

Amund Egeland Gaudernack
NTNU
Norwegian University of
Science and Technology
Faculty of Natural Sciences
Department of Chemical Engineering

Amund Egeland Gaudernack

Model Predictive Control for a De-oiling Hydrocyclone

A Theoretical and Experimental Study

June 2021



Norwegian University of
Science and Technology

Model Predictive Control for a De-oiling Hydrocyclone

A Theoretical and Experimental Study

Amund Egeland Gaudernack

MTKJ

Submission date: June 2021

Supervisor: Sigurd Skogestad, IKP

Co-supervisor: Christian Holden, MTP
Mishiga Vallabhan, MTP

Norwegian University of Science and Technology
Department of Chemical Engineering

Abstract

The emissions of produced water is a growing concern due to the negative environmental impact. A potential solution is de-oiling hydrocyclones, which is the most commonly used technology to treat produced water. The Compact Separator Laboratory (CSL) is a produced water treatment plant, which allows for the study of hydrocyclone performance and the development of control methods to maintain their efficiency.

The primary purpose of this project has been to implement Model Predictive Control (MPC) in the CSL. MPC is an advanced control algorithm, which makes use of a model of the process as well as any available measurements. Implementing MPC may be beneficial as it could help automatically keep the oil content of the produced water below the maximum level of 30 ppm.

A previously developed first-principles hydrocyclone model has been used as the foundation for the model in the MPC. However, certain alterations have been made, notably an experimental polynomial approximation of one of the model parameters. A nonlinear MPC was developed and tested in simulation, before it was implemented in the CSL. It has been tested for two disturbances, namely increases in the inlet flow rate and the inlet oil concentration.

A second-degree polynomial of the internally separated oil in the hydrocyclone was experimentally obtained. The experimentally implemented MPC handled an increase in the inlet oil concentration well. When increasing the flow rate, on the other hand, the MPC failed to act efficiently.

Sammendrag

Utslipp av produsert vann er en voksende bekymring på grunn av den negative miljøpåvirkningen. En potensiell løsning er de-oiling hydrosykloner, som er den mest brukte teknologien for å rense produsert vann. Compact Separator Laboratory (CSL) er et renseanlegg for produsert vann, som muliggjør for undersøkelsen av hydrosykloners ytelse og utviklingen av kontrollmetoder for å vedlikeholde deres effektivitet.

Den primære hensikten til dette prosjektet har vært å implementere Model Predictive Control (MPC) i CSL. MPC er en avansert kontrollalgoritme, som tar bruk av en modell av prosessen, samt tilgjengelige målinger. Implementeringen av MPC kan være gunstig, ettersom det kan hjelpe å automatisk holde oljenivået til det produserte vannet under maksimalgrensen på 30 ppm.

En tidligere utviklet førsteprinsipps-modell har blitt brukt som grunnlaget for modellen i MPC-en. Imidlertid har enkelte endringer blitt gjort, mest merkbart en eksperimentell polynomtilnærming av en av modellens parametre. En ikke-lineær MPC ble utviklet og testet i simulering, før den ble implementert i CSL. Den har blitt testet for to forstyrrelser, nemlig økninger i strømningshastigheten og oljekonsentrasjonen ved innløpet.

Et andregradspolynom av den internt separerte oljen i hydrosyklonen ble eksperimentelt funnet. Den eksperimentelt implementerte MPC-en taklet en økning i oljekonsentrasjonen ved innløpet bra. Ved en økning i strømningshastigheten ved innløpet derimot, klarte ikke MPC-en å handle effektivt.

Preface

This thesis has been written as part of the completion of a M.Sc. degree in Chemical Engineering.

I would like to thank my supervisors Sigurd Skogestad and co-supervisor Christian Holden for the very helpful guidance provided during the course of my work. A special thanks to my co-supervisor Mishiga Vallabhan, who helped me greatly with both theoretical and experimental work. I would also like to thank José Matias, who assisted me with coding and debugging.

Acronyms

CSL Compact Separator Laboratory

CV Controlled Variable

DAQ Data Acquisition

DV Disturbance Variable

FOPDT First-Order-Plus-Dead-Time

IPOPT Interior Point Optimizer

MIMO Multiple-Input-Multiple-Output

MPC Model Predictive Control

MV Manipulated Variable

NHL Norsk Hydroteknisk Laboratorium

NLP Non Linear Programming

OCP Optimal Control Problem

OIW Oil-In-Water

PDR Pressure Drop Ratio

SISO Single-Input-Single-Output

SUBPRO Subsea Production and Processing

VI Virtual Instrument

Nomenclature

Symbol	Units	Description
F_s	-	Flow split
M_h	-	Control horizon
N	-	Time horizon
o	-	Oil
P	bar, Pa	Pressure
P_h	-	Prediction horizon
Q_{ex}	m ³ /h	Excess flow rate
Q_{in}	m ³ /h	Inlet flow rate
Q_O	m ³ /h	Overflow rate
Q_{sep}	m ³ /h	Separated flow rate
Q_U	m ³ /h	Underflow rate
r_U	m	Underflow pipe radius
r_U	m	Overflow pipe radius
V_{HC}	m ³	Hydrocyclone volume
V_O	m ³	Oil-rich volume
V_U	m ³	Water-rich volume
w	-	Water
Z_{in}	%	Inlet flow valve opening
Z_O	%	Overflow valve opening
Z_U	%	Underflow valve opening
$\beta_{in,o}$	-,ppm	Inlet oil fraction/concentration
$\beta_{U,o}$	-,ppm	Underflow oil fraction/concentration
$\beta_{O,o}$	-,ppm	Overflow oil fraction/concentration
ρ	kg/m ³	Density
η	-	Separation efficiency

Contents

1	Introduction	1
1.1	Background	1
1.2	Literature Review	2
1.3	Project Aim	2
1.4	Specialization Project	3
1.5	Thesis Overview	4
2	Process Description	5
2.1	Compact Separator Laboratory	5
2.2	Experimental Setup	7
2.2.1	Mastersizer - Offline Sensor	11
2.3	Startup Procedure	11
3	Hydrocyclone Separation	13
3.1	Background	13
3.2	Properties of Produced Water	13
3.3	Principle	14
3.4	Performance	15
3.4.1	Separation Efficiency	15
3.4.2	Flow Split	16
3.5	PDR Control	17
4	Modeling	18
4.1	Modeling Approaches	18

4.2	Dynamic Mass Balance Model	19
4.2.1	Volume Partitioning	19
4.2.2	Mass Balance	20
4.2.3	Polynomial Fitting	20
4.2.4	Volume Fractions	21
4.2.5	Experimental vs Simulated Model	22
4.2.6	Simulation Parameters	24
5	Model Predictive Control	25
5.1	Background	25
5.2	Principle	26
5.3	Process Variables	27
5.4	Constraints	27
5.5	Objective Function	28
5.5.1	Setpoint Tracking	28
5.5.2	Soft Constraints	29
5.6	MPC Implementation in CasADi	30
5.6.1	Step 1 - Defining the Model	30
5.6.2	Step 2 - Formulating the NLP	31
5.6.3	Step 3 - IPOPT Solver	32
5.7	LabVIEW Implementation	32
6	Results	36
6.1	Simulated MPC	36
6.2	Polynomial Approximation	40
6.2.1	Experimental Data	40
6.2.2	2nd Degree Polynomial	42
6.3	Experimental MPC	44
6.3.1	Change in Inlet Concentration	44
6.3.2	Change in Inlet Flow Rate	46
7	Discussion	48
7.1	Polynomial Approximation	48
7.2	Experimental MPC	49

CONTENTS

7.3 Comparison with Direct Feedback Results	51
8 Conclusion	55
8.1 Future Work	56
References	57
A Matlab Code	61
A.1 Simulated MPC	61
A.1.1 Main	61
A.1.2 Model	65
A.1.3 Controller (Multiple Shooting)	67
A.1.4 Parameters	73
A.1.5 Pressure-Flow Relationship	75
A.1.6 Pressure Equations	76
A.2 Experimental MPC	78
A.2.1 Main	78
A.2.2 Controller	78
A.3 Polynomial Fitting	86
B LabVIEW	93
C Sampling Procedure	94
D Mastersizer Procedure	95

List of Figures

2.1	Compact Separator Laboratory	6
2.2	Process flowsheet	7
2.3	Photo of the hydrocyclone	9
2.4	Oil-In-Water sensor	10
3.1	Simplified hydrocyclone sketch	14
3.2	Separation efficiency [%], as a function of $Q_{in}^{[8]}$	16
3.3	Typical PDR control scheme	17
4.1	Partitioned hydrocyclone system	20
5.1	MPC principle ^[25]	26
5.2	LabVIEW implementation overview	33
5.3	Front panel of the LabVIEW MPC program	35
6.1	Simulated MPC results - $\Delta u_{max} = 1\%$	38
6.2	Simulated MPC results - $\Delta u_{max} = 5\%$	39
6.3	Data used for the polynomial approximation of Q_{sep}	41
6.4	Experimentally determined relationship between η and Q_O	42
6.5	Experimental MPC results - increase in $\beta_{in,o}$	45
6.6	Experimental MPC results - increase in Q_{in}	47
7.1	Direct feedback control - increase in Q_{in}	52
7.2	Direct feedback control - increase in $\beta_{in,o}$	53
8.1	PDR control scheme with MPC as supervisory layer	56

LIST OF FIGURES

B.1 LabVIEW - MPC block diagram 93

List of Tables

- 2.1 Flowsheet terminology 8
- 2.2 Relationship between rpm and $\beta_{in,o}$ 11

- 4.1 Simulation model parameters 24

- 5.1 Key variables 27
- 5.2 LabVIEW - MPC inputs 34
- 5.3 LabVIEW - MPC outputs 34

- 6.1 Simulated MPC - parameters and constraints 37
- 6.2 Data used for polynomial approximation 43
- 6.3 Experimental MPC parameters 44

Chapter 1

Introduction

1.1 Background

Produced water is a commonly occurring bi-product in the oil- and gas industry. It is also referred to as oilfield wastewater and consists of various organic and inorganic components. The discharge produced water may negatively affect the environment and has become an increasing global concern. Therefore, it is desirable to treat the produced water, to minimize the environmental impact of the oil and gas industry.^[1] In Norway, it is legally required that the produced water is separated to a concentration less than 30 mg/L, before being discharged to the sea.^[2]

Since the 1980s, hydrocyclone separation has been the most frequently applied technology for the treatment of produced water.^[3] De-oiling hydrocyclones work by using a centrifugal force to separate the oil droplets from the produced water, which enters through a tangential inlet chamber. The separated oil exits through the stream referred to as the 'overflow', while the treated water exits through the 'underflow'.^[4]

The Compact Separator Laboratory (CSL) has been used as part of this project. It is a part of Subsea Production and Processing (SUBPRO), which is a centre for research-based innovation at NTNU.^[5] The laboratory provides for the inves-

tigation of hydrocyclone performance, and has been used to research potential control structures in order to maintain their efficiency.

1.2 Literature Review

In the last few years, there has been some research into control-oriented modeling of de-oiling hydrocyclones. In Durdevic et al. (2015) a set of First-Order-Plus-Dead-Time (FOPDT) models were experimentally identified.^[6] A control-oriented model of an inline de-oiling hydrocyclone was developed based on mass balance equations in Das and Jäschke (2018).^[7] A first-principles mathematical model for a hydrocyclone was derived in Vallabhan et al. (2020),^[8] which forms the basis of the model used in this thesis. Two non-linear control methods were proposed in Vallabhan and Holden (2020), namely feedback linearization control and sliding mode control.^[9]

Control strategies for hydrocyclones are often based on the Pressure Drop Ratio (PDR),^[6] which will be explained in section 3.5. An issue with the traditional PDR centered control approach, is that it struggles to maintain efficiency when increases in the inlet concentration occur.^{[10][11]}

Due to this problem, three new control schemes have been proposed, a feed-back/cascade, feed-forward and Model Predictive Control (MPC). They have been implemented in simulation,^[10] based on the previously developed first-principles mathematical model of the hydrocyclone.^[8] The feed-back and feed-forward schemes have also been implemented and tested experimentally. MPC on the other hand, has not yet been experimentally implemented.

1.3 Project Aim

The main aim of this project is to implement MPC in the Compact Separator Laboratory. MPC is an advanced process control algorithm, which has the goal of computing the trajectory of one or more inputs u , also referred to as the Manipulated Variable (MV). The MV trajectory which is calculated leads

to the optimal output of the plant, y , which is called the Controlled Variable (CV). The purpose of implementing MPC in this case is to automatically limit the underflow oil concentration beneath the maximum allowed level of 30 ppm. This may be achieved by varying the opening of the hydrocyclone's overflow valve, which is denoted as Z_O . Z_O is considered as the MV in this case, meaning the goal of the MPC is to calculate the optimal overflow valve opening. The oil fraction in the underflow is the CV. The disturbances considered are the inlet flow rate Q_{in} and the inlet oil fraction $\beta_{in,o}$, both of which are controllable in the laboratory.

For the MPC to work properly, it is essential to have a reasonably accurate model of the system. If the model is flawed, using MPC may only make matters worse.^[12] The already developed mathematical model is used as a starting point.^[8] However, certain alterations may have to be made, in order for it to work when implementing the MPC in the CSL.

1.4 Specialization Project

The work presented in this master thesis is to a certain extent a continuation of my specialization project, which was completed in the fall semester of 2020.^[13] The project was partly centered around experimentally determining the oil droplet size distribution at the inlet of the hydrocyclone. An offline sensor was used to investigate how the distribution varied as a result of changing the inlet pressure drop.

Additionally, the overflow rate Q_O was estimated by two different methods, as it unlike the underflow and overflow is not being measured by a flowmeter. The two methods used were the hydrocyclone mass balance, and the valve equation, which were both compared to an experimental measurement of the overflow. Having available methods of estimating Q_O is useful, as this flow rate may be included when modeling the hydrocyclone.

1.5 Thesis Overview

The remainder of this thesis is structured in the following manner:

Chapter 2 contains a description of the experimental process, along with a few practical aspects related to the Compact Separator Laboratory.

Chapter 3 presents the basic theory related to hydrocyclone technology and typical control strategies.

Chapter 4 introduces and explains the model of the hydrocyclone.

Chapter 5 explains the basic principle of MPC, in addition to the methods used for implementation in this project.

Chapter 6 presents the results and the main analysis surrounding them.

Chapter 7 contains further analysis and discussion related to the results and methods, and a comparison with some previously obtained control results.

Chapter 8 gives the conclusions of the project, as well as some suggestions for future work.

Chapter 2

Process Description

This chapter presents an overview of the laboratory and the examined process. Some practical aspects related to the performed experiments are also included.

2.1 Compact Separator Laboratory

The examined process is part of a produced water treatment plant, which is located at Norsk Hydroteknisk Laboratorium (NHL), Trondheim. The plant is known as the Compact Separator Laboratory (CSL), and was completed in the summer of 2017.^[14] It has been used as part of a SUBPRO backed project, with the aims of researching and developing models and control methods related to hydrocyclone separation. The research may have industrial benefits, as advanced control structures may help maintain hydrocyclone efficiency, in the presence of disturbances.^[5] Increased hydrocyclone efficiency could reduce the environmental footprint of the oil-and gas industry.

The experimental test rig has the ability to generate three different disturbances, namely changes in inlet oil concentration, flow rate and droplet distribution. The first two disturbances will be considered when implementing MPC in the laboratory. An overview of the CSL is shown in Figure 2.1



Figure 2.1: Compact Separator Laboratory

2.2 Experimental Setup

The flowsheet in Figure 2.2 gives an overview of the examined process:^[15]

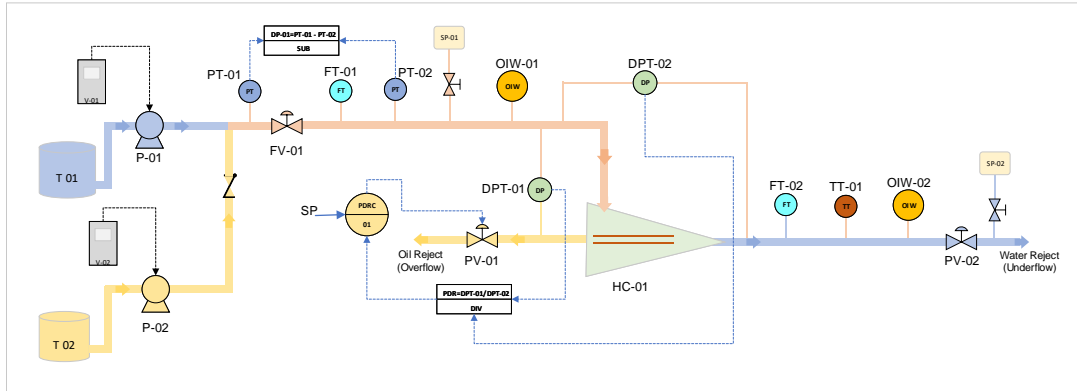


Figure 2.2: Process flowsheet

Table 2.1 contains an explanation of the terminology used in the flowsheet, that is the names and descriptions of the instruments and equipment.

Table 2.1: Flowsheet terminology

Name	Description
T-01	Water tank
T-02	Oil barrel
P-01	Water pump
P-02	Oil pump
V-01	Variable speed drive (P-01)
V-02	Variable speed drive (P-02)
FV-01	Inlet valve
PT-01	Pressure transmitter
FT-01	Flow transmitter
PT-02	Pressure transmitter
SP-01	Sampling point 1
OIW-01	Inlet OIW sensor
DPT-01	Overflow pressure drop transmitter
PV-01	Overflow valve
DPT-02	Underflow pressure drop transmitter
HC-01	Hydrocyclone
FT-02	Flow transmitter
TT-01	Temperature transmitter
OIW-02	Underflow OIW sensor
PV-02	Underflow valve
SP-02	Sampling point 2

The produced water is generated by sending tap water from a 5000 liter water tank, by using the water pump (P-01). Oil is then injected from an oil barrel, by use of the oil pump (P-02). The injected oil then mixes with the water, before it is sent to the inlet of the hydrocyclone (HC-01). Both the oil and water flow rates are controllable by their respective variable speed drives, V-01 and V-02. This allows us to generate two of the previously mentioned disturbances, namely the inlet concentration and flow rate. The inlet concentration may be increased

by increasing the speed of the oil pump, thus injecting a larger amount of oil. The total flow rate may similarly be increased by the water pump. If the goal is to keep the inlet concentration constant at the same time, the oil pump speed may have to be increased as well.

The flow rate of produced water is measured by a flowmeter (FV-01), as is the flow rate of the water reject (underflow) (FV-02). However, the overflow is not being measured. As mentioned previously, there are two methods available to estimate the overflow, namely the valve equation and the overall mass balance of the hydrocyclone.^[13]

The hydrocyclone contains two liners, which are placed in parallel. They have a common inlet and two outlets. A photo of the hydrocyclone is included below.



Figure 2.3: Photo of the hydrocyclone

The inlet valve (FV-01), may be varied in order to change the inlet droplet distribution. Decreasing the valve opening leads to an increase in pressure drop (DP-01), which results in decreasing droplet distributions.^[15] This was part of the focus of the specialization project, however for this thesis we are not considering the droplet distribution.^[13]

Two Oil-In-Water-sensors are present to measure the oil concentration at the inlet and underflow. These measurements allow for the investigation of the hydrocyclone's efficiency. It is also a benefit to be able to measure the inlet concentration, as it is one of the two considered disturbances.

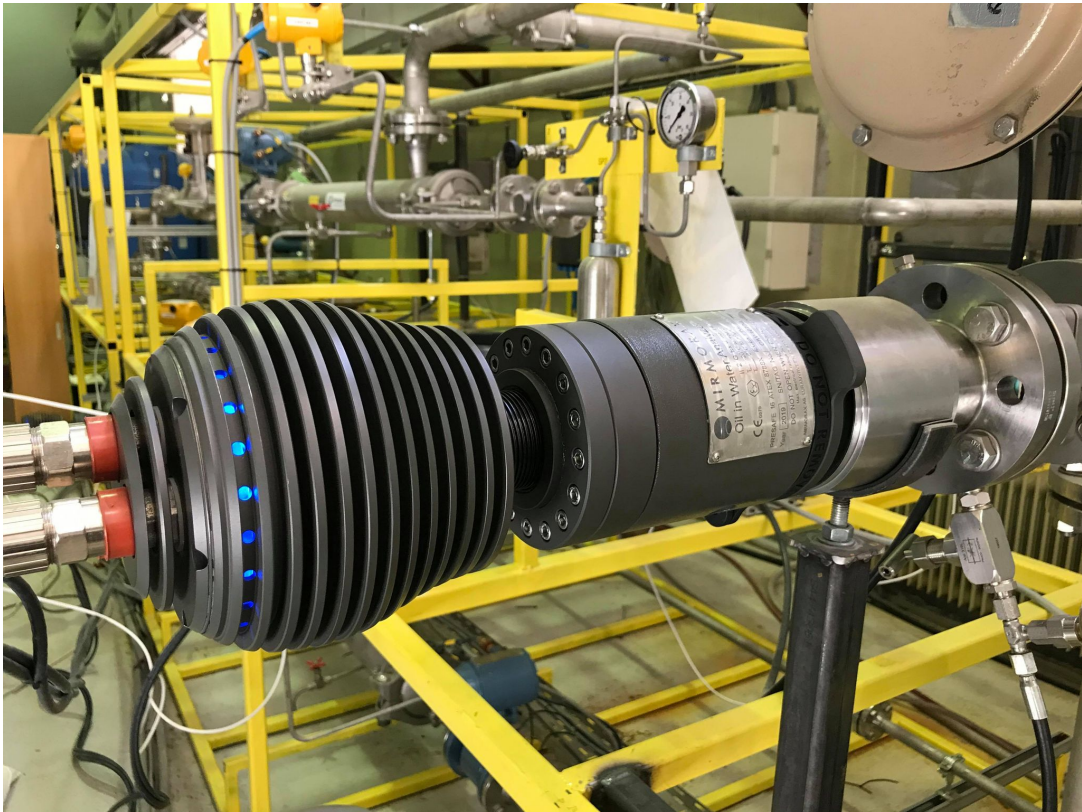


Figure 2.4: Oil-In-Water sensor

In addition to online concentration measurements, the laboratory also has the capability of offline measurements. There are two sampling points (SP-01 and

SP-02) which make this possible. SP-01 and the accompanying sampling procedure are shown in Appendix C.

The inlet OIW-sensor has been too unstable to rely on for accurate measurements of the inlet oil concentration. Therefore, a previously obtained relationship between the oil pump speed (rpm) and inlet concentration, $\beta_{in,o}$, has been used.^[16] The relationship is given in Table 2.2:

Table 2.2: Relationship between rpm and $\beta_{in,o}$

P-02 speed [rpm]	$\beta_{in,o}$ [ppm]
130	165
170	294
200	346
250	455
280	547

All the available measurements are logged by use of the graphical programming language LabVIEW and corresponding Data Acquisition (DAQ) cards. The LabVIEW program is also used to implement control structures and give signals to the laboratory, such as opening a valve or increasing the water pump speed.

2.2.1 Mastersizer - Offline Sensor

The offline sensor is called the 'Mastersizer 3000', which is a device that may be used to measure both the concentration and droplet distribution of a sample. Appendix D gives the procedure used for operating the Mastersizer. It has been used on some occasions to validate the assumed inlet oil concentration.

2.3 Startup Procedure

The general startup procedure of the lab is described in this section. Firstly, it is important to make sure there is a sufficient amount of water in the tank, and

oil in the barrel, and that the air supply is on. When the preliminary actions are performed, the water pump may be activated. For the experiments in this project, we have usually set the pump speed to 1950 rpm, which corresponds to an inlet flow rate of approximately 3.9-4.0 m³/h. The next step is to wait for the measurements of the underflow OIW-sensor to stabilize. The oil pump may then be activated, and once again we wait for the sensor readings to stabilize. When all these steps are finished, experiments may be performed. In this project the experimental actions mostly involve changing the values of certain variables, such as the inlet flow rate and oil concentration.

Chapter 3

Hydrocyclone Separation

This chapter presents some of the most relevant theory concerning de-oiling hydrocyclones, produced water, and control methods applied for hydrocyclones.

3.1 Background

Hydrocyclones have been used to treat produced water in the oil and gas industry since the 1980s. They are sometimes called 'liquid-liquid de-oiling hydrocyclones' or 'enhanced gravity separators'.^[4] They provide numerous benefits compared to conventional gravity separators, which were the most used technology previously. Some of the benefits include the ease of use, the low maintenance required, and their efficiency and compactness.^[3]

3.2 Properties of Produced Water

The properties of produced water depend on several factors. They include the product and location of the field, and the operating conditions and production method. Typically its contents include minerals, oil compounds and gases, which are either dispersed or dissolved. Additionally, production chemicals and solids may be present.^[1]

In the CSL, EXXSOL D60 is used as an oil replacement. The EXXSOL D60 is mixed with tap water to create the produced water. It mainly consists of paraffins, isoparaffins, and cycloparaffins, which are names often used for long-chain hydrocarbons.^[17]

3.3 Principle

Figure 3.1 is a simplified sketch of a hydrocyclone:

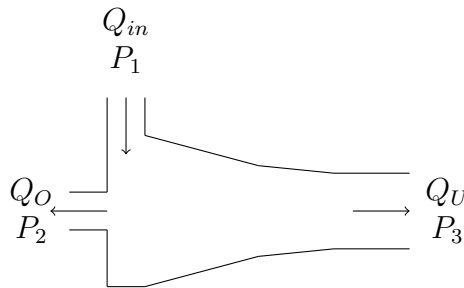


Figure 3.1: Simplified hydrocyclone sketch

As Figure 3.1 illustrates, a hydrocyclone contains an inlet stream Q_{in} and two outlet streams Q_O and Q_U . The produced water stream Q_{in} enters a tangential inlet, which results in a high velocity vortex system. The vortex creates a high acceleration field, forcing the oil towards the centre and the water towards the walls of the cyclone. The oil-rich stream exits through the overflow Q_O , while the treated water exits through the underflow Q_U .^{[18][4]}

A hydrocyclone may have several liners, which may be removed or added depending on flow rate requirements.^[8] The hydrocyclone considered in the CSL has two liners.

Some hydrocyclones have a swirl element placed at the inlet. This forces the flow into a swirl motion, meaning the fluids achieve both angular and axial velocity.^[7] This type is referred to as an 'inline hydrocyclone'. However, the one in the CSL is without a swirl element. In that case, the hydrocyclone's geometry itself induces swirling.^[8]

3.4 Performance

When determining the performance of a de-oiling hydrocyclone, there are two main criteria to consider, namely the separation efficiency and the flow split.

3.4.1 Separation Efficiency

The separation efficiency η is given by Equation (3.1):^[18]

$$\eta = 1 - \frac{\beta_{U,o}}{\beta_{in,o}} \quad (3.1)$$

$\beta_{U,o}$ and $\beta_{in,o}$ represent the fraction of oil in the underflow and inflow respectively. The value of η should be high, indicating a low fraction of oil in the underflow, compared to the oil fraction in the inlet flow. The separation efficiency may alternatively be expressed in terms of flow rate, as $\frac{Q_{sep}}{Q_{in,o}}$.^[8] Q_{sep} represents the internally separated oil flow in the hydrocyclone, and $Q_{in,o}$ represents the inlet oil flow rate.

The inlet flow rate is one of the factors determining the hydrocyclone's separation efficiency. The hydrocyclone in the CSL has a minimum inlet flow rate of 1.44 m³/h, and a maximum of 4.53 m³/h, according to the vendor. However, as the hydrocyclone contains two liners, $Q_{in,min}$ is in fact 2.88 m³/h. Within the range of $Q_{in,min}$ and $Q_{in,max}$, the separation efficiency is nearly constant. On the other hand, outside of this range there is a significant reduction in efficiency. Both when the value is below the minimum and over the maximum, the drop-off is present.^[18] Figure 3.2 illustrates this point.^[8]

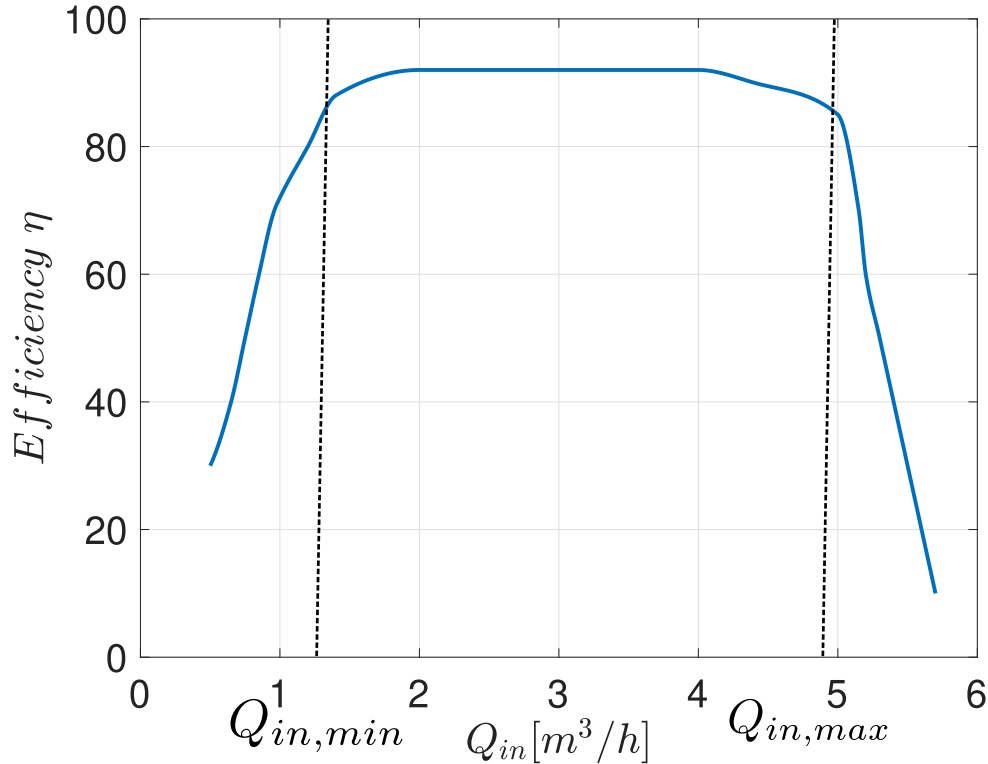


Figure 3.2: Separation efficiency [%], as a function of Q_{in} [8]

3.4.2 Flow Split

Flow split is the other main aspect relevant when assessing hydrocyclone performance. It is also the other factor influencing the separation efficiency. Equation (3.2) gives the flow split:^[18]

$$F_s = \frac{Q_o}{Q_{in}} \quad (3.2)$$

Contrary to the separation efficiency, the flow split should be low. A small value of F_s indicates a low flow rate Q_o . F_s is typically kept in the range of 2-3%, however it may be as low as 1%. The separation efficiency increases rapidly from 0 to 1%. A too low flow split leads to an increased amount of oil in the

underflow. The efficiency stops increasing, and remains nearly constant when increasing the flow split above approximately 3%.^[18]

3.5 PDR Control

As previously mentioned, control methods for hydrocyclones are often based on the Pressure Drop Ratio (PDR), which is defined as:

$$\text{PDR} = \frac{P_1 - P_2}{P_1 - P_3} \quad (3.3)$$

where P_1 , P_2 and P_3 represent the pressures at the inlet, overflow and underflow, as shown in Figure 3.1. The purpose of the PDR control strategy is to keep the flow split F_s at a certain reference value. F_s is linearly correlated with the PDR. Therefore we can choose a setpoint for the PDR value, instead of the flow split. A benefit of this approach is that pressure measurements are generally cheaper and more accurate than flow measurements.^[19] A typical PDR control scheme is shown in Figure 3.3:^[15]

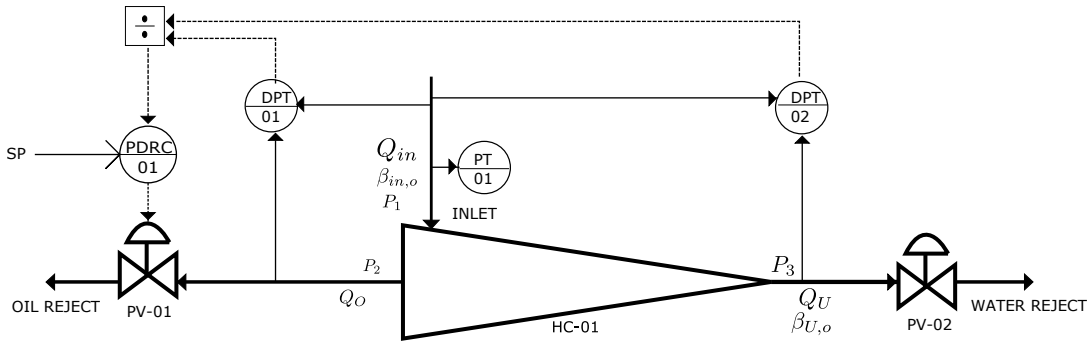


Figure 3.3: Typical PDR control scheme

Chapter 4

Modeling

This chapter introduces a dynamic mass-balance model of the de-oiling hydro-cyclone, which is largely based on Vallabhan et al. (2020).^[8]

4.1 Modeling Approaches

Before MPC can be implemented, a model of the system must be determined. The accuracy of the model is highly important, as there is no way to tune the MPC to compensate for an inaccurate model. In many cases a model may have to be updated, due to changes in the process. The changes may occur because of changes in operating conditions or deterioration of the plant.^[12]

There are two main ways to develop a process model. One way is by using mathematical equations to describe the physical behaviour of the system, which is also known as white-box modeling.^[20] Another approach is known as data-driven modeling, which is based on analysing a system's input and output. In contrast to a physical model, it is not formed from any knowledge of the physical process itself. The data-driven approach is therefore particularly useful when there is little understanding about the system which is being modelled.^[21] Another name for this approach is black-box modeling.

Black- and white-box modeling may be combined, which is an approach known

as grey-box modeling. This method combines the theoretical knowledge of first-principles modeling, with available data. Grey-box modeling was used for a de-oiling hydrocyclone system in Bram et al. (2020), where first principles were used to define the model’s structure and the parameters were identified by a data-driven approach.^[19]

4.2 Dynamic Mass Balance Model

The hydrocyclone in question may be represented by a series of model equations, based on its total mass balance. This model was first proposed in Vallabhan et al. (2020),^[8] and serves as the foundation for the model described in this section. However, some alterations have been made, in order to fit the experimental conditions. Notably, the parameter Q_{sep} has to be experimentally identified, which will be further described in section 4.2.3. Therefore, the approach used may be described as grey-box modeling, since we are combining first-principles modeling with parameter identification.

It is worth noting that there are essentially two models being used, one for implementing MPC in simulation and one for the actual experimental implementation. However, the differences are not that substantial. They are described in section 4.2.5.

4.2.1 Volume Partitioning

The hydrocyclone may be partitioned into an oil-rich and a water-rich volume, denoted by V_O and V_U respectively. V_O is located at the center of the hydrocyclone, where most of the oil will exit through the overflow. V_U is the remainder of the of the hydrocyclone’s volume, which contains a higher fraction of water. The total volume is given by $V_{HC} = V_U + V_O$. Note that while the total volume is constant, the amount of oil and water inside V_O and V_U varies. The partitioned hydrocyclone system is illustrated by Figure 4.1:

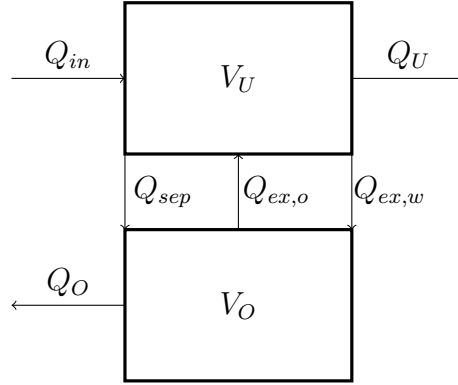


Figure 4.1: Partitioned hydrocyclone system

The subscript o indicates oil flow while w indicates water flow. Q_{sep} represents the flow rate of the internally separated oil. $Q_{ex,w}$ denotes the excess flow rate of water, which V_O receives due to not being completely filled with oil. There will therefore be some water flowing from V_U to fill up the free volume in V_O . Similarly, $Q_{ex,o}$ represents the oil flow rate to the water-rich volume.

4.2.2 Mass Balance

The mass balance is defined for each of the control volumes, with respect to the volume of oil. This may be formulated by the following equations:

$$\frac{dV_{U,o}}{dt} = Q_{in,o} - Q_{sep} - Q_{U,o} + Q_{ex,o} \quad (4.1)$$

$$\frac{dV_{O,o}}{dt} = Q_{sep} - Q_{O,o} - Q_{ex,o} \quad (4.2)$$

4.2.3 Polynomial Fitting

As previously mentioned, there is no way to measure the flow rate of the internally separated oil, Q_{sep} . It is therefore approximated by a second degree polynomial, as given in Equation (4.3):

$$\frac{Q_{sep}}{Q_{in,o}} = (p_2 Q_O^2 + p_1 Q_O + p_0) \quad (4.3)$$

It is assumed that Q_O is the only factor impacting the internal separation efficiency $\frac{Q_{sep}}{Q_{in,o}}$.

Two polynomial approximations are considered. One was obtained in Vallabhan et al. (2020),^[8] where Z_O was varied from 1 to 100 %, with $Z_U = 40$ %, $P_1 = 6$ bar and $\beta_{in,o} = 1000$ ppm. The resulting values of the polynomial coefficients were $p_0 = 0.8414$, $p_1 = 5190$ and $p_2 = -4.821 \cdot 10^7$. This result was achieved by simulation, and is the one used when implementing the simulated MPC. However, for the experimental implementation of MPC, a new polynomial has been experimentally calculated. The results will be presented in section 6.2. In this case, the separation efficiency η is used, which is calculated according to Equation (3.1).

The MATLAB function 'polyfit' has been used for the polynomial approximations. This function finds the polynomial of a specified degree which best fits the data, according to the method of least squares.^[22]

4.2.4 Volume Fractions

The oil volume fractions in the overflow and underflow are found by dividing the oil volume by V_O and V_U respectively, as shown in Equation (4.4) and (4.5):

$$\beta_{O,o} = \frac{V_{O,o}}{V_O} \quad (4.4)$$

$$\beta_{U,o} = \frac{V_{U,o}}{V_U} \quad (4.5)$$

The inlet flow rate of oil is given by Equation (4.6):

$$Q_{in,o} = \beta_{in,o} Q_{in} \quad (4.6)$$

The oil-streams of the underflow and overflow may be calculated as the product of the oil fraction and the flow rate of the respective streams:

$$Q_{O,o} = \beta_{O,o}Q_O \quad (4.7)$$

$$Q_{U,o} = \beta_{U,o}Q_U \quad (4.8)$$

Equation (4.1) and (4.2) may then be rewritten in terms of change in oil fraction instead of volume change. $Q_{ex,o}$ and $Q_{ex,w}$ are assumed to be negligible for the purpose of this model, which further simplifies it to the form given by Equation (4.9) and (4.10):

$$\frac{d\beta_{O,o}}{dt} = \frac{1}{V_O}(Q_{sep} - \beta_{O,o}Q_O) \quad (4.9)$$

$$\frac{d\beta_{U,o}}{dt} = \frac{1}{V_U}(Q_{in,o} - Q_{sep} - \beta_{U,o}Q_U) \quad (4.10)$$

The model may be formulated in state-space form, by representing the two states $\beta_{O,o}$ and $\beta_{U,o}$ as x_1 and x_2 :

$$\dot{x}_1 = \frac{1}{V_O}(Q_{sep} - x_1Q_O) \quad (4.11)$$

$$\dot{x}_2 = \frac{1}{V_U}(Q_{in,o} - Q_{sep} - x_2Q_U) \quad (4.12)$$

4.2.5 Experimental vs Simulated Model

When implementing MPC in simulation, the model being used is exactly as described by Equation (4.9) and (4.10). However, when implementing MPC in the Compact Separator Laboratory, there are some differences that must be considered. For instance, in simulation it was assumed that the hydrocyclone

consists of only one liner. The hydrocyclone in the CSL consists of two liners, meaning that Equation (4.6) must be changed to the following:

$$Q_{in,o} = \frac{\beta_{in,o} Q_{in}}{2} \quad (4.13)$$

Another crucial difference is that when simulating, we can not make use of the flowmeter measurements of Q_U . It is therefore estimated by using the valve equation:

$$Q_U = C_{v1} Z_U \sqrt{\frac{2}{\rho_w} (P_3 - P_{atm})} \quad (4.14)$$

ρ_w is the density of water, P_3 is the pressure at the underflow, P_{atm} is the atmospheric pressure and Z_U is the underflow valve opening. C_{v1} represents the valve constant for the underflow. In the laboratory, this estimation is not necessary, as Q_U is measured directly by a flowmeter.

Q_O is estimated by the same method:

$$Q_O = C_{v2} Z_O \sqrt{\frac{2}{\rho_o} (P_2 - P_{atm})} \quad (4.15)$$

ρ_o is the density of the oil and P_2 is the pressure at the overflow. Z_O is the overflow valve opening, which importantly serves as the control input, u . C_{v2} is the overflow valve constant.

When simulating, P_2 and P_3 are calculated by using the MATLAB function 'fsolve', which is nonlinear system solver.^[23] 'fsolve' is used to solve the above-mentioned valve equations, with respect to P_2 and P_3 .

The valve equation is also used for the experimental MPC, as Q_O needs to be estimated. However, the pressure P_2 may be then be read directly from the pressure transmitter available in the CSL.

4.2.6 Simulation Parameters

Table 4.1 contains the values of the model parameters used for the MPC simulations:

Table 4.1: Simulation model parameters

Parameter	Value	Unit
C_{v1}	$7.854 \cdot 10^{-5}$	m^2
C_{v2}	$3.141 \cdot 10^{-6}$	m^2
$P_{2,init}$	3	bar
$P_{3,init}$	4	bar
r_O	0.001	m
r_U	0.005	m
V_{HC}	$2.090 \cdot 10^{-4}$	m^3
V_O	$5.224 \cdot 10^{-7}$	m^3
V_U	$2.084 \cdot 10^{-4}$	m^3
Z_U	50	%
ρ_o	910	kg/m^3
ρ_w	1000	kg/m^3

Chapter 5

Model Predictive Control

This chapter presents a theoretical overview of Model Predictive Control (MPC). It also includes a description of the developed MPC and the methods used to implement it.

5.1 Background

In various process industries, Model Predictive Control is an often used control approach, especially for Multiple-Input-Multiple-Output (MIMO) systems.^[12] MPC systems were first developed in the 1970's, and has since been used extensively in industry, in addition to being a field often researched in academia.^[24]

There are several advantages related to using MPC. One advantage is the ability to handle constraints, both on the system's inputs and outputs. If the model being used is sufficiently accurate, it may capture the interactions between the inputs, outputs and disturbances, and provide early warnings of any potential errors. However, the advantages are highly dependent on the model accuracy, as an inaccurate model could only make things worse.^[24]

For this particular project, the purpose of implementing MPC is to keep the oil concentration of the underflow stream, $\beta_{U,o}$, below 30 ppm. A couple of different approaches to achieving this goal is discussed later in this chapter.

We may distinguish between nonlinear MPC and linear MPC. The main difference consists of whether the model is linear or not. In this case, the model is nonlinear due to the valve equation. However, for the sake of simplicity, it is usually referred to in this report as simply MPC, as opposed to NMPC.

5.2 Principle

The main idea of MPC is that the controller combines a dynamic model and plant measurements in order to predict future output values.^[24] This principle is demonstrated in Figure 5.1.^[25]

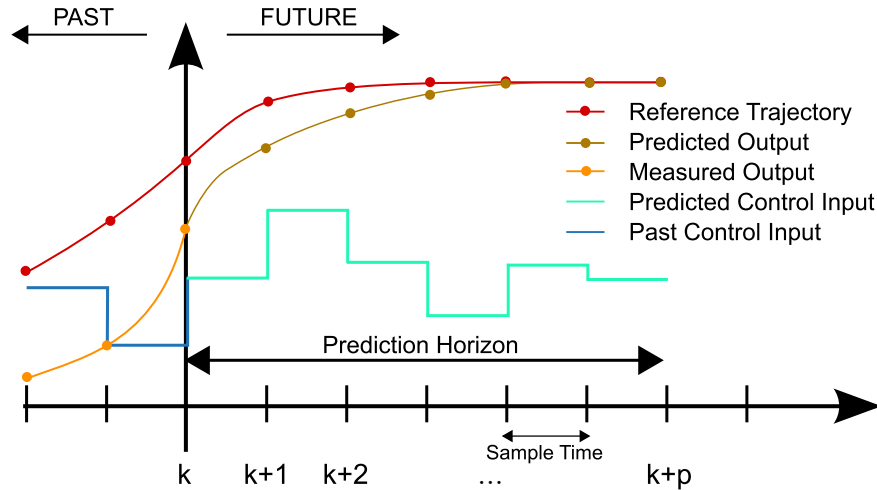


Figure 5.1: MPC principle^[25]

The controller calculates a series of input changes, usually in order to keep the output at a certain set point. Of the input sequence, only the first calculated value is applied to the actual process. Then a new sequence is calculated, and again the first value is fed to the process. The calculation is achieved according to an objective function, J . Further details about the objective function will be provided in section 5.5.

5.3 Process Variables

In MPC applications, the input variable is often referred to as the Manipulated Variable (MV), while the output variable is called the Controlled Variable (CV). Measured disturbances are sometimes called Disturbance Variable (DV)s. The aforementioned variables for this particular project are included in Table 5.1:

Table 5.1: Key variables

Type	Description	Symbol
MV	Overflow valve opening	Z_O
CV	Underflow oil fraction	$\beta_{U,o}$
DV1	Inlet flow rate	Q_{in}
DV2	Inlet oil fraction	$\beta_{in,o}$

This is a Single-Input-Single-Output (SISO) system, since only one MV and one CV are being considered. As mentioned in chapter 4, there are two states, x_1 and x_2 . They represent the oil fraction in the overflow and underflow respectively, and will be used interchangeably with the symbols $\beta_{O,o}$ and $\beta_{U,o}$.

5.4 Constraints

There are two main groups of constraints, which are called equality and inequality constraints. As previously mentioned, one of the main benefits of MPC is the ability to handle inequality constraints, which is the only type of constraint present in this project.

It is often further distinguished between two additional types of constraints. The first type is called hard constraints, which may never be violated. The other type is known as soft constraints, which will be explained in section 5.5.2.

There are certain constraints which must be upheld for practical reasons. For one, the valve opening Z_O , must be within its physical limitations. In other

words, the opening of Z_O must be within in the range 0 to 100 %. This constraint is formulated in the following way:

$$u_{min} \leq u \leq u_{max} \quad (5.1)$$

When implementing MPC, it is often beneficial to restrict excessive input movement. Therefore, an additional constraint is added on the change in input, Δu :

$$-\Delta u_{max} \leq \Delta u \leq \Delta u_{max} \quad (5.2)$$

5.5 Objective Function

The objective function is optimized in order to find the optimal sequence of changes in the MV, the first of which is fed to the process.^[24] The choice of objective function is important as it will largely impact the behaviour of the controller, which should achieve the goal of keeping x_2 below 30 ppm. A couple of different options have been considered, which are presented in this section.

5.5.1 Setpoint Tracking

Setpoint tracking is a commonly used MPC approach, which has been implemented previously in simulation.^[10] It involves the use of an objective function as given in Equation (5.3):

$$J = \min \int_0^{P_h} \left(Q(x_2 - x_2^{SP})^2 + R\Delta u^2 \right) dt \quad (5.3)$$

The principle of this approach is to minimize the difference between the state value, x_2 and a chosen setpoint, x_2^{SP} . The larger the deviation between the achieved value and the setpoint, the more cost is added.

Additionally, it is often a goal to minimize the change in input, represented by Δu . The reason for minimizing Δu is to help stabilize the MPC, and avoid unnecessarily high input usage.

Q and R are tuning parameters. They allow one to choose whether to emphasize the setpoint deviation or input usage, for this particular cost function. Usually, Q is selected as larger than R , meaning a larger emphasis on setpoint deviation. For this particular project, Q must be selected as many orders of magnitude larger than R . The reason why is related to the units of the process variables. x_2 , which represents the underflow oil fraction, has very low numerical values. For instance if $x_2 = 33$ ppm and $x_2^{SP} = 30$ ppm, the numerical value given to the cost function is only $3 \cdot 10^{-6}$, since we are operating with fractions. Δu , on the other hand, may have values between 0 and 1. Therefore Q should be selected as much larger than R , in order to scale for the units.

There is a potential disadvantage related to using setpoint tracking. The disadvantage is that the MPC acts in way to keep x_2 at exactly 30 ppm. However, the goal is to keep x_2 below 30 ppm, not at an exact setpoint. Therefore, it may be disadvantageous to use setpoint tracking.

5.5.2 Soft Constraints

To achieve the stated goal, a better solution may be to use soft constraints. In contrast to hard constraints, soft constraints may be violated. However, the degree of violation is included as a cost in the objective function.^[24]

To include a cost in the objective function, a slack variable s is introduced. s is then added to the cost function, as replacement for the setpoint tracking term of Equation (5.3). Equation (5.4) shows a cost function with soft constraints:

$$J = \min \int_0^{P_h} \left(Qs^2 + R\Delta u^2 \right) dt \quad (5.4)$$

The slack variable s essentially represents the deviation between the value of x_2 and a given setpoint. It may be beneficial to implement soft constraints such

that the value of does not exceed an upper limit nor go beneath a lower limit. The reason for including a lower limit as well, may be justified by not wanting to have too much flow in Q_O .

This method is also known as range control, where the output is not supposed to reach a certain setpoint. Instead, the output value is kept between a lower and an upper limit. The lower and upper limits may be included as hard constraints. However, the use of hard constraints on the outputs could lead to infeasible solutions for the optimization problem.^[24] Therefore, soft constraints as described above, is likely a better option.

5.6 MPC Implementation in CasADi

CasADi is a free, open source framework for numerical optimization, which has been used to implement MPC in this project.^[26] It is available for C++, Python and Octave/MATLAB, with the choice of language having little significance on the performance.^[27] For this project, MATLAB has been used, with the complete code provided in appendix A. Implementation examples of using CasADi, including multiple shooting, which has been used as a template, are given in the attached link.^[28]

The rest of this section contains a rudimentary overview of how CasADi has been used in this project. For a more detailed on how to use CasADi, the reader might find Rawlings, Mayne and Diehl (2019)^[29] and the CasADi manual^[27] useful.

5.6.1 Step 1 - Defining the Model

The first step of using CasADi for MPC implementation is to formulate the state-space model. As mentioned in chapter 4, the model in this particular project is given as:

$$\begin{aligned} \dot{x}_1 &= \frac{1}{V_O}(Q_{sep} - x_1 Q_O) \\ \dot{x}_2 &= \frac{1}{V_F}(Q_{in,o} - Q_{sep} - x_2 Q_U) \end{aligned} \tag{5.5}$$

x_1 and x_2 represents the oil fractions in the overflow and underflow respectively. The equations used to calculate Q_{sep} , Q_O and $Q_{in,o}$ are defined, as is the input u , and the constraints on u and Δu . The states x_1 and x_2 , slack variable s , the model parameters, and cost function J are also defined.

5.6.2 Step 2 - Formulating the NLP

The next step is to formulate a Non Linear Programming (NLP) problem. This is achieved by transforming an Optimal Control Problem (OCP), meaning the cost function J and the associated constraints, into an NLP problem.^[29] The NLP problem is given in the following form:

$$\begin{aligned} \min f(x) \\ g^L \leq g(x) \leq g^U \\ x^L \leq x \leq x^U \end{aligned} \tag{5.6}$$

g^L and g^U represent the lower and upper bound of the constraints, while x^L and x^U are the lower and upper limits for the variables.

There are three commonly used techniques to achieve the transformation, namely direct single shooting, direct multiple shooting and collocation methods. They fall under two categories, sequential and direct approaches. Single shooting falls under former category, while the multiple shooting and collocation belong to the latter category.^[29] For this project, direct multiple shooting has been used.

Direct multiple shooting was first proposed in Bock and Plitt (1984).^[30] The method may be summarized by the following steps, as described in Tamini and Li (2009):^[31]

1. Divide the time horizon into equally sized sub-intervals,
2. For each sub-interval parameterize the control function and initial states, and evaluate the state trajectories,
3. Add continuity constraints,
4. For each sub-interval, compute the objective function.

5.6.3 Step 3 - IPOPT Solver

The final step is to solve the NLP problem. IPOPT is used, which is an NLP solver included in CasADi. The method that IPOPT uses is known as interior point line search filter method. It attempts to find a local solution of an NLP problem as defined in Equation (5.6).^[32]

5.7 LabVIEW Implementation

For the experimental implementation of MPC, an additional step is to implement it in LabVIEW. This sections contains an outline of how the implementation is achieved.

LabVIEW is a graphical programming language which is used for measurements and control of the CSL. It allows for the creation a of Virtual Instrument (VI). A VI consists of user interface, called a front panel, a block diagram and an icon and connector pane. The user interface consists of controls and indicators, which are used to control and read the systems inputs and outputs. The block diagram contains the graphical code.^[33]

An already existing VI has been modified to implement the MPC. The MPC itself is not included directly in LabVIEW, rather LabVIEW calls a MATLAB script which performs the optimization. An overview of the LabVIEW implementation is illustrated by Figure 5.2, while the entire block diagram is provided in appendix B.

At the center of the implementation is a script node, which accesses and runs the MATLAB code. However, the code only runs if the case structure is set to 'True', which is controlled by an indicator switch in the front panel. Otherwise, the node simply clears the MATLAB workspace. It is beneficial to have a simple way of turning off the controller, as we do not want to run the MPC at all times. The MV is sent to the daq, and all the input and output data which is necessary for analysis is stored in an Excel file.

Timed while loop

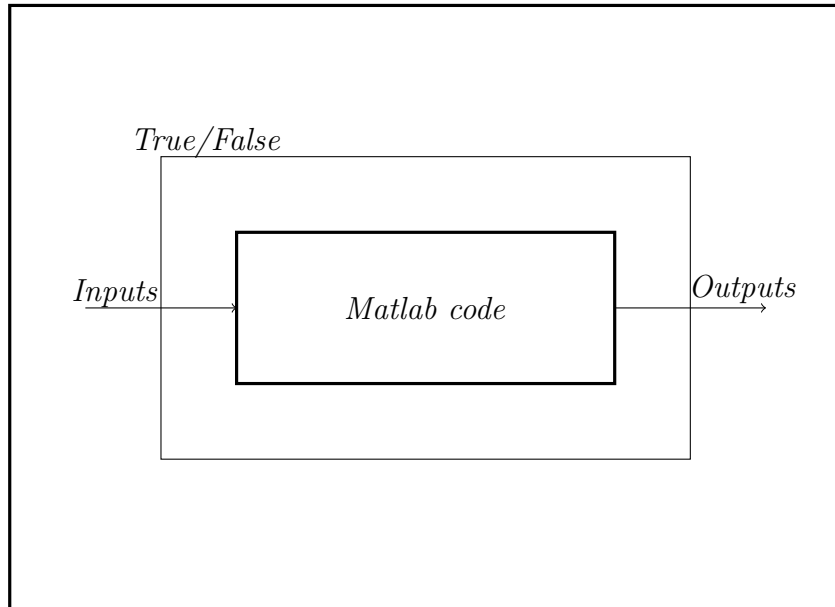


Figure 5.2: LabVIEW implementation overview

The inputs consists of measured and estimated values needed in the model, MPC parameters and constraints. Many of the inputs have been included as control inputs in LabVIEW, in order to enable customization and live experimentation with different values for the parameters and constraints.

Table 5.2 contains a description of the inputs which are given to the MPC:

Table 5.2: LabVIEW - MPC inputs

Input	Description
C_v	Valve constant
Q_{in}	Measured inlet flow rate
Q_O	Estimated overflow rate
Q_U	Measured underflow rate
u_{min}	Minimum input value
u_{max}	Maximum input value
Δu_{max}	Maximum input change
$\beta_{in,o}$	Estimated inlet oil concentration
$\beta_{u,o}$	Measured underflow oil concentration
M_h	Control horizon
P_h	Prediction horizon
P_2	Overflow pressure
Q	Tuning parameter
R	Tuning parameter
x_2^U	Upper ppm limit of $\beta_{U,o}$

Table 5.3 gives a description of the MPC's outputs:

Table 5.3: LabVIEW - MPC outputs

Output	Description
u	Optimal Z_O value
t	Time used to find optimal solution
Solver status	Indicates optimization success/failure
Iterations	Number of iterations optimizer used
Objective	Numerical value of objective function

The key output is naturally the calculated optimal Z_O value. The other values, for instance the solver status, may help indicate how the MPC is performing.

Additionally, the predicted future values of x_1 and x_2 are included, along with the future control inputs.

The part of the LabVIEW front panel related to the MPC is shown in Figure 5.3

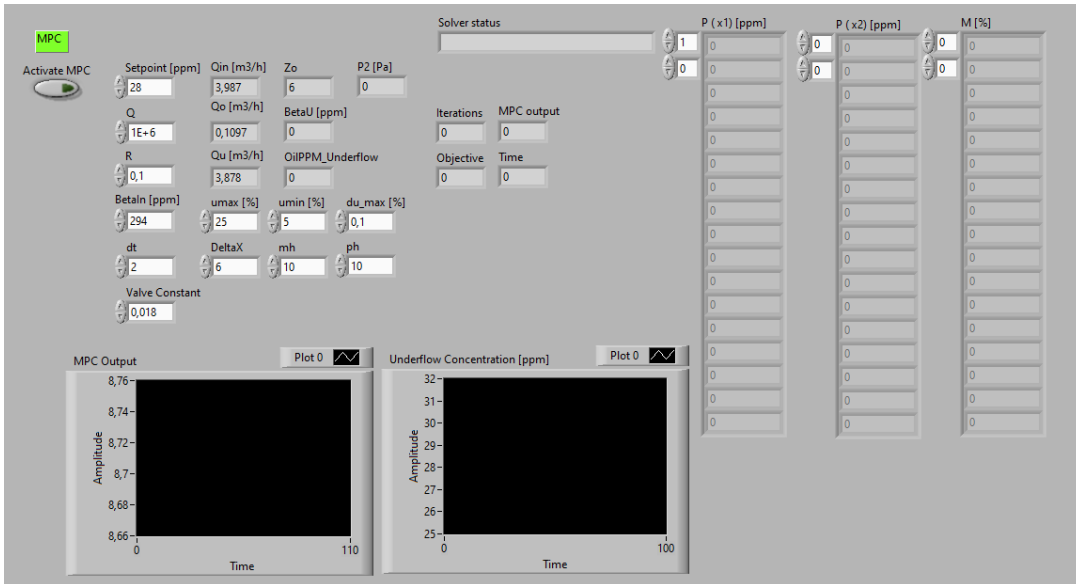


Figure 5.3: Front panel of the LabVIEW MPC program

Chapter 6

Results

The results of this project have been obtained both by simulation in MATLAB and by performing experiments in the Compact Separator Laboratory. Firstly, the simulated MPC results are presented. Then the results related to the polynomial approximation are given. Finally, the results of testing the MPC experimentally are shown.

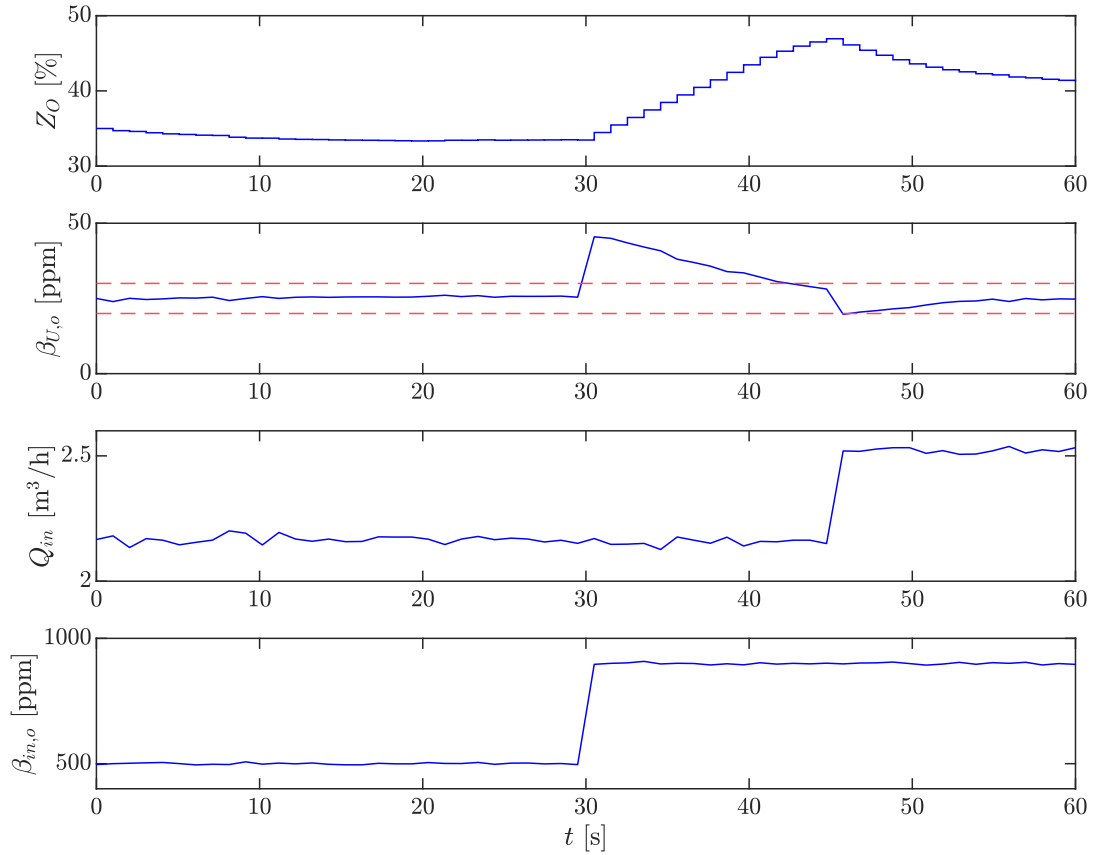
6.1 Simulated MPC

The MPC with soft constraints as described in chapter 5, has been simulated in MATLAB. The values of the parameters and constraints that have been used are given in Table 6.1:

Table 6.1: Simulated MPC - parameters and constraints

Parameter	Value
u_{min}	0%
u_{max}	100%
Δu_{max}	1%
u_0	35%
Q	10^6
R	0.1
M_h	10
N	60
P_h	15
x_2^U	30 ppm
x_2^L	20 ppm
x_2^{init}	25 ppm

When performing the simulations, the inlet concentration $\beta_{in,o}$ is increased from 500 to 900 ppm at $t = 30$, while the inlet flow rate Q_{in} is increased from 2.16 m³/h to 2.52 m³/h at $t = 45$. White Gaussian noise has been added to both Q_{in} and $\beta_{in,o}$, by use of the MATLAB function 'awgn'.^[34] The simulated results are shown in Figure 6.1:

Figure 6.1: Simulated MPC results - $\Delta u_{max} = 1\%$

When the increase in $\beta_{in,o}$ occurs, there is as expected a corresponding increase in $\beta_{U,o}$. The increase leads to $\beta_{U,o}$ surpassing the setpoint of 30 ppm, which is indicated by the upper red line. The controller reacts by increasing the overflow valve opening, Z_O . As the maximum change in Z_O per iteration is 1%, several steps are needed before $\beta_{U,o}$ returns below the upper limit.

The increase in Q_{in} on the other hand, has the effect of lowering the underflow oil concentration. The controller then reacts in the opposite manner, namely by closing Z_O . This occurs because the controller wants to avoid that $\beta_{U,o}$ goes below the lower limit of 20 ppm, which is indicated by the lower red line.

A notable observation that was made, is that the controller appears to stabilize

when $\beta_{U,o}$ is in the midpoint between the upper and lower limits. This is observed towards the end of the simulation, when Z_O keeps decreasing, even though $\beta_{U,o}$ is within the acceptable range of values. The same behaviour is also observed for the first half of the simulation, before there are any steps in the disturbances. In that time period, there are minor decreases in Z_O , before it stabilizes when $\beta_{U,o}$ is approximately 25 ppm.

The value of Δu_{max} likely has a large impact on the behaviour of the controller. Figure 6.2 shows simulated results using the same parameters as in Table 6.1, with the exception of Δu_{max} being 5% instead of 1%.

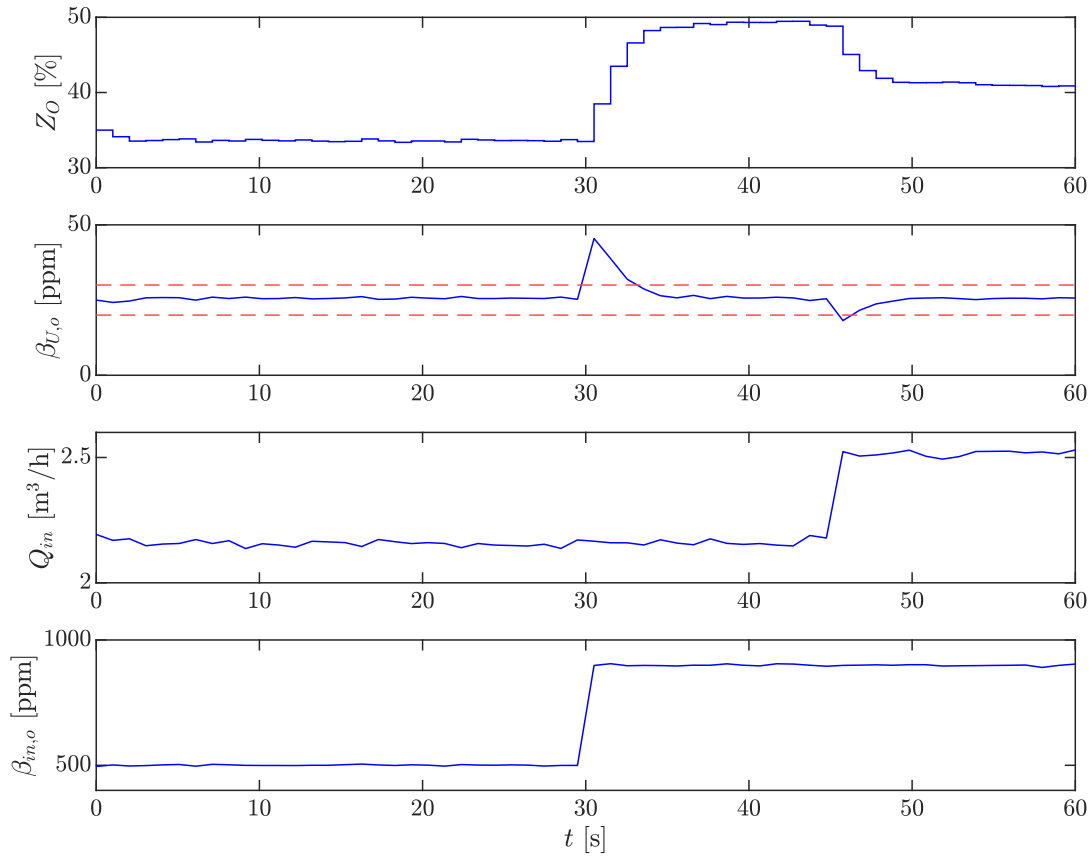


Figure 6.2: Simulated MPC results - $\Delta u_{max} = 5\%$

We observe that changing the constraint on the input change Δu_{max} , does in fact

have a large impact on the controller's behavior. The controller increases Z_O in larger steps, which brings $\beta_{U,o}$ down below the maximum level faster.

The controller is also somewhat more unstable in the region before the disturbances occur, that is from $t = 0$ to 30 s. This means a lower value of Δu_{max} could help stabilize the control action.

6.2 Polynomial Approximation

Two steps have been performed to obtain the polynomial approximation of Q_{sep} . The first step was to gather the necessary experimental data in the Compact Separator Laboratory. The second step was to use the data to obtain the polynomial expression.

6.2.1 Experimental Data

The experimental data used for approximating Q_{sep} was gathered by performing steps on Z_O , which led to varying values of $\beta_{U,o}$ and Q_O . In this case, the values of Q_O were estimated based on the hydrocyclone mass balance. Q_{in} was kept constant at approximately 4.0 m³/h. $\beta_{in,o}$ was assumed constant at 300 ppm, according to the pump-concentration relationship. The value of inlet concentration was also verified by use of the offline Mastersizer sensor. The resulting data is shown in Figure 6.3:

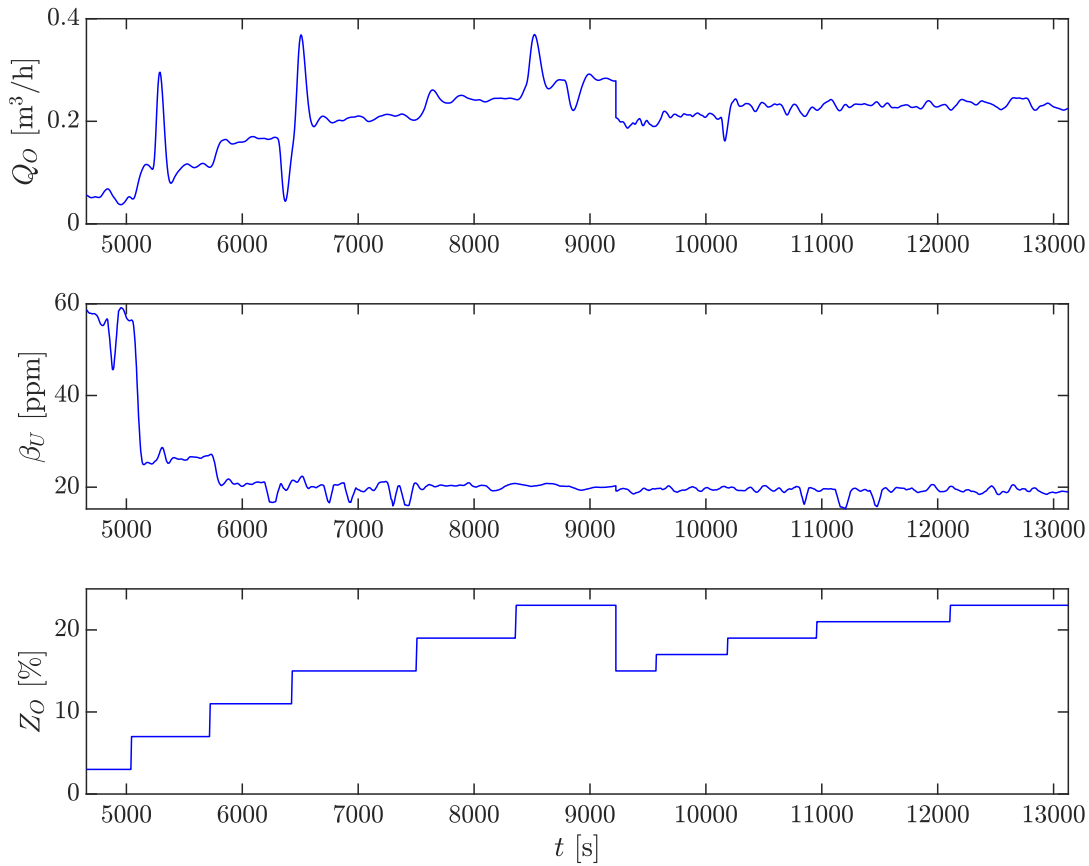


Figure 6.3: Data used for the polynomial approximation of Q_{sep}

Steps were first performed from $Z_O = 3\%$ to 23% , in increments of 4 percentage points. Then additional steps were added for values of Z_O between 15% and 23% , in increments of 2 percentage points. Overflow valve openings from 3% to 23% covers most of the relevant range for the hydrocyclone. At higher values of Z_O , the impact of increasing the valve opening is minimal. At values of Z_O lower than 3% , a significant amount of oil will be sent through the underflow. Therefore, it is undesirable to operate in those ranges of Z_O -values.

The figure shows a significant drop in $\beta_{U,o}$ for the first few steps. For instance when Z_O is increased from 3% to 7% , there is approximately a halving in $\beta_{U,o}$. As the valve opening increases further, the corresponding effect on $\beta_{U,o}$ decreases.

For the last few steps there is virtually no change in $\beta_{U,o}$ at all.

A reason why $\beta_{U,o}$ decreases the most for the first few steps is likely due to the relative change in Z_O . Consider for instance the first 6 steps in Z_O . The absolute increase in Z_O is the same at 4 percentage points. However, the relative increase is smaller for each step. The most noticeable change occurs from 3% to 7%, when the valve opening more than doubles in absolute value. At the later steps, the change in $\beta_{U,o}$ is insignificant.

6.2.2 2nd Degree Polynomial

Figure 6.4 illustrates the results of the experimentally found polynomial for approximating the separation efficiency:

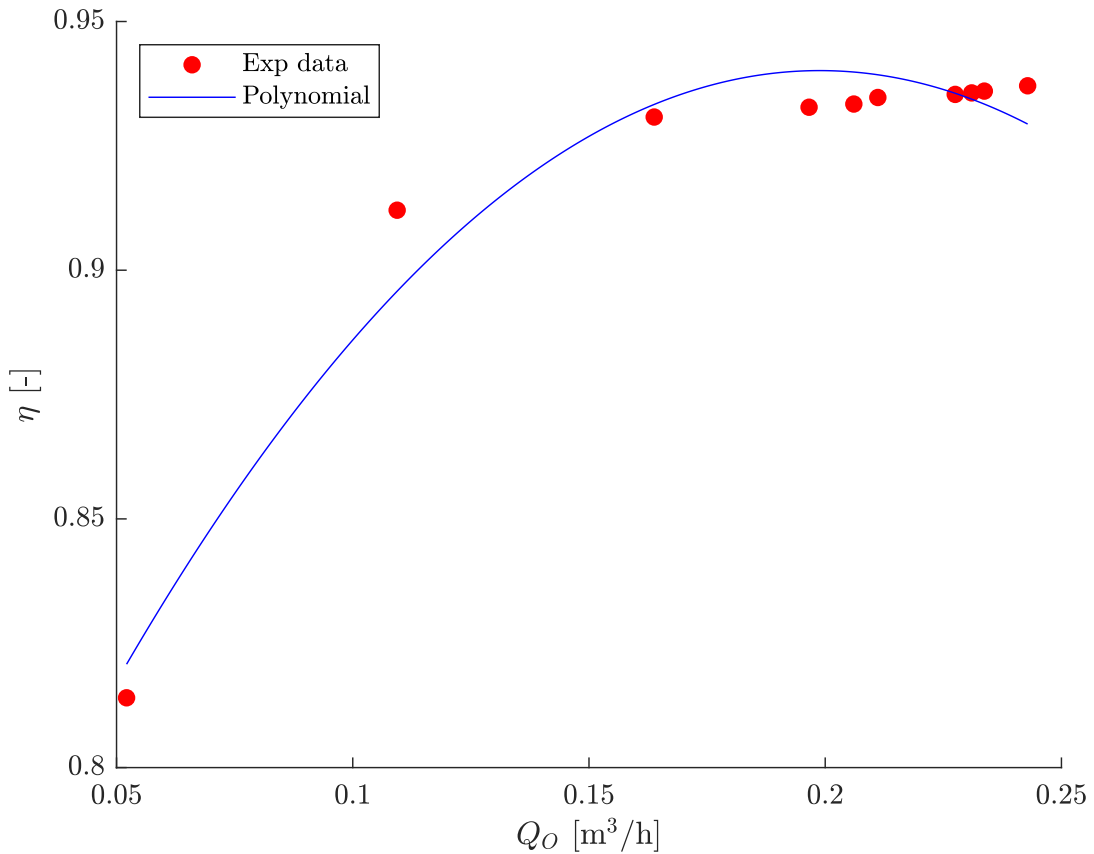


Figure 6.4: Experimentally determined relationship between η and Q_O

The red dots represent the separation efficiency η , which was calculated based on the experimental data shown in Figure 6.3. The calculation of η was based on the average values of $\beta_{U,o}$, for each step of Z_O . $\beta_{in,o}$ was assumed to be constant. The blue line represents the resulting 2nd degree polynomial, which is given in Equation (6.1):

$$\eta(Q_O) = -5.5455Q_O^2 + 2.2052Q_O + 0.7209 \quad (6.1)$$

As previously mentioned, the separation efficiency may be formulated in terms of flow rate as $\frac{Q_{sep}}{Q_{in,o}}$. This gives the following estimate for Q_{sep} :

$$Q_{sep} = Q_{in,o}(-5.5455Q_O^2 + 2.2052Q_O + 0.7209) \quad (6.2)$$

The experimental data is also given in Table 6.2.

Table 6.2: Data used for polynomial approximation

$\eta[-]$	Q_O [m ³ /h]
0.8140	0.05210
0.9121	0.1094
0.9308	0.1638
0.9328	0.1966
0.9334	0.2061
0.9347	0.2111
0.9353	0.2275
0.9356	0.2310
0.9360	0.2337
0.9371	0.2428

The results shown in Table 6.2 and Figure 6.4, further illustrate a previously mentioned point, namely that the separation efficiency increases rapidly for lower values of Q_O , but that the increase tapers off for higher values. If one

ignores the data point at $Q_O \approx 0.05 \text{ m}^3/\text{h}$, the remaining points would show a nearly linear relationship between η and Q_O , which further demonstrates this fact.

6.3 Experimental MPC

As with the simulated MPC, the experimental MPC has been evaluated for two disturbances, namely Q_{in} and $\beta_{in,o}$. However, for practical reasons, we are now considering only one disturbance at the time.

6.3.1 Change in Inlet Concentration

The values of the parameters and constraints that have been used are given in Table 6.3:

Table 6.3: Experimental MPC parameters

Parameter	Value
u_{min}	5%
u_{max}	25%
Δu_{max}	0.1%
u_0	6%
C_v	0.018
x_2^U	28 ppm
x_2^L	22 ppm
Q	10^6
R	0.1

The values differ somewhat from the ones used in simulation. The overflow valve opening has been constrained to the region 5% to 25%, as this is deemed the relevant range. This was previously demonstrated in Figure 6.3. If Z_O is below 5%, a lot of oil will likely be sent through the underflow, which is undesirable. For values of Z_O higher than 25%, the separation efficiency will

not increase. Additionally, a lot of water will be sent through the overflow, which is undesirable. Δu_{max} was set to 0.1%, as small changes are sufficient to achieve relatively large changes in the output. This constraint value will additionally help the controller act in a stable manner.

For the first test of the experimental MPC, Q_{in} was kept constant, while $\beta_{in,o}$ was increased from 294 ppm to 346 ppm. The increase is achieved by increasing the rpm of the oil pump from 170 to 200, as per Table 2.2. The results are shown in Figure 6.5:

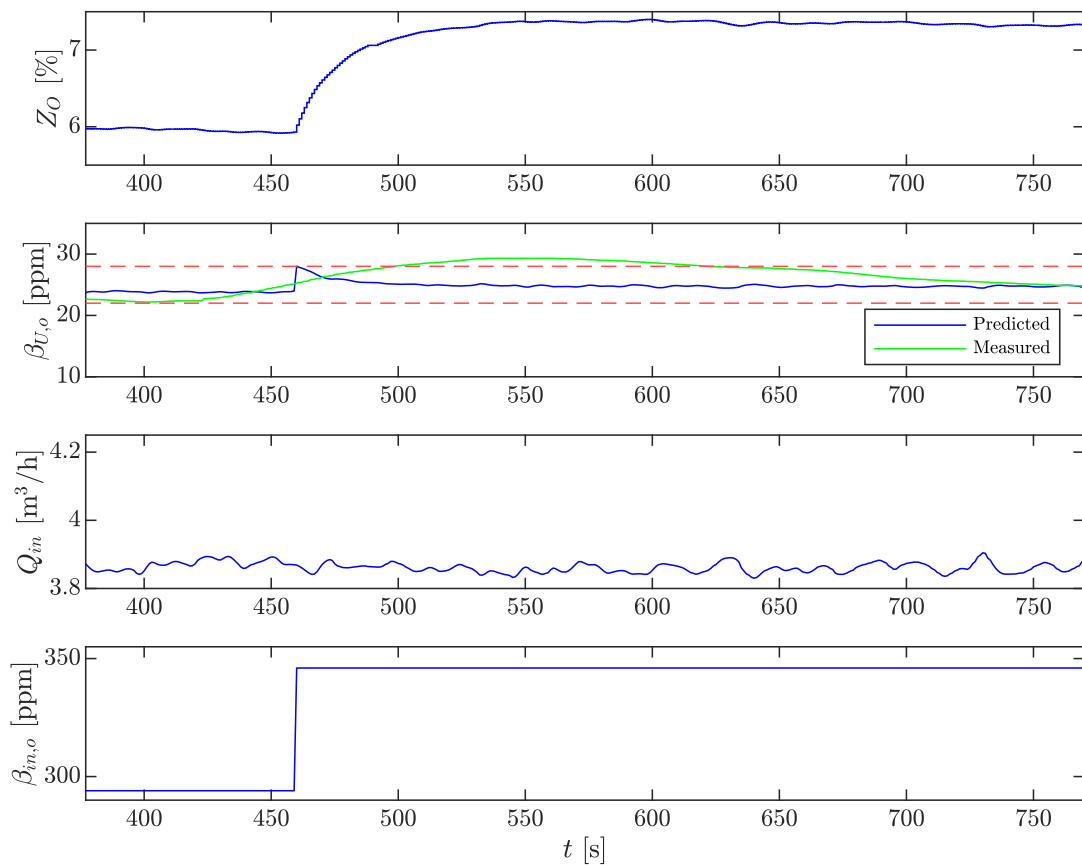


Figure 6.5: Experimental MPC results - increase in $\beta_{in,o}$

The increase in $\beta_{in,o}$ leads to an immediate increase in the predicted $\beta_{U,o}$ values, which is given by the blue line. The controller immediately acts by opening

Z_O . The increase in Z_O is very gradual, due to the constraint on Δu , and the relatively small changes that are necessary to achieve the reduction in predicted $\beta_{U,o}$ -values. The control action quickly leads to the predicted value returning to its starting point of approximately 25 ppm.

A similar response is observed in the measured output value, which is measured by the underflow OIW-sensor. However, the response is much slower. The actual concentration takes a lot longer to change, as can be seen from the gradual increase and decrease. As $\beta_{in,o}$ immediately changes, the predicted values immediately increase. In reality, there is some delay before the changes in inlet concentration occur, which explains the slower response of the measured value.

6.3.2 Change in Inlet Flow Rate

The experimental MPC was also tested for an increase in the inlet flow rate. Q_{in} was increased from approximately 3.98 to 4.18 m³/h, while $\beta_{in,o}$ was kept constant at an estimated 356 ppm. The same values as given in Table 6.3 were used, with the exception of the valve constant C_v , which was given a value of 0.015. The initial valve opening was different as well.

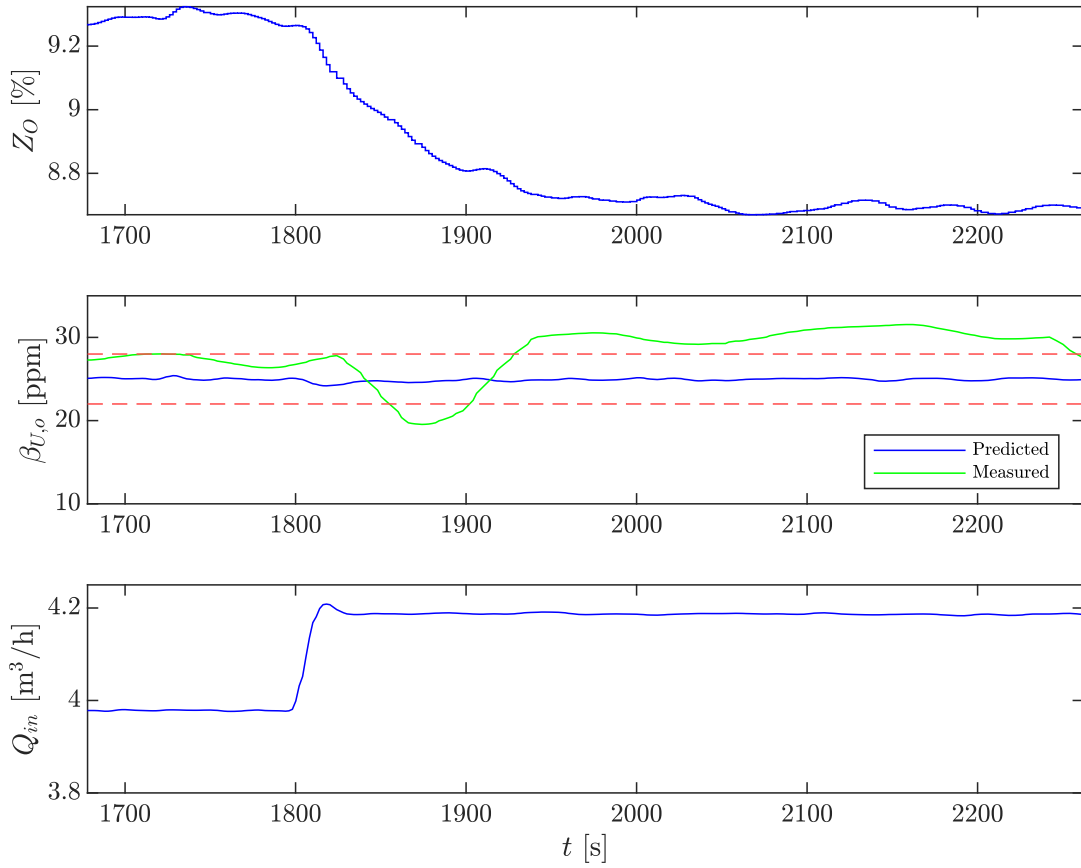


Figure 6.6: Experimental MPC results - increase in Q_{in}

The increase in Q_{in} led to a minor decrease in the predicted value of $\beta_{U,o}$, which is observed at $t \approx 1800$ s. The controller quickly acts by closing the valve, to keep the predicted value stable. The increased inlet flow rate also leads to a decrease in the measured value, albeit after some more time. The closing of Z_O eventually leads to $\beta_{U,o}$ increasing again. The measured value increases above where it was when it started, which is probably a sign of model mismatch.

Figure 6.6 illustrates that the MPC evidently does not handle changes in Q_{in} well. A probable reason is that the approximation of Q_{sep} was found while keeping Q_{in} . This could mean that the model works poorly when operating at other values of Q_{in} .

Chapter 7

Discussion

In this chapter, further analysis of the results is presented, along with discussion about the methods used and the assumptions made. Finally, we compare the experimental MPC results with some previously obtained results, from Vallabhan et al. (2021).^[16]

7.1 Polynomial Approximation

There are a few important points to consider related to the polynomial expression. As mentioned in the previous chapter, the data was gathered while keeping Q_{in} at a constant value. This means that the approximation of Q_{sep} may be inaccurate when operating at different values of Q_{in} . The inlet concentration was also assumed to be constant when gathering data for the polynomial. To improve the accuracy of the approximation, the polynomial could be expanded in a couple of different ways. One option is to add several polynomials, which are valid for different value ranges of the inlet concentration and flow rate. It was assumed that the internal separation $\frac{Q_{sep}}{Q_{in,o}}$ depends solely on Q_O . In reality, Q_U also affects the internal separation. Therefore, another way to improve the approximation of Q_{sep} is to expand the polynomial to also include Q_U .

Another point to consider is that the Q_O -values are not measured, but estimated from the mass balance of the hydrocyclone. The mass balance method is con-

sidered to give reasonably accurate estimates of Q_O , however it may have led to some inaccuracy of the approximation. Ideally, there would be a flowmeter measuring the Q_O stream as well.

In Figure 6.3, sudden peaks are observed in Q_O , for instance at $t \approx 6500$ s. The peaks occurred due to operating the sampling point SP-01, leading to momentary increases in Q_O . The sampling was done in order to validate the assumed inlet concentration using the offline Mastersizer sensor, since the inlet OIW-sensor did not give accurate readings.

One might notice that the experimental plots which have time on the x-axis do not start at $t = 0$. The reason for this is that some time is required for the experimental setup to stabilize. The starting time is the point at which the conditions in the laboratory were considered stable enough to perform the experiments.

7.2 Experimental MPC

There are several experimental conditions which should be considered, when evaluating the experimental MPC results. One frequently occurring problem was the instability of the underflow OIW-sensor. The sensor measurements are expected to oscillate somewhat, but at times the oscillation was too high. This lead to some difficulties in obtaining valid experimental results. There may be several reasons for the instability of this sensor. One could be that the sensor needs a good amount of time to stabilize. In industrial applications this is not a problem, as they would be continuously active. However, the CSL does not have the capacity to run continuously. The frequent restarting of the experimental rig may be a reason for why the sensor was not always giving accurate and stable measurements.

The inlet sensor was not used in any of the experiments. Instead a previously derived relationship between the oil pump speed and inlet oil concentration was used. This was considered necessary, as the inlet OIW-sensor proved to not accurately measure the oil concentration. However, there are some issues

related to using this approximation. If the relationship is inaccurate, that is an issue, since $\beta_{in,o}$ is an important part of the model and one of the considered disturbances. Another problem is that the actual increase in $\beta_{in,o}$ is gradual, as opposed to an immediate increase. The MPC would act differently if the values of $\beta_{in,o}$ were to increase gradually, as opposed to an immediate increase, which was the case in the performed experiments. An additional issue with this approximation is that it does not capture the natural fluctuations in $\beta_{in,o}$.

The experimental results indicate that the implemented MPC handles changes in the inlet oil concentration well, but not changes in the inlet flow rate. Ideally, more experiments should be done to verify these findings, preferably with the inlet OIW-sensor working properly. If more results could replicate these findings, we could be more confident in these two mentioned conclusions.

Q_O was estimated by the valve equation for the MPC, instead of the mass balance method, which was used for the polynomial approximation. The valve equation was used in order to include the Manipulated Variable, Z_O , in the model. An issue with this approach is that we need a value for the valve constant C_v . It was experimented with different values of C_v , according to what seemed to correspond with the mass balance estimated Q_O -values. The C_v was also adjusted according to what would make the model predict reasonably correct values of $\beta_{U,o}$.

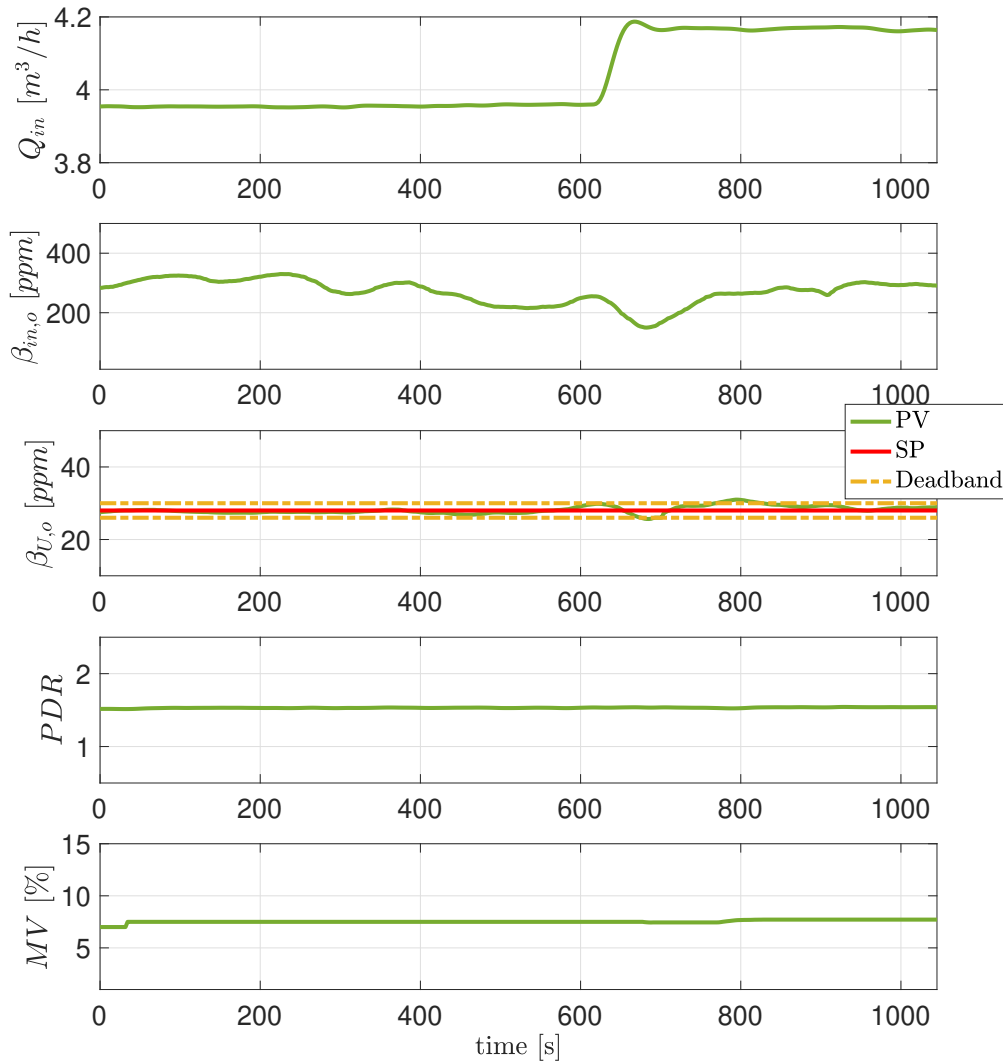
Another issue discovered during some of the experiments, was that the actual valve opening did not correspond exactly with the commanded value from LabVIEW. The values of Z_O in the plots are the commanded values, however, the actual values may have differed somewhat.

There were some notable differences between the simulated and experimental MPC results. The deviation may be explained by the model differences, specifically the polynomial for Q_{sep} and the number of hydrocyclone liners. Other influential differences include the values of Q_{in} , $\beta_{in,o}$ and Z_U .

7.3 Comparison with Direct Feedback Results

It is considered useful to compare the experimental MPC results with previously obtained results, which were gathered by using an alternative control method. The method we are comparing with is direct feedback control.

In this control method, a simple PI-controller with a dead-band was used, in order to directly control the hydrocyclone separation. Figure 7.1 shows the experimental results of direct feedback control when increasing Q_{in} .^[16]

Figure 7.1: Direct feedback control - increase in Q_{in}

The controller has a setpoint of 28 ppm, and a dead-band of ± 2 ppm. Figure 7.1 illustrates that this method appears to deal with an increase in Q_{in} more successfully than the MPC.

Figure 7.2 shows the response to an increase in $\beta_{in,o}$:^[16]

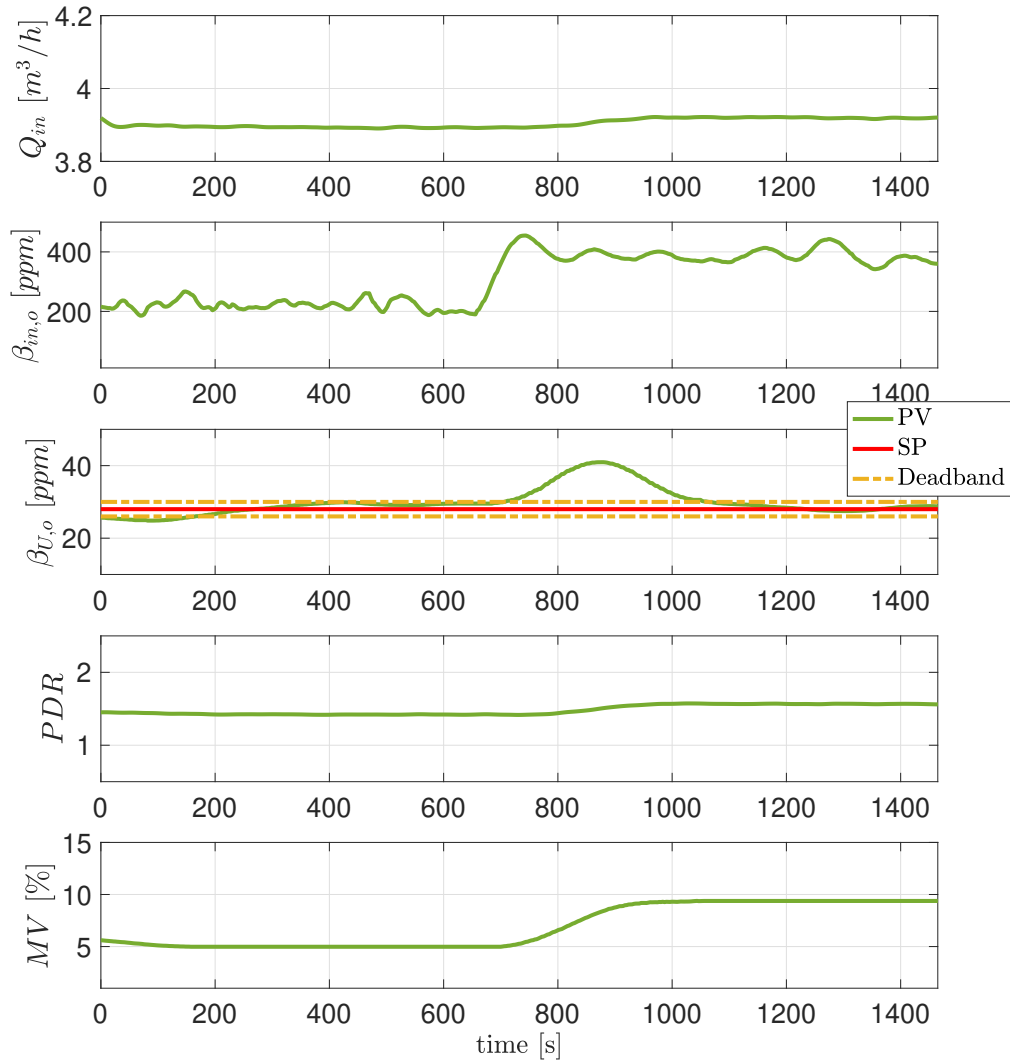


Figure 7.2: Direct feedback control - increase in $\beta_{in,o}$

Figure 7.2 illustrates that the controller also handles an increase in the inlet concentration well, similarly to the experimental MPC.

Since the direct feedback controller seemingly outperforms the MPC, it may be used as alternative control method for the hydrocyclone. However, there are a couple of disadvantages related to direct feedback control. First of all it is sensitive to measurement noise. As the measured values of $\beta_{in,o}$ illustrate, there appears to be a significant amount of measurement noise present. Therefore it is necessary to include a dead-band. The dead-band works by stopping the control action when the process value reaches within its values. The control valve then receives the previous value.

Chapter 8

Conclusion

The main aim of this project was to implement Model Predictive Control for a de-oiling hydrocyclone in the Compact Separator Laboratory (CSL).

A nonlinear Model Predictive Control was first implemented in simulation. The objective function used soft constraints to include both an upper and a lower limit on the oil concentration in the underflow. The implementation was done in MATLAB, by use of the open-source optimization software CasADi. The MPC uses direct multiple shooting to transform the control problem into a Non Linear Programming (NLP) problem, which is solved by the NLP solver IPOPT. The next step consisted of implementing MPC in the CSL. Before this step, a polynomial approximation of the internally separated oil was experimentally obtained. The experimental MPC was implemented in a manner similar to the simulated one, but included the additional step of creating a LabVIEW program, to communicate with both MATLAB and the CSL.

The MPC was tested for two disturbances, namely the inlet flow rate Q_{in} and inlet oil concentration $\beta_{in,o}$. As seen in Figure 6.5, the MPC appears to handle changes in $\beta_{in,o}$ well. Figure 6.6 showed that the MPC struggled to deal with a step in Q_{in} , which is likely mainly due to limitations of the 2nd degree polynomial of Q_{sep} .

8.1 Future Work

As mentioned, the MPC appears to handle changes in $\beta_{in,o}$ well, but not changes in Q_{in} . This is in contrast with PDR control, which does not work when changes in $\beta_{in,o}$ occur. Therefore, an interesting opportunity would be to combine MPC with a PDR controller. More specifically, it may act as a supervisory layer to the PDR controller. A suggested control scheme is illustrated in Figure 8.1:

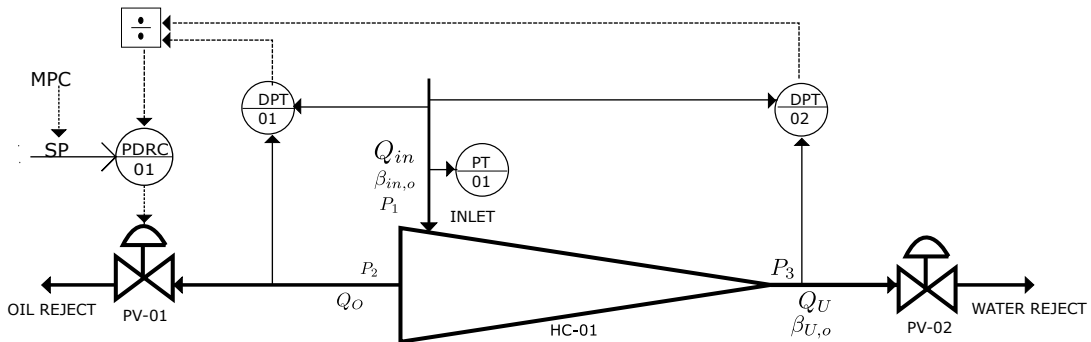


Figure 8.1: PDR control scheme with MPC as supervisory layer

It would be interesting to investigate how the MPC deals with a third disturbance, namely the droplet distribution. One could also consider an additional input, namely the underflow valve opening Z_U , which for this project has been kept constant.

There are several potential ways to improve the model. One of the main shortcomings of the model is the polynomial approximation of Q_{sep} , which is only valid for one value of Q_{in} . It could therefore be beneficial to expand the polynomial.

Lastly, a clear way to increase the MPC's performance, is to somehow improve the performance of the inlet OIW-sensor, to a point where it accurately measures $\beta_{in,o}$.

References

- [1] Fakhru'l-Razi Ahmadun et al. "Review of technologies for oil and gas produced water treatment". In: *Journal of Hazardous Materials* 170 (2009), pp. 530–551.
- [2] Ospar Convention. *Discharges*. Accessed: 2020-10-20. URL: <https://www.ospar.org/work-areas/oic/discharges>.
- [3] AB Sinker, M Humphris, and N Wayth. "Enhanced Deoiling Hydrocyclone Performance without Resorting to Chemicals". In: *Society of Petroleum Engineers* (1999).
- [4] Maurice Stewart and Ken E. Arnold. *Produced Water Treatment Field Manual*. Gulf Professional Publishing, 2011.
- [5] *SUBPRO Annual Report 2019-2020*. URL: <https://www.ntnu.edu/documents/1263635085/0/SUBPRO+Annual+Report+2019-2020.pdf/e1fd4df5-c008-4ec8-6341-9b9d0004bc66?t=1587985175506>.
- [6] Petar Durdevic et al. "Control Oriented Modeling of a De-oiling Hydrocyclone". In: *International Federation of Automatic Control* 48-28 (2015), pp. 291–296.
- [7] Tamal Das and Johannes Jäschke. "Modeling and control of an inline deoiling hydrocyclone". In: *International Federation of Automatic Control* 51-8 (2018), pp. 138–143.

REFERENCES

- [8] Mishiga Vallabhan K G, Christian Holden, and Sigurd Skogestad. “A First-Principles Approach for Control-Oriented Modeling of De-oiling Hydrocyclones”. In: *Industrial & Engineering Chemistry Research* 59.42 (2020), pp. 18937–18950. DOI: 10.1021/acs.iecr.0c02859. eprint: <https://doi.org/10.1021/acs.iecr.0c02859>. URL: <https://doi.org/10.1021/acs.iecr.0c02859>.
- [9] Mishiga Vallabhan and Christian Holden. “Non-linear control algorithms for de-oiling hydrocyclones”. In: *2020 28th Mediterranean Conference on Control and Automation (MED)*. 2020, pp. 85–90. DOI: 10.1109/MED48518.2020.9183115.
- [10] Mishiga Vallabhan et al. “Feedforward, cascade and model predictive control algorithms for de oiling hydrocyclones”. 2021.
- [11] *Hydrocyclones: A Solution to Produced Water Treatment*. Vol. All Days. OTC Offshore Technology Conference. OTC-5594-MS. Apr. 1987. DOI: 10.4043/5594-MS. eprint: <https://onepetro.org/OTCONF/proceedings-pdf/870TC/A11-870TC/OTC-5594-MS/2017665/otc-5594-ms.pdf>. URL: <https://doi.org/10.4043/5594-MS>.
- [12] M.L. Darby, M Harmse, and M Nikolaou. “MPC: Current Practice and Challenges”. In: *Control Engineering Practice* 20 (2012), pp. 328–342.
- [13] Amund Egeland Gaudernack. *Droplet Distribution and Overflow Estimation of a Hydrocyclone Laboratory*. Trondheim, Dec. 2020. URL: https://folk.ntnu.no/skoge/diplom/prosjekt20/gaudernack/TKP4850_SpecializationProject.pdf.
- [14] Magnus Hellem and Jens Djupvik. “Completion of Compact Separator Laboratory”. MA thesis. Trondheim: Norwegian University of Science and Technology, 2017.
- [15] Mishiga Vallabhan et al. “Experimental test setup for de-oiling hydrocyclones”. 2021.
- [16] Mishiga Vallabhan, Christian Holden, and Sigurd Skogestad. “De-oiling hydrocyclones: An experimental study of new control schemes”. 2021.

-
- [17] *Exxsol D60*. ExxonMobil. URL: https://www.exxonmobilchemical.com/en/library/library-detail/2652/exxsol_d60_fluid_product_safety_summary_en.
- [18] T Husveg et al. “Performance of a deoiling hydrocyclone during variable flow rates”. In: *Minerals Engineering* 20 (2007), pp. 368–379.
- [19] Mads Bram et al. “Control-Oriented Modeling and Experimental Validation of a Deoiling Hydrocyclone System”. In: *Processes* 8.9 (2020). URL: <https://www.mdpi.com/2227-9717/8/9/1010>.
- [20] D Solomatine and A Ostfeld. “Data-driven modelling: some past experiences and new approaches”. In: *Journal of Hydroinformatics* 10.1 (2008), pp. 3–22.
- [21] Shahab Araghinejad. *Data-Driven Modeling: Using MATLAB® in Water Resources and Environmental Engineering®*. Springer, 2014.
- [22] Matlab. *Polynomial curve fitting*. Accessed: 2021-05-26. URL: <https://se.mathworks.com/help/matlab/ref/polyfit.html>.
- [23] Matlab. *Fsolve*. Accessed: 2021-06-01. URL: <https://se.mathworks.com/help/optim/ug/fsolve.html>.
- [24] Dale E. Seborg, Duncan A. Mellichamp, and Thomas F. Edgar. *Process Dynamics and Control*. Third. Wylie Series in Chemical Engineering. John Wiley & Sons, 2011. ISBN: 9780470646106. URL: <http://www.worldcat.org/isbn/9780470646106>.
- [25] Wikimedia Commons. *File:MPC scheme basic.svg* — *Wikimedia Commons, the free media repository*. [Online; accessed 26-January-2021]. 2020. URL: https://commons.wikimedia.org/w/index.php?title=File:MPC_scheme_basic.svg&oldid=465567294.
- [26] Joel Andersson. “A General-Purpose Software Framework for Dynamic Optimization”. PhD thesis. Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium: Arenberg Doctoral School, KU Leuven, Oct. 2013.

REFERENCES

- [27] R Core Team. *User Documentation for CasADi v3.3.0-194.a1d1a5d78*. R Foundation for Statistical Computing. Vienna, Austria, 2018.
- [28] *CasADi Github*. URL: <https://github.com/casadi/casadi/tree/master/docs/examples/matlab>.
- [29] Dale B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design 2nd Edition*. Nob Hill Publishing, 2019.
- [30] H.G. Bock and K.J. Plitt. “A MULTIPLE SHOOTING ALGORITHM FOR DIRECT SOLUTION OF OPTIMAL CONTROL PROBLEMS”. In: (1984).
- [31] Jasem Tamimi and Pu Li. “Nonlinear Model Predictive Control Using Multiple Shooting Combined with Collocation on Finite Elements”. In: *IFAC Proceedings Volumes 42.11 (2009)*. 7th IFAC Symposium on Advanced Control of Chemical Processes, pp. 703–708. ISSN: 1474-6670. DOI: <https://doi.org/10.3182/20090712-4-TR-2008.00114>. URL: <https://www.sciencedirect.com/science/article/pii/S1474667015303578>.
- [32] Y Kawajir et al. *Introduction to Ipopt: A tutorial for downloading, installing, and using Ipopt*. Accessed: 2021-04-30. URL: <https://projects.coin-or.org/Ipopt/browser/stable/3.11/Ipopt/doc/documentation.pdf?format=raw>.
- [33] *LabVIEW User Manual*. LabVIEW. URL: <https://www.ni.com/pdf/manuals/320999e.pdf>.
- [34] Matlab. *awgn*. Accessed: 2021-06-03. URL: <https://se.mathworks.com/help/comm/ref/awgn.html>.

Appendix A

Matlab Code

This chapter contains the Matlab code that has been used in this Master's thesis. It is further divided into the code used for the simulated MPC, the experimental MPC and the polynomial fitting.

A.1 Simulated MPC

Used to do the simulations as shown in Figure 6.1 and 6.2

A.1.1 Main

```
% Main simulated MPC file - nonlinear MPC (Casadi)

clc
close all
addpath('C:\Users\amund\OneDrive\Skrivebord\NTNU\
        TKP4555\Matlab\casadi-windows-matlabR2016a-v3.5.3')
import casadi.*

par = initHC();

T = 60;      % Time horizon
```

APPENDIX A. MATLAB CODE

```
N = 60;      % Sampling intervals
dt = T/N;

%% Initial conditions

x1 = 0.03;      % Overflow concentration - initial
               value
x2 = 25E-6;     % Underflow concentration - initial
               value
x0 = [x1;x2];

u0 = 0.35;     % Initial value of the MV (Zo)

% Initial pressure values

P2 = 3*10^5;
P3 = 4*10^5;

%% Time (for plotting)

t = linspace(0, T, N);

%% Initializing values of u and x

xi = zeros(2,N);
ui = zeros(1,N);

%% The first initial values will be given the initial
   condition

xi(:,1) = x0;
ui(1) = u0;
```

```
%% Simulated values for in-flow and concentration (d1
& d2)

Qin1 = [0.0006.*ones(1,N/4) 0.0006.*ones(1,N/4)
        0.0006.*ones(1,N/4) 0.0007.*ones(1,N/4)];
Betain1 = [0.0005.*ones(1,N/2),0.0009.*ones(1,N/2)];
Qin = awgn(Qin1, 110);
Betain = awgn(Betain1, 110);

xSP = 30E-6;

Q = 1E6;
R = 0.1;

%% Loop

elapsedTime = [];
for i = 2:N
    disp('Iteration number:')
    disp(i-1)
    PQ = PressFlowRelation(0.5,ui(i-1),P2,P3,Qin(i),
        par);
    P2 = PQ(:,3);
    P3 = PQ(:,4);
    xi(:,i) = ModelNMPC(xi(:,i-1), ui(i-1), Qin(i),
        Betain(i), dt, P2, P3);
    tic
    [ui(i)] = SimulatedMultipleShooting(xi(:,i), ui(i-1),
        Qin(i), Betain(i), dt, Q, R, xSP, P2, P3);
    elapsedTime = [elapsedTime;toc];
end
```

```
%% Plotting results

set(groot, 'defaultTextInterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');

xL = 20E-6;

figure('visible', 'on')
subplot(4,1,1)
stairs(t, ui.*100, 'b')
% title('MV - Overflow Valve Opening')
ylim([30 50])
ylabel('$Z_0$ [%]')

subplot(4,1,2)
plot(t, xi(2,:).*1e6, 'b')
hold on
yline(xSP*1e6, '--r')
hold on
yline(xL*1e6, '--r')
% title('CV - Oil Concentration Underflow ($x_2$)')
ylim([0 50])
ylabel('$\beta_{U,o}$ [ppm]')

subplot(4,1,3)
plot(t, Qin.*3600, 'b')
ylim([2.0 2.6])
% title('D1 - Flowrate Inlet')
ylabel('$Q_{in}$ [m$^3$/h]')
```

```
subplot(4,1,4)
plot(t, Betain.*1e6, 'b')
% title('D2 - Oil Concentration Inlet')
xlabel('$t$ [s]')
ylim([400 1000])
ylabel('$\beta_{in,o}$ [ppm]')

% sgtitle('Simulated MPC Results', 'FontSize', 12)

% saveas(gcf,[pwd '/Figures/SimulatedMPC_2'],'eps')
```

A.1.2 Model

```
%% Model of the plant

function xend = ModelNMPC(x0,D,Qin,Betain,dt, P2, P3)

addpath('C:\Users\subsea\Desktop\Amund\casadi-windows-
matlabR2016a-v3.5.5')
import casadi.*

%% Parameters

par = initHC();
Vo = par.VRF();
Vf = par.Vol_HC-par.VRF;
Zu = 0.5;          % Constant underflow valve opening

%% Model variables

x1 = SX.sym('x1');
x2 = SX.sym('x2');
x = [x1;x2];
```



```
%% Model inputs

u = SX.sym('u'); % Overflow valve opening

%% Model equations

cv1 = pi*0.005^2;
cv2 = pi*0.001^2;

Qu = cv1*Zu*sqrt(2/par.Rho_w*(P3-par.Patm));
Qo = cv2*u*sqrt(2/par.Rho_o*(P2-par.Patm));

% Qsep polynomial approximation

p2 = -9.447e+07;
p1 = 9024;
p0 = 0.7648;

% Oil inflow
Qino = Betain.*Qin;
Qsep = Qino.*(p2*Qo.^2+p1*Qo+p0);

%% Model equations

xdot1 = (1/Vo)*(Qsep-x1*Qo);
xdot2 = (1/Vf)*(Qino-Qsep-x2*Qu);

xdot = [xdot1;xdot2];

ode = struct('x',x,'p',u,'ode',xdot);
```

```
%% Integrator

opts = struct('tf',dt);
F = integrator('F', 'cvodes', ode, opts);
sim = F('x0',x0,'p',D);

xend = full(sim.xf);
end
```

A.1.3 Controller (Multiple Shooting)

```
function u_opt = SimulatedMultipleShooting(x0,u0,Qin,
    Betain,h,Q,R,xSP,P2,P3)

import casadi.*

%% Model variables

x1 = SX.sym('x1');    %fraction of oil in overflow
x2 = SX.sym('x2');    %fraction of oil in underflow

x = [x1;x2];
nx = 2;

%% System inputs

u = SX.sym('D'); % $[h^{-1}]$  - Dilution Rate:  $F/V$ 
du = SX.sym('du'); %input variation
s = SX.sym('s');

%% System constraints

u_min = 0;
```

```
u_max = 1;

du_max = 0.05;
du0 = 0;

s_min = 0;
s_max = inf;
s0 = 0;

% control horizon
mh = 10;
% prediction horizon
ph = 15;

%% Parameters
par = initHC();
Vo = par.VRF();
Vf = par.Vol_HC-par.VRF;
Zu = 0.5;          % Constant underflow valve opening

%% Model variables

x1 = SX.sym('x1');
x2 = SX.sym('x2');
x = [x1;x2];

%% Model inputs
u = SX.sym('u'); % Overflow valve opening

cv1 = pi*0.005^2;
cv2 = pi*0.001^2;
```

```
%% Model equations

Qu = cv1*Zu*sqrt(2/par.Rho_w*(P3-par.Patm));
Qo = cv2*u*sqrt(2/par.Rho_o*(P2-par.Patm));

% Qsep polynomial approximation
p2 = -9.447e+07;
p1 = 9024;
p0 = 0.7648;

% Oil inflow
Qino = Betain.*Qin;
Qsep = Qino.*(p2*Qo.^2+p1*Qo+p0);

xdot1 = (1/Vo)*(Qsep-x1*Qo);
xdot2 = (1/Vf)*(Qino-Qsep-x2*Qu);

xdot = [xdot1;xdot2];

% Objective term
L = Q*s^2 + R*du^2; % Soft Constraints

Uu=[u;du;s];

% Formulate discrete time dynamics
% CVODES from the SUNDIALS suite

ode = struct('x',x,'p',Uu,'ode',xdot,'quad',L);

deltaX = 10E-6;
```

```
%% Integrator

opts = struct('tf',h);
F = integrator('F', 'cvodes', ode, opts);

% preparing symbolic variable

w={};

% preparing numeric variables and bounds
w0 = [];
lbw = [];
ubw = [];

g={};
lbg = [];
ubg = [];
J = 0;

% Formulate the NLP
% initial state as variable

Xk = MX.sym('X0', nx);
w = {w{:}, Xk};
lbw = [lbw; x0(1); x0(2)];
ubw = [ubw; x0(1); x0(2)];
w0 = [w0; x0(1); x0(2)];

%loop for discretized dynamic
for k=0:ph-1

    % variable for the control in this control
```

```
    interval
    Uk = MX.sym(['U_' num2str(k)]);
    DUk = MX.sym(['DU_' num2str(k)]);
    Sk = MX.sym(['S_' num2str(k)]);
    Uk1=[Uk;DUk;Sk];
    w = {w{:}, Uk1};

% bounds
lbw = [lbw; u_min; -du_max; s_min];
ubw = [ubw; u_max; du_max; s_max];

%initial
w0 = [w0;u0;du0;s0];

if k <= mh
    if k == 0
        DUk = Uk - u0;
    else
        DUk = Uk(1) - Uk_1;
    end

    % Add input movement constraint
    g = {g{:}, DUk};
    lbg = [lbg; -du_max];
    ubg = [ubg; du_max];

else
    Uk(1) = Uk_1;
    DUk = 0;
end
```

```
% Integrate till the end of the interval

Fk = F('x0',Xk,'p',Uk1);

J = J + Fk.qf;

Pk = Fk.xf;

Xk = MX.sym(['X_' num2str(k+1)], nx);

w = {w{:}, Xk};
lbw = [lbw; 0; 0];
ubw = [ubw; 1; 1];
w0 = [w0; x0(1); x0(2)];

%calculating delta U

if k ~= ph - 1
    % for computing du
    Uk_1 = Uk1(1);
end
% Add continuity constraint (closing shooting
gap)
g = {g{:}, Pk-Xk};
lbg = [lbg; zeros(nx,1)];
ubg = [ubg; zeros(nx,1)];

% Soft constraint
```

```

    g = {g{:}, (Xk(2)-Sk-xSP)};
    lbg = [lbg; -deltaX];
    ubg = [ubg; 0];

end

% Create an NLP solver
prob = struct('f', J, 'x', vertcat(w{:}), 'g', vertcat
    (g{:}));
solver = nlpsol('solver', 'ipopt', prob);

% Solve the NLP
sol = solver('x0', w0, 'lbx', lbw, 'ubx', ubw, 'lbg',
    lbg, 'ubg', ubg);
w_opt = full(sol.x);

%first input
u_opt = w_opt(nx + 1);

end

```

A.1.4 Parameters

```

function [par] = initHC()
par.Patm=101325; %Atmosphere pressure
par.Rho_in = 989;
par.Rho_o = 910;%850; %
    density of oil kg/m3
par.Rho_w = 1000; %density
    of oil kg/m3
par.D = [5e-6;12e-6;13e-6;14e-6;15e-6;16e-6;20e-6;25e
    -6;30e-6;35e-6;40e-6;45e-6;50e-6;55e-6;60e-6];
par.Cd = 20; % 1; This is the value of Cd for sphere

```



```
when the Reynolds number is greater
      %than 10 and less than 100 where the
      separated flow
      %starts Reference plot in 'E. Loth Drag of
      non-spherical
      %solid particles of regular and irregular
      shape'For laminar flow
par.vis = 10^-6; %kinematic viscosity
par.Rov = 0.001; %
      overflow radius
par.Rin = 0.0035; %par.Rin = 0.0018;
      % [m] % inlet
      radius
par.R1 = 0.02; % [m]
      Cylindrical radius
par.L1 = 0.04; % [m]
      Cylindrical length
par.R2 = 0.01; % [m]
      first cone radius
par.alpha = 0.1745 ; %
      first cone angle in radians
par.R1Half = sqrt(par.R1^2/2);
par.L2 = 0.0567;%(par.R1-par.R2)/tan (par.alpha);
par.R3 = 0.005; % [m]
      second cone radius
par.beta = 0.0131 ; %
      second cone angle in radians
par.R2Half = sqrt(par.R2^2/2);
par.L3 = 0.3820;%(par.R2-par.R3)/tan (par.beta);
par.Rfac=0.3718;
par.Rfac1=0.27;
par.L4=0.6;
```

```

par.Ru = 0.0127;
par.Ro = 0.00635;
par.Aup=pi*0.0127^2; % 1 inch pipe at the underflow
par.Aop=pi*0.00635^2; % 1/2 inch pipe at the overflow
par.Cdu=0.1;
par.Cdo=0.02;
par.Vol_HC= (pi*((par.R1)^2)*par.L1)+(pi*(1/3)*((par.
    R1)^2+(par.R2)^2+(par.R1)*(par.R2))*(par.L2))+ (pi
    *(1/3)*((par.R2)^2+(par.R3)^2+(par.R2)*(par.R3))*(
    par.L3)+(pi*(par.R3)^2*(par.L4));
par.VRF=(par.Vol_HC*(0.001/par.R1)^2);% Volume of
    reverse flow oil core -----VolumeofHC*(par.Rov/par.
    R1)^2
par.K1=1/par.VRF;
par.K2=1/(par.Vol_HC-par.VRF);

```

A.1.5 Pressure-Flow Relationship

```

function Q_Flow = PressFlowRelation(xu,xo,P2,P3,Qin,
    par)
x0 = [P2,P3];
if xo<0.1
    xo=0.05;
end

Cv1=par.Aup*par.Cdu;%Pipe dia underflow
Cv2=par.Aop*par.Cdo;%pipe dia overflow

%% Qin as a sum of Qo and Qu
%options = optimoptions('fsolve','Display','off');

options = optimoptions('fsolve','Display','off','
    OptimalityTolerance',1e-10,'FiniteDifferenceType','

```

```
    central');  
x = fsolve(@(x) PressEquations_withQin(Qin,xu,xo,par,x  
    ),x0,options);  
Qu=abs(Cv1*xu*sqrt(2/par.Rho_w*(x(2)-par.Patm)));  
Qo=abs(Cv2*xo*sqrt(2/par.Rho_o*(x(1)-par.Patm)));  
P2=abs(x(1));  
P3=abs(x(2));  
Q_Flow =[Qu Qo P2 P3];
```

A.1.6 Pressure Equations

```
function F = PressEquations_withQin(Qin,Zu,Zo,par,x)  
Ain=pi*par.Rin^2;  
Cv1=par.Aup*par.Cdu;  
Cv2=par.Aop*par.Cdo;  
Au=pi*par.R3^2;  
Ao=pi*par.Rov^2;  
  
alpha1=0.175;  
%% INLET PRESSURE%%  
  
% F(1)=P1+(K7)*(K1*2*(x(2)-par.Patm)+K2*2*(x(1)-par.  
    Patm)+K3*sqrt(2*(x(2)-par.Patm)/par.Rho_w)*sqrt(2*(x  
    (1)-par.Patm)/par.Rho_o))-x(1)*(1+K6)+K6*par.Patm;  
% F(2)=P1+(K4)*(K1*2*(x(2)-par.Patm)+K2*2*(x(1)-par.  
    Patm)+K3*sqrt(2*(x(2)-par.Patm)/par.Rho_w)*sqrt(2*(x  
    (1)-par.Patm)/par.Rho_o))-x(2)*(1+K5)+K5*par.Patm;  
  
K_1=(5*par.Rho_w*(alpha1*Qin*par.R1)^2)/(4*(par.R3*pi*  
    par.Rin^2)^2);  
K_2=(par.Rho_o*(alpha1*Qin*par.R1)^2*par.Rov^2)/(4*((  
    par.R3*par.Rfac1)^2*pi*par.Rin^2)^2);  
F(1)=x(2)-x(1)+((1/(2*Au^2))*(Cv1*Zu)^2*((2*(x(2)-par.
```

```
Patm)))) - ((1/(2*Ao^2))*(Cv2*Zo)^2*((2*(x(1)-par.Patm
))) + K_1 - K_2;
F(2) = Qin - (Cv1*Zu*sqrt(2*(x(2)-par.Patm)/par.Rho_w)) - (
Cv2*Zo*sqrt(2*(x(1)-par.Patm)/par.Rho_o));
```

end

A.2 Experimental MPC

A.2.1 Main

```
%% Labview test

addpath('C:\Users\subsea\Desktop\Amund\casadi-windows-
        matlabR2016a-v3.5.5')
import casadi.*

[timeElapsed, uOpt, obj, P, M, iterations, success,
  exitFlag, x1Horizon, firstx2, lastx2] =
  MultipleShootingLabview([Beta0;BetaU], Zo, Qin,
  Betain, dt, Q, R, x2SP, Qu, u_min, u_max, du_max, mh
  , ph, deltaX, P2, valveConstant);
```

A.2.2 Controller

```
function [timeElapsed, uOpt, obj, P, M, iterations,
  success, exitFlag, x1Horizon, firstx2, lastx2] =
  MultipleShootingLabview(x0, u0, Qin, Betain, h, Q, R
  , x2SP, Qu, u_min, u_max, du_max, mh, ph, deltaX, P2
  , valveConstant)

tic % Starting timer
import casadi.*

%% Model variables

x1 = SX.sym('x1'); % Overflow oil fraction, Beta0
x2 = SX.sym('x2'); % Underflow oil fraction, BetaU
x = [x1;x2];
nx = 2; % Number of states
```

```
%% System inputs

u = SX.sym('D');      % Input
du = SX.sym('du');    % Input change
s = SX.sym('s');      % Slack

%% System constraints

du0 = 0;

s_min = 0;
s_max = inf;
s0 = 0;

%% Model Equations

par = initHC();
Vo = par.VRF();      %
Vf = par.Vol_HC-par.VRF;
Rho0 = 810;         % kg/m3

%% Model variables

x1 = SX.sym('x1');
x2 = SX.sym('x2');
x = [x1;x2];

%% Model inputs

u = SX.sym('u'); % Overflow valve opening

%% Experimental polynomial for Qsep
```

```
p2 = -5.5455 ;% (-1.322e+08, -5.671e+07)
p1 = 2.205;% (6748, 1.13e+04)
p0 = 0.7209;% (0.7378, 0.7919)

Qo = valveConstant*pi*0.0127^2*u*sqrt((2/Rho0)*(P2-par
.Patm));

% Oil inflow
Qino = Betain.*Qin/2;

%% Polynomial is valid for [m3/h], therefore
   converting the units

QoPerHour = Qo*3600;
QinoPerHour = Qino*3600;

QsepPerHour = QinoPerHour.*(p2*QoPerHour.^2+p1*
QoPerHour+p0);

Qsep = QsepPerHour/3600;

%% Model equations

xdot1 = (1/Vo)*(Qsep-x1*Qo);
xdot2 = (1/Vf)*(Qino-Qsep-x2*Qu);

xdot = [xdot1;xdot2];

% Objective term
L = Q*s^2 + R*du^2;
```

```
Uu=[u;du;s];

ode = struct('x',x,'p',Uu,'ode',xdot,'quad',L);

%% Integrator

opts = struct('tf',h);
F = integrator('F', 'cvodes', ode, opts);

% preparing symbolic variable

w={};

preparing numeric variables and bounds
w0 = [];
lbw = [];
ubw = [];

g={};
lbg = [];
ubg = [];
J = 0;

% Formulate the NLP
%initial state as variable
Xk = MX.sym('X0', nx);
w = {w{:}, Xk};
lbw = [lbw; x0(1); x0(2)];
ubw = [ubw; x0(1); x0(2)];
w0 = [w0; x0(1); x0(2)];

for k=0:ph-1
```



```
% variable for the control in this control
interval
Uk = MX.sym(['U_' num2str(k)]);
DUk = MX.sym(['DU_' num2str(k)]);
Sk = MX.sym(['S_' num2str(k)]);
Uk1 = [Uk; DUk; Sk];
w = {w{:}, Uk1};

lbw = [lbw; u_min; -du_max; s_min];
ubw = [ubw; u_max; du_max; s_max];

w0 = [w0; u0; du0; s0];

if k <= mh
    if k == 0
        DUk = Uk - u0;
    else
        DUk = Uk(1) - Uk_1;
    end

    % Add input movement constraint
    g = {g{:}, DUk};
    lbg = [lbw; -du_max];
    ubg = [ubg; du_max];

else
    Uk(1) = Uk_1;
    DUk = 0;
end

% Integrate till the end of the interval
```

```
Fk = F('x0',Xk,'p',Uk1);

J = J + Fk.qf;

%loop
Pk = Fk.xf;

% States variables for state at end of interval
% create states

Xk = MX.sym(['X_' num2str(k+1)], nx);

w = {w{:}, Xk};
lbw = [lbw; 0; 0];
ubw = [ubw; 1; 1];
w0 = [w0; x0(1); x0(2)];

%calculating delta U

if k ~ = ph - 1
    % for computing du
    Uk_1 = Uk1(1);
end
% Add continuity constraint (closing shooting
gap)
g = {g{:}, Pk-Xk};
lbg = [lbg; zeros(nx,1)];
ubg = [ubg; zeros(nx,1)];
```

```
% Soft constraint
g = {g{:}, (Xk(2)-Sk-x2SP)};
lbg = [lbg; -deltaX];
ubg = [ubg; 0];

end

%% Create an NLP solver

prob = struct('f', J, 'x', vertcat(w{:}), 'g', vertcat
    (g{:}));

solver = nlpso1('solver', 'ipopt', prob);

%% Solve the NLP

sol = solver('x0', w0, 'lbx', lbw, 'ubx', ubw, 'lbg',
    lbg, 'ubg', ubg);
w_opt = full(sol.x);

%% Various outputs

uOpt = w_opt(nx + 1);

obj = full(sol.f);

x1Horizon = w_opt(1:5:end);

P = w_opt(2:5:end);    % x2

M = w_opt(3:5:end);    % Control inputs
```

```
stats = solver.stats();    % information

iterations = stats.iter_count; % Number of iterations
    performed

success = stats.success;    % 1 = success 0 = fail

exitFlag = stats.return_status();

QoValve = full(Qo);

firstx2 = P(2);

lastx2 = P(11);

timeElapsed = toc; % ending timer

end
```

A.3 Polynomial Fitting

```
%% Polynomial fitting %%

clc
close all

%% Data

T = readtable('ExpData/24.3.21.xlsx');

start = 1500;    % Values before this measurement are
                 discarded
Qo = T{start:2150,2};
betaU = T{start:2150,3};
betaIn = 300;
betaIn2 = 420;
Zo = T{start:2150,4};

%% Plotting experimental results

set(groot, 'defaultTextInterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');

%% Data 2

T2 = readtable('ExpData/26.3.21.xlsx');

start2 = 810;
Qo2 = T2{start2:1650,2};
betaU2 = T2{start2:1650,3};
```

```
Zo2 = T2{start2:1650,4};

%% DataCombined (For Plotting)

QoCom = [Qo; Qo2];
betaUCom = [betaU; betaU2];
ZoCom = [Zo; Zo2];

%% Time (for plotting)

time1 = T{start:2150,6};
time2 = T2{start2:1650,6};
time2 = T2{start2:1650,6}+max(time1)-min(time2);
time = [time1;time2];

%% Plotting experimental results COMBINED

figure('visible','on');

subplot(3,1,1)
plot(time, QoCom, 'b')
ax=gca;
ax.XAxis.Exponent=0;
xlim([min(time), max(time)])
ylabel('$Q_0$ [m3/h]')
% title('Overflow Flowrate')

subplot(3,1,2)
plot(time, betaUCom, 'b')
ax=gca;
ax.XAxis.Exponent=0;
```

```
xlim([min(time), max(time)])
ylabel('$\beta_U$ [ppm]')
% title('Underflow Oil Concentration')

subplot(3,1,3)
plot(time, ZoCom, 'b')
ax=gca;
ax.XAxis.Exponent=0;
xlim([min(time), max(time)])
ylim([0 25])
xlabel('$t$ [s]')
ylabel('$Z_0$ [%]')
% title('Overflow Valve Opening')

% sgtitle('Experimental Data - Polynomial Fitting')

saveas(gcf, [pwd '/Figures/ExperimentalDataPolynomial'
], 'eps')

%% Calculating average values

QoAvg1 = mean([Qo(1:67)]);
QoAvg2 = mean([Qo(126:177)]);
QoAvg3 = mean([Qo(197:264)]);
QoAvg4 = mean([Qo(312:431)]);
QoAvg5 = mean([Qo(466:540)]);

QoAvgList = [QoAvg1 QoAvg2 QoAvg3 QoAvg4 QoAvg5];

% Discarded values:
% QoAvg6 = mean([Qo(677:762)]);
% QoAvg7 = mean([Qo(803:850)]);
```

```
% QoAvg8 = mean([Qo(893:906)]);
% QoAvg9 = mean([Qo(944:1018)]);
% QoAvg10 = mean([Qo(1044:1068)]);
%
% QoAvgList2 = [QoAvg6 QoAvg7 QoAvg8 QoAvg9 QoAvg10];

betaUAvg1 = mean([betaU(1:67)]);
betaUAvg2 = mean([betaU(127:175)]);
betaUAvg3 = mean([betaU(214:249)]);
betaUAvg4 = mean([betaU(364:400)]);
betaUAvg5 = mean([betaU(433:535)]);

betaUAvgList = [betaUAvg1 betaUAvg2 betaUAvg3
    betaUAvg4 betaUAvg5];

% Discarded values:
%
% betaUAvg6 = mean([betaU(677:762)]);
% betaUAvg7 = mean([betaU(803:850)]);
% betaUAvg8 = mean([betaU(893:906)]);
% betaUAvg9 = mean([betaU(944:1018)]);
% betaUAvg10 = mean([betaU(1044:1068)]);
%
% betaUAvgList2 = [betaUAvg6 betaUAvg7 betaUAvg8
    betaUAvg9 betaUAvg10];

%% Calculating average values - Dataset 2

QoAvg11 = mean([Qo2(1:100)]);
QoAvg12 = mean([Qo2(110:257)]);
QoAvg13 = mean([Qo2(285:440)]);
QoAvg14 = mean([Qo2(468:662)]);
```


APPENDIX A. MATLAB CODE

```
QoAvg15 = mean([Qo2(690:837)]);

QoAvgListData2 = [QoAvg11 QoAvg12 QoAvg13 QoAvg14
    QoAvg15];

betaUAvg11 = mean([betaU2(1:100)]);
betaUAvg12 = mean([betaU2(110:257)]);
betaUAvg13 = mean([betaU2(285:440)]);
betaUAvg14 = mean([betaU2(468:662)]);
betaUAvg15 = mean([betaU2(690:837)]);

betaUListData2 = [betaUAvg11 betaUAvg12 betaUAvg13
    betaUAvg14 betaUAvg15];

%% Calculating separation efficiency

etaList = [];
for i = 1:length(betaUAvgList)
    eta = 1 - (betaUAvgList(i)/betaIn);
    etaList = [etaList, eta];
end

% etaList2 = [];
% for i = 1:length(betaUAvgList2)
%     eta2 = 1 - (betaUAvgList2(i)/betaIn2);
%     etaList2 = [etaList2, eta2];
% end

etaListData2 = [];
for i = 1:length(betaUListData2)
    eta3 = 1 - (betaUListData2(i)/betaIn); % Betain
```

```
        is 300 for this aswell
    etaListData2 = [etaListData2, eta3];
end

%% Polynomial fitting

QoAvgList3 = sort([QoAvgList QoAvgListData2]);

etaList3 = sort([etaList etaListData2]);

p2 = polyfit(QoAvgList3, etaList3, 2); % 2nd degree
    polynomial

xlist = linspace(min(QoAvgList3), max(QoAvgList3),
    10000);

pol2 = polyval(p2,xlist);

p3 = polyfit(QoAvgList3, etaList3, 3); % 3rd degree
    polynomial

pol3 = polyval(p3,xlist); %

%% Plotting

figure
scatter(QoAvgList3, etaList3, 'filled', 'r')
hold on
plot(xlist, pol2, 'b')
```

```
xlabel('$Q_0$ [m3/h]')
ylabel('$\eta$ [-]')
legend('Exp data','Polynomial')
legend('Location', 'northwest')
% title('Polynomial Expression')
saveas(gcf, [pwd '/Figures/PolynomialApproximation'],'
    epsc')
```

Appendix B

LabVIEW

Untitled 1
Last modified on 29.05.2021 at 15.31
Printed on 29.05.2021 at 15.32

Page 6 

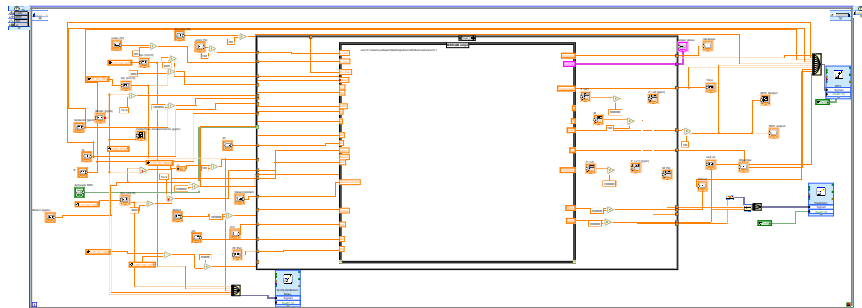


Figure B.1: LabVIEW - MPC block diagram

Appendix C

Sampling Procedure


The procedure for operating the sampling points in the CSL.^[14]

1. Check that the sampling bomb is drained and de-pressurized.
2. Open inlet valve until the pressure in the sampling bomb matches the process pressure, then close inlet valve.
3. Open vent valve until the pressure in the sampling bomb is at atmospheric level.
4. Extract sample from sample valve
5. Fully drain the sampling bomb, then close sample valve and vent valve.

Appendix D

Mastersizer Procedure

The following is a procedure for operating the offline Mastersizer sensor; developed by Anders Andersen and Marcin Dudek.

	PROCEDURE	Last revision : 12.09.2018
	Title: Mastersizer 3000	Page 1 of 6
Faculty of Natural Sciences and Technology	Procedure number: 2-0669-01-02	
Department: Chemical Engineering	Developed by: Anders Andersen, Marcin Dudek	
Revision number: 2	Approved by:	
	Maintained by:	


REVISION STATUS :			
DOCUMENT STATUS : CONTROLLED			
REV No. – DATE	REVISION made by	CHECKER	APPROVER
0 – 20.07.2016	Anders Andersen, Marcin Dudek		
1 – 25.11.2016	Marcin Dudek		
2 – 12.09.2018	Marcin Dudek		
This document is subjected to AUDIT.			

ORIGINATOR: Anders Andersen, Marcin Dudek

REVISION MADE BY: Marcin Dudek

APPROVER: Camilla I. Dagsgård – Laboratory manager at UL

REVISION HISTORY	
DATE	AMENDMENT DESCRIPTION
25.11.2016	Added instructions for Hydro SV accessory
12.09.2018	Added notes on obscuration

 NTNU Faculty of Natural Sciences and Technology	Procedure: Mastersizer 3000	
Procedure number: 2-0669-01-02	Revision no.: 2	Page 2 of 6

1. PURPOSE

Mastersizer 3000 (Malvern Instruments) is used for quick and accurate particle size distribution analysis. It can work with both emulsions and dispersions.

2. SCOPE

Mastersizer 3000 uses laser diffraction for measuring the size of particles or droplets in a dispersion (from 0,01 to 3500 μm), by measuring the intensity of scattered light from those particles in a continuous phase. Data is then processed and presented as a size distribution.

3. RESPONSIBILITIES

The person responsible for the instrument is also responsible for updating this procedure.

4. DEFINITIONS AND ABBREVIATIONS


SOP – Standard operating procedure

5. EQUIPMENT

Mastersizer 3000
Hydro EV accessory
Beakers (400 – 1000 ml)
Hydro SV accessory
Washing station for Hydro SV

CHEMICALS

Water
Crude oil
Organic solvents (Toluene, Isopropanol)
Solid particles

 NTNU Faculty of Natural Sciences and Technology	Procedure: Mastersizer 3000	
Procedure number: 2-0669-01-02	Revision no.: 2	Page 3 of 6

6. METHOD

This procedure describes performing a manual measurement. It is possible to create an SOP (Standard operating procedure) in an analogous way, to quickly repeat similar measurements. The person responsible for the instrument can also create SOP on request, during training.

The instrument is equipped with two measurement accessories: Hydro EV (400-600 ml) and Hydro SV (6-7 ml). Depending on the amount of available sample, the user can choose whichever accessory fits best for his purposes.

Before measurements make sure that the sample and solvents used for cleaning after, are compatible with the tubing (can be changed, but normally is PVC) and sealing material (Viton®). Incompatible solvent may damage the instrument.

It is necessary to know the refractive index and the density of the dispersed material, as well as the refractive index of the dispersant before the measurement starts.


The instrument needs to be turned on 30 min before measurements to ensure thermal stability in the cell.

1. Turn on the computer and then the software on the desktop (Mastersizer 3000).
2. In the bottom-right corner check, if the instrument and accessory is connected properly (Mastersizer 3000 and Hydro EV)
3. Click **New – Measurement File**
4. Click **Manual Measurement** from the **Measurement** ribbon at the top
 - a. Name the sample
 - b. Select particle type:
 - i. Emulsions: spherical
 - ii. Dispersions: non-spherical (recommended, however not necessary)
 - c. Select material by:
 - i. Using the existing database (you can edit the database by adding known materials or chemicals)
 - ii. Manual input of the refractive index and the density
 - d. Select dispersant by:
 - i. Database (same as in c.)
 - ii. Manual input of the refractive index
 - e. Measurement duration

Not that important, anywhere between 10-20 s (usually 10s) depending on how much time you have

- f. Blue laser light measurement

Blue laser is used for very fine particles. If you expect that your sample contains particles smaller than 150 nm, then check that option. If not, leave it unchecked, as it will prolong your measurement.

 NTNU Faculty of Natural Sciences and Technology	Procedure: Mastersizer 3000	
Procedure number: 2-0669-01-02	Revision no.: 2	Page 4 of 6

g. Sequence

Depends on how much time you want to spend on each sample. 3 runs per measurement is sufficient, more than 5 is unnecessary.

h. Obscuration

Heavily dependent on your sample and its concentration, may require some initial screening. See Additional Notes in the end of the procedure for more information.

Standard values are:

- i. 10-20% for wet sample (emulsions, dispersions)
- ii. 1-10% for dry sample
- iii. 5-12% for crude oil emulsion (50-100 ppm concentration)

i. Accessories – stirrer

- i. Anywhere between 2000-3000 rpm
- ii. Note: if you have a full beaker and stir too fast, spillage may occur

j. Cleaning

- i. Normal – 3 sequences (for solid particles should be enough)
- ii. For crude oil emulsions (custom procedure)

k. Analysis Model

- i. General Purpose – most likely use this
- ii. Narrow – if you expect only a single peak (very monodisperse system)

l. Result type

It is recommended to use the volume distribution, however the choice is up to the user.


m. User sizes

It is recommended to use the default sizes.

Instructions for Hydro EV (points 5 to 10)

Instructions for Hydro SV (points 11 to 15)

5. Add beaker with dispersant (pure continuous phase) and lower the head. Make sure that the beaker is not full – 60-80% volume is usually enough for the measurement and to prevent spillage during mixing.
 6. Click **Initialize the instrument** to initialize and **Start** again to measure the background. The background is of good quality, when the indicator of energy on the 1st and 20th detector is less than 100 and 20, respectively.
 7. Add the sample into the beaker until you have reached sufficient obscuration (indicator on the left side of the screen), wait 30-50 seconds and then start measurements. After the measurement is done, you may preform additional measurements or skip to cleaning.
 8. Start cleaning by clicking the **Clean** button (even though it may seem greyed out)
 9. Follow the instructions on the screen. If you rinse the apparatus with organic solvents, make sure you use the portable fume hood and half-mask with appropriate filters. When the cleaning sequence is complete, stop the stirrer and exit the measurement window.
-

 NTNU Faculty of Natural Sciences and Technology	Procedure: Mastersizer 3000	
Procedure number: 2-0669-01-02	Revision no.: 2	Page 5 of 6

10. Remember to save your measurement file in your folder.

Cleaning:

- Dispersion with solid particles – rinsing the system several times with water is sufficient
- Crude oil emulsions – rinsing twice with isopropanol/toluene (50/50), then isopropanol once and then 4-5 times with tap water.


**Make sure to clean any spillage to prevent damaging the equipment.
After measurements are complete, remember to clean the tray!**

There is a possibility of connecting temperature control unit to the cell, to measure size distributions in various temperatures.

Instructions for Hydro SV

- Carefully add the dispersant (pure continuous phase) to the cell by using a pipette or syringe. Avoid creating gas bubbles in the cell. Do not scratch the glass of the cell. Insert the accessory back into the instrument. You may add magnetic stirrer in the cell to have and control the mixing in the cell.
- Click **Initialize the instrument** to initialize and **Start** again to measure the background. The background is of good quality, when the indicator of energy on the 1st and 20th detector is less than 100 and 20, respectively.
- Take out the accessory from the instrument. Remove some dispersant and add your sample directly in the cell. This method may require some experience with the sample, as you need to be in a specific range of the obscuration (indicator on the left side of the screen). After adding the sample, put the accessory back into the instrument. If the appropriate level of obscuration is reached, wait 30-50 seconds and then start measurements. After the measurement is done, you may perform additional measurements or skip to cleaning.
- Cleaning is performed with a washing station. Take out the accessory and unlock the cell. Put the cell in the right position in the washing station and flush the cell several times with an appropriate solvent:
 - Dispersion with solid particles – rinsing the system several times with water is sufficient
 - Crude oil emulsions – rinsing twice with toluene, then twice with isopropanol, then 4-5 times with tap water. Finish the cleaning with flushing the cell with isopropanol.
- Put the cell on a fibreless cloth and let it dry.

The results can be accessed by opening the measurement file. The software produces graphs, different distributions and various parameters, however the raw data can still be exported to a text file or excel sheet.

 NTNU Faculty of Natural Sciences and Technology	Procedure: Mastersizer 3000	
Procedure number: 2-0669-01-02	Revision no.: 2	Page 6 of 6

It is possible to install the software on user's personal computer and access the files there. Ask the person responsible for the instrument for more details.

7. SAFETY EQUIPMENT

Safety goggles

Lab coat

Nitrile gloves

Half-mask

Portable fume hood

8. REFERENCES

Mastersizer 3000 manual

Malvern educational materials

ADDITIONAL NOTES

Obscuration tells you how much light is lost during your measurement. The software typically adjusts for 10-20% of obscuration, which means that the value above or below this range can yield inaccurate results. For example, if the concentration is too high, there is a risk of multiscattering (light is scattered off many particles). This will result in larger angle of scattering and a signal from very small particles (below 1 μm). Sometimes it can also be spotted by the presence of bimodal, non-continuous distribution of particles. Conversely, if the dispersion concentration is too low, the model in the software 'overadjusts' and can shift the distribution towards larger sizes of drops.
