unit could be a part of the solution for all of them. In the case of low reservoir pressure, there would be desirable to lower the pressure at the well heads to get more of the present resources out of the reservoir. Then the pressure has to be increased again for the oil and gas to be transported to a platform or to shore. This boosting could be done by a two phase pump, and this is a solution which is common today. The draw back of this solution is that the two phase pump has low efficiency and is hard to maintain. It would therefore be better to separate the gas and the liquid sub sea, and then use a conventional pump and compressor for boosting. Hopefully, in a longer prospect, the same technology could be used for sub sea separation of unnecessary components as for example water, and then inject it back to an empty reservoir.

The size of the separation unit would of course be important for all sub sea applications, but this is especially important when the system is going to be installed at deep water. A compact unit would be easier to install and easier to maintain. The draw back would be that the regularity of the system would be smaller than for a larger system.

1.2 *A brief description of the separator unit studied*

As shown in Figure 1.1, the separator unit studied in this work actually consists of five different units. The main units for making this system compact are the degasser and the deliquidizer. These units are based on new in-line technology where a centrifugal force is set up to separate components with different density. The reason why this is called “in-line technology” is because the units could be designed to have almost the same dimensions as the transport pipe. The centrifugal force needed to separate the phases is made by the flow itself. If the flow rate gets too small, there would not be enough power to run the swirl element inside the degasser or the deliquidizer.

The small separator tank would do most of the separation, but the separation would not be satisfactory regards to the demands of the compressor and the pump. The degasser and the deliquidizer would therefore do further separation of the two phases. The quality of the liquid coming out of the degasser is made satisfactory through the configurations of the degasser. This means that the quality of the liquid phase coming out of the degasser is more important than the quality of the gas phase separated. The gas phase separated in the degasser must therefore pass the deliquidizer to get separated further.

A more complicated method is used to achieve a dry gas into the compressor, and at the same time not letting too much gas going to the pump. Inside the deliquidizer there is a center pipe

which is designed to take most of the gas phase and nothing of the liquid phase. The liquid phase and the rest of the gas phase hit the back wall of the deliquidizer, and then enter the boot section. The boot section works as a small flash tank, and the gas phase gets sucked back into the pipe section of the deliquidizer through the gas recycle line. This mechanism is driven by the fact that the flow through the deliquidizer puts up a pressure difference between the boot section and the pipe section of the deliquidizer.

The return streams of liquid and gas after the pump and compressor respectively, are not going to be used under normal operation. They are introduced to increase the regularity of the total separator unit, and would be used to secure that the pump and the compressor have enough feed or that the flow rate through the degasser and deliquidizer is above a certain limit.

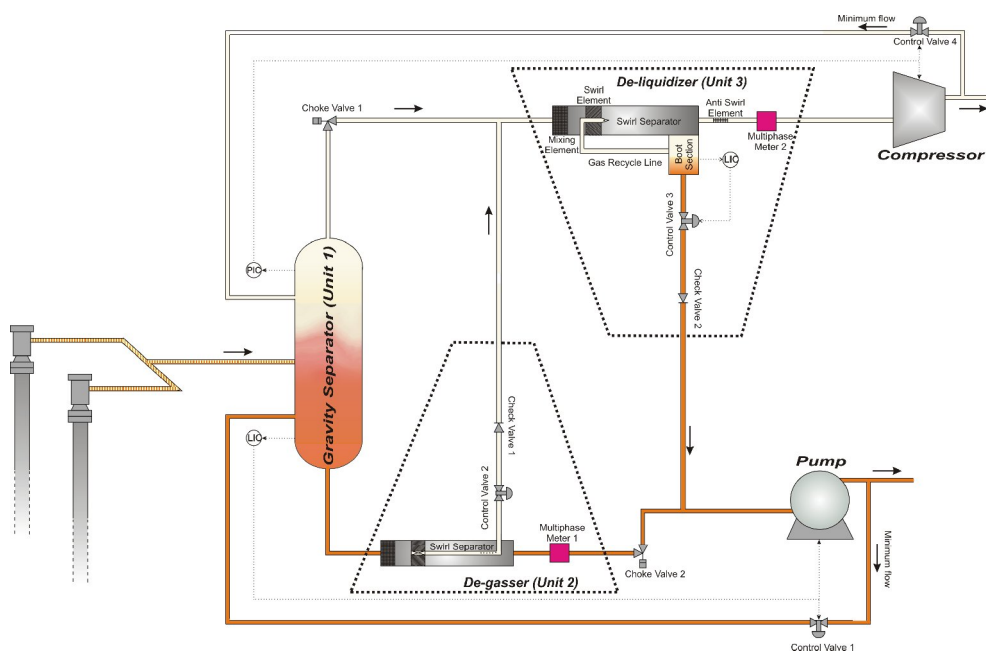


Figure 1.1: A schematic view of the compact separator unit

1.3 The main tasks for the controller system

There are four valves used for controlling purposes, together with the pump and the compressor. When controlling this system, some variables are more important than others. Because the system is made as small as possible, the regularity is an important issue. Some criterions have to be met to avoid a trip of the system.

The most obvious criterions are that the liquid levels in the separator tank and the boot section of the deliquidizer are held between full and empty. Another important criterion is to stay above the surge area of the compressor and pump. Then the controller system has to make sure that the compressor and pump have enough feed. There is also important that the flow rate into the degasser and deliquidizer is above a certain limit, and this criterion would probably be in evidence before the feed to the pump and the compressor reaches the low limit.

To measure the flows, flow meters have to be installed in front of the degasser and the deliquidizer. These devices are not given in Figure 1.1.

There is also important to secure that the gas going to the compressor is dry enough, and that the liquid going to the pump does not contain more gas than the requirements of the pump. This is done by making sure that the liquid outlet of the degasser and the gas outlet of the deliquidizer are the cleanest streams. To achieve a dry gas to the compressor, it is important to have a good level control of the boot section of the deliquidizer, and to make sure that not too much liquid enters the deliquidizer. To avoid that too much gas is going to the pump, it is important to keep the liquid content in the gas outlet of the degasser high enough, which implies that the gas content is low in the liquid outlet of the degasser. To achieve this, there has to be a multiphase meter at the gas outlet of the degasser. This device is not given in Figure 1.1.

1.4 The purpose of the project

The main task of this work was to implement control on an already developed Matlab model of the compact separator unit. Two different types of control systems were going to be developed independently of each other, and then compared. The first control system implemented was based on decoupled PI control loops, while the second control system was based on model predictive control (MPC). MPC is a multivariable control system, and decoupling of the system is not necessary. A lot of simulation studies have been done, and the performances of the two controllers have been compared. There were also supposed to be some simulations on start-up and shut down of the system, but the time has shown not to be adequate for this.

Together with the implementation of a MPC controller on the system, there also has been done a small literature study on MPC control.

2 The mathematical model

2.1 *The origin of the model and its weaknesses*

The model was developed in the summer of 2007 by a group of three persons, the author of this report as one of the participants. Some of the physical equations were given by StatoilHydro at the beginning of the project, but still a lot had to be done by the project group. It was decided that the model were going to be implemented in Matlab, and after 8 weeks of hard work, the first outline of a complete model was finished. There was no real separator unit to take any physical values from, and the parameters are just fitted to a thought lab scale unit. Unfortunately the first model showed to be a little bit too simple due to lack of some important dynamic behaviors. Because of this, some model improvements had to be included in this project work.

2.2 *Model improvements performed*

The reader is encouraged to use Figure 1.1 for help in this section. This figure gives a good overview of the complete system.

2.2.1 **Introduction of degasser dynamics**

The degasser is the unit which has the simplest physical model in the complete sub sea separator model. It is simply a set of algebraic equations which returns a split factor of the inlet stream, based on the flow capacity of the gas outlet of the degasser. More information on this is given in Appendix A, where the complete mathematical model is given.

The absence of dynamics in this unit gave some problems when implementing control on the system. There was therefore decided to indirectly introduce dynamics to this unit by implement a state equation to the valve at the gas outlet of the degasser.

2.2.2 **Implementation of the return streams**

A major weakness of the original model was that the return streams were not included. This had of course to be solved before further work could be done. Especially because the return streams are important for controlling the system.

A small problem arises when the return streams are to be included. They are the last calculated variables, but are still needed to give the inflow to the separator tank. This set of algebraic equations could have been solved simultaneously to give a solution, but the computation time turned out to be very long. An easy solution to this problem was to introduce fast dynamics of the return streams. Then the inflow into the separator tank could be based on past values of the return streams. This solution has shown to be satisfactory.

2.2.3 Introduction of pump and compressor dynamics

To achieve a more realistic model, there have also been introduced state equations for the compressor and pump power. There is not realistic that a pump or a compressor reacts immediately when it is tolled to change power.

3 Basic controller theory

3.1 A review on the SIMC-tuning rules

This chapter is just going to give a short description of the SIMC-tuning rules which have been used in this work. The information is taken from Skogestad (2003), and the reader is requested to look in this paper for a deeper understanding of the theory behind the rules.

3.1.1 SIMC-tuning of a first-order process

The transfer equation for a general first-order process is given in equation (3.1)

$$g(s) = k \frac{e^{-\theta s}}{(\tau_1 s + 1)} \quad (3.1)$$

Figure 2.1 shows a first order response on the output after a step change on the input. From this response, the three different parameters which describe a first order process could be found. These parameters are the gain (k), the dead time (θ) and the time constant (τ_1). The gain is the change on the output after a new steady state is reached, relative to the change performed on the input. The time from the input change is done until there is happening anything at the output, is called the dead time. The time constant is the time from there is seen a change on the output until approximately 63% of the output change is done.

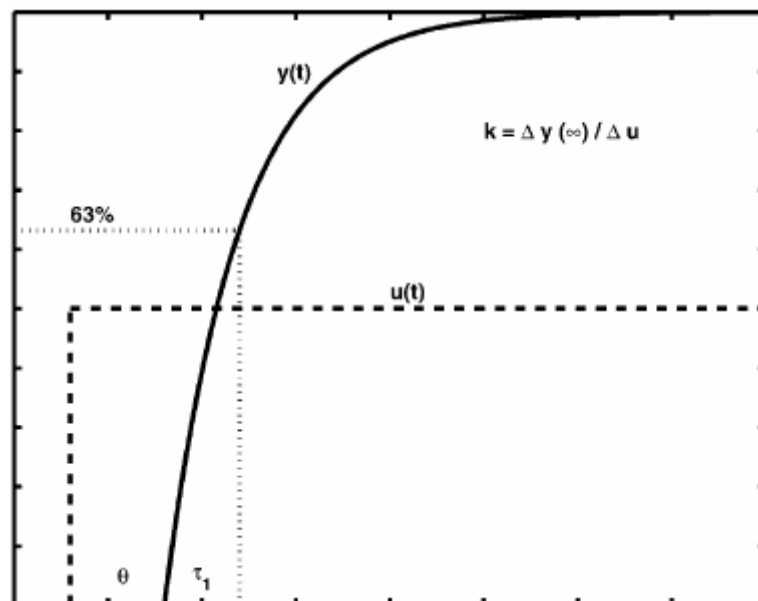


Figure 2.1: A first order response after a step change on the input, Skogestad (2003)

After the three parameters for the first order process are found, the PI-controller can be tuned by using the following equations:

$$K_c = \frac{1}{k} \frac{\tau_1}{\tau_c + \theta} \quad (3.2)$$

$$\tau_I = \min \{ \tau_1, 4(\tau_c + \theta) \} \quad (3.3)$$

There is only one parameter left for the user to decide, and that is the wanted closed loop response τ_c . In Skogestad (2003) it is stated that a value of $\tau_c = \theta$ would give a fast response with good robustness.

3.1.2 SIMC-tuning of an integrating process

The transfer equation for a general integrating process is given in equation (3.4).

$$g(s) = k' \frac{e^{-\theta s}}{s} \quad (3.4)$$

Figure 2.2 shows the response for an integrating process after a step change on the input. There are two parameters which have to be identified in equation (3.4), and that are the slope of the integrator (k') and the dead time of the process.

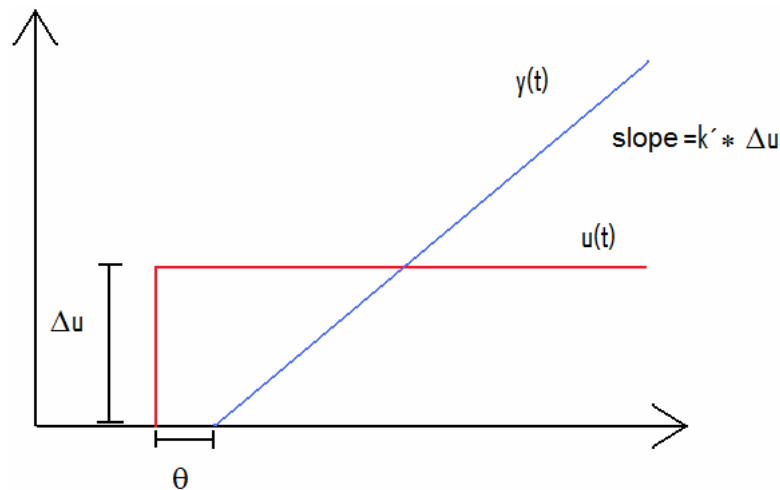


Figure 2.2: An integrating response after a step change on the input

When the two parameters are identified, the PI-controller can be tuned by using the following equations:

$$K_c = \frac{1}{k'} \frac{1}{\tau_c + \theta} \quad (3.5)$$

$$\tau_I = 4(\tau_c + \theta) \quad (3.6)$$

Again there is only one parameter left for the user to decide, and that is the τ_c .

3.2 Basic theory on MPC

The basic theory on MPC given in this chapter is taken from Maciejowski (2002).

The basic idea of MPC is illustrated in Figure 2.3. There is assumed discrete time setting, and the current time is labelled as time step k . The set-point trajectory is given as $S(t)$, and shows which trajectory the output should follow, ideally. For the output to reach the set-point trajectory, a reference trajectory $r(t|k)$ is defined. This reference trajectory gives the closed-loop behaviour of the plant, and is normally based on the assumption that the reference trajectory approaches the set-point exponentially. Then, if the output follows the reference trajectory exactly in the future, the error i steps later would be,

$$\varepsilon(k+i) = e^{-iT_s/T_{ref}} \varepsilon(k) \quad (3.7)$$

where T_s is the sampling time, T_{ref} is the closed-loop time constant and $\varepsilon(k)$ is the current error. The reference trajectory could then be calculated as

$$r(k+i|k) = s(k+i) - \varepsilon(k+i) \quad (3.8)$$

$$= s(k+1) - e^{-iT_s/T_{ref}} \varepsilon(k) \quad (3.9)$$

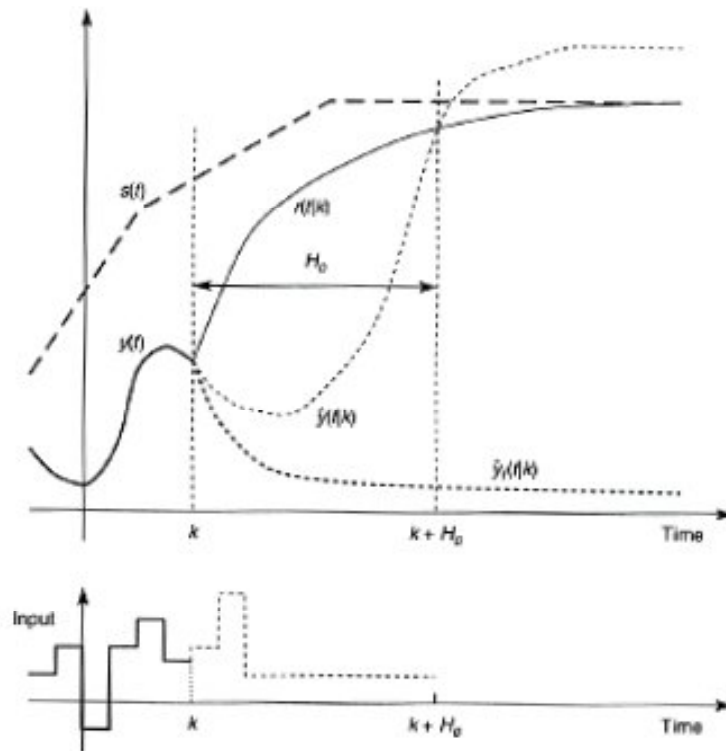


Figure 2.3: The basic idea of MPC, Maciejowski (2002)

The MPC-controller has an internal model of the plant which is used to predict the behaviour of the plant. The goal of the controller is to reach the reference trajectory at some decided coincidence points. In the simplest case, which is illustrated in Figure 2.3, there is only one coincidence point, namely at the end of the prediction horizon, time $k + H_p$. Then the controller would calculate the best possible input trajectory $\hat{u}(k+i|k)$ ($i = 0, 1, \dots, H_p - 1$) to reach this coincidence point. The word “best” in this case is a little bit floating, because the cost function which is going to be minimized could be different from application to application.

Nevertheless, the most common approach is to minimize a cost function based on the outputs deviation from the reference trajectory, together with a term which penalizes movements of the inputs. An example on a cost function which has these two elements is given in equation (3.10)

$$V(k) = \sum_{i=H_w}^{H_p} \|\hat{y}(k+i|k) - r(k+i|k)\|_{Q(i)}^2 + \sum_{i=0}^{H_u-1} \|\Delta\hat{u}(k+i|k)\|_{R(i)}^2 \quad (3.10)$$

There are several points which should be noticed in this equation. It is not normal to penalize deviation from the reference trajectory immediately. This is because there would take some time between applying an input and seeing any effect.

Another important point is to understand that the notation $\|x\|_Q^2$ refers to the quadratic form $x^T Q x$. Here x is a vector, and Q is a symmetric matrix which normally also is diagonal. If Q is a diagonal matrix, the elements would tell how much a deviation from the reference trajectory are penalized for each of the different outputs. The elements in the diagonal matrix R give the movement penalties for each of the inputs.

The cost function could consist of even more terms, and sometimes there is desirable to have set-points also for some of the inputs. Then the cost function would have an additional term which penalizes deviation from the set-points of the inputs. There is also possible to have some soft constraints which could be violated in some cases. Then a term with these constraints also has to be included in the cost function. The appurtenant matrix would tell how important it would be to keep inside the limits of each of the soft constraints.

In addition to the cost function, the problem could consist of some hard constraints. This could for example be high and low limits on the inputs. A valve could not be more than fully open or completely closed. These hard constraints would simply give the problem a limit inside where there is possible to find a solution.

4 Implementation and tuning of the controllers

4.1 Choice of input/output pairings

The pairings for the PI-controller loops are given in Table 4.1. The referred control valves are given in Figure 1.1. Some of the pairings are more obvious than others, for example that control valve 3 is used for controlling the liquid level in the boot section of the deliquidizer. There is also obvious that control valve 2 is used to control the liquid content out of the gas outlet of the degasser. To control the liquid level in the separator tank, two choices seems to be good. Either the return stream of liquid or the pump power could be used. The return stream should only be used when it is necessary to keep the system running. Then the choice is to use the pump power for level control in the separator tank, and the return stream is used to secure the flow rate into the degasser. The same argumentation is used when choosing the pairings for the separator pressure and the flow rate into the deliquidizer.

Table 4.1: Pairings for the PI-controller loops

Control loop:	Input:	Output:
1	Power pump	Liquid level in separator tank
2	Control valve 3	Liquid level in boot section, deliquidizer
3	Power compressor	Pressure in separator tank
4	Control valve 1	Flow rate into degasser
5	Control valve 4	Flow rate into deliquidizer
6	Control valve 2	Liquid content in gas outlet degasser

4.2 Tuning of the PI-controllers

A graphical approach has been used to approximate a first order or integrated response to each of the loops. Then the controllers have been tuned based on the rules given in chapter 3.

4.2.1 Tuning of control loop 1

The integrated response of control loop 1, after a step in u of magnitude 0.1, is given in Figure 4.1. The two parameters θ and the slope were found through graphical analyses. The values were:

$$\theta \approx 1.6 \quad (4.1)$$

$$\text{slope} \approx -0.006 \quad (4.2)$$

It is not so important to have fast set-point tracking of the liquid level in the separator tank. The value for τ_c is therefore set to 3. After inserting this into equation (3.5) and (3.6), these tuning parameters were found:

$$K_c = -72.46 \quad (4.3)$$

$$\tau_I = 18.4 \quad (4.4)$$

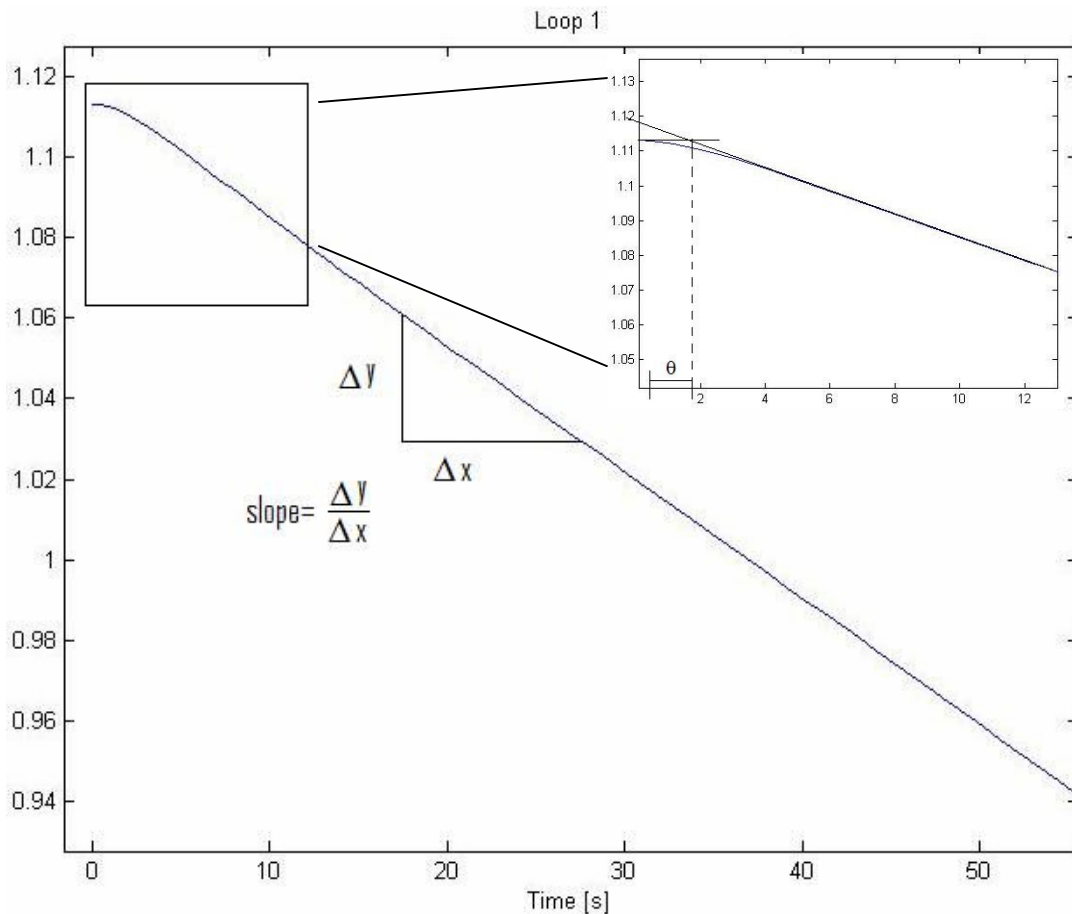


Figure 4.1: Step response on control loop 1

4.2.2 Tuning of control loop 2

Figure 4.2 shows the step response of control loop 2. The different parameters are given under.

$$K = \frac{\Delta y}{\Delta u} = \frac{0.05123}{0.0422} = -12.1392 \quad (4.5)$$

$$\tau_1 \approx 25 \quad (4.6)$$

$$\theta \approx 0 \quad (4.7)$$

The response has no dead time, and it is not possible to choose $\tau_c = \theta$. Because of the small volume of the boot, it is important to have a good set-point tracking of the level. $\tau_c = 2$ has shown to be efficient for this purpose. When using equation (3.2) and (3.3), we got these control parameters:

$$K_c = -1.030 \quad (4.8)$$

$$\tau_1 = 8 \quad (4.9)$$

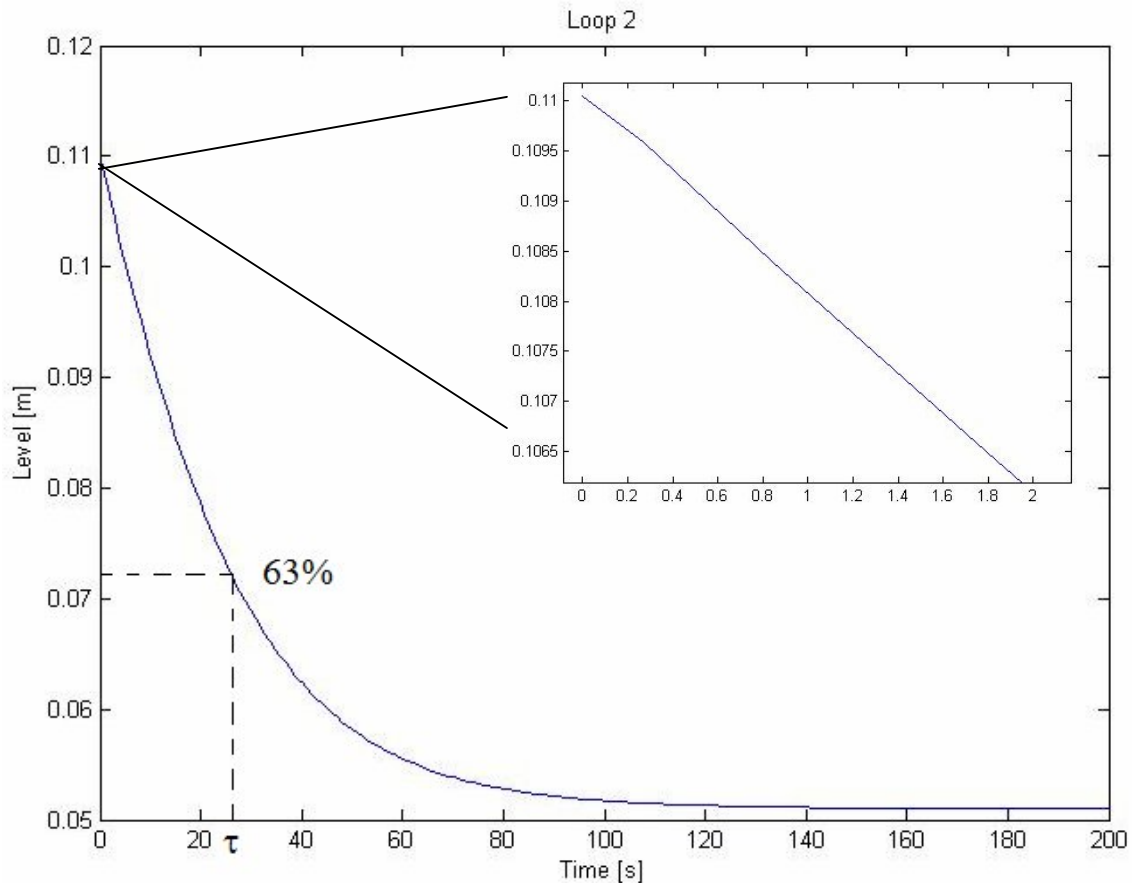


Figure 4.2: Step response on control loop 2

4.2.3 Tuning of control loop 3

Figure 4.3 shows the step response of control loop 3. The different parameters are given under.

$$K = \frac{\Delta y}{\Delta u} = \frac{-0.5926}{2.000} = -0.2962 \quad (4.10)$$

$$\tau_1 \approx 10.6 \quad (4.11)$$

$$\theta \approx 1.1 \quad (4.12)$$

It is desirable to have fast control of the pressure in the separator tank, and $\tau_c = \theta$ was chosen. The calculated control parameters are:

$$K_c = -20.71 \quad (4.13)$$

$$\tau_I = 8.8 \quad (4.14)$$

Unfortunately these tuning parameters gave a lot of oscillations for some of the cases studied. A discussion on this matter would be given in Chapter 6. The oscillations disappeared when setting $K_c = \frac{-20.71}{6}$ and $\tau_I = 8.8 \cdot 2$.

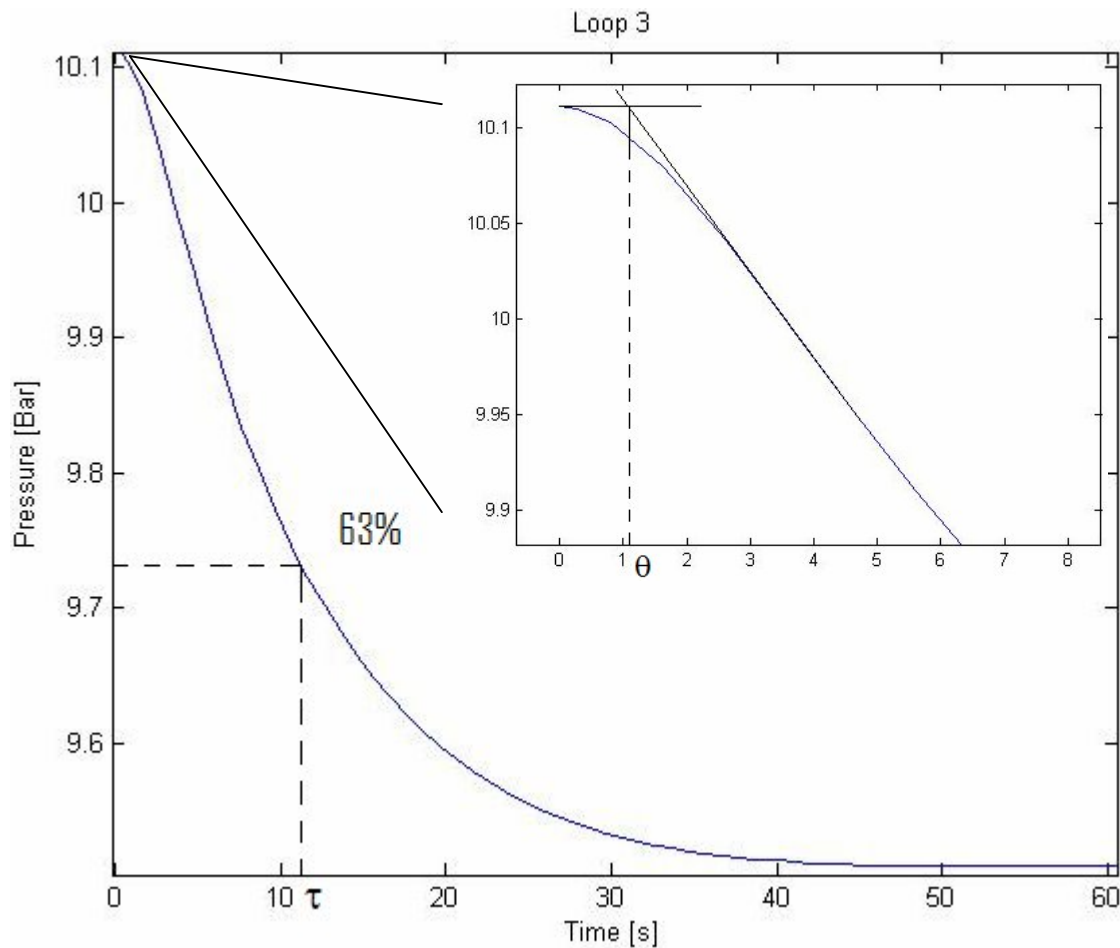


Figure 4.3: Step response on control loop 3

4.2.4 Tuning of control loop 4

Figure 4.4 shows the step response of control loop 4. The different parameters are given under.

$$K = \frac{\Delta y}{\Delta u} = \frac{0.0016}{0.1} = 0.0159 \quad (4.15)$$

$$\tau_1 \approx 4.5 \quad (4.16)$$

$$\theta \approx 0.8 \quad (4.17)$$

It is important to have fast control of the flow rate into the degasser, so $\tau_c = \theta$ was chosen.

The calculated control parameters are:

$$K_c = 176.9 \quad (4.18)$$

$$\tau_I = 4.5 \quad (4.19)$$

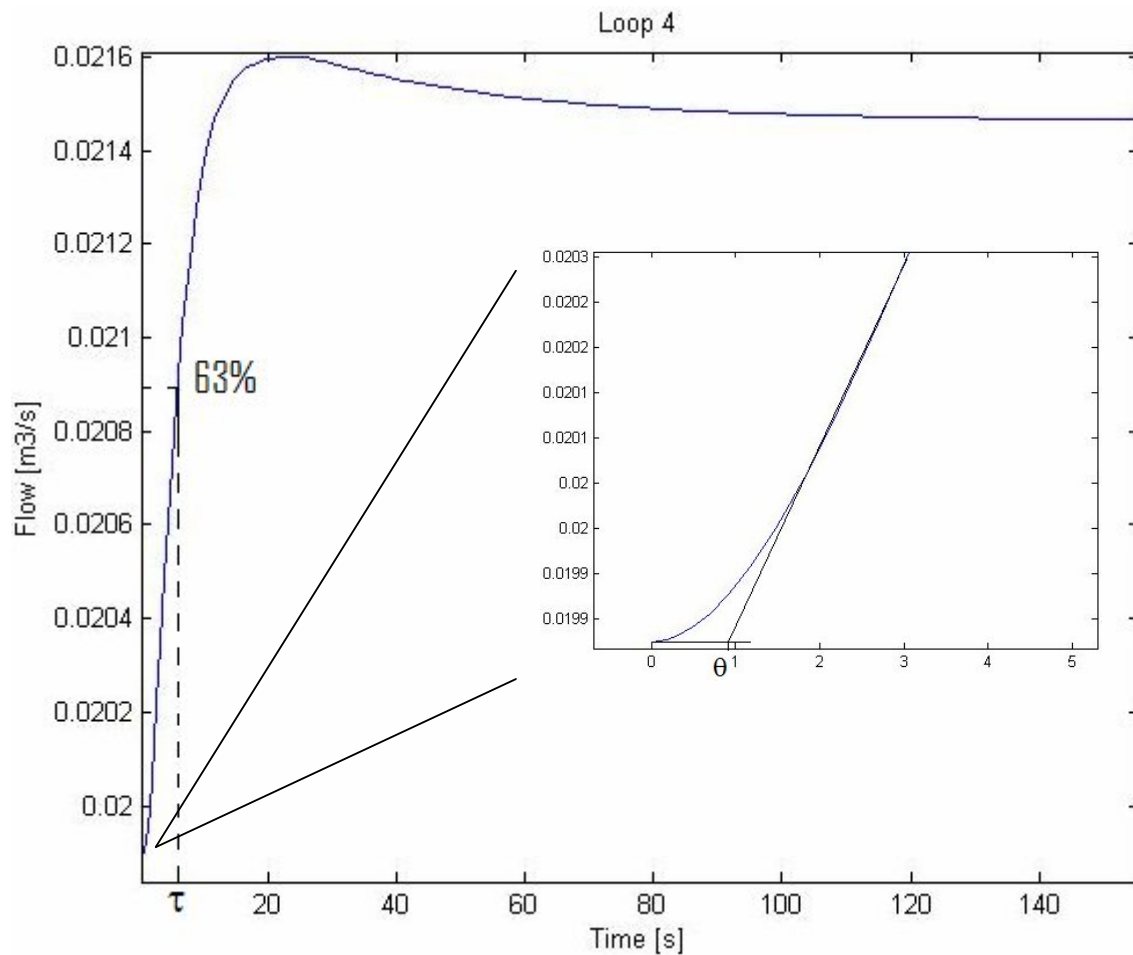


Figure 4.4: Step response on control loop 4

4.2.5 Tuning of control loop 5

Figure 4.5 shows the step response of control loop 5. The different parameters are given under.

$$K = \frac{\Delta y}{\Delta u} = \frac{0.0024}{0.01} = 0.0243 \quad (4.20)$$

$$\tau_1 \approx 10 \quad (4.21)$$

$$\theta \approx 0.9 \quad (4.22)$$

It is important to have fast control of the flow rate into the deliquidizer, so $\tau_c = \theta$ was chosen. The calculated control parameters are:

$$K_c = 228 \quad (4.23)$$

$$\tau_i = 7.2 \quad (4.24)$$

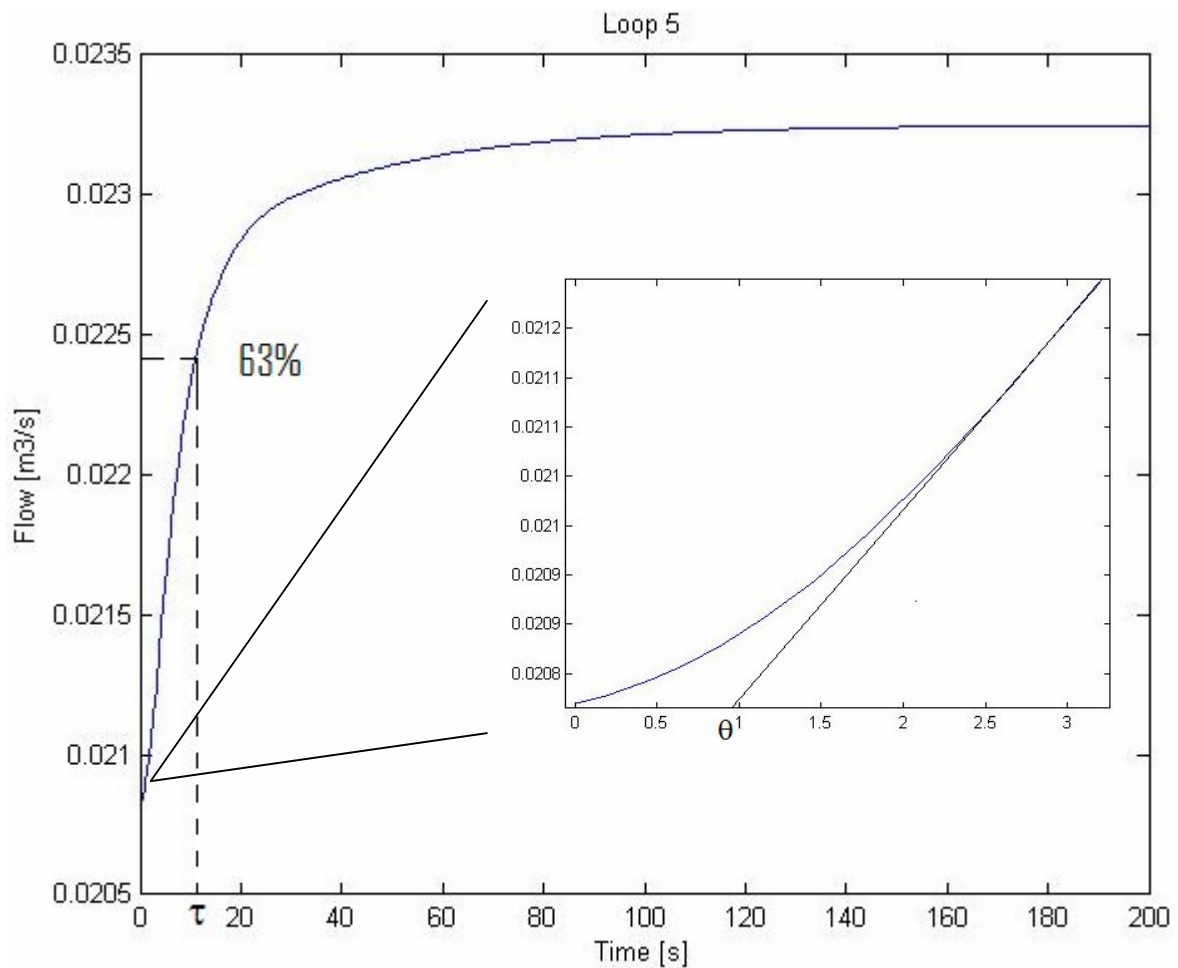


Figure 4.5: Step response on control loop 5

4.2.6 Tuning of control loop 6

Figure 4.6 shows the step response of control loop 6. The different parameters are given under.

$$K = \frac{\Delta y}{\Delta u} = \frac{-0.0057}{0.07} = -0.0813 \quad (4.25)$$

$$\tau_1 \approx 3.2 \quad (4.26)$$

$$\theta \approx 0 \quad (4.27)$$

The response has no dead time, and it is impossible to choose $\tau_c = \theta$. It is important that there is good set-point tracking of the gas quality, and $\tau_c = 4$ has shown to be efficient for this purpose. The calculated control parameters are:

$$K_c = 9.84 \quad (4.28)$$

$$\tau_I = 3.2 \quad (4.29)$$

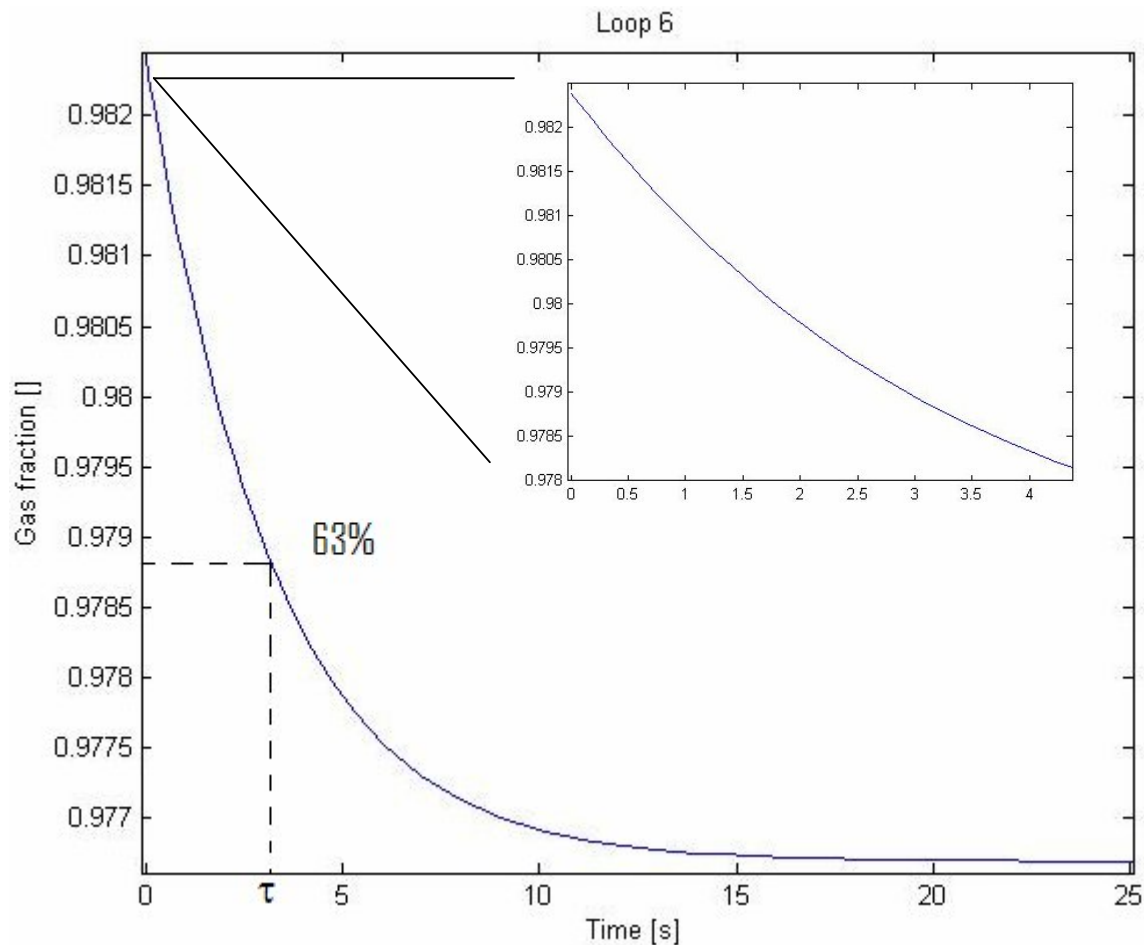


Figure 4.6: Step response on control loop 6

4.3 Implementation of the MPC-controller

4.3.1 The control structure

At first the MPC-controller were going to be implemented on top of a PI-controller structure. The MPC was then going to use the set-points as manipulated variables (inputs). This was changed at an early stage of the work, and it was decided to use MPC without any PI-controller in the layer beneath. This choice was justified by the fact that the most important issue about this project was to look at the difference between the two methods. In a real application PI-controllers in the bottom layer would have been nice for stabilizing the system, but also as backup if the MPC-controller failed.

In SEPTIC (see Chapter 4.3.2) an output variable has to be defined either as an integrator or not as an integrator. In this system three variables behaved differently depending on which input was manipulated. This is a little bit hard to explain, but have something to do with the strong correlations in the system. Table 4.2 gives an overview of which outputs this happened to. A simple solution to this problem was to use PI-controllers on the two most connected input/output parameters. Control valve 3 for level control in the boot section of the deliquidizer and control valve 2 to secure the correct liquid content in the gas outlet of the degasser.

Table 4.2: Some outputs behavior to different inputs

Input:	Behavior:
Control valve 1	Integrators
Control valve 2	Non-integrators
Control valve 3	Non-integrators
Control valve 4	Integrators
Power pump	Integrators
Power compressor	Integrators

4.3.2 The MPC software, SEPTIC

StatoilHydro has its own MPC-tool which is called SEPTIC. This program has been used for the MPC-application in this project. SEPTIC uses linear step-response models as standard, but it is possible to use non-linear models if needed. In this project, linear step-response models have shown to be appropriate.

It is not possible for SEPTIC to directly communicate with Matlab. To solve this, an OPC-server has been used. The OPC-server are standing “between” Matlab and SEPTIC, and gets the calculated values from the two programs. More about the file structure and the Matlab code are given in Appendix C.

4.3.3 An example on the step-response models

In Figure 4.7 one of the step-response models in SEPTIC is shown. This is one out of 32 models which are implemented in SEPTIC. The large number of models arises from the fact that SEPTIC needs the response from each input and disturbance to each output. Since there are 4 inputs, 4 disturbances and 4 outputs, the total number of models is $4*4+4*4=32$.

The models were made very easily by applying a step to the Matlab model in open loop. SEPTIC logged the responses, and the models could be made.

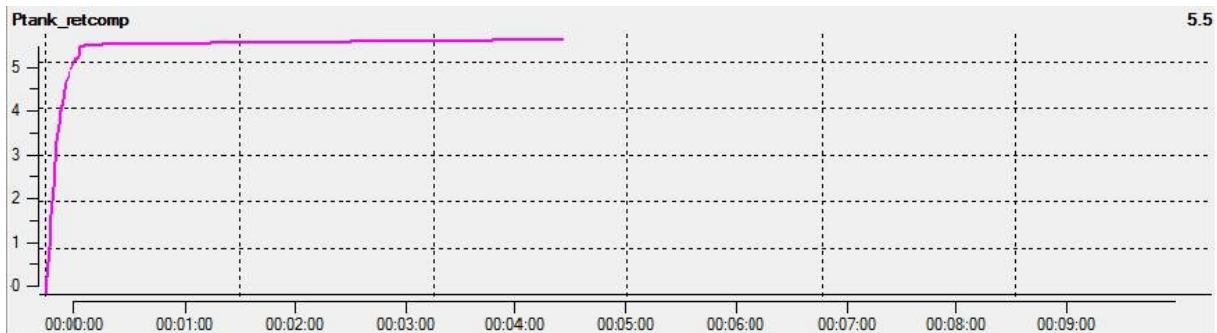


Figure 4.7: An example of a step-response model in SEPTIC

4.4 Tuning of the MPC-controller

All the tuning parameters for the MPC-controller are given in Table 4.3.

Table 4.3: Tuning parameters for the MPC-controller

Tuning parameters	Manipulated variables (Inputs)				Controlled variables (Outputs)			
	Valve 1	Valve 3	Pump	Comp.	L. tank	P. tank	Inflow deg.	Inflow del.
HighPrio	-	-	-	-	1	1	-	-
LowPrio	-	-	-	-	1	1	1	1
SetPntPrio	-	-	-	-	3	2	-	-
IvPrio	4	4	-	-	-	-	-	-
Span	0.1	0.1	0.05	0.05	0.1	0.2	0.001	0.001
MaxUp	0.1	0.2	0.5	0.5	-	-	-	-
MaxDown	-0.1	-0.2	-0.5	-0.5	-	-	-	-
MovePenalty	400	400	7	4	-	-	-	-
Fulf	40	80	-	-	10	50	-	-
HighPenalty	-	-	-	-	500	500	-	-
LowPenalty	-	-	-	-	500	500	500	400

4.4.1 Choice of priorities

The priorities are given at the upper rows in Table 4.3. Since there are 4 manipulated variables, at least 4 of the priorities could be in use at any time. In normal operation all the priorities are held, but if for example the level in the separator tank passes the high limit, the MPC-controller would use more action to get it back again. If more than 4 variables are out of any of their limits, the MPC-controller would choose to work with the 4 variables with the highest priority numbers.

The most important to control are the high and low limits. The system has high and low limits for level and pressure in the separator tank, while it is enough with low limits on the flow rates into the degasser and deliquidizer. These limits got priority 1.

The second most important to control is the pressure in the separator tank and it has been given the priority 2. Priority 3 has been given to the liquid level in the separator tank.

As can be seen in Table 4.3, two of the controlled variables have not got any set points. This is because it has been chosen to give ideal values to the two return streams instead. There are only 4 manipulated variables, and that means that only 4 parameters can have a set-point. By having ideal values on the return streams, it is possible to keep the use of them as low as possible. They have the lowest priority number, because these are the variables which are going to “save” the system if the flow rates get to small or the compressor or pump need more feed.

4.4.2 Choice of tuning parameters

The span is a scaling parameter and is just set once. The meaning of the span is to scale all parameters such that it is easier to do the sharp tuning afterwards. MaxUp and MaxDown tell how large steps the manipulated variables are allowed to take. These are based on physical limitations of the system. Since the real system does not exist yet, the parameters are decided by qualified guesses.

MovePenalty say how much a change in a manipulated variable should be penalized. These values are set very high for the return streams, compared to the pump and compressor power. This is because the MPC-controller should try to use the compressor or the pump, rather than the return streams.

The High and Low penalties are set very high. These are soft constraints, but they should not be crossed.

Fulf is the last tuning parameter, and is the penalty of deviation from the set-points. The Fulf is a little bit lower for the level in the separator tank, compared to the other variables. This is because it is desirable to have slow level control in the separator tank. Then disturbances would be smoothed out, instead of being transferred down stream.

5 Simulation results

The Matlab code developed in this project runs 18 different cases, and then plots and saves the results. All these plots can of course not be included in this chapter, and are instead attached to this report as emf files. A list of the different cases is given in Appendix B, and a selection of the most important results is given here.

5.1 Pressure in the separator tank

Figure 5.1 is taken from Case 1. In this case the set-point of the separator pressure is increased after 100 seconds, and then decreased again after 300 seconds. The figure shows how the set-point tracking are in each application. It also shows the use of compressor power.

In Figure 5.2 there also has been a set-point change in pressure, but this time the change have been the opposite way. This result comes from Case 3.

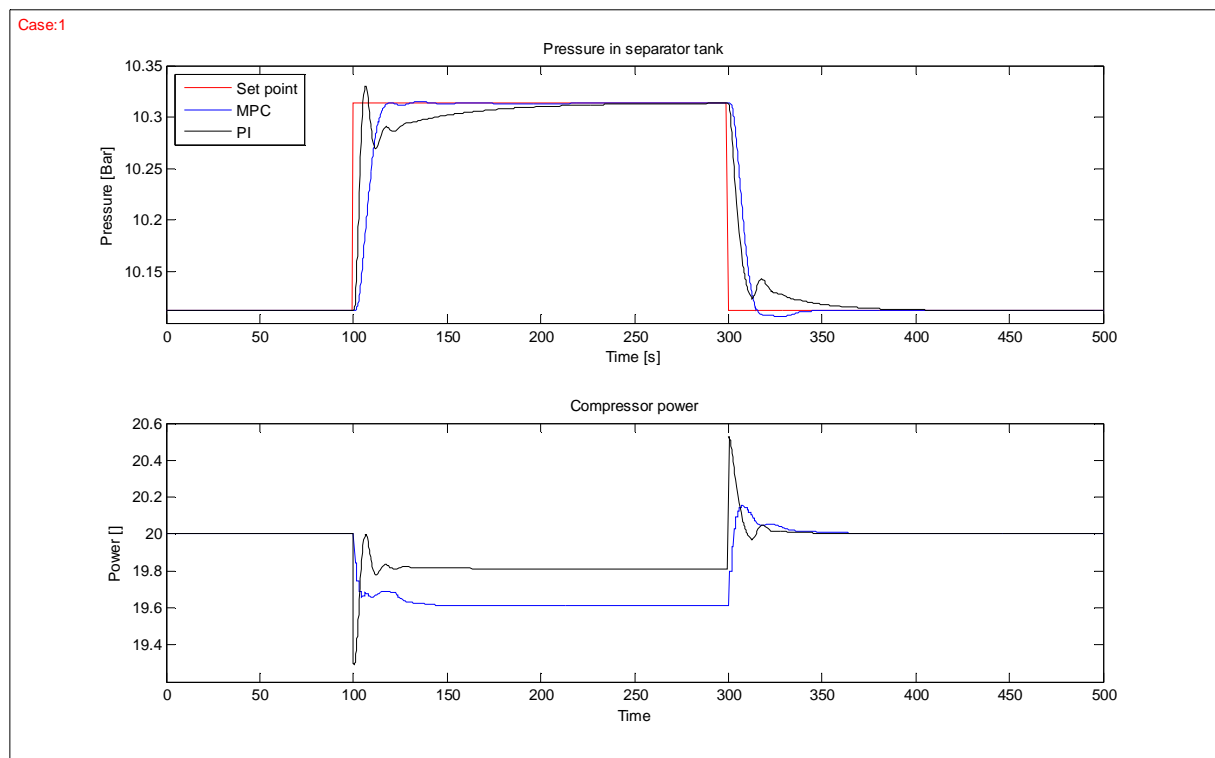


Figure 5.1: Pressure in separator and compressor power after a set-point change in pressure

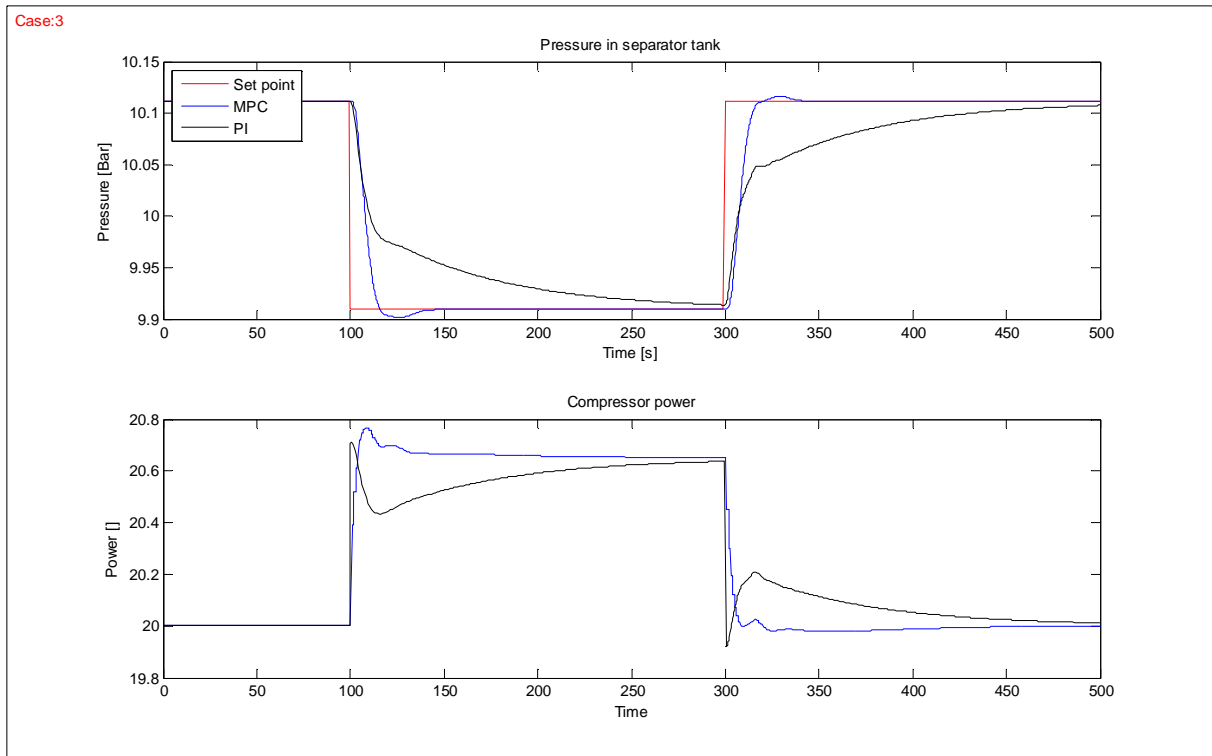


Figure 5.2: Pressure in separator and compressor power after a set-point change in pressure

5.2 Liquid level in the separator tank

Figure 5.3 is also taken from Case 1. The figure shows how the liquid level in the separator changes with time, and how the pump power is used.

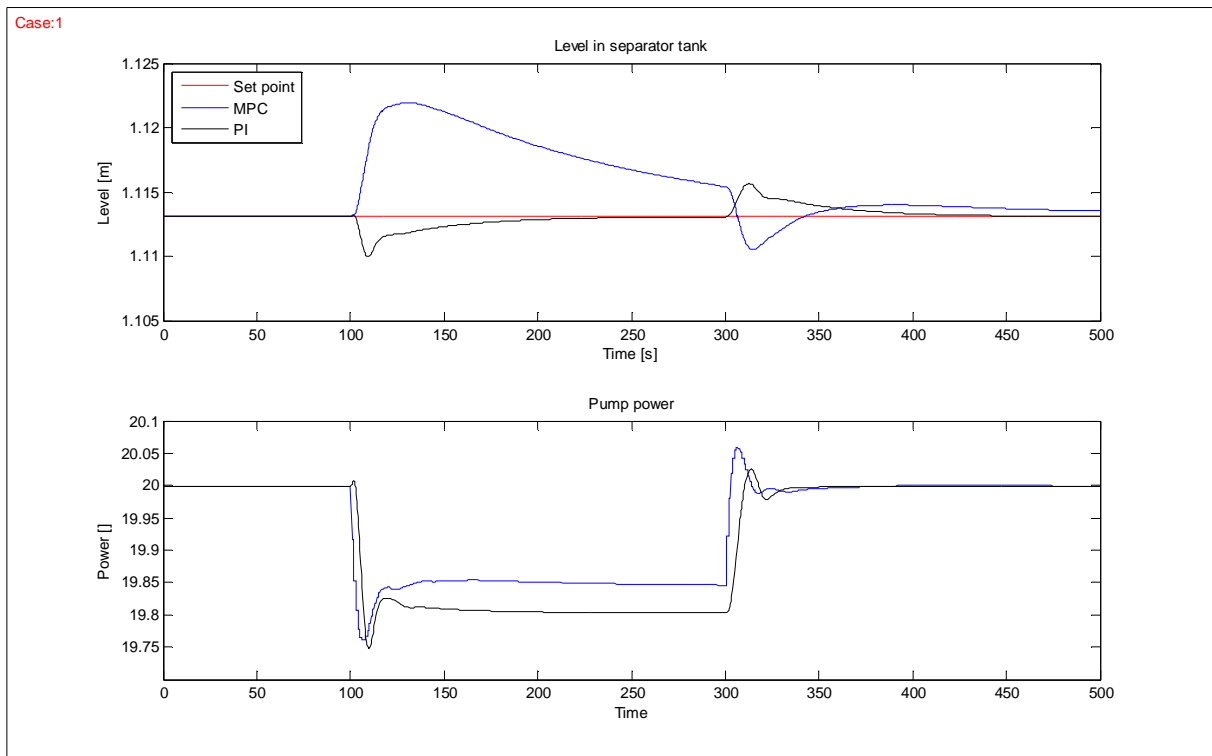


Figure 5.3: Liquid level in separator and pump power after a set-point change in pressure

5.3 Liquid level in the boot section of the deliquidizer

Another result from Case 1 is given in Figure 5.4. Here the liquid level in the boot section of the deliquidizer is shown. It is important to remember that this variable is controlled by a PI-controller in both applications.

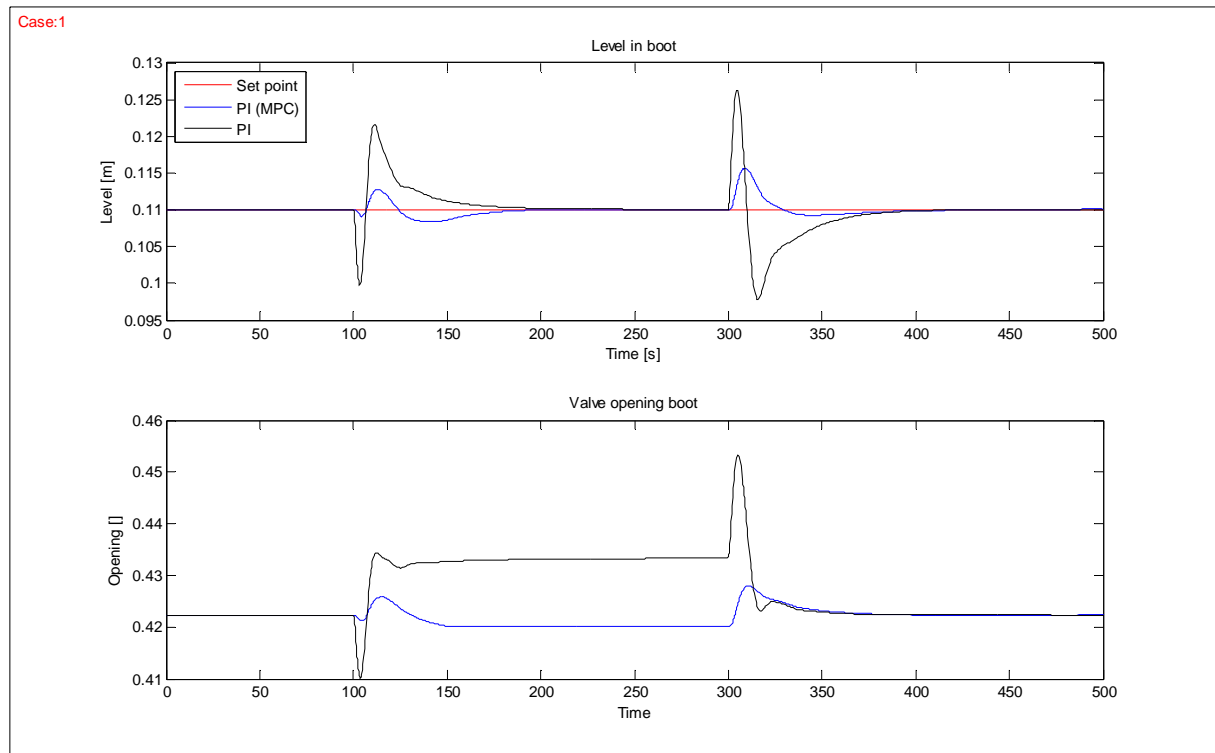


Figure 5.4: Liquid level in boot section of deliquidizer after a set-point change in pressure

5.4 Set-point change level, separator tank

In Case 5 there is a set-point change in the liquid level of the separator. After 100 seconds the set-point is increased, and after 300 seconds it decreased again. Figure 5.5 shows how the two applications try to reach the new set-points, while Figure 5.6 shows how this is affecting the level in the boot of the deliquidizer.

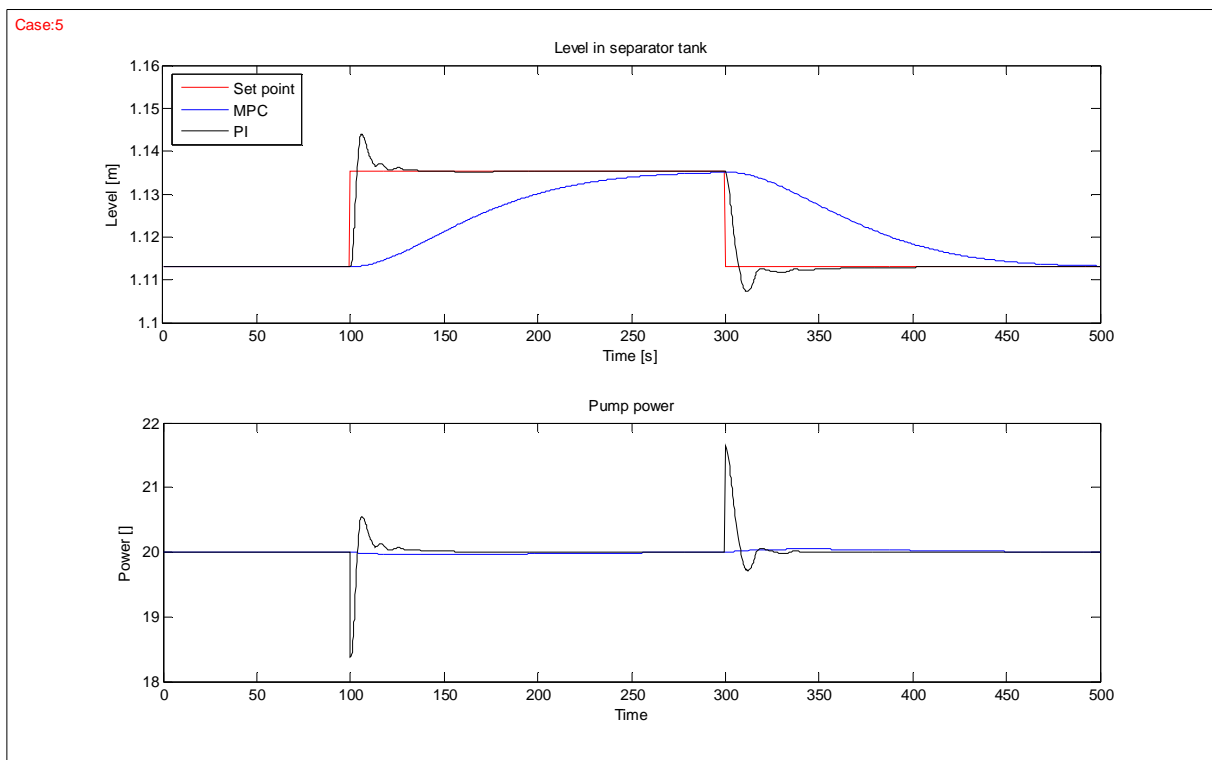


Figure 5.5: Liquid level in separator and pump power after a set-point change in level

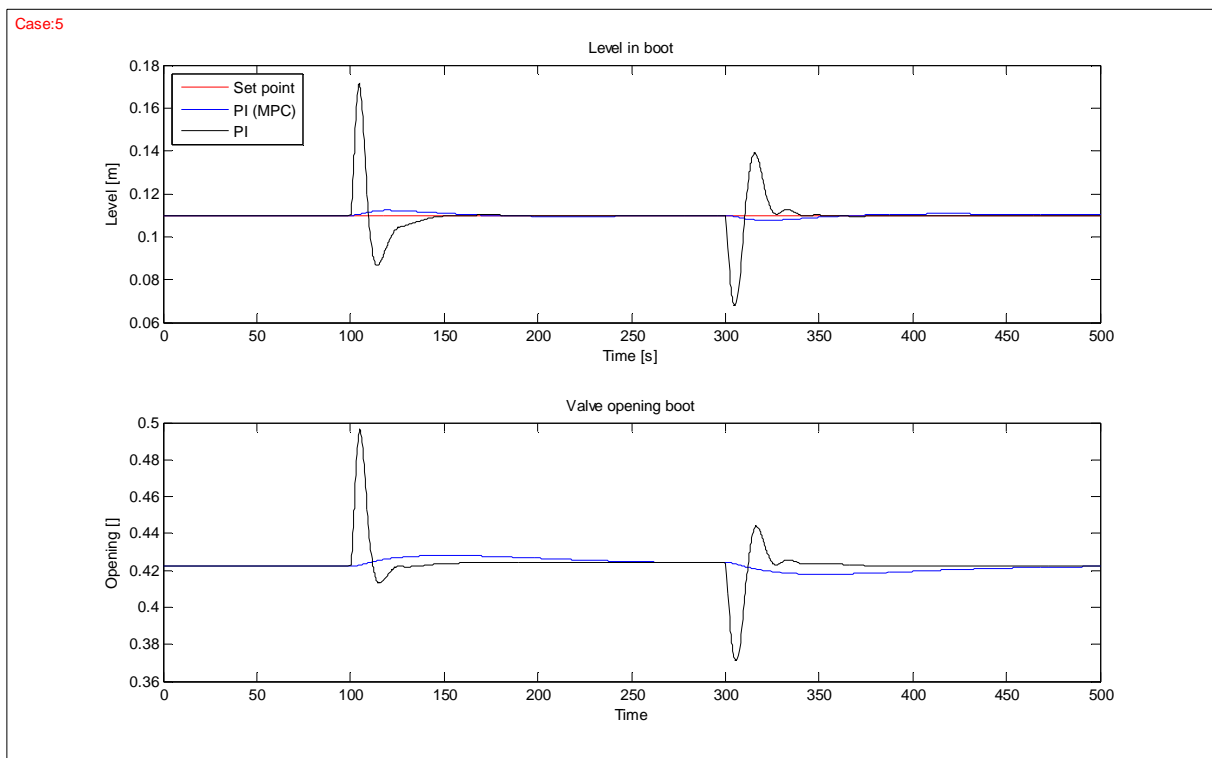


Figure 5.6: Liquid level in boot of deliquidizer after a set-point change in separator level

5.5 Reduced inflow to the system

In Case 27 the total inflow to the system is reduced. This is an interesting case because it will show how the two applications are using the return streams. Figure 5.7 shows the flow rate into the degasser, while Figure 5.8 shows the flow rate into the deliquidizer.

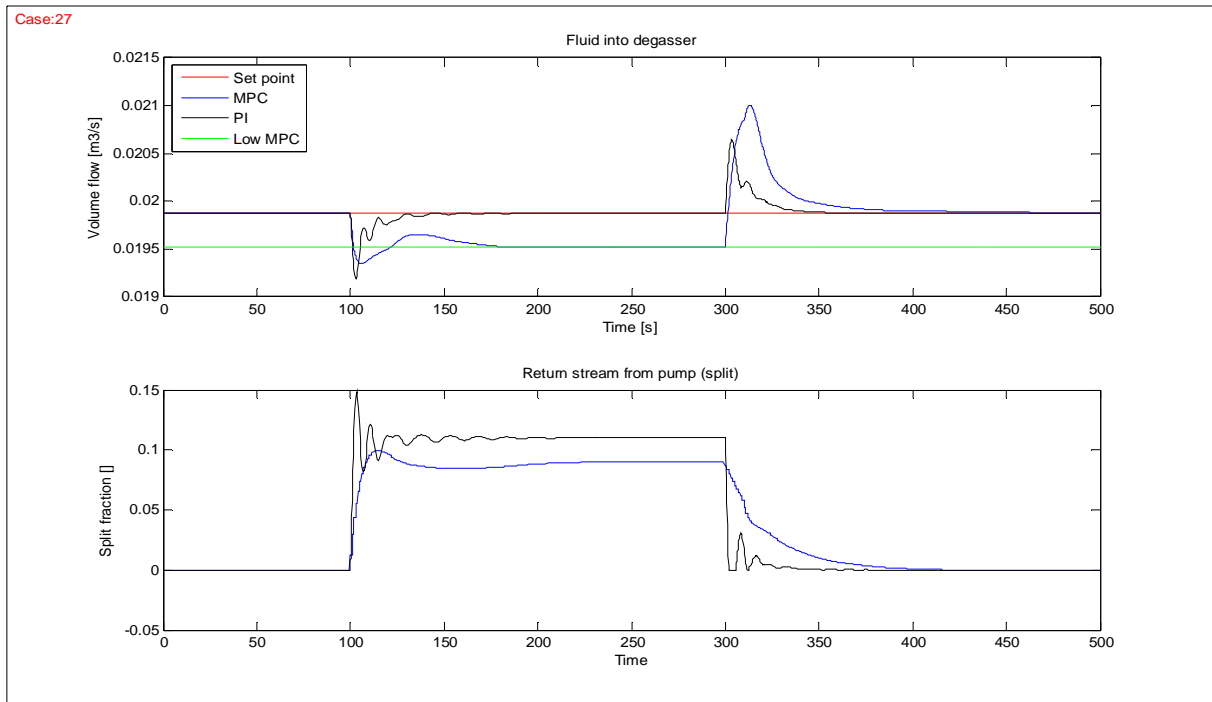


Figure 5.7: Flow rate into degasser when reducing the total inflow to the system

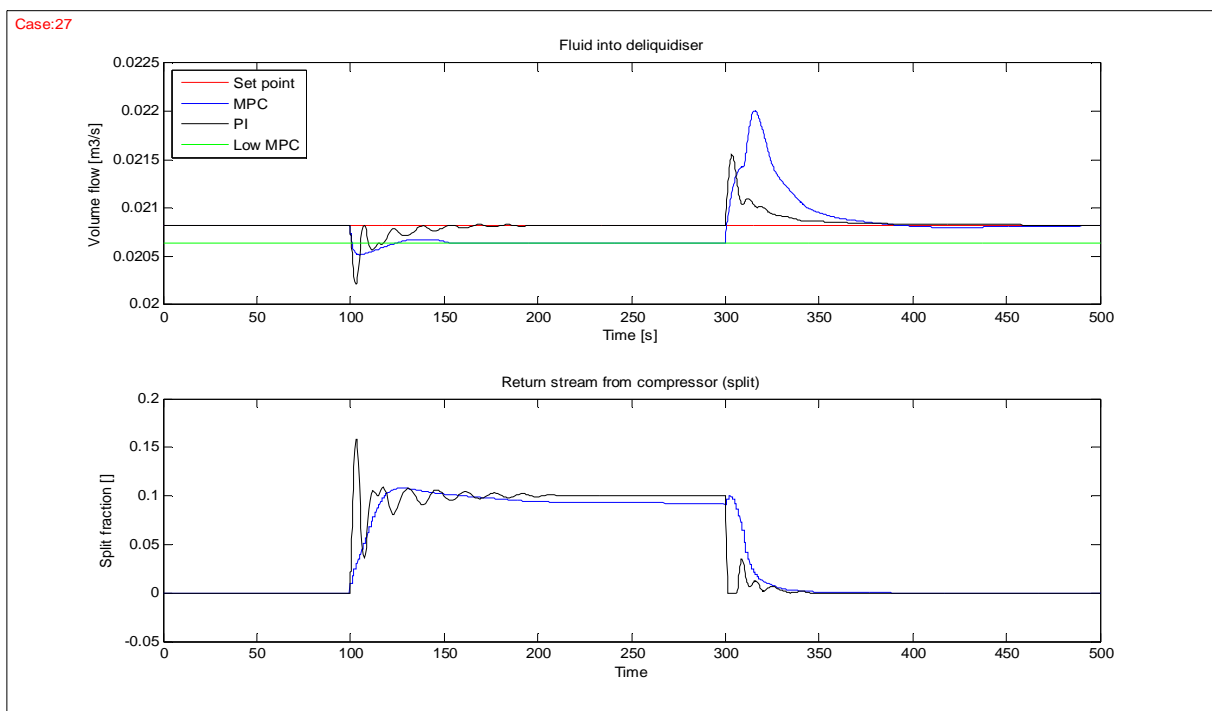


Figure 5.8: Flow rate into deliquidizer when reducing the total inflow to the system

6 Discussion

Almost every case simulated seemed to be interesting for discussion. Unfortunately this would give a very long report. Therefore the most important results are discussed in this chapter. These results are very clear and were confirmed throughout the simulations.

6.1 Pressure in the separator tank

Two different set-point changes for pressure are given in Figure 5.1 and Figure 5.2. The MPC-controller is performing well in both cases, while the PI-controller is much slower in Case 3. The first explanation to this could be the non-linearity of the model, but then the linear MPC-controller should have had the same problems. Another explanation is that the gain is different for the two cases. This is plausible because the return streams have a set-point of zero. Then the system behaves differently for the two different set-point changes because they represent two different “areas” of the system. In Chapter 4.2.3 it was described that the first tuning parameters made the system oscillate for some of the cases. This was of course because the system only was tuned for a change in one direction, and that these tuning parameters gave too rough control in the other direction. A solution to this problem could be to implement a split range PI-controller.

The MPC-controller does not have the same problems because it knows about the restrictions on the return streams. This is one of the properties of the MPC-controller. It would therefore use other inputs more actively when the restrictions on the return streams are met.

6.2 Liquid level in the separator tank

Figure 5.3 shows how different the two applications do set-point tracking of the liquid level in the separator tank. The MPC-controller has a much smoother set-point tracking than the PI-controller. It is easy to see that the closed-loop response τ_c of the PI-controller could have been given a higher value. This would have smoothed out the response of the PI-application as well.

A much more interesting observation in Figure 5.3 is that the use of pump power is very similar for the two applications. The MPC-controller reduces the power despite that the liquid level is raising, and the reduction is almost similar for the two applications. One could think that the power is reduced to control another output variable, and this could of course be the case. But when studying the other plots for Case 1 (given as emf files together with this report), it is obvious that the MPC-controller first of all is reducing the power to control liquid level. This is one of the most important properties of the MPC-controller. It predicts that the liquid level would come back to its set-point over time without using the pump power to actively. Physically this happens because the set-point of the pressure is raised. The increased pressure would help pushing out more and more water. The MPC-controller has also been told that it is more important to have fast set-point tracking of the pressure, and indirectly uses the liquid to raise the pressure faster. Said another way, the MPC-controller knows that reducing the pump power would increase the pressure in the tank.

6.3 Liquid level in the boot section of the deliquidizer

The control of liquid level in the boot section of the deliquidizer is performed by a PI-controller in both applications. The tuning parameters are also the same. In Figure 5.4 it is possible to see how different the two applications perform. The control of the boot is very depending on the liquid control of the separator tank. Since the PI-application has harder set-point tracking of the liquid level in the separator tank, the effect is a worse set-point tracking in the boot.

But this is not the whole truth. A multivariable control system would use every input in a much smoother way than a decoupled PI-controller system. This is discussed more in detail in Chapter 6.6.

6.4 Set-point change level, separator tank

Figure 5.5 and 5.6 shows a set-point change of the level in the separator tank, and how this is affecting the liquid level in the boot. The fast set-point tracking in the separator for the PI application give large deviations from set-point of the liquid in the boot. The MPC-controller is of course very slow to reach the new level set-point, but it is presumed that this is not so important. As already mentioned, a larger closed loop response for the separator PI-controller would probably give better results for the boot.

6.5 Reduced inflow to the system

This is a very interesting case because it shows the use of the return streams. When reducing the total flow into the system, the deliquidizer and degasser do not get enough feed to keep up a good enough separation. To compensate, the return streams have to be used.

Figure 5.7 and 5.8 shows how the reduced inflow is affecting the inflows to the degasser and deliquidizer. The green lines give the low limits which are defined for the MPC-controller. These are just soft-constraints, and we can see that the limits are crossed for a small period. What is interesting is that the PI-controller crosses the limits even more than the MPC-controller. The PI-controller does not know that any limitations exist, and the set-point must therefore be set high enough to avoid the limits. When using a MPC-controller it is possible to lay closer to the limitations, and give high penalty when getting to close. The advantage of this is that the return streams are used less. This could be seen in Figure 5.7 and 5.8. The use of return streams are less for the MPC-application, at least when a new steady-state is reached.

The oscillations give another indication on the hard PI-tuning for some cases. A discussion around this was given in Chapter 6.1.

6.6 The use of manipulated variables

Almost every figure in Chapter 5 shows a smoother use of the manipulated variables for the MPC-controller. This is generally the case for the rest of the case studies also. This could be explained by noticing that the separator system is a very interactive process. Every single unit is coupled together with the other units, either directly or indirectly. A change in any variable could therefore affect every variable in the whole system. Especially in times where the return streams are in use. The effect of this would be that some of the PI-controller loops would work against each other. A typical example of two controllers which are working against each other, are the level and pressure controllers of the separator tank.

Some of the problems mentioned could be solved by breaking the coupling between the controllers. This could be done by introducing cascade control on the most coupled loops. An example of this is given in Figure 6.1. After the decoupling the liquid flow rate would be the manipulated variable rather than the valve position, and the liquid level would no longer be affected by the gas flow (Skogestad 2007). This could also have been done in this case, just that we have a pump and a compressor instead of valves.

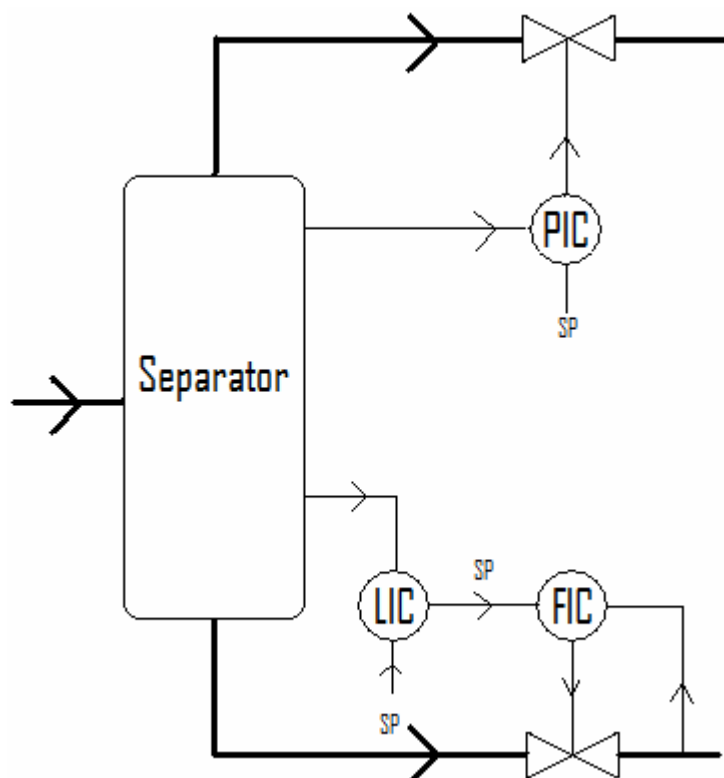


Figure 6.1: An example on cascade control

Another solution to the problems caused by an interactive process is to use multivariable control. The results in Chapter 5 show the effect of this. Almost every figure shows a less aggressive use of the manipulated variable, and at the same time better control of the controlled variable. Multivariable control also gives the user much more freedom when it comes to tuning. It is much easier to design the controller such that it takes care of the most important matters. This is done through the priorities and all possibilities for penalizing. The draw back is that it is much more complicated to tune a MPC-controller.

7 Conclusions

The separator unit studied in this project is a very interactive system. It is very clear that the MPC-controller performed better than the PI-controller application. Some of the most important advantages are listed under:

- Better set-point tracking in boot of deliquidizer
- Better set-point tracking of pressure in separator tank
- Possible to lay closer to the low flow rate limits for the degasser and deliquidizer
- Smoother use of the manipulated variables
- Easier to avoid oscillations
- Less use of the return streams
- It is possible to design the controller to take care of the most important matters at all times (through priorities and penalties)
- Possible to have ideal values for manipulated variables
- It is not necessary to have a set-point for a controlled variable
- The controller know about restrictions and take this into consideration in the calculations

One of the disadvantages of the MPC-controller is that it is much more difficult to implement and tune. A mathematical model of the process must also be present. Nobody knows if the model used in this project is good enough for controlling purposes, since the separator unit never has been built and tested.

Another disadvantage of the MPC-controller is that fewer people know how it is working and what they should do to get a stable system. The MPC-controller also needs more computer power and has a higher probability to crash than a PI-controller. **It would therefore be useful to look at an application with PI-controllers at the bottom layer, and a MPC-controller on top.** Then the PI-controllers would stabilize the system, and also secure it if something happened to the MPC-controller.

Trondheim, 26.11.2007

Christian Ellingsen

Suggestions to further work

- A better model of the separator tank where the separation is a function of flow rate, composition, etc.
- Model the pipe sections between every unit to get more of the dynamics in the system
- Implement phase transformation as a function of pressure (still assume constant temperature)
- Do lab experiments to get more knowledge about how the degasser and the deliquidizer are working (separation profiles and performance as a function of flow rate)
- Implement better and more realistic models for the pump and the compressor
- Try the PI-application with cascade control as discussed in Chapter 6.6
- Test a control structure with PI-controllers in the bottom layer and a MPC-controller on top
- Run more difficult simulation studies (larger steps, combinations of steps, etc.)
- Run start-up and shut-down simulation studies

References

Maciejowski, J.M., *Predictive Control with Constraints*, 1.edition, Prentice Hall, Great Britain 2002.

Skogestad, S., *Simple analytic rules for model reduction and PID controller tuning*, Journal of Process Control, 13:291-309, 2003.

Skogestad, S. and Postlethwaite, I., *Multivariable Feedback Control, Analysis and Design*. JohnWiley and Sons, Ltd, 2.edition, Great Britain 2005.

Appendix A – The mathematical model

In this appendix all the model equations are given. Most of it was completed before the start of the project, but it is included here so that the reader could look in the equations if something about the model or the results are unclear. Any detailed explanation of the model and its notation would not be done, but some helpful information is listed in Chapter A.5.

A.1 The separator tank

The differential equations:

$$\dot{m}_{SEP,L} = w_{SEP,L,in} + w_{SEP,L,return} - w_{SEP,L,out,bottom} - w_{SEP,L,out,top} \quad (A.1)$$

$$\dot{m}_{SEP,G} = w_{SEP,G,in} + w_{SEP,G,return} - w_{SEP,G,out,bottom} - w_{SEP,G,out,top} \quad (A.2)$$

The algebraic equations:

$$V_{SEP,L} = \frac{m_{SEP,L}}{\rho_L} \quad (A.3)$$

$$l_{SEP,L} = \frac{A_{SEP}}{V_{SEP,L}} \quad (A.4)$$

$$V_{SEP,G} = V_{SEP} - V_{SEP,L} \quad (A.5)$$

$$P_{SEP} = \frac{n_{SEP,G}RT}{V_{SEP,G}} \quad (A.6)$$

$$n_{SEP,G} = \frac{m_{SEP,G}}{M_G} \quad (A.7)$$

$$\rho_{SEP,G} = \frac{m_{SEP,G}}{V_{SEP,G}} \quad (A.8)$$

$$P_{SEP,bottom} = P_{SEP} + \rho_L g l_{SEP,L} \quad (A.9)$$

$$K_{SEP,bottom} = K_{SEP,bottom,max} C_{SEP,bottom}^{u_{SEP,bottom}-1} \quad (A.10)$$

$$q_{SEP,L,out,bottom} = \sqrt{\frac{power_{PUMP} - (P_{PUMP,out} - P_{SEP,bottom})}{C_{PUMP}}} \quad (A.11)$$

$$G_{SEP,G} = \frac{\rho_{SEP,G}}{\rho_w} \quad (\text{A.12})$$

$$K_{SEP,top} = K_{SEP,top,max} C_{SEP,top}^{u_{SEP,top}-1} \quad (\text{A.13})$$

$$q_{SEP,G,out,top} = K_{SEP,top} \sqrt{\frac{P_{SEP} - P_{DL}}{G_{SEP,G}}} \quad (\text{A.14})$$

$$w_{SEP,L,out,bottom} = q_{SEP,L,out,bottom} \rho_L \quad (\text{A.15})$$

$$w_{SEP,G,out,top} = q_{SEP,G,out,top} \rho_{SEP,G} \quad (\text{A.16})$$

$$w_{SEP,L,out,top,real} = \alpha w_{SEP,L,out,bottom} \quad (\text{A.17})$$

$$w_{SEP,L,out,bottom,real} = (1 - \alpha) w_{SEP,L,out,bottom} \quad (\text{A.18})$$

$$w_{SEP,G,out,bottom,real} = \beta w_{SEP,G,out,top} \quad (\text{A.19})$$

$$w_{SEP,G,out,top,real} = (1 - \beta) w_{SEP,G,out,top} \quad (\text{A.20})$$

A.2 The degasser

Just algebraic equations:

$$P_{DEG} = P_{SEP,bottom} - G_L \left(\frac{q_{SEP,L,bottom}}{K_{SEP,bottom}} \right)^2 \quad (\text{A.21})$$

$$q_{DEG,L,in} = \frac{w_{SEP,L,out,bottom,real}}{\rho_L} \quad (\text{A.22})$$

$$q_{DEG,G,in} = \frac{w_{SEP,G,out,bottom,real}}{\rho_{DEG,G}} \quad (\text{A.23})$$

$$q_{DEG,in} = q_{DEG,L,in} + q_{DEG,G,in} \quad (\text{A.24})$$

$$rate_{DEG} = 100 \frac{q_{DEG,in}}{q_{DEG,in,ref}} \quad (\text{A.25})$$

$$sep_efficiency_{DEG,G} = \frac{1}{100} (a_4 \cdot rate_{DEG}^4 + a_3 \cdot rate_{DEG}^3 + a_2 \cdot rate_{DEG}^2 + a_1 \cdot rate_{DEG} + a_0) \quad (\text{A.26})$$

$$sep_efficiency_{DEG,L} = \frac{1}{100} (b_4 \cdot rate_{DEG}^4 + b_3 \cdot rate_{DEG}^3 + b_2 \cdot rate_{DEG}^2 + b_1 \cdot rate_{DEG} + b_0) \quad (A.27)$$

$$C_{eff,G} = \frac{(1 - sep_efficiency_{DEG,G})}{sep_efficiency_{DEG,G}} \quad (A.28)$$

$$C_{eff,L} = \frac{(1 - sep_efficiency_{DEG,L})}{sep_efficiency_{DEG,L}} \quad (A.29)$$

$$q_{DEG,L_phase,in} = \frac{(q_{DEG,L,in} - q_{DEG,G,in} C_{eff,G})}{(1 - C_{eff,L} C_{eff,G})} (1 + C_{eff,L}) \quad (A.30)$$

$$K_{DEG,sep} = K_{DEG,sep,max} (C_{DEG,sep}^{u_{DEG,sep}-1} - C_{DEG,sep}^{-1}) \quad (A.31)$$

$$G_{DEG,sep} = \frac{\rho_{DEG,G}}{\rho_w} \quad (A.32)$$

$$IF (P_{DEG} \geq P_{DL})$$

$$q_{DEG,sep} = K_{DEG,sep} \sqrt{\frac{(P_{DEG} - P_{DL})}{G_{DEG,sep}}} \quad (A.33)$$

ELSE

$$q_{DEG,sep} = 0 \quad (A.34)$$

END

$$IF (q_{DEG,sep} \leq q_{DEG,G_phase,in})$$

$$q_{DEG,G_phase,sep} = q_{DEG,sep} \quad (A.35)$$

$$q_{DEG,G,sep} = q_{DEG,G_phase,sep} \cdot sep_efficiency_{DEG,G} \quad (A.36)$$

$$q_{DEG,L,sep} = q_{DEG,G_phase,sep} - q_{DEG,G,sep} \quad (A.37)$$

$$w_{DEG,G,sep} = \rho_{DEG,G,sep} q_{DEG,G,sep} \quad (A.38)$$

$$w_{DEG,L,sep} = \rho_L q_{DEG,L,sep} \quad (A.39)$$

ELSE

$$q_{DEG,sep} = \frac{(q_{DEG,G_phase,in}(\rho_L - \rho_{DEG,G}))}{2\rho_L} \pm \frac{\sqrt{q_{DEG,G_phase,in}^2(\rho_L - \rho_{DEG,G})^2 + 4\rho_w\rho_L K_{DEG,sep}^2(P_{DEG} - P_{DL})}}{2\rho_L} \quad (A.40)$$

$$q_{DEG,G_phase,sep} = q_{DEG,G_phase,in} \quad (A.41)$$

$$q_{DEG,L_phase,sep} = q_{DEG,sep} - q_{DEG,G_phase,sep} \quad (A.42)$$

$$q_{DEG,G,sep} = q_{DEG,G_phase,sep} \cdot sep_efficiency_{DEG,G} + q_{DEG,L_phase,sep} (1 - sep_efficiency_{DEG,L}) \quad (A.43)$$

$$q_{DEG,L,sep} = q_{DEG,G_phase,sep} (1 - sep_efficiency_{DEG,G}) + q_{DEG,L_phase,sep} \cdot sep_efficiency_{DEG,L} \quad (A.44)$$

$$w_{DEG,G,sep} = q_{DEG,G,sep} \rho_{DEG,G} \quad (A.45)$$

$$w_{DEG,L,sep} = q_{DEG,L,sep} \rho_L \quad (A.46)$$

$$END \quad (A.47)$$

$$w_{DEG,G,out} = w_{DEG,G,out,bottom,real} - w_{DEG,G,sep} \quad (A.48)$$

$$w_{DEG,L,out} = w_{SEP,L,out,bottom,real} - w_{DEG,L,sep} \quad (A.49)$$

$$q_{DEG,G,out} = \frac{w_{DEG,G,out}}{\rho_{DEG,G}} \quad (A.50)$$

$$q_{DEG,L,out} = \frac{w_{DEG,L,out}}{\rho_L} \quad (A.51)$$

$$\rho_{DEG,mixed,out} = \frac{w_{DEG,G,out} + w_{DEG,L,out}}{q_{DEG,G,out} + q_{DEG,L,out}} \quad (A.52)$$

$$\rho_{DEG,mixed,sep} = \frac{w_{DEG,G,sep} + w_{DEG,L,sep}}{q_{DEG,G,sep} + q_{DEG,L,sep}} \quad (A.53)$$

$$q_{DEG,out} = q_{DEG,in} - q_{DEG,sep} \quad (A.54)$$

$$G_{DEG,out} = \frac{\rho_{DEG,mixed,out}}{\rho_w} \quad (A.55)$$

$$K_{DEG,out} = K_{DEG,out,max} \left(C_{DEG,out}^{u_{DEG,out}-1} - C_{DEG,out}^{-1} \right) \quad (A.56)$$

$$P_{DEG,out} = P_{DEG} - G_{DEG,out} \left(\frac{q_{DEG,out}}{K_{DEG,out}} \right)^2 \quad (\text{A.57})$$

A.3 The deliquidizer

The differential equations:

$$\dot{m}_{DL,L} = w_{SEP,L,out,top,real} + w_{DEG,L,sep} - w_{DL,L,boot,DiscL} - w_{DL,L,DiscG} \quad (\text{A.58})$$

$$\dot{m}_{DL,G} = w_{SEP,G,out,top,real} + w_{DEG,G,sep} - w_{DL,G,boot,DiscL} - w_{DL,G,DiscG} \quad (\text{A.59})$$

The algebraic equations:

$$V_{DL,L} = \frac{m_{DL,L}}{\rho_L} \quad (\text{A.60})$$

$$m_{DL,L,boot} = m_{DL,L} - A_{DL,L} L_{DL} \rho_L \quad (\text{A.61})$$

$$l_{DL,boot} = \frac{m_{DL,L,boot}}{\rho_L A_{DL,boot}} \quad (\text{A.62})$$

$$V_{DL,G} = V_{DL} - V_{DL,L} \quad (\text{A.63})$$

$$P_{DL} = \frac{m_{DL,G} RT}{M_G V_{DL,G}} \quad (\text{A.64})$$

$$\rho_{DL,G} = \frac{m_{DL,G}}{V_{DL,G}} \quad (\text{A.65})$$

$$P_{DL,boot,bottom} = P_{DL} + \rho_L g l_{DL,boot} \quad (\text{A.66})$$

$$q_{DL,G,in} = \frac{w_{SEP,G,out,top,real} + w_{DEG,G,sep}}{\rho_{DL,G}} \quad (\text{A.67})$$

$$q_{DL,L,in} = \frac{w_{SEP,L,out,top,real} + w_{DEG,L,sep}}{\rho_L} \quad (\text{A.68})$$

$$sep_efficiency_{DL,G} = \frac{1}{100} (c_4 \cdot rate_{DL}^4 + c_3 \cdot rate_{DL}^3 + c_2 \cdot rate_{DL}^2 + c_1 \cdot rate_{DL} + c_0) \quad (\text{A.69})$$

$$sep_efficiency_{DL,L} = \frac{1}{100} (d_4 \cdot rate_{DL}^4 + d_3 \cdot rate_{DL}^3 + d_2 \cdot rate_{DL}^2 + d_1 \cdot rate_{DL} + d_0) \quad (\text{A.70})$$

$$C_{eff_L} = \frac{(1 - sep_efficiency_{DL,L})}{sep_efficiency_{DL,L}} \quad (A.71)$$

$$C_{eff_G} = \frac{(1 - sep_efficiency_{DL,G})}{sep_efficiency_{DL,G}} \quad (A.72)$$

$$q_{DL,L_phase,in} = \frac{(q_{DL,L,in} - q_{DL,G,in} C_{eff,G})}{(1 - C_{eff,L} C_{eff,G})} (1 + C_{eff,L}) \quad (A.73)$$

$$q_{DL,G_phase,in} = \frac{(q_{DL,G,in} - q_{DL,L,in} C_{eff,L})}{(1 - C_{eff,L} C_{eff,G})} (1 + C_{eff,G}) \quad (A.74)$$

$$u_{DL,G_phase} = \frac{q_{DL,G_phase,in} + \gamma_{DL} q_{DL,L_phase,in}}{A_{DL}} \quad (A.75)$$

$$u_{DL,L_phase} = \frac{u_{DL,G_phase}}{\gamma_{DL}} \quad (A.76)$$

$$A_{DL,L_phase} = \frac{q_{DL,L_phase,in}}{u_{DL,L_phase,in}} \quad (A.77)$$

$$A_{DL,G_phase} = A_{DL} - A_{DL,L_phase} \quad (A.78)$$

$$r_{DL,G_phase} = \sqrt{\frac{A_{DL,G_phase}}{\pi}} \quad (A.79)$$

$$r_{DL,G,0} = r_{DL,G_phase} (1 - \alpha) \quad (A.80)$$

$$r_{DL,G,1} = r_{DL,G_phase} (1 + \alpha) \quad (A.81)$$

$$IF(r_{DL,pipe,DiscG} < r_{DL,G,0})$$

$$A_{DL,L_phase,DiscG} = 0 \quad (A.82)$$

$$A_{DL,G_phase,DiscG} = A_{DL,pipe,DiscG} \quad (A.83)$$

$$ELSEIF(r_{DL,pipe,DiscG} \geq r_{DL,G,0}) \& (r_{DL,pipe,DiscG} \leq r_{DL,G,1})$$

$$A_{DL,L_phase,DiscG} = \frac{\pi}{3(r_{DL,G,1} - r_{DL,G,0})} (2r_{DL,G,1}^3 - 3r_{DL,G,1}^2 r_{DL,G,0} + r_{DL,G,0}^3) \quad (A.84)$$

$$A_{DL,G_phase,DiscG} = A_{DL,pipe,DiscG} - A_{DL,L_phase,DiscG} \quad (A.85)$$

ELSE

$$A_{DL,L_phase,DiscG} = \frac{\pi}{3(r_{DL,G,1} - r_{DL,G,0})} (2r_{DL,G,1}^3 - 3r_{DL,G,1}^2 r_{DL,G,0} + r_{DL,G,0}^3) + \pi(r_{DL,pipe}^2 - r_{DL,G,1}^2) \quad (A.86)$$

$$A_{DL,G_phase,DiscG} = A_{DL,pipe,DiscG} - A_{DL,L_phase,DiscG} \quad (A.87)$$

END

$$G_L = \frac{\rho_L}{\rho_w} \quad (A.88)$$

$$K_{DL,boot,DiscL} = K_{DL,boot,DiscL,max} C_{DL,boot,DiscL}^{u_{DL,boot,DiscL}^{-1}} \quad (A.89)$$

$$q_{DL,boot,DiscL} = K_{DL,boot,DiscL} \sqrt{\frac{P_{DL,boot,bottom} - P_{DEG,out}}{G_L}} \quad (A.90)$$

$$A_{DL,L} = A_{DL,L_phase} sep_efficiency_{DL,L} + A_{DL,G_phase} (1 - sep_efficiency_{DL,G}) \quad (A.91)$$

$$w_{DL,G,boot,DiscL} \quad (A.92)$$

$$w_{DL,L,boot,DiscL} = q_{DL,boot,DiscL} \rho_L \quad (A.93)$$

$$q_{DL,DiscG} = \sqrt{\frac{power_{compressor} - (P_{compressor,out} - P_{DL})}{C_{compressor}}} \quad (A.94)$$

$$u_{DL,L_phase,DiscG} = \frac{q_{DL,DiscG}}{\gamma_{DL} A_{DL,G_phase,DiscG} + A_{DL,L_phase,DiscG}} \quad (A.95)$$

$$u_{DL,G_phase,DiscG} = \gamma_{DL} u_{DL,L_phase,DiscG} \quad (A.96)$$

$$q_{DL,G_phase,DiscG} = u_{DL,G_phase,DiscG} A_{DL,G_phase,DiscG} \quad (A.97)$$

$$q_{DL,L_phase,DiscG} = u_{DL,L_phase,DiscG} A_{DL,L_phase,DiscG} \quad (A.98)$$

$$q_{DL,G,DiscG} = q_{DL,G_phase,DiscG} sep_efficiency_{DL,G} + q_{DL,L_phase,DiscG} (1 - sep_efficiency_{DL,L}) \quad (A.99)$$

$$q_{DL,L,DiscG} = q_{DL,L_phase,DiscG} sep_efficiency_{DL,L} + q_{DL,G_phase,DiscG} (1 - sep_efficiency_{DL,G}) \quad (A.100)$$

$$w_{DL,G,DiscG} = \rho_{DL,G} q_{DL,G,DiscG} \quad (A.101)$$

$$w_{DL,L,DiscG} = \rho_L q_{DL,L,DiscG} \quad (A.102)$$

A.4 The return streams

Algebraic equations:

$$w_{G,return} = split_{pump,out} (w_{DEG,G,out} + w_{DL,G,boot,DiscL}) + split_{compressor,out} w_{DL,G,DiscG} \quad (A.103)$$

$$w_{L,return} = split_{pump,out} (w_{DEG,L,out} + w_{DL,L,boot,DiscL}) + split_{compressor,out} w_{DL,L,DiscG} \quad (A.104)$$

A.5 Comments to the equations

Here some of the most complicated equations are explained.

Equation (A.17) to (A.20) describes the carry over and carry under for the separator tank. More complicated equations should be implemented.

Equation (A.30) has a long derivation which not is included here. It is based on a mass balance where two homogeneous phases are mixed together as a function of flow rate. The same is the case with equation (A.73) and (A.74).

Equation (A.35) to (A.46) are the split equations for the degasser. At first the flow out of the gas outlet is calculated as there was only gas going through it. If this amount is less than the amount of gas coming into the degasser, the rest of the gas have to go out of the liquid outlet. If the calculated amount is higher than the amount of gas coming in, some liquid should go out of the gas outlet instead of the excess of gas which not is real. Equation (A.40) has a long derivation which not is included.

Equation (A.75) to (A.87) describe most of the separation in the deliquidizer. It is assumed that there would be a radial separation profile inside the deliquidizer. In the middle there would be just a gas phase and at the wall just a liquid phase. Somewhere in between both phases would be present. The separation depends of the area of the different regions, and this is what is modeled in these equations.

Appendix B – The different cases

In Table B.1 all the different cases are listed. All results from each of the cases are given as emf files together with this report. The reason why the cases are listed with only odd numbers is because there are two step changes in each case (the second back to the origin). In the Matlab code the second step has even numbers, but this is not visible when the files are maid.

Table B.1: List of the different cases carried out

Case	Variable changed	Magnitude
1	Set-point of separator pressure	+ 2 %
3	Set-point of separator pressure	- 2 %
5	Set-point of separator liquid level	+ 2 %
7	Set-point of separator liquid level	- 2 %
9	Set-point of liquid level in boot of deliquidizer	+ 10 %
11	Set-point of liquid level in boot of deliquidizer	- 10 %
13	Set-point of gas quality out of the degasser	+ 0.5 %
15	Set-point of gas quality out of the degasser	- 0.5 %
17	Up stream gas rate	+ 10 %
19	Up stream gas rate	- 10 %
21	Up stream liquid rate	+ 10 %
23	Up stream liquid rate	- 10 %
25	Up stream total rate	+ 10 %
27	Up stream total rate	- 10 %
29	Down stream pressure, after compressor	+ 2 %
31	Down stream pressure, after compressor	- 2 %
33	Down stream pressure, after pump	+ 2 %
35	Down stream pressure, after pump	- 2 %

Appendix C – The main Matlab script

The Matlab script given in this appendix was made in connection with the project work. It is not directly important for the results, but gives some information about how the project work was carried out. It also gives some hints about how the OPC-server is working. The connection of the OPC-server was one of the biggest challenges in the project.

```
% Matlab <-> OPC-server <-> Septic
clear all
close all

global DL DEG SEP PUMP COMPRESSOR SYSTEM i time controller_type

% Initialize taglist. defTaglist.m must contain the same tags as
% taglist.txt used by the OPC server
Taglist=defTaglist;

% Remember to start up the OPC server manually

% Connect to the running OPC server:
hOPC=OPCConnect('OPCData.OpcdataCtrl.1','Statoil.OPC.server',[],Taglist);

% Which case(s) are we going to run?
for cases=[1:2:35]; % 35 cases

% Need to tell the matlab-model that the PI-controllers are not going to be
% used:
controller_type = 0;

% Total time for simulation, in seconds
totalTime = 500;

% Defines t0:
t=0;

% The clock
SetOPCValue(hOPC,Taglist,'TIME',t);

% Defining the step interval (SEPTIC sees every second)
interval=1;

% ODE-solver options
options = odeset('OutputFcn', @successful_step, ... % Function called at
                'MaxStep', 1, ... % each successful step
                'RelTol', 1e-3, ...
                'AbsTol', 1e-6);

for j = cases

    % Starting setptic
    dos('CompSep.lnk &');
```

```

teller=1;

% Reset structures each time
SEP = struct();
DEG = struct();
DL = struct();
PUMP = struct();
COMPRESSOR = struct();

% Time vector for the PI-application
time(1) = 0;

% The equations need a step counter.
i=1;

% Initial conditions
m_SEP_G=1.563496103111137; % Mass of gas in separator tank
m_SEP_L=163.9162857717156; % Mass of liquid in separator tank
m_DL_G=0.040512456587563; % Mass of gas in deliquidiser
m_DL_L=1.581589808630985; % Mass of liquid in deliquidiser
u_DEG_sep=0.699728816488114; % Valve opening at gas outlet of degasser.
% (Describing the dynamics of the degasser)
system_G_return=0; % The amount of gas returned to separator
system_L_return=0; % The amount of liquid returned to separator
deg_rho_mixed_sep=21.560574084683402; % Density gas outlet degasser
power_PUMP=19.999198786079116; % Relative power on pump
power_COMPRESSOR=20.00121499443047; % Relative power on compressor

% Initial state-vector
Y0=[m_SEP_G;
m_SEP_L;
m_DL_G;
m_DL_L;
u_DEG_sep;
system_G_return;
system_L_return;
deg_rho_mixed_sep;
power_PUMP;
power_COMPRESSOR];

% The matlab compsep model is controlled using 6 PI-controllers. These are
% not going to be used in Septic, but the file which gives the controller
% settings is used because it is needed to give the initial values of all
% our MV-variables:
controller_settings

% quality parameters
quality_param

% All the constants in the system
constantsSYSTEM
constantsSEP
constantsDEG
constantsDL
constantsPUMP
constantsCOMPRESSOR

% All the variables for the system:

```

```

set_variablesSYSTEM
set_variablesSEP
set_variablesDL
set_variablesDEG

% Solving all the algebraic equations for the system. We are hopefully at
% steady-state:
computePressures_DL_DEG_SEP
separator_model
degasser
deliquidiser
level_pressure_controller_separator

% Initiate measurements for Septic:
% All MV-variables:
SetOPCValue(hOPC,Taglist,'Split_return_pump',PUMP.bias_split_out);
SetOPCValue(hOPC,Taglist,'Split_return_compressor',...
    COMPRESSOR.bias_split_out);
SetOPCValue(hOPC,Taglist,'Power_pump',PUMP.bias_power);
SetOPCValue(hOPC,Taglist,'Power_compressor',COMPRESSOR.bias_power);
SetOPCValue(hOPC,Taglist,'Valve_opening_degasser',DEG.bias_u_DEG_sep);
SetOPCValue(hOPC,Taglist,'Valve_opening_deliquidiser',...
    DL.bias_u_boot_DiscL);
% All CV-variables:
SetOPCValue(hOPC,Taglist,'Inflow_deliquidiser',DL.q_in(i));
SetOPCValue(hOPC,Taglist,'Inflow_degasser',DEG.q_in(i));
SetOPCValue(hOPC,Taglist,'Pressure_tank',SEP.P(i));
SetOPCValue(hOPC,Taglist,'Level_tank',SEP.h_liquid(i));
SetOPCValue(hOPC,Taglist,'Quality_degasser',DEG.gvf(i));
SetOPCValue(hOPC,Taglist,'Level_boot',DL.l_boot(i));
% All DV-variables:
SetOPCValue(hOPC,Taglist,'Gas_into_compsep',SYSTEM.w_G_in);
SetOPCValue(hOPC,Taglist,'Liquid_into_compsep',SYSTEM.w_L_in);
SetOPCValue(hOPC,Taglist,'Temperature',SYSTEM.Temp);
SetOPCValue(hOPC,Taglist,'Pressure_out_compressor',COMPRESSOR.P_out);
SetOPCValue(hOPC,Taglist,'Pressure_out_pump',PUMP.P_ut);
% All the setpoints:
SetOPCValue(hOPC,Taglist,'Sp_pressure_tank',SEP.P_ref);
SetOPCValue(hOPC,Taglist,'Sp_level_tank',SEP.h_liquid_ref);
SetOPCValue(hOPC,Taglist,'Sp_level_boot',DL.l_boot_ref);
SetOPCValue(hOPC,Taglist,'Sp_gvf_deg',DEG.gvf_ref);
SetOPCValue(hOPC,Taglist,'Sp_rate_deg',DEG.q_in_ref);
SetOPCValue(hOPC,Taglist,'Sp_rate_dl',DL.q_in_ref);

% Pause just to make sure that SEPTIC is completely opened
pause(3)

for t=t:interval:t+totalTime

    % Read MVs from SEPTIC via OPC server
    PUMP.bias_split_out=GetOPCValue(hOPC,Taglist,'Split_return_pump');
    COMPRESSOR.bias_split_out=GetOPCValue(hOPC,Taglist,...
        'Split_return_compressor');
    PUMP.bias_power=GetOPCValue(hOPC,Taglist,'Power_pump');
    COMPRESSOR.bias_power=GetOPCValue(hOPC,Taglist,'Power_compressor');
    DEG.bias_u_DEG_sep=GetOPCValue(hOPC,Taglist,'Valve_opening_degasser');
    DL.bias_u_boot_DiscL=GetOPCValue(hOPC,Taglist,...
        'Valve_opening_deliquidiser');
    % Read DVs from SEPTIC via OPC server
    SEP.w_G_in=GetOPCValue(hOPC,Taglist,'Gas_into_compsep');

```

```

SEP.w_L_in=GetOPCValue(hOPC,Taglist,'Liquid_into_compsep');
COMPRESSOR.P_out=GetOPCValue(hOPC,Taglist,'Pressure_out_compressor');
PUMP.P_ut=GetOPCValue(hOPC,Taglist,'Pressure_out_pump');

% Simulate for one second
if teller==1
    [T, Y]=ode15s(@diff_equations, [t,t+interval],Y0, options);
else
    % Want to use the last states as new initial conditions
    numSteps=length(T);
    Ystep=Y(numSteps,:);
    [Tstep,Ystep]=ode15s(@diff_equations,[t,t+interval],Ystep,options);
    T=[T(1:(numSteps-1));Tstep];
    Y=[Y(1:(numSteps-1),:);Ystep];
end

% Set process values into OPC server
SetOPCValue(hOPC,Taglist,'Inflow_deliquidiser',DL.q_in(i));
SetOPCValue(hOPC,Taglist,'Inflow_degasser',DEG.q_in(i));
SetOPCValue(hOPC,Taglist,'Pressure_tank',SEP.P(i));
SetOPCValue(hOPC,Taglist,'Level_tank',SEP.h_liquid(i));
SetOPCValue(hOPC,Taglist,'Quality_degasser',DEG.gvf(i));
SetOPCValue(hOPC,Taglist,'Level_boot',DL.l_boot(i));

% Steps in Set-point or DV
if (teller==1/5*totalTime)
    % Different for case 25 and case 27
    if j==25 || j==27
        perturbate
        SetOPCValue(hOPC,Taglist,perturbation_text1{j},perturbed1);
        SetOPCValue(hOPC,Taglist,perturbation_text2{j},perturbed2);
    else
        perturbate
        SetOPCValue(hOPC,Taglist,perturbation_text{j},perturbed);
    end
    % Step back again
elseif teller==3/5*totalTime
    if j==25 || j==27
        j=j+1;
        perturbate
        SetOPCValue(hOPC,Taglist,perturbation_text1{j},perturbed1);
        SetOPCValue(hOPC,Taglist,perturbation_text2{j},perturbed2);
        j=j-1;
    else
        j=j+1;
        perturbate
        SetOPCValue(hOPC,Taglist,perturbation_text{j},perturbed);
        j=j-1;
    end

    % Plot when finished
elseif teller==totalTime

    % Array of handles to active figures
    figure_handler = [];
    scrsz=get(0,'ScreenSize');

    figure_handler(1)=figure('position',[scrsz(1) scrsz(2)...
        4*scrsz(3)/5 4*scrsz(4)/5]);

```



```

subplot(2,1,1)
plot(T,SEP.P_ref_plot,'r',T,SEP.P)
title('Pressure in separator tank')
xlabel('Time [s]')
ylabel('Pressure [Pa]')
hold on

// Then two pages of script for plotting follows. This is not
included here.

% Run the PI application (mostly made last summer)
main_compsep_system

% Plot the PI results on top of the SEPTIC results
figure_handler(1)=figure(2);
subplot(2,1,1)
plot(T,SEP.P,'k')
legend('Set point','MPC','PI',2);
title('Pressure in separator tank')
xlabel('Time [s]')
ylabel('Pressure [Pa]')

subplot(2,1,2)
plot(T,COMPRESSOR.power,'k')
title('Compressor power')
xlabel('Time')
ylabel('Power [W]')

// Then two more pages of script for plotting follows. This is also
not included here.

% Old filename. Now its modified and just saving
plot_and_save_system
end
% Give the time to the OPC between every second of simulation
SetOPCValue(hOPC,Taglist,'TIME',T(end));

% Make sure that the OPC server has the right values before proceeding
while GetOPCValue(hOPC,Taglist,'TIME') < T(end)
end

% Update counter
teller=teller+1;
end

% Close SEPTIC between each case
dos('Process -k qtSeptic.exe');
end
end

% Close the OPC-server after last case
OPCDisconnect(hOPC);

```