NTNU
Norwegian University of
Science and Technology

Risk-based health-aware control of Åsgard subsea gas compression station

Julie Berge Ims

Chemical Engineering and Biotechnology
Submission date: June 2018
Supervisor: Johannes Jäschke, IKP
Co-supervisor: Adriaen Verheyleweghen, IKP

Norwegian University of Science and Technology
Department of Chemical Engineering

Preface

This master's thesis is the final work of the five year Master's program within Industrial Chemistry and Biotechnology at the Norwegian University of Science and Technology (NTNU), resulting in a M.Sc. degree in Chemical Engineering. The thesis was written in the spring of 2018 at the Department of Chemical Engineering, NTNU.

I would like to thank my supervisor, Associate Professor Johannes Jäschke, for valuable counselling and the opportunity to work on this very exciting project. Also, I wish to express my sincerest gratitude to my co-supervisor, PhD candidate Adriaen Verheyleweghen, for continuous assistance and feedback throughout the process. You have given me the ability to explore and investigate interesting topics and yet forced me into structuring my priorities to improve my work.

Furthermore, I would like to thank my friends and fellow students at NTNU for five wonderful years. Finally, I am grateful to my parents, Anne Berit Berge Ims and Leif Aarthun Ims, for all the love and support I have received throughout my education. I would like to give special thanks to my father, Leif Aarthun Ims, for his continued guidance on this thesis.

Declaration of Compliance

I hereby declare that this is an independent work according to the exam regulations of the Norwegian University of Science and Technology.

Trondheim, 2018-06-12

Julie Berge Ims

Julie Berge Ims

Abstract

Safe and efficient operation of subsea processing systems imposes strict requirements with respect to equipment design and reliability. This is to avoid accidental shutdowns, which can lead to expensive maintenance engagements. For that reason, health monitoring methods are applied to monitor and evaluate the condition of the overall system in real-time. However, when finding the optimal operation policy, the health condition is generally not reviewed directly. As a consequence, this may cause overly restrictive operation. This study will suggest to combine control and condition monitoring of the Åsgard gas compression station, in order to prevent the operation policy from being sub-optimal. In this manner, the obtained optimal plan of action for operation will seek to sustain the reliability of the subsea system. This makes it possible to forecast the health of the system and manage the operation accordingly, rather than just reacting to it.

This thesis proposes a model predictive control (MPC) approach for integrating health monitoring and control. The scheme will seek to ensure safe operation and an economic optimal control policy for the subsea station. Risk measures that estimate the risk of failure are used for condition monitoring purposes. In this work, Conditional Value-at-Risk (CVaR) with respect to the random variable remaining useful life (RUL) of equipment is incorporated into the optimal control problem to assess the condition of system equipment. CVaR estimates the risk of failure in a conservative manner by bringing the extreme RUL of equipment outcomes into focus for a confidence level, α. The theoretical analysis shows that minimization of unavailability of equipment coincides with the maximization of CVaR with respect to RUL of equipment. Control of the predicted CVaR with respect to RUL of equipment is employed to enforce safe operation until the next maintenance engagement.

The numerical simulations show that the predicted CVaR with respect to the RUL of equipment decreases with time until the next maintenance engagement, which is scheduled to happen in five years. The average RUL of the 0.1% worst RUL outcomes has been calculated to be to just above five years at the startup of the plant. A higher confidence level gives rise to higher values for CVaR with respect to RUL of equipment. In this approach, maximizing profit in terms of gas production while maximizing average RUL of the 0.1% worst RUL outcomes gives a production profile where the gas production rate decreases with time.

The overall conclusion from this work is that health-aware control with risk measures for condition monitoring has the potential to manage the reliability of a subsea plant. Nevertheless, the accuracy of the system model and the implementation of the risk measure estimate influence the ability of the controller to predict the risk of failure.

Sammendrag

Sikker og effektiv drift av prosessystemer på havbunnen stiller strenge krav til reliabilitet og design av utstyr. Dette er for å unngå tilfeldig driftsstans av prosessanlegg og kostbart vedlikeholdsarbeid. Av denne grunn brukes overvåkingsmetoder for å observere og evaluere tilstanden til hele systemet i sanntid. Generelt sett vurderes ikke systemts tilstand direkte når den optimale kontrolstrategien for driften av systemet utformes. Dette kan imidlertid føre til altfor restriktiv drift av prosessanlegget. Denne oppgaven vil foreslå å kombinere kontroll og tilstandsovervåking av gasskompresjonsstasjonen på Åsgard feltet for å forhindre at kontrollstrategien blir suboptimal. På denne måten vil den optimale kontrolstrategien for driften av systemet forsøke å opprettholde reliabiliteten av prosessanlegget på havbunnen. Denne metoden gjør det mulig å administrere driften av anlegget i henhold til prognoser for tilstanden til systemet, i stedet for å bare respondere på observasjoner.

Denne oppgaven benytter modell prediktiv kontroll (MPC) til å integrere tilstandsovervåking og kontroll. Denne metoden vil prøve å sikre trygg drift av anlegget og en økonomisk optimal kontrollstrategi for undervannsanlegget. Tilstandsovervåkingen vil benytte risikoevalueringer for å anslå risikoen for svikt i systemet. I dette arbeidet er Conditional Value-at-Risk (CVaR) med hensyn til den stokastiske variabelen for gjenværende levetid (RUL) av utstyr, innarbeidet i det optimale kontroll problemet for å vurdere tilstanden til utstyret. CVaR anslår risikoen for svikt på en konservativ måte ved å fokusere på de ekstreme tilfellene av RUL for et gitt konfidensnivå, α. Den teoretiske analysen viser at minimialisering av utilgjengelighet av utstyr sammenfaller med maksimering av CVaR med hensyn til RUL. Kontroll av CVaR med hensyn til RUL er anvendt for å opprettholde trygg drift av anlegget frem til neste planlagte vedlikeholdsarbeid.

De numeriske simuleringene viser at CVaR med hensyn til RUL synker med tiden frem til neste planlagte vedlikeholdsarbeid om fem år. Gjennomsnittlig RUL av de 0,1% verste utfallene av RUL er kalkulert til å være litt over fem år ved oppstart av anlegget. Et høyere konfidensnivå resulterer i høyere verdier for CVaR med hensyn til RUL. Ved å maksimere inntekt i form av gassproduksjon parallellt med å maksimere gjennomsnittlig RUL av 0,1% dårligste utfallene av RUL, blir utfallet at produksjonsraten av gass synker med tiden.

Den overordnede konklusjonen fra dette arbeidet er at MPC kombinert med risikomålinger for tilstandsovervåking har potensial til å håndtere reliabiliteten til prosessystemer på havbunnen. Systemmodellens nøyaktighet og implementeringen av mål på risiko påvirker kontrollerens evne til å forutsi risikoen for svikt.

Table of Contents

List of Tables

List of Figures

List of Symbols

N_μ	Dimensionless viscosity number.	-
q_{max}	Maximum allowable flow in compressor to avoid choking.	m^3/s
q_{min}	Minimum allowable flow in compressor to prevent surge.	m^3/s
α	Confidence level for risk measurements.	$\%$
α_s	Cyclone separation efficiency.	$\%$
β^k	Discounting factor for risk measures at time k.	-
δ_l	Liquid film thickness in the cyclone wall.	m
$\dot{m}_1$	Mass flow rate entering the valve.	kg/s
$\dot{m}_{1,g}$	Gas mass flow rate entering the valve.	kg/s
$\dot{m}_{1,l}$	Liquid mass flow rate entering the valve.	kg/s
$\dot{Q}_l$	Volumteric liquid flow in the cyclone wall.	m^3/s
η	Compressor efficiency.	-
Γ	Volumteric liquid flow in the cyclone wall per unit wetted perimeter.	m^2/s
γ	Adiabatic constant.	-
$\hat{\theta}$	Angle to indicate direction of gas flow in the wetted perimeter.	^{o}
κ	Constant.	-
λ_w	Scale parameter in Weibull distribution.	-
μ_l	Dynamic viscosity for liquid film on cyclone wall in the separator.	cP
ω	Weighting factor between the reliability and the economic objective.	-

Φ	Objective function in the optimization problem.	-
ϕ_e	Economic objective.	-
ϕ_r	Reliability objective.	-
ψ	Random variable that denotes remaining useful life of equipment.	Years
ρ_1	Density of flow entering the choke valve.	kg/m^3
$\rho_{3,avg}$	Average density upstream to the compressor.	kg/m^3
$\rho_{3,g}$	The density of the gas in the gas stream entering the compressor.	kg/m^3
$\rho_{3,l}$	The density of the condensate in the gas stream entering the compressor.	kg/m^3
ρ_3	Total density of the gas stream entering the compressor.	kg/m^3
ρ_g	Density of gas flow in the separator.	kg/m^3
ρ_l	Density of liquid film in the separator.	kg/m^3
σ	Interfacial/surface tension between two phases in the separator.	mN/m
σ_c	Temporary variable applied in the calculation of the compressibility factor.	-
$\tau_{i,tg}$	Tangential component of shear stress acting on the film due to the gas flow.	Pa
$\tau_{w,tg}$	Tangential component of shear stress acting on the wall due to the liquid film.	Pa
$\mathbf{A}$	Non-anticipativity constraints in the stochastic optimization problem.	-
$\mathbf{c}_3$	Composition of chemical compounds in the inlet stream to the compressor.	-
$\mathbf{f}$	Vector with differential equations.	-
$\mathbf{g}$	Vector with algebraic equations.	-
$\mathbf{p}$	Vector with random parameters.	-
$\mathbf{u}$	Vector with control inputs.	-
$\mathbf{u}_{lb}$	Vector with lower bounds for input variables.	-
$\mathbf{u}_{ub}$	Vector with upper bounds for input variables.	-
$\mathbf{x}$	Vector with differential variables.	-
$\mathbf{x}_0$	Vector with initial states at time t = 0.	-
$\mathbf{x}_{lb}$	Vector with lower bounds for differential variables	-
$\mathbf{x}_{ub}$	Vector with upper bounds for differential variables.	-
$\mathbf{z}$	Vector with algebraic variables.	-

$\mathbf{z}_{lb}$	Vector with lower bounds for algebraic variables.	-
$\mathbf{z}_{ub}$	Vector with upper bounds for algebraic variables.	-
θ	Angle between vertical and tangential gas velocity on the cyclone wall.	^{o}
φ	Arbitrary static risk measure.	-
A	Constant for linear model for Re-entrainment number.	-
a	Model-constant.	-
a_l	Centrifugal acceleration acting on the liquid film in the cyclone wall.	m/s^2
A_{choke}	Cross section area of the choke valve.	m^2
B	Constant for linear model for for Re-entrainment number.	-
$c(t)$	Discounting term for NPV calculations.	-
C_d	Valve constant.	-
c_{choke}	Choke constant.	kg/s$\sqrt{bar}$
$C_{p,3}$	Heat capacity heat capacity for the inlet gas stream to the compressor.	J/K
$C_{p,4}$	Heat capacity heat capacity for the outlet gas stream from the compressor.	J/K
c_{Paris}	Lumped parameter in Paris' law for crack propagation.	s/J^2
D	Separator diameter.	m
D_{comp}	Material constant.	-
E	Dimensionless Re-entrainment number.	-
$f(z)$	Valve characteristic.	-
F_d	Drag force from the gas flow on the liquid wave peak in the separator.	N
F_k	Filtration at time k.	-
F_ψ	Cumulative distribution function for ψ.	-
f_ψ	Probability density function for ψ.	-
F_ψ^{-1}	Inverse cumulative distribution function for ψ.	-
F_σ	Retaining force of the surface tension in the separator.	N
$f_{g,i}$	Friction factor for gas on the liquid film.	-
$f_{i,w}$	Friction factor for liquid on the cyclone wall.	-
f_{wood}	Woods correction factor.	-

g	Gravitational constant.	m/s^2
GVF_3	Gas-volume fraction of the gas upstream to the compressor.	-
H	Compressor head.	Nm/kg
h	Bearing crack length in the wet-gas compressor.	mm
h_0	Initial bearing crack-length in the wet-gas compressor.	mm
hi	Characterisitc health indicator.	-
i	Discounting rate for NPV calculations.	-
K	Constant.	-
k	Variable used to track each time step.	-
K_w	Shape parameter in Weibull distribution.	-
l	Variable used to track each scenario.	-
M	Molar mass for the stream.	kg/kmol
m	Constant for friction factor expression.	-
N	Number of time periods. It is the length of the horizon.	-
P_1	Pressure upstream to the valve.	bar
p_l	Probability of occurrence for scenario l in the stochastic optimization problem.	$\%$
P_w	Wetted perimeter.	m
P_2	Pressure downstream from the valve.	bar
P_3	Pressure upstream to the compressor.	bar
P_{out}	Pressure through the pipeline to topside.	bar
P_{pc}	Pseudo-critical mixture pressure.	K
P_{pr}	Pseudo-reduced pressure.	-
Pow	Compressor power.	Watt
q_3	Volumetric flow upstream to the compressor.	m^3/s
q_α	Quantile in the Weibull distribution with confidence level, α.	Years
R	Static risk measure.	-
$R_{k,N}$	Dynamic risk measure defined at time k.	-
Re_l	Reynolds number for liquid film on the cyclone wall.	-

S	Number of scenarios.	-
Srg	Surge condition for compressor choking.	-
Stw	Stonewall condition for compressor choking.	-
T	Temperature in compressor.	K
t	Time variable.	Years
t_0	Initial time.	Years
t_f	Time until the next maintenance intervention.	Years
t_k	Time at time period k.	Years
T_3	Temperature upstream to the compressor.	K
T_4	Temperature downstream from the compressor.	K
T_{comp}	Motor Torque in the wet-gas compressor.	Nm
T_{pc}	Pseudo-critical mixture temperature.	K
T_{pr}	Pseudo-reduced temperature.	-
u_l	Velocity of the liquid film on the cyclone wall.	m/s
u_z	Vertical gas velocity on the cyclone wall.	m/s
u_{choke}	Choke opening.	-
u_{comp}	Compressor speed.	-
$u_{g,tg}$	Tangential gas velocity in the cyclone.	m/s
$u_{l,tg}$	Tangential velocity of the liquid film on the cyclone wall.	m/s
u_s	Superficial gas velocity in the separator unit.	m/s
w	Adiabatic constant.	-
Z	Compressibility factor of the gas upstream to the compressor.	-
CVaR_α	Conditional Value-at-Risk with confidence level, α.	Years
R	Gas constant.	J/K mol
VaR_α	Value-at-Risk with confidence level, α.	Years

Abbreviations

AFC	Axial flow cyclone
CDF	Cumulative distribution function
CVaR	Conditional Value-at-Risk
DAE	Differential algebraic equation
GVF	Gas Volume Fraction
ICDF	Inverse cumulative distribution function
MPC	Model predictive control
NLP	Non-linear program(ming)
NPV	Net present value
NTNU	Norwegian University of Science and Technology
OCP	Optimal control problem
PDF	Probability density function
PHM	Prognostics and health monitoring
VaR	Value-at-Risk
RUL	Remaining useful life

Chapter 1

Introduction

Subsea processing technology is developed to overcome many challenges associated with topside oil and gas operations. This technology enables production from reservoirs and fields previously deemed too remote in arctic or hostile environments. The main purpose with subsea processing systems is to strengthen the economic result from the operation through cost reduction and increase in production. Subsea technology is a mature technology that has been applied in a number of oil fields on the Norwegian continental shelf, i.e. Troll and Åsgard (McClimans et al., 2006). However, since each field is unique, unsolved problems still exists (Moreno-Trejo and Markeset, 2011a,b). Each field brings new industrial challenges and demands in terms of design, operation and reliability of the subsea systems.

1.1 Motivation

New challenges arise when operating oil and gas systems on the seabed. Inaccessibility of the subsea plant is one of the most crucial challenges when shifting topside equipment to the seabed. Maintenance engagements are considerably rare for subsea systems, as they require specialized intervention ships to carry out operations on the bottom of the ocean. Consequently, unplanned shutdowns, which may cause expensive maintenance engagements are attempted to be avoided at every opportunity. Because of this, safe and efficient operation of subsea plants imposes strict requirements both with respect to equipment design and reliability. Information about the condition of the system can be used to establish effective maintenance policies and to forecast the health condition of system components in the future. Condition monitoring techniques are applied to evaluate the health of the subsea system in real-time. Unfortunately, if the decision-making process does not handle the information from the condition monitoring system explicitly, the process might be overly restrictive (Verheyleweghen and Jäschke, 2017b). This study will suggest to combine control and condition monitoring in order to prevent sub-optimal operation. This is to obtain an optimal control policy for operation, without compromising the reliability of the plant.

1.2 Scope of Work

This master thesis focuses on the subsea gas compression station at the Åsgard field, as depicted in Figure 1.1. This study proposes a model predictive control (MPC) approach for integrating condition monitoring and control. In this manner, the obtained optimal plan of action for operation will seek to sustain the reliability of the subsea station. Financial risk measures are not commonly employed in process control. However, risk controlling techniques that consider the risk of failure are used for condition monitoring purposes for the subsea plant. Numerical tests are performed to evaluate if the chosen risk measure is suitable for risk control of the subsea plant. In detail, this involves implementing a suited risk measure in the form of MATLAB code in the optimal control problem. Additionally, the closed-loop MPC is implemented in MATLAB to add random disturbance to the optimization.

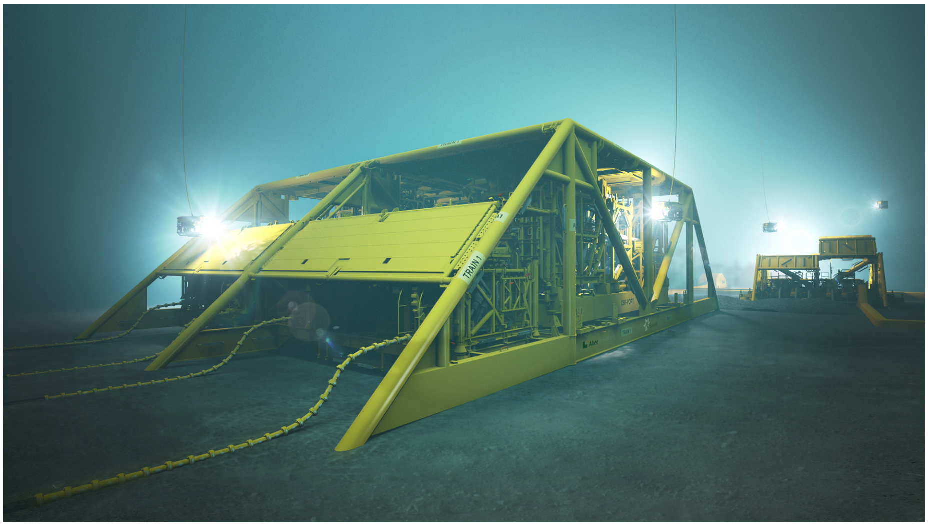


Figure 1.1: Artist rendition of the Åsgard gas compression station. Copyright: Aker Solutions.

1.3 Previous Work

The term health-aware control is a design that integrates prognostics and health monitoring (PHM) and control to ensure reliable and efficient operation of systems subject to instrumental faults and hazards (Escobet et al., 2012). In this approach, the obtained control policy for operation will seek to sustain the reliability of the system. Condition monitoring methods are generally combined with PHM to assess the system condition. In recent years, there have been several attempts at combining condition monitoring techniques and MPC. This is a predictive control scheme that combines feedback control with periodic optimization of the system model subject to constraints in order to generate optimal control policies (Morari and Lee, 1999). There have been several attempts at incorporating

PHM in the objective function or the constraints in the optimization. Pereira et al. (2010) attempted to distribute the control effort in a simulated tank level control system by imposing constraints directly on the accumulated actuator degradation in the optimization problem. Salazar et al. (2016) employed PHM explicitly in the constraints in the pumps in a drinking water system. Sanchez et al. (2015) proposed to minimize damage on wind turbine blades by including a prognosis that was based on fatigue in the objective function.

This master thesis is a continuation of a project work on health-aware control of the subsea gas compression station at the Åsgard field (Ims, 2017). This study was originally based on a mathematical model of the Åsgard subsea gas compression station implemented in MATLAB by supervisor Adriaen Verheyleweghen. First efforts at optimizing the subsea system was handled as part of the work with the project thesis (Ims, 2017). Two health propagation models for condition monitoring were investigated, the degradation of equipment and hazard functions for remaining useful life (RUL) of equipment. In the first method, Paris' law for crack propagation was used to predict degradation of equipment. Constraints were applied on allowable accumulated degradation of system to enforce safe operation. In the second approach, the hazard function for RUL of equipment acted as a chance constraint on RUL of equipment. Constraints were applied on allowable cumulative hazard to ensure reliable operation until the next maintenance engagement. The results in terms of the economic outcome are somewhat unexpected. The operational strategy is more profitable when the cumulative hazard function for RUL of equipment is used to monitor the health condition development

A detailed separator model has also been implemented as part of the work with the project to provide accurate predictions of liquid carry over to the wet-gas compressor. A new separator model is proposed to reduce uncertainty and enable enhanced production through less conservative operations. The mathematical model designed is based on a correlation between the cyclone separation efficiency and the dimensionless re-entrainment number.

1.4 Outline

Chapter 1 gives a brief introduction to the study. The motivation for this work and scope is described here. This chapter also includes an overview of previous work on the same subject. Chapter 2 discusses the topics of optimization theory and optimal control. The chapter provides details on relevant optimization problem formulations. Towards the end, the features and structure of the model predictive control framework are presented. Chapter 3 provides a detailed description of risk measures, both static and dynamic, for risk controlling purposes. The topic of risk control in optimization problems is considered at the end of Chapter 3. Chapter 4 presents a process description of the subsea gas compression station at the Åsgard field. The model equations for the choke, the separator and the compressor are given here. Chapter 5 addresses the full optimal control problem in detail. Chapter 6 presents the results obtained working with this study. Chapter 7 contains concluding remarks and suggestions for future work.

Chapter 2

Optimization and Optimal Control

”*For since the fabric of the universe is most perfect, and is the work of a most wise Creator, nothing whatsoever takes place in the universe in which some relation of maximum and minimum does not appear.*”

— Leonhard Euler

It is an indisputable fact that people optimize by making decisions for the sole intention of maximizing their quality of life in some way or another (Kiranyaz et al., 2014). For this reason, optimization has found applications in a number of areas. Investors aim to form portfolios that obtain a high rate of return while preventing extreme risk. Manufacturers seek ultimate productivity from operation and design of their processes. Engineers intend to improve system performance of their model by modifying parameters (Nocedal and Wright, 2006).

Optimization has been a fundamental concept in human history long before mathematical models and computers were developed. The conception of optimization is the process of locating the optimum of systems (Kiranyaz et al., 2014). The underlying idea of optimization originates from the work of Euler and Lagrange (Nocedal and Wright, 2006) in the 1800s. Advances in the theory of optimization were managed by the likes of Gauss and Newton. Newton and Gauss presented iterative techniques for shifting against an optimal state. George Dantzig introduced a general linear programming formulation and invented the Simplex method in 1947 (Gill et al., 2008). This led to optimization being introduced in other areas outside mathematics. Further study in the area of optimization led to the formation of dynamic optimization by Bellman (1954), as well as the unfolding of nonlinear solvers such as IPOPT (Wächter and Biegler, 2006). Dynamic optimization is also known as the modern term dynamic programming which employs solving sub-problems inside larger decision problems. With these advancements, optimization has become a widespread tool utilized in various areas like science, engineering and finance.

Optimization can be applied in numerous areas within chemical engineering. Process control is one field in particular which benefits greatly from optimization. Optimization problems solved in process control are most frequently referred to as optimal control problems (OCPs). The complexity and structure of the optimization problem will influence what solver methods are suitable for the OCP. Furthermore, in process control, minor alterations in operating conditions can have enormous impact on system performance. For that reason, it is convenient to have a systematic approach for locating the optimal operating conditions which yield the most profitable outcome. Model predictive control (MPC) is an optimization based control strategy often employed in process industries (Lucia et al., 2013b). The MPC scheme yields an optimal sequence of control inputs which is obtained by means of mathematical optimization. This work will apply the MPC framework in order to obtain the optimal trajectory of control inputs for the subsea gas station at the Åsgard field. Section 2.1 will give a brief introduction to optimization theory. Relevant forms of optimization problems such as dynamic optimization as well as dynamic stochastic optimization will also be presented here. At last, Section 2.2 will discuss the MPC scheme applied for optimization of the Åsgard subsea gas compression station.

2.1 Optimization Theory

In computer science and mathematics, the general understanding of an optimization problem is finding the optimal solution out of all possible solutions. Optimization problems are categorized based on whether the variables involved are discrete or continuous. The focus of this work will be continuous optimization problems formulated with constraints. The canonical form of a general continuous optimization problem is

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{z},\mathbf{u}} \quad & \Phi(\mathbf{x},\mathbf{z},\mathbf{u},\mathbf{p}) \\
\text{s.t.} \quad & \mathbf{f}(\mathbf{x},\mathbf{z},\mathbf{u},\mathbf{p}) \leq 0 \\
& \mathbf{g}(\mathbf{x},\mathbf{z},\mathbf{u},\mathbf{p}) = 0.
\end{aligned}
\tag{2.1}
$$

$\mathbf{f}(\mathbf{x},\mathbf{z},\mathbf{u},\mathbf{p}) \leq 0$ are labeled the inequality constraints and $\mathbf{g}(\mathbf{x},\mathbf{z},\mathbf{u},\mathbf{p}) = 0$ are referred to as the equality constraints (Biegler, 2010). The latter two terms define the feasible set of solutions for the optimization problem. $\Phi(\mathbf{x},\mathbf{z},\mathbf{u},\mathbf{p}) \in \mathbb{R}$ is the objective function to be optimized. The objective function is integrated into the optimization problem formulation to enable differentiation of the feasible solutions. By convention, the common design of an mathematical optimization problem defines a matter of minimization. A maximization problem can be considered by employing the negative of the objective function.

Problems of the general form given in Equation 2.1 can be classified according to the characteristics of the objective function and constraints, i.e. linear, nonlinear and convex (Kiranyaz et al., 2014). Optimization problems in which the objective function or some of the constraints are nonlinear are referred to as non-linear optimization problems. Chapter 4 will present model equations of nonlinear nature. The process of solving an optimization problem of non-linear characteristics is called nonlinear programming (NLP).

The idea of convexity is significant in optimization (Kiranyaz et al., 2014). In general, many real-life optimization problems possess this characteristic. Convex optimization problems are simpler to solve both in theory and practice. The convex term can be adapted to both functions and sets. A set $S \in \mathbb{R}^n$ is considered a convex set if the direct line between any two points x and y in S, lies exclusively within S (Biegler, 2010). Figure 2.1 display an example of a convex set S and a non-convex set S' with two points x and y.

(a) Convex set S with two points x and y.

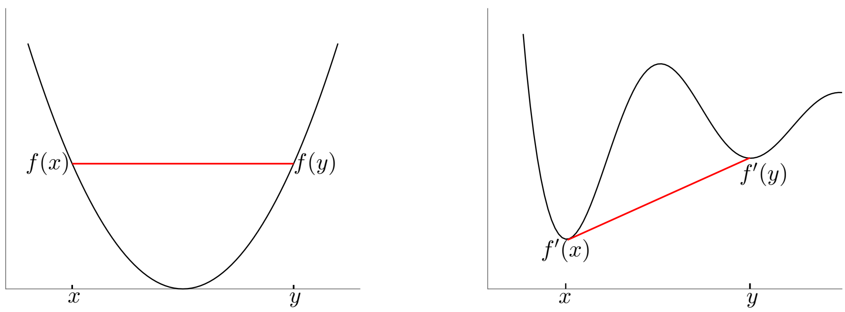
(b) Non-convex set S' with two points x and y.

Figure 2.1: Example of a convex set, S, and a non-convex set, S', with two points, x and y, explicitly marked.

An arbitrary function f is considered a convex function if its domain S is a convex set and if for any two points x and y in S, the following feature is fulfilled (Biegler, 2010):

$$f(\kappa x + (1-\kappa)y) \leq \kappa f(x) + (1-\kappa)f(y), \qquad \forall\, \kappa \in [0,1]. \tag{2.2}$$

Figure 2.2 display an example of a convex function f and a non-convex function f' defined on a set S with two points x and y.



(a) Convex function f defined on a set S with two points x and y.

(b) Non-convex function f' defined on a set S with two points x and y.

Figure 2.2: Example functions of a convex function, f, and a non-convex function, f', with two points, x and y, explicitly marked.

The solution obtained when solving the optimization problem given in Equation 2.1 is in fact a global solution if the objective function in the optimization problem and the feasible domain are both convex (Kiranyaz et al., 2014). This is applicable when solving convex problems with both local and global solvers. However, solving a non-convex optimization problem with a local solver, cannot guarantee that the minimum point obtained is a global minimum. Solving a non-convex optimization problem with a global solver will, in most cases, locate a global minimum. Nevertheless, this is rather computational demanding and not recommended. The concept of convexity is not discussed any further as it is considered to be outside the scope of this thesis.

2.1.1 Dynamic Optimization

Mathematical optimization problems emerging from multistage decision processes occur in various areas like science, engineering and finance. Multistage optimization problems can be broken down into a sequence of simpler sub-problems to facilitate complex problems. This is a simplification that allows for decomposing of complex decision processes into a string of elementary decision stages over time. This is commonly referred to as dynamic optimization (Bellman, 1954). Dynamic optimization problems solved in process control are called dynamic optimal control problems. Dynamic optimal control is a rather widespread form of OCP in which the optimal state of the system alters with time. The process of solving dynamic optimization problems is called dynamic programming.

The design of a general dynamic optimization problem gives rise to some trivial assumptions. Suppose that the dynamic system to be optimized can be described by differential-algebraic equations (DAEs). The DAEs are expressed with respect to an independent variable representing time, t. On that regard, defined initial conditions at time $t = 0$ are necessary for finding a solution to the dynamic optimization problem. The order of the differential equations determine the number of initial conditions that are required for solving the DAE. In the area of process engineering, DAEs are generally stated as initial value problems with initial conditions (Biegler, 2010),

$$\begin{aligned} \mathbf{f}\big(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}\big) &= \frac{d\mathbf{x}}{dt} \\ \mathbf{g}\big(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}\big) &= 0 \\ \mathbf{x}(0) &= \mathbf{x}_0. \end{aligned} \tag{2.3}$$

Equation 2.3 introduces some generic notation. $\mathbf{g}$ denotes the algebraic equations and $\mathbf{f}$ denotes the differential equations. $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ represents the differential variables while $\mathbf{x}_0$ are initial states at time $t = 0$. $\mathbf{z}(t) \in \mathbb{R}^{n_z}$ denotes the algebraic variables. $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ are the control variables. $\mathbf{x}(t), \mathbf{z}(t)$ and $\mathbf{u}(t)$ are functions of time, $t \geq 0$. $\mathbf{p} \in \mathbb{R}^{n_p}$ represents the time-independent parameters. Based on the set of DAEs given in Equation 2.3, assume that with designated values of $\mathbf{x}(t)$, $\mathbf{u}(t)$ and $\mathbf{p}$, $\mathbf{z}(t)$ can be obtained exclusively by $\mathbf{g}$ (Biegler, 2010).

In a dynamic environment, the objective function defines the target of every decision stage in the optimization. Assume that the optimization problem has a fixed time horizon, t_f.

The objective function can be formulated as

$$\int_0^{t_f} \Phi\big(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}\big)\, dt. \tag{2.4}$$

The constraints defined in Equation 2.3 and the objective function defined by Equation 2.4 are merged to formulate a dynamic optimization problem. Assuming that the dynamic optimization problem is a matter of minimization, the problem can be formulated as (Biegler, 2010)

$$\begin{aligned}
\min_{\mathbf{x},\mathbf{z},\mathbf{u}} \quad & \int_0^{t_f} \Phi(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p})dt \\
\text{s.t.} \quad & \mathbf{f}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}) = \frac{d\mathbf{x}}{dt} \\
& \mathbf{g}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}) = 0 \\
& \mathbf{x}(0) \ = \ \mathbf{x}_0.
\end{aligned} \tag{2.5}$$

Equation 2.5 may yield solutions which are not within a safe operating domain or physically not feasible. Because of this, bounds must be imposed on particular variables in order to limit the scope of operation. The limits can be inflicted based on the design of the system operation, for instance the maximum allowable pressure inside a compressor. The bounds can be enforced to guarantee a physically consistent system. In this regard, the following equation will yield feasible solutions that abide by the lower- and upper bounds for $\mathbf{x}$, $\mathbf{z}$ and $\mathbf{u}$ and fulfills the constraints specified in Equation 2.3,

$$\begin{aligned}
\min_{\mathbf{x},\mathbf{z},\mathbf{u}} \quad & \int_0^{t_f} \Phi(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p})dt \\
\text{s.t.} \quad & \mathbf{f}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}) = \frac{d\mathbf{x}}{dt} \\
& \mathbf{g}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}) = 0 \\
& \mathbf{x}(0) \ = \ \mathbf{x}_0 \\
& \mathbf{x}_{\text{lb}} \leq \mathbf{x} \leq \mathbf{x}_{\text{ub}} \\
& \mathbf{z}_{\text{lb}} \leq \mathbf{z} \leq \mathbf{z}_{\text{ub}} \\
& \mathbf{u}_{\text{lb}} \leq \mathbf{u} \leq \mathbf{u}_{\text{ub}}.
\end{aligned} \tag{2.6}$$

In the equation above the subscript lb represents lower bounds and the subscript ub indicates upper bounds.

There are two different schemes for solving dynamic optimization problems, indirect and direct methods (Verheyleweghen and Jäschke, 2017b). The indirect method provides solutions with continuous input profiles, while the direct methods are based on time discretization and give approximate solutions. Despite this, efficient solution algorithms and easy implementation of the direct methods make this approach more applicable. In the context of this work, a direct method will be used to solve the dynamic optimization problem for the subsea station. The direct methods solve the optimization problem numerically by transforming the DAE to a non-linear programming (NLP) problem via discretization.

Furthermore, existing direct methods can be divided into three forms, based on how the dynamics of the DAE are threaded (Verheyleweghen and Jäschke, 2017b). That is single shooting, multiple shooting and direct collocation. The direct collocation method will be used to solve this particular optimization problem. In this manner, the state trajectories are approximated by orthogonal polynomials (Diehl, 2011). Further details of the direct collocation method will not be discussed here.

The discretization of a continuous problem refers to dividing the time into a fixed set of intervals. Assume that the initial time horizon, t_f, can be separated into N number of time periods and that each time step is denoted k. For this reason, the continuous objective function from Equation 2.4 can be estimated by a Riemann sum by separating the time horizon into a finite number of distinct points,

$$\int_0^{t_f} \Phi\big(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}\big) dt = \sum_{k=1}^{N} \Phi_k(\mathbf{x}(t_{k+1}), \mathbf{z}(t_{k+1}), \mathbf{u}(t_k), \mathbf{p})\Delta t_k. \tag{2.7}$$

$\mathbf{x}(t_{k+1})$ and $\mathbf{z}(t_{k+1})$ represents the values of the differential and algebraic variables at the termination of the time period k. Δt_k denotes the duration of time period k. $\mathbf{x}_0$ is presumed to be provided as it is not a decision variable. Hence, $\mathbf{x}$ and $\mathbf{z}$ are evaluated at t_{k+1} and the input $\mathbf{u}$ is sampled at t_k.

The dynamic optimization problem defined by Equation 2.6 together with the new objective function in Equation 2.7 give rise to a set of optimization problems. This implies that the optimization problem in each time step k is only conditional on information from previous time steps. The following Equation 2.8 displays an advanced formulation of the dynamic optimization problem (Biegler, 2010):

$$\begin{aligned}
\min_{\mathbf{x},\mathbf{z},\mathbf{u}} \quad & \sum_{k=1}^{N} \Phi_k(\mathbf{x}_{k+1}, \mathbf{z}_{k+1}, \mathbf{u}_k, \mathbf{p})\Delta t_k, \\
\text{s.t.} \quad & \mathbf{f}_k(\mathbf{x}_k, \mathbf{z}_k, \mathbf{u}_k, \mathbf{p}) = \mathbf{x}_{k+1}, \quad \forall k = 1, .., N \\
& \mathbf{g}_k(\mathbf{x}_k, \mathbf{z}_k, \mathbf{u}_k, \mathbf{p}) = 0, \quad \forall k = 1, .., N \\
& \mathbf{x}_{\text{lb}} \leq \mathbf{x} \leq \mathbf{x}_{\text{ub}}, \quad \forall k = 1, .., N \\
& \mathbf{z}_{\text{lb}} \leq \mathbf{z} \leq \mathbf{z}_{\text{ub}}, \quad \forall k = 1, .., N \\
& \mathbf{u}_{\text{lb}} \leq \mathbf{u} \leq \mathbf{u}_{\text{ub}}, \quad \forall k = 0, .., N-1.
\end{aligned} \tag{2.8}$$

Equation 2.8 has made some abbreviations in terms of notation. $\mathbf{y}(t_k)$ is reduced to $\mathbf{y}_k$, in which y is one of the variables.

2.1.2 Dynamic Stochastic Optimization

In the area of process control, uncertainties can arise from system measurements or model mismatch. Problems of optimization under uncertainty are characterized as stochastic. A stochastic problem formulation is considered to ensure robustness against disturbance and uncertainty in the system model. Dynamic stochastic optimization will be core for the optimization routine for the Åsgard subsea gas compression station.

There are several techniques for incorporating uncertainty in optimization problems. In the context of this work, scenario decomposition techniques are employed to account for uncertainty in physical parameters in the optimization routine for the subsea system (Lucia et al., 2013b). A scenario-based approach to uncertainty will convert the distributions for the uncertain physical parameters, **p**, to discrete values by having a finite number of parameter realizations (Lucia et al., 2013b; Hans et al., 2015). A scenario is a combination of different parameter realizations with associated probability of occurrence as illustrated in Figure 2.3. The scenario will act as a path from the root to the leaf of the scenario tree (Verheyleweghen and Jäschke, 2017a).

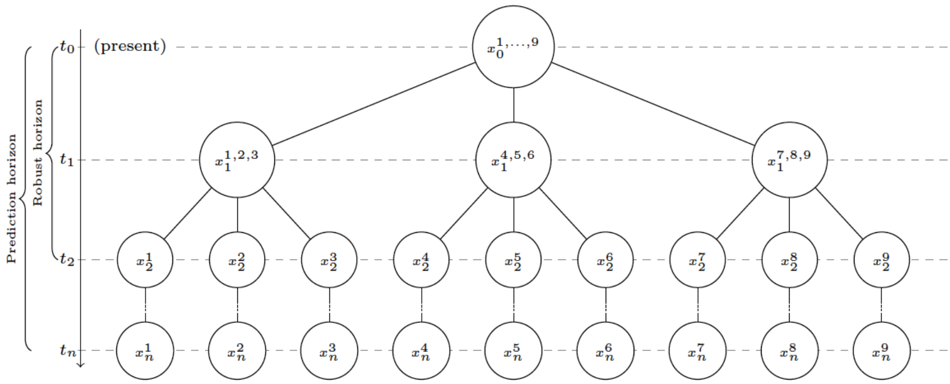


Figure 2.3: Scenario tree with robust horizon N_R = 2, prediction horizon N = n and number of scenarios S = 9 (Verheyleweghen and Jäschke, 2017c).

It is rather challenging to create a scenario tree that captures all aspects of the uncertainty in the system. However, it is preferable to build the tree as small as possible for complex optimization problems. The size of the optimization problem will grow exponentially with the number of uncertain parameters evaluated and the prediction horizon (number of stages, N). A robust horizon, N_R is introduced to limit the problem by branching the tree until a certain stage (Lucia et al., 2013a). Then, the uncertainty is assumed to be constant until the end of the prediction horizon. In the context of this work, the scenario tree is generated using combinations of minimum, maximum and expected uncertain parameter realization (Lucia et al., 2013b).

The deterministic equivalent of the dynamic stochastic optimization problem with a scenario-based approach to uncertainty can be expressed as

$$
\begin{aligned}
\min_{\mathbf{x}_{l,k},\mathbf{u}_{l,k},\mathbf{z}_{l,k}} \quad & \sum_{l=1}^{S} p_l \sum_{k=1}^{N} \Phi_k(\mathbf{x}_{l,k+1}, \mathbf{z}_{l,k+1}, \mathbf{u}_{l,k}, \mathbf{p})\Delta t_k, \\
\text{s.t.} \quad & \mathbf{f}_{l,k}(\mathbf{x}_{l,k}, \mathbf{z}_{l,k}, \mathbf{u}_{l,k}, \mathbf{p}) = \mathbf{x}_{l,k+1}, && \forall\, l = 1,...S,\ k = 1,..,N \\
& \mathbf{g}_k(\mathbf{x}_{l,k}, \mathbf{z}_{l,k}, \mathbf{u}_{l,k}, \mathbf{p}) = 0, && \forall\, l = 1,...S,\ k = 1,..,N \\
& \mathbf{x}_{\text{lb}} \leq \mathbf{x} \leq \mathbf{x}_{\text{ub}}, && \forall\, l = 1,...S,\ k = 1,..,N \\
& \mathbf{z}_{\text{lb}} \leq \mathbf{z} \leq \mathbf{z}_{\text{ub}}, && \forall\, l = 1,...S,\ k = 1,..,N \\
& \mathbf{u}_{\text{lb}} \leq \mathbf{u} \leq \mathbf{u}_{\text{ub}}, && \forall\, l = 1,...S,\ k = 0,..,N-1 \\
& \sum_{l=1}^{S} \mathbf{A}_{l,k}\mathbf{u}_{l,k} = 0, && \forall\, l = 1,...S,\ k = 1....N.
\end{aligned}
\tag{2.9}
$$

In Equation 2.9, S denotes the number of scenarios and p_l is the probability of occurrence for scenario l. **A** represents the non-anticipativity constraints, which are imposed such that decisions at the nodes in the scenario tree which are based on the same information are equal (Lucia et al., 2013b). Figure 2.4 illustrates how the non-anticipativity constraints are enforced between the connecting nodes in the scenario tree. Scenario decomposition is a technique for solving large multistage problems by relaxing the non-anticipativity constraints and solving the resulting scenario sub-problems in parallel. Increasing penalties are added in the sub-problems which will eventually ensure non-anticipativity. The advantage of this method is that the sub-problems are much smaller and easier to solve. The drawback is that convergence of the master problem can be very slow.

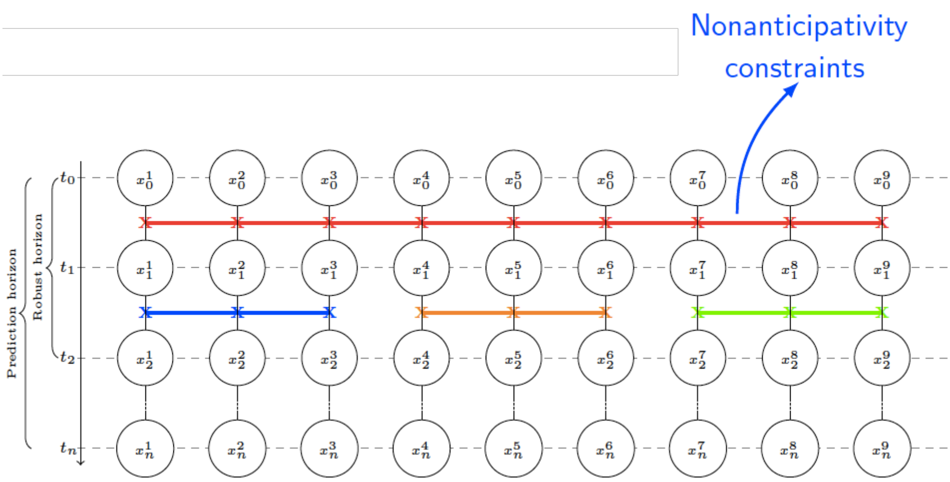


Figure 2.4: Scenario tree with robust horizon N_R = 2, prediction horizon N = n and number of scenarios S = 9 illustrating the connection between the non-anticipativity constraints. (Verheyleweghen and Jäschke, 2017c).

The dynamic stochastic optimization problem given in Equation 2.9 may be solved with a nonlinear programming (NLP) solver such as IPOPT (Wächter and Biegler, 2006).Note that the uncertainty that is taken into consideration in the stochastic problem formulation is the uncertainty in physical parameters. In the context of this study, the time of failure for the subsea station is assumed to be a stochastic variable as well. Uncertainties in regard to system reliability will be discussed in Chapter 3.

2.2 Model Predictive Control

Model predictive control (MPC) is an optimization-based control strategy often employed in process industries (Lucia et al., 2013b). It is an advanced method of process control recognized for its excellent ability for controlling complex systems. The MPC principle is based on repeated optimization of the model of the plant, subject to constraints (Morari and Lee, 1999). The model predictive controller applies the model of the system to predict its future behaviour and optimize future inputs (Lucia et al., 2013b).

Model predictive control is a combination of optimal control and a closed-loop method. Optimal control is generally referred to as open-loop control. Open-loop control solves an optimal control problem (OCP) and computes a sequence of input signals (Lucia et al., 2013b). The sequence of input signals obtained from the open-loop optimization is applied to the actual system. In general terms, assuming that the dynamic optimization problem is solved with a direct method, the OCP can be formulated as Equation 2.8 (Biegler, 2010). In the context of this study, a dynamic stochastic optimization problem is implemented to obtain efficient solutions with scenario based methods to account for uncertainty in physical parameters (Lucia et al., 2013b). Because of this, the optimal control problem that is solved in the open-loop optimization can be written as Equation 2.9.

In model predictive control, the optimal control is linked with a closed-loop method to overcome deviations between the predicted and the actual behavior of the system. Deviations arise due to disturbances and model mismatch. The closed-loop method implements the first control input achieved from the open-loop optimization. In addition, the most recent measurements from the open loop-optimization will act as new initial conditions for the differential states. In the context of this work, the closed loop solves the open-loop optimization problem repeatedly with a receding time horizon. This is done by decreasing the prediction horizon by one time step for each open-loop optimization (Seborg et al., 2010). The closed-loop simulation introduces random disturbance on control inputs to obtain the optimal control strategy for operation of the subsea system.

Figure 2.5 illustrates the interplay of the open-loop optimization and the closed-loop method of the actual plant. The figure illustrates how the model predictive controller predicts the optimal inputs after systematically resetting the initial conditions in the optimal control problem to the most recent measurements. The most recent input is implemented as the first control input in the optimal control problem. The open-loop optimization is done to predict optimal control inputs for a particular prediction horizon, N (Verheyleweghen and Jäschke, 2017c).

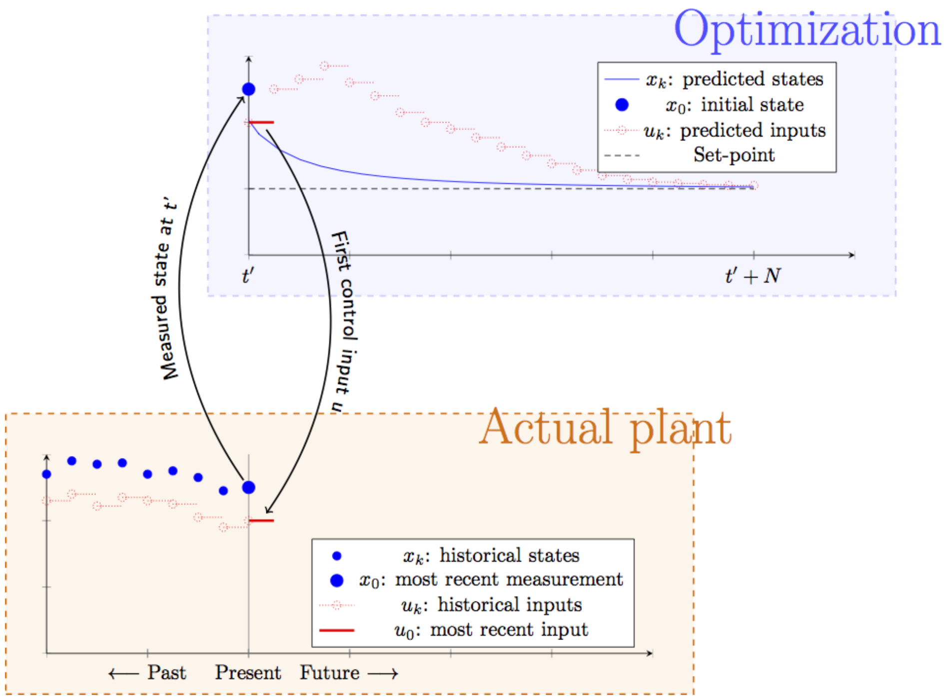


Figure 2.5: Illustration of the sequence of events in a model predictive controller (Verheyleweghen and Jäschke, 2017c).

Chapter 3

Reliability and Risk Management

Safe and efficient operation of subsea plants imposes strict requirements both with respect to equipment design and reliability. Information about the condition of the system can be used to make effective maintenance policies and to forecast the health condition of system components in the future. Condition monitoring techniques can be applied to evaluate the health of the subsea system in real-time. This study suggests to combine condition monitoring methods and optimal control. In doing so, the obtained optimal control strategy for operation will seek to ensure safe operation. This makes it possible to forecast the health of the system and manage the operation accordingly, rather than just reacting to it.

In recent years, there have been several attempts at combining condition monitoring techniques and model predictive control (MPC). Condition monitoring methods are coupled with prognostics and health monitoring (PHM) in order to improve the maintenance policy so that the predicted remaining useful life (RUL) of equipment can be increased. There have been several attempts at incorporating PHM in the objective function or the constraints in the OCP. The work carried out in the project thesis investigated the use of accumulated compressor degradation to estimate the condition of equipment for the Åsgard subsea gas compression station (Ims, 2017). However, by imposing hard constraints on the degradation of equipment, essential factors for ensuring a reliable operation are omitted. The degradation of equipment ignores the potential loss of production. Loss of production is defined as when RUL of equipment is shorter than the time until the next maintenance engagement (Verheyleweghen and Jäschke, 2018). RUL of equipment is a random variable that accounts for uncertainty in equipment reliability. Because of the stochastic nature of reliability and degradation, it is impossible to set hard constraints on the RUL of equipment (Verheyleweghen and Jäschke, 2018). Regular chance constraints on RUL of equipment can be imposed to ensure that the probability of RUL of equipment being greater than the time until the next maintenance intervention, is sufficient. However, optimal control problems with regular chance constraints neglect the effect of occurrence of extreme events. Safe control and decision making operations require the attention of unlikely events that can yet have disastrous consequences if realized (Singh et al., 2017).

The theory of financial risk estimates, which stem from the area of stochastic finance, can be applied to transcend the limitations of regular chance constraints in optimal control problems (Herceg et al., 2017). The combination of financial risk assessments and MPC is commonly referred to as risk-averse MPC. The application and design of risk-averse MPC has in recent years been introduced into the field of process engineering (Herceg et al., 2017). This study will investigate the use a risk-based OCP formulation to quantify the effect of tail risk, that is, the impact of extreme RUL of equipment outcomes. Risk assessments will be included into the optimization of the subsea station for condition monitoring purposes. Percentile limitations on RUL of equipment in the form of risk measures will be investigated as means for steering system reliability in real-time. Section 3.1 will discuss the discovery of faults in the system which are applied as health indicators for the RUL of equipment. Section 3.2 will elaborate on necessary properties for an acceptable risk measure and suggest suitable risk measures to employ for condition monitoring. Section 3.3 will discuss the application of risk measures in the optimization routine of the Åsgard subsea gas compression station.

3.1 Diagnostics and Prognostics

This study will apply risk monitoring techniques in conjunction with optimal control in order to limit the risk of failure. Failure can be any kind of unavailability of the system. Unavailability can be interpreted as the degree to which a system or component is not operational and accessible when required for use (Geraci et al., 1991). The concept of diagnostics deals with the discovery and surveillance of faults and hazards in a system (Verheyleweghen and Jäschke, 2017a). Prognostics on the other hand concerns the ability to anticipate health development and estimate the RUL of equipment (Verheyleweghen and Jäschke, 2017a). A variety of diagnostics and prognostics techniques are applied to monitor vulnerable parts in subsea systems. For example, measurements of electrical resistance can be applied to estimate corrosion and erosion rates (Verheyleweghen and Jäschke, 2017a). Vibration monitoring of rotating instruments are generally employed to evaluate faults on the impeller blades, shaft and bearings (Heng et al., 2009).

In order to limit the scope of this thesis, a simplifying assumption has been made that only the most vulnerable components in the system are considered. For that reason, a characteristic health indicator hi is considered for diagnostics of the condition of the equipment. Chapter 4 will give a description of hi specific to the Åsgard subsea gas compression station. Furthermore, propagation models for RUL of equipment in the form of risk measures will be investigated for condition monitoring. The random variable RUL of equipment is denoted ψ and is assumed to be Weibull distributed. The Weibull distribution is a commonly used distribution in reliability engineering with probability density function (Song et al., 2017),

$$f_\psi(t) = \lim_{dt \to 0} \frac{\mathrm{P}(t \leq \psi < t + dt)}{dt} = \begin{cases} \frac{K_w}{\lambda_w}\left(\frac{t}{\lambda_w}\right)^{(K_w-1)} e^{-\left(\frac{t}{\lambda_w}\right)^{K_w}} & t \leq 0 \\ 0 & t > 0, \end{cases} \tag{3.1}$$

where $K_w = K_w(hi)$ is the shape parameter and $\lambda_w = \lambda_w(hi)$ is the scale parameter (Jiang and Murthy, 2011).

3.2 Risk Measure

In recent years, attention has been paid to financial risk assessments and their ability to manage risk in areas outside finance. One interesting feature in particular is that financial risk measures can be expressed as percentile conditions for random variables. For that reason, risk measure formulations can be considered as prognostics models for condition monitoring of subsea plants. A risk measure can be designed to quantify the random RUL of equipment, $\psi = \psi(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{p})$, by a functional $R : \psi \to \mathbb{R}$ that can function as a substitute for gross RUL distribution (Capolei et al., 2015). Consequently, $R(\psi(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{p}))$ is referred to as a risk measure with respect to RUL of equipment. Risk quantification allows for efficient decision processes. Specifically, risk assessment of two RUL scenarios ψ_1 and ψ_2; implies comparison of the numerical values of $R(\psi_1)$ and $R(\psi_2)$.

R is a substitute for the distribution of ψ, of which various R formulations comprehend with different aspects of the RUL distribution (Capolei et al., 2018). The quality of the risk assessment depends on the traits of the risk measure in question. Hence, it is significant to have a particular set of attributes that define a capable risk measure. In the context of this work, the coherence and aversion axioms introduced by Artzner et al. (1999); Rockafellar (2007); Krokhmal et al. (2011) will be of interest.

3.2.1 Coherent Averse Measures of Risk

Coherent averse measures of risk are functionals $R : \psi \to \mathbb{R}$. Axiomatic analysis of risk measures was proposed by Artzner et al. (1999):

A1 Risk aversion:

- $R(c) = -c$ for constants c (constant equivalence)
- $R(\psi) >$ -$E[\psi]$ for non-constant ψ (aversion).

A2 Positive homogeneity:
R($\lambda\psi$) = λ $R(\psi)$ for all ψ and all constants $\lambda > 0$

A3 Sub-additivity:
$R(\psi_1 + \psi_2) \leq R(\psi_1) + R(\psi_2)$ for all ψ_1 and ψ_2

A4 Closure:
$\forall c \in \mathbb{R}$, the set $\{\psi | R(\psi) \leq$ c$\}$ is closed

A5 Monotonicity
$R(\psi_1) \geq R(\psi_2)$ when $\psi_2 \geq \psi_1$

It is vital to elaborate on these axioms in order to better grasp the underlying concepts of a proper risk measure. Axiom (A1) expresses the principle of risk aversion. A risk-averse controller does not have confidence in the expected value of a stochastic variable, $E[\psi]$, and prefers a deterministic value for RUL. The risk of a deterministic RUL yields the following relation: $R(c)$ = -c, which implies $R(E[\psi])$ = -$E[\psi]$. This means that $R(\psi) >$ -$E[\psi]$ can be rephrased to $R(\psi) > R(E[\psi])$ for $\psi \neq c$ and constant c (Capolei et al., 2015).

The positive homogeneity axiom (A2) guarantees consistency under scaling. In financial risk management, positive homogeneity suggests that the risk of a portfolio is proportional to its magnitude (Klüppelberg et al., 2014). In the context of this work, this axiom implies that the risk of failure is proportional to the control input. However, this might not be the case for complex subsea systems. In addition, if units of ψ are converted to a new currency, the risk is unambiguously scaled accordingly. Hence, this axiom facilitates that the units of measurements of $R(\psi)$ are equal to those of ψ (Capolei et al., 2015).

The sub-additivity axiom (A3) conveys the fundamental principle for risk attenuation through diversification (Capolei et al., 2015). In financial risk management, sub-additivity suggests diversification to be beneficial. Hence, the risk of adding two separate portfolio risks are always riskier than the risk of two joint portfolios (Klüppelberg et al., 2014). Axiom (A3) in conjunction with the constant equivalence attribute from axiom (A1), $R(c)$ = -c, results in the property of translational invariance,

$$R(\psi + c) = R(\psi) - c. \tag{3.2}$$

In financial risk management, translation invariance suggests that adding a particular quantity of funds reduces the risk by the same amount (Klüppelberg et al., 2014). The translation invariance principle presents a reasonable approach for defining a satisfactory risk (Artzner et al., 1999; Rockafellar, 2007). The closure axiom (A4) implies that the risk measure, $R(\psi)$, is finite and continuous (Rockafellar and Uryasev, 2013).

The monotonicity axiom (A5) implies that ψ_1 is viewed as riskier than ψ_2, given that all possible realizations of ψ_2 is greater than every realization of ψ_1 (Capolei et al., 2015). In terms of financial risk management, monotonicity suggests that a portfolio with greater future returns on investments has less risk. That is, if portfolio P_1 constantly has worse values than portfolio P_2 under almost all scenario realizations, then the risk of P_1 ought to be greater than the risk of P_2 (Klüppelberg et al., 2014). In the context of this work, this axiom implies that if the controller input $\mathbf{u}_1$ is more gentle than input $\mathbf{u}_2$, then $\mathbf{u}_1$ will yield less risk. In general, this axiom suggests that small valve openings and low compressor speed should provide little risk of failure.

Risk measures that comply with axioms (A1)-(A4) are referred to as averse measures of risk (Krokhmal et al., 2011; Rockafellar, 2007). However, risk measures that fulfill axioms (A2)-(A5) and the constant equivalence property in axiom (A1) are referred to as coherent risk measures as stated in Artzner et al. (1999); Krokhmal et al. (2011). Note that if risk measures comply with the positive homogeneity axiom (A2) and the sub-additivity axiom (A3), it implies convexity of the risk measure in question (Rockafellar, 2007; Krokhmal et al., 2011). As previously mentioned, the convexity feature is rather significant in optimization problems, as it permits the optimizer to locate globally optimal solutions (Capolei et al., 2015). In the context of this study, ψ = $\psi(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{p})$ is non-convex in terms of the control input $\mathbf{u}$. Hence, the optimization problem is non-convex and the use of local/convex solvers can only be expected to yield local minimums.

In literature, multiple coherent averse risk measures have been investigated for the purpose of optimization. Coherent averse risk measures such as Conditional Value-at-Risk (CVaR) introduced by Rockafellar and Uryasev (2000) are universally accepted for managing financial risk in optimization problems. CVaR is an extension to Value-at-Risk (VaR) introduced by Morgan, JP (1994); Jorion (2006). However, VaR is not a coherent averse risk measure and does not qualify as a proper risk measure. Despite this, the underlying concepts of VaR are rather essential for the interpretation of CVaR.

3.2.2 Value-at-Risk

Value-at-Risk (VaR) is possibly the most renowned risk measure in financial risk management in present time (Morgan, JP, 1994; Jorion, 2006). It can be interpreted as the minimum expected value for a random variable given a certain confidence level, α. In the context of this study, VaR is estimated with respect to remaining useful life (RUL) of equipment, ψ. Assuming that the probability density function (PDF) for ψ given in Equation 3.1 is continuous and strictly monotonic. The cumulative distribution function (CDF), F_ψ, can be written as

$$F_\psi(x) = \mathrm{P}[\psi \leq x] = \int_{-\infty}^{x} f_\psi(t)dt. \tag{3.3}$$

Note that P is the probability operator and f_ψ is the probability density function of ψ. The Value-at-Risk with confidence level α ∈(0,1) of a random RUL variable, ψ, is defined as (Jorion, 2006)

$$\mathrm{VaR}_\alpha(\psi) = q_\psi(\alpha). \tag{3.4}$$

$q_X(\alpha)$ is the quantile with confidence level α. The quantile function specifies, for a given probability α in the probability distribution of the random variable, the value x for which P[$\psi \leq x$] = α. Hence, the quantile function can be mathematically expressed as (Rockafellar and Uryasev, 2000)

$$\begin{aligned} q_\psi(\alpha) &= \inf\{x \in \mathbb{R} : \mathrm{P}[\psi \leq x] = \alpha\} \\ &= \inf\{x \in \mathbb{R} : F_\psi(x) = \alpha\} \\ &= F_\psi^{-1}(\alpha) \\ &= x. \end{aligned} \tag{3.5}$$

$F_\psi^{-1}(\alpha)$ is the inverse cumulative distribution function (ICDF) with confidence level α. This results in the following expression for calculating Value-at-Risk at level $\alpha \in$ (0,1) of a random RUL of equipment variable, ψ:

$$\mathrm{VaR}_\alpha(\psi) = q_\psi(\alpha) = \inf\{x \in \mathbb{R} : F_\psi(x) = \alpha\} = F_\psi^{-1}(\alpha). \tag{3.6}$$

Conceptually, $\mathrm{VaR}_\alpha(\psi)$ denotes the minimum expected value for RUL of equipment with a confidence level, α. This risk measure is defined in such a way that the probability of values for RUL of equipment greater than $\mathrm{VaR}_\alpha(\psi)$ is less than or equal to α. Thus, the chance of a values for RUL of equipment less than $\mathrm{VaR}_\alpha(\psi)$ is less than or equal to 1-α. Figure 3.1 illustrates the RUL of equipment distribution with the value for VaR_α explicitly marked at α.

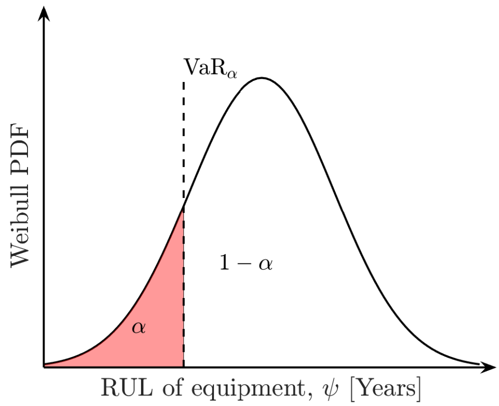


Figure 3.1: Illustration of a probability density function of RUL of equipment, ψ, with the value for VaR_α explicitly marked at confidence level, α.

VaR_α is expressed as a quantile in Equation 3.6 which acts as a chance constraints on RUL of equipment (Krokhmal et al., 2011). This is also referred to as the failure probability constraint in the area of reliability (Rockafellar and Royset, 2010). However, VaR_α does not consider the tail of the RUL distribution. The RUL outcomes beneath the α-quantile are not taken into account when calculating VaR_α. As a consequence, extreme RUL outcomes are neglected which may result in catastrophic consequences. In addition, VaR_α lacks highly desired properties such as convexity and sub-additivity, which may limit its application (Artzner et al., 1999). This will not be elaborated on as it is considered to be out of the scope of this work.

3.2.3 Conditional Value-at-Risk

Conditional Value-at-Risk (CVaR) is introduced as an extension to Value-at-Risk to overcome deviations in VaR calculations. CVaR was introduced by Rockafellar and Uryasev (2000) and fulfills all axioms for a coherent averse measure of risk according to section 3.2.1. Rockafellar and Uryasev (2002) defined CVaR_α as the average of VaR_α,

$$\begin{aligned} \mathrm{CVaR}_\alpha(\psi) &= \frac{1}{\alpha}\int_0^\alpha \mathrm{VaR}_\gamma(\psi)d\gamma \\ &= \frac{1}{\alpha}\int_0^\alpha F_\psi^{-1}(\gamma)d\gamma. \end{aligned} \tag{3.7}$$

CVaR is as an extension of VaR and serves as an approximation of the chance constraint in Equation 3.6. The new risk measure serves with the same purpose, which is to limit, with confidence level α, the probability of having RUL of equipment greater than the time until the next maintenance engagement. However, CVaR_α, in contrast to VaR_α, considers the tail of the RUL distribution beneath the α-quantile. CVaR_α calculates the average RUL that occur beneath VaR_α which implies that the most unlikely and worst possible outcomes are emphasized for low values of α. Essentially, CVaR estimates risk in a more conservative manner by bringing the extreme RUL outcomes into focus. Figure 3.2 illustrates the RUL of equipment distribution with values for CVaR_α and VaR_α explicitly marked at α.

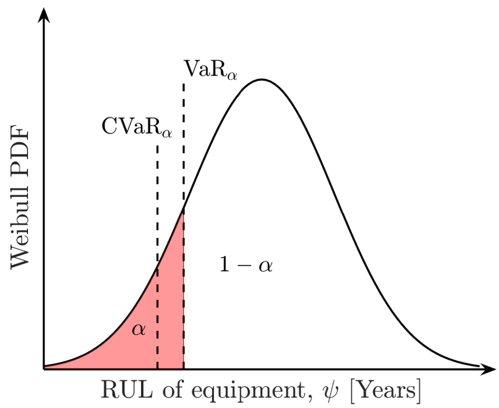


Figure 3.2: Illustration of a probability density function of RUL of equipment, ψ, with the values for VaR_α and CVaR_α explicitly marked at confidence level, α.

Note that both VaR_α and CVaR_α are examples of static risk measures which are calculated for one time period. The next section will present dynamic risk measures as a natural extension to static risk measures.

3.2.4 Dynamic Risk Measure

In financial risk management, dynamic risk measures are applied in dynamic portfolio selection problems, in which investment arrangements can change over time (Chen et al., 2017). Dynamic risk measures, also referred to as multi-period risk measures, are risk estimates that reckon with a longer time horizon than the static. Dynamic risk measures are risk estimates which are adjusted as new information becomes available. More explicitly, the risk measures are conditional on available information at the time of the risk evaluation (Acciaio and Penner, 2011). In this study, dynamic risk measures will be applied in the dynamic optimal control problem in the model predictive controller, in which the optimal input strategy may change over time.

Dynamic decision problems are often expressed as discrete, multi-stage, control problems (Chen et al., 2017). Because of this, the dynamic risk function considered here will be evaluated at a number of distinct points. The dynamic risk function will adopt the same notation as a dynamic optimization problem from Section 2.1.1, in which k denotes a particular time period and N is the number of time steps. As a consequence, a dynamic risk measure is a conditional risk function that can be defined at time k as $R_{k,N}$ (Ruszczyński, 2010). As time progresses from the startup of the plant at k = 1 to $k = N$, the risk function $\{R_{k,N}\}_{k=1}^{N}$ will provide an estimate of the risk associated with the remaining time until the next maintenance engagement.

Section 3.2.1 introduced the coherence and aversion axioms as necessary characteristics for a proper static risk measure. Similar axiomatic analysis is essential when shifting to a dynamic environment. A dynamic risk measure, $\{R_{k,N}\}_{k=1}^{N}$, is a conditional risk function that must attain axioms (A1)-(A5) for each time-interval k = 1,...,N, in order to qualify as a coherent averse risk measure (Chen et al., 2017). However, additional features must be examined when shifting to a dynamic environment in a model predictive controller. Conditions like information monotonicity and dynamic time consistency are significant particularly in relation to the optimal control problem (Chen et al., 2017). The principle of information monotonicity is used to differentiate risk measures subject to various information processes. Information processes are captured by so called filtrations, $F_k \in \mathbb{F}$, that represent the information available at time k. The conditional risk measure under an arbitrary filtration process at time k is denoted by $R_{k,N}(\psi_{k,N}|\{F_k, ...F_N\})$, where $\psi_{k,N}$ = $(\psi_k, ..., \psi_N)$ denotes the RUL process over the periods from k to N. A risk measure is said to be F_k-adapted if the risk assessment at time k, is independent of information to be disclosed in the future. A dynamic risk measure is said to be information monotone according to the following definition (Pflug and Romisch, 2007)

Definition 3.2.1. A dynamic risk measure $\{R_{k,N}\}_{k=1}^{N}$ is information monotone if for any two filtrations $\{F_1, .., F_N\}$ and $\{F_1^{'}, .., F_N^{'}\}$ we have that $F_s \in F_s^{'}$, $s = k, ..., N$ and $R_{k,N}(\psi_{k,N}|\{F_k^{'}, .., F_N^{'}\}) \leq R_{k,N}(\psi_{k,N}|\{F_1, .., F_N\})$.

In financial risk management, Definition 3.2.1 supports the concept that for a given portfolio, more accessible information will never cause a rise in risk disclosure but usually give more effective control of risk (Chen et al., 2017). The idea of information monotonicity is analogous to the principle of non-anticipativity introduced in Section 2.1.2.

Dynamic time consistency deals with consistency in the form of both risk measures and optimal control strategies. Wang (1999) originally described the concept of dynamic time consistency as risk assessments where past and future evaluations do not contradict each other (Riedel, 2004). Essentially, consistency over time ensures that subsequent knowledge will not affect past risk evaluations or control solutions (Chen et al., 2017). The idea is based on a rather simple understanding: given two input policies $\mathbf{u}_1$ and $\mathbf{u}_2$, if $\mathbf{u}_1$ is riskier than $\mathbf{u}_2$ under a specific risk measure in the future, then $\mathbf{u}_1$ is riskier than $\mathbf{u}_2$ under the same measure today (Wang, 1999). A dynamic risk measure $R_{k,N}$ is dynamically time consistent according to the following definition (Chen et al., 2017)

Definition 3.2.2. If for any two points on the time horizon, $k = 1 < \tau < \theta \leq N$, and input policies $\mathbf{u}_1$ and $\mathbf{u}_2$, the condition $R_{\theta,N}(\psi_{\theta,N}(\mathbf{u}_1)) \leq R_{\theta,N}(\psi_{\theta,N}(\mathbf{u}_2))$ implies that $R_{\tau,N}(\psi_{\tau,N}(\mathbf{u}_1)) \leq R_{\tau,N}(\psi_{\tau,N}(\mathbf{u}_2))$, then the dynamic risk measure, $\{R_{k,N}\}_{k=1}^{N}$, is dynamically time consistent.

Adopting the concept in Definition 3.2.2 to control would suggests that the optimal control policy determined at t = 0 indicates its optimality in the future as well (Chen et al., 2017). However, in practice, the optimal input policy for an optimal control problem may fail to satisfy dynamic time consistency. Moreover, in most cases, the dynamic time consistency of optimal control policies relies on the consistency of the dynamic risk measure. Implementing a dynamic time consistent risk measure in the optimal control problem will most often result a dynamically time consistent optimal strategy for inputs (Chen et al., 2017). On the other hand, applying a dynamic time inconsistent risk measure will give a inconsistent optimal input policy.

Multi-period risk measures existing in literature today may be sorted into three categories: terminal, additive and recursive risk measures (Chen et al., 2017). Terminal risk measures are risk measures formulated in terms of the terminal outcome of the RUL of equipment. Additive risk measures arise when the risk evaluations are performed separately in different time periods before they are combined as one. The main difference between terminal and additive is that in the case of terminal risk measures the period-wise losses of profit are aggregated prior to applying the risk measure. However, the additive risk measure aggregates the risk measures instantaneously. There is a catch to the previous method: most terminal risk measures are not dynamically time consistent (Chen et al., 2017). Since additive risk measures are simple extensions if terminal risk measures, the additive risk measures may also be dynamically time inconsistent. Finally, recursive risk measures arise from assessing dynamic risk exposure over time recursively. Unfortunately, incorporating recursive risk measures to dynamic optimization problems may lead to rather complex numerical matters (Chen et al., 2017).

Terminal

$$R_{k,N}(\psi_{k,N}) \;=\; \varphi_k\big(\psi_N|F_N\big) \tag{3.8}$$

Additive

$$R_{k,N}(\psi_{k,N}) \;=\; \sum_{k=1}^{N} \beta^k \, \varphi\big(\psi_k|F_k\big) \tag{3.9}$$

Recursive

$$R_{k,N}(\psi_{k,N}) \;=\; \sum_{s=k+1}^{N} E\big[\varphi_s \, \big(\psi_s|F_{s-1}\big)|F_k\big] \tag{3.10}$$

ψ_k represents the distribution of RUL of equipment at time k. φ denotes an arbitrary static risk measure, i.e. $\varphi = CVaR_\alpha$. F_k represents filtration at time k. β^k is a discount factor from $k = 1, ..., N$.

3.3 Risk Control

Up to this point, the main focus has been risk measures and their properties. Furthermore, this thesis will stress the importance of the risk measure's sustainability for the formulation of risk control problems. Risk measures will be incorporated into the optimal control problem to obtain an optimal control strategy for operation in-line with specified risk preferences.

Rockafellar and Uryasev (2000, 2002) presented Conditional Value-at-Risk (CVaR) as a means for estimating and minimizing risk in control problems. CVaR estimates the risk in a more conservative manner by bringing the extreme RUL outcomes into focus as it calculates the average of the α-percent lowest profit realizations. Consequently, CVaR is excellent at attenuating unlikely events which can yet have disastrous effects if realized to enforce safe control and decision-making operations. CVaR adheres to the coherence and aversion axioms proposed by Artzner et al. (1999); Rockafellar (2007); Krokhmal et al. (2011) and qualifies as proper risk measure. For this reason, CVaR will be applied for calculating the risk of failure with respect to RUL of equipment.

As seen in previous sections, the adaption of static risk measures to a multi-stage setting is fairly complicated. The characteristics of the dynamic risk measures for multi-stage problems are essential to ensure efficient risk control. Information monotonicity, coherence and dynamic time consistency arise as significant features for dynamic risk measures in optimal control (Chen et al., 2017). By adapting an additive multi-period risk measure formulation of CVaR into the model predictive control scheme, the concepts of information monotonicity and coherency are assumed to be satisfied. However, the additive form of CVaR fails to fulfill dynamic time consistency. The time-inconsistency of the dynamic risk measure, will give a dynamically time inconsistent optimal control strategy for operation and thus result in a sub-optimal solution. However, the degree of sub-optimality might be small, and can be tolerated by the decision maker in exchange for an optimization problem that is much easier to solve. For that reason, the concept of dynamic time consistency will be given a lower priority and the additive risk measure will be applied for estimating the risk of failure in optimal control. The resulting risk estimate that will be employed in the optimization can be expresses as

$$\begin{aligned} R_{k,N}(\psi_{k,N}) &= \sum_{k=1}^{N} \beta^k \, \mathrm{CVaR}_\alpha\big(\psi_k | F_k\big) \\ &= \sum_{k=1}^{N} \beta^k \, \frac{1}{\alpha} \int_0^\alpha \mathrm{VaR}_\gamma(\psi_k | F_k) d\gamma \\ &= \sum_{k=1}^{N} \beta^k \, \frac{1}{\alpha} \int_0^\alpha F_{\psi_k}^{-1}(\gamma) d\gamma. \end{aligned} \tag{3.11}$$

For simplicity, the α-domain is discretized by assuming that $\lim \Delta\gamma \to 0$ and the final risk estimate can be expressed as

$$R_{k,N}(\psi_{k,N}) = \sum_{k=1}^{N} \left(\beta^k \, \frac{1}{\alpha} \sum_{j=0}^{\alpha} \left(F_{\psi_k}^{-1}(j) \, \Delta\gamma \right) \right). \tag{3.12}$$

Equation 3.12 formulates the dynamic risk measure that will be included into the optimization of the subsea station for condition monitoring purposes. This is a type of percentile limitations on RUL of equipment that will be integrated into the optimal control problem to ensure safe operation in in real-time. Chapter 5 will elaborate on how this particular risk measure is integrated in the OCP formulation.

Chapter 4

Model Development

The underlying process explored in this work is the subsea gas compression station at the Åsgard field. A brief process description of the subsea plant will be given in Section 4.1. Section 4.2 will present the equation structures and fundamental assumptions for the derivation of the model for the subsea plant. The models for the choke, the separator and the compressor are discussed in this chapter. The model equations for the compressor and the choke valve are provided by Verheyleweghen and Jäschke (2017b). The separator model is developed as part of the study with the project thesis (Ims, 2017). The model equations and corresponding assumptions are included in this thesis to better grasp the fundamental process explored in this study.

4.1 Process Description

The subsea gas compression station at the Åsgard field is the very first compressor to be installed and operated on the seabed. It is considered to be pioneering compression technology (Setekleiv et al., 2016). The purpose of the gas compression station is to boost the pressure of the reservoir stream such that it will surpass pressure drop in transportation pipes to topside facilities. However, the maturity level of the technology is limited for multiphase (Verheyleweghen and Jäschke, 2017b). Hence, the gas and liquid components are separated to allow an increase in pressure. A process diagram of the subsea gas compression station is illustrated in Figure 4.1. The system consists of a well choke that controls the reservoir stream entering the gas compression station. A separator downstream from the well choke separates liquid from gas. Incomplete separation causes liquid droplets to exit the separator with the gas through the gas outlet. The liquid pressure is subsequently boosted by a pump and the gas pressure is increased in a wet-gas compressor (Verheyleweghen and Jäschke, 2017b).

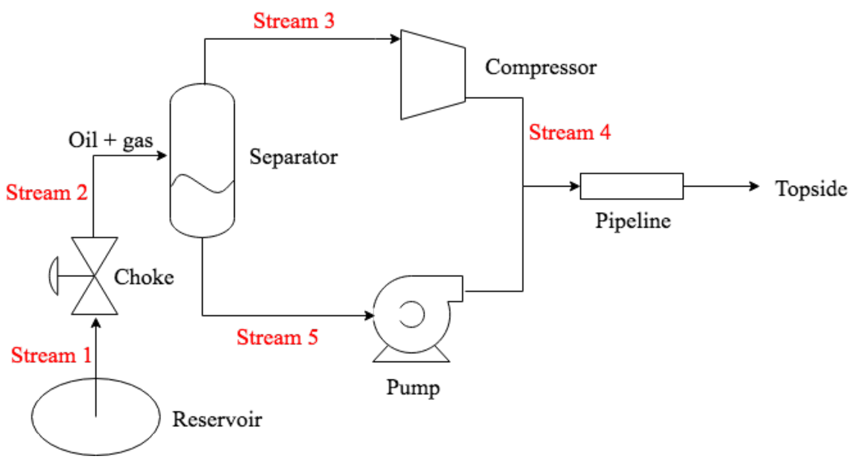


Figure 4.1: Process diagram of the subsea gas compression station in the Åsgard field adapted from Verheyleweghen and Jäschke (2017a).

4.2 Model Description

This section will describe the equations defining the model for the Åsgard subsea gas compression station. The models for the choke, the separator and the compressor are discussed in this section. Simplified thermodynamics are used for the compressor model. Also, the model for the degradation of the bearings in the wet-gas compressor is reviewed here. The pipeline and the pump are not modeled, but will be included in future work. Lastly, it is assumed that the fluid in the system can be described as liquid and gas.

4.2.1 Choke

The system make use of a well choke to enable control of the flow of hydrocarbons from the reservoir. The mass flow through a valve is assumed to be given by the standard valve equation (Grimholt and Skogestad, 2015),

$$\dot{m}_1 = f(z)\, C_d\, A_{choke}\, \sqrt{\rho_1 (P_1 - P_2)}. \tag{4.1}$$

$\dot{m}_1$ is the mass flow rate entering the valve and C_d represents the valve constant. P_1 is the inlet pressure and P_2 represents the outlet pressure from the valve. A_{choke} denotes the cross sectional area of the valve and ρ_1 is the density of the flow. $f(z)$ is the valve characteristics, where z denotes the valve opening. z ranges between 0 and 1 when completely open (Grimholt and Skogestad, 2015). In the context of this work, linear valve characteristics is assumed,

$$f(z) = z, \qquad \text{where} \;\; z \in [0, 1]. \tag{4.2}$$

The choke opening is a control input and is denoted $z = u_{choke}$. For simplicity, A_{choke} and ρ_1 are assumed to be constant across the valve. The resulting flow through the valve is given by a simplified valve equation,

$$\dot{m}_1 = \dot{m}_{1,l} + \dot{m}_{1,g} = u_{choke} c_{choke} \sqrt{P_1 - P_2}, \tag{4.3}$$

where c_{choke} is a choke constant, and u_{choke} denotes the opening of the choke valve (Verheyleweghen and Jäschke, 2017b). $\dot{m}_{1,l}$ and $\dot{m}_{1,g}$ denotes inlet mass flow rate for liquid and gas, respectively.

4.2.2 Separator

The presented separator model is developed as part of the project on ”Modelling of Åsgard subsea gas compression station for condition monitoring purposes”, which was conducted by Julie Berge Ims during the autumn of 2017 (Ims, 2017). The content of this section is based on said project, but is repeated here for the convenience of the reader.

A variety of well developed separator models are available for gas-liquid separation. In this particular case, it is essential to develop a detailed separator model to be able to accurately predict liquid carry over. A detailed model reduces uncertainty and provides opportunity to shift constraints in the optimal control problem. Shifting constraints can make the operation less conservative and thus more profitable. In terms of subsea operating conditions, the separator model must also be able to handle higher pressure and higher flow rates. In this regard, the separator model developed for the subsea gas compression station at the Åsgard field is based on a Statoil patented separator for liquid-gas separation of an inlet flow which predominantly contains gas. The separator unit is developed to be able to separate the last liquid droplets from a gas flow, both at high flow rates and high pressure (Fredheim et al., 2013).

The Statoil patented separator consists of a spinlet inlet configuration and axial flow cyclones (AFCs) (Fredheim et al., 2013; Aguilera and Carlui, 2013). The separator is a vertical standing vessel with an inlet for the liquid-gas flow and outlets for gas- and liquid flows. The inlet is a spinlet arrangement for flow distribution to receive and make the flow move in rotational movements around the vertical axis of the main container towards a porous pipe configuration. The axial flow cyclone exploits centripetal forces to separate light and heavy components in the fluid. The fluid is likely to follow a helical path where heavier components will accumulate at the outer peripheral of the helical trail, while lighter components will gather in the center along the vertical axis. Gravitational forces will also contribute to separate heavier components, whereas lighter components may rise towards the gas outlet (Fredheim et al., 2013).

The separator may contain a wired mesh demister in the gas outlet between the container wall and the upper end of the tubular wall (Fredheim et al., 2013). An illustration of a separator with wired mesh pads is shown in Figure 4.2. However, due to the maturity level of the technology, mesh pads are currently not considered an option for subsea processing systems due to the risk of clogging (Setekleiv et al., 2016).

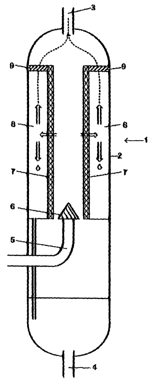


Figure 4.2: Illustration of a separator unit (with mesh pads) patented by Statoil (Fredheim et al., 2013).

A steady state model was developed by Austrheim (2006) for a scrubber with a mesh pad used for primary separation and axial flow cyclones for separating the last droplets from the gas stream. In view of the risk of clogging for subsea processing systems, this study will only concentrate on the axial flow cyclone section of the steady state model. A mathematical model based on flow development, fluid properties and cyclone geometry has been developed to correlate the dimensionless re-entrainment number and separation efficiency in a cyclone (Austrheim, 2006). This mathematical correlation has been fundamental in this particular separator model.

Re-entrainment Number

The performance degradation of the AFC applied in Austrheim (2006) was dominated by some type of re-entrainment mechanism rather than insufficient separation of small droplets. The separation efficiency is governed by the re-entrainment of liquid which has settled on the separator wall. Various mechanisms for re-entrainment of liquid into a gas stream is described in Austrheim (2006). Figure 4.3 illustrates the re-entrainment mechanisms "Roll wave" and "Wave undercut".

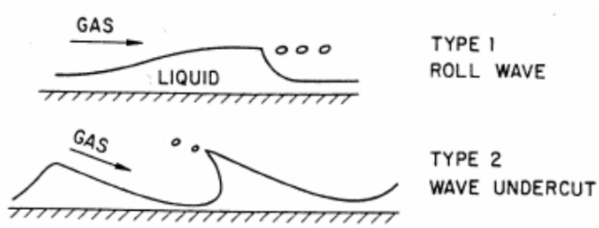


Figure 4.3: Re-entrainment mechanisms in the axial flow cyclone (Austrheim, 2006).

The ”Roll wave” mechanism is associated with droplets that are cut of a roll-wave peak. This is the dominant mechanism in liquid film with high Reynolds number and in the transition regime (Austrheim, 2006). The ”Wave undercut” mechanism is connected to cutting a wave peak. This is a governing mechanism in liquid film with relatively low Reynolds number. In the context of this study, it assumed that the liquid film on the cyclone wall is in the transition regime. A force balance was applied as a criterion for the eruption of re-entrainment. The retaining force, F_σ, of the surface tension, σ, between the two phases was evaluated with the drag force from the gas flow on the liquid wave peak, F_d. Roll wave re-entrainment was presumed to be feasible if the drag force acting on the wave top exceeded the retaining force (Austrheim, 2006),

$$F_d \geq F_\sigma. \tag{4.4}$$

The outburst of such re-entrainment mechanism depends on the Reynolds number of the liquid film, Re_L, on the cyclone wall and the dimensionless viscosity number, N_μ (Austrheim, 2006). The criterion for the eruption of entrainment in the transition regime was expressed as

$$\begin{aligned} \frac{\mu_l\ u_{g,s}}{\sigma}\sqrt{\frac{\rho_g}{\rho_l}} &\geq 11.78 N_\mu^{0.8}\ \mathrm{Re}_L^{-1/3} \quad && \text{for } N_\mu \geq \frac{1}{15}, \\ \frac{\mu_l\ u_{g,s}}{\sigma}\sqrt{\frac{\rho_g}{\rho_l}} &\geq 1.35 \mathrm{Re}_L^{-1/3} && \text{for } N_\mu \leq \frac{1}{15}. \end{aligned} \tag{4.5}$$

In Equation 4.5, ρ_l is the density in the liquid film on the cyclone wall and ρ_g is the gas density. $u_{g,s}$ is the superficial gas velocity. μ_l denotes the viscosity of the liquid film and σ denotes the interfacial tension between the liquid and the gas phase. Furthermore, it is assumed that liquid carry-over is a constant fraction of entrained liquid. Thus the liquid flow, $\dot{Q}_l$, on the cyclone wall must be corrected for this (Austrheim, 2006). Hence, the expression for the Reynolds number, Re_L for the liquid film on the cyclone wall results in:

$$Re_L = \frac{\rho_l\ u_l \delta_l}{\mu_l} = \frac{\rho_l\ \Gamma}{\mu_l} = \frac{\rho_l\ \dot{Q}_l\ \alpha_s}{\mu_l\ \mathrm{P}_w}. \tag{4.6}$$

Neither the liquid film thickness, δ_l, nor the liquid film velocity, u_l, are known at this stage. The product of the two quantities, Γ, is the volumetric liquid flow, $\dot{Q}_l$, per unit wetted perimeter, P_w. The liquid flow, $\dot{Q}_l$, is assumed to be constant and equal to 10% of the volumetric gas flow, $\dot{Q}_g$. ($\dot{Q}_l \alpha_s$) is the corrected volumetric liquid flow. α_s is here the separation efficiency in the axial flow cyclone. The wetted perimeter of the cyclone, P_w, must take into account the direction of the gas flow (Austrheim, 2006). If the cyclone body is flattened to a rectangle, the circumference in the container is the length of the short side as depicted in Figure 4.4.

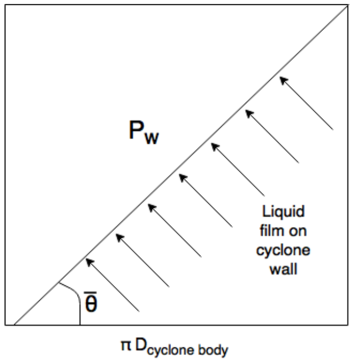


Figure 4.4: Illustration of the lower section of the cyclone when it is flattened. The wetted perimeter of the cyclone is marked as the diagonal. The figure is adapted from Austrheim (2006).

The wetted perimeter can be defined with the equation

$$\mathrm{P}_w = \frac{\pi\ D}{\cos\hat{\theta}}, \tag{4.7}$$

where the angle, $\hat{\theta}$, is used to indicate the direction of the gas flow. It denotes the relative angle to the swirl and is assumed to be approximately $\hat{\theta}$ = 45°. D is the diameter of the cyclone. Furthermore, the force balance in equation 4.4 accounts for changes in shear stress acting on the liquid wave due to the drag force from the gas flow through the dimensionless viscosity number, N_μ (Austrheim, 2006). This parameter is used to analyze the viscous force induced by internal flow. The viscosity number is defined through the following relation:

$$N_\mu = \frac{\mu_l}{\sqrt{\rho_l\ \sigma\ \sqrt{\frac{\sigma}{a_l\ \Delta\rho}}}}. \tag{4.8}$$

$\Delta\rho = \rho_l - \rho_g$ and a_l is the centrifugal acceleration acting on the liquid film,

$$a_l = \frac{2\ u_{l,tg}^2}{D}, \tag{4.9}$$

where the tangential velocity component of the liquid film, $u_{l,tg}$, is unknown at this stage. The tangential components of the shear stress acting on the wall due to the liquid film and on the liquid film due to the gas are $\tau_{w,tg}$ and $\tau_{i,tg}$, respectively. The tangential shear stresses are defined as

$$\tau_{i,tg} = f_{g,i}\frac{\rho_g\ u_{r,tg}^2}{2}, \tag{4.10}$$

$$\tau_{w,tg} = f_{l,w}\frac{\rho_l\ u_{l,tg}^2}{2} = \tau_{i,tg}. \tag{4.11}$$

Assumptions about the gas velocity relative to the liquid film velocity are defined as

$$u_{g,tg} >> u_{l,tg} \Rightarrow u_{r,tg} \approx u_{g,tg}. \tag{4.12}$$

Based on these assumptions the tangential liquid velocity can be expressed as

$$u_{l,tg} \;=\; \sqrt{\frac{f_{g,i}\;\rho_g\;u_{g,tg}^2}{f_{l,w}\;\rho_l}}. \tag{4.13}$$

$u_{g,tg}$ is the tangential gas velocity which will be discussed later. The f's are friction factors which have not yet been measured for liquid flow on a cyclone wall (Austrheim, 2006). However, friction factors developed for annular flow in pipes were used by Austrheim (2006) and the same approximation is done in this study as well. $f_{g,i}$ is the friction factor for gas on the liquid film and is expressed through the following relation:

$$f_{g,i} = 0.005\left[1 + 300\frac{2\delta_l}{D}\right]. \tag{4.14}$$

$f_{g,i}$ is the friction factor for liquid on the wall and is expressed as

$$f_{i,w} = \left(K \cdot Re_L^m\right)^2. \tag{4.15}$$

$K = 3.73$ and $m = -0.47$ for $2 < Re_L < 100$. $K = 1.962$ and $m = -1/3$ for $100 < Re_L < 1000$. Note that this study will assume that the liquid film on the cyclone wall is in the transition regime. Consequently, K and m for the friction factor for liquid on the cyclone wall are equal to 1.926 and -1/3, respectively. The friction factors depend on the thickness of the liquid film, δ_l, on the cyclone wall. Liquid film thickness, δ_l, can be found from

$$\Gamma \;=\; \frac{\dot{Q}_l}{P_w} = u_l\;\delta_l \quad \Rightarrow\; \delta_l = \frac{\dot{Q}_l}{P_w\;u_l}, \tag{4.16}$$

where liquid film velocity can be expressed as

$$u_l = \frac{u_{l,tg}}{cos(\hat{\theta})}. \tag{4.17}$$

The expression for the liquid film thickness, δ_l, can thus be simplified based on Equation 4.7, 4.16 and 4.17,

$$\delta_l = \frac{\dot{Q}_l}{\pi\;D\;u_{l,tg}}\cos^2\hat{\theta}. \tag{4.18}$$

The tangential gas velocity, $u_{g,tg}$, increases with radius, similar to a solid body rotation. The gas viscosity is low relative to the liquid. Hence, the velocity profile for the tangential gas velocity close to the cyclone wall will resemble a loss-free-vortex profile (Austrheim, 2006). The tangential gas velocity in the cyclone can therefore be considered as something between a loss-free vortex and a solid body rotation. However, the gas velocity at the liquid-gas interface on the cyclone wall is more important for re-entrainment analyzes. The wall gas velocity is illustrated in Figure 4.5 where θ is the angle of the tangential gas velocity at the cyclone wall. θ is assumed to be constant and equal to 45^o.

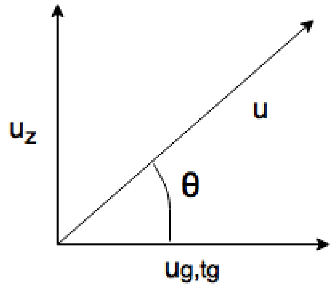


Figure 4.5: Illustration of flow coordinates at the cyclone wall adapted from Austrheim (2006).

Furthermore, the superficial gas velocity, $u_{g,s}$, is assumed to be a factor 0.8 less than the vertical gas velocity, u_z, close to the cyclone wall in the middle section of the cyclone (Austrheim, 2006). For this reason, the superficial gas velocity can be calculated with respect to the tangential gas velocity, $u_{g,tg}$, and θ according to

$$u_{g,s} = 0.8\ \big(u_z\big) = 0.8\ \big(u_{g,tg} \cdot \tan(\theta)\big). \tag{4.19}$$

Based on all these expressions, a dimensionless re-entrainment number, E, has been developed to characterize the cyclone separation efficiency, α_s, where the separation is governed by re-entrainment (Austrheim, 2006),

$$E(\alpha_s, u_{l,tg}, a) = \frac{\frac{\mu_l\ u_{g,s}}{\sigma}\left(\frac{\rho_g}{\rho_l}\right)^{0.8}}{N_\mu^a\ \mathrm{Re}_L^{-1/3}}. \tag{4.20}$$

Excellent correlation between the cyclone separation efficiency and the dimensionless re-entrainment number may indicate that the separation efficiency is governed by liquid re-entrainment, not insufficient separation of smaller droplets (Austrheim, 2006). The correlation between the cyclone separation efficiency and the dimensionless re-entrainment number is expressed as

$$\alpha_s = A \cdot E(\alpha_s, u_{l,tg}, a)\ +\ B. \tag{4.21}$$

a is a constant used to fit the re-entrainment number with the separation efficiency. It proved to be appropriate with a = 0.4 for this model. A and B are constants for the linear model. In the context of this work, A and B, are assumed to be equal to -0.1345 and 1.01, respectively.

Numerous approximations are made for this separation unit. One assumption in particular is the fact that the inlet flow predominantly contains gas. In reality, the separator should be able to handle various GVFs and tangential gas velocities. The physical properties of the gas and liquid phase will also vary. This should be addressed in future work.

4.2.3 Compressor

The system is equipped with a compressor in order to increase the pressure in the gas flow downstream from the separator. The compressor is modeled as a standard polytropic compressor (Verheyleweghen and Jäschke, 2017b). The polytropic relation is given by

$$\frac{T_4}{T_3} = \left(\frac{P_4}{P_3}\right)^{\frac{1}{w}}. \tag{4.22}$$

T_4 and T_3 is the outlet and inlet temperature to the compressor, respectively. The outlet and inlet pressure to the compressor are denoted P_4 and P_3, respectively. w is given in terms of the adiabatic correlations,

$$w = \eta \cdot \frac{\gamma}{1-\gamma}, \tag{4.23}$$

$$\gamma = \frac{1}{2}\left(\frac{C_{p,3}}{C_{p,3} - \mathrm{R}} + \frac{C_{p,4}}{C_{p,4} - \mathrm{R}}\right). \tag{4.24}$$

Here, R is the standard gas constant and η is the compressor efficiency. $C_{p,3}$ and $C_{p,4}$ denote the heat capacity for the inlet and outlet gas stream, respectively. The heat capacity, C_p, can be a expressed as a polynomial with respect to temperature, T, using the expression

$$C_p = (b_1 + b_2 T + b_3 T^2 + b_4 T^{-4})\mathrm{R}, \tag{4.25}$$

where T is the temperature in the respective stream. b_1 - b_4 are polynomial parameters based on the chemical composition in the stream. The compressor efficiency, η, may be expressed in terms of the volumetric flow upstream to the compressor, q_3, and the compressor speed, u_{comp},

$$\eta = f(q_3, u_{comp}) = \frac{c_1\,\hat{q}^2 + c_2\,\hat{q} + c_3}{\hat{q}^2 + c_4\,\hat{q} + c_5}, \tag{4.26}$$

$$\hat{q} = \frac{q_3}{u_{comp.}}. \tag{4.27}$$

c_1 - c_5 are polynomial parameters. The function f is given by a polynomial fit to the compressor map from Aguilera and Carlui (2013). This is unique to each compressor. Furthermore, the compressor head, H, is given by

$$H = w\frac{Z \cdot \mathrm{R} \cdot (T_4 - T_3)}{gM}, \tag{4.28}$$

where g denotes the gravitational constant and M represents the molar mass in the stream (Verheyleweghen and Jäschke, 2017b). Z is the compressibility factor of the gas upstream to the compressor. Moreover, the compressor head, H, can also be expressed as a polynomial function of the volumetric inlet flow to the compressor, q_3, according to the following relation:

$$H = \left(c_6\hat{q}^2 + c_7\,\hat{q} + c_8\right) \cdot f_{wood}. \tag{4.29}$$

f_{wood} is the Woods correction factor. c_6 - c_8 are polynomial parameters. The compressibility factor, Z, can be found by utilizing Dranchuk and Abou-Kassems equation of state (Dranchuk et al., 1975),

$$
\begin{aligned}
Z = 1 &+ \left(A_1 + \frac{A_2}{T_{pr}} + \frac{A_3}{T_{pr}^3} + \frac{A_4}{T_{pr}^4} + \frac{A_5}{T_{pr}^5}\right) \cdot \sigma_c \\
&+ \left(A_6 + \frac{A_7}{T_{pr}} + \frac{A_8}{T_{pr}^2}\right) \cdot \sigma_c^2 - \left(\frac{A_7}{T_{pr}} + \frac{A_8}{T_{pr}^3}\right) \cdot \sigma_c^5 \cdot A_9 \\
&+ A_{10}(1 + A_{11}\sigma_c^2)\left(\frac{\sigma_c^2}{T_{pr}^3}\right)\exp(-A_{11}\sigma_c^2).
\end{aligned} \tag{4.30}
$$

A_1-A_{11} are polynomial parameters for the compressibility factor. σ_c is given by

$$\sigma_c = 0.27\left(\frac{P_{pr}}{T_{pr}}\right)Z. \tag{4.31}$$

P_{pr} and T_{pr} represents the pseudo-reduced pressure and temperature, respectively. Kay (1936) proposed that the pseudo-reduced pressure, P_{pr}, can be calculated using simple mole relations,

$$P_{pr} = \frac{P_3}{P_{pc}} = \frac{P_3}{\mathbf{P}_c \times \mathbf{c}_3}, \tag{4.32}$$

where P_3 is the pressure upstream to the compressor, P_{pc} is the pseudo-critical mixture temperature, $\mathbf{P}_c$ are the critical temperatures for the components in the mixture and $\mathbf{c}_3$ is the composition of chemical compounds in the stream. Kay (1936) proposed an equivalent calculation method for the pseudo-reduced temperature, T_{pr},

$$T_{pr} = \frac{T_3}{T_{pc}} = \frac{T_3}{\mathbf{T}_c \times \mathbf{c}_3}, \tag{4.33}$$

where T_3 is the temperature upstream to the compressor, T_{pc} is the pseudo-critical mixture temperature and $\mathbf{T}_c$ are the critical temperatures for the components in the stream. It is apparent that Z in Equation 4.30 is conditional on Z into σ_c, which itself relies on Z. Consequently, the equations explaining the model give rise to a semi-implicit index-1 DAE (Verheyleweghen and Jäschke, 2017b).

f_{woods} is a Woods correction factor, that considers liquid at the inlet of the wet gas compressor (Hundseid et al., 2008).

$$f_{wood} = \frac{1}{\frac{\rho_{3,avg}}{\rho_3}\sqrt{GVF_3 \cdot \frac{\rho_{3,avg}}{\rho_3}}}. \tag{4.34}$$

ρ_3 is the density upstream to the compressor. The average density, $\rho_{3,avg}$, of the wet gas compressor entry is

$$\rho_{3,avg} = GVF_3 \cdot \rho_3 + (1 + GVF_3)\rho_{3,l}. \tag{4.35}$$

$\rho_{3,l}$ represents the density of the condensate in the gas stream entering the compressor. GVF_3 denotes the gas-volume fraction of the gas stream upstream to the compressor,

$$GVF_3 = \frac{q_{3,g}}{q_{3,g} + q_{3,l}}, \tag{4.36}$$

where $\rho_{3,g}$ represents the density of the gas in the gas stream entering the compressor. Furthermore, the compressor power can be calculated using the energy balance

$$Pow = \frac{H\ q_3\ \rho_3\ g}{\eta}. \tag{4.37}$$

In addition, compressor surging or choking are undesired physical phenomena which may occur in a wet gas compressor (Verheyleweghen and Jäschke, 2017b). For this reason, the variables Srg and Stw are employed to signal surge as Stonewall conditions (compressor choking), respectively. Values less than zero for Srg and Stw suggest either surge or choke.

$$Srg = \hat{q} - q_{min} \tag{4.38}$$

$$Stw = q_{max} - \hat{q} \tag{4.39}$$

q_{min} denotes the minimum allowable flow in the compressor in order to prevent surge. q_{max} denotes the maximum allowable flow in the compressor order to avoid compressor choking.

Compressor Bearing Degradation

This particular system is complex with a large number of components for which diagnostics and prognostics can be challenging. In order to limit the scope of this thesis, a simplifying assumption has been made that only the most vulnerable components in the system are considered. The bearings in the wet-gas compressor are considered to be vital in the operation, and should be replaced immediately if broken. The bearings are prone to faults as they have multiple moving parts and a complex mechanical setup (Verheyleweghen and Jäschke, 2017a). For that reason, the only dynamics of interest is the wet-gas compressor bearing degradation model (Verheyleweghen and Jäschke, 2017b). The compressor bearings will degrade according to Paris' law of crack propagation. Paris' crack propagation model is commonly used for surface defects (Paris and Erdogan, 1963). This model states that the crack length, h, will develop according to

$$\frac{dh}{dn_{cycles}} \ = \ D_{comp} \cdot (\Delta K_{comp})^n, \tag{4.40}$$

where n is a numerical exponent, n_{cycles} is the number of cycles, D_{comp} is a material constant and ΔK_{comp} denotes the range of strain. This can be reformulated into a model for the development of a bearing crack length, h,

$$\frac{dh}{dt} \ = \ h \cdot \ c_{Paris} \left(T_{comp}^2 \cdot u_{comp}\right) \ = \ h \cdot \ c_{Paris} \cdot \left(\frac{Pow^2}{u_{comp}}\right), \tag{4.41}$$

where it is assumed that the torque, T_{comp}, can be used as health indicator for gross strain (Bechhoefer et al., 2008). c_{Paris} is a lumped parameter and is estimated from past values (Verheyleweghen and Jäschke, 2017a).

The bearing crack-length is applied as a health indicator for the remaining useful life (RUL) of equipment. RUL of equipment is assumed to be Weibull distributed with shape and scale-parameters. It is assumed that the shape parameter, K_w, and the scale parameter, λ_w, depend on degradation of equipment, h (Verheyleweghen and Jäschke, 2018),

$$K_w(h) = k_a \ + \ k_b \cdot h \ + \ k_c \cdot h^2 \tag{4.42}$$

$$\lambda_w(h) = \lambda_a - \lambda_b \cdot h - \lambda_c \cdot \sqrt{h} \tag{4.43}$$

Note that degradation of equipment is a function of inputs, h = $h(u)$. In the context of this study, system failures which are independent of operational decisions are neglected. This should however be addressed in future work.

Chapter 5

Optimal Control Problem Formulation

The optimal control problem (OCP) for the subsea gas compression station at the Åsgard field is formulated in the same manner as the dynamic stochastic optimization problem presented in Equation 2.9. Section 5.1- 5.3 will specify the objective function, constraints and bounds for the optimization of this particular subsea station. The full optimal control problem for the open-loop optimization will be presented in Section 5.4. The optimization is conducted with respect to a particular time horizon. The start-up of the plant is at t = 0. The next maintenance engagement is scheduled to take place at time $t = t_f$. At last, the parameters in the Weibull distribution for the remaining useful life (RUL) of equipment variable will be discussed in Section 5.5.

5.1 Objective Function

In context of this study, the main target with the optimization of this subsea system is to improve the economic outcome from the operation through cost reduction and increase in production. However, safe and efficient operation imposes stringent requirements with respect to equipment reliability. Hence, the ultimate objective is twofold:

1. Prevent premature failure of the subsea system

2. Maximize net profit from operation.

The first objective is referred to as the reliability objective, ϕ_r. The reliability objective is defined as minimizing unavailability of the system in terms of loss of production. Chapter 3 presented Conditional Value-at-Risk (CVaR) as a risk measure estimate for assessing the risk of failure. Consequently, minimizing unavailability of the system corresponds to maximizing CVaR with respect to RUL of equipment, ψ, for a given confidence level, α, over the expected lifetime of the operation. The reliability objective can be formulated as

$$
\begin{aligned}
\phi_r = -R_{k,N}(\psi_{k,N}) &= -\sum_{k=1}^{N} \beta^k \ \mathrm{CVaR}_\alpha\big(\psi_k|F_k\big) \\
&= -\sum_{k=1}^{N} \beta^k \ \frac{1}{\alpha}\int_0^\alpha \mathrm{VaR}_\gamma(\psi_k|F_k)d\gamma \\
&= -\sum_{k=1}^{N} \beta^k \ \frac{1}{\alpha}\int_0^\alpha F_{\psi_k}^{-1}(\gamma)d\gamma \\
&= -\sum_{k=1}^{N} \bigg(\beta^k \ \frac{1}{\alpha}\sum_{j=0}^{\alpha}\Big(F_{\psi_k}^{-1}(j)\ \Delta\gamma\Big)\bigg).
\end{aligned}
\tag{5.1}
$$

The inverse cumulative distribution function (ICDF) for the Weibull distribution, F_ψ^{-1}, with respect to ψ, is the quantile function,

$$
F_\psi^{-1} = q_\psi(\alpha, \lambda_w, K_w) = \lambda_w(-\ln(1-\alpha))^{1/K_w}. \tag{5.2}
$$

The final reliability objective can be expressed as

$$
\begin{aligned}
\phi_r &= -\sum_{k=1}^{N} \beta^k \ \frac{1}{\alpha}\int_0^\alpha F_{\psi_k}^{-1}(\gamma)d\gamma \\
&= -\sum_{k=1}^{N} \bigg(\beta^k \ \frac{1}{\alpha}\sum_{j=0}^{\alpha}\Big(F_{\psi_k}^{-1}(j)\ \Delta\gamma\Big)\bigg) \\
&= -\sum_{k=1}^{N} \bigg(\beta^k \ \frac{1}{\alpha}\sum_{j=0}^{\alpha}\Big(q_{\psi_k}(j, \lambda_w, K_w)\ \Delta\gamma\Big)\bigg) \\
&= -\sum_{k=1}^{N} \bigg(\beta^k \ \frac{1}{\alpha}\sum_{j=0}^{\alpha}\Big(\lambda_w(-\ln(1-j))^{1/K_w}\ \Delta\gamma\Big)\bigg).
\end{aligned}
\tag{5.3}
$$

Note that $\phi_r = \phi_r(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{p})$, $\lambda_w = \lambda_w(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{p})$ and $K_w = K_w(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{p})$. The second objective is referred to as the economic objective, $\phi_e = \phi_e(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{p})$. The economic objective can be written in the following manner (Verheyleweghen and Jäschke, 2018)

$$
\phi_e(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{p}) = \mathbb{E}\bigg(\int_0^{t_f} \mathrm{cost}(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{p}) \cdot c(t)\ dt\bigg). \tag{5.4}
$$

$\mathbb{E}$ is the expected value operator, cost($\mathbf{x}$,$\mathbf{z}$,$\mathbf{u}$,$\mathbf{p}$) is the cost associated with the states, inputs and parameters and $c(t)$ is the discounting term. The profit is weighted from time $t = 0$ to $t = t_f$ by discounting future value of money at a periodic rate of return, called the discount rate. This is a way to measure profit by including present and all future discounted cash flows, called the Net present value (NPV) (Kurt, 2016). Equation 5.5 gives a general definition of NPV:

$$\begin{aligned}\text{NPV}(\mathbf{x},\mathbf{z},\mathbf{u},\mathbf{p},i,N) &= \sum_{k=1}^{N}\Big(\text{cost}(\mathbf{x}_{k+1},\mathbf{z}_{k+1},\mathbf{u}_k,\mathbf{p})\cdot c(t)\Big)\\ &= \sum_{k=1}^{N}\Big(\text{cost}(\mathbf{x}_{k+1},\mathbf{z}_{k+1},\mathbf{u}_k,\mathbf{p})\cdot(1\,+\,i)^{-t_k}\Big).\end{aligned} \tag{5.5}$$

Here, i is the discount rate and N is the number of time periods. The objective is to maximize NPV of the production which is measured in terms of the gas production rate downstream from the compressor. Therefore, the economic objective can be expressed as

$$\begin{aligned}\phi_e(\mathbf{x},\mathbf{z},\mathbf{u},\mathbf{p}) &= \int_0^{t_f}\Big(-\frac{\dot{m}_{gas}(\mathbf{x},\mathbf{z},\mathbf{u},\mathbf{p})}{(1+i)^t}\Big)\,dt\\ &= \sum_{t=1}^{N}\Big(-\frac{\dot{m}_{gas}(\mathbf{x}_{k+1},\mathbf{z}_{k+1},\mathbf{u}_k,\mathbf{p})}{(1+i)^{t_k}}\Big).\end{aligned} \tag{5.6}$$

Note that $\dot{m}_{gas} = \dot{m}_{gas}(\mathbf{x},\mathbf{z},\mathbf{u},\mathbf{p})$. The dynamic stochastic optimization problem formulated in Equation 2.9 is formulated as a minimization problem. The economic objective and the reliability objective are matters of maximization and hence the negative of the objectives are applied. The reliability objective and the economic objective generate a multi-objective function in the dynamic stochastic optimization problem. Equation 2.9 employs scenarios to incorporate uncertainty in physical parameters into the optimization routine. For that reason, the scenario-based deterministic equivalent of the objective function can be formulated as

$$\begin{aligned}\sum_{l=1}^{S}\sum_{k=1}^{N}\Phi &= \sum_{l=1}^{S}\sum_{k=1}^{N}\phi_e+\omega\phi_r\\ &= \sum_{l=1}^{S}\sum_{k=1}^{N}\Big(-\frac{\dot{m}_{gas}}{(1+i)^{t_k}}\Big)+\omega\Big(-\beta^k\frac{1}{\alpha}\sum_{j=0}^{\alpha}\Big(\lambda_w(-\ln(1-j))^{1/K_w}\Delta\gamma\Big)\Big).\end{aligned} \tag{5.7}$$

ω is applied as a weighting factor between the two objectives. The weighing factor sets the decision maker's attitude towards risk. As a consequence, the choice of the weight of risk and the risk measure in the objective is fundamental for the nature and formulation of the problem (Dupačová and Kozmík, 2015; Kozmık, 2015). In the context of this work, the weighting factor is assumed to be constant and equal to 0.5. The discount term in the dynamic risk measure, β^k, is assumed to be constant in all time intervals. The value for β^k is assumed to be equal to 0.05.

5.2 Constraints

The subsea system is described by a set of nonlinear Differential Algebraic Equations (DAEs) given in Chapter 4. In order to enforce feasible solutions, the set of DAEs are incorporated into the optimization problem formulation as constraints on the algebraic equations, **g**, and the differential equations, **f**. Section 2.1.1 also made assumptions that the optimization problem could be formulated as an initial value problem. The initial bearing crack-length is set to 0.01 mm and the initial condition for the time variable is equal to 0. Consequently, the following constraints are imposed on system variables to enforce a feasible solution:

$$\begin{aligned}\mathbf{f}_{l,k}(\mathbf{x}_{l,k},\mathbf{z}_{l,k},\mathbf{u}_{l,k},\mathbf{p}) &= \mathbf{x}_{l,k+1}\\ \mathbf{g}_{l,k}(\mathbf{x}_{l,k},\mathbf{z}_{l,k},\mathbf{u}_{l,k},\mathbf{p}) &= \mathbf{x}_{l,k+1}\\ h_0 &= 0.01\\ t_0 &= 0.0.\end{aligned} \tag{5.8}$$

5.3 Upper and Lower Bounds

The optimal control problem with objective function given in Equation 5.7 and constraints given in Equation 5.8 may yield solutions which are not within a safe operating domain or physically not feasible. As a consequence, bounds are enforced on inputs, u_{comp} and u_{choke}, related to allowable operating range for flow through the compressor and the choke. Bounds on surge, Srg, and Stonewall, Stw, conditions for compressor choking according to the allowable operating range must be imposed. Limitations on P_{out} is necessary to ensure flow trough the pipeline to the topside (Verheyleweghen and Jäschke, 2017b). The resulting lower-and upper bounds are

$$\begin{aligned}0.75 \le u_{\text{comp}} &\le 1.05\\ 0 \le u_{\text{choke}} &\le 1\\ 0 &\le Srg\\ 0 &\le Stw\\ 150 \text{ bar} &\le P_{\text{out}}.\end{aligned} \tag{5.9}$$

5.4 Optimal Control Problem

The objective function defined by Equation 5.7 together with the constraints from Equation 5.8 and the variable bound in Equation 5.9 will give rise to a set of optimization problems,

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{z},\mathbf{u}} \quad & \sum_{l=1}^{S} p_l \sum_{k=1}^{N} \Big(-\frac{\dot{m}_{gas}}{(1+i)^{t_k}} \Big) + \omega \Big(-\beta^k \frac{1}{\alpha} \sum_{j=0}^{\alpha} \Big(\lambda_w (-\ln(1-j))^{1/K_w} \Delta\gamma \Big) \Big) \\
\text{s.t.} \quad & \mathbf{f}_{l,k}(\mathbf{x}_{l,k}, \mathbf{z}_{l,k}, \mathbf{u}_{l,k}, \mathbf{p}) = \mathbf{x}_{l,k+1} \\
& \mathbf{g}_{l,k}(\mathbf{x}_{l,k}, \mathbf{z}_{l,k}, \mathbf{u}_{l,k}, \mathbf{p}) = 0 \\
& h_0 \ = \ 0.01 \\
& t_0 \ = \ 0.0 \\
& 0.75 \ \leq \ u_{\text{comp}} \ \leq \ 1.05 \\
& 0 \ \leq \ u_{\text{choke}} \ \leq \ 1 \\
& 0 \ \leq Srg \\
& 0 \ \leq Stw \\
& 150 \text{ bar } \ \leq P_{\text{out}}.
\end{aligned}
\tag{5.10}
$$

The probability of occurrence, p_l, is assumed to be equal for all scenarios.

5.5 Weibull Parameters

The remaining useful life (RUL) of equipment variable is assumed to be Weibull distributed with some interesting parameters, the scale parameter, λ_w, and the shape parameter, K_w. Assumptions were made that both shape and scale parameters were dependent on the degradation variable, h. The characteristics of the scale and shape parameters are key in investigating the RUL of equipment distribution of the system. Different shape parameters affect the failure rate in the following manner (Jiang and Murthy, 2011):

1. $K_w < 1$ Suitable for modelling early failure due to problems with production

2. K_w = 0 Suitable for modelling failure due to pure coincidence

3. $K_w > 1$ Suitable for modelling wear-out failure due to degradation of equipment after some time

The shape parameter represents the slope of the Weibull distribution and in the context of this work, it is appropriate to use a positive shape parameter. The failures occurring are assumed to be ”wear-out”-failures as they will commence due to ”the aging process”. The scale parameter represents the variance of the Weibull distribution. The scale parameter was assumed to be greater than zero. Also, it is assumed to decrease over time since the RUL of equipment distribution has a greater variance earlier in the production. Due to the lack of failure data, the final relationship between the shape, K_w, and scale, λ_w, parameter and the crack length, h, is assumed to be

$$K_w = 4.55 \ + \ 0.1h \ + \ 0.1h^2, \tag{5.11}$$

$$\lambda_w = 5.7 - 1.2h - 2.8\sqrt{h}. \tag{5.12}$$

For implementation in a real systems, the parameter values must be adjusted to reflect the expected degradation profile of the given system. Historical data from the OREDA database or similar can be used for this purpose (OREDA Participants, 2002).

Chapter 6

Results and Discussion

This chapter will review the results of the performed study. Section 6.1 will give a brief overview of the model predictive control (MPC) scheme that is employed for optimization of the subsea station. Details on the implementation in MATLAB will be assessed in Section 6.2. Results obtained from the open-loop optimization will be analyzed in Section 6.3 At the end, the optimal control strategy for the subsea station achieved from closed-loop MPC simulation will be discussed in Section 6.4.

6.1 Model Predictive Control Framework

The ambition of this study is to employ a model predictive control (MPC)-like framework to obtain a control policy that maximizes the net present value (NPV) of production without jeopardizing the reliability of the subsea gas compression station at the Åsgard field. For that reason, health monitoring methods are applied to monitor the condition of the overall system in real-time. This study investigates the idea of risk minimization to manage condition monitoring in optimization of a subsea system. A risk measure that considers the risk of failure is used to assess the health of the subsea plant. Essentially, integrating risk monitoring techniques into the optimization procedure refers to ensuring that the subsea system stays operational until the next maintenance intervention. In the context of this work, the next maintenance engagement is scheduled to take place five years after the start-up of the plant. However, the simulation is conducted with initial time horizon t_f = 1 as a simplification in the calculations. Initial values for the two differential variables, wet-gas compressor bearing crack length, h, and time, t, was set to 0.01 mm at t = 0 years, respectively. The simulation was carried out with a fixed compressor strain on the bearing fault in the wet-gas compressor.

A scenario-based method is employed to account for the uncertainty in the physical parameter c_{Paris} in the bearing crack length propagation model in Equation 4.41. The scenarios represent discrete parameter realizations, namely the 90% percentile, the 10% percentile, and the nominal value. The stochastic optimal control problem (OCP) expressed in Equation 5.10 is solved with initial prediction horizon N = 20 and a robust horizon of N_R = 1 for the scenario tree. This OCP is solved in the open-loop optimization and computes a sequence of input signals to the actual plant. The closed loop MPC solves the optimization problem repeatedly with a receding time horizon. The closed-loop simulation introduces random disturbance on control inputs in order to obtain the optimal operational control strategy for the subsea system. The numerical results obtained in this study are based on the chosen set of parameters and assumptions for this system. Simulation parameters are listed in Appendix A.

6.2 Implementation in MATLAB

The original system is implemented in MATLAB by Adriaen Verheyleweghen and serves as the open-loop optimization algorithm in the MPC. The open-loop optimization problem is implemented in MATLAB using the open-source external software package CasADi (Andersson, 2013). The optimization problem is solved with IPOPT (Wächter and Biegler, 2006). The original script is modified for risk controlling purposes. Risk monitoring is enforced through risk measure estimates which are stated as algebraic equations and added into the original system. The closed-loop MPC is implemented to add random disturbance to the open-loop optimization. The initial separator model provided by Adriaen Verheyleweghen was altered with a new separator model developed through the study with the project thesis (Ims, 2017). The resulting MATLAB code is provided in Appendix B.

6.3 Open-loop Optimization with Risk Control

The model predictive controller repeatedly solves an open-loop optimization problem to predict the optimal control policy. The open-loop optimization algorithm is modified with risk monitoring techniques to assess the unavailability of equipment. Minimizing unavailability of equipment can be rephrased to maximizing an additive multi-period risk measure formulation of Conditional Value-at-Risk (CVaR) with respect to remaining useful life (RUL) of equipment. As a consequence, the optimal control strategy is obtained by maximizing profit and CVaR with respect to RUL of equipment over the expected lifetime of the operation.

6.3.1 Risk Control

Value-at-Risk and Conditional Value-at-Risk calculations in financial risk management usually employ a confidence level α = 1-5%. In the context of this study, the confidence level, α, is applied as a tuning parameter in the optimization problem. Table 6.1 presents the predicted values for VaR$_\alpha$ and CVaR$_\alpha$ at t = 0 obtained from the first open-loop optimization with a nominal value for the uncertain parameter c_{paris} = 1.0. The table shows an obvious trend that both VaR$_\alpha$ and CVaR$_\alpha$ are increasing with increasing α. Even though higher values for VaR$_\alpha$ and CVaR$_\alpha$ may imply shrinking probability for failure, the increasing level of confidence may be non satisfactory for subsea operation. The confidence level ought to be selected at a lower value to obtain a more reliable control strategy for the subsea plant.

Table 6.1: Value-at-Risk and Conditional Value-at-Risk for different confidence levels, α. The values are obtained from the first open-loop optimization at $t = 0$.

α [%]	VaR$_\alpha$ [Years]	CVaR$_\alpha$ [Years]
1	9.83	8.09
2	11.46	9.41
3	12.54	10.29
4	13.38	10.96
5	14.07	11.52

Confidence levels of 1-5% are assumed to be non satisfactory for subsea processing systems. Operating on the seabed gives rise to higher demands in terms of safety and reliability. Maintenance engagements are considerably rare for subsea systems, as it requires specialized intervention ships to carry out operations on the bottom of the ocean. Consequently, unplanned shutdowns which may cause expensive maintenance engagements are avoided at every opportunity. In this study, it is assumed that $\alpha = 0.1\%$ would give an acceptable risk level. Consequently, CVaR$_\alpha$ represents the average RUL of equipment of the 0.1% lowest RUL of equipment outcomes. Figure 6.1 illustrates the probability density function (PDF) of RUL of equipment with values for VaR$_\alpha$ and CVaR$_\alpha$ explicitly marked at $\alpha = 0.1\%$. The values are obtained from the first open-loop optimization at $t = 0$ with a nominal value for the uncertain parameter $c_{paris} = 1.0$.

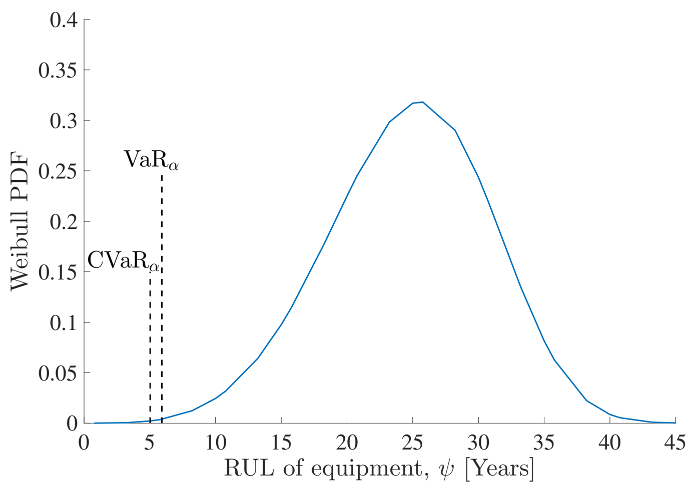
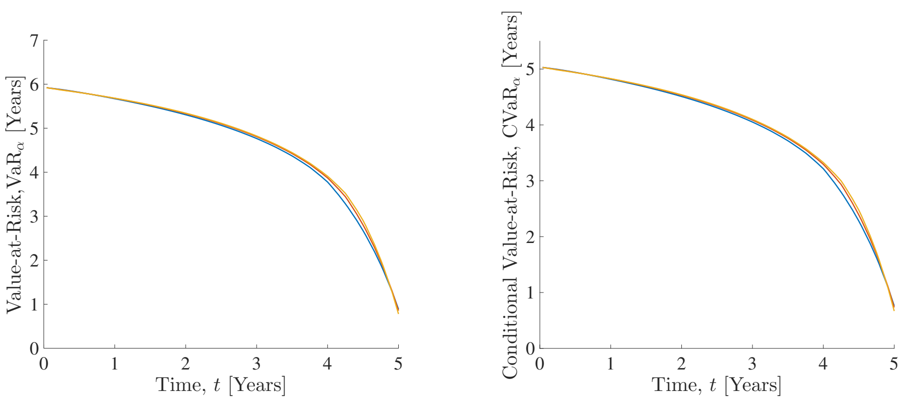


Figure 6.1: The probability density function of RUL of equipment, ψ, with values for VaR$_\alpha$ and CVaR$_\alpha$ explicitly marked at $\alpha = 0.1\%$. The values are obtained from the first open-loop optimization at $t = 0$.

Figure 6.1 depicts the predicted values at $t = 0$ for VaR_α and CVaR_α from the first open-loop optimization. The plot shows the that expected minimum value of RUL of equipment (VaR_α) is just below six years and that the average RUL beneath that (CVaR_α) is just above five years. This may imply that with this particular set of parameters and assumptions, the chance of RUL of equipment less than five years at $t = 0$ is sufficiently small for subsea operation. It is assumed that the obtained CVaR_α has a confidence level of $\alpha = 0.1\%$. The constraints are satisfied so the solution achieved from the optimization should produce a level of confidence for the risk of failure equal to $\alpha = 0.1\%$. The actual level of confidence can be checked by running Monte Carlo simulations to test if CVaR_α represents the average RUL of the 0.1 % lowest RUL outcomes for the given time horizon. This should be verified in future work.

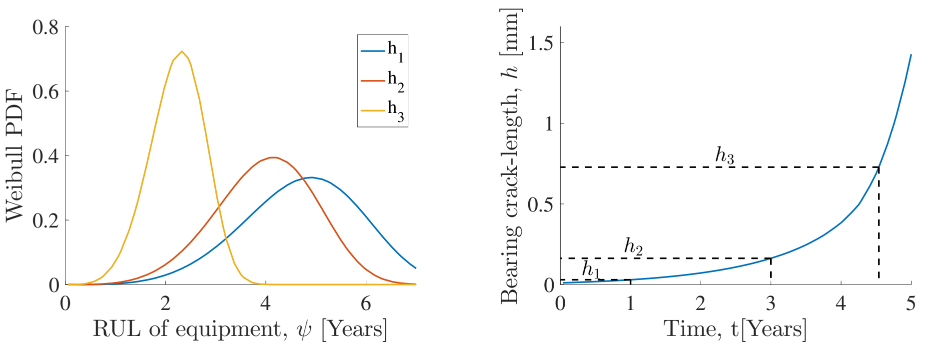
From the previous analysis, it is assumed that $\alpha = 0.1\%$ gives an acceptable risk level. The results from the first open-loop optimization for VaR_α and CVaR_α with $\alpha = 0.1\%$ are illustrated in Figure 6.2. The plots show the predicted behaviour for VaR_α and CVaR_α with three lines indicating the individual scenarios. Figure 6.2 illustrates how VaR_α and CVaR_α decreases with time with confidence level $\alpha = 0.1\%$. Essentially, VaR_α and CVaR_α decreases with decreasing RUL of equipment. This may be to prevent an overly conservative operation. Nevertheless, the non-convex nature of the system equations resulted in a non-convex optimization problem. Solving a non-convex optimization problem with the local solver, IPOPT, cannot guarantee that the minimum point obtained is a global minimum. In this regard, there might exist better solutions with the same α.



(a) VaR_α , as a function of time, t, with $\alpha = 0.1\%$. **(b)** CVaR_α , as a function of time, t, with $\alpha = 0.1\%$.

Figure 6.2: The state profiles for VaR_α and CVaR_α from the first open-loop optimization with confidence level $\alpha = 0.1\%$.

RUL of equipment is employed to account for uncertainty in equipment and is assumed to be Weibull distributed with shape and scale parameters. The bearing crack-length is applied as a health indicator for the RUL of equipment distribution. The risk captured by CVaR can be directly affected by shaping the RUL-distribution. The RUL of equipment-distribution can be formed by influencing the states x by adjusting the inputs u. Figure 6.3a shows the predicted degradation of the bearing crack-length from the first open-loop optimization with a nominal value for c_{paris} = 1.0. The corresponding RUL of equipment distribution at degradation levels h_1, h_2 and h_3 are depicted in Figure 6.3b. Essentially the expected RUL increases with decreasing degradation. That is reasonable as a smaller crack-length would suggest that the system is operational for a longer time, compared to a larger crack-length.



(a) RUL of equipment, ψ, distribution at degradation level h_1, h_2 and h_3.

(b) Bearing crack-length at degradation level h_1, h_2 and h_3.

Figure 6.3: Evolution of the degradation of equipment, h, and the RUL of equipment- distributions at degradation levels h_1, h_2 and h_3 with $\alpha = 0.1\%$.

6.3.2 Optimization of Production

Maximizing of CVaR$_\alpha$ with respect to RUL of equipment, without any profit evaluation, can lead to decisions that are overly restrictive. Consequently, this study investigates an optimal control policy with respect to both risk and profit. The profit is measured as the net present value (NPV) of gas production downstream from the wet-gas compressor. The predicted gas production profile from the first open-loop optimization is illustrated in Figure 6.4 with three lines indicating the individual scenarios. The optimization found it profitable to maximize gas production in the beginning. The NPV concept in the objective function in Equation 5.10 favours early gas production rather than late production. After approximately 3.5 years of operation, the system realized that the predicted loss of profit until the next maintenance intervention is rather small and less valuable. The plant responded by increasing the inputs to squeeze more gas production out of the system. This may indicate that the specified maintenance horizon might have been too short. Preferably, the next maintenance should have been planned later.

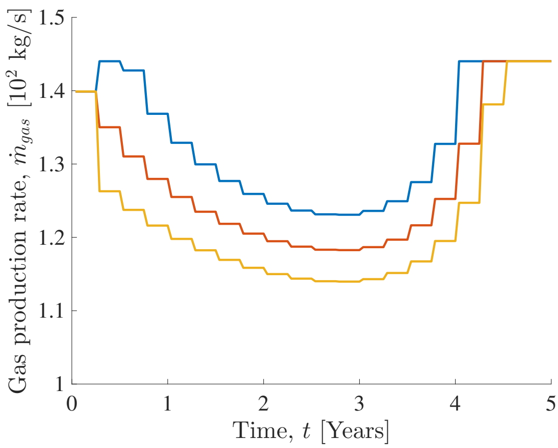


Figure 6.4: Open-loop state profile for the gas production rate, $\dot{m}_{gas}$, as a function of time, t, at t = 0.

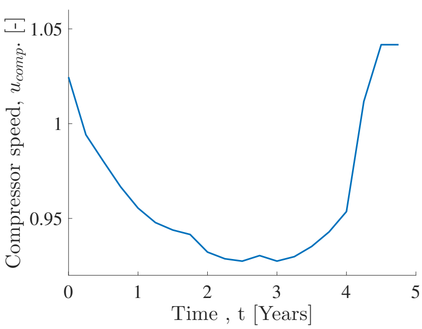
Another interpretation of this production profile is that perhaps the weighting, ω, should have been different. ω, is a tuning parameter in the optimization routine and affects the outline of the control strategy. The choice of weight in the objective function in Equation 5.7 is essential to the nature and formulation of the problem. In the context of this study, ω is assumed to be constant and equal to 0.5 to enforce a risk averse operation. Higher values for ω would result in a more conservative operation. On the other hand, applying significantly lower values for ω would give a control strategy where risk is weighted very little. As a consequence, the plant would end up favouring maximum production for the entire operation horizon. Employing a time-dependent ω might have given a consistently declining production rate. This should be discussed in future work.

There are several instrumental factors to this production profile. An important contributor to the behaviour of the controller could be that net present value of gas production downstream from the wet-gas compressor is measured in terms of money, while the risk of failure is measured in years. A simplified assumption is made that the conversion rate between money and years is 1:1. In relation to Equation 5.7, the summation of two different currencies is not desirable and may cause errors. Improvements to the proposed relation between the risk of failure and profit is imperative. This must be addressed in future work.

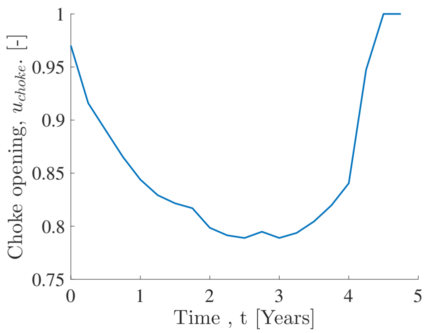
Previous study on health-aware control of the subsea gas compression station at the Åsgard field was handled as part of the work with the project thesis (Ims, 2017). Here, the degradation of equipment were used for condition monitoring purposes. Paris' law for crack propagation was used to predict degradation of equipment. Based on a rational mindset, the degradation of equipment as a health propagation model ought to yield a more economically profitable operation as it imposes constraints directly on the fault indicator. Incorporating risk control into the optimization routine for the subsea system would suggest a more conservative operational strategy as it seeks to limit loss of production. However, the outcome is a little contradicting as constraints on the bearing crack-length yields a less profitable control policy than integrating risk control in the optimization routine.

6.4 Closed-loop Optimization with Risk Control

The closed-loop model predictive controller adds random disturbances on the inputs to the open-loop optimization problem in order to obtain an optimal control strategy for operation of the subsea plant. In the context of this work, a shrinking time horizon will be used by decreasing the prediction horizon by one time step for each open loop optimization. Figure 6.5 presents the optimal control policy from the closed loop simulation for the two control inputs, compressor speed, u_{comp}, and choke opening, u_{choke}. The plots correlate with the gas production rate results obtained from open-loop optimization in Figure 6.4. The optimization found it profitable to maximize gas production in the beginning. After approximately 3.5 years of operation, the system realized that the predicted loss of profit until the next maintenance intervention is rather small and less valuable. Figure 6.5 shows that the plant responded by increasing the inputs to squeeze more gas production out of the system. The reasons for this production outline were discussed in the previous section.



(a) Compressor speed, u_{comp}, as a function of time, t, with disturbance.



(b) Choke opening, u_{choke}, as a function of time, t, with disturbance.

Figure 6.5: Closed loop state profiles with noise for the compressor speed, u_{comp}, and the choke opening, u_{choke}, with confidence levels, $\alpha = 0.1\%$.

The optimal control control problem is solved with a dynamically time inconsistent risk measure. As a consequence, the resulting optimal control policy depicted in Figure 6.5 is dynamically time inconsistent. This gives a sub-optimal solution. However, the degree of sub-optimality might be small. Closed-loop simulations without disturbance provided an invariable optimal control policy throughout the optimization routine in-spite of the dynamic time inconsistency. This was tested by applying the nominal value for all the scenario realizations for the random variable c_{paris}. This may indicate that the additive form of CVaR perhaps could yield a dynamic time-consistent optimal control policy after all.

Nevertheless, dynamic time inconsistency of the dynamic risk measure must be addressed in future work.

Chapter 7

Concluding Remarks and Further Work

7.1 Concluding Remarks

This master thesis proposed a model predictive control (MPC) approach for integrating health monitoring and control to achieve an economic optimal control policy, without jeopardizing the safety of the Åsgard gas compression station. Risk controlling techniques that consider the risk of failure were used for condition monitoring purposes. The risk measure Conditional Value-at-Risk (CVaR) with respect to remaining useful life (RUL) of equipment was implemented in the form of MATLAB code in the optimization routine in the MPC. The optimized strategies obtained with the open-loop optimization were predictive control strategies without disturbance. For that reason, a closed loop was implemented in MATLAB with receding horizon to include random disturbances on inputs. Due to the lack of data from the real subsea gas compression station at the Åsgard field, it was impossible to derive exact parameters for the risk measure used to estimate the risk of failure. Based on a particular set of parameters and assumptions made for this system, the optimal control policy sought safe operation until the next maintenance intervention. Two tuning parameters, the weighting between the reliability and the economic objective, β, and the confidence level for the risk measure, α, were used to tune the system.

The numerical simulation showed that the average RUL of the 0.1% worst RUL outcomes was calculated be to just above five years at $t = 0$. As expected, the predicted CVaR with respect to RUL of equipment decreases with time until the next maintenance engagement, which is scheduled to happen in five years. Implementing the risk measure with higher confidence levels gave rise to higher values for CVaR with respect to RUL of equipment. However, maximizing of CVaR with respect to RUL itself, without any profit evaluation, can lead to decisions that are overly restrictive. Consequently, this study found an optimal control policy with respect to both risk and mean profit. By doing so, the optimization found it profitable to decrease the gas production rate with time.

The overall conclusion from this work, is that health-aware control with risk measures for condition monitoring has the possibility to master the reliability of a subsea plant. Nevertheless, the accuracy of the system model and the implementation of the risk measure estimate influence the controllers ability to predict the risk of failure. The non-convex nature of the system equations resulted in non-convex optimization problem. Solving a non-convex optimization problem with a local solver, IPOPT, cannot guarantee that the minimum point obtained is a global minimum. In this regard, there might exist better solutions with the same α. In addition, the dynamic risk measure chosen for this study is dynamic time inconsistent which in turn yielded a dynamic time inconsistent optimal control policy. However, the degree of sub-optimality might be small as the closed-loop simulations without disturbance provided an invariable optimal control policy throughout the optimization.

7.2 Further Work

There are many possible paths to follow in future research to improve this approach to risk control and optimization. A reasonable next step would be to look for improvements to the proposed relation between the reliability objective and the economic objective. As of now, the two objectives are not in the same currency and the relation between the risk of failure and the profit must be analyzed. Furthermore, attention must be paid to the dynamic time inconsistency of the dynamic risk measure which is applied in the optimization of this subsea system. The additive form of the multi-period risk measure CVaR is generally not dynamically time consistent. This ought to be improved in future work. In general, two approaches can be adopted to master time-inconsistency: either analyze the optimal control policies in detail and enforce conditions to overcome dynamic time inconsistency, or introduce modifications of the risk measure and then generate dynamic time consistent strategies (Chen et al., 2017).

Furthermore, the formulation of the optimal control problem can be changed to minimizing the unavailability whilst constraining the minimum expected economic profit. This indicates that the objective is to minimize the risk of failure while constraining the minimum expected economic profit from operation.

Bibliography

Acciaio, B., Penner, I., 2011. Dynamic risk measures. In: Advanced mathematical methods for finance. Springer, pp. 1–34.

Aguilera, P., Carlui, L., 2013. Subsea wet gas compressor dynamics. Master's thesis, Department of Energy and Process Engineering.

Andersson, J., 2013. A General-Purpose Software Framework for Dynamic Optimization. Ph.D. thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium.

Artzner, P., Delbaen, F., Eber, J.-M., Heath, D., 1999. Coherent measures of risk. Mathematical finance 9 (3), 203–228.

Austrheim, T., 2006. Experimental characterization of high-pressure natural gas scrubbers. Ph.D. thesis, The University of Bergen.

Bechhoefer, E., Bernhard, A., He, D., 2008. Use of paris law for prediction of component remaining life. In: Aerospace Conference, 2008 IEEE. IEEE, pp. 1–9.

Bellman, R., 1954. Dynamic Programming and a New Formalism in the Calculus of Variations. Proceedings of the National Academy of Sciences of United States of America 40 (4), 231–235.

Biegler, L. T., 2010. Nonlinear programming: concepts, algorithms, and applications to chemical processes. Vol. 10. Siam.

Capolei, A., Christiansen, L. H., Jørgensen, J. B., 2018. Risk minimization in life-cycle oil production optimization. arXiv preprint arXiv:1801.00684.

Capolei, A., Foss, B., Jørgensen, J. B., 2015. Profit and risk measures in oil production optimization. Proceedings of the 2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production. Florianópolis,Brazil 48 (6), 214–220.

Chen, Z., Consigli, G., Liu, J., Li, G., Fu, T., Hu, Q., 2017. Multi-period risk measures and optimal investment policies. In: Optimal financial decision making under uncertainty. Springer, pp. 1–34.

Diehl, M., 2011. Numerical optimal control. Optimization in Engineering Center (OPTEC).

Dranchuk, P., Abou-Kassem, H., et al., 1975. Calculation of z factors for natural gases using equations of state. Journal of Canadian Petroleum Technology 14 (03).

Dupačová, J., Kozmík, V., 2015. Structure of risk-averse multistage stochastic programs. OR spectrum 37 (3), 559–582.

Escobet, T., Puig, V., Nejjari, F., 2012. Health aware control and model-based prognosis. In: Control & Automation (MED), 2012 20th Mediterranean Conference on. IEEE, pp. 691–696.

Fredheim, A. O., Gjertsen, L. H., Rusten, B. H., Austrheim, T., Johnsen, C. G., Sep. 24 2013. Separator unit. US Patent 8,540,788.

Geraci, A., Katki, F., McMonegal, L., Meyer, B., Lane, J., Wilson, P., Radatz, J., Yee, M., Porteous, H., Springsteel, F., 1991. IEEE standard computer dictionary: Compilation of IEEE standard computer glossaries. IEEE Press.

Gill, P. E., Murray, W., Saunders, M. A., Tomlin, J. A., Wright, M. H., 2008. George b. dantzig and systems optimization. Discrete Optimization 5 (2), 151 – 158.

Grimholt, C., Skogestad, S., 2015. Optimization of Oil Field Production Under Gas Coning Conditions Using the Optimal Closed-Loop Estimator. IFAC-PapersOnLine 48 (6), 39–44.

Hans, C. A., Sopasakis, P., Bemporad, A., Raisch, J., Reincke-Collon, C., Dec 2015. Scenario-based model predictive operation control of islanded microgrids. In: 2015 54th IEEE Conference on Decision and Control (CDC). pp. 3272–3277.

Heng, A., Zhang, S., Tan, A. C., Mathew, J., 2009. Rotating machinery prognostics: State of the art, challenges and opportunities. Mechanical systems and signal processing 23 (3), 724–739.

Herceg, D., Sopasakis, P., Bemporad, A., Patrinos, P., 2017. Risk-averse model predictive control. arXiv preprint arXiv:1704.00342.

Hundseid, O., Bakken, L. E., Grüner, T. G., Brenne, L., Bjorge, T., 2008. Wet gas performance of a single stage centrifugal compressor. In: ASME Turbo Expo 2008: Power for Land, Sea, and Air. American Society of Mechanical Engineers, pp. 661–670.

Ims, J., 2017. Modelling of Åsgard subsea gas compression station for condition monitoring purposes. Project Thesis, Department of Chemical Engineering, Norwegian University of Science and Technology.

Jiang, R., Murthy, D., 2011. A study of weibull shape parameter: properties and significance. Reliability Engineering & System Safety 96 (12), 1619–1626.

Jorion, P., 2006. Value at Risk, 3rd Ed.: The New Benchmark for Managing Financial Risk. McGraw-Hill Education.

Kay, W., 1936. Gases and vapors at high temperature and pressure-density of hydrocarbon. Industrial & Engineering Chemistry 28 (9), 1014–1019.

Kiranyaz, S., Ince, T., Gabbouj, M., 2014. Optimization Techniques: An Overview. In: Multidimensional Particle Swarm Optimization for Machine Learning and Pattern Recognition. Springer, pp. 13–44.

Klüppelberg, C., Straub, D., Welpe, I. M., 2014. Risk-A Multidisciplinary Introduction. Springer.

Kozmık, V., 2015. Multiperiod Risk Measures.

Krokhmal, P., Zabarankin, M., Uryasev, S., 2011. Modeling and optimization of risk. Surveys in operations research and management science 16 (2), 49–66.

Kurt, D., 2016. Net Present Value (NPV) Definition | Investopedia. Investopedia. Retrieved 2018-04-25.

Lucia, S., Finkler, T., Engell, S., 2013a. Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. Journal of Process Control 23 (9), 1306–1319.

Lucia, S., Subramanian, S., Engell, S., 2013b. Non-conservative robust nonlinear model predictive control via scenario decomposition. In: Control Applications (CCA), 2013 IEEE International Conference on. IEEE, pp. 586–591.

McClimans, O., Fantoft, R., et al., 2006. Status and new developments in subsea processing. In: Offshore Technology Conference. Offshore Technology Conference.

Morari, M., Lee, J. H., 1999. Model predictive control: past, present and future. Computers & Chemical Engineering 23 (4-5), 667–682.

Moreno-Trejo, J., Markeset, T., 2011a. Identifying challenges in the development of subsea petroleum production systems. In: IFIP International Conference on Advances in Production Management Systems. Springer, pp. 287–295.

Moreno-Trejo, J., Markeset, T., 2011b. Mapping factors influencing the selection of subsea petroleum production systems. In: IFIP International Conference on Advances in Production Management Systems. Springer, pp. 242–250.

Morgan, JP, 1994. Riskmetrics.

Nocedal, J., Wright, S. J., 2006. Numerical Optimization, 2nd Edition. Springer, New York, NY, USA.

OREDA Participants, 2002. Offshore reliability data handbook, 4th. ed. SINTEF, Trondheim.

Paris, P., Erdogan, F., 1963. A critical analysis of crack propagation laws. Journal of basic engineering 85 (4), 528–533.

Pereira, E. B., Galvão, R. K. H., Yoneyama, T., July 2010. Model Predictive Control using Prognosis and Health Monitoring of actuators. In: 2010 IEEE International Symposium on Industrial Electronics. pp. 237–243.

Pflug, G. C., Romisch, W., 2007. Modeling, measuring and managing risk. World Scientific, Singapore.

Riedel, F., 2004. Dynamic coherent risk measures. Stochastic processes and their applications 112 (2), 185–200.

Rockafellar, R. T., 2007. Coherent Approaches to Risk in Optimization Under Uncertainty. Tutorials in Operations Research 3, 38–61.

Rockafellar, R. T., Royset, J. O., 2010. On buffered failure probability in design and optimization of structures. Reliability Engineering & System Safety 95 (5), 499–510.

Rockafellar, R. T., Uryasev, S., 2000. Optimization of Conditional Value-at-Risk. Journal of Risk 2, 21–41.

Rockafellar, R. T., Uryasev, S., 2002. Conditional value-at-risk for general loss distributions. Journal of banking & finance 26 (7), 1443–1471.

Rockafellar, R. T., Uryasev, S., 2013. The fundamental risk quadrangle in risk management, optimization and statistical estimation. Surveys in Operations Research and Management Science 18 (1-2), 33–53.

Ruszczyński, A., 2010. Risk-averse dynamic programming for markov decision processes. Mathematical programming 125 (2), 235–261.

Salazar, J. C., Weber, P., Nejjari, F., Theilliol, D., Sarrate, R., 2016. MPC framework for system reliability optimization. In: Advanced and Intelligent Computations in Diagnosis and Control. Springer, pp. 161–177.

Sanchez, H., Escobet, T., Puig, V., Odgaard, P. F., 2015. Health-aware model predictive control of wind turbines using fatigue prognosis. IFAC-PapersOnLine 48 (21), 1363–1368.

Seborg, D., Mellichamp, D., Edgar, T., Doyle, F., 2010. Process Dynamics and Control. John Wiley & Sons.

Setekleiv, E., Anfray, J., Boireau, C., Gyllenhammar, E., Kolbu, J., et al., 2016. An evaluation of subsea gas scrubbing at extreme pressures. In: Offshore Technology Conference. Offshore Technology Conference.

Singh, S., Chow, Y.-L., Majumdar, A., Pavone, M., 2017. A Framework for Time-Consistent, Risk-Sensitive Model Predictive Control: Theory and Algorithms. arXiv preprint arXiv:1703.01029.

Song, K. Y., Chang, I. H., Pham, H., 2017. A software reliability model with a weibull fault detection rate function subject to operating environments. Applied Sciences 7 (10), 983.

Verheyleweghen, A., Jäschke, J., 2017a. Framework for Combined Diagnostics, Prognostics and Optimal Operation of a Subsea Gas Compression System. IFAC-PapersOnLine 50 (1), 15916 – 15921, 20th IFAC World Congress.

Verheyleweghen, A., Jäschke, J., 2017b. Health-aware operation of a subsea gas compression system under uncertainty. Foundations of Computer Aided Process Operations / Chemical Process Control 2017 . FOCAPO/CPC Tucson, AZ. 2017-01-08 - 2017-01-12.

Verheyleweghen, A., Jäschke, J., 2017c. Using operational degrees of freedom to optimize remaining useful life. VI Oil and Gas Production Optimization Workshop, Rio de Janeiro, Brazil.

Verheyleweghen, A., Jäschke, J., 2018. Risk-based health-aware production optimization with application to a subsea oil and gas production systems.

Wächter, A., Biegler, L. T., Mar 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming 106 (1), 25–57.

Wang, T., 1999. A class of dynamic risk measures. University of British Columbia 21.

Appendices

Appendix A

Simulation Parameters

Table A.1 lists some necessary parameters for the simulation of the model predictive controller for the subsea gas compression station at the Åsgard field.

Table A.1: Simulation parameters

Parameter	Description	Value	Unit
t_f	Time until the next maintenance intervention.	1	Years
S	Number of scenarios.	3	-
N	Number of time periods. It is the length of the horizon.	20	-
p_l	Probability of occurrence for scenario l.	1/3	%
k_a	Parameter in shape parameter equation.	4.55	-
k_b	Parameter in shape parameter equation.	0.1	-
k_c	Parameter in shape parameter equation.	0.1	-
λ_a	Parameter in scale parameter equation.	5.7	-
λ_b	Parameter in scale parameter equation.	1.2	-
λ_c	Parameter in scale parameter equation.	2.8	-
β^k	Discounting factor for risk measures at time k.	0.05	-
$\dot{m}_1$	Mass flow rate entering the valve.	0.9	kg/s
c_{choke}	Choke constant.	0.1671	kg/s$\sqrt{bar}$
P_1	Pressure upstream to the valve.	100	bar
u_{choke}	Initial choke opening.	0.565	-
σ	Interfacial/surface tension between two phases.	2.2	mN/m
μ_l	Viscosity in liquid film in separator.	0.096	cP
D	Separator diameter.	2	m
K	Parameter in expression for friction factor $f_{g,i}$.	1.926	-
m	Parameter in expression for friction factor $f_{g,i}$.	-1/3	-
$u_{g,tg}$	Tangential gas velocity in cyclone.	3.75	m/s
θ	Angle of gas velocity on AFC wall.	45	^{o}
$\hat{\theta}$	Angle to indicate direction of gas flow.	45	^{o}
A	Parameter for the correlation between E and α.	-0.1345	-
B	Parameter for the correlation between E and α.	1.01	-
a	Parameter for the Re-entrainment number,E.	0.4	-

c_{Paris}	Nominal value for parameter in Paris' law.	2.4	s/J^2
g	Gravitational constant.	9.81	m/s^2
u_{comp}	Initial compressor speed.	0.85	-
R	Gas constant.	8.31446	J/K mol
T_3	Temperature upstream to the compressor.	350	K
h_0	Initial bearing degradation.	0.01	mm
c_1	Parameter in the expression for η.	0.582	-
c_2	Parameter in the expression for η.	-2.398	-
c_3	Parameter in the expression for η.	2.752	-
c_4	Parameter in the expression for η.	- 3.969	-
c_5	Parameter in the expression for η.	4.303	-
c_6	Parameter in the expression for H.	-0.9937	-
c_7	Parameter in the expression for H.	2.256	-
c_8	Parameter in the expression for H.	1.888	-
A_1	Parameter in the expression for Z.	0.3265	-
A_2	Parameter in the expression for Z.	-1.0700	-
A_3	Parameter in the expression for Z.	-1.0700	-
A_4	Parameter in the expression for Z.	0.01569	-
A_5	Parameter in the expression for Z.	-0.05165	-
A_6	Parameter in the expression for Z.	0.5475	-
A_7	Parameter in the expression for Z.	-0.7361	-
A_8	Parameter in the expression for Z.	0.1844	-
A_9	Parameter in the expression for Z.	0.1056	-
A_{10}	Parameter in the expression for Z.	0.6134	-
A_{11}	Parameter in the expression for Z.	0.7210	-
q_{min}	Minimum allowable flow to prevent surge.	1.163	m^3/s
q_{max}	Maximum allowable flow to svoid Stonewall cond.	2.286	m^3/s
i	Discount rate for net present value calculations.	0.015	-
ω	Weighting factor in objective function.	0.5	-
t_0	Initial time.	0	Years

Appendix B

MATLAB code

B.1 Stream Definition

```
function stream = def_stream(name,x,init,Ti,Pi,mgdoti,mcdoti)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%@Course       : Master Thesis Spring 2018
%@Task         : Function that creates a stream object for the system.
%@input        : name of the stream (name), composition of the stream (x),
%                boolean variable indicating if it is inital stream (init),
%                temperature in the stream (Ti), pressure in the stream(Pi),
%                gas flow rate (mgdoti), liqiud flow rate (mcdoti)
%@output       : stream object (stream)

%@author       : Adriaen Verheyleweghen
%@organization: Department of Chemical Engineering, NTNU, Norway
%@requires     : MATLAB R2016a (not tested in other releases)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

addpath('/Users/juliebergeims/downloads/casadi-matlabR2015a-v3.0.0')
import casadi.*

nm = num2str(name);                                   % SCALING FACTORS
T = MX.sym(['T_',nm]);                                % Temperature
P = MX.sym(['P_',nm]);                                % Pressure
mgdot = MX.sym(['mgdot_',nm]);                        % Mass flow rate
mcdot = MX.sym(['mcdot_',nm]);                        % Mass flow rate

z = struct();                                         % Decision variables
p = struct();                                         % Extra parameters
algs = struct();                                      % Residuals Equaitons

%% Declare variables
```

```
Z     = MX.sym(['Z_',nm]);                       % Compressibility
Cp    = MX.sym(['Cp_',nm]);                      % Heat capacity
rho   = MX.sym(['rho_',nm]);                     % Density of gas
Vgdot = MX.sym(['Vgdot_',nm]);                   % Gas volumetric flow
Vcdot = MX.sym(['Vcdot_',nm]);                   % Liquid Volumetric flow
GVF   = MX.sym(['GVF_',nm]);                     % Gas-volume-fraction

%% Concatenate variables

z.T     = T;
z.P     = P;
z.mgdot = mgdot;
z.mcdot = mcdot;

z.Z     = Z;
z.Cp    = Cp;
z.rho   = rho;
z.Vgdot = Vgdot;
z.Vcdot = Vcdot;
z.GVF   = GVF;

%% Define properties

% Molar masses [kg/mol]
M_ =[...
    16.04, ... % C1
    30.07, ... % C2
    44.10, ... % C3
    58.12, ... % n-C4
    72.15, ... % n-C5
    86.18, ... % n-C6
    18.02, ... % H2O
    44.01, ... % CO2
    28.01, ... % N2
    58.12, ... % i-C4
    72.15, ... % i-C5
    ];

M = M_*x';
p.M = M;
p.x = x;

% Critical temperatures [K]
Tc_ =[...
    190, ... % C1
    305, ... % C2
    370, ... % C3
```

```
    425, ... % n-C4
    469, ... % n-C5
    507, ... % n-C6
    647, ... % H2O
    304, ... % CO2
    126, ... % N2
    408, ... % i-C4
    460, ... % i-C5
    ];

% Pseudo-critical mixture temperature [K]
%
%   Calculated from Kay's rule:
%   W.B. Kay - "Gases and Vapors At High Temperature and Pressure-Density
%   of Hydrocarbon" (1936)
%
%   Known to be innaccurate, see for instance
%   R.P. Sutton - "Compressibility Factors for High-Molecular-Weight
%   Reservoir Gases" (1985)

Tpc = Tc_*x';

% Pseudo-reduced temperature [-]
Tpr = (T*1E2) / Tpc;

% Critical pressures [Pa]
Pc_ =[...
    4.60551724137931E6, ... % C1
    4.88137931034483E6, ... % C2
    4.25034482758621E6, ... % C3
    3.80000000000000E6, ... % n-C4
    3.37241379310345E6, ... % n-C5
    3.01379310344828E6, ... % n-C6
    22.0620689655172E6, ... % H2O
    7.38344827586207E6, ... % CO2
    3.39000000000000E6, ... % N2
    3.64896551724138E6, ... % i-C4
    3.39000000000000E6, ... % i-C5
    ];

% Pseudo-critical mixture pressure [Pa]
Ppc = Pc_*x';

% Pseudo reduced pressure [-]
Ppr = (P*1E7) / Ppc;

% Compressibility factor
%
```

```
%   Calculated using Dranchuk and Abou-Kassem EOS
%   P.M. Dranchuk and H. Abou-Kassem - " Calculation of Z Factors For
%   Natural Gases Using Equations of State" (1975)

A1 = 0.3265;
A2 = -1.0700;
A3 = -0.5339;
A4 = 0.01569;
A5 = -0.05165;
A6 = 0.5475;
A7 = -0.7361;
A8 = 0.1844;
A9 = 0.1056;
A10 = 0.6134;
A11 = 0.7210;

% Compressibility factor (Z)
%    is calculated using Dranchuckand Abou Kassems eq
%    where tmp is a temporary variable using in the calculation
tmp = 0.27*Ppr/(Z*Tpr);

algs.Z = -Z + 1+(A1+A2/Tpr+A3/Tpr^3+A4/Tpr^4+A5/Tpr^5)*tmp + ...
      (A6+A7/Tpr+A8/Tpr^2)*tmp^2 - ...
      (A7/Tpr+A8/Tpr^2)*A9*tmp^5 + ...
       A10*(1+A11*tmp^2)*(tmp^2/Tpr^3)*exp(-A11*tmp^2);

% Heat capacity
%  Cp = ( c1 + c2*T + c3*T^2 + c4 *T^-2)*R where R is gas constant
%
%   Coefficients  from:
%   "http://www.personal.utulsa.edu/~geoffrey-price/Courses
%   /ChE7023/HeatCapacity-HeatOfFormation.pdf
%   coefficients for different molecules

coeffs =[...
    1.702,  9.081E-3,  -2.164E-6,       0; ... % C1
    1.131, 19.225E-3,  -5.561E-6,       0; ... % C2
    1.213, 28.785E-3,  -8.824E-6,       0; ... % C3
    1.935, 36.915E-3, -11.402E-6,       0; ... % n-C4
    2.464, 45.351E-3, -14.111E-6,       0; ... % n-C5
    3.025, 53.722E-3, -16.791E-6,       0; ... % n-C6
    3.470,  1.450E-3,          0, 0.121E5; ... % H2O
    5.457,  1.045E-3,          0,-1.157E5; ... % CO2
    3.280,  0.593E-3,          0, 0.040E5; ... % N2
    1.677, 37.853E-3, -11.945E-6,       0; ... % i-C4
    2.464, 45.351E-3, -14.111E-6,       0; ... % i-C5
    ];
```

```
% Calculating the heat capacity according to the composition
c = x*coeffs;
R = 8.31446;
algs.Cp = -(Cp*1E2) + (c(1) + c(2)*(T*1E2) +...
    c(3)*(T*1E2)^2 + c(4)*(T*1E2)^(-2))*R;

% Density
algs.rho = -(rho*1E2) + (P*1E7)*(M*1E-3)/(8.3144*Z*(T*1E2));

% Volumetric flow rate
rho_condensate = 10;
algs.Vgdot = -(mgdot*1E2) + (Vgdot)*(rho*1E2);
algs.Vcdot = -(mcdot*1E2) + (Vcdot)*(rho_condensate*1E2);

% GVF
algs.GVF = -GVF + Vgdot/(Vgdot+Vcdot);

%% Define stream object

stream      = struct();
stream.z    = z;
stream.p    = p;
stream.algs = algs;

if init

%% Initialize

test = [T-Ti;P-Pi;mgdot-mgdoti;mcdot-mcdoti];

% Creating a rootfinding function to solve
rf_function = Function('rf_function',{casadi_struct2vec(z)},...
                                  {vertcat(test,casadi_struct2vec(algs))});

% Create a solver for rootfinding problem
rf = rootfinder('rf','kinsol',rf_function);

% Initial guess
Zi = 0.89;
Cpi = 40;
rhoi = 68;
Vgdoti = 179.743;
Vcdoti = 19.97;
GVFi = 0.9;

z0 = [Ti,Pi,mgdoti,mcdoti,Zi,Cpi,rhoi,Vgdoti,Vcdoti,GVFi]';

% Converting a CasADi matrix to a MATLAB dense matrix
```

```
stream.z0 = full(rf(z0));

end
```

B.2 Choke

```matlab
function [choke,outlet] = choke(namein,nameout,inlet,Zi)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%@Course       : Master Thesis Spring 2018
%@Task         : Function that creates a choke object for the system.
%@input        : name of the choke (namein), name of the output stream
%                (nameout), conditions for the inlet stream (inlet),
%                controller input for choke opening (Zi)
%@output       : choke object (choke), outlet stream object (outlet)

%@author       : Adriaen Verheyleweghen
%@organization: Department of Chemical Engineering, NTNU, Norway
%@requires     : MATLAB R2016a (not tested in other releases)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    addpath('/Users/juliebergeims/downloads/casadi-matlabR2015a-v3.0.0')
    import casadi.*

    nm = num2str(namein);

    outlet = def_stream(nameout,inlet.p.x,false,[],[],[],[]);

    % Declearing necessary variables
    Z     = MX.sym(['Z_',nm]);                          % Cchoke opening
    dP    = (inlet.z.P*1E7)-(outlet.z.P*1E7);           % Change in pressure
    m     = (inlet.z.mgdot*1E2)+(inlet.z.mcdot*1E2);    % Total mass flow
    Cv    = 0.4*.4179402833086;                         % Choke constant

    % Creating a struct to hold the input (choke valve opening) variable
    u       = struct();
    u.Z     = Z;

    % Algebraic equations
    algs.m     = m - Cv*Z*sqrt(dP);
    algs.mgdot = outlet.z.mgdot - inlet.z.mgdot;
    algs.mcdot = outlet.z.mcdot - inlet.z.mcdot;
    algs.T     = outlet.z.T     - inlet.z.T;

    % Creating a choke struct
        choke          = struct();
    choke.u        = u;
    choke.algs     = algs;

    %% Initialize
    test = [Z-Zi;casadi_struct2vec(inlet.z)-inlet.z0];

    % Creating a rootfinding function to solve
```

```
    rf_function = Function('rf_function',...
        {vertcat(casadi_struct2vec(inlet.z),...
        casadi_struct2vec(outlet.z),Z)},{vertcat(test,...
        casadi_struct2vec(outlet.algs),casadi_struct2vec(algs))});

    % Create a solver for rootfinding problem
    rf = rootfinder('rf','kinsol',rf_function);

    % Initial guess
    z0in = inlet.z0;
    z0out = z0in;
    z0out(2) = z0out(2)*Zi;

    % Converting a CasADi matrix to a MATLAB dense matrix
    z0 = full(rf(vertcat(z0in,z0out,Zi)));
    outlet.z0 = z0(length(inlet.z0)+1:2*length(inlet.z0));
end
```

.

B.3 Separator

```
function [separator,outletg,outletl] = separator(namein,nameoutg,nameoutl,inlet)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%@Course      : Master Thesis Spring 2018
%@Task        : Function that creates a separator object for the system.
%@input       : name of the separator (namein), name of the gas output
%               stream (nameoutg), name of the liquid output stream
%               (outputl), conditions for the inlet stream (inlet)
%@output      : separator object (separator), gas outlet stream object
%               (outletg), liquid outlet stream object (outetl)

%@author      : Adriaen Verheyleweghen
%@modified    : Julie Berge Ims
%@organization: Department of Chemical Engineering, NTNU, Norway
%@requires    : MATLAB R2016a (not tested in other releases)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    addpath('/Users/juliebergeims/downloads/casadi-matlabR2015a-v3.0.0')
    import casadi.*

    nm = num2str(namein);

    % Creating streams for outlet gas and outlet liquid stream
    outletg = def_stream(nameoutg,inlet.p.x,false,[],[],[],[]);
    outletl = def_stream(nameoutl,inlet.p.x,false,[],[],[],[]);

    % Declare necessary variables
    u_l_tg = MX.sym(['u_l_tg_',nm]);            % Tangential liquid vel
    delta_l = MX.sym(['delta_l_',nm]);          % Film thickness on wall
    alpha = MX.sym(['alpha_',nm]);              % Cyclone efficiency
    re_ent = MX.sym(['re_ent_',nm]);            % Re-entrainment number
    f_l_w = MX.sym(['f_l_w_',nm]);              % Frict. factor film/wall
    f_g_i = MX.sym(['f_g_i_',nm]);              % Frict. factor gas/film
    N_my = MX.sym(['N_my_',nm]);                % Viscosity number
    a_l = MX.sym(['a_l_',nm]);                  % Force on liquid film
    Re_l = MX.sym(['Re_l_',nm]);                % Reynold nr liq. stream

    % Algebraic variables
    z        = struct();
    z.u_l_tg = u_l_tg;
    z.theta_l = delta_l;
    z.alpha = alpha;
    z.re_ent = re_ent;
    z.f_l_w = f_l_w;
    z.f_g_i = f_g_i ;
    z.N_my = N_my;
    z.a_l = a_l;
```

```
z. Re_l = Re_l;

% Declare necessary parameters
D = 2 * 10^(-2);                            % Inner vessel diameter
rho_l = 10 * 10^(2);                        % Liquid density
rho_g = inlet.z.rho * 10^(2);               % Gas density
Vgdot = (inlet.z.Vgdot)*10^(-5);            % Gas volumetric flow rate

% Declare necessary parameters for the separator
Vldot = 0.1*Vgdot;                          % Volumetric liquid flow
angle = pi/4;                               % Angle film/cyclone body
my_l = 9.6e-8*rho_l;                        % Viscosity of liquid
sigma = 2.2e-3;                             % Interfacial tension
u_g_tg = 3.75;                              % Tangential gas velocity
u_g_s = 0.8* u_g_tg * tan(angle);           % Superficial gas velocity


% --- provided that the Re_L is in the transition regime
%     laminar flow          2 < Re_L < 100   K = 3.73   m = -0.47
%     transisiton regime   100< Re_L <1000   K = 1.926  m = -1/3

K = 1.926;
m = -1/3;

% Model fitting parameters Austrheim(2007)
a_const = 0.4;
A = -0.1345;
    B = 1.01;

% Declare necessary expressions
algs       = struct();

algs.u_l_tg   = - u_l_tg+sqrt((f_g_i*1E-2)* rho_g * ((u_g_tg)^2)/    ...
                             (f_l_w * rho_l));
algs.theta_l = - (delta_l*1E05)+((cos(angle))^2) * Vldot/(pi*D*u_l_tg);
algs.alpha    = - alpha + A * re_ent + B;
algs.Re_l     = - Re_l*1E2 + alpha * Vldot * rho_l * cos(angle)/     ...
                 (pi *D * my_l);
algs.a_l      = - a_l + (( u_l_tg)^2 )/ D/2;
algs.N_my     = - N_my*1E-3 + my_l/ sqrt(rho_l * sigma *             ...
                                   sqrt(sigma/(a_l * (rho_l-rho_g))) );
algs.f_g_i    = - f_g_i*1E-2 + 0.005*(1 + 300 * (delta_l*1E05)/D/2);
algs.f_l_w    = - f_l_w+ (K*(Re_l*1E2)^m)^2;
algs.re_ent   = - re_ent + (my_l * u_g_s *((rho_g/rho_l)^0.8 )/      ...
                 (sigma * ((N_my*1E-3)^a_const) * (Re_l*1E2) ^(-1/3)));


algs.gas     = struct();
```

```
algs.gas.mg = outletg.z.mgdot - inlet.z.mgdot*alpha;
algs.gas.mc = outletg.z.mcdot - inlet.z.mcdot*(1-alpha);
algs.gas.T  = outletg.z.T     - inlet.z.T;
algs.gas.P  = outletg.z.P     - inlet.z.P;

algs.liq    = struct();
algs.liq.mg = outletl.z.mgdot - inlet.z.mcdot*(1-alpha);
algs.liq.mc = outletl.z.mcdot - inlet.z.mcdot*alpha;
algs.liq.T  = outletl.z.T     - inlet.z.T;
algs.liq.P  = outletl.z.P     - inlet.z.P;

    separator         = struct();
separator.z       = z;
separator.algs    = algs;

%% Initialize
 test = [casadi_struct2vec(inlet.z)-inlet.z0];

% Vector for all variables
vec = [ u_l_tg delta_l alpha re_ent f_l_w f_g_i N_my a_l Re_l]';

% Creating a rootfinding function to solve
rf_function = Function('rf_function',...
    {vertcat(casadi_struct2vec(inlet.z),...
             casadi_struct2vec(outletg.z),...
             casadi_struct2vec(outletl.z),vec)},...
    {vertcat(casadi_struct2vec(outletg.algs),...
             casadi_struct2vec(outletl.algs),...
             casadi_struct2vec(algs),test)});

% Create a solver for rootfinding problem
rf = rootfinder('rf','kinsol',rf_function);

% Initial guess
a=0.95;
z0in = inlet.z0;
z0in(9) = 0.1;
z0outg = z0in;
z0outg(3) = z0outg(3)*a;
z0outg(4) = z0outg(4)*(1-a);
z0outg(8) = z0outg(8)*a;
z0outg(9) = z0outg(9)*(1-a);

z0outl = z0in;
z0outl(4) = z0outl(4)*a;
z0outl(3) = z0outl(3)*(1-a);
z0outl(9) = z0outl(9)*a;
z0outl(8) = z0outl(8)*(1-a);
```

```
    vec_guess=[0.2205 5.2609 0.900 0.9749 0.1230 0.7 1.7 1.2155 1.655653]';

    % Converting a CasADi matrix to a MATLAB dense matrix
    z0 = full(rf(vertcat(z0in,z0outg,z0outl,vec_guess)));
    outletg.z0 = z0(1*length(inlet.z0)+1:2*length(inlet.z0));
    outletl.z0 = z0(2*length(inlet.z0)+1:3*length(inlet.z0));
    separator.z0 = z0(end-8:end);

end
```

.

B.4 Compressor

```
function [comp,outlet] = compressor(namein,nameout,inlet,Ni,hi)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%@Course       : Master Thesis Spring 2018
%@Task         : Function that creates a coompressor object for the system.
%@input        : name of the inlet stream (namein), name of the output
%                stream (nameout), conditions for the inlet stream (inlet),
%                inital compressor speed (Ni), inital bearing crack length
%                in the compressor (hi)
%@output       : compressor object (comp), outlet stream from compressor
%                (outlet)

%@author       : Adriaen Verheyleweghen
%@modified     : Julie Berge Ims
%@organization: Department of Chemical Engineering, NTNU, Norway
%@requires     : MATLAB R2016a (not tested in other releases)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    addpath('/Users/juliebergeims/downloads/casadi-matlabR2015a-v3.0.0')
    import casadi.*

    nm = num2str(namein);
    % Creating a stream 'object' for the outlet stream
    outlet = def_stream(nameout,inlet.p.x,false,[],[],[],[]);

    % Compressor variables and parameters
    N = MX.sym('N');                     % Normalized compressor speed
    h = MX.sym('h');                     % Compressor bearing crack length

    gamma = MX.sym(['gamma_',nm]);       % Average adiabatic ratio
    k     = MX.sym(['k_',nm]);           % k = n/(n+1)
    nu    = MX.sym(['nu_',nm]);          % Compressor efficiency
    H     = MX.sym(['H_',nm]);           % Compressor head
    Pow   = MX.sym(['Pow_',nm]);         % Compressor power consumption
    srg   = MX.sym(['srg_',nm]);         % Indicates surge
    stw   = MX.sym(['stw_',nm]);         % Indicates Stonewall
    Paris = MX.sym(['Paris_',nm]);       % Lumped parameter
    a     = MX.sym(['a_',nm]);           % Wear parameter

    % Struct for equations
    algs = struct();                     % Algebraic equations
    odes = struct();                     % Ddifferential equations

    % Structs for variables
    z = struct();                        % Algebraic variables
    x = struct();                        % Differential variables
    u = struct();                        % Inputs
    p = struct();                        % Parameters
```

```
u.N      = N;
x.h      = h;
z.gamma = gamma;
z.k      = k;
z.nu     = nu;
z.H      = H;
z.Pow    = Pow;
z.srg    = srg;
z.stw    = stw;
z.Paris = Paris;
p.a      = a;

% Extract variables from streams
T1     = inlet.z.T                         *1E2;
T2     = outlet.z.T                        *1E2;
P1     = inlet.z.P                         *1E7;
P2     = outlet.z.P                        *1E7;
Cp1    = inlet.z.Cp                        *1E2;
Cp2    = outlet.z.Cp                       *1E2;
q1     = (inlet.z.Vgdot+inlet.z.Vcdot);
M1     = inlet.p.M                         *1E-3;
Z1     = inlet.z.Z;
rho1   = inlet.z.rho                       *1E2;
GVF1   = inlet.z.GVF;

% Parameters
g      = 9.80665;          % Gravitational constant
R      = 8.31446;          % Universal gas constant
rho_l = 10;                % Density of liquid

c1 = 0.582;                % Parameters for the fit nu = f(q)
c2 = -2.398;
c3 = 2.75;
c4 = - 3.969;
c5 = 4.303;

c6 = -0.9937;              % Parameters for the fit H = f(q)
c7 = 2.256;
c8 = 1.888;

qmin = 1.163;              % Minimum allowable flow at N=1 (surge line)
qmax = 2.286;              % Maximum allowable flow at N=1 (choke line)

%% Define the residuals
head_par = 4;                                  % Head adjustment parameter

qN       = q1/N;                    % Flow corresponding to N=1 (fan laws)
```

```
rho_avg1 = GVF1*rho1+(1-GVF1)*rho_l;                     % Average density
f_wood   = 1/(rho_avg1/rho1*sqrt(rho_avg1/rho1*GVF1)); % Woods cor.fac.

algs.nu = nu - (c1*qN^2 + c2*qN + c3)/(qN^2 + c4*qN + c5);
algs.T  = (H*1E3)  - k*Z1*R/(g*M1)*(T2-T1);
algs.H  = H  - head_par*(c6*qN^2 + c7*qN + c8)*N^2*f_wood;
algs.k  = k  - nu*gamma/(gamma-1);
algs.gamma = gamma - 1/2*( Cp1/(Cp1-R) + Cp2/(Cp2-R));
algs.P  = T2/T1 - (P2/P1)^(1/k);

algs.Pow = (Pow*1E7) - (H*1E3)*q1*rho1*g/nu;
algs.srg = srg - (qN-qmin);
algs.stw = stw - (qmax-qN);
algs.mg = inlet.z.mgdot - outlet.z.mgdot;
algs.mc = inlet.z.mcdot - outlet.z.mcdot;
algs.Paris = Paris - Pow^2/N;

%% Define the dynamics
odes.dh = a*2.4*Paris*h;

%% Collect the variables and equations into struct
comp = struct();
comp.z = z;
comp.x = x;
comp.u = u;
comp.p = p;
comp.algs = algs;
comp.odes = odes;

%% Initialize
test = [N-Ni;h-hi;casadi_struct2vec(inlet.z)-inlet.z0];

% Creating a rootfinding function to solve
rf_function = Function('rf_function',...
    {vertcat(casadi_struct2vec(inlet.z),casadi_struct2vec(outlet.z),...
    casadi_struct2vec(z),N,h)},{vertcat(test,...
    casadi_struct2vec(outlet.algs),casadi_struct2vec(algs))});

% Create a solver for rootfinding problem
rf = rootfinder('rf','kinsol',rf_function);

% Initial guess
guess = [1.2,4,0.77,9,0.96,0.4,075,1.1]';

z0in = inlet.z0;
z0out = z0in; z0out(2) = z0out(2)*1.4;

% Converting a CasADi matrix to a MATLAB dense matrix
```

```
    z0         = full(rf(vertcat(z0in,z0out,guess,Ni,hi)));
    outlet.z0 = z0(length(inlet.z0)+1:2*length(inlet.z0));
    comp.z0    = z0(2*length(inlet.z0)+1:end-2);
end
```

B.5 Subsea Model

```
function [x_struct,z_struct,u_struct,x0_struct,z0_struct,u0_struct,...
    p_struct,ode,alg]= subsea_model(u_choke, u_comp)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%@Course       : Master Thesis Spring 2018
%@Task         : Function that creates an object that defines the subsea
%                system for with a choke, a separator and a separator.
%                Four streams connecting the units are also created.
%@input        : Choke opening (u_choke), compressor speed (u_comp)
%@output       : Differential states (x_struct), algebraic states (z_struct)
%                control inputs (u_struct), initial conditions for
%                differential states (x0_struct, initial conditions for
%                alfebraic states (z0_struct), inital values for inputs
%                (u0_struct), parameters (p_struct), differential eguations
%                (ode), algebraic equations (alg)

%@author       : Adriaen Verheyleweghen
%@modified     : Julie Berge Ims
%@organization: Department of Chemical Engineering, NTNU, Norway
%@requires     : MATLAB R2016a (not tested in other releases)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


    % Link to Casadi installation folder
    addpath('/Users/juliebergeims/downloads/casadi-matlabR2015a-v3.0.0')
    import casadi.*
    addpath([pwd,'/functions'])

    % Declare necessary parameters for stream 1
    x = [.92,.05,.02,.005,.005,0,0,0,0,0,0];     % Composition of the fluid
    T1 = 3.5;                                    % Inlet temperature 350 K
    P1 = 1.0;                                    % Inlet pressure 100 bar
    mgdot1 = 0.9;                                % Inlet gas mass flow rate
    mcdot1 = 0;                                  % Inlet liquid mass flow

    % Declare other necessary parameters for the process
    comp_0 = 0.01;                               % Init. comp. bearing degr.


    % First well: def_stream(name,x,init,Ti,Pi,mgdoti,mcdoti)
    stream1 = def_stream('1',x,true,T1,P1,mgdot1,mcdot1);

    % Choke: choke(namein,nameout,inlet,Zi)
    [chk,stream2] = choke('chk','2',stream1,u_choke);

    % Separator: separator(namein,nameoutg,nameoutl,inlet)
    [sep,stream3,stream5] = separator('sep','3','5',stream2);
```

```
% Compressor: compressor(namein,nameout,inlet,Ni,hi)
[comp,stream4] = compressor('comp34','4',stream3,u_comp,comp_0);

%% Concatenate the decision variables

% Algebraic variables
z_struct             = struct();

% Remove unneeded variables (to reduce size of system)
z_struct.stream1     = rmfield(stream1.z,                              ...
                            {'Cp'});
z_struct.stream2     = rmfield(stream2.z,                              ...
                            {'Cp','Vcdot','GVF'});
z_struct.stream3     = rmfield(stream3.z,                              ...
                            {});
z_struct.stream4     = rmfield(stream4.z,                              ...
                            {'rho','Vgdot','Vcdot','Z','GVF'});
z_struct.stream5     = rmfield(stream5.z,                              ...
                            {'Cp','rho','Vgdot','Vcdot','Z','GVF'});
z_struct.separator   = sep.z;
z_struct.compressor = comp.z;

% Same for the struct of initial conditions
z0_struct             = struct();
z0_struct.stream1     = rmfield(casadi_vec2struct(stream1.z,         ...
                             stream1.z0),                             ...
                             {'Cp'}                                    ...
                             );
z0_struct.stream2     = rmfield(casadi_vec2struct(stream2.z,         ...
                             stream2.z0),                             ...
                             {'Cp','Vcdot','GVF'}                      ...
                             );
z0_struct.stream3     = rmfield(casadi_vec2struct(stream3.z,         ...
                            stream3.z0),                              ...
                            {}                                         ...
                            );
z0_struct.stream4     = rmfield(casadi_vec2struct(stream4.z,         ...
                            stream4.z0),                              ...
                            {'rho','Vgdot','Vcdot','Z','GVF'}          ...
                            );
z0_struct.stream5     = rmfield(casadi_vec2struct(stream5.z,         ...
                            stream5.z0),                              ...
                            {'Cp','rho','Vgdot','Vcdot','Z','GVF'}     ...
                            );
z0_struct.separator   = sep.z0;
z0_struct.compressor = comp.z0;
```

```
    % Differential variables
    x_struct = struct();
    x_struct.compressor = comp.x;

    x0_struct = struct();
    x0_struct.compressor = comp_0;

    % Inputs
    u_struct = struct();
    u_struct.compressor = comp.u;
    u_struct.choke      = chk.u;

    u0_struct = struct();
    u0_struct.compressor = u_comp;
    u0_struct.choke      = u_choke;

    % Parameters
    p_struct = struct();
    p_struct.a = comp.p.a;
    p_struct.GVF = stream1.z.GVF;

    %% Define equations

    % Algebraic equations
    sep_algs = sep.algs;
    sep_algs.gas = rmfield(sep_algs.gas,{'T','P'});
    sep_algs.liq = rmfield(sep_algs.liq,{'T','P'});

    alg = [                                                      ...
           casadi_struct2vec(rmfield(stream1.algs,               ...
                             {'Cp'}                              ...
                             ));                                 ...
           casadi_struct2vec(rmfield(stream2.algs,               ...
                             {'Cp','Vcdot','GVF'}                ...
                             ));                                 ...
           casadi_struct2vec(rmfield(stream3.algs,               ...
                             {}                                  ...
                             ));                                 ...
           casadi_struct2vec(rmfield(stream4.algs,               ...
                             {'rho','Vgdot','Vcdot','Z','GVF'}   ...
                             ));                                 ...
           casadi_struct2vec(rmfield(stream5.algs,               ...
                             {'Cp','rho','Vgdot','Vcdot','Z','GVF'} ...
                             ));                                 ...
           casadi_struct2vec(rmfield(chk.algs,                   ...
                             {'T'}                               ...
                             ));                                 ...
           casadi_struct2vec(sep_algs);                          ...
```

```
           casadi_struct2vec(rmfield(comp.algs,{}));                          ...
          ];

    % Differential equations
    ode = [                                                                   ...
           casadi_struct2vec(comp.odes);                                      ...
          ];

    % Boundary conditions
    s1 = {...
          {'stream1','T'},                                                    ...
          {'stream2','T'},                                                    ...
          {'stream3','T'},                                                    ...
          {'stream5','T'},                                                    ...
          {'stream1','P'},                                                    ...
          {'stream2','P'},                                                    ...
          {'stream3','P'},                                                    ...
          {'stream5','P'},                                                    ...
         };

    v1 = {...
          3.5,                                                                ...
          3.5,                                                                ...
          3.5,                                                                ...
          3.5,                                                                ...
          1.0,                                                                ...
           .9,                                                                ...
           .9,                                                                ...
           .9,                                                                ...
         };

    for i=1:length(s1)
        cell = s1{i};
        z                    = z_struct.(cell{1}).(cell{2});
        z_struct.(cell{1})   = rmfield(z_struct.(cell{1}),cell{2});
        z0_struct.(cell{1})  = rmfield(z0_struct.(cell{1}),cell{2});
        alg = substitute(alg ,z,v1{i});
        ode = substitute(ode ,z,v1{i});
    end
end
```

B.6 Open-loop Optimization

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%@Course       : Master Thesis Spring 2018
%@Task         : Open-loop optimization to obtain optimal control policy for
%              : the subsea system.
%@input        : none
%@output       : none

%@author       : Adriaen Verheyleweghen
%@modified     : Julie Berge Ims
%@organization: Department of Chemical Engineering, NTNU, Norway
%@requires     : MATLAB R2016a (not tested in other releases)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Provide path to Casadi installation
addpath('/Users/juliebergeims/downloads/casadi-matlabR2015a-v3.0.0')
import casadi.*

% Creating a model for optimization
[dae_x,dae_z,dae_u,~,dae_z0,dae_u0,dae_p,dae_ode,dae_alg] =              ...
                                      subsea_model(u_choke, u_comp);

% Size of time step
dt = 0.05;

% Vector with time steps
hlist = repmat(dt,1,N);

% Sum of all time steps
tf = sum(hlist);

%% Variable declarations

% Stochastic variables: GVF and degradation speed
dae_stoc.a      = dae_p.a;
dae_stoc.GVF    = dae_z.stream1.GVF;
dae_z.stream1   = rmfield(dae_z.stream1,'GVF');
dae_z0.stream1 = rmfield(dae_z0.stream1,'GVF');


% CVaR variables
CVaR   = struct( 'diff',MX.sym('diff'),                                  ...
                 'var',MX.sym('var'),                                    ...
                 'cvar',MX.sym('cvar')                                   ...
                 );

% Profit variable
```

```
Profit = struct('profit',MX.sym('profit'));

% RUL variables
RUL  = struct('lambda',MX.sym('lambda'),                                    ...
                  'K',MX.sym('K')                                           ...
             );

% Scale parameter
la = 5.7;
lb = 1.2;
lc = 2.8;
lambda = @(h) la - lb*h - lc*sqrt(h) ;

% Shape parameter
ka = 4.55;
kb = 0.1;
kc = 0.1;
K   = @(h)    ka + kb*h + kc*(h)^2 ;

% Weibull quantile function
var = @(lambda,k,alpha) (lambda*(-log(1-alpha))^(1/k));

% Differential variables
dae_x.clock = struct('time',MX.sym('time'));

% ODE expressions
dae_ode=[dae_ode 1];

% Algebraic variables
dae_z.RUL     =  struct('lambda',RUL.lambda,                                 ...
                        'K',RUL.K);

dae_z.CVaR    =  struct('var',CVaR.var,                                      ...
                        'cvar',CVaR.cvar);

dae_z.Profit  =  struct('profit',Profit.profit);

% Algebraic initial conditions
dae_z0.RUL    =  struct('lambda',la,                                         ...
                        'K',ka);

dae_z0.CVaR   =  struct('var',1,                                             ...
                        'cvar',0.8);

dae_z0.Profit = struct('profit',0.08);

% Algebraic expressions
 dae_alg = [dae_alg;                                                        ...
```

```
            dae_z.RUL.lambda     - lambda(dae_x.compressor.h);           ...
            dae_z.RUL.K          - K(dae_x.compressor.h);                ...
            dae_z.CVaR.var       - var(lambda(dae_x.compressor.h),       ...
                                   K(dae_x.compressor.h),alpha);         ...
            dae_z.CVaR.cvar      - cvar_func(lambda(dae_x.compressor.h), ...
                                   K(dae_x.compressor.h),alpha);         ...
            dae_z.Profit.profit - (dae_z.stream4.mgdot)*(dt*(1+0.015)^   ...
                                   (-60*dae_x.clock.time))               ...
           ];

%% Optimal Control Problem set-up

% DAE-struct
dae      = struct;
dae.x    = casadi_struct2vec(dae_x);
dae.z    = casadi_struct2vec(dae_z);
dae.p    = casadi_struct2vec(dae_u);
dae.s    = casadi_struct2vec(dae_stoc);
dae.ode  = casadi_struct2vec(dae_ode);
dae.alg  = dae_alg;

% Objective function
dae.quad = ( -dae_z.Profit.profit - dae_z.CVaR.cvar*0.025)';

%% Direct Collocation set-up

% Number of variables
nx = size(dae.x,1);
nu = size(dae.p,1);
nz = size(dae.z,1);
ns = size(dae.s,1);

% Degree of interpolating polynomial
d = 3;

% Obtain collocation points of specific order and scheme.
tau_root = casadi.collocation_points(d, 'radau');

% Obtain a function for collocation
collfun = simpleColl(dae,tau_root);
nlp = {};

%% Scenario-based collocation

% Three scenarios, based on "one" uncertain parameter, a
for scen=1:3

    % Symbolic primitive with given dimensions nx
```

```
    X0s = MX.sym('X0',nx);

    % Cell array [N+1,1] of empty matrices. Symbolic primitives
    Xs = cell(N+1,1);
    for i=1:N+1
        Xs{i} = MX.sym(['X_' num2str(i)],nx);
    end

    % Cell array [N,1] of empty matrices.Symbolic primitives with for-loop
    XCs = cell(N,1);
    Zs  = cell(N,1);
    Us  = cell(N,1);
    Ss  = cell(N,1);

    for i=1:N
        XCs{i} = MX.sym(['XC_' num2str(i)],nx,d);
        Zs{i}  = MX.sym(['Z_'  num2str(i)],nz,d);
        Us{i}  = MX.sym(['U_'  num2str(i)],nu,1);
        Ss{i}  = MX.sym(['S_'  num2str(i)],ns,1);
    end

    V_block = struct();
    V_block.X  = Sparsity.dense(nx,1);
    V_block.XC = Sparsity.dense(nx,d);
    V_block.Z  = Sparsity.dense(nz,d);
    V_block.U  = Sparsity.dense(nu,1);

    % Bounds on states and constraints
    lbx = {};
    ubx = {};
    lbg = {};
    ubg = {};

    % Objective function
    f = 0;

    % List of constraints
    g = {};

    % List of all decision variables (determines ordering)
    V    = {};
%% Define bounds

    % Bounds at 0<t<tf
    x_lb_k = casadi_vec(dae_x, -inf);
    x_ub_k = casadi_vec(dae_x,  inf);
    u_lb_k = casadi_vec(dae_u, 0,'N',0.75,'Z',0);
    u_ub_k = casadi_vec(dae_u, inf,'N',1.05,'Z',1);
```

```
    z_lb_k = casadi_vec(dae_z, -inf,'srg',0,'stw',0,                          ...
                                'stream4',{'P',1.5});
    z_ub_k = casadi_vec(dae_z, inf);

    % repeat for each collocation point
    z_lb_k = repmat(z_lb_k',d)';
    z_ub_k = repmat(z_ub_k',d)';

    % Gather all bounds at 0<t<tf
    lbx_k  = {casadi_vec(V_block,  0,'X',x_lb_k,'U',u_lb_k,'Z',z_lb_k)};
    ubx_k  = {casadi_vec(V_block,inf,'X',x_ub_k,'U',u_ub_k,'Z',z_ub_k)};

    % Bounds at t=tf
    x_lb_tf = {x_lb_k};
    x_ub_tf = {x_ub_k};

    % Initial guess
    guess       = [];
    x_guess     = casadi_struct2vec(dae_x0);
    z_guess     = casadi_struct2vec(dae_z0);
    u_guess     = casadi_struct2vec(dae_u0);

    for k=1:N

        % Add decision variables  xc: collocation points
        V = [V {casadi_vec(V_block,'X',Xs{k},'XC',XCs{k},                     ...
                               'Z',Zs{k},'U',Us{k})}];

        % Vector with inital guess
        guess =[guess;[repmat(x_guess,d+1,1);repmat(z_guess,d,1);u_guess]];

        lbx = [lbx lbx_k];
        ubx = [ubx ubx_k];

        if k==1
            tmp = {Xs{k}-X0s};
            g   = [g tmp];
            lbg = [lbg {zeros(size(tmp{:}))}];
            ubg = [ubg {zeros(size(tmp{:}))}];

            % Enforce nonanticipativity
            if scen==1
                U1 = Us{1};
            else
                tmp = {Us{k}-U1};
                g = [g tmp];
                lbg = [lbg {zeros(size(tmp{:}))}];
                ubg = [ubg {zeros(size(tmp{:}))}];
```

```
            end
        end

        % Obtain collocation expressions
        coll_out = collfun.call({hlist(k),Xs{k},XCs{k},Zs{k},Us{k},Ss{k}});

        tmp = coll_out(2);
        g   = [g tmp];                                    % System dynamics
        lbg = [lbg {zeros(size(tmp{:}))}];
        ubg = [ubg {zeros(size(tmp{:}))}];

        tmp = coll_out(3);
        g   = [g tmp];                              % Algebraic constraints
        lbg = [lbg {zeros(size(tmp{:}))}];
        ubg = [ubg {zeros(size(tmp{:}))}];

        tmp = {Xs{k+1}-coll_out{1}};
        g   = [g tmp];                            % Gap closing constraints
        lbg = [lbg {zeros(size(tmp{:}))}];
        ubg = [ubg {zeros(size(tmp{:}))}];

        % Cost function
        f = f + coll_out{4};
    end

    % Add final x to decision variables
    V     = [V     , Xs(end)];
    guess = [guess ; 1;1];

    % Bounds for final t
    lbx = [lbx x_lb_tf];
    ubx = [ubx x_ub_tf];

    %% Define the NLP
    nlp{scen} = struct('x',vertcat(V{:}),                               ...
                       'f',f,                                           ...
                       'g',vertcat(g{:}),                               ...
                       'p',vertcat(X0s,Ss{:})                           ...
                       );
end

% Generate the full-space NLP by combining the individual scenario problems
nlp_full_space = nlp{1};
for i = 2:scen
    nlp_full_space.x = horzcat(nlp_full_space.x,nlp{i}.x);
    nlp_full_space.g = vertcat(nlp_full_space.g,nlp{i}.g);
    nlp_full_space.p = vertcat(nlp_full_space.p,nlp{i}.p);
    nlp_full_space.f = nlp_full_space.f + nlp{i}.f;
```

```
end

nlpfun = Function('nlp',nlp_full_space,char('x','p'),char('f','g'));

opts = struct('warn_initial_bounds',false,                                ...
              'gather_stats',true,                                        ...
              'print_time',false,                                         ...
              'ipopt',struct('linear_solver','mumps',                     ...
                             'max_iter',5E2,                              ...
                             'warm_start_init_point','yes',               ...
                             'mu_init',1E-5,                              ...
                             'replace_bounds','yes',                      ...
                             'print_level',5,                             ...
                             'tol',1E-8)                                  ...
              );

% Creating nlp sovler for the nlp_full_space system with the options 'opts'
solver = nlpsol('solver','ipopt',nlp_full_space,opts);

% Scenario realizations
b1 = [scenparam(1),1];
b2 = [scenparam(2),1];
b3 = [scenparam(3) ,1];

% Scenarios
s11 = [b1;repmat(b1,N-1,1)]';
s22 = [b2;repmat(b2,N-1,1)]';
s33 = [b3;repmat(b3,N-1,1)]';

%% Solve the NLP

x0 = casadi_struct2vec(dae_x0);
x0_orig= repmat(guess(:),1,scen);

res = solver('x0',x0_orig,                                                ...
             'p',[full(x0);s11(:);full(x0);s22(:);full(x0);s33(:)],       ...
             'lbg',zeros(size(nlp_full_space.g)),                         ...
             'ubg',zeros(size(nlp_full_space.g)),                         ...
             'lbx',repmat(vertcat(lbx{:}),1,scen),                        ...
             'ubx',repmat(vertcat(ubx{:}),1,scen)                         ...
             );

%% Plotting

close all
figure('units','normalized','outerposition',[0 0 1 1])
hold on
```

```
input_comp = [];
input_choke = [];
for i = 1:scen

    % Plotting Variables
     vars_Z = {'stream4.mgdot','RUL.K','RUL.lambda','Profit.profit',...
               'CVaR.cvar','CVaR.var'...
               };
    vars_X = {'compressor.h'};
    vars_U = {'compressor.N','choke.Z'};

    dim = size(casadi_struct2vec(V_block));
    tmp = DM(reshape(full(res.x),[res.x.size1(),scen]));
    res_split = vertsplit(tmp,dim(1));

    while true % Plotting

        % number of plots
         n_plots = length(vars_Z)+length(vars_X)+length(vars_U);
        flr = floor(sqrt(n_plots));
        cei = ceil(sqrt(n_plots));
        if flr==sqrt(n_plots)
            plot_dims = [flr,flr];
        elseif flr*cei>=n_plots
            plot_dims = [flr,cei];
        else
            plot_dims = [cei,cei];
        end

        counter = 1;

        res_Z  = {};
        res_X  = {};
        res_XC = {};
        res_U  = {};

        for r = res_split(1:end-1)
            r = full(r{:});
            r = r(:,i);
            rs = casadi_vec2struct(V_block,r(:));
            res_Z  = [res_Z  {rs.Z} ];
            res_X  = [res_X  {rs.X} ];
            res_XC = [res_XC {rs.XC}];
            res_U  = [res_U  {rs.U} ];
        end
        res_Z  = full([res_Z{:}])';
        res_X  = full([res_X{:}])';
        res_XC = full([res_XC{:}])';
```

```
        res_U  = full([res_U{:}])';

        input_comp(i)  = res_U(1,1);
        input_choke(i) = res_U(1,2);
        %% Plot Z
        [nr,nc] = size(res_Z);
        mat_Z = mat2cell(res_Z,[nr],ones(nc,1));
        indices_Z = casadi_vec2struct(dae_z,1:nc);

        t_z = [];
        tmp = N*(cumsum([0 hlist(1:end-1)]));
        for t=1:length(tmp); t_z =[t_z tmp(t)+tau_root*hlist(t)*N]; end
        t_z = t_z'/N;

        z_lb = full(z_lb_k(:,1));
        z_ub = full(z_ub_k(:,1));
        for j = 1:length(vars_Z)
            subplot(plot_dims(1),plot_dims(2),counter)
            counter = counter+1;
            hold on
            str = strsplit(vars_Z{j},'.');
            index = full(indices_Z.(str{1}).(str{2}));
            plot(t_z,ones(size(t_z)).*z_lb(index),'r--')
            plot(t_z,ones(size(t_z)).*z_ub(index),'r--')
            plot(t_z,mat_Z{index})
            xlim([0,N*dt])
            title(vars_Z(j))
            xlabel('Time')
            ylabel(str{2})
        end
        %% Plot X

        [nr,nc] = size(res_XC);
        mat_XC = mat2cell(res_XC,[nr],ones(nc,1));
        indices_XC = casadi_vec2struct(dae_x,1:nc);

        t_xc = [];
        tmp = N*(cumsum([0 hlist(1:end-1)]));
        for t=1:length(tmp); t_xc =[t_xc tmp(t)+tau_root*hlist(t)*N]; end
        t_xc = t_xc'/N;

        xc_lb = full(x_lb_k(:,1));
        xc_ub = full(x_ub_k(:,1));
        for i = 1:length(vars_X)
            subplot(plot_dims(1),plot_dims(2),counter)
            counter = counter+1;
            hold on
            str = strsplit(vars_X{i},'.');
```

```
                index = full(indices_XC.(str{1}).(str{2}));
                plot(t_xc,ones(size(t_xc)).*xc_lb(index),'r--')
                plot(t_xc,ones(size(t_xc)).*xc_ub(index),'r--')
                plot(t_xc,mat_XC{index})
                xlim([0,N*dt])
                title(vars_X(i))
                xlabel('Time')
                ylabel(str{2})
            end
            %% Plot U
            [nr,nc] = size(res_U);
            mat_U = mat2cell(res_U,[nr],ones(nc,1));
            indices_U = casadi_vec2struct(dae_u,1:nc);

            t_u = N*(cumsum([0 hlist(1:end-1)]));
            t_u = t_u'/N;

            u_lb = full(u_lb_k(:,1));
            u_ub = full(u_ub_k(:,1));
            for i = 1:length(vars_U)
                subplot(plot_dims(1),plot_dims(2),counter)
                counter = counter+1;
                hold on
                str = strsplit(vars_U{i},'.');
                index = full(indices_U.(str{1}).(str{2}));
                plot(t_u,ones(size(t_u)).*u_lb(index),'r--')
                plot(t_u,ones(size(t_u)).*u_ub(index),'r--')
                stairs(t_u,mat_U{index})
                xlim([0,N*dt])
                title(vars_U(i))
                xlabel('Time')
                ylabel(str{2})
            end
            hold off
            break
        end
    end
```

.

B.7 Closed-loop Control

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%@Course      : Master Thesis Spring 2018
%@Task        : Closed-loop model predictive controller. Solves open-loop
%               optimization problem periodically and adds random
%               disturbance to the inputs
%@input       : none
%@output      : none

%@author      : Julie Berge Ims
%@organization: Department of Chemical Engineering, NTNU, Norway
%@created     : February 2018
%@requires    : MATLAB R2016a (not tested in other releases)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
clc

% Provide path to Casadi installation
addpath('/Users/juliebergeims/downloads/casadi-matlabR2015a-v3.0.0')
import casadi.*

%% Parameterization of open-loop

% Initial values for the differential states
dae_x0  = struct('compressor',struct('h',0.01)                         ,...
                 'clock',struct('time',0.0));

% Initial process input
u_comp = 0.85;                                    % Compressor speed [input]
u_choke = 0.565;                                  % Choke opening    [input]

% Number of time steps for prediction horizon
N  = 20;

% CVaR parameters
alpha = 0.001;                                   % Confidence level for risk

% Uncertain parameter realizations
scenparam = [0.9 1 1.1];
%% Closed-lopp simulation

run OL_main
%close all

% DAE-struct for integration
int_dae = struct('x', dae.x,                                            ...
```

```
                  'z', dae.z,                                                ...
                  'p',[dae.p ; dae.s],                                       ...
                  'ode', dae.ode,                                            ...
                  'alg', dae.alg );

% Integrator using DAE-integrator "idas"
DAE_integrator = integrator('integrator',                                    ...
                             'idas',int_dae, struct('t0',0,'tf',0.05));

% Integrating the DAE-struc
sol = DAE_integrator('x0', res_X(1,:)',                                      ...
                      'z0', res_Z(1,:)',                                     ...
                       'p', [res_U(1,:),scenparam(3) , 1]');

% Obtaining initial measured state(X) and first control input (U)
x_temp(1,:)        = sol.xf.full()';
u_temp_comp(1,:)   = input_comp(:);
u_temp_choke(1,:)  = input_choke(:);

% Repeat process with shrinking horizon
for w = 2:1:20

    % Using first control input from the OL opt
    u_comp = res_U(1,1);
    u_choke= res_U(1,2);

    % Updating initial conditions
    dae_x0.compressor.h = x_temp(w-1,1) + rand(1,1)*x_temp(w-1,1)/10;
    dae_x0.clock.time   = x_temp(w-1,2) + rand(1,1)*x_temp(w-1,2)/10;

    % OL opt by decreasing the prediction horizon by on time step
    N = N-1;
    run OL_main
    close all

    % Integr. the DAE-struct
    sol = DAE_integrator('x0',sol.xf.full(),                                 ...
                          'z0',sol.zf.full(),                                ...
                          'p', [res_U(1,:),scenparam(3), 1]');

    % Obtaining the CL states and inputs
    x_temp(w,:) = sol.xf.full()';
    u_temp_comp(w,:)   = input_comp(:);
    u_temp_choke(w,:) = input_choke(:);
end
```

.

B.8 Conditional Value-at-Risk

```matlab
function cvar_out = cvar_func(lambda,k,alpha )
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%@Course       : Master Thesis Spring 2018
%@Task         : Calculating Conditional Value-at-Risk at time t for lambda,
%                k and alpha
%@input        : scale parameter(lambda) and shape parameter (k) in the
%                Weibull distribution, confidence level for risk (alpha)
%@output       : value for Conditional Value-at-Risk

%@author       : Julie Berge Ims
%@organization: Department of Chemical Engineering, NTNU, Norway
%@created      : February 2018
%@requires     : MATLAB R2016a (not tested in other releases)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Weibull quantile function
    quantile = @(lambda,k,alpha) (lambda*(-log(1-alpha))^(1/k));

    % VaR-variable declaration
    var_out = 0;

    % Step-size in integration
    dy = 0.0001;

    % Integrating VaR from 0.0 to alpha
    for i = 0.00:dy:alpha
        var_out = var_out + (quantile(lambda,k, i))*dy;
    end

    % Dividing the integral by alpha
    cvar_out = var_out/alpha;

end
```

.

B.9 CasADi Function for Collocation

```
function [G] = simpleColl(dae,tau_root)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%@Task         : Function that generates a function for collocation
%@input        : differential algebraic equations (dae), struct (tau_root)
%
%@output       : collocation function (G)

%@author       : Joris Gillis, Rien Quirynen, Joel Andersson,
%                Sebastien Gros and Moritz Diehl
%@modified     : Adriaen Verheyleweghen
%@organization: Faculty of Engineering, University of Frieburg, Germany
%@requires     : MATLAB R2016a (not tested in other releases)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  import casadi.*

  daefun = Function('fun',dae,char('x','z','p','s'),...
                              char('ode','alg','quad'));

  % Degree of interpolating polynomial
  tau_root = [0, tau_root];

  d = length(tau_root)-1;

  % Coefficients of the collocation equation
  C = zeros(d+1,d+1);

  % Coefficients of the continuity equation
  D = zeros(d+1,1);

  % Dimensionless time inside one control interval
  tau = SX.sym('tau');

  % For all collocation points
  for j=1:d+1
    % Construct Lagrange polynomials to get the polynomial basis at the
    % collocation point
    L = 1;
    for r=1:d+1
      if r ~= j
        L = L * (tau-tau_root(r))/(tau_root(j)-tau_root(r));
      end
    end
    lfcn = Function('lfcn', {tau},{L});
    out = lfcn.call({1.0});
```

```

    % Evaluate the polynomial at the final time to get the coefficients
    % of the continuity equation
    D(j) = full(out{1});

    % Evaluate the time derivative of the polynomial at all collocation
    % points to get the coefficients of the continuity equation
    tfcn = lfcn.tangent();
    for r=1:d+1
      out = tfcn.call({tau_root(r)});
      C(j,r) = full(out{1});
    end
  end

  % Time step
  h   = MX.sym('h',1);

  % State variable
  CVx   = MX.sym('x',dae.x.size1(),1);

  % Helper state variables
  CVCx = MX.sym('x',dae.x.size1(),d);

  % Algebraic variables
  CVz   = MX.sym('z',dae.z.size1(),d);

  % Fixed parameters (controls)
  CVp   = MX.sym('p',dae.p.size1());
  CVs   = MX.sym('s',dae.s.size1());

  X = [CVx CVCx];

  g_alg   = {};
  g_cont = {};

  % For all collocation points
  quad_k = 0;
  for j=2:d+1

    % Get an expression for the state derivative at the collocation point
    xp_jk = 0;
    for r=1:d+1
      xp_jk = xp_jk + C(r,j)*X(:,r);
    end
    % Add collocation equations to the NLP
    out = daefun.call({CVCx(:,j-1),CVz(:,j-1),CVp,CVs});
    ode = out{1};
    alg = out{2};
```

```
    quad = out{3};
    quad_k = h*quad_k+quad;
    g_cont = [g_cont {h*ode - xp_jk}];
    g_alg  = [g_alg {alg}];
  end
  % Get an expression for the state at the end of the finite element
  xf_k = 0;
  for r=1:d+1
    xf_k = xf_k + D(r)*X(:,r);
  end
  G = Function('G',{h,CVx,CVCx,CVz,CVp,CVs},...                        % Inputs
     {xf_k,vertcat(g_cont{:}),vertcat(g_alg{:}),quad_k});              % Outputs

end
```

.

B.10 CasADi Struct

```
function [out] = casadi_struct(s,varargin)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%@Course       : Master Thesis Spring 2018
%@Task         : Function for creating a struct from a set of input
%                arguments
%@input        : input arguments (varargin), struct (s)
%
%@output       : struct (out)

%@author       : Joris Gillis, Rien Quirynen, Joel Andersson,
%                Sebastien Gros and Moritz Diehl
%@modified     : Adriaen Verheyleweghen
%@organization: Faculty of Engineering, University of Frieburg, Germany
%@requires     : MATLAB R2016a (not tested in other releases)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  import casadi.*

  out = struct;

  origs = s;

  if ischar(varargin{1})
      default = 0;
  else
      default = varargin{1};
      varargin = varargin(2:end);
  end

  tmp = {default,varargin{:}};

  subs = {};

  for i = fliplr(1:length(tmp)/2)
      c = varargin{2*i};
      if isa(c,'cell')
          try
              subs = {subs{:},varargin{2*i-1},s.(varargin{2*i-1}),c};
              s = rmfield(s,varargin{2*i-1});
              tmp = {tmp{1:2*i-1},tmp{2*(i+1):end}};
          catch err
              rethrow(err)
          end
      end
  end

```

```
  for j = 1:length(subs)/3
    arg = subs{3*j};
    out.(subs{3*j-2}) = casadi_struct(subs{3*j-1},arg{:});
  end

  for k=fieldnames(s)'
    k = k{1};
    found = -1;
    for l=1:length(varargin)/2
        if strcmp(varargin{2*l-1},k)
           found = l;
           break;
        end
    end
    if found>0
      if isa(s.(k),'struct')
          e = casadi_struct(s.(k),varargin{2*found:end});
      else
          e = varargin{2*found};
      end
      if isscalar(e)
        dims = size(s.(k));
        e = repmat(e,dims(1),dims(2));
      end
      dims = size(s.(k));
      assert(size(e,1)==dims(1))
      assert(size(e,2)==dims(2))
    else
      if isa(s.(k),'struct')
          e = casadi_struct(s.(k),tmp{1:end});
      else
          dims = size(s.(k));
          e = default*DM.ones(dims(1),dims(2));
      end
    end
    out.(k) = e;
  end
  out = orderfields(out,origs);
end
```

.

B.11 CasADi Vector

```
function [out] = casadi_vec(varargin)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%@Course      : Master Thesis Spring 2018
%@Task        : Function for creating a vector from a set of input
%               arguments
%@input       : input arguments (varargin)
%
%@output      : vector (out)

%@author      : Joris Gillis, Rien Quirynen, Joel Andersson,
%               Sebastien Gros and Moritz Diehl
%@organization: Faculty of Engineering, University of Frieburg, Germany
%@requires    : MATLAB R2016a (not tested in other releases)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  out = casadi_struct2vec(casadi_struct(varargin{:}));
end
```

.

B.12 CasADi Struct to Vector

```
function [out] = casadi_struct2vec(s)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%@Course       : Master Thesis Spring 2018
%@Task         : Function that converts a struct to a vector
%@input        : struct (s)
%
%@output       : vector (out)

%@author       : Joris Gillis, Rien Quirynen, Joel Andersson,
%                Sebastien Gros and Moritz Diehl
%@organization: Faculty of Engineering, University of Frieburg, Germany
%@requires     : MATLAB R2016a (not tested in other releases)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  flat = {};
  if isstruct(s)
    for f=fieldnames(s)'
      flat = {flat{:} casadi_struct2vec(s.(f{1}))};
    end
    out = vertcat(flat{:});
  elseif iscell(s)
    for i=1:length(s)
       flat = {flat{:} casadi_struct2vec(s{i})};
    end
    out = vertcat(flat{:});
  else
    try
      out = vec(s);
    catch
      import casadi.*
      out = vec(DM(s));
    end
  end
end
```

.

B.13 CasADi Vector to Struct

```
 function [out] = casadi_vec2struct(s,vec)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%@Course       : Master Thesis Spring 2018
%@Task         : Function that converts a vector to a struct
%@input        : vector (vec), struct (s)
%
%@output       : struct (out)

%@author       : Joris Gillis, Rien Quirynen, Joel Andersson,
%                Sebastien Gros and Moritz Diehl
%@modified     : Adriaen Verheyleweghen
%@organization: Faculty of Engineering, University of Frieburg, Germany
%@requires     : MATLAB R2016a   (not tested in other releases)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  import casadi.*
  assert(isvector(vec))
  try
      vec.sparsity();
  catch
      vec = DM(vec);
  end
  flat = {};
  if isstruct(s)
    out = struct;
    sizes = 0;
    for f=fieldnames(s)'
      dim = size(casadi_struct2vec(s.(f{1})));
      sizes = [sizes,sizes(end)+dim(1)];
    end
    comps = vertsplit(vec,sizes);
    i = 1;
    for f=fieldnames(s)'
      out.(f{1}) = casadi_vec2struct(s.(f{1}),comps{i});
      i = i+1;
    end
  elseif iscell(s)
    out = cell(size(s));
    sizes = 0;
    for i=1:length(s)
      n = size(casadi_struct2vec_new(s{i}),1);
      sizes = [sizes,sizes(end)+n];
    end
    comps = vertsplit(vec,sizes);
    for i=1:length(s)
      out{i} = casadi_vec2struct(s{i},comps{i});
```

```
    end
      else
    out = reshape(vec,size(s));
  end
end
```

.