

Sindre O. Skrede

Application of Advanced Methods for ILI Data Denoising and Reliability Assessment of Corroding Pipelines

Master's thesis in Marine Technology

Supervisor: Bernt J. Leira

June 2020



Norwegian University of
Science and Technology

Sindre O. Skrede

Application of Advanced Methods for ILI Data Denoising and Reliability Assessment of Corroding Pipelines

Master's thesis in Marine Technology
Supervisor: Bernt J. Leira
June 2020

Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

Preface

This Master Thesis is the result of work conducted by Sindre O. Skrede during the tenth semester at The Norwegian University of Science and Technology (NTNU), in the spring of 2020. The theory review of this thesis is, in part, based on the project thesis conducted in the Autumn semester of 2019. The relevant part is referred to in the text.

The most time-consuming part of this work has been understanding and treatment of challenging datasets, resulting in a filtering algorithm suitable for the treatment of entire folders of data. As parts of the filtering algorithm were vulnerable to missing data, there was more development required than anticipated before the real-case datasets could be analysed correctly. This has in part lead to lowering of ambitions for the capacity investigation as the project went on. Communication with and getting guidance from the supervisors have also been more challenging due to the circumstances with the Coronavirus, but I believe we have made the best of the current situation.

I would, in particular, like to thank the following for being available for discussion of problems with the thesis work:

- Bernt J. Leira for being the main supervisor, and for guidance regarding the structural reliability.
- Zhenhui Liu for input regarding data filtering and processing.

A thanks is also given to Aker Solutions AS for facilitating the project, lending out software and providing datasets for the development.

Abstract

This project aims to establish methods for processing data and for estimating the current and future capacity of corroding pipelines, based on in-line inspection (ILI) datasets consisting of remaining wall thickness (RWT) and stand-off (SO) data. Pipeline corrosion is a random process dependent on a variety of factors, and is virtually impossible to predict precisely with known methods. For ensuring the integrity of pipelines, ILI represent a possibility to investigate the current state of the corrosion. The datasets from an ILI of a corroded pipeline are, however, often concerned with uncertainties, noise, and misreadings. For this not to add further uncertainties into a capacity calculation, some data filtering is required. Also, the results from these inspections are only giving an uncertain momentary image. Future growth is hard to determine from this with deterministic methods without adding safety factors, giving an underestimation of the capacity. Reliability methods provide better means to take the inherent randomness of the corrosion process into account and, at the same time, control the level of conservatism added. However, these methods are dependent on modelling the underlying processes correctly, which means that detailed information about the corrosion process and the pipeline in question is needed.

The data filtering methodology developed in this project is a further development of the program rbpGen by Aker Solutions AS, based on methodology from DNVGL-RP-F101. The program developed was used on a fictional test dataset and a real case dataset from a pipeline. The basics behind pipeline corrosion and developments within reliability methods used on corroding pipelines were reviewed and acted as guidelines for the methodology used for estimating the reliability of the real-case dataset. The results indicate that the new program produces slightly better data filtering than rbpGen, and is more stable with regards to missing data treatment. The reliability analysis executed in this project can take multiple random parameters and failure modes into account, providing reasonable results. A drawback is the computational efficiency, which limits the accuracy of the analysis. The results obtained from the reliability analysis is only valid for the dataset in question, i.e. 100 m of a pipeline, and indicate that the pipeline section has sufficient capacity at present state until at least five years from inspection time. The analysis is partly based on parameter descriptions from the literature, making the time-dependent reliability estimates questionable.

Samandrag

Dette prosjektet har som mål å etablere metodar for dataprossesering og estimering av noverande og framtidig kapasitet av korroderande røyrledningar, basert på data frå in-line inspeksjonar (ILI) av gjennverande veggtjukkleik (RWT) og sensor-avstand til rørveggen (SO). Korrosjon på røyrledningar er ein tilfeldig prosess avhengig av fleire faktorar, og er tilnærma umogleg å predikere eksakt med kjende metodar. ILI er ein måte å undersøkje den noverande korrosjonstilstanden til røyrledningar og dermed sikre integriteten, men datasetta frå ILI inneheld ofte usikkerheit i tillegg til støy og feilavlesingar. Datafiltrering er dermed nødvendig for å unngå å ta inn usikkerheit i kapasitetsberekningane. I tillegg gir ILI-data også berre eit usikkert augenblinksbilete av situasjonen. Framtidig korrosjonsvekst er vanskeleg å bestemme frå dette biletet med deterministiske metodar utan å legge til store sikkerheitsfaktorar, noko som vil føre til at kapasiteten blir underestimert. Pålitelighetsmetodar gir betre verktøy til å handtere den naturlige tilfeldigheita som er i korrosjonsprossessen, og samtidig kan ein kontrollere nivået på sikkerheitsmarginen. Bruk av desse metodane krevjer at dei underliggende prosessane er modellert riktig, noko som betyr at det er nødvendig med detaljert informasjon om korrosjonsprosessen og om røyrledninga.

Metoden for datafiltrering som er utvikla i dette prosjektet er ei vidareføring av programmet rbpGen, som er utvikla av Aker Solutions AS. Programmet rbpGen er igjen basert på metodologi frå DNVGL-RP-F101. Programmet utvikla i dette prosjektet vart brukt på eit fiktivt datasett og på eit faktisk datasett frå ein røyrledning. Prinsippa bak røyrledningskorrosjon og utvikling innen pålitelighetsmetodar brukt på korroderande røyrledningar har blitt studert, og har blitt brukt som rettleiing for metodane som estimerer påliteligheta av det faktiske datasettet. Resultata indikerer at det nye programmet produserer noko betre datafiltrering samanlikna med rbpGen, og handterer manglande data meir stabilt. Pålitelighetsanalysen utført i dette prosjektet kan handtere fleire stokastiske parametre og sviktmodar, og gir fornuftige resultat. Ei begrensing er effektiviteten til programmet, som gjer at nøyaktigheita til berekningane blir avgrensa. Resultata frå pålitelighetsanalysen er berre gyldig for datasettet som er analysert, som representerer 100 meter av ein røyrledning, og indikerer at røyrledningen har tilstrekkelig kapasitet på noverande tidspunkt til minst fem år fram i tid. Analysen er delvis basert på parameterbeskrivelsar frå litteraturen, noko som gjer dei tidsavhengige resultatata diksutable.

Contents

Preface	i
Abstract	ii
Samandrag	iii
1 Introduction	1
1.1 Background	1
1.2 Scope of the thesis	2
2 Theoretical Background: Reliability methods	4
2.1 Level I analysis	5
2.1.1 Description of limit state design methods	5
2.1.2 Limitations with Level I methods - relationship to level II	6
2.2 Level II analysis - Safety index methods	7
2.2.1 Basic variables, Reliability Index, and the Failure Surface	7
2.2.2 Cornell Safety Index	8
2.2.3 Hasofer and Lind's Reliability Index	8
2.2.4 Correlated variables	9
2.2.5 Relations between Level II and Level III methods	10
2.3 Level III analysis	11
2.3.1 FORM and SORM	12
2.3.2 Monte Carlo Simulations: General overview	14
2.3.3 Importance Sampling	14
2.3.4 Directional simulation	15
3 Literature review: Corrosion of pipelines	17
3.1 General description of Corrosion damage	17
3.1.1 Common forms of corrosion damage	17
3.1.2 Parameters deciding the corrosion	19
3.1.3 Growth models	20
3.2 Review of DNV-GL RP F101: Standard approach	21
3.2.1 Brief review of the calibrated safety factors approach used in DNVGL-RP-F101	22
3.2.2 Brief review of the Allowable stress design approach	24
3.3 Assessment of long axial internal corrosion defects according to DNV-GL RP F101 .	25
3.3.1 Level 0: Feature list	25
3.3.2 Level 1: Detailed WT and SO data	26
3.3.3 Level 2: Establish river bottom profile	26
3.3.4 Level 3: Calculate capacity	27
3.3.5 Level 4: Corrosion development	28
3.4 Reliability Analyses on corroding pipelines: An overview	29
3.4.1 Limit states	29
3.4.2 Deterministic corrosion growth models	30
3.4.3 Probabilistic growth models	30
3.4.4 Hierarchical Modelling of defect growth	31
3.5 Markov Chain models: Corrosion growth and usage in reliability analyses	33
3.6 Monte Carlo Simulation for reliability analysis on corroding pipelines	34
3.6.1 Limit States	34
3.6.2 Formulation for the system reliability analysis	35
3.7 FORM-based Reliability Analysis on pipelines	37
3.7.1 Limit state formulation	37

3.7.2	FORM Formulations	38
3.7.3	Equivalent limit state functions	38
3.7.4	System failure probabilities	39
3.8	Notes on the adequacy of the probabilistic models	41
3.9	Inspection method	42
4	Methodology description	43
4.1	An Overview: Description of inputs and of the overall flow of the method	43
4.1.1	Data layout and description	43
4.2	Methodology in <i>rbpGen</i> from Aker Solutions	45
4.2.1	Brief algorithm description	45
4.2.2	GUI layout and outputs	46
4.2.3	The DNV-Filter	47
4.2.4	The Median filter	49
4.2.5	The Gaussian filter	49
4.2.6	RBP/MAV Calculation	49
4.3	Description of the program developed in this project	50
4.3.1	Packages and dependencies	51
4.3.2	Program set-up and inputs	51
4.3.3	Data filtering algorithm description and output	52
4.3.4	Changes specific to the data filtering	53
4.3.5	Missing data treatment	54
4.3.6	Listing of defects	55
4.4	Example calculations estimating the reliability of the structure	56
4.4.1	Limit state formulation	56
4.4.2	Random Parameters	58
4.4.3	Reliability analysis set-up	59
5	Results and specific discussion of results	60
5.1	Verification of the new filtering program	60
5.1.1	Fabricated test files	60
5.1.2	Real test files: Data analysis of three real sections	64
5.1.3	Real test file: Section 1	65
5.1.4	Real test file: Section 2	67
5.1.5	Real test file: Section 3	69
5.2	An overview analysis of the pipeline dataset	71
5.2.1	Defect Listing results	73
5.3	Reliability analysis using generated defect list	75
5.3.1	MCS initial state, 10000 cycles, 99 defects	75
5.3.2	Time dependent MCS: 10000 cycles, 15 defects	76
5.3.3	Time-dependent MCS: 10000 cycles, 60 defects	77
5.3.4	Time-dependent MCS: 100 000 cycles, 4 defects	77
6	General Discussion	78
6.1	Discussion regarding the data treatment	78
6.2	About the defect listing algorithm and the real case dataset	78
6.3	About the Reliability analysis	79
7	Conclusion and Recommendations	81
7.1	Regarding methodology and results	81
7.2	Further work	81

References **83**

A Flowcharts **I**

 A.1 Flowchart of the entire program I

 A.2 Flowchart of the defect listing algorithm II

 A.3 Flowchart of the MCS-program III

B Plots **IV**

 B.1 Contour plots IV

 B.2 Results from MCS with zero probability of failure V

List of Figures

2.1	Illustration of load and resistance in level I analysis.	6
2.2	Simple 2D-case of a failure surface	7
2.3	Failure function with Hasofer-Lind safety index (2D).	9
2.4	2D-illustration of a failure surface, the FORM-approximation and the 2nd order SORM approximation.	13
2.5	2D-illustration of a failure surface and the approximation using spherical segments.	16
3.1	Flowchart over the assessment procedure in the recommended practice, taken from DNV-GL (2019a).	21
3.2	Overview of the levels of deliveries from UT inspection.	25
3.3	Assessment flow diagram of the capacity calculations performed in level 3.	27
3.4	Hierarchical representation of uncertainties for modelling system deterioration.	32
3.5	Principle of ultrasonic inspection for measuring of remaining wall thickness.	42
4.1	First part of an example data file.	44
4.2	Example of a filled out GUI for rbpGen.	46
4.3	Example of filtered data using the DNV-filter.	47
4.4	Plots of RBP- and MAV-datasets using rbpGen with the specified example inputs.	47
4.5	Flowchart of the ILI data processing following DNV-RP F101.	48
4.6	Weight factors applying linear filters for a kernel size of 5.	50
4.7	Illustration of MAV-calculation from a RBP.	50
4.8	Illustration of the output data structure, here presented with the output from the DNV-filter.	53
4.9	Illustration of the defect list data structure.	54
5.1	Contour of the dataset with no missing data, made using rbpGen.	60
5.2	Contour of the dataset with no missing data, made using the new program.	60
5.3	Results from File 1 using rbpGen	61
5.4	Results from File 1 using the new program.	61
5.5	Contour of the dataset with four columns missing, made using rbpGen.	62
5.6	Contour of the dataset with four columns missing, made using the new program.	62
5.7	Results from File 1 using rbpGen	63
5.8	Results from File 2 using the new program.	63
5.9	Quality-plots for the test-sections. The value on the y-axis is the number of missing datapoints per mm Odometer value.	64
5.10	Contour from rbpGen of the section between 3000 and 4000 mm.	65
5.11	Contour from the new program, of the section between 3000 and 4000 mm.	65
5.12	Results of the section between 3000 and 4000 mm using rbpGen.	66
5.13	Results of the section between 3000 and 4000 mm using the new program.	66
5.14	Contour from rbpGen of the section between 20000 and 21000 mm.	67
5.15	Contour from the new program, of the section between 20000 and 21000 mm.	67
5.16	Results of the section between 20000 and 21000 mm using rbpGen.	68
5.17	Results of the section between 20000 and 21000 mm using the new program.	68
5.18	Contour from rbpGen of the section between 40000 and 41000 mm.	69
5.19	Contour from the new program, of the section between 40000 and 41000 mm.	69
5.20	Results of the section between 40000 and 41000 mm using rbpGen.	70
5.21	Results of the section between 40000 and 41000 mm using the new program.	70
5.22	Quality plot of the entire dataset. Y-axis denotes the number of missing data per mm Odometer value.	71
5.23	Overview of the RBP for the entire dataset.	72
5.24	Overview of the MAV of the entire dataset.	72
5.25	Grouping of defects from the defect listing in the odometer interval [0, 25000]mm.	73
5.26	The worst pressure parameters in each iteration (not necessarily appearing together).	75

5.27	Minimum wall thickness for each iteration.	75
5.28	Resulting plots for MCS-run with 10000 cycles, 9 defects.	76
5.29	Resulting plots for MCS-run with 10000 cycles, 60 defects.	77
A.1	Flowchart of the entire program.	I
A.2	Flowchart of the defect listing algorithm.	II
A.3	Flowchart of the Monte Carlo Simulation program	III
B.1	Grouping of defects from the defect listing in the odometer interval of [0,100000]mm.	IV
B.2	Contour plots of the entire dataset.	IV
B.3	Resulting plots for MCS-run with 100 000 cycles, 5 defects.	V

List of Tables

2.1	Design limit states (Moan, 1994).	5
3.1	Safety class and target annual failure probability for ultimate limit state (ULS) (DNV-GL, 2019a).	22
4.1	Constants in the burst capacity equations.	57
4.2	Distribution of parameters for the observed defects, with error margins specified from the inspection company.	58
4.3	Distribution of material parameters and pipeline characteristics.	58
5.1	Overview of the most severe defects along the pipeline, i.e. with depth larger than 5 mm. RWT is the remaining wall thickness.	74

List of Abbreviations

ASD	Allowable Stress Design
CoV	Coefficient of Variance
FE	Finite Elements
FEM	Finite Element Modelling
FORM	First Order Reliability Method
GEVD	Generalised Extreme Value Distribution
ILI	In-line Inspection
IS-MCS	Importance-Sampled Monte Carlo simulation
LRFD	Load and Resistance Factor Design
MAV	Moving Average
MCMC	Markov Chain Monte Carlo
MCS	Monte Carlo Simulation
MFL	Magnetic Flux Leakage
MIC	Microbiologically-Influenced Corrosion
pdf	Probability Density Function
RBP	River Bottom Profile
RP	Recommended Practice (from DNV-GL)
RWT	Remaining Wall Thickness
SO	Stand-Off
SORM	Second Order Reliability Method
SVCR	Single-Value Corrosion Growth Rate
ULS	Ultimate Limit State
UT	Ultrasonic Technology
WT	Wall Thickness

1 Introduction

Pipelines are crucial components in extracting hydrocarbons from the seabed, in the transportation of these to shore and further transportation from production sites to markets. As a result, pipelines can be found in connection to production sites all around the world, either subsea or onshore. A problem relevant to the integrity of all structures made of metal is corrosion growth, which can lead to structural integrity problems. This is especially true for pipelines, which generally fall within the metallic-structure category, and where the transportation of fluids can have a strong influence on the behaviour and deterioration of the materials. Compared to most structures, the pressurised fluid within also adds additional consequences of failure, such as leakage damaging the environment, large financial loss due to shut-down, and in some severe cases danger to human lives. Since the consequences of failure can be large, there are strong incentives to manage the integrity of pipelines so that repair can be performed only when necessary, as such operations are also connected with high costs. Inspection methods have been developed for this reason. Although the results are imperfect and the execution expensive, they represent a way to avoid unnecessary repair work while ensuring integrity. However, due to the uncertainty of the results, methods for treating the inspection data represent an important part of the integrity assessment. These methods consist of filtering the raw data, integrity assessment on the filtered data, and future growth and capacity estimates based on the datasets.

1.1 Background

Crack growth can, in broad terms, be caused by cyclic loading or by metallic corrosion. For pipelines, the latter is the most important, since they are normally not subjected to cyclic loading in any significant degree. Pipeline corrosion is defined as the deterioration of the pipe material, caused by electrochemical or chemical reactions between the metal and its environment (Vanaei et al., 2017). The corrosion itself depends on the characteristics of the metal and the properties of the environment surrounding the component. In general, pipelines are vulnerable due to their surroundings, which can be highly corrosive, and due to the fluid transported. The external corrosion is often mitigated by coating the structure, but this is not as common for the internal conditions. The internal environment may be corrosive depending on the chemical composition of the carried fluid, which can cause integrity problems.

Corrosion is highly stochastic, and modelling of it is concerned with large degrees of uncertainty. For estimating the remaining lifetime without significant uncertainties, which calls for a high degree of conservatism, inspection data is needed. In the industry, inspection data can be obtained through a variety of methods. The method considered here is In-line Inspection (ILI) tools using the ultrasonic tool (UT), which reports remaining wall thickness (RWT) along the pipeline. An alternative to the UT is the Magnetic flux leakage (MFL) tool, which directly detects and reports defects. This alternative is not considered in this project, as the treatment of these inspection data demands a different approach. Nevertheless, all inspection data have in common that they are subjected to a significant degree of uncertainty. This means that a situational image from inspection cannot be used directly; it needs to be treated intelligently in some way to efficiently make decisions based on it.

Methods to treat inspection data are given in various design standards. DNV-GL has developed a recommended practice regarding corroded pipelines (DNV-GL, 2019a), where standard methods for assessing defects along a pipeline are defined and justified. It also provides an algorithm for treating inspection data to give a conservative overview of the defects in a pipeline. This algorithm has been applied by Aker Solutions, making a program for handling inspection data (Liu et al., 2018). The method used here will be described in detail in a following section.

In this project subsea pipelines are considered, as they are the most relevant for the Norwegian

industry, though the filtering methodology should also be valid for other systems. The subsea pipelines are generally hard to access, so replacing components or performing maintenance work is concerned with large and costly operations, making the threshold for intervention high. Having a detailed picture of the situation so that no unnecessary handling is performed is important within the industry, and inspection data is playing a vital role in creating this picture. However, the execution of inspections themselves are also costly, so creating good models reducing the need for these is appealing as well.

Estimation of the capacity of a pipeline that has corroded is another issue concerning the industry today. There are possibilities for doing simplified calculations on the momentary image provided by inspection, but the capacity may be underestimated by applying conservative values. Even larger uncertainties are evident when trying to estimate how the pipeline will behave in the future, where corrosion growth needs to be taken into account.

Structural reliability assesses these problems by introducing stochastic theory into the structural calculations, treating each uncertain variable as a random parameter with a certain distribution. The entire survival/failure-question is looked at as a sum of all the probabilities and improbabilities that can happen, i.e., as a stochastic process. The results of these calculations give the structure a certain probability of failure. It is a method that gives control over the degree of safety for a structure, making it possible to find the best compromise between safety against failure and cost/performance. These methods have been applied to the problem of pipeline corrosion in the literature, but their adoption within the industry today is still limited.

1.2 Scope of the thesis

The overall goal of this project is to review the filtering methods provided in the DNV-GL RP-F101, and on the implementation of these in the program rbpGen by Aker Solutions, looking at limitations and identifying possible fields of improvement. Improving the algorithm and performing a case study on actual pipeline data for verification is also of interest. It is necessary to perform a literature review covering the basics of the corrosion damage so that the understanding is ensured. The next step of the literature review is to look into the DNV-GL RP-F101 in detail to gain an overview of the program's working methods today. Then some more advanced methods for treatment of inspection data and calculation of both current and future capacity will be looked into to identify development possibilities and to know what to focus on in an algorithm. A particular effort has been put on reviewing probabilistic or reliability-based methods, which are expected to handle analyses concerned with high degrees of uncertainties well.

To summarise, the scope of the literature review will be:

- Description of the most common forms of corrosion damage relevant for pipelines. This includes describing the governing environmental factors and description of growth models that are meant to capture the phenomenon.
- Review filtering methods and capacity calculation in DNV-GL RP-F101 (DNV-GL, 2019a).
- Provide an overview of the basics within the usage of reliability methods on pipeline corrosion.
- Review some common growth models and some examples of usage of reliability methods on pipeline corrosion.

Part of this project is also concerned with software development using Python, based on the filtering program rbpGen by Aker Solutions. It is written in Python 2.7, for further development it will be beneficial to transfer it into Python 3. Using the filtering algorithm as basis, the program should be made to treat large amounts of data with one click. Identifying possible improvements and trying them out on actual data will be attempted.

In short, the programming part of the project will focus on:

- Transfer rbpGen from Python 2.7 to Python 3, making algorithm changes and set-up changes that are convenient for modelling.
- Make a program able to process larger amounts of data with one click.
- Identifying possible improvements compared to the old data processing.
- Create contour plots for verification compared to the old method.
- Make a method that can find defects based on a dataset of filtered wall thickness data.
- Use a reliability-based approach to assess the integrity of the pipeline considered.
- Identify and recommend possible reliability methods or model implementations that can be used for future integrity assessment.

The methodology from rbpGen and the methodology used in this project will be described, and the programs will be compared. The output from the filtering program will further be used in an integrity analysis of a real case dataset. These results will be presented, along with discussion and conclusions regarding the validity and limitations of the results.

2 Theoretical Background: Reliability methods

Uncertainties are present in all models trying to model reality. This is also evident for structural problems, which may be subjected to high degrees of uncertainty, especially regarding material qualities, fabrication tolerances, and corrosion damage. In some cases, good results may be obtained when modelling using deterministic models, but otherwise, reliability methods present themselves as better alternatives. They introduce stochastic theory into structural calculations and model important characteristics of a model as random parameters. Using such methods allows for better control of conservatism and structural performance of the structure compared to deterministic models. It is also a better approach when dealing with new designs with limited previous experience. In this project, reliability methods applied on corroding pipelines are reviewed in the literature review in section 3, and one form of reliability methods (Monte Carlo Simulation) has been used in the model built in this project. This model is described in section 4. This section will present the underlying principles behind most reliability methods and present some of the most commonly used forms. The description in this section is largely based on the project thesis written by the author (Skrede, 2019).

The reliability methods are categorised according to their level of detail in the information used from the initial stage. A common categorisation is to group them into three categories, the so-called "levels". The level definition in this project is specified as follows:

- Level I: Partial coefficient methods
- Level II: Safety index methods
- Level III: Probability of failure methods

where increasing level means an increasing level of information used in the analysis.

This section is organised according to the level description, where the most effort has been put into describing the level III methods, being the most relevant for use with computer simulations. Not all methods described have been applied or can be considered directly relevant for the remainder of this project. However, part of this project aims to develop a method for computation of remaining capacity on a corroded pipeline, partly using a reliability method. During this development, keeping in mind the various possibilities and limitations regarding reliability methods are considered important, and it will also make the discussion of the results and further development easier. So, for this reason, a more complete description than strictly necessary has been included.

2.1 Level I analysis

The group of reliability analysis methods requiring the least data and being the simplest to execute is the Level I analyses. These methods are deterministic in nature, using one characteristic value to describe each uncertain variable. Each of these variables is multiplied with a partial coefficient, contributing to a total safety margin for the system. As the level of conservatism needed varies with the consequence of failure, the partial coefficients will typically have different specified values depending on the limit state of concern. Structural codes are often using level I methods, where the coefficients are determined based on more advanced probabilistic methods. Typical examples of design limit states, for which different coefficients often are specified, are given in Table 2.1.

Table 2.1: Design limit states (Moan, 1994).

Limit State of Unfitness	Main Characteristics
SLS - Serviceability Limit States	Functional specifications, deformations without loss of equilibrium and corrosion damage
ULS - Ultimate Limit States	Loss of equilibrium, load exceeding ultimate capacity either for single members or for structure, transformation into mechanisms
PLS - Progressive Collapse Limit States (ULS for damaged structures)	Accidental loss or overloading of single members, rendering the structure in a state where progressive failure may take place.
FLS - Fatigue Limit States	Cumulated effects from cyclic loading

The characteristic design values are decided using statistical confidence for survival from more advanced analyses; typical conservative values are often specified in codes. Which percentile each parameter is decided from usually depends on the consequence of failure and the possibilities for inspection.

It is evident that these methods are heavily dependent on experience with similar structures, and based on data from reference projects. For example, the overall design of the structure must be well defined in the codes applied, and the climate variations regarding loads must be well documented. Hence, applying these methods for new problems must be done with caution, and care must be taken that the assumptions made are valid for the new case.

2.1.1 Description of limit state design methods

As the name indicates, limit state design concerns itself with the limiting capacity of a structure. The engineer must decide which states will govern the collapse of the structure, but all the relevant scenarios must be checked. With both the capacity and loads of the structure uncertain, the design condition will be an equality problem.

$$R\left(\frac{f_k}{\gamma_m}, \dots\right) \geq S(\gamma_f Q_k) \quad (2.1)$$

where f_k and Q_k are the characteristic values of material strength and loads, γ_m and γ_f are partial factors for material and loads. As this equality problem consists of uncertain variables, the problem itself is considered stochastic. It can be described with two probability distributions, $f_R(r)$ and $f_S(s)$, where R is the resistance, and S is the load. When the load effect S exceeds the resistance R, the structure experiences failure. What this failure implies depends on the limit state formulation in question. An illustration of probability distributions of the load effect and the resistance is given in Figure 2.1. This figure is only for illustration, the shapes can take any form.

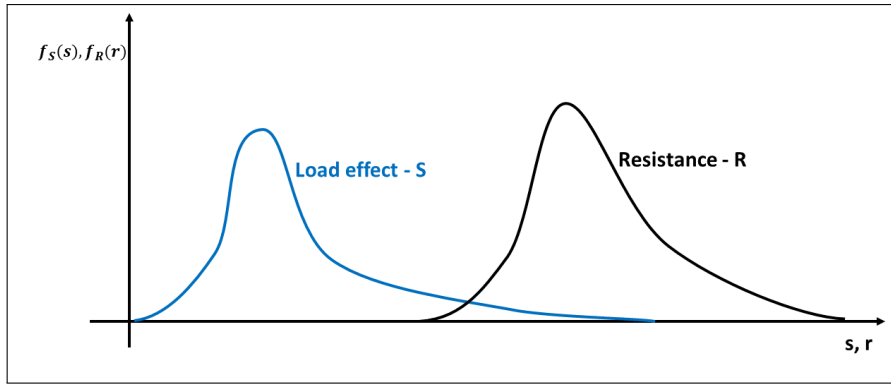


Figure 2.1: Illustration of load and resistance in level I analysis. Image inspired by Thoft-Christensen and Baker (1982).

It is important to notice that both load effect and resistance must be of the same dimension, e.g., bending moment, for this relation to be valid. It might be natural to think that the area of overlap will be the probability of failure, but this will be incorrect since the distributions are not functions of the same variable. As can be seen from the axes, the load is a function of s and the resistance a function of r . The reliability (being the complement of the probability of failure) equals:

$$\mathbb{R} = 1 - P_f = 1 - \int_{-\infty}^{\infty} F_R(x)F_S(x)dx \quad (2.2)$$

closed-form solution of this integral does not exist in general, and in most cases approximate methods must be applied.

2.1.2 Limitations with Level I methods - relationship to level II

The Level I methods are simple in nature and are highly dependent on the quality of the underlying calculations. This is often ensured when using structural codes, which are built on testing and applications of advanced calculations. Since they are having a wide range of validity, the values are usually set to be overly conservative. The use of these methods will always be a trade-off between simple engineering and a more expensive or less optimal structure. This means that an existing structure designed using overly conservative codes can have possibilities for lifetime extension, by applying more advanced methods to calculate the actual level of conservatism more accurately. This aspect is promising, especially when performing fatigue calculations on structures already in operation. Simpler methods from previous times are also often associated with limited knowledge and high safety factors, which can give additional lifetime when looking at actual fatigue damage or by using more advanced methods to estimate the damage more accurately.

Using level II methods involves applying a more probabilistic approach compared to level I. The level II methods involve mapping from the physical space to the standard normal space and then locating a design point in this new space. The design point can then be related to a certain probability of failure using an approximation. Extending this, it is also possible to obtain partial coefficients that can be used in level I methods. The failure modes governing the design will be the same for level I as for level II, but more accurate calculations may give better results regarding the capacity.

2.2 Level II analysis - Safety index methods

To explain the level II methods, some fundamentals need to be reviewed. Compared to the level I methods, these methods are more probabilistic, applying second-moment statistical information for describing each uncertain variable. The level II methods rely on using basic variables to describe the structure and using the reliability index to estimate the probability of failure. These concepts are described in the first part of this section. Further, the attention is focused on the different simplifications and idealisations applied to the problems in order to make them solvable. Finally, the relation to the level III analysis is briefly described.

The use of the level II methods generally results in an approximation of a failure probability. The error made in the approximation is associated with idealisation of the failure domain, along with coarse simplifications to the probability distributions of the parameters involved. The parameters are only described by their means and standard deviations in the level II methods, which can be a significant source of error.

2.2.1 Basic variables, Reliability Index, and the Failure Surface

The starting point of the analysis is to identify the basic variables, which are the parameters defining the problem. Examples of basic variables in a structural reliability problem can be material strength, size of imperfections, or external loads. These variables cannot be determined exactly, so they must be described as random variables each characterised by a mean and a standard deviation. The mutual correlation between the basic variables is described by a correlation matrix. With n basic variables in place, denoted as $\bar{X} = [X_1, \dots, X_n]$, a limit state function can be defined. This function can then be set up to implicitly define one safe and one unsafe region within the n -dimensional basic variable space, ω (Thoft-Christensen and Baker, 1982),

$$f(\bar{x}) = f(x_1, x_2, \dots, x_n) = \begin{cases} > 0, & \text{if } \bar{x} \in \omega_s \\ < 0, & \text{if } \bar{x} \in \omega_f \end{cases} \quad (2.3)$$

where ω_s and ω_f denotes the safe and unsafe sets, respectively. For a safe structure, the safety margin M will be equal to the function value, $M = f(\bar{X})$. As the function value f is stochastic, the safety value is naturally stochastic as well.

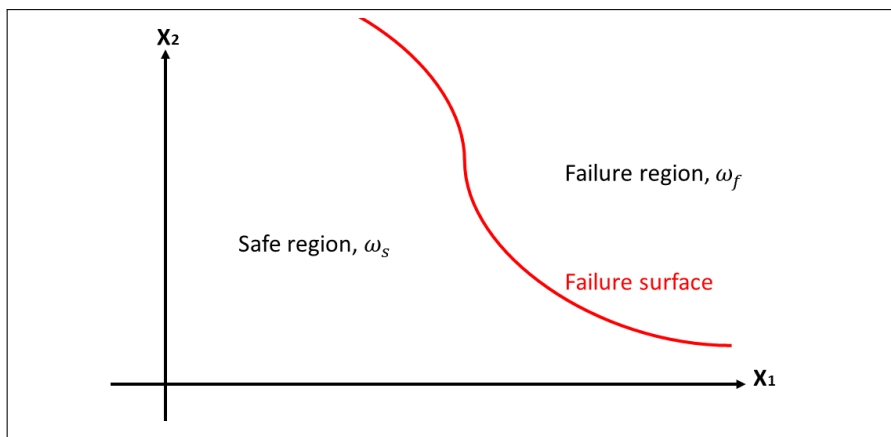


Figure 2.2: Simple 2D-case of a failure surface splitting the domain into one safe and one unsafe region.

Because the failure surface can be a nonlinear function of high order, it is normal to approximate it by a hyperplane when applying level II methods. The most common method is to apply the Hasofer and Lind safety index, described more extensively in Section 2.2.3 below. In short, the approach consists

of transforming the failure surface into the Gaussian space, and then approximating the transformed failure surface with a tangent hyperplane at the point closest to the origin. This point represents the highest probability parameter combination leading to failure, and is often called "design point" in the literature. The distance from the design point to the origin, β , is called the reliability index. The relation between β and the probability of failure, P_f , is shown in Equation 2.4 below.

$$P_f = \Phi(-\beta) \Leftrightarrow \beta = -\Phi^{-1}(P_f) \quad (2.4)$$

In cases where the failure function is linear in the basic variables, and these variables are normally distributed, Equation 2.4 gives exact results. For most cases, though, the resulting geometry is an approximation of the real surface. However, if the design point is defined correctly and the failure surface is not too curved, this method provides good results.

2.2.2 Cornell Safety Index

As mentioned, the basic variables in level II analyses are described by their mean, μ , and their standard deviation, σ . The mutual correlation is described in a correlation matrix. If the safety margin M is linear in the basic variables it can be explicitly described by the parameters μ_M as expected value and σ_M as standard deviation, calculated from the parameters defining the resistance and the load. Then, the Cornell reliability index β is defined by:

$$\beta = \frac{\mu_M}{\sigma_M} \quad (2.5)$$

which is the safety index definition introduced by Cornell in 1969 (Thoft-Christensen and Baker, 1982). From this, the probability of failure can be found using Equation 2.4.

If the safety margin is non-linear in \bar{X} , the reliability index definition in Equation 2.5 is invalid because M will not follow a normal distribution. Though approximate values for μ_M and σ_M can be found from a linearised safety margin M by a Taylor expansion about the point $X_1, \dots, X_n = \mu_1, \dots, \mu_n$. The safety margin is linearised, as shown in Equation 2.6:

$$M = f(\bar{X}) \approx f(\mu_1, \dots, \mu_n) + \sum_{i=1}^n \frac{\partial f}{\partial X_i} (X_i = \mu_i) \quad (2.6)$$

then the approximate values for μ_M and σ_M are found as:

$$\mu_M \approx f(\mu_1, \dots, \mu_n)$$

$$\sigma_M^2 \approx \begin{cases} \sum_{i=1}^n \left(\frac{\partial f}{\partial X_i}\right)^2, & \text{if uncorrelated} \\ \sum_{i=1}^n \sum_{j=1}^n \frac{\partial f}{\partial X_i} \frac{\partial f}{\partial X_j} Cov[X_i, X_j], & \text{if correlated.} \end{cases}$$

Naturally, this approximation will be heavily dependent on the location of the linearisation point μ_1, \dots, μ_n . The reliability index β will change when different points are used, leading to different reliability for the same structure. This is a problem when considering the adequacy of the result, and has led to several improved formulations for the safety index.

2.2.3 Hasofer and Lind's Reliability Index

This reliability index is an alternative to the Cornell safety index defined in Equation 2.5. Hasofer and Lind's reliability index has function invariance, meaning that it gives one result for each failure

function. This solves the most obvious problem with Cornell in a simple way. This safety index is more geometric and is described here for uncorrelated variables, to make the equations simpler.

The starting point is as in Equation 2.3, where the failure surface is defined by n basic variables, \bar{X} . This set of basic variables is normalised, which means that a linear mapping to the z -space is performed:

$$Z_i = \frac{X_i - \mu_{X_i}}{\sigma_{X_i}} \quad (2.7)$$

The failure function is also mapped to the z -space, which divides the domain into one safe and one failure region. The Hasofer and Lind reliability index (also denoted β) is simply defined as the shortest distance from the origin to the failure surface in the normalised z -space. This distance is found in the general case by iteration, and the point on the failure surface closest to the origin is denoted the design point. In the z -space, shorter distance to the origin is assumed to be correlated with higher probability density, as the space is normalised around the respective means of the basic variables. The point on the surface closest to the origin then corresponds to the most probable parameter combination leading to failure. The resulting failure probability is estimated by the transformation found in Equation 2.4, similarly as for the Cornell index.

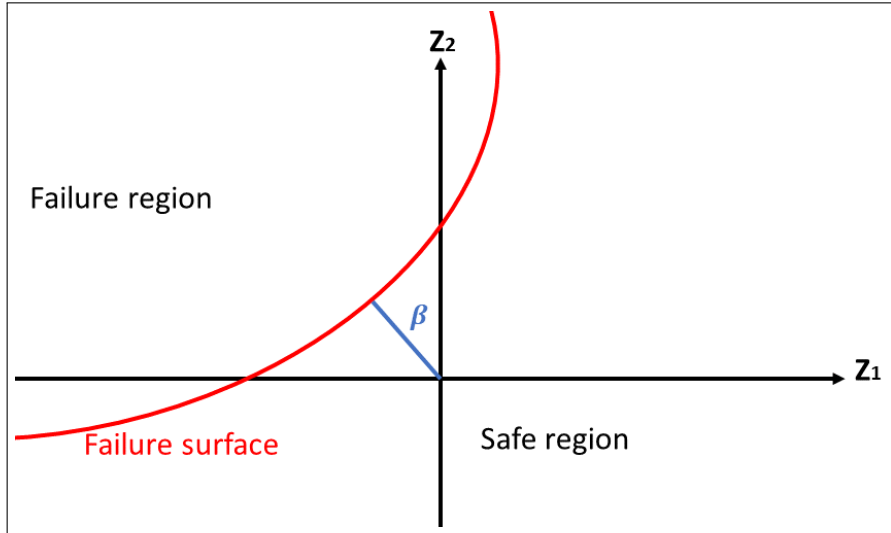


Figure 2.3: Failure function with Hasofer-Lind safety index (2D).

2.2.4 Correlated variables

So far, the formulas and methods have assumed uncorrelated variables. This makes the calculations easier to understand and to execute, but it is rarely a good representation of a real problem. The concept of correlation is simple: Let X_i and X_j be two random variables with expected value denoted by μ_i and μ_j , respectively. Then the covariance between them will equal

$$Cov[X_i, X_j] = E[(X_i - \mu_{X_i})(X_j - \mu_{X_j})]. \quad (2.8)$$

If $Cov[X_i, X_j] = 0$, the parameters are uncorrelated, in any other case they are correlated.

When treating the Hasofer-Lind safety index with correlated variables, the variables need to be transformed into uncorrelated before the normalisation can be performed. I.e., the correlated variable set \mathbf{X} must be replaced with the uncorrelated set \mathbf{Y} . This is done by defining each Y_i as $X_i + \sum_{j=1}^n cov(X_i, X_j)$, so that all covariance terms are taken care of in the parameter definition. This means that each Y_i is potentially a function of all elements in the set \mathbf{X} .

Then the normalised variables can be found as:

$$Z_i = \frac{Y_i - E[Y_i]}{\sigma_{Y_i}}, \quad (2.9)$$

where each Y_i will be a function of the basic variables X_1, \dots, X_n and their respective probability parameters. The reliability index β can then be found using Hasofer-Lind, which can be defined in short as shown in Equation 2.10 below.

$$\beta = \min_{\mathbf{Z} \in \omega_z} (\mathbf{Z}\mathbf{Z}^T)^{\frac{1}{2}} \quad (2.10)$$

where ω_z denotes the failure region in the z-space (Thoft-Christensen and Baker, 1982).

2.2.5 Relations between Level II and Level III methods

In level II analyses the basic variables making up the problems were defined only by their means and their standard deviations, in addition to the correlation matrices. Idealisations on the failure surfaces were also applied to make the problems manageable. The underlying distributions defining the basic variables are either not known or simplified to fit into the methods. These approaches were used traditionally, partly because of the need for hand calculations and limited computational capacities. In level III analysis, the full (joint) probability distributions of the basic variables are to be used. The methods utilises some of the same relations as level II, but the amount of input data is larger and more accurately described for a level III analysis. Correspondingly, the results gain more confidence when originating from a level III analysis compared to the lower levels.

2.3 Level III analysis

The level III methods use the joint probability distributions of all the basic variables, which places a significant demand on the amount of information necessary. Also, since the amounts of data causes intricate relations, these calculations are virtually impossible to do by hand calculations. These methods include numerical integration, approximate analytical methods, and simulation methods. The results from the latter are converging towards the "true" reliability as far as the probability models can determine. However, the rate of convergence differs and can be very slow for low probabilities or computationally costly models. The analytical methods usually represent more approximate models representing the problem.

With significant advances in computational power during the last years, new possibilities both regarding simulation and estimation of complex models have presented themselves. In earlier years, Monte Carlo simulations were advised against, considered too time-consuming even to be an alternative (Thoft-Christensen and Baker, 1982). Today, they are widely used within reliability analysis, even with complex FEM-models to calculate structural integrity. The limitations previously apparent are no longer as relevant, and more sophisticated analyses can be performed applying modern algorithms. This section will review some standard level III approaches for computing reliability, considering the approximate analytical first- and second-order reliability methods and some simulation methods. Numerical integration in the common form is not treated in this project due to its limited use in reliability analysis.

When considering level III methods, the starting point is commonly referred to as the fundamental problem. This problem describes the survivability of a system and relates this to a probability of failure. The failure function is denoted $G(\bar{x})$, and can consist of multiple failure modes if present. \bar{x} is the n-dimensional vector of basic variables. The failure region in the n-dimensional space is where G is less than zero, and the total contents of this region will equal the probability of failure. This means that an integral over this region must be performed to find the reliability. For most multidimensional problems of higher order, this integral is impossible to calculate using purely analytical methods, so a resort to more approximate methods is needed. The methods described in this section are approximations of this integral.

2.3.1 FORM and SORM

The first and second-order reliability methods (FORM and SORM) are described in this section, representing the most common forms of semi-analytical methods for estimating the reliability within level III analyses. Usually, these are considered good alternatives to simulation methods even though they rely on simplifications of the failure domain, making the results approximate. Verification from simulation methods is, in general, recommended when using these approaches. The logical algorithms behind these two methods are similar; the difference lies in the hyperplane approximating the failure domain.

1. As a starting point, a limit state function $G(\bar{x})$ is formulated as a function of N basic variables. Consider $\bar{x} = x_1, \dots, x_N$ as the set of basic physical variables making up the problem (e.g., loads or material strength) described by joint probability distributions.
2. The FORM and SORM methods rely on transforming the limit state function into the Gaussian space, by use of e.g. Rosenblatt (analytical) or Nataf (approximate) transformations. These transformations are extensively described in other papers, including Melchers (1999) which this description is based on. Here they are not treated in detail, but the Rosenblatt transformation is used for illustration. Each conditional cumulative distribution $F_i(x_i|x_{i-1}\dots)$ is set to the standard Gaussian value of the new variable set u_i , or

$$\begin{aligned}\Phi(u_1) &= F_1(x_1) \\ \Phi(u_2) &= F_2(x_2|x_1) \\ &\dots\end{aligned}\tag{2.11}$$

where $\bar{u} = u_1, u_2, \dots, u_n$ are the coordinates in the standard Gaussian space. The limit state function can then be expressed from these coordinates, effectively transforming the limit surface to the standard Gaussian space. Mathematically, the transformation is: $G(\bar{x}) = g(\bar{u})|\mathbf{J}|$, where \mathbf{J} denotes the Jacobian.

3. Having an expression for the failure surface in the u-space, an optimisation algorithm is needed to find the point closest to the origin. This point represents the *design point*, which is the point on the failure surface with the highest probability of occurring. With a multi-dimensional problem, multiple design points might exist. With one clear design point, most of the failure probability will be dominated by this contribution. If multiple design points exist, the approximated failure surface must take this into account.
4. After the design point is found, the methods have different approaches. FORM approximates the failure surface by a first-order function in the u-space, naming the distance from the origin to this point as the safety index, β_F . The safe domain is approximated as a tangential first-order hyperplane with the smallest distance to the origin equal to β_F . The probability of failure is estimated as $P_f = \Phi^{-1}(-\beta_F)$. Then the approximated limit state surface can be transformed back to the physical variables, and the design parameters are found. This method produces good results if the problem contains relatively few parameters and has a low curvature failure surface, but is insufficient for problems with a highly curved failure surface or with multiple failure points.

In SORM, we utilise the same design point, but instead of approximating using a linear function, the limit state function is approximated by a higher-order surface. The domain outside this surface represents the probability of failure. The most common approach in SORM is to approximate the actual surface by parabolic or quadratic functions. Both the design point and the shape of the real failure surface should be included in the approximated surface, but the best approach will vary with each problem. One alternative is to approximate a parabola from the curvature of the actual failure surface along with the design point. Another is to point-fit a

parabola to selected points on the failure surface along with the design point (Der Kiureghian et al., 1987).

Approximating the content enclosed by a higher-order surface is no easy task, but there are some approximate solutions applicable to such problems. One approach is to utilise sampling around the design point to obtain the contents of the failure surface (Melchers, 1999). Another is to determine the limit state curvatures and apply asymptotic equations for estimating the contents (Breitung Karl, 1984). A third method is to approximate the curvatures iteratively using the gradients of the limit-state function (Der Kiureghian et al., 1987).

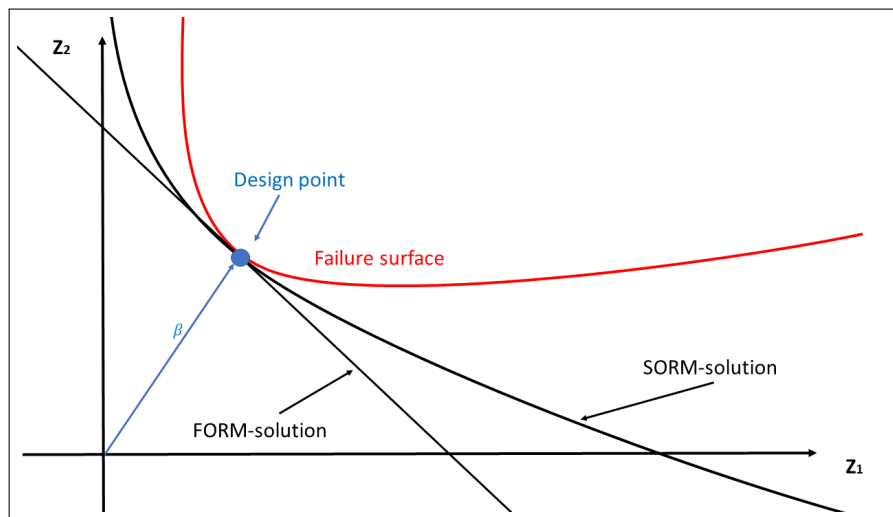


Figure 2.4: 2D-illustration of a failure surface, the FORM-approximation and the 2nd order SORM approximation.

The analytical methods presented here are expected to produce good results for sufficiently simple failure surfaces. With SORM, estimating the contents of the failure surface might present itself as a problem, but it can be worked around by applying various approximations. An approach combining FORM/SORM with simulational methods is to use the result of the FORM/SORM analysis as a starting point for a more extensive simulation e.g., by Monte Carlo simulation. This approach will make the simulation faster and more reliable, reducing the number of cycles needed to obtain accurate results.

2.3.2 Monte Carlo Simulations: General overview

Monte Carlo simulation (MCS) techniques are widely used for estimation of failure probabilities, either as the main analysis method or as a complimentary tool. The basic idea behind the simulation is quite simple, it mainly consists of conducting a large number of "experiments" and estimating the probability of failure as (Melchers, 1999):

$$p_f \approx \frac{n(G(\hat{\mathbf{x}}_i) \leq 0)}{N} \quad (2.12)$$

where $G(\hat{\mathbf{x}}_i)$ is the limit state function checked at each iteration, n denotes the number of failures experienced, and N is the number of tests conducted. The accuracy of this analysis will depend on the number of cycles, the variance of the individual distributions, and the failure probability. The crude Monte Carlo approach is the simplest of the methods, where the set is sampled uniformly over the entire domain. The sample of x_1, \dots, x_n is generated from the respective conditional distributions, and the problem can be checked for failure using the limit state function.

A common problem with Monte Carlo simulations is computational efficiency. DNV-GL suggests, as a rule of thumb, $\frac{100}{p_f}$ number of simulations for crude Monte Carlo simulation (Ronold, 2016), which quickly leads to infeasible computational time for low probabilities or complex problems. With low probabilities of failure, it is logical that a large number of simulation is needed even to observe failure at all, and for effective and accurate estimates of the failure domain, the number increases still. The number of simulations required also depends on the variance of the individual components. There are, however, several ways to improve the efficiency of a Monte Carlo simulation. The most common method is to locate areas of interest and focus the analysis around a specific point. Methods exemplified in this project are importance sampling (2.3.3) and directional simulation (2.3.4).

2.3.3 Importance Sampling

The crude Monte Carlo simulation distributes the random samples uniformly over the entire domain, aiming to estimate the size and location of the failure domain. However, since the failure domain is often located in the tails of the distributions making only a small part of the entire domain interesting, much of the computational effort is primarily wasted on simulating uninteresting safe areas. Importance sampling techniques represent methods to focus the simulations executed to the area of interest, making the analysis more efficient. This is done by sampling from a distribution weighted around the critical region. The calculations are then corrected to take into account that they are sampled from this distribution. A brief outline of the basics of importance sampling follows in this section.

The fundamental integration problem solved when using Monte Carlo simulation for estimation of structural reliability is:

$$p_f = J = \int \dots \int I[G(\mathbf{x}) \leq 0] f_x(\mathbf{x}) d\mathbf{x}, \quad (2.13)$$

where crude Monte Carlo simulation estimates J as the sum of observations less than zero over the total number of observations. $I[\]$ is the indicator function equaling 1 when the condition is fulfilled and 0 otherwise.

The integral presented in Equation 2.13 can be rewritten, so that an importance-sampling density function is included. By performing a variable change from \mathbf{x} to the variable \mathbf{v} sampled from the importance density distribution, we end up with:

$$J = \int \dots \int I[G(\mathbf{x}) \leq 0] \frac{f_x(\mathbf{x})}{h_v(\mathbf{x})} h_v(\mathbf{x}) d\mathbf{x} = \mathbb{E} \left(\frac{I(G(\mathbf{v}) \leq 0) f_x(\mathbf{v})}{h_v(\mathbf{v})} \right), \quad (2.14)$$

where \mathbf{v} is sampled by use of the importance probability distribution h_v , \mathbb{E} denotes the expected value of \mathbf{v} . By applying this transformation, h_v effectively governs the sampling, and the variance, covariance, and correlation of the expression $(\frac{If}{h})$ can be calculated.

Since h_v governs the sampling, the choice of this distribution is important. In particular, the variance of $\frac{If}{h}$ has a large impact on the results. This is calculated as:

$$Var\left(\frac{If}{h}\right) = \left(\int \dots \int I[\dots]f_x(\mathbf{v})d\mathbf{v}\right)^2 - J^2 \quad (2.15)$$

This expression needs the value of J , which is what we want to calculate with the Monte Carlo simulation in the first place (ref. Equation 2.13). However, good results can be achieved by applying approximate values of J , meaning that such an expression can form a basis for an optimisation algorithm.

The limitations in applying importance sampling are that important regions may be neglected. The method can, for example, yield bad results with multiple design points if h_v is poorly chosen. Convergence can be achieved to a wrong design point even by using optimisation, meaning that analyses must be performed with a solid amount of scepticism. Often, the best practice is simply to apply a priori knowledge to perform an educated guess about the area of interest, and model the importance probability density function (pdf) from that. Though, with high-dimensionality problems this is not always possible, and care must be taken when modelling the analysis.

2.3.4 Directional simulation

The starting point in such an analysis is, as with the other methods, a limit state surface described by n random variables (\mathbf{X}). A common approach is to transform the problem into standard normal space, e.g., by applying a Rosenblatt transformation (described in Equation 2.11). We then have the standard normal random vector $\mathbf{U} = [U_1, \dots, U_n]$, in terms of which the limit state surface can be expressed.

The aim is to integrate over the failure domain in the standard normal space; this is done with the following scheme:

1. Find a number of points on the failure surface, effectively describing the problem. These are located in a corresponding set of directions.
2. Perform the reliability analysis by estimating the failure domain by a set of one-dimensional integrals in the directions specified.

To achieve the first point above, a directional vector \mathbf{A} is defined, which describes the various directions the integration is going to take place over. An integral in the standard normal space is described exactly by use of the χ_n^2 -distribution, meaning that for each direction in A we can obtain a value of the integral in this direction, from the origin to the failure surface. The individual probability of failure, p_j , will be the complement of this value. By applying the tabulated value for the integral, the individual contributions will equal:

$$p_j = 1 - \chi_n^2(r) \quad (2.16)$$

where r is the distance from the origin to the corresponding point on the failure surface. By applying this method, each point estimates the failure surface by an individual hypersphere. This means that if the failure surface is spherical, only one point is needed to find the failure probability exactly.

Having a set of N points evenly distributed over the failure surface, each estimating the failure probability by Equation 2.16, we can estimate the total failure probability as the mean of all the contributions:

$$p_f \approx E(\hat{p}_f) = \frac{1}{N} \sum_{j=1}^N \hat{p}_j \quad (2.17)$$

The simulation method described here is the most basic method, having a resemblance to the crude Monte Carlo simulation. Applying uniformly distributed directions is not the most efficient method for most problems, and cannot be expected to yield good results when dealing with more complex limit states. For simpler problems the approach should yield good results.

The directions are determining which parameter combinations are checked. Similarly as when using MCS, large parts of the domain may be unreasonable to check. For controlling which parts of the domain that should be checked, importance sampling can be applied in this method as well. A simple scheme is, e.g., described by Melchers (1999), where an equal approach to the one described for MCS in 2.3.3 is used. The problems are still evident for complex, nonlinear failure surfaces with multiple extreme points, which this method has problems with analysing.

Katsuki and Frangpol (1994) present a more advanced approach where the entire domain is divided into active and passive subdomains, and the probability of failure is calculated as the sum of contributions from each subdomain. Inside each subdomain, the limit surface is estimated as a series of hyperspherical segments, with radius and p_f calculated as in the basic case. This technique takes multiple design points into account, along with highly nonlinear failure surfaces. However, the errors are greater when considering more linear failure surfaces. This method is called the *Hyperspace division method* (HDM).

Improvements can also be made at identifying the directions over which the domain is to be analysed. This can have a large influence in determining the efficiency and accuracy of the analysis. For example, attempts were made by Nie and Ellingwood (2000), where the HDM approach is applied along with certain optimisation algorithms to identify ideal directions. These directions can be reused for structural problems of the same dimension.

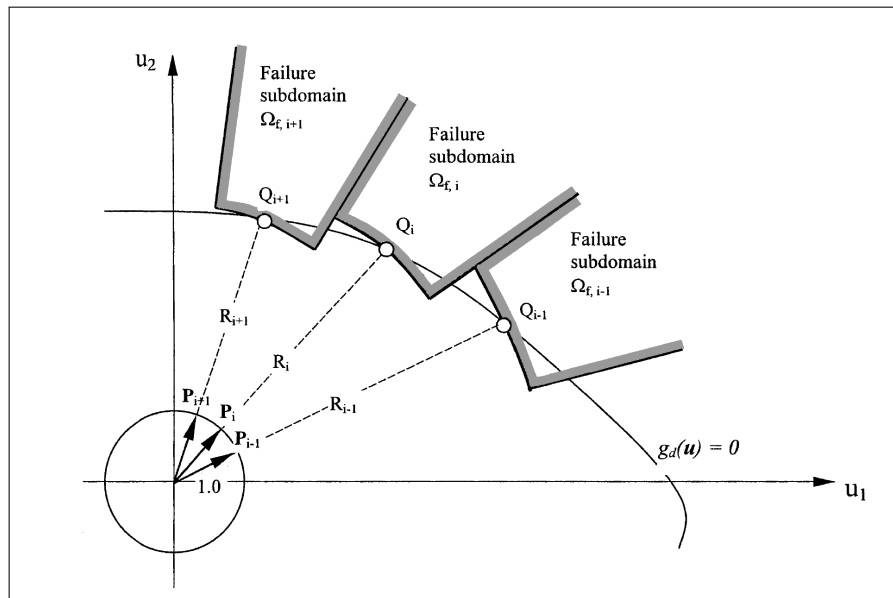


Figure 2.5: 2D-illustration of a failure surface and the approximation using spherical segments. Image taken from Nie and Ellingwood (2000).

3 Literature review: Corrosion of pipelines

As the thesis is regarding pipeline corrosion, it is necessary to review some of the available literature to gain a better overview of the important characteristics and mechanical principles regarding pipeline corrosion. The first part of this section will concern itself about the problem of corrosion damage and describe some of the factors governing the phenomenon. Then, a short review of the methods recommended in DNV-GL RP F101 is included (DNV-GL, 2019a), to highlight parts of a practical industry-standard method for calculating fatigue damage in use today. The last part of this section will review some methods applied in the literature for constructing reliable models for the prediction of corrosion damage and for estimating remaining capacity. As this project partly focuses on reliability methods for estimating corrosion damage, the literature examples reviewed are applying different forms of reliability methods as well, as the working methods of these models applied to pipelines are of interest.

3.1 General description of Corrosion damage

As mentioned, metallic corrosion is caused by chemical or electrochemical interaction between the metal and the environment. When considering subsea pipelines, the environment includes both the fluid transported, which can cause corrosion on the inside, and the surrounding sea water, leading to corrosion on the outside of the pipeline. Pipelines are usually covered in some coating, giving protection against corrosion from the outside. This means that external corrosion damage usually coincides with coating damage. The internal conditions are governed by the characteristics of the fluid and the local variations within the pipeline. The quality of the metal may, for example, vary significantly within a pipeline, and the influence of welds will also be important in deciding the corrosion characteristics.

Corrosion damage can have significant consequences for a pipeline system, as their primary purpose is to contain and transport fluid safely. As such, a leak comes in direct conflict with the primary objective of the structure. Additionally, depending on the size of the defect combined with the internal loading, corrosion may lead to structural instability problems, compromising the safety of parts of the pipeline. Onshore pipelines that fail may cause loss of human life, especially if the transported fluid is flammable or explosive. This is highly relevant when dealing with certain hydrocarbons, for example. Human lives are usually not directly at stake when considering the majority of the lengths of subsea pipelines. Failure of pipelines in these conditions can instead lead to severe damage to the environment, especially to the plants and creatures living in the surroundings. However, human lives need to be taken into account when risers or pipelines close to platforms or shore are considered. Leaks are also, regardless of the surroundings, almost always concerned with significant cleanup and repair costs in addition to reputational damage. This means that companies have large incentives to minimise the occurrence of such events. On the other hand, it is also expensive to over-dimension structures, so care must be taken to ensure a sufficient, but not a too high level of conservatism.

3.1.1 Common forms of corrosion damage

In general, there exist several types of corrosion damage. It is common to categorise different types of corrosion damage according to some characteristics, as the different forms vary in severity and potential consequence. The mechanics that decide which form of corrosion a certain pipeline is going to experience are not fully understood, and resort is often taken to applying statistical models. Inspection can reveal the momentary image, at least to a certain error margin, but further growth is not possible to predict exactly by any known method today.

As there are various applications of corrosion methods, the ways of classification of corrosion damage differ as well. When looking at the mechanics behind the corrosion, it is clear that somewhat

advanced categorisation is needed. Vanaei et al. (2017) divide corrosion damage into:

- Pitting corrosion

The most common form of corrosion damage in oil and gas pipelines. It is defined as a severe, localised deterioration of a surface area, leading to a pit formation on the pipe wall. It can appear both on the internal and external pipe wall and is highly stochastic in nature. May occur because of (but not limited to):

- Pipe material defects or surface defects
- Mechanical damage to the protective layer
- Localised chemical damage due to aggressive substances, such as chlorides
- Improper materials selection

- Cavitation

Occurring on the internal surface of the pipeline, where the working pressure of the fluid drops below its vapour pressure. This depends on the operating conditions, component layout, pump sections, and gross geometry changes.

- Erosion corrosion

Caused by the relative movement between the fluid and/or particles in the flow against the pipe wall. Fluid turbulence can especially cause high erosion rates, which means that manufacturing errors such as poor finishing can cause major corrosion damage in the following lifetime of the pipeline.

- Stray current corrosion

Caused by localised electrical current from neighboring sources. Not relevant for subsea pipelines.

- Microbiologically-influenced corrosion (MIC)

MIC is defined as the deterioration of metal as a direct or indirect result of the metabolic activity of microorganisms.

Such a division is fitting for design approaches, and especially when considering the cause of corrosion. In this report, the structural integrity of an already installed pipeline is to be investigated, and a more coarse division for corrosion damage is needed. As actual data is analysed, the underlying reasons are hard to combine into an analysis. Though they would probably work well for determining e.g., crack growth rate, they are hard to obtain from a set of inspection data alone. A more common approach is to divide the observed defects into two groups, which will have a different effect on the capacity:

- Pitting corrosion

Localised deterioration of the pipe wall material. This is usually the most severe when considering capacity.

- Uniform/Channeling corrosion

Can be a continuous chain of pits grown together, or caused by errors in the manufacturing process leading to localised lower wall thickness over a span of length.

A combination of these two can also be observed, where there are localised pit formations within a longer span of uniform corrosion damage. The limit on pit depth will be lower in such a case, with less pipe wall thickness to grow through.

3.1.2 Parameters deciding the corrosion

The process governing the growth of corrosion defects contains inherent uncertainties and cannot be modelled exactly using available methods. Limited knowledge of the system in question can also further increase the uncertainty of the system. Multiple inspection runs can highlight some important characteristics of the system and can be applied to estimate future corrosion damage. Nevertheless, having a clear understanding of the underlying factors causing the corrosion will also be of benefit. At present, the effect on corrosion from individual parameters are known, but knowledge about their interaction is lacking. This means that corrosion growth calculations will be approximations, regardless of the prior information available.

External corrosion is a problem that occurs on all metallic marine structures such as ships and offshore structures. Subsea pipelines also experience internal corrosion, which occurs due to interaction between the transported fluid and the metallic walls of the pipeline. Internal corrosion is especially frequent around girth welds due to discontinuities, but can in general occur at all points along the pipeline.

According to Paul (2012), the important environmental factors in a marine environment include:

Salinity: The chloride (Cl^-) concentration of sea water can vary from about 5.8 g/kg to about 24 g/kg. It is dependent on the pipeline location, sea currents, weather, etc. Chloride ions have a strong influence on corrosion of metals, as they dissolve the passive layer, and most metallic structures submerged in sea water will need some coating and/or cathodic protection.

Sulphate level: The sulphate (SO_4^{2-}) concentration varies from 0.8 g/kg to 3.4 g/kg. It can both increase or decrease the corrosion rate based on other factors.

Dissolved oxygen level: Higher concentration of dissolved oxygen enhances the cathodic reaction of water reduction, and hence promotes corrosion of the metal.

pH: pH in sea water is usually neutral or slightly alkaline. But a corroding pipeline can experience localised acidity due to corrosion effects and the presence of hydrocarbons from crude petroleum products. The effect on corrosion is dependent on the pH: acidic environments enhance corrosion while a pH between 8.5 and 12 enhances the formation of a passive layer. A pH above 12 strongly promotes corrosion.

Temperature: Higher temperature enhances the dissolution of ions to the surroundings and helps in convection mass transfer. These processes promote depolarisation, which increases the corrosion rate.

A newly installed subsea pipeline usually has its coating intact, which eliminates most of the external environmental factors. When the metal comes in contact with the sea water, corrosion can happen. External corrosion tends to be local, covering an irregular area of the pipe and forming a long groove pattern after the protective coating fails. Prevention of external corrosion is possible, and by carefully selecting materials, having high manufacturing quality and sufficient coating and cathodic protection, this problem may be greatly reduced.

The internal corrosion rate is depending on both the characteristics of the transported fluid and the pipeline. The characteristics of the transported fluid may change as a pipeline is installed for service over a long period of time. For example, as an oil well ages, the fluid usually contains increasing amounts of water and gas flow. The presence of highly corrosive substances such as CO_2 , H_2S , calcium, and chlorine compounds may greatly increase the corrosion rate, mainly due to erosion-corrosion (Hernández-Rodríguez et al., 2007). The internal corrosion is also dependent on the layout of the pipeline. Elbow joints, sharp edges, and discontinuities are especially vulnerable to corrosion. This also applies to girth welds, which can be vulnerable in case of bad finishing. The choice of material is also important, as different steels have different resistance to certain chemical substances.

When monitoring a corroding pipeline, keeping these effects in mind is important even though most of these effects can only be mitigated against in the design stage. They may, e.g., cause changes in corrosion rate, which again may have an influence on inspection intervals and the lifetime of the structure.

3.1.3 Growth models

Corrosion growth models are important in establishing inspection intervals, estimating the remaining lifetime and reliability of a pipeline, and also in decision-making regarding the repair and/or replacement of pipeline components. The deterministic growth models can take a variety of forms, but the most common consists of modelling growth as either a linear or a non-linear process where the corrosion growth is assumed to follow a set pattern. Some deterministic growth models are described in 3.4.2. The probabilistic methods for estimation of corrosion growth are generally more advanced and give more options regarding conservatism and accuracy estimates. Examples of these models can be found in Section 3.4.3.

The corrosion growth models can either be based on inspection data from multiple inspection runs, on a prior assumption that the corrosion growth will follow a certain growth rate or be accurately modelled by a statistical process. It is also possible to combine these methods, e.g., with an updated growth model. Applying the past inspection data may seem like an appealing alternative, as it should intuitively predict the behaviour of a certain pipeline accurately. However, past data is only an accurate representation of the past, and future extrapolation will be subjected to significant uncertainties (Vanaei et al., 2017).

Another limitation on growth models based on inspection data is that they rarely take the growth of new defects into account. Qin et al. (2015) developed a method that takes both growth of existing defects and the generation of new defects into account, applying the non-homogeneous Poisson-process for modelling of the generation of new defects. The growth of depths of individual defects is modelled using the non-homogeneous gamma-process. This may be a more accurate description of reality, and they conclude that the generation of new defects has a non-negligible effect on the capacity. However, this does not discard all existing growth models; it states that they could be more accurately formulated. And what impact this non-negligible effect has on the different analyses is not known.

3.2 Review of DNV-GL RP F101: Standard approach

DNV-GL has developed a recommended practice for assessing pipelines with active corrosion defects (DNV-GL, 2019a). It is developed for two different cases, covering most subsea pipelines;

- Internal pressure loading only.
- Internal pressure loading combined with longitudinal compressive stresses.

The recommended practice (RP) adopts two different approaches for estimating the reliability of a pipeline. The document is split between these two approaches:

Calibrated safety factors approach: Approach using calibrated safety factors, which takes natural spread in material properties, wall thickness, and internal pressure variations into account.

Allowable Stress Design (ASD) approach: Approach based on allowable stress design (ASD). The capacity of the pipeline is calculated and is scaled down using a single usage factor. Uncertainties are left up to the judgement of the user.

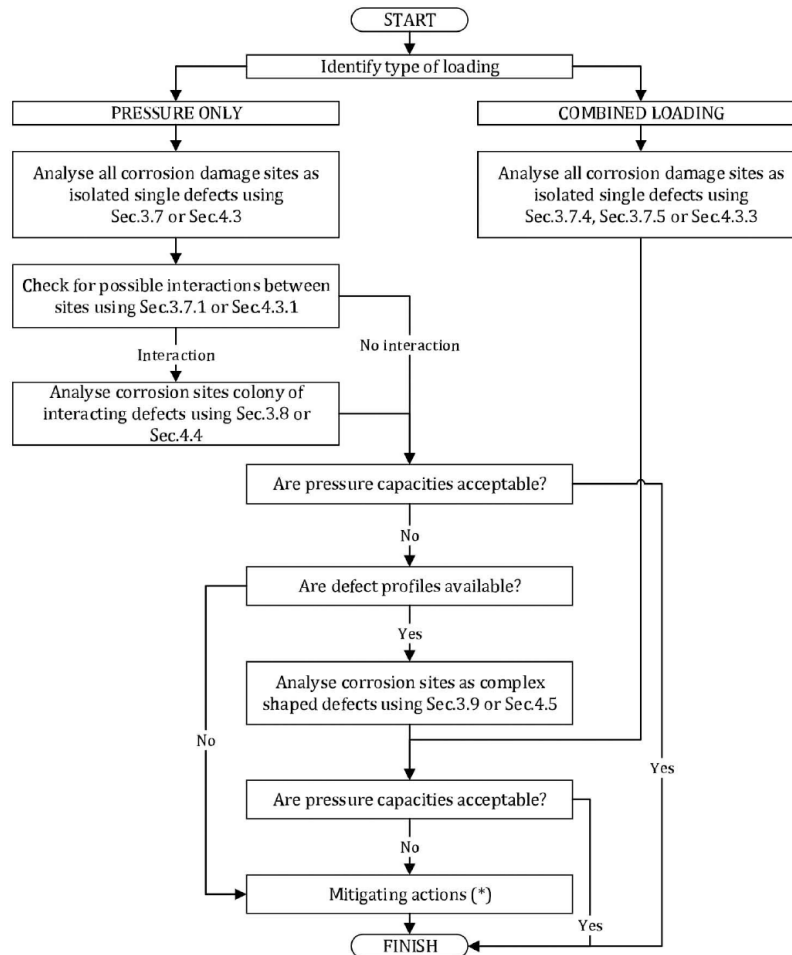


Figure 3.1: Flowchart over the assessment procedure in the recommended practice, taken from DNV-GL (2019a).

In addition to the capacity assessment approaches, the RP contains five appendices. In this project, appendix D: *Assessment of long axial internal corrosion defects* is of interest. This section presents an algorithm for assessing inspection data from a pipeline subjected to long axial internal corrosion. This algorithm has been adopted by Aker Solutions (Liu et al., 2018) in the development of computational

tools in treating inspection data. The algorithm for data filtering is looked at in more detail in Section 4, while the capacity evaluation from DNV-GL is reviewed below.

3.2.1 Brief review of the calibrated safety factors approach used in DNVGL-RP-F101

This section utilises calibrated safety factors in combination with probabilistic calibrated equations based on the Load and Resistance factor design (LRFD) methodology. The calculations are deterministic in that the input parameters are set to fixed values, though the calibration itself is probabilistic. Lines can be drawn to the Level I methods discussed in Section 2.1, where underlying calibrated coefficients are used to estimate the resistance of a structure.

The safety factors are calibrated with particular care taken to consider uncertainties in defect depth and material properties. They are also dependent on the required safety level of the structure. The reliability levels are as shown in Table 3.1.

Table 3.1: Safety class and target annual failure probability for ultimate limit state (ULS) (DNV-GL, 2019a).

Safety Class	Target annual failure probability
Very high	$< 10^{-6}$
High	$< 10^{-5}$
Medium	$< 10^{-4}$
Low	$< 10^{-3}$

Subsea oil and gas pipelines without frequent human activity are normally classified as safety class medium. The parts of pipelines close to shore or near frequent human activities are set to safety class very high. More detailed information and guidelines regarding safety classification are found in the relevant standard, e.g. in DNVGL-ST-F101 (DNV-GL, 2019b) for submarine pipeline systems.

The LRFD methodology used to determine the capacity equations is a well-known scheme of designing steel structures and structural components. It differs from the ASD approach in the way the safety factors are added, as shown in the inequalities in the following (Galambos, 1981);

$$R_n/F.S. \geq \sum_1^i Q_{ni} \quad (3.1)$$

$$\phi R_n \geq \sum_1^i \gamma_i Q_{ni} \quad (3.2)$$

where Equation 3.1 shows the ASD-approach and Equation 3.2 shows the LRFD-approach. R denotes resistance, Q denotes external load, and the remaining parameters are safety factors. As can be seen, the difference between the approaches is that the ASD-approach uses one safety factor for the entire inequality while the LRFD takes one factor per load factor as well, to account for uncertainties in the load. More detailed description of the probalistic method behind the LRFD-approach is described by Galambos (1981).

DNV-GL utilises the approach given in Equation 3.2, and defines the following parameters:

- γ_m = Partial safety factor for model prediction.
- γ_d = Partial safety factor for corrosion depth.
- ϵ_d = factor for defining a fractile value for the corrosion depth.
- $StD[d/t]$ = standard deviation of the measured (d/t) ratio (tool-specific).

and these are again based on the safety class (e.g., from Table 3.1), inspection method and inspection accuracy. Typical values are given in the RP.

So, with the partial factors decided from the safety levels and from the inspection tool in question, the defects are ready to be assessed. This is done by checking for failure modes using relevant equations for each defect. The equations are specific with regard to loading and defect configuration, and presenting them all here makes little sense for this project. For more detailed study, checking the RP (DNV-GL, 2019a) is advised.

A step-by-step guide from the RP is provided below. It illustrates an assessment procedure for estimating the pressure resistance of a corroded pipeline:

- Step 1 Calculate the pressure resistance for each defect or corroded section as a single defect. p_{corr} will be the pressure resistance for the most severe defect.
- Step 2 Present the pressure resistance for all the selected defects/sections in a histogram with pressure on the horizontal axis and the number of defects/sections on the vertical axis.
- Step 3 Define a pressure limit, $p_{lim} = 1.20p_{corr}$. The defects/sections below p_{lim} are the defects/sections that will contribute to the system effect.
- Step 4 Count the number of defects with capacity less than p_{lim} and find the pressure resistance reduction factor from the $N - \gamma_s$ curve. This curve is based on probabilistic modelling of the limit state and the pressure resistance factor, γ_s . It has been derived by the use of SORM, which is briefly described in Section 2.3.1.
- Step 5 Calculate the pressure resistance, including system effect:

$$p_{corr,syst} = \gamma_s \cdot p_{corr}$$

- Step 6 If the pressure resistance, including system effect, is unacceptable or considered too conservative, a more detailed calculation for determination of pressure resistance, including system effect, can be carried out according to Appendix D reviewed in Section 3.3 below.

This guide is a more simplified version of the one given in appendix D, and does not require detailed inspection data.

Every single defect must be checked for interaction effects, which may lower the pipeline capacity. They must also be checked for effects coming from irregular shape (more on this in Section 3.2.2). This makes it a tedious operation to check each section of the pipeline, as every defect must be evaluated considering multiple effects. The lowest pressure resistance found is set as the system pressure resistance.

3.2.2 Brief review of the Allowable stress design approach

The approach based on ASD calculates a failure pressure of the pipeline, which is multiplied with a safety factor based on the original design factor. The formula used in practice is shown in Equation 3.1. It differs from the calibrated safety factors approach in applying a single safety factor, which has both its benefits and its drawbacks. The LRFD has advantages regarding the consistency in which conservatism is added, where each load effect is made conservative before entering the equations. The ASD calculates the load effects as accurately as possible and adds conservatism to the final result only. The approach may be more flexible when using more advanced calculations, but the single coefficient is multiplied with all contributions. Using the LRFD may make the total level of conservatism lower, since the coefficients may be individually adjusted. However, which method provides the best results will vary with the case and the chosen analysis method.

Following the ASD-procedure in the RP, the failure pressure is calculated for each defect, which has to be controlled for:

1. Assessment of a single defect with no interaction.
2. Assessment of interacting defects, dependent on the size and position of each defect, along with the spacing between them.
3. Assessment of a complex-shaped defect. This is to determine if the defect acts as a single irregular patch or if local pits within the patch dominate.

The classification of each defect depends on both the loading condition, on the size and position of the defects, and the internal spacing between them. The rules give classification guidelines for this assessment. Calculating the total resistance following these criteria gives a safe working pressure, which is then scaled using the single safety factor, as previously mentioned.

3.3 Assessment of long axial internal corrosion defects according to DNV-GL RP F101

This part of the RP represents a more complete guidance on the treatment of inspection data than the standard methods. In the RP it is placed in Appendix D, as it is a method that is harder to implement in practice than the others. If the previously mentioned approaches give infeasible results, this approach gives a more detailed picture that might verify the structure for operation. It is more detailed and has a lower predicted level of conservatism.

The proposed methodology addresses:

- An evaluation of the inspection results, and further establishment of a two-dimensional representation of the internal corrosion along the pipeline, the so-called River bottom profile (RBP).
- Calculation of pipeline capacity, i.e., integrity assessment controlling the pressure resistance accounting for the system effect.
- Assessment of internal corrosion development.

In using this approach, it is important to have data of reasonable quality, and have sufficient amounts of data.

The deliveries from ultrasonic technology (UT) inspection are divided into levels. Level 0 denotes the standard delivery; the higher levels are extensions to this standard delivery. The levels 1-4 are successive, i.e., level 2 requires delivery of level 1, and so on. An overview of these levels is found in Figure 3.2 below.

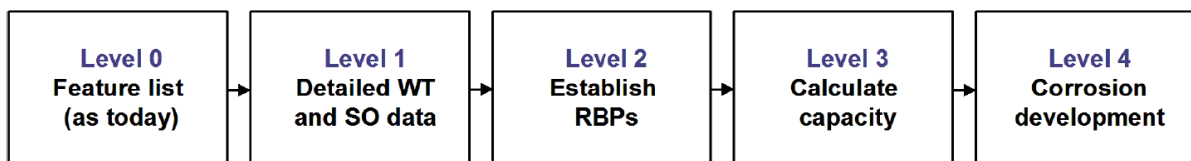


Figure 3.2: Overview of the levels of deliveries from UT inspection. Image taken from DNV-GL (2019a).

The methodology is split according to these levels of delivery. The RP Specifies the expected delivery from all the levels, and levels 1-4 also have corresponding algorithms for calculations.

3.3.1 Level 0: Feature list

The results from this level are the delivery from the inspection companies and represent more of a specification listing. The term feature list means listing the defects with geometry and relevant data. The RP gives what an inspection report from a UT inspection run should include as a minimum. This level of delivery must include:

- Echo loss reporting to a certain detail. Certain plots are specified.
- Probability of detection and tool limitations.
- The accuracy of length, width, and depth measurements.
- Measured wall thickness of uncorroded areas, preferably compared to previous inspection results.
- Threshold values for depth, length, and width measurements.

- A figure to show how the features are reported (example specified).

This specification is the minimum requirement and specifies a quite coarse delivery, and most inspection companies exceed this in their deliveries. However, it gives a good framework for further development of programs to treat the inspection data and for calculation of capacity.

3.3.2 Level 1: Detailed WT and SO data

This part specifies the file format of the wall thickness (WT) and stand-off (SO) data files, which are given for each joint. These have to be specified for the entire joint around the entire circumference. They represent the raw data from a UT inspection run. These datasets are important in reliability assessments, as they represent the basic data for the pipeline joint. The files are organised so that axial distance along the pipeline is shown in the first column, and the remainder of the columns contain remaining wall thickness around the circumference. The first few lines of such a file can be seen in Figure 4.1.

3.3.3 Level 2: Establish river bottom profile

This part is concerned about the transformation of the detailed WT and SO data over to a RBP. The RBP is a two-dimensional representation of the remaining wall thickness as a function of length along the pipeline joint. It can be established either for a finite patch around the circumference or as a representation for the entire pipe. The sensor data contains, in most cases, significant amounts of noise and misreadings. For example, girth welds may show up as sudden changes in wall thickness. The RP gives an algorithm to filter data so that some of the misreadings can be resolved. However, to provide consistent results, the filtering requires data of overall good quality, with heavy echo loss and misreadings restricted to small areas.

The algorithm will be more extensively elaborated in Section 4.2.3, brief concepts will be discussed in the following.

In short, the algorithm can be summarised as follows:

Starting point: From the level 1 delivery from the inspection companies, WT and SO data files of good quality must be given.

Establish RWT_{SO} : RWT_{SO} stands for "Remaining wall thickness", and is derived from the treatment of the WT and SO files. The establishment is dependent on assuming that $WT + SO = WTSO$ is close to constant and that relatively large abrupt variations in this value compared to its neighbouring elements indicate erroneous data. It is recommended to let it vary within the $\pm 99\%$ upper quantile of the measurement accuracy plus half the fabrication tolerance. Erroneous data is replaced using a replacement scheme.

Producing the RBP-dataset: The river bottom profile is produced using an algorithm that takes missing values in the RWT_{SO} into account.

The use of RBP to estimate the defects is good when assessing long cracks, as it does not take width effects into account. It may be too conservative when estimating the interaction effects, as the defects take the "worst possible path" over the circumference. The advantage can be observed when dealing with complex-shaped defects, varying in depth over the length. In such cases, the RBP-approach can be considered less conservative than the most used alternative; to approximate metal loss as rectangular patches with a constant depth equal to the deepest point of the defect. However, for this to be valid, the RBP must be used to determine the effective geometry of the defect. Previous comparisons between these methods exist in the literature, for example, as shown by Hennig et al. (2012).

3.3.4 Level 3: Calculate capacity

This part concerns itself with capacity calculations on the pipelines. It makes use of the RBP established in Section 3.3.3, and enters this into an extensive algorithm. An overview of the algorithm provided from the RP is shown in Figure 3.3.

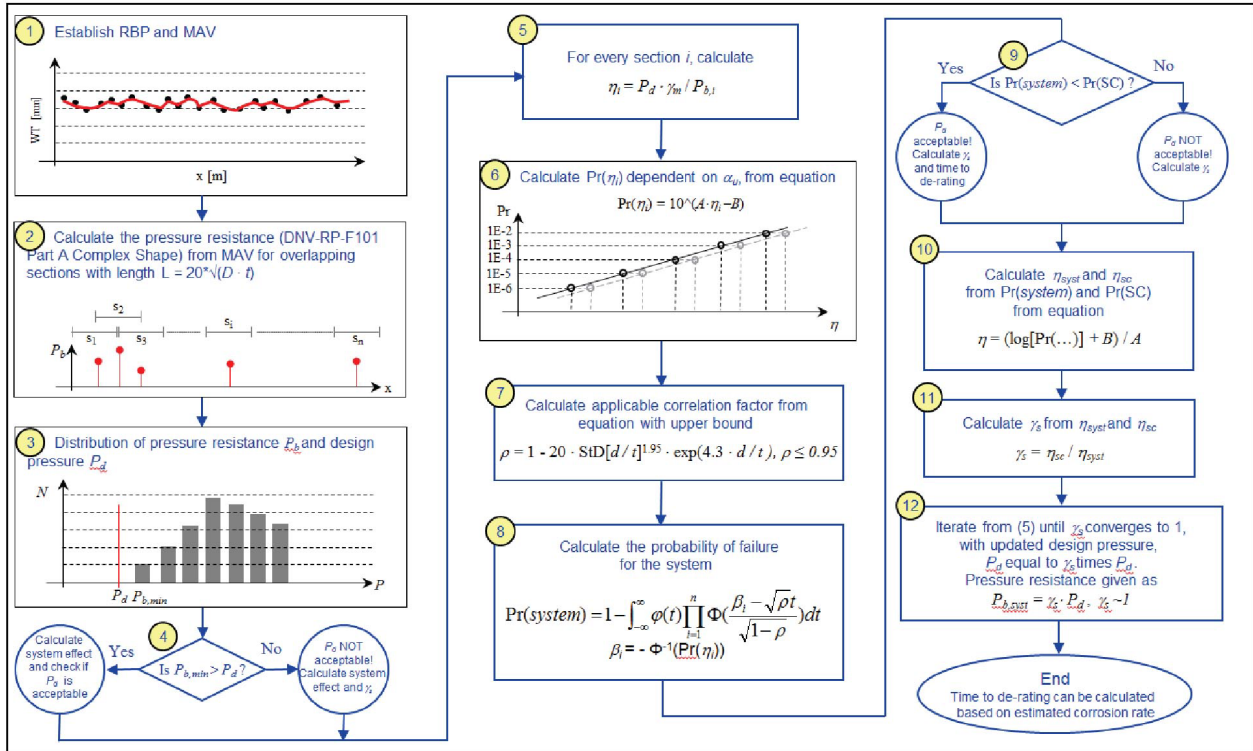


Figure 3.3: Assessment flow diagram of the capacity calculations performed in level 3. Image taken from DNV-GL (2019a).

The entire algorithm is too extensive to review in detail here, but the first few steps will be mentioned. The first step is to establish a moving average (MAV) profile, which is a smoothed version of the RBP. The smoothing is performed by averaging over a length equal to $k \cdot \sqrt{D \cdot t}$, where $k = 0.5$, D denotes the diameter of the pipeline and t denotes the thickness. This is done to minimise the influence of noise or low wall thickness readings from single bins on the total capacity. Further, pressure resistance calculations are done on the MAV file, comparing it to the rated pressure of the pipeline. These capacity calculations are done with the pipeline split up in sections of length equal to $20 \cdot \sqrt{D \cdot t}$. These sections overlap so that defects spanning over section boundaries are captured. The remainder of the algorithm concerns system effects and correlation, which are important when looking at the probability of failure for the pipeline. The calculations will, in short, determine if the pipeline is fit for its rated pressure in the condition it is in today.

3.3.5 Level 4: Corrosion development

Here, the future corrosion growth is estimated, based on the data from the previous levels. The corrosion development calculations used here require access to multiple inspection runs, and provide simple estimates for the future corrosion rate. The following methods are given:

Calculation of bias: Having access to previous inspection data from the same pipeline, bias in the wall thickness can be estimated. This is found by comparing the wall thickness an uncorroded area for each of the two inspection runs. The bias can be different for a corroded area, due to variations in cleaning, material variations, etc., which can cause the sound waves to give different results. However, it is better to use this estimated bias than to disregard it entirely.

Calculation of corrosion rate: Corrosion rate is estimated as the difference between two inspection runs. This means that future growth is extrapolated from the previous observations.

Pressure resistance versus time: Based on the estimated corrosion rate, future pressure resistance can be calculated using the capacity calculations from Level 3.

3.4 Reliability Analyses on corroding pipelines: An overview

Corrosion growth is, in most practical cases, regarded as a highly stochastic process hard to model exactly using deterministic models. By obtaining inspection data of high quality, the momentary corrosion with corresponding capacity can be found. However, significant uncertainties in the inspection data make these results unreliable, and ensuring conservatism might mean underestimating the resulting capacity. Using these methods does not give a qualitative measure of conservatism either; it is simply an estimate based on the safety factors applied. Another thing that might be harder to model is the future growth, which will govern the future capacity of the pipeline. Reliability-based methods are expected to yield superior results compared to deterministic methods, especially when dealing with problems of stochastic nature, being more flexible, accurate, and providing a means of controlling conservatism. They are applicable to most engineering problems faced, though application on pipeline corrosion is limited within the industry today. Since pipeline corrosion is a process concerned with high degrees of uncertainty, the reliability analyses should provide good methods for estimating the capacity and model error in one. They should also be beneficial concerning cost-control and improvement of safety since the accuracy is controllable.

This section starts with defining the most common limit state formulations for reliability analyses on pipelines. Further, some growth models applied in the petroleum industry are reviewed. These include deterministic models and some examples of probabilistic models. Also, modelling of defect growth using the hierarchical Bayes approach and using the Markov Chain model are described, which are methods applicable when uncertain data from multiple inspection runs are available. The last part of this section reviews two reliability-approaches found in the literature; one application of Monte Carlo simulation and one application of FORM for calculating reliability on corroded pipelines. Here, the application and implementation of the methods are discussed, not the basics of the methods themselves, which are covered by Section 2.

3.4.1 Limit states

The starting point for most reliability analyses is to define the limit states. These can vary with the model in question since not all failure modes are as relevant when considering pipelines. A pipeline is, in essence, a series system, where failure at any defect along the length implies failure of the entire system. This has some important consequences when dealing with the pipeline's total probability of failure, as it has to take multiple active defects into account. The probability of failure at one point will depend on the internal pressure, local metal properties, and crack geometry, including depth. Relevant failure modes will depend on the case and model in question.

Reliability analyses need to distinguish between different failure modes. The reasoning behind this is governed by failure consequence considerations and failure mode. In general, a pressurised steel pipe can fail due to two different causes at an active corrosion defect, namely by small leaks and bursts. These are further described as follows (Wenxing Zhou, 2011):

Small leak: A defect penetrates the pipe wall, causing a small leak.

Burst: The pipe wall undergoes plastic collapse at the defect location prior to the defect penetrating the pipe wall. A burst can be further divided into **Large leak** and **Rupture**. The rupture is an instability phenomenon, where the through-wall flaw axially extends itself due to the pressure. It can be discerned from the large leak by some fracture-initiation criterion based on the mechanical properties and the internal pressure.

The most severe of these failure modes is the rupture. In addition to having an unstable defect growth, which leads to a large leakage, the risk of ignition is significant when considering onshore facilities (Nessim et al., 2009). The different failure modes are considered mutually exclusive in most models. The presence of a small leak is assumed to be detected rather quickly, so there is no chance that it develops into a burst. Whether a burst develops into a large leak or a rupture depends on

the local conditions, but they are exclusive, being two different states of a burst. In reality, it is possible that small leaks can develop into bursts if the remaining ligament experiences severe crack growth leading to plastic collapse, but it is unlikely given good detection capacity.

In considering a pipeline with multiple active corrosion defects, the failure at different defects will be statistically correlated (Gong and Wenxing Zhou, 2017). This can be due to local pipe variations (e.g. pipe wall thickness, yield strength, etc.), environmental factors and pressure variations over the length of the pipeline. The concept of correlation is important in any reliability analysis, and it must be handled accordingly when setting up an analysis.

3.4.2 Deterministic corrosion growth models

The corrosion growth models are descriptions of how corrosion defect geometry is expected to develop over time. They are important in determining inspection intervals and deciding when pipe replacement projects should be initiated since they essentially represent a forecast for how the integrity of the pipeline develops over time. An overly conservative growth rate will lead to unnecessary handling, while the opposite may lead to unforeseen pipeline failure. The most important phenomenon to describe is the pitting corrosion (Vanaei et al., 2017), as this may relatively fast lead to leakages or hazardous situations. The deterministic models are easy to evaluate and set up but may lose some of the inherent randomness in the corrosion process.

The simplest model is the single-value corrosion growth rate (SVCR) model. It assumes constant corrosion growth, applying a rate recommended by relevant standards, such as the rate suggested by NACE (Vanaei et al., 2017) at 0.4 mm/year. This model is obviously very limited in regards to accuracy, but its simplicity makes it a widespread model nonetheless.

Another simple method, almost equal to the SVCR, is the linear growth model, which depends on a growth factor estimated either from standards or from data from multiple inspection runs. The linear growth model is the most widely used model due to its simplicity (Al-Amin and Wenxing Zhou, 2014). An example of one algorithm estimating the future growth using data from two inspection runs can be found in 3.3.5, which is the approach recommended by DNV-GL (2019a). A different approach is the one adopted by Gong and Wenxing Zhou (2017), who used one set of inspection data and implemented the linear growth model on the lengths and depths of the defects. The resulting reliability is time-dependent. The formulations here are as follows:

$$l(t) = l_0 + g_l t \quad (3.3a)$$

$$d(t) = d_0 + g_d t \quad (3.3b)$$

where l_0 and d_0 denotes the initial defect length and depth respectively, and g_l and g_d denotes the growth rate of the length and depth, respectively. The growth rates are applied as random parameters following a Weibull distribution, simulating the inherent randomness in corrosion growth. The correlation in corrosion growth rate at different defects within each segment is also taken into account.

A deterministic alternative to the linear model is the nonlinear growth rate model. It depends on extensive knowledge of the surroundings and of the pipe material:

$$h(t) = \alpha(t - t_{ini})^\nu \quad (3.4)$$

where α and ν are soil-dependent parameters, t_{ini} is the initiation time for the corrosion (Vanaei et al., 2017).

3.4.3 Probabilistic growth models

Corrosion is, in most cases, an uncertain process. This is, in general, a valid statement for corrosion growth on pipelines, which has inherent randomness in its nature. The corrosion growth models

should cover these phenomena, and this is why probabilistic growth models have been developed. Though they can take uncertainties into account, they have some important limitations; the data adopted must be statistically homogeneous, and the external conditions must be nominally consistent (Vanaei et al., 2017). But even if the conditions are not completely fulfilled (e.g., the presence of welds may contradict statistical homogeneity), the statistical models may give reasonable estimates for corrosion growth.

The probabilistic models commonly utilised for oil and gas transmission pipelines include the following (Vanaei et al., 2017):

The Markov Model: A corrosion distribution following the Markov model uses a continuous-time, non-homogeneous linear growth, or pure birth approach. It allows modelling of future corrosion growth from observed properties. This method is more extensively described in Section 3.5.

Monte Carlo Simulation: MCS is well known for its versatility within engineering, and it is also applicable for modelling defect growth. If a dataset of corrosion defects is given, the future growth may be estimated using growth models with stochastic growth parameters. Distributions of the uncertain parameters must be assumed; a common approach is to use historical data for finding good models. The future growth may then be estimated using the MCS, either by finding a most likely outcome or a range of outcomes (D. J. F. Kiefner and Kolovich, 2007). A limitation with this is that the MCS may be computationally expensive, depending on the model in question.

Time dependent GEVD model: Time-dependent Generalised Extreme Value Distribution (GEVD) uses a time-varying corrosion rate distribution for a general textural soil (Vanaei et al., 2017). The parameters in this model must be estimated from empirical data, making it hard to use for most offshore applications, as the external conditions are different from those of buried onshore pipelines.

Time independent GEVD model: This is similar to the time-independent GEVD model; the difference is that the parameters of the GEV-distribution are independent of time, equal to their value at time t_0 (e.g. inspection time).

Gamma process: The Gamma distribution is a two-parameter continuous distribution defined by the two parameters θ and k , which denote the scale and shape parameters, respectively. The shape of the distribution changes drastically when altering the shape parameter, making it a suitable fit for various experimental data. The Gamma process is a stochastic process with independent, non-negative increments following the Gamma distribution. This makes it suitable for modelling of degradation over time, which will accumulate in one direction. An example is an implementation suggested by Mahmoodian and Alani (2014), where a stochastic Gamma process model is presented to account for temporal variability in a concrete pipeline.

3.4.4 Hierarchical Modelling of defect growth

When data from multiple inspection runs are available, possibilities arise for implementing more advanced approaches for modelling defect growth. The Hierarchical Bayes analysis can be implemented with various growth models, though the overall set-up and reasoning behind using them are the same. Even if multiple ILI-datasets are available, these are heavily influenced by errors, making the estimated growth rates uncertain. Also, individual defects may have growth paths that are not captured by adjusting the growth models. The use of hierarchical models combined with Bayes models makes it possible to account for random effects within the stochastic deterioration process and account for uncertainties in the inspection datasets.

For description of these hierarchical models, a hierarchy of deterioration uncertainty must be established. Figure 3.4 shows a schematic overview of four hierarchical levels. The top level represents

the system-wide uncertainties, capturing all model uncertainties, while the three lower levels apply to each local point considered in the analysis. The system-wide uncertainties considered in the top level are typically epistemic uncertainties connected to the corrosion model itself. The second level contains location-specific or inter-element effects. They can be caused by external factors, such as temperature changes or exposure to stress, but they also capture aleatory effects due to variables unknown or not described in the models. The third level captures temporal uncertainties that are not included in the two levels above. This uncertainty will be dependent on the time increment Δt , and can be said to capture the drift in uncertainties as time goes on. The lowest level captures the observation uncertainty, i.e., the uncertainty in the inspection tool itself and the reportability of the results.

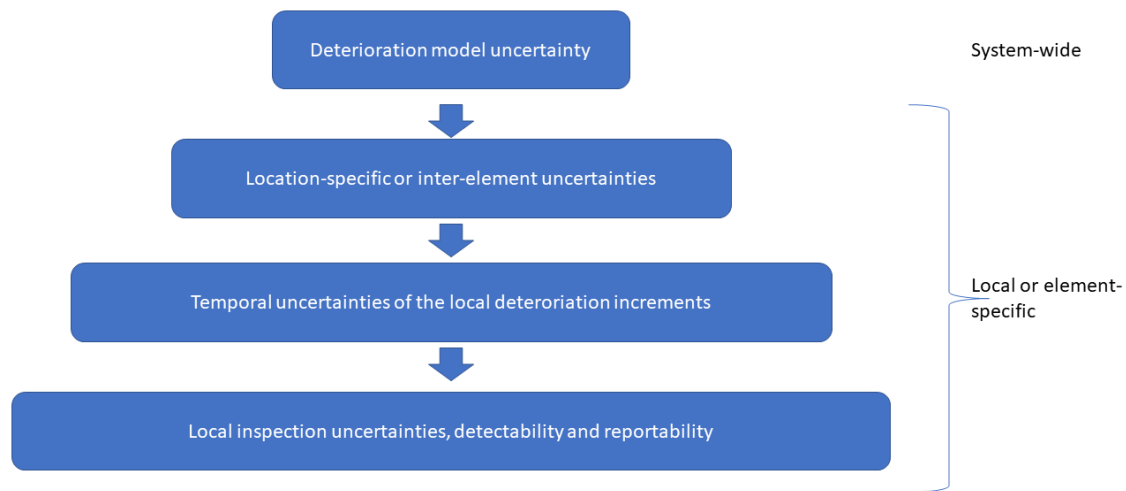


Figure 3.4: Hierarchical representation of uncertainties for modelling system deterioration. Image is inspired by Maes et al. (2009).

Implementation of the model involves setting up a Bayesian framework mirroring the uncertainty hierarchy described in Figure 3.4, where every uncertainty in each layer is described. This means that the large, complex problem of quantifying uncertainties is broken down into local, simple conditional relationships and sub-models. Each uncertainty is described by simple relations, and contain elements of randomness described by carefully chosen distributions. These sub-systems combined should provide a satisfactory synthesis of the current and future behaviour of the entire system.

Maes et al. (2009) describes the build-up of a framework, where the lower level uncertainties consist of both aleatory uncertainties characterised by random processes and of epistemic uncertainties which are implemented into the system-wide deterioration model. On the top level, this system ends up being described by four statistical parameters. As inspection data becomes available, the model is updated to suit the new information. The model is updated using the Markov Chain Monte Carlo (MCMC), which can update the posterior pdfs of upper-level variables given lower-level input.

3.5 Markov Chain models: Corrosion growth and usage in reliability analyses

The Markov chain models are built around the Markov processes, which are commonly used for describing state-changing systems over time. The most basic condition of these processes is that a process developing through time follows the *Markov property*, which states that the distribution of each development increment is independent of what has happened in the past, and is only dependent on the current state. Hence, given the initial state, and by assuming random process parameters, a Markov chain can be set up to model a time-dependent process such as corrosion.

The following is a brief mathematical formulation of a system that can be modelled using Markov chain theory. Consider a system S , which can be in one of the following states S_0, S_1, S_2, \dots , which is a finite, countable set. The time T is continuous, and the transition of the system from one state to another occurs at random priority unknown moments of time T . $S(T)$ denotes the state of the system S at time T , while $P_i(T)$ denotes the probability of the system being in the state S_i at the time T :

$$P_i(T) = P\{S(T) = S_i\} \quad (3.5)$$

A random process that evolves in the system S can be considered a *Markovian process* if the probability of each of the system states in the future only depends on the current state of the system. This means that the future is not depending on how or when the system entered its current state, only that it is in it. This property does not disconnect the future state from the past states, because they are still the reason behind the system ending up in its current place. Said in a slightly different manner; "For a Markov random process, the *future* depends on the *past* only via the *present*." (Timashev et al., 2008).

For pipeline corrosion modelling, a Markov chain can only go in one direction, since the defects cannot "heal" themselves. They are suitable for modelling reliability from ILI data, as these data do represent a present state where the exact path to this state is uncertain.

For calculation of system reliability, Markov Chains are commonly combined with Monte Carlo simulations. Such analysis schemes are frequently called Markov Chain Monte Carlo. Qin et al. (2015) estimated pipeline reliability from inspection data, starting with a gamma process-based growth model, where the model parameters were estimated using a maximum-likelihood method. This growth model was used in combination with a Bayesian-updating scheme for the evaluation of data from multiple inspection runs. The MCMC simulation was then employed to evaluate the posterior distributions from the power growth model. Using Monte Carlo to generate the random parameters for each increment, of which the distributions depend on the assumed growth model, the reliability could be estimated. The Markov chain was also used to account for missing data by estimating growth from a prior inspection.

Al-Amin and Wenxing Zhou (2014) applied a slightly different approach for modelling corrosion growth. They employed a power-law growth model, where the uncertain growth model parameters were estimated by using Bayesian updating in combination with MCMC. The reliability analysis was then employed on this MCMC-generated dataset for estimation of system reliability.

More complete descriptions of Markov processes can be found in the literature. For example, Stroock (2005) has written an introduction to Markov processes, starting with simple random walk theory and continuing with Markov processes both in discrete and in continuous time. The transition matrix for systems is also described, which is important when setting up such a model. A more extensive in-depth description is considered beyond the scope of this project.

3.6 Monte Carlo Simulation for reliability analysis on corroding pipelines

One of the most popular forms of reliability analysis is the Monte Carlo simulation. It can be implemented for most engineering problems due to its simplicity, and it is also useful in analysing corroding pipelines. This section briefly reviews a methodology used to evaluate time-dependent system reliability on a pipeline containing multiple active corrosion defects by Wenxing Zhou (2010), and all equations and descriptions are based on the methodology description found there. The methodology behind the Monte Carlo simulation itself is described in Section 2.3.2, and is not treated in detail in this section. Instead, the overall set-up and definition of limit states and stochastic variables are considered.

3.6.1 Limit States

The limit states defined by Wenxing Zhou (2010) at a given corrosion defect are small leak, burst and rupture. The small leak limit state is given as:

$$g_1 = 0.8t - d_{max} \quad (3.6)$$

where t is pipe wall thickness and d_{max} is maximum defect depth. $g_1 \leq 0$ represents a small leak. The reasoning behind using $0.8t$ in the equation that it is consistent with industry practice. The burst limit state is given as:

$$g_2 = r_b - p \quad (3.7)$$

where r_b is burst pressure resistance of a pipe containing a corrosion defect; p is the internal pressure, and $g_2 \leq 0$ represents a burst failure. For evaluation of the burst pressure, the PCORRC model (Leis and Stephens, 1997) was selected due to its simplicity. It is calculated as follows:

$$r_b = \chi \frac{2\sigma_u t}{D} \left[1 - \frac{d_{max}}{t} \left(1 - \exp \left(\frac{-0.157l}{\sqrt{\frac{D(t-d_{max})}{2}}} \right) \right) \right] \quad (3.8)$$

where σ_u is the pipe ultimate tensile strength; D is pipe diameter; l is defect length, and χ is a multiplicative error term. The values adopted for χ was based on values given by Leis and Stephens (1997), which indicated mean and standard deviation of χ equal to 0.97 and 0.105 based on full-scale burst test data reported in the literature. χ was further assumed to follow a lognormal distribution.

The limit state for rupture is defined as:

$$g_3 = r_{rp} - p \quad (3.9)$$

where r_{rp} is the rupture pressure, i.e. pressure resistance of a pipe containing a through-wall flaw resulting from a burst. The through-wall flaw will develop unstable axial extension if $g_3 \leq 0$. The rupture pressure model was suggested by CSA (2007) and Nessim et al. (2009), and was based on the flow stress-dependent failure criterion developed by J. F. Kiefner, Maxey, et al. (1973). The rupture pressure r_{rp} is given by:

$$r_{rp} = \frac{2t\sigma_f}{MD} \quad (3.10a)$$

$$M = \begin{cases} \sqrt{1 + 0.6275 \frac{l^2}{Dt} - 0.003375 \frac{l^4}{Dt}}, & \frac{l^2}{Dt} \leq 50 \\ 3.293 + 0.032 \frac{l^2}{Dt}, & \frac{l^2}{Dt} \geq 50 \end{cases} \quad (3.10b)$$

where M is the Folias factor, and σ_f is the flow stress defined as $0.9\sigma_u$. The model error was ignored.

By utilizing the limit state functions g_1 , g_2 and g_3 , the limit states can be defined as:

Small leak: $g_1 \leq 0 \cap g_2 > 0$

Burst: $g_1 > 0 \cap g_2 \leq 0$

Large leak: $g_1 > 0 \cap g_2 \leq 0 \cap g_3 > 0$

Rupture: $g_1 > 0 \cap g_2 \leq 0 \cap g_3 \leq 0$

where \cap denotes the joint event. All the limit states depend on time, as the pipe resistance is deteriorating over time due to corrosion growth. The defect growth was characterised by random variables for simplicity.

3.6.2 Formulation for the system reliability analysis

To ensure the reliability of the entire pipeline, Wenxing Zhou (2010) states that it is demanded that the failure probability for a given reference segment length (e.g. 1 km) is below the allowable limit for a reference period of time (e.g. one year). The system considered is a single pipeline segment containing n ($n > 1$) active corrosion defects. The internal pressure loading on the defects is assumed identical and fully correlated. This assumption is valid because the pressure drop over a relatively short pipeline segment is generally low, and because the pressure fluctuations at different points along a segment length can be considered completely in sync. Due to this correlation in loading, failure at different defects will also be correlated, even if the resistances at individual defects are independent.

The following is defined:

- \mathbf{X} represent the vector of random variables involved in the limit state functions, except for the internal pressure.
- $F_P(p)$ denote the cumulative probability distribution function of the annual maximum internal pressure.
- $P_{sla}(\tau)$, $P_{lla}(\tau)$ and $P_{rpa}(\tau)$ denote the annual probabilities of small leak, large leak and rupture, respectively, within the time interval $[\tau - 1, \tau]$ (τ in years).

Included in the vector \mathbf{X} are the wall thickness, tensile strength, model error, defect length, defect maximum depth, defect maximum depth growth rate, and defect length growth rate. These parameters follow different assumed distributions, and more thorough description and justifications are not included in this review.

The internal pressure is the loading factor that is important in this analysis. It is characterised as a simple discrete stochastic process consisting of a sequence of identically distributed random variables each acting over a deterministic period of one year. The magnitude of the pressure follows the annual maximum internal pressure distribution.

The annual failure probabilities conditional on \mathbf{X} can then be formulated as:

$$P_{sla|\mathbf{X}}(\tau) = \begin{cases} \bar{P}_{b|\mathbf{X}}(\tau - 1) & \tau - 1 < T_{sl} \leq \tau \\ 0 & \text{otherwise} \end{cases} \quad (3.11a)$$

$$P_{lla|\mathbf{X}}(\tau) = \begin{cases} \bar{P}_{b|\mathbf{X}}(\tau - 1)I_{ll|\mathbf{X}} & 1 \leq \tau < T_{sl} \\ 0 & \tau \geq T_{sl} \end{cases} \quad (3.11b)$$

$$P_{rpa|\mathbf{X}}(\tau) = \begin{cases} \bar{P}_{b|\mathbf{X}}(\tau - 1)I_{rp|\mathbf{X}} & 1 \leq \tau < T_{sl} \\ 0 & \tau \geq T_{sl} \end{cases} \quad (3.11c)$$

$$\bar{P}_{b|\mathbf{X}}(\tau) = \prod_{k=1}^{\tau} F_P(r_{bm}(k)) \quad (3.11d)$$

$$\bar{P}_{b|\mathbf{X}}(0) = 0 \quad (3.11e)$$

$$I_{ll|\mathbf{X}}(\tau) = \begin{cases} F_P(r_{rpm})(\tau) - F_P(r_{bm})(\tau) & r_{bm}(\tau) < r_{pm}(\tau) \\ 0 & r_{bm}(\tau) \geq r_{pm}(\tau) \end{cases} \quad (3.11f)$$

$$I_{rp|\mathbf{X}}(\tau) = \begin{cases} 1 - F_P(r_{rpm}(\tau)) & r_{bm}(\tau) < r_{pm}(\tau) \\ 1 - F_P(r_{bm}(\tau)) & r_{bm}(\tau) \geq r_{pm}(\tau) \end{cases} \quad (3.11g)$$

$$T_{sl} = \min\{T_{sl1}, T_{sl2}, \dots, T_{sln}\} \quad (3.11h)$$

$$r_{bm}(k) = \min\{r_{bi}(k)\} \quad (i = 1, 2, \dots, n), (k = 0, 1, 2, \dots, \tau) \quad (3.11i)$$

where T_{sli} ($i = 1, 2, \dots, n$) is the time required for the i th defect to reach a small leak state given \mathbf{X} ; $\bar{P}_{b|\mathbf{X}}(\tau)$ is the probability of no burst failure within the time interval $[0, \tau]$ conditional on \mathbf{X} ; $P_{lla|\mathbf{X}}(\tau)$ and $P_{rpa|\mathbf{X}}(\tau)$ are the probabilities of large leak and rupture respectively within the time interval $[\tau - 1, \tau]$ given \mathbf{X} ; $r_{bm}(\tau)$ is the minimum value of the burst pressures associated with the n defects at time τ , and $r_{rpm}(\tau)$ is the rupture pressure at the defect that has the minimum burst pressure at time τ .

If there are no interactions between individual defects, and the internal pressure undergoes no systematic changes, the unconditional annual failure probabilities can be evaluated as follows:

$$P_{sla}(\tau) = \int P_{sla|\mathbf{X}}(\tau) \mathbf{f}_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \quad (3.12a)$$

$$P_{lla}(\tau) = \int P_{lla|\mathbf{X}}(\tau) \mathbf{f}_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \quad (3.12b)$$

$$P_{rpa}(\tau) = \int P_{rpa|\mathbf{X}}(\tau) \mathbf{f}_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \quad (3.12c)$$

These cumulative probabilities can be directly evaluated using the brute Monte Carlo simulation, in the paper by Wenxing Zhou (2010) this is done for a sensitivity study on a hypothetical case. Results from this analysis suggest that spatial variations in pipe properties have negligible impact on the system reliability. In contrast, the spatial variability of internal pressure, initial defect sizes and defect growth rates can have a significant impact.

3.7 FORM-based Reliability Analysis on pipelines

This section reviews the methodology developed by Gong and Wenxing Zhou (2017). It applies FORM to evaluate the time-dependent system reliability of a pressurised steel pipeline containing multiple active corrosion defects. It is based on methodology from Der Kiureghian (2005), and aims to develop an efficient FORM-based methodology for this problem.

The analysis consists of two key components. It first constructs two equivalent linearised time-dependent limit functions in the standard normal space. The equivalent limit states are based on individual FORM-assessments of each of the multiple defects. Secondly, a procedure is developed for calculating incremental probabilities of leak and burst of the pipeline, valid for small time steps. By using this, future reliability can be estimated based on the linearised equivalent limit states in addition to an estimate of the growth rate.

3.7.1 Limit state formulation

In the methodology by Gong and Wenxing Zhou (2017), two time-dependent limit states are formulated for each defect; one for penetration of the pipe wall (small leak) and one for plastic collapse (burst). The different forms of burst are not considered in this model. The different limit states are shown in Equation 3.13 and 3.14.

$$g_i^s(t) = \phi wt_i - d_i(t) \quad (3.13)$$

$$g_i^c(t) = p_{ci}(t) - p_i \quad (3.14)$$

where g_i^s is the limit state for small leak and g_i^c is for collapse. The subscript i means the parameters are localised at the i th defect. Regarding the parameters of Equation 3.13, wt_i is the pipe wall thickness and $d_i(t)$ is the maximum defect depth at time t . ϕ is a reduction factor to account for increased crack growth probability in the remaining ligament at a defect. In Equation 3.14, p_{ci} represents the burst capacity at time t and p_i denotes the internal pressure. In reality, the internal pressure is best represented as a time-dependent random process, but such an approach would make the calculations too complicated. So, in the model it is characterised by a time-independent random process. For evaluation of the burst capacity, they have decided to use the ASME B31G Modified model by J. F. Kiefner and Vieth (1989), given in Equation 3.15 below:

$$p_{ci} = \xi_i \frac{2wt_i(\sigma_{yi} + 68.95)}{D_i} \left[\frac{1 - \frac{0.85d_i}{wt_i}}{1 - \frac{0.85d_i}{M_i wt_i}} \right], \frac{d_i}{wt_i} \leq 0.8 \quad (3.15)$$

$$M_i = \begin{cases} \sqrt{1 + 0.6275 \frac{l_i^2}{D_i wt_i} - 0.003375 \frac{l_i^4}{D_i wt_i}}, & \frac{l_i^2}{D_i wt_i} \leq 50 \\ 3.3 + 0.032 \frac{l_i^2}{D_i wt_i}, & \frac{l_i^2}{D_i wt_i} \geq 50 \end{cases} \quad (3.16)$$

where ξ is the model error associated with the B31G Modified model; D is the pipe outer diameter; σ_y and $\sigma_y + 68.95$ [MPa] are the yield strength and the empirically defined flow stress of the pipe material, respectively; M is the so-called Folias factor; d is the defect depth and l is the defect length (implicitly dependent on time). This equation is an empirical formula for calculating the capacity.

The probabilities $P_l(t)$ and $P_b(t)$ are defined as probabilities of leak and burst, respectively, within the time interval $[0, t]$. Because of the competing nature of the limit states, the probabilities need to be formulated accordingly. Defining t_i^s as the time the i th defect just penetrates the pipe wall, and t_i^c as the time plastic collapse takes place at the i th defect. Then, the probabilities of failure can be defined as:

$$P_l(t) = Prob \left[(0 \leq \min_i \{t_i^s\} \leq t) \cap (\min_i \{t_i^s\} < \min_i \{t_i^c\}) \right] \quad (3.17)$$

$$P_b(t) = Prob \left[(0 \leq \min_i \{t_i^c\} \leq t) \cap (\min_i \{t_i^c\} < \min_i \{t_i^s\}) \right] \quad (3.18)$$

3.7.2 FORM Formulations

The methodology by Gong and Wenxing Zhou (2017) is built on the following relation; for a series system with n statistically dependent limit states, the probability of failure can be evaluated as:

$$P_f = 1 - \Phi_n(\boldsymbol{\beta}, \mathbf{R}) \quad (3.19)$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_n]^T$ denotes the reliability indices of the n limit state functions; $\mathbf{R} = \{r_{ij}\}$, ($i, j = 1, 2, \dots, n$) is a correlation coefficient matrix, giving the relation between linearised safety margins associated with the limit state functions i and j in the standard normal space; and $\Phi_n(\cdot)$ is the n -dimensional standard normal cumulative distribution.

The correlation coefficients are found as $r_{ij} = \boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_j$, where $\boldsymbol{\alpha}_i$ and $\boldsymbol{\alpha}_j$ are the unit normal vectors of the limit state functions i and j , respectively, at their respective design points in the standard normal space.

3.7.3 Equivalent limit state functions

In the methodology by Gong and Wenxing Zhou (2017), the pipeline containing n defects is described by two series systems, each system with n limit state functions describing the small leak and burst states respectively. For use in FORM it is of benefit to compress the series systems of n limit states into two equivalent limit states describing the leak and burst, because it greatly simplifies the mathematical formulations that follow. The methodology used in the paper for this construction is described in the following.

The union of all the random variables (e.g. wall thickness, yield strength, internal pressure and defect sizes) involved in the limit states are represented by the m -dimensional vector \mathbf{X} . The probabilistic characteristics of individual elements in \mathbf{X} can vary with location, giving a representation of the variation in, e.g., corrosion growth conditions along the pipeline.

Considering the series system describing the wall penetration first:

- At a given time t , FORM can be used to evaluate $Prob[g_i^s(t) \leq 0]$. This gives the corresponding reliability index $\beta_i^s(t)$, representing the cumulative probability of the i th defect penetrating the pipe wall within the time interval $[0, t]$.
- Define $\boldsymbol{\beta}^s(t) = [\beta_1^s(t), \beta_2^s(t), \dots, \beta_n^s(t)]^T$ and let $\mathbf{R}^s(t)$ denote the correlation matrix of the linearised safety margins associated with $g_1^s(t), g_2^s(t), \dots, g_n^s(t)$ in the standard normal space.
- $1 - \Phi_n(\boldsymbol{\beta}^s(t), \mathbf{R}^s(t))$ then gives the cumulative probability of any of the n defects penetrating the pipe wall within the time interval $[0, t]$, $P^s(t)$.
- The reliability index $\beta^{se}(t)$ is defined, corresponding to $P^s(t)$. I.e. $\beta^{se}(t) = -\Phi^{-1}(P^s(t))$.
- A linearised equivalent limit state function at time t , $G^{se}(t)$ is then constructed, with reliability index equaling $\beta^{se}(t)$. It follows that:

$$G^{se}(t) = \beta^{se}(t) - (\boldsymbol{\alpha}^{se}(t))^T \mathbf{u} \quad (3.20)$$

where \mathbf{u} denotes \mathbf{X} transformed to the standard normal space, and $\boldsymbol{\alpha}^{se}(t)$ is the equivalent unit normal vector associated with $G^{se}(t)$.

- The unit normal vector $\boldsymbol{\alpha}^{se}$ represents the sensitivity of the reliability index with respect to the random parameters involved. It is evaluated as:

$$\alpha_k^{se}(t) = \frac{\frac{\partial \beta^{se}(t)}{\partial u_k}}{\sqrt{\sum_{k=1}^m \left(\frac{\partial \beta^{se}(t)}{\partial u_k} \right)^2}}, \quad (k = 1, 2, \dots, m). \quad (3.21)$$

$\frac{\partial \beta^{se}}{\partial u_k}$ is found by utilising the chain rule:

$$\frac{\partial \beta^{se}(t)}{\partial u_k} = \sum_{i=1}^n \frac{\partial \beta^{se}(t)}{\partial \beta_i^s(t)} \frac{\partial \beta_i^s(t)}{\partial u_k} \quad (3.22)$$

where $\frac{\partial \beta_i^s(t)}{\partial u_k}$ is found from the FORM-analysis conducted on the single limit states. $\frac{\partial \beta^{se}(t)}{\partial \beta_i^s}$ is found by the finite differences method.

The same approach can be applied to the n collapse limit states, where all $g_i^c(t)$ are combined into $G^{ce}(t)$ with corresponding reliability index $\beta^{ce}(t)$ and equivalent unit normal vector $\boldsymbol{\alpha}^{ce}(t)$. As with the probability of penetration, β^{ce} corresponds to the cumulative probability of plastic collapse for any of the n defects in the time interval $[0, t]$.

In the further calculations the equivalent limit states are used, meaning that the n active defects on the pipeline are now represented by one equivalent defect. This simplifies the calculations of probabilities for leak and burst, P_l and P_b

3.7.4 System failure probabilities

The conditional formulations from Equation 3.17 and 3.18 gives the probabilities for failure at any point within the time interval $[0, t]$. For getting the complete picture getting the development over time, an incremental investigation is needed. The following equations are used by Gong and Wenxing Zhou (2017):

$$P_l(\tau + \Delta t) = P_l(\tau) + \Delta P_l(\tau, \Delta t) \quad (3.23a)$$

$$P_b(\tau + \Delta t) = P_b(\tau) + \Delta P_b(\tau, \Delta t) \quad (3.23b)$$

where $\Delta P_l(\tau, \Delta t)$ and $\Delta P_b(\tau, \Delta t)$ ($0 \leq \tau < t$) are incremental probabilities of leak and burst, respectively. Using geometric relations, the incremental probabilities are found as:

$$\Delta P_l(\tau, \Delta t) = \Phi_3([\beta^{se}(\tau), -\beta^{se}(\tau + \Delta t), \beta^{ce}(\tau)]^T, \mathbf{R}^{sc}(\tau, \Delta t)) \quad (3.24a)$$

$$\Delta P_b(\tau, \Delta t) = \Phi_3([\beta^{ce}(\tau), -\beta^{ce}(\tau + \Delta t), \beta^{se}(\tau)]^T, \mathbf{R}^{cs}(\tau, \Delta t)) \quad (3.24b)$$

where $\mathbf{R}^{sc}(\tau, \Delta t)$ is the 3 x 3 matrix of correlation coefficients for $G^{se}(\tau + \Delta t)$, $G^{se}(\tau)$ and $G^{ce}(\tau)$, and $\mathbf{R}^{cs}(\tau, \Delta t)$ is the 3 x 3 matrix of correlation coefficients for $G^{ce}(\tau + \Delta t)$, $G^{se}(\tau)$ and $G^{ce}(\tau)$. The correlation coefficients between the functions are estimated by the scalar products of their respective unit normals.

It is assumed that $G^{se}(\tau)$ and $G^{se}(\tau + \Delta t)$ are fully correlated, and that $G^{ce}(\tau)$ and $G^{ce}(\tau + \Delta t)$ are fully correlated. This means that Equation 3.24 can be simplified to:

$$\Delta P_l(\tau, \Delta t) = \int_{-\text{inf}}^{\beta^{ce}(\tau)} \int_{\beta^{se}(\tau + \Delta t)}^{\beta^{se}(\tau)} \phi_2(\theta_1, \theta_2, r^{sc}(\tau)) d\theta_1 d\theta_2 \quad (3.25a)$$

$$\Delta P_b(\tau, \Delta t) = \int_{-\text{inf}}^{\beta^{se}(\tau)} \int_{\beta^{ce}(\tau + \Delta t)}^{\beta^{ce}(\tau)} \phi_2(\theta_1, \theta_2, r^{sc}(\tau)) d\theta_1 d\theta_2 \quad (3.25b)$$

where $\phi_2(\cdot, \cdot, \cdot)$ is the pdf of the bivariate normal distribution, and $r^{sc}(\tau) = (\boldsymbol{\alpha}^{se}(\tau))^T \boldsymbol{\alpha}^{ce}(\tau)$ is the correlation coefficient between $G^{se}(\tau)$ and $G^{ce}(\tau)$. With the formulations for $\Delta P_l(\tau, \Delta t)$ and $\Delta P_b(\tau, \Delta t)$, $P_l(t)$ and $P_b(t)$ can be calculated recursively, using Equation 3.23 starting from $P_l(0)$ and $P_b(0)$.

3.8 Notes on the adequacy of the probabilistic models

The probabilistic models included in this section are extensive to set up and depend, of course, on the underlying probability distributions. Applied correctly, they represent methods to estimate the capacity of a pipeline more detailed than by applying the deterministic methods. The most prominent drawback is that they are hard to set up, so they are usually only an alternative when the simpler models predict too low capacity.

Limitations with the probabilistic methods include that they have a hard time dealing with discontinuities or variations internally in the pipeline. Future growth is hard to determine even when applying these models, so there are uncertainties on these as well. Changes in external conditions are also hard to implement in such models. When they are based on inspection data, it is also hard to implement the growth of new defects, which may be an important phenomenon when the capacity is concerned.

3.9 Inspection method

For the last part of the literature review, some information about the inspection method is included for completeness. The datasets which are to be analysed in the following sections are generated from this tool, and it will be of interest to present some information about the accuracy and possible errors of these data.

The inspection tool used for obtaining the data treated in this project is the UT tool. The technology is based on sending ultrasonic pulses through a stand-off medium (e.g. oil), to be partly reflected off the inner wall and fully reflected from the outer surface (Camp, 2006). The returning echo can be analysed, where the difference in transmission times and sound velocity can be used to derive the wall thickness. The sound velocity is known both for the medium and for the steel, making this calculation possible. With a sufficient amount of sensors, the wall thickness of the entire pipeline can be measured.

The stand-off distance (i.e. the distance between the sensor and the pipe wall) can be calculated as well, since the sound velocity in the medium is known, and some of the pulse sent out is reflected off the inner pipe wall. This information is useful, as it allows a distinction between external and internal metal loss. The stand-off signal will change only when an internal metal loss is present (NDT Global, 2020). The working method for the ultrasonic inspection tool is shown in Figure 3.5.

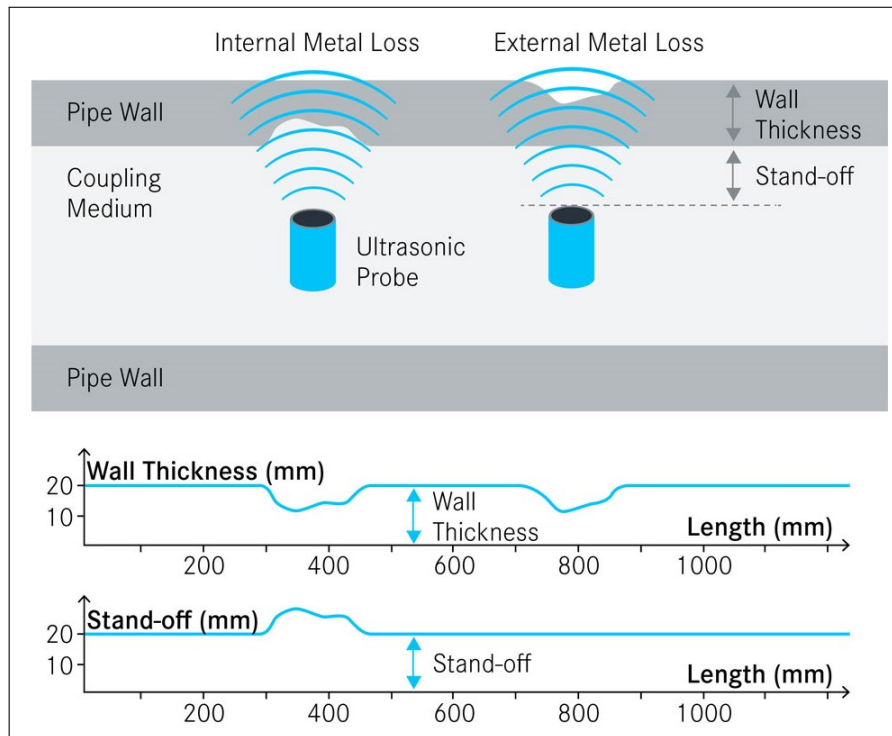


Figure 3.5: Principle of ultrasonic inspection for measuring of remaining wall thickness. Image taken from NDT Global (2020).

4 Methodology description

This section aims to present the methodology applied in this project. The first part of the methodology concerns itself with data filtering of imperfect ILI data, which in general is influenced by noise and misreadings. The starting point of the development done in this project is software developed by Aker Solutions AS for integrity assessment of corroded pipeline segments. This software is again based on methodology from DNV-GL RP F101 (DNV-GL, 2019a), which is reviewed in Section 3.3 as part of the literature review in this project. As this software was written for Python 2.7, an effort has been made to transfer the data filtering algorithm to Python 3.6, enabling more flexibility in programming and also making it possible to use some of the more recent additions to the Python library. The second part of the software aims to provide estimates for the reliability of the pipeline, where the focus is to establish methodology, with little focus on the actual results themselves.

So, this section is split into parts. The first part of this section looks into the software developed by Aker Solutions AS, in addition to reviewing some of the work done by Liu et al. (2018), where different filters from the Python library are compared against the algorithm from the RP.

The following part describes the development of the data filtering methods done in this project, i.e. the transfer over to Python 3. Further development includes an algorithm for identification of defects from the filtered dataset, which was made for interest as an alternative to the RBP-approach from the RP. The output from this program is used in the following reliability analysis.

Since the project is looking at reliability methods on pipelines, it was also of interest to include example calculations using some of the principles reviewed above. The method chosen was the Monte Carlo simulation, used on the list of defects developed from the filtered data. The development of this model is described in the last part of this section.

Flowcharts over the programs developed can be found in Appendix A.

4.1 An Overview: Description of inputs and of the overall flow of the method

ILI data contain, in general, significant uncertainties connected to noise, misreadings and missing data. These measurement errors can have various causes, including systematic error from the ILI tool, varying inspection conditions, environmental factors within the pipeline, system errors and reporting method of data (Al-Amin, W. Zhou, et al., 2012). Attempts have been made to try out different image denoising tools for removing at least some of the noise of the measurements, though with varying luck (Liu et al., 2018). These standard denoising filters do form sort of a lower and higher bound on the wall thickness results, which does promise some possibilities in the future. However, their investigations suggest that the algorithm from DNV-GL provides the best results out of the alternatives reviewed (Liu et al., 2018), as they are used both for missing data treatment and for noise reduction. For reducing more general image problems like noise, though, the image denoising might provide viable alternatives.

4.1.1 Data layout and description

The input to the data filtering is "raw" inspection data, meaning that it contains imperfections. Using this for capacity calculations would, in general, give wrong results. There are two different datasets treated in the analysis: Wall thickness (WT) data and stand-off (SO) data. The pipeline is split up in sections of finite length, and for each section one set of WT-data and one corresponding set of SO-data are given. These data represent the complete datasets for a certain UT inspection run.

Since the data is to be processed by an algorithm, the setup needs to be standardised to make the modelling simpler. The solution chosen in the software is to use CSV-files equally set up for both WT and SO-data following a layout where the first column shows the odometer value and the

remaining columns contain the dataset values. The top row gives the circular position along the circumference.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Odometer	100.0°	101.01010	102.02020	103.03030	104.04040	105.05050	106.06060	107.07070	108.08080	109.09090	110.10101	111.11111	112.12121
2	0.5	17.3438	17.3259	17.245	17.3578	17.3014	17.1206	17.6557	17.4423	17.5584	17.773	17.5365	17.5426	17.3962
3	1	17.4848	17.1502	17.44	17.7788	17.7323	17.7775	17.1467	17.8097	17.1522	17.8447	17.2638	17.5636	17.6757
4	1.5	17.4141	17.1502	17.6203	17.3225	17.4861	17.5497	17.9492	17.5813	17.7857	17.5404	17.1305	17.3904	17.7349
5	2	17.2706	17.5931	17.3945	17.8931	16.9631	17.5761	18.133	17.1728	17.2303	17.7718	17.5178	17.3215	17.773
6	2.5	17.617	17.155	17.9414	17.4697	17.5163	18.004	17.4332	17.5327	17.4447	17.4669	17.1795	17.7206	17.8012
7	3	17.7287	17.5333	17.6741	17.62	17.6252	17.1833	17.8638	17.5205	17.3776	17.545	17.668	17.475	17.4862
8	3.5	17.7024	17.552	17.2178	17.3872	17.5482	17.1189	17.146	17.5163	17.7059	17.892	17.4174	17.6451	17.4827
9	4	17.6072	17.2431	17.3934	16.9096	17.5737	16.8473	17.7509	17.5053	17.8544	17.3977	17.2957	17.2518	17.4618
10	4.5	17.3545	17.064	17.7713	17.3157	17.8813	17.4807	18.2891	17.3584	17.478	17.254	17.2744	17.321	17.4506
11	5.01	17.8058	17.9434	17.424	17.074	17.3983	17.475	17.567	17.3531	17.3988	17.8215	17.5901	17.6145	17.3979
12	5.51	17.2755	17.6338	17.7779	17.2421	17.4922	17.3626	17.8051	17.551	17.691	17.8465	17.662	17.7693	17.9542
13	6.01	17.5871	17.4502	17.7572	17.4948	17.3047	17.7386	17.8247	17.2178	17.8046	17.3948	17.4641	17.1161	17.4516
14	6.51	17.667	18.0921	17.8041	16.9383	17.1647	17.7759	17.5631	17.0899	17.88	17.5428	17.5308	17.3646	17.4938
15	7.01	17.6233	17.398	17.5119	17.6231	17.1303	17.7014	17.3613	17.0701	17.5961	17.6735	17.474	17.3353	17.6069
16	7.51	17.1376	17.4807	17.5103	17.5074	17.4764	17.6852	17.6136	16.9294	17.5297	17.8898	17.4444	17.8851	17.5672
17	8.01	17.4812	17.23	17.658	18.2395	17.3024	17.5715	17.3067	17.7733	17.3448	17.66	17.4867	17.4585	17.5644
18	8.51	17.7002	18.1024	17.2339	17.3989	17.4645	17.4214	17.6842	17.5534	17.4385	17.5298	17.1767	17.6126	17.2478
19	9.01	17.7251	17.5175	17.7445	17.2085	17.7632	17.1044	17.2868	18.0248	17.4176	17.3874	17.568	17.7245	17.6222
20	9.51	17.221	16.8904	17.5126	17.7496	17.6359	17.6433	17.4375	17.6732	17.1621	17.4272	17.3994	17.6742	17.5449

Figure 4.1: First part of an example data file.

Since this setup was used in the original software by Aker Solutions AS, and the datasets analysed in this project were previously set up according to this layout, there was no reason to alter this setup further. The file names are also playing a part in the processing, as it is a wish to automate the analysis of an entire folder of datasets with a simple click. The file names are of the form:

$$[Project_name] - [Odometer_start_value].[Data_type].csv$$

where *Project_name* typically identifies the inspection; *Odometer_start_value* is the distance between the starting point and the start of the section considered (which also is reflected in the Odometer-column within the file); and *Data_type* is either *So* or *Wt*. There must be pairwise *Wt*- and *So*-data with the same file names.

4.2 Methodology in *rbpGen* from Aker Solutions

This description is largely based on an internal report regarding the theory behind the program *rbpGen* (Aker Solutions AS, 2017b). The software filters data according to a pre-set algorithm, which is either based on the methodology provided in the RP (DNV-GL, 2019a), on a simple median-approach or image denoising tools readily made in the Scipy-package (Jones et al., 2020) included in most Python scientific installations. The software is GUI-based, which makes it convenient and user-friendly without requiring much detailed knowledge of the program. It is written in Python 2.7 and uses the Matplotlib module (Hunter, 2007) for plotting the results. The GUI itself is achieved using PyQt4 (Thompson, 2007). The end product has been made into an executable using Pyinstaller (Cortesi, 2019), which makes it easily available on most computers without further installations. For array calculations, additions and maintaining the data structure of results, the Numpy package (Walt et al., 2011) has been utilised.

4.2.1 Brief algorithm description

Simply stated, the algorithm works as follows:

1. Read ILI datasets, which gives WT and SO matrices.
2. Establish WTSO-matrix as the sum of WT and SO.
3. Treat defected sensor data, which is done by replacing columns of which the median is zero with the nearest column with a non-zero median.
4. Perform filtering on the WTSO and SO matrices. The user can choose between the algorithm from DNV-GL, the Gaussian filter or the Median filter. These filters are described below. The output after this step is a matrix of remaining wall thickness, *RWT*.
5. Generate RBP file from the RWT. The RBP denotes the minimum remaining wall thickness around the circumference for each length step and is described in Section 3.3.3. It can be produced for a certain degree-interval specified by the user, or take the whole circumference into account.
6. Generate MAV file, as described in 3.3.4. This is a simple smoothing algorithm, taking away the influence of single bins to the total capacity.

For the capacity calculations on corroded pipelines defect depth is an important parameter (Vanaei et al., 2017). Further implementation of the methodology from Appendix D in RP F101 (DNV-GL, 2019a) involves the application of capacity calculations on the MAV-file. Aker Solutions has written a program performing these calculations, called Corripipe Aker Solutions AS (2017a). This program uses the MAV for finding the maximum defect depth within each defect, which then is entered into the capacity equations.

4.2.2 GUI layout and outputs

An example of a filled-out analysis run is shown in Figure 4.2. The parameters present in the GUI-example are arbitrary and not representing any real case. The files analysed are, however, real examples from a pipeline.

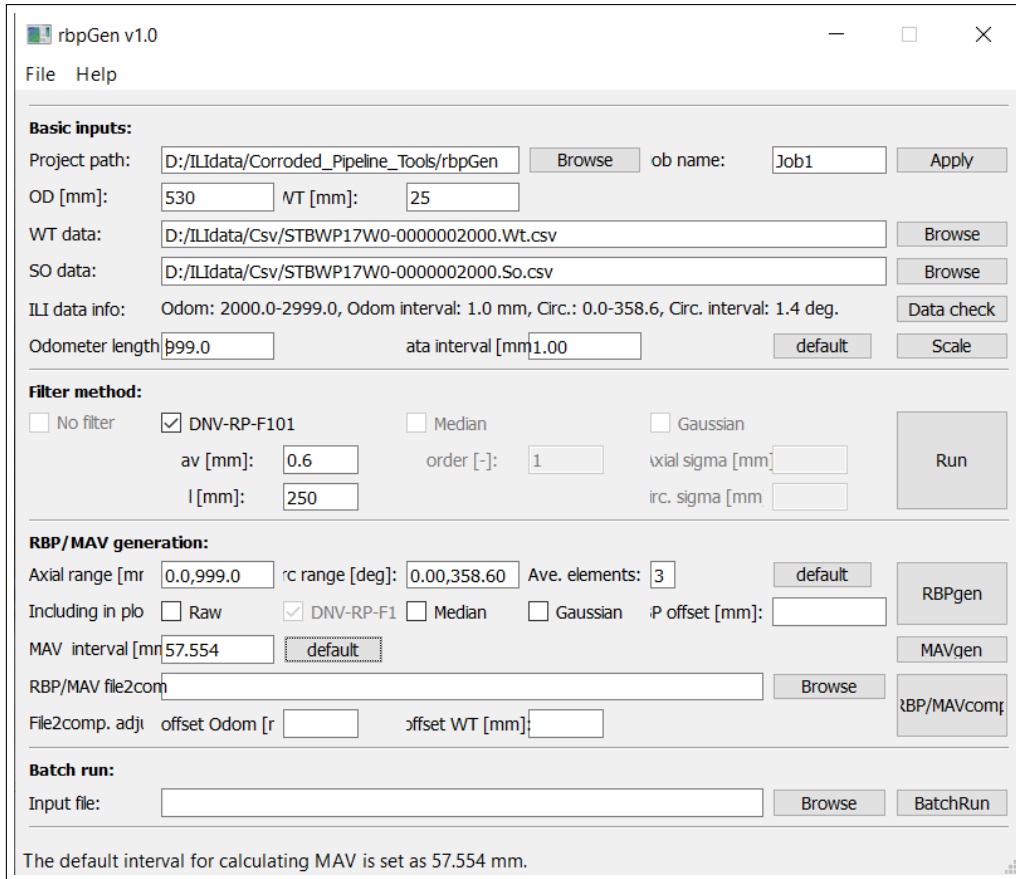


Figure 4.2: Example of a filled out GUI for rbpGen. The numbers are fictional or from default settings. The filter chosen here is the DNV-filter.

By filtering the data using the DNV-filter, the output datasets can be plotted, as shown in Figure 4.3. They are, however, not completely representing the dataset as the program sets lower bounds on the values plotted. So in case of values close to zero, these values are ignored. This can be observed in Figure 4.4, where the RBP gets values below zero some places, while the minimum specified in the corresponding contours shown in Figure 4.3 is 4 mm.

The RWT- and WTSO-datasets shown in Figure 4.3 can further be applied to create RBP- and MAV-files. Examples of these plots are shown in Figure 4.4 below.

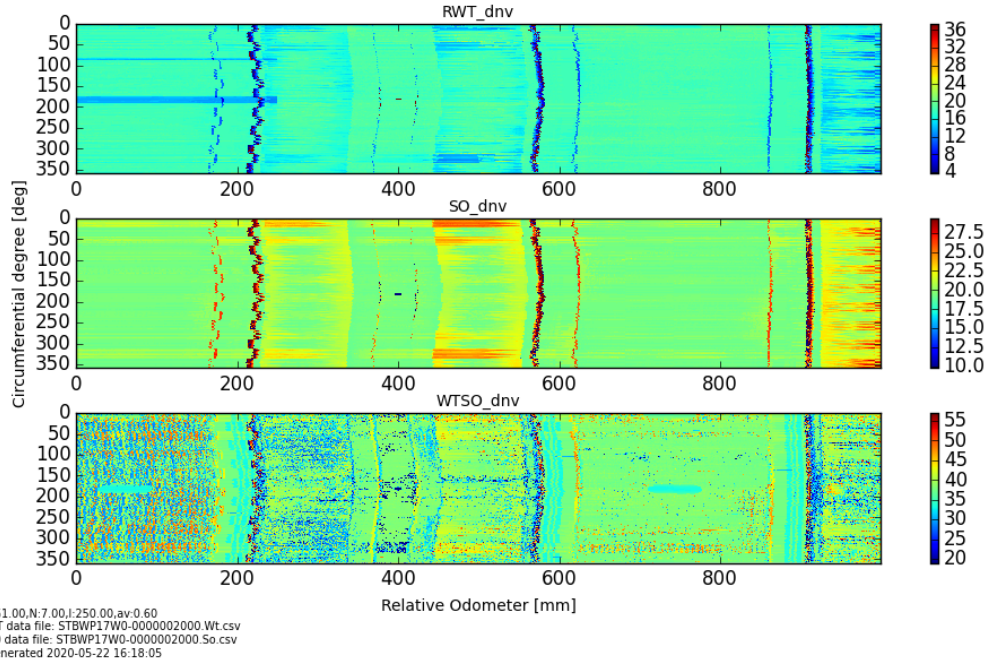


Figure 4.3: Example of filtered data using the DNV-filter. The estimated RWT, SO and WTSO datasets are plotted.

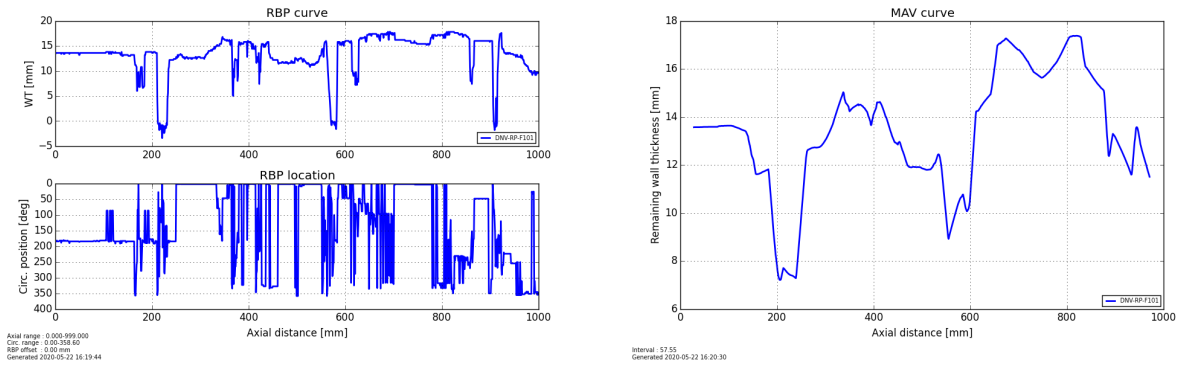


Figure 4.4: Plots of RBP- and MAV-datasets using rbpGen with the specified example inputs.

4.2.3 The DNV-Filter

As stated previously, the DNV-filter implemented in the program is based on the algorithms provided by DNV-GL in their RP F101 (DNV-GL, 2019a). It is built on the assumption that the WTSO will not vary much over the dataset, and that too high variation in an area indicates erroneous data. The output is an array with remaining wall thickness, representing the conditions on the pipeline. This dataset is used to create the RBP and the MAV.

The starting point is the input, described in Section 4.1.1, i.e. a pair of WT- and SO-data. These are combined into the WTSO, which is entered into the filtering algorithm. For filtering, the WTSO-array is iterated over in two directions. The allowable variation for a single WTSO-bin, $WTSO[i, j]$, is allowed to have a variation from the reference derived from the input parameter av . If it is within the allowable range, the remaining wall thickness is generally reproduced from the input arrays as:

$$RWT_{SO}[i, j] = WTSO[i, j] - SO[i, j] = WT[i, j] \quad (4.1)$$

where i, j denotes the row and column index of the element. Reference values for the WTSO- and the SO-values are taken as the median of each column. This is done for the entire joint, $WTSO_{joint\ ref}[j]$

and $SO_{joint\ ref}[j]$, and for l shorter sections, $WTSO_{section\ ref}$ and $SO_{section\ ref}$. The WTSO- and SO-values are allowed to vary within the $\pm 99\%$ upper quantile of the measurement uncertainty together with half the fabrication tolerance. This variation is called *allowable WTSO variation* denoted by the parameter av .

If the $WTSO[i, j]$ is outside the valid range, it is replaced using a replacement scheme. It estimates the erroneous data based on the median of WTSO-data for a given length ($\pm k/2$ elements around the element in question). The length k is recommended to have as low as possible, DNV-GL (2019a) recommends a length of about 150 mm, giving 50-75 data points. The estimate of WTSO and RWT becomes:

$$movWTSO_{median}[i, j] = Median(WTSO[i - \frac{k}{2}, j] : WTSO[i + \frac{k}{2}, j]) \quad (4.2)$$

$$RWT_{SO}[i, j] = movWTSO_{median}[i, j] - SO[i, j] \quad (4.3)$$

Since missing data is reported as zero in the datasets, these values will also lie outside the valid range, and be replaced using the same median scheme. The $movWTSO_{median}$ is controlled for deviations from the reference value in the same way as $WTSO[i, j]$. If it is outside the allowable range, the element is replaced by the lower or upper bound depending on whether the original data is on the lower or upper side. The reference values themselves may also be erroneous; these are controlled against the N neighbouring reference values. The lower and upper bounds are set as the minimum and maximum of the N neighbours, respectively. DNV-GL (2019a) recommends a neighbourhood size of $N = 7$, i.e. applying three reference values to either side of the element in question.

A flowchart for the DNV-filter algorithm can be found in Figure 4.5 below.

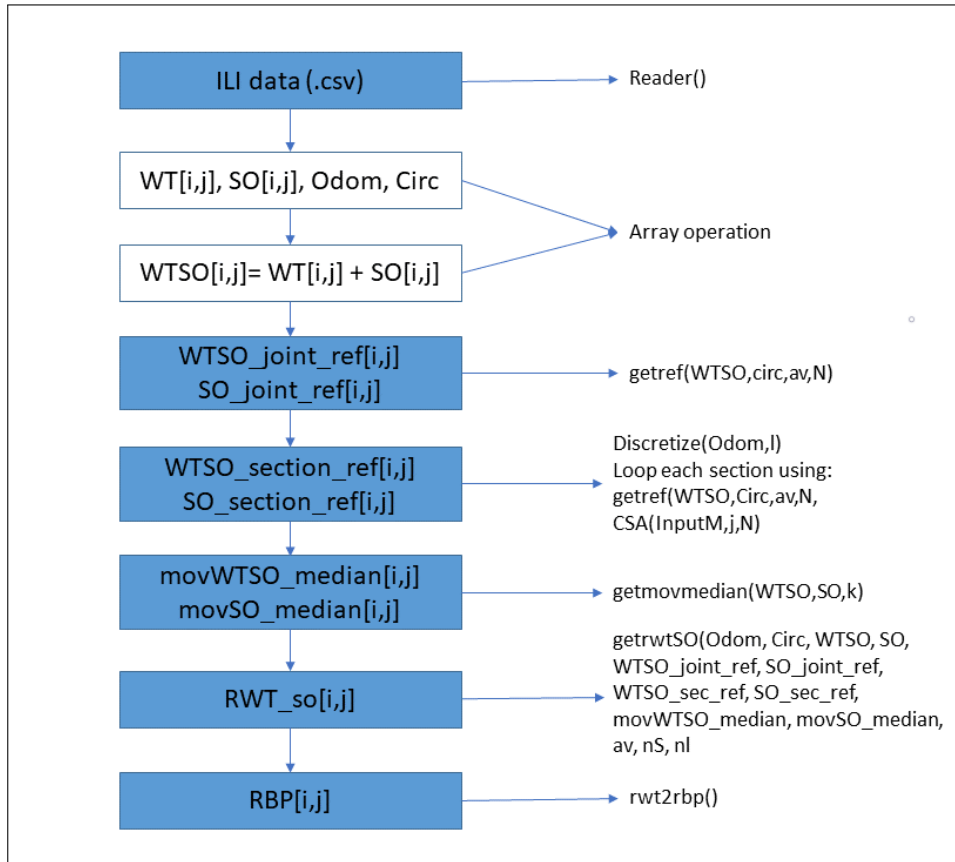


Figure 4.5: Flowchart of the ILI data processing following DNV-RP F101. Image inspired by Aker Solutions AS (2017b).

4.2.4 The Median filter

The median filter applied can filter the same data as the DNV-filter, and uses the assumption that the WTSO-values should not change much. In the program, two alternatives are implemented; one custom made approach using a 1D median filter and one applying the 2D median filter from Scipy. These are briefly described below.

- The 1D median filter approach is implemented using the median of each row or column of the WTSO data as the reference value of the WTSO. The output RWT-array is then calculated subtracting SO from WTSO. Further, the same approach is used to calculate the RBP from this result. In mathematical terms, the RWT is derived as follows (Aker Solutions AS, 2017b):

$$WTSO[i, j] = WT[i, j] + SO[i, j] \quad (4.4)$$

$$WTSO_{median}[i, j] = \begin{cases} median(WTSO[:, j]), & \text{(column based)} \\ median(WTSO[i, :]), & \text{(row based)} \end{cases} \quad (4.5)$$

$$RWT_{median}[i, j] = WTSO_{median}[i, j] - SO[i, j] \quad (4.6)$$

- The 2D median filter is implemented using `scipy.ndimage.median_filter`, which is an image denoising tool from Scipy (The SciPy community, 2019). It creates a two-dimensional median filter, meaning that each element is replaced with the median of its neighbouring elements. The kernel size determines the number of neighbors that is included in the filtering of each element, and is set to $k = 3$ when using this filter. According to Aker Solutions AS (2017b), numerical tests show that it is not suitable for processing ILI data since it produces overly conservative results.

4.2.5 The Gaussian filter

The Gaussian filter is applied using the `scipy.ndimage.gaussian_filter`, which is a two-dimensional image denoising filter from Scipy (The SciPy community, 2019). The filter is implemented as a sequence of one-dimensional convoluting filters, which means that it uses the local neighbours multiplied with a Gaussian weight factor to calculate the value of each element. The kernel size decides the number of neighbours included in the calculation, and the distance weighting is decided by the standard deviation σ , which can be specified differently for each direction. Numerical tests show that using equal standard deviations in both directions gives the best results (Liu et al., 2018), and this is the default setting in the program. As default, σ is set to $0.5\sqrt{D \cdot t}$

The procedure for calculating the RWT-array is as follows (Liu et al., 2018):

$$WTSO[i, j] = WT[i, j] + SO[i, j] \quad (4.7)$$

$$WTSO_{Gaussian}[i, j] = ndimage.gaussian_filter(WTSO, \sigma_{axial}, \sigma_{circ}) \quad (4.8)$$

$$RWT_{Gaussian}[i, j] = WTSO_{Gaussian}[i, j] - SO[i, j] \quad (4.9)$$

where σ_{axial} and σ_{circ} denotes the kernel size in axial and circular direction, respectively. The RBP-calculations after filtering follows the same approach as the DNV-filter. An illustration of weight factors applying different linear filters are shown below in Figure 4.6, the Gaussian weighting applied here is item d.

4.2.6 RBP/MAV Calculation

From the RWT data, the RBP is calculated. It can be calculated using either raw WT data or data from one of the filters (RWT_{so} , RWT_{median} or $RWT_{Gaussian}$). For calculating the RBP, the minimum wall thickness for each point along the length direction is found by iterating over the

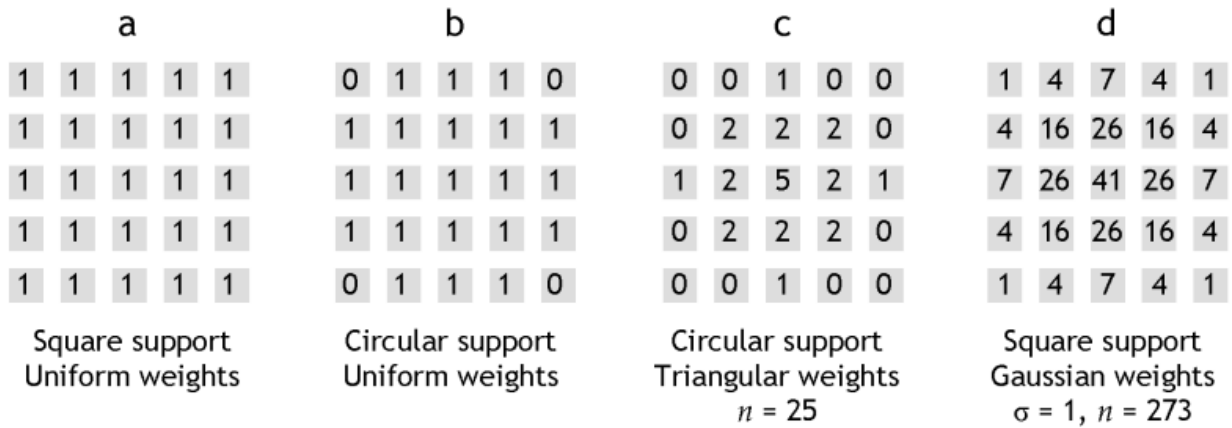


Figure 4.6: Weight factors applying linear filters for a kernel size of 5. Image is taken from Hall (2012). The item (d) denotes the Gaussian filter, which is applied here.

circumference. The circumference can either be narrowed to a user-defined interval, or the entire circumference may be taken into account.

After the RBP-file is created, the moving average profile (MAV) can be calculated. Here, the RBP is averaged over a length equal to $k\sqrt{D \cdot t}$, where $k = 0.5$. This is done using the convolution function in Numpy, *numpy.convolve*. Since it needs to average over a set number of elements, the start and endpoints are different. An illustration of the MAV-calculation is shown below.

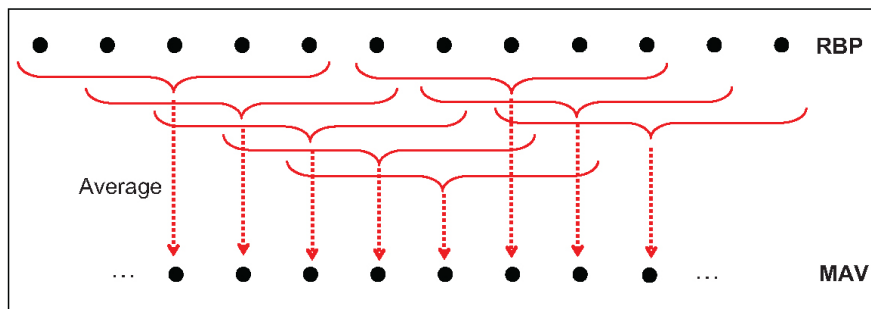


Figure 4.7: Illustration of MAV-calculation from a RBP. Image taken from DNV-GL (2019a).

The next step, according to the RP, is to perform capacity calculations. In corrPipe (Aker Solutions AS, 2017a), this is calculated using the RBP- and MAV-files as basis. This is briefly described in Section 3.3.4, but will not be considered further in this project.

4.3 Description of the program developed in this project

The program made in this project is built on parts of rbpGen by Aker Solution, reusing some of the codes applied there. As mentioned, rpgGen was written for Python 2.7, but Python has been significantly updated since, with increased speed and versatility. The transfer over to Python 3 means that some of the old program codes cannot be used directly, due to package changes within the Python environment. This means that many of the packages compatible with Python 2 are updated for Python 3 (e.g. Pyqt4 \rightarrow Pyqt5), while some packages are replaced by others, making parts of the codes unusable.

The work in this project does not concern itself with user-friendliness. More effort were put into processing power, programming convenience and speed instead. This meant that some changes were

implemented. The goal was not to directly transfer the program into Python 3, but instead, make use of the filtering framework in a program which can process large chunks of data with little user interference. It is also a goal to keep versatility and clarity in modelling, making it easy to implement changes and see results through the different steps of the program. Also, an alternative to the RBP-approach was developed, where defects are identified and gathered in lists. This method is described in detail later in this section.

To sum up, this section aims to describe the new program implementation in Python 3, changes made to the algorithm, justification for these changes and a description of the development attempted.

4.3.1 Packages and dependencies

All modelling in this program has been done using the Spyder IDE (The Spyder Website Contributors, 2018), which is suitable for scientific development in Python. For reading and structuring data throughout the algorithm, Pandas (McKinney, 2010) has been applied. Pandas is a Python library of data structures and statistical tools and is a tool suitable for data processing for scientific purposes. In this project, it was used for easing data manipulation before entering it into the filters. Since some of the filtering codes were reused either entirely or in modified form, it was convenient to define input and output of these as Numpy arrays, to save some time in the modelling phase and to keep the output of the different filters of the same data type. However, this may have caused the memory consumption of the program to go up compared to sticking with one data type throughout the entire algorithm. This solution was chosen as a compromise between easier modelling and speed.

Other packages that were used include Matplotlib for plotting, methods from Sklearn (Pedregosa et al., 2011) for the attempted treatment of missing data and imports from Scipy used in the filtering methods.

4.3.2 Program set-up and inputs

As mentioned, the GUI-based approach is inconvenient when investigating different methods; it is more a way of presenting a finished product. Instead, it was chosen to make a controller script, where all inputs are defined. Because this program is not intended to be used by others in its current form, this should be a good solution. The inputs that need to be defined are as follows:

- Project folder name - Folder containing all programs, and which also contains subfolders.
- Data folder name - Folder containing the .csv-files to be analysed (set as a subfolder of Project-folder in this implementation).
- Plot folder name - Also laid as a subfolder of the Project folder.
- Pipe dimensions - Outer diameter and thickness of the pipeline investigated.
- Filter parameters - av , N , l and k for the DNV-filter. The Gaussian filter has only variance, while the median filter uses the parameter k .
- `data_size` - Number of datasets that are to be analysed (mostly for development purposes). If it is not specified, all the datasets in the data-folder are analysed.
- `start_data` - The starting-point of the datasets. The files have names includes their odometer values, this parameter chooses the starting data in a sorted list of filenames. This enables analysis of a chunk of data, in the middle of the data-span.
- `fname` - Can be specified as "dnv_f", "gaussian_f" or "median_f". Specifies which of the filters that is to be used.

After these inputs are specified, the code can be run. Within the code itself, there are choices made that needs to be addressed directly in the code if changes are to be made. As this is a development project, these choices have not yet been automated.

4.3.3 Data filtering algorithm description and output

With input specified, the program algorithm for filtering the raw data works in the following way:

1. **Defining input data:** All filenames in the data folder are saved to two lists, one containing the WT-filenames and one with the SO-filenames. Using the *data_size* and *start_data* parameters, these lists are limited.
2. **Reading input:** The two lists are iterated over, reading one set of WT- and SO-data for each iteration. The raw data is loaded as Dataframes and structured in dictionaries. Each file is allocated in a specific Dataframe.
3. **Structuring output:** The size of the output will be dependent on the input shapes. For preserving data capacity, space is preallocated. The output data structure is displayed in Figure 4.8.
4. **Start iteration:** Iterates over the input dictionaries, treating one set of WT- and SO-data at each iteration:
 - Finds quality matrices, *Wt_q* and *So_q*, where 1 means missing data and 0 means data is present.
 - Replaces missing data with KNNImputer, with weighting set to *distance*, and the number of neighbours set to 7. A brief description of the KNNImputer can be found in Section 4.3.5.
 - Runs the filtering algorithm specified by the *fname* parameter. The filters themselves are for the most part unchanged from the previous version by Aker Solutions, and thorough description of these can be found in Sections 4.2.3, 4.2.4 and 4.2.5. The modifications of significance are described in Section 4.3.4. After filtering, the RBP and MAV are generated. The results are appended to the *result_dict* structure, which is a collection of the results obtained.
 - After filtering the dataset, the *RWT_SO* dataset is entered into the defect-lister algorithm, giving out a list of defects in the joint. The defect list is appended to a complete list of all joints considered. This algorithm is described in Section 4.3.6.
5. After the iteration is complete; two data structures contain the results obtained: *result_dict*, which contains the results shown in Figure 4.8, and *tot_defects*, which is a nested list of defects contained in each joint. The structure of *tot_defects* can be found in Figure 4.9.

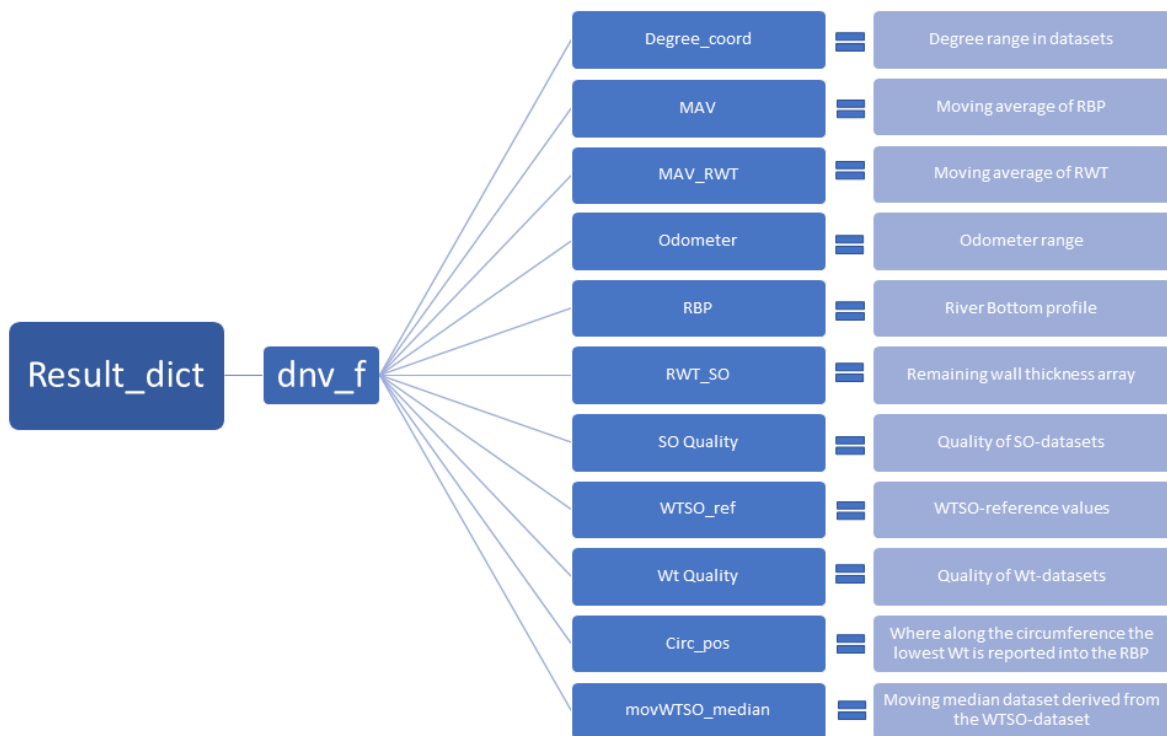


Figure 4.8: Illustration of the output data structure, here presented with the output from the DNV-filter. The data type is a nested dictionary, where the objects at the lowest level are Numpy arrays.

4.3.4 Changes specific to the data filtering

During the modelling phase, some changes have been done to the filters compared to the original program described in Section 4.2. These changes include:

- In the top layer, the input is changed from Numpy arrays to dataframes, because this data type works well with .csv-files. This meant some of the functions used throughout the filtering had to be adapted to suit the dataframes. These functions include *dnvrpf101*, *getref*, *getmovmedian*, *medianfilter* and *gaussianfilter*. The other functions are still using Numpy arrays, for these the input dataframes are transformed into Numpy arrays at the function call.
- As the function *getref* now takes input as a dataframe, it is easier (and faster) to use the built-in median calculation provided by Pandas, rather than using the original iteration.
- The replacement of zero-median-columns is removed. This problem is treated beforehand in the data preparation when missing data are replaced. This is discussed in Section 4.3.5.
- In the function *getref*, the lower and upper bound of *TmpWTSO* is established. Previously, this was found taking the minimum and maximum of the array *TmpWTSO_median*. This array was discovered to contain zeros due to preallocation, which could cause instabilities. The lower and upper bound was instead found from the array *WTSO_joint_ref*, which at the initiation time contains the same values as *TmpWTSO_median*, excluding the zero values.

During the work with the real datasets in this project, it became evident that the filtering is unstable when dealing with datasets where large parts of data are missing. The filtering methods gave unreasonable wall thickness for some of these cases, which again causes the capacity calculations to

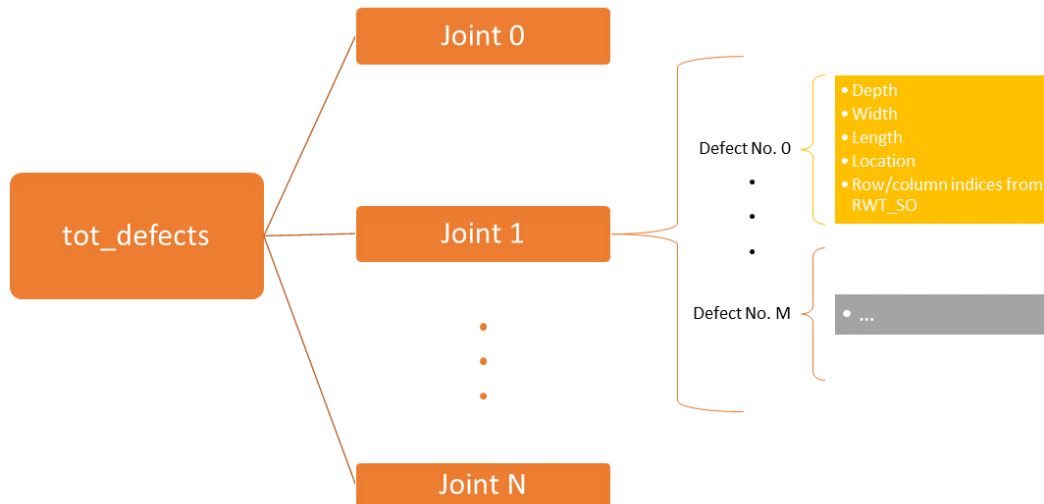


Figure 4.9: Illustration of the defect list data structure. The objects *tot_defects* and *Joint i* are lists, the defects are objects with properties as shown for the single defect no. 0 in joint no. 1 containing M defects. These properties are defined for each individual defect, and every joint contains its own list of such defects.

predict poor performance. The solution became to take out the minimum wall thickness from the raw datasets of each section, and add a condition after the filtering where all elements lower than the lowest observed section value are set to the section minimum. With this in place, the output after the filtering should make sense, even when the filtering has failed in making the dataset less conservative compared to the raw data.

4.3.5 Missing data treatment

Before the datasets are entered into the filters, some choices have been made regarding the missing data. Since the problem with missing data is considered different from noise and misreadings, it is treated before the "data filtering" part of the program.

In the RP from DNV-GL it is recommended to replace columns where the median is zero with the nearest column with a non-zero median. This is done in the same step as the rest of the data filtering, which may be causing some of the instabilities observed. It is also an approach that can mask bad datasets because it could give single columns more influence on the total result, removing important parts of the dataset. So, an attempt was made in this project to improve the treatment of missing data, and to disconnect it from the remainder of the data filtering.

What was used in the new program is the KNNimputer from Sklearn (Pedregosa et al., 2011). It is an imputer made for the treatment of machine learning datasets, where missing values might cause algorithms to fail. It is designed to keep features in the data and to use a nearest-neighbours-approach in estimating missing results. The imputer makes use of *n_neighbors* neighbours to estimate each missing data element. The neighbouring elements can either be averaged uniformly or weighted as a function of distance. A drawback when using this method is that it only uses the values from the same column as where the missing data is located, which may cause instabilities when entire columns have unreasonable results mixed with missing data. In case of entire missing columns, the

column is deleted, something that causes the rest of the program to break down. A condition was introduced to mitigate against this; when the KNNimputer fails, the missing column is interpolated from the neighbouring columns using the linear interpolator provided by pandas.

Combining these two approaches is possible as well, e.g. by choosing the interpolator when the columns are bad, and using the imputer otherwise. However, developing a condition for determining exactly when a column is "bad" has not been done in this project. So in this version, the interpolator is only used when the imputer breaks down.

4.3.6 Listing of defects

As an alternative to the RBP-approach from DNV-GL (DNV-GL, 2019a), which is reviewed in Section 3.3 and applied in rbpGen from Aker Solution, a method for identifying single defects within each joint has been developed. This is developed for use in the reliability analysis for the estimation of capacity. It analyses the dataset and classifies areas with wall thickness less than a nominal thickness value as defects, described by length, width and depth. The output is a list of all defects within the joint, with dimensions and location as properties. Applying this method makes it possible to take size and location of defects into account to a larger extent than when using the RBP-approach, where the "worst path" is assumed, and no information about the defect width is given. By looking at defects individually, this method is expected to yield slightly less conservative results than the RBP-method, along with giving possibilities to control the defect investigation to a larger degree. Defects with unreasonably low width can, for example, be mitigated against with this method.

The defect listing is done on the *RWT_SO* dataset, which is filtered wall thickness data. If the dataset is large, it should be split up as the iterations over the datasets are time-consuming. In the example provided in Section 5.2.1, for example, the dataset which covers 100 000 mm of data is split up in sections with length equal to 2000 mm. Different section lengths can give different results, so the program should be run using different specifications. The input to the function is also the nominal thickness. For the datasets analysed in this project, the best results were obtained by setting the nominal thickness to a "running mean" value, i.e. the mean thickness of the nearest sections.

The algorithm for listing defects for one joint is described below:

1. Inputs to be treated by the function are pipe wall thickness; RWT-array for the joint in question; and WT and SO-quality-matrices.
2. A matrix of defect depths is derived as $-(RWT - nom_thickness)$. All elements with a value smaller than 1 mm are set to zero, i.e. these parts of the pipe are considered undamaged.
3. Iteration is started over the defect depth matrix, defining each patch of a continuous defect as one single defect. The defects are saved to a list of defects, with their width, lengths and row/column-locations within the RWT-array as parameters. This means that complex-shaped defects are replaced with a regular-shaped estimation, with only a few parameters characterising the shape.
4. The list of defects is iterated over, finding overlapping/neighbouring defects. If found, they are combined, and one of them is deleted from the list. This iteration continues until the list is stable.
5. The defect depth matrix is treated using the quality matrices, for removal of extreme depths generated by the imputer. This can happen within large patches of missing data.
6. The maximum depth within each defect is found from the defect depth matrix. The maximum depth is taken as the mean of the $k = 0.5\sqrt{D \cdot t}$ largest depths within each defect, for some

averaging. If the defect is smaller than 3x3 elements, the depth is set to zero, and the defect is deleted.

7. The function returns the defect list for the joint in question. Each defect is characterised by length, width, depth and location as shown in Figure 4.9.

4.4 Example calculations estimating the reliability of the structure

A simplified reliability analysis is to be performed on the dataset. It is first executed using the current configuration and modelling the pressure as a random parameter following the Gumbel distribution, which is commonly applied to model annual maxima. Growth rates are then introduced into the problem, and the time-dependent reliability of the pipeline is estimated. Since there is only one dataset available, the growth parameters are estimated to values found in the literature. The basis is the defect list generated by the method described in Section 4.3.6.

The analysis was conducted using the brute Monte Carlo approach, as it is a method reasonably simple to set up and execute while providing good results if allowed to run for long enough time. The computational resources available in this project are, however, limited, so the accuracy of the analysis will be limited. The theory and specifics behind a MCS is described in Section 2.3.2. The limit state equations are based on the ones used in the literature examples (Wenxing Zhou, 2011, Wenxing Zhou, 2010), and on the burst equation from DNV-GL RP-F101.

4.4.1 Limit state formulation

The analysis is set up using a differentiation between the three limit states relevant for pipeline failure, i.e. small leak (sl), large leak (ll) and rupture (rp), where large leak and rupture are the two states a burst can develop into. These limit states are more extensively described in 3.4.1. Mathematically speaking, the limit states are set up as follows:

$$g_1 = t - d_{max} \quad (4.10)$$

where t denotes wall thickness; and d_{max} denotes maximum defect depth within a defect. The case of small leak is represented by $g_1 \leq 0$.

The burst limit state is given by:

$$g_2 = r_b - p \quad (4.11)$$

where r_b is the burst pressure resistance of the pipeline containing a part-through-wall defect, which will be a function of defect geometry and the pipe characteristics; and p is the internal pressure. For evaluating the burst pressure resistance, the burst capacity equation from DNV-GL RP F101 (DNV-GL, 2019a) was selected. It was established originally using the NG-18 (J. F. Kiefner, Maxey, et al., 1973) and B31G (ASME, 2012) equations. It is expressed as:

$$r_b = Y_{lab} Y_{FEA} P_{fit} \quad (4.12)$$

where

$$P_{fit} = P_0 R \text{ and } P_0 = Y_B \frac{2t \cdot UTS}{D - t} \quad (4.13)$$

where P_0 denotes the corrosion-free pipe capacity; R is the reduction factor due to corrosion; and the terms Y_{lab} and Y_{FEA} are model uncertainties given from comparing the predicted capacities with laboratory tests and finite element (FE) analyses, respectively, and are given by random parameters with distributions as shown in 4.3. The term Y_B accounts for the boundary conditions of the pipe, and is set to 1.08 for this analysis (assuming constrained pipe). The reduction factor, R , is given as:

$$R = \frac{1 - \frac{d_{max}}{t}}{1 - \frac{d_{max}}{t} \cdot \frac{1}{f_1(L/\sqrt{Dt})}} f_2(L/\sqrt{Dt}, t/D, d_{max}/t, \sigma_y/\sigma_u) \quad (4.14)$$

where f_1 is chosen as:

$$f_1 = 1 + 0.002X^3 \quad (4.15a)$$

$$X = \frac{L}{\sqrt{Dt}} \quad (4.15b)$$

and f_2 is calculated as:

$$f_2 = HQ \quad (4.16)$$

where H and Q are given in Equation 4.17 and 4.18, respectively:

$$H = \begin{cases} 1 - H_1H_2, & X \leq c_3 \\ 1, & X > c_3 \end{cases} \quad (4.17a)$$

$$H_1\left(\frac{t}{D}, \frac{d_{max}}{t}\right) = b_1 + b_2\frac{t}{D} + b_3\frac{d_{max}}{D} + b_4\frac{t^2}{D^2} \quad (4.17b)$$

$$H_2\left(X, \frac{d_{max}}{t}\right) = c_1X^{c_2}\left(1 - \frac{X}{c_3}\right)^{c_4} \cdot \left(1 + \frac{X}{c_3}\right) \cdot \left(\frac{d_{max}}{t}\right)^{c_5} \quad (4.17c)$$

$$Q = \begin{cases} 1 - Q_1Q_2, & X \leq e_3 \\ 1, & X > e_3 \end{cases} \quad (4.18a)$$

$$Q_1(B) = e_1B^2 + e_2B, \quad B = \frac{1.3\sigma_y}{UTS} - 1 \quad (4.18b)$$

$$Q_2\left(X, \frac{d_{max}}{t}\right) = X^{e_4}\left(1 - \frac{X}{e_3}\right)^{e_5} \cdot \left(1 + \frac{X}{e_3}\right)\left(\frac{d_{max}}{t}\right)^{e_6} \quad (4.18c)$$

the constants b_i , c_i , and e_i are given in Table 4.1.

Table 4.1: Constants in the burst capacity equations.

$b_1 = 1.6$	$c_1 = 0.4$	$e_1 = -0.5$
$b_2 = -3$	$c_2 = 1.15$	$e_2 = -1$
$b_3 = -2$	$c_3 = 25.5$	$e_3 = 16$
$b_4 = 2$	$c_4 = 8$	$e_4 = 0.5$
	$c_5 = 1.5$	$e_5 = 5$
		$e_6 = 1$

A burst can develop into a rupture if certain conditions is met. Here, the rupture limit state is defined as:

$$g_3 = r_{rp} - p \quad (4.19)$$

where r_{rp} is the rupture pressure resistance of a pipeline containing a through-wall defect resulting from a burst. This defect will develop axial instability if $g_3 \leq 0$. The rupture resistance model was suggested by CSA (2007) and Nessim et al. (2009), based on a flow-stress dependent failure criterion developed by J. F. Kiefner, Maxey, et al. (1973). The rupture pressure resistance is given by:

$$r_{rp} = \frac{2t\sigma_f}{MD} \quad (4.20a)$$

$$M = \begin{cases} \sqrt{1 + 0.6275\frac{l^2}{Dt} - 0.003375\frac{l^4}{Dt}}, & \frac{l^2}{Dt} \leq 50 \\ 3.293 + 0.032\frac{l^2}{Dt}, & \frac{l^2}{D^2t^2} \geq 50 \end{cases} \quad (4.20b)$$

where M is the Folias factor, and σ_f is the flow stress defined as $0.9\sigma_u$. The model error in the rupture model was ignored for this analysis.

When the growth rates are introduced, the limit state functions will be functions of time. They depend on the pressure, in addition to the lengths and depths of the defects, which all are time-dependent.

4.4.2 Random Parameters

The random parameters are taking observational variance and variance in pipeline geometry and material into account. The observational accuracy is specified in the report by the inspection company (NDT Global, 2020), where the error margin for 90% certainty is given. The 90% confidence transferred to standard deviations gives a value of 1.644854σ . The mean will be set equal to the depth of each defect for simplicity. An overview of the defect parameters are given in Table 4.2.

Table 4.2: Distribution of parameters for the observed defects, with error margins specified from the inspection company.

Parameter	Symbol	Accuracy at 90% certainty	Distribution	σ
Defect maximum depth	d_{max}	± 0.6 mm	Normal	0.365
Defect length	l	± 1.5 mm	Normal	0.912
Wall thickness	t	± 0.6 mm	Normal	0.365

The pipe material parameters are taken from the inspection report, where the pipeline information is listed. The steel grade is API 5L X65, the material info is found from PM International Suppliers (2020). The distribution of the ultimate strength is specified in the RP (DNV-GL, 2019a) as normally distributed, with a coefficient of variance (CoV) of 6%. With all defects assumed independent, the material properties are assumed independent as well, even though it would be more correct to take spatial correlation into account. This is done for simplicity. The annual maximum pressure distribution is found from DNV-GL (2019a), where it is assumed to follow a Gumbel distribution with mean equal to $1.05 \cdot p_d$, where p_d denotes the design pressure.

The growth rates are taken as linear growth rates from the literature, as only one dataset is available. The growth rates of the defect depths are estimated from CSA (2007), where growth rates between 0.033 and 0.40 mm/year are indicated. They are assumed to follow a lognormal distribution fit between these values. The growth rates in the length direction are arbitrarily assumed and assumed to follow a lognormal distribution as well. With the spatial distribution of defects ignored, the possibility of defects growing together is neglected. The growth rates are set as a specific value to each of the defects in the initial stage, as to represent the pipeline characteristics in that spot.

Table 4.3: Distribution of material parameters and pipeline characteristics.

Parameter	Symbol	Distribution	Mean	CoV
Internal pressure	p	Deterministic (Gumbel)	$p_d = 13.8$ MPa	3.0 %
Diameter	D	Deterministic	304.8 mm	-
Ultimate tensile strength	σ_u	Normal	578.8 MPa	6%
Yield strength	σ_y	Normal	483.8 MPa	8%
Model error FEA	Y_{FEA}	Normal	1.0	2%
Model error lab	Y_{lab}	Normal	1.0	8%
Defect length growth rate	g_l	Lognormal	3.5 mm/yr	50%
Defect depth growth rate	g_d	Lognormal	0.20 mm/yr	75%

The tensile strengths are calculated from specified minimum yield strength (SMYS) and specified minimum ultimate strength (SMUS) as $\sigma_Y = 1.08 \cdot SMYS$ and $\sigma_u = 1.09 \cdot SMUS$.

4.4.3 Reliability analysis set-up

The current defect depth is set as a random parameter with mean equal to the observed depth, and with a variation that represents the measurement variation. As mentioned, the depths from the current configuration will be subjected to random growth rates each year, so each defect decreases with an increment for each year. The pipeline material and the geometric characteristics of the pipeline are also set as random parameters but specified in the initial stage. The design pressure of the pipeline is 138 bar, so the mean of the Gumbel distribution is set to this value.

The defects are found using the tool specified in Section 4.3.6. The defects are checked independently, and the annual probability of failure of the pipeline section is calculated by looking at the pipeline as a series system, where failure at one defect location means failure of the entire section. The pressure distribution over the pipeline section is assumed constant, i.e. that the pressure at different defects is fully correlated. The failure probabilities of small leak, large leak and rupture are considered mutually exclusive so that they can be calculated independently of each other.

The probability of failure is estimated as:

$$p_f = \frac{N_f}{N_{cycles}} \quad (4.21)$$

where p_f is the probability of failure; N_f is number of series system that have failed and N_{cycles} is the total number of series systems generated.

5 Results and specific discussion of results

5.1 Verification of the new filtering program

The first stage of the project consisted of transferring the algorithm to Python 3 and adjusting the algorithm slightly while still getting comparable results. The transfer is verified checking the contours and the RBP-plots of the original program compared to the modified one. This is done by using two test datasets, and two examples of real inspection data. The filtering is done using the DNV-GL algorithm, and the results are shown in this section.

5.1.1 Fabricated test files

The fabricated test files are two sets of data with a nominal thickness of 17.5mm, with amounts of noise added, and with a corrosion depth of 8.75 mm specified. The difference between the datasets is that *File 1* is complete, while *File 2* has had four columns of data removed. Hence, this verification shows how the filtering program deals with missing columns. The programs are expected to treat missing data slightly differently, but the filtering on good quality datasets should yield comparable results.

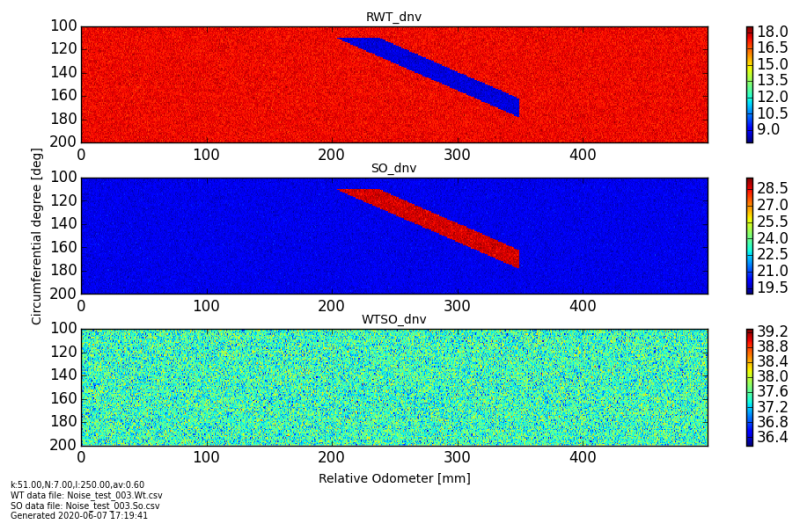


Figure 5.1: Contour of the dataset with no missing data, made using rbpGen.

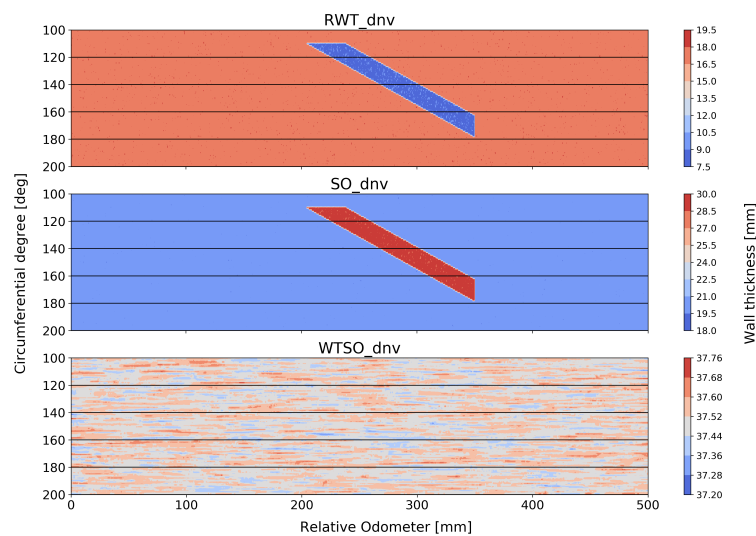
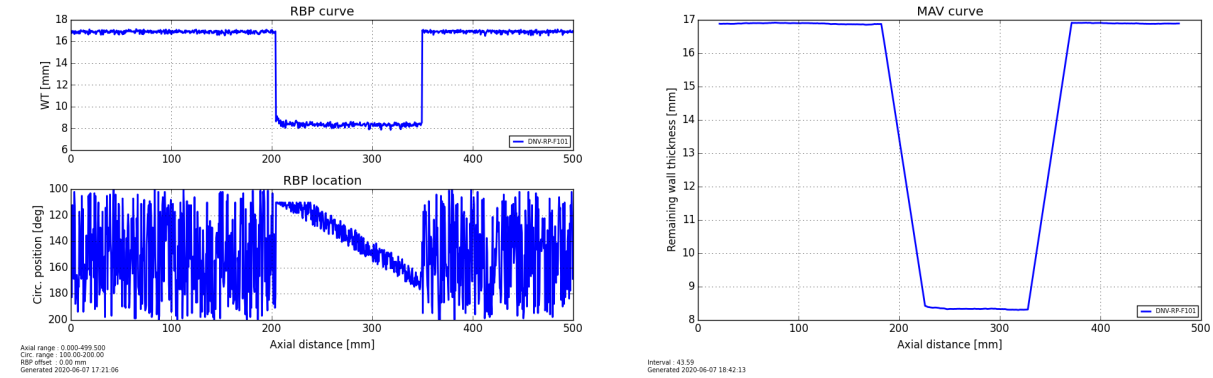


Figure 5.2: Contour of the dataset with no missing data, made using the new program.

By comparing the figures 5.1 and 5.2, it can be seen that the results are close to equal. The differences in programs are largest on the treatment of missing data, and this shows that the filtering method works as intended when the datasets are complete.

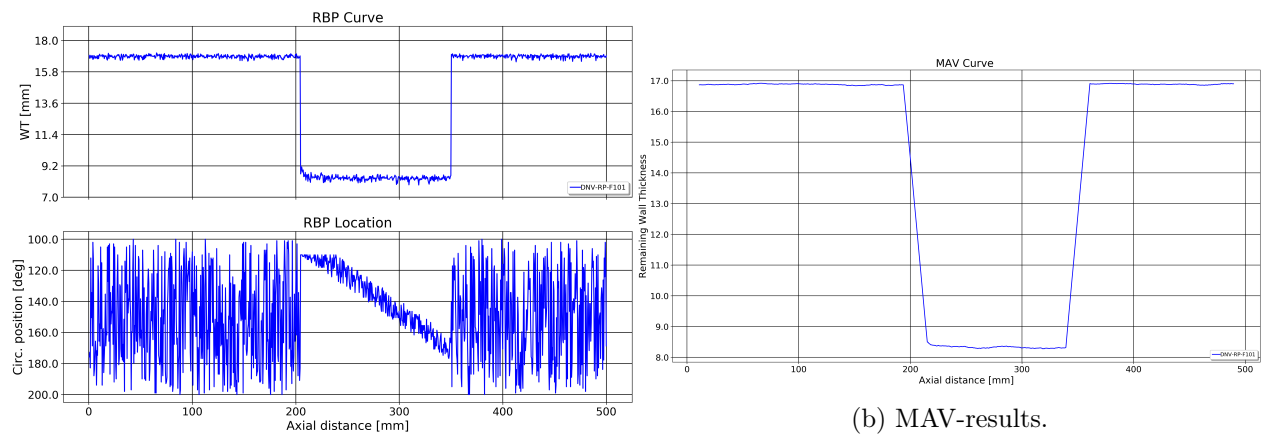
The RBP- and MAV-files from this run are shown below.



(a) RBP-results.

(b) MAV-results.

Figure 5.3: Results from File 1 using rbpGen



(a) RBP-results.

(b) MAV-results.

Figure 5.4: Results from File 1 using the new program.

As the numerical results from the RBP- and MAV-files shown in Figure 5.3 and 5.4 suggest, the noise filtering seem to work in the same way.

The contours produced when analysing File 2 with some columns missing are shown below:

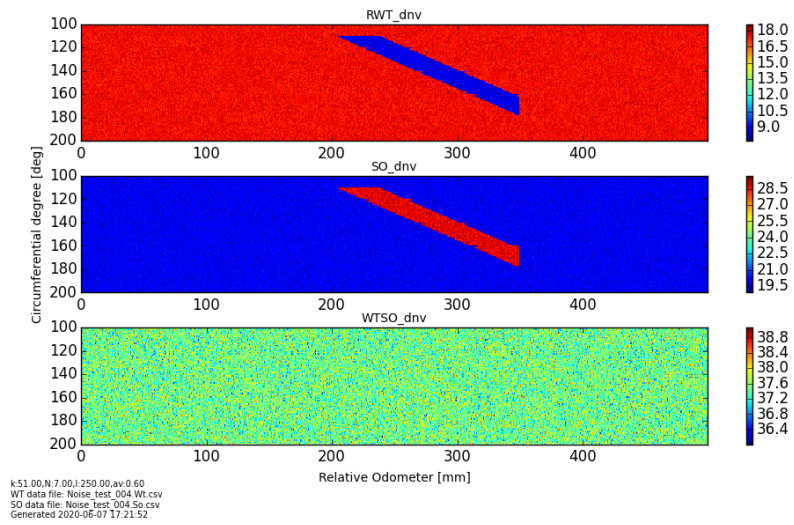


Figure 5.5: Contour of the dataset with four columns missing, made using rbpGen.

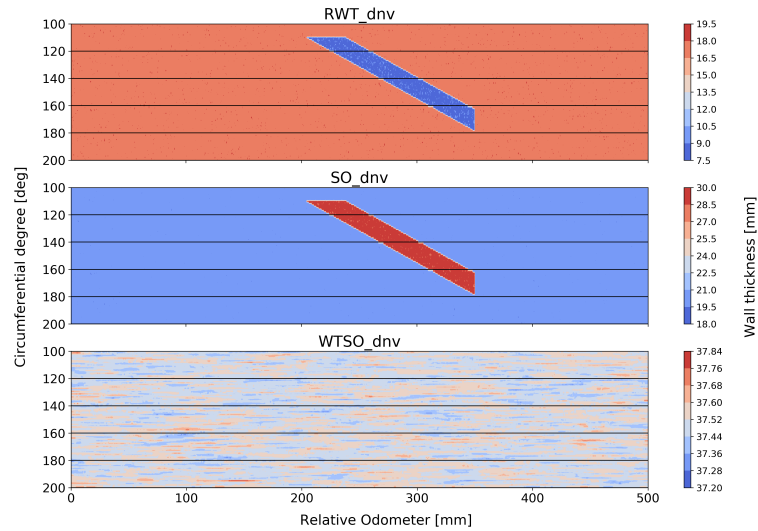
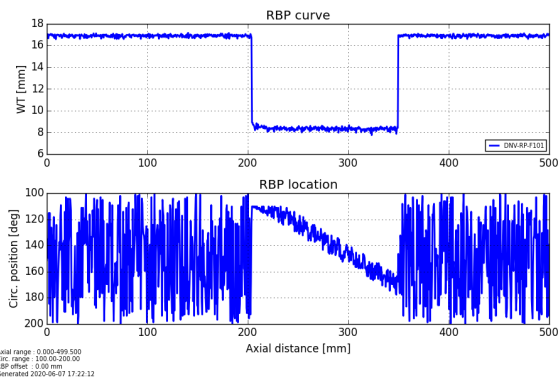


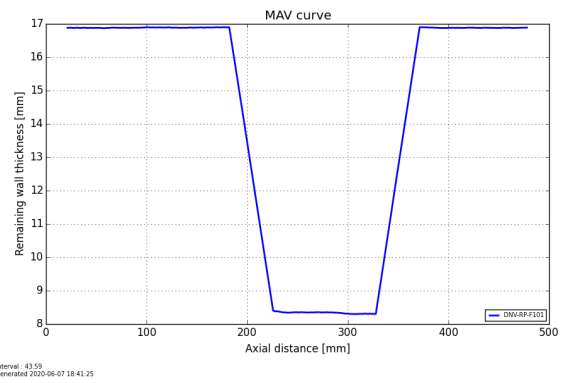
Figure 5.6: Contour of the dataset with four columns missing, made using the new program.

The contours produced seem equal, the differences in colour palettes and in label levels should account for most of the differences. This suggests that the results are comparable.

Looking at the RBP- and MAV-plots from the fabricated test files:

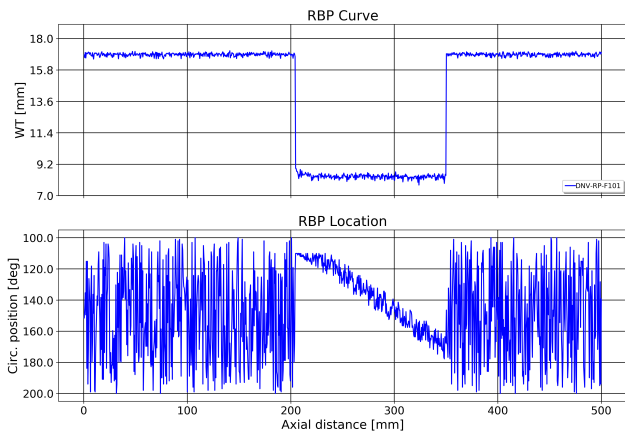


(a) RBP-results.

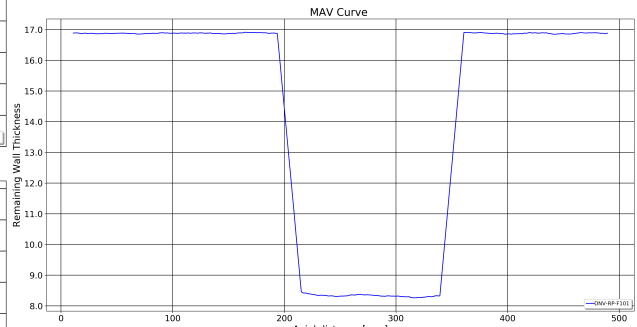


(b) MAV-results.

Figure 5.7: Results from File 1 using rbpGen



(a) RBP-results.



(b) MAV-results.

Figure 5.8: Results from File 2 using the new program.

These results also indicate that both algorithms are capable of treating missing columns if the overall quality in the rest of the dataset is good. It also indicates that the program made in Python 3 produces results comparable to the implementation in Python 2.

5.1.2 Real test files: Data analysis of three real sections

The three sections chosen are taken from a generated dataset from an ILI inspection on a pipeline, where the outer diameter is 304.8 mm, and the nominal wall thickness varies along the length. The pipeline is heavily corroded, and the datasets have many imperfections and chunks of missing data. This comparison is performed to show how the different algorithms treat missing data and fluctuations of more random nature. Two sections were initially chosen, but when it was discovered that the data quality was rather poor for both a third section was added.

The new data filtering algorithm contains a "fail-safe" explained in Section 4.3.4. In short, it limits the remaining wall thickness to the lowest observed value from the raw dataset. It is disabled for this part, as the filtering methods are better compared without this feature.

The section datasets are randomly picked, the first starts at $x = 3000$ mm along the pipeline, the second at $x = 20000$ mm and the third at $x = 40000$ mm. The length of all sections are 1000 mm. They have varying quality along the length, as shown in Figure 5.9 below:

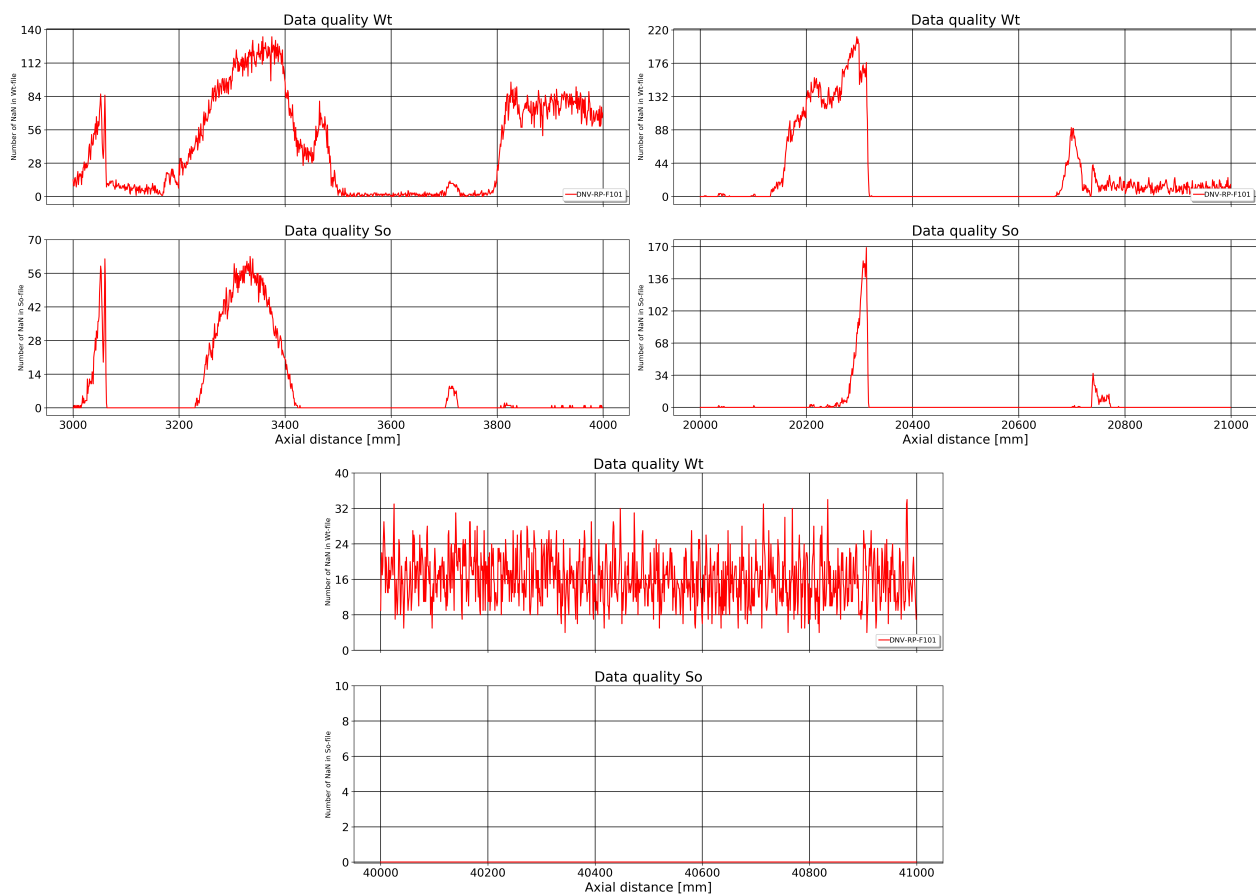


Figure 5.9: Quality-plots for the test-sections. The value on the y-axis is the number of missing datapoints per mm Odometer value.

Where the fabricated data just had some missing columns along the entire length, these datasets contain significantly higher amounts of both missing data and noise. This is expected to have influence on the results, and also to show some differences between the programs.

5.1.3 Real test file: Section 1

The results presented here are from the first section tested, which lies within the odometer interval [3000, 4000] mm. The wall thickness of this analysis is set to 25 mm. First, the contours are plotted:

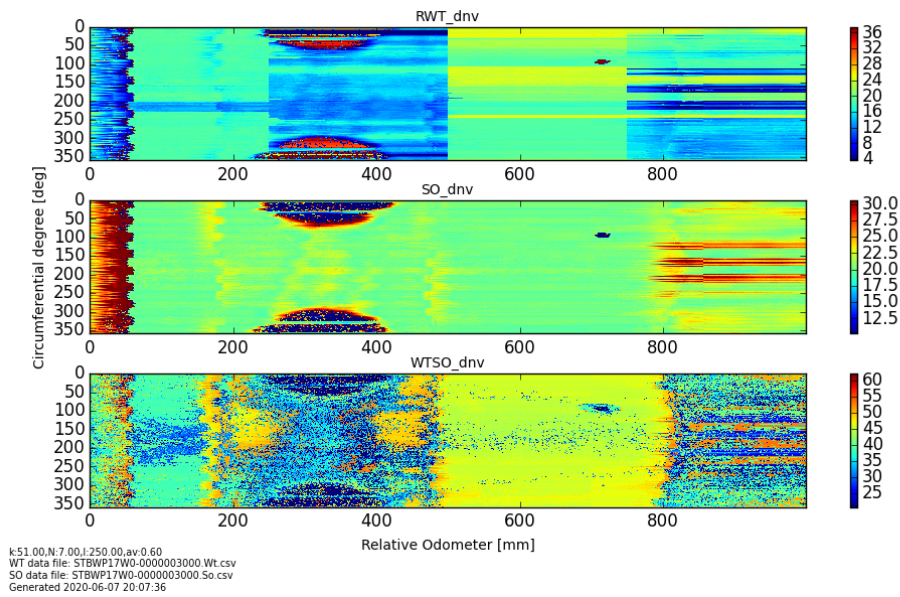


Figure 5.10: Contour from rbpGen of the section between 3000 and 4000 mm.

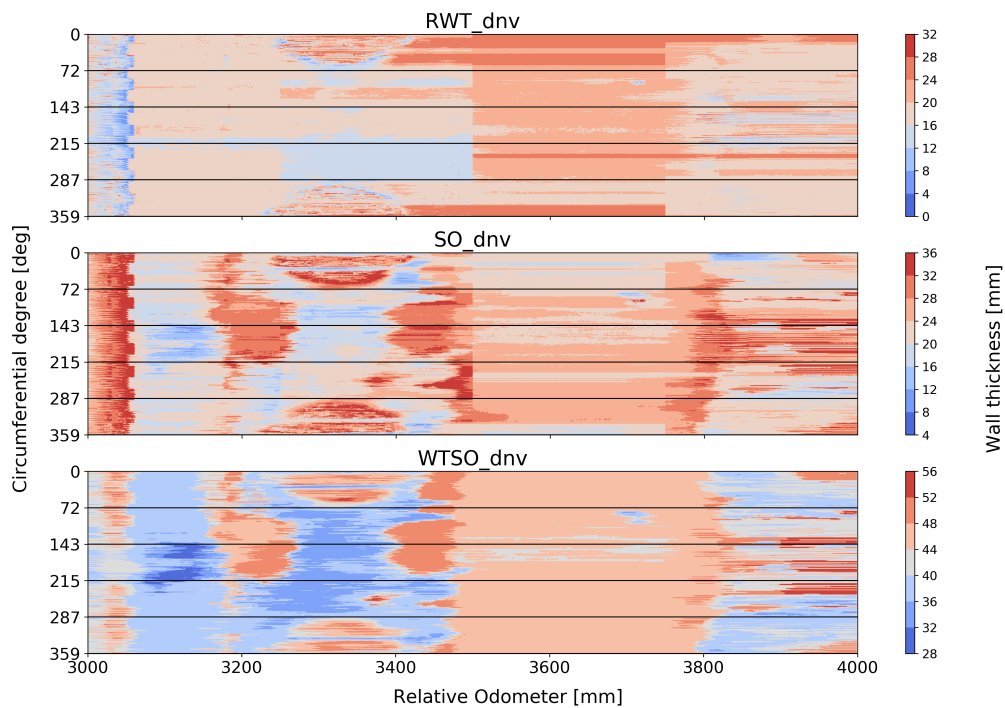


Figure 5.11: Contour from the new program, of the section between 3000 and 4000 mm.

The patterns in the contours are kept by both implementations. It can be noted that the contours from rbpGen have a lower limit of 4 mm while the new program goes all the way down to zero. This is due to the specifications of the plot and not the underlying data.

The RBP- and MAV-plots from rbpGen are plotted for this section as well and are shown in the figures below.

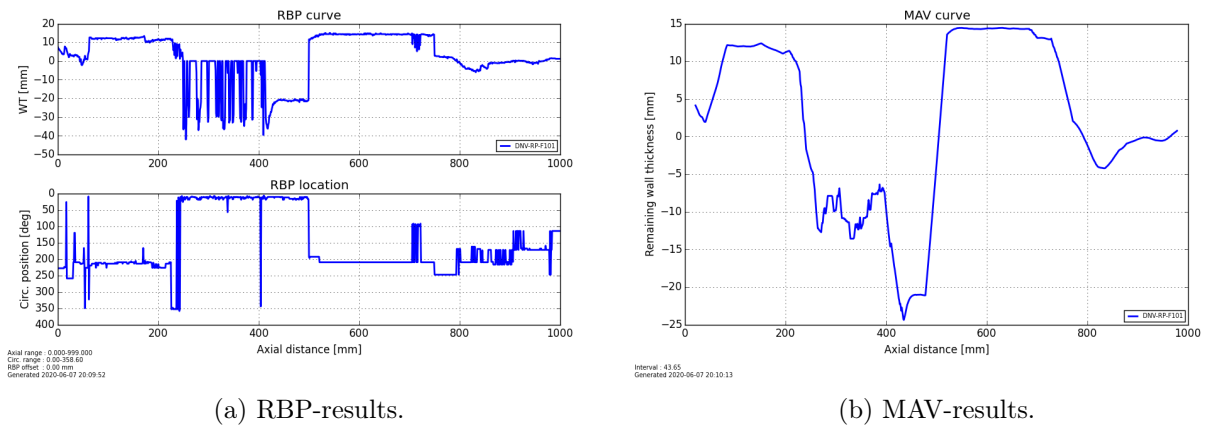


Figure 5.12: Results of the section between 3000 and 4000 mm using rbpGen.

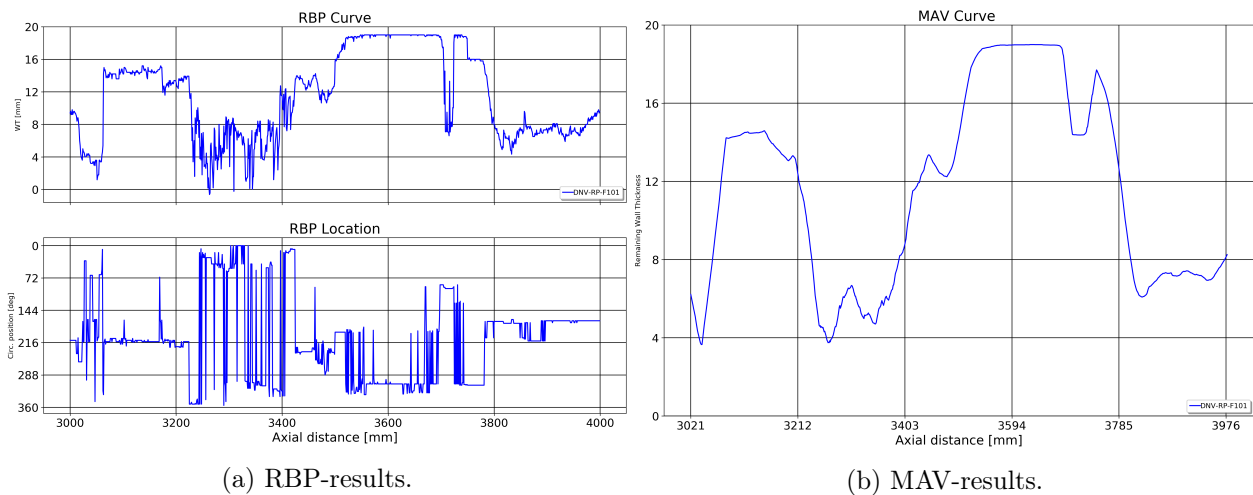


Figure 5.13: Results of the section between 3000 and 4000 mm using the new program.

As we can see from the quality plots found in Figure 5.9, the missing data in this dataset has a peak between 3200 and 3400 mm, which coincides with the seemingly largest problems in the RBPs. So these problems must be kept in mind when looking at the results. However, when comparing the RBP- and MAV-results from the new implementation in Figure 5.13 to the ones generated from rbpGen in Figure 5.12, it can be seen that the new version gives results that are more feasible than the original one. There are 255 data points at each odometer value, so about half the data in this area is missing. So how reliable these generated results are is questionable, and this uncertainty should be taken into account in capacity calculations.

5.1.4 Real test file: Section 2

The results presented here are from the second section tested, which lies within the odometer interval [20000, 21000] mm. The wall thickness of this analysis is set to 17.6 mm. First, the contours are plotted:

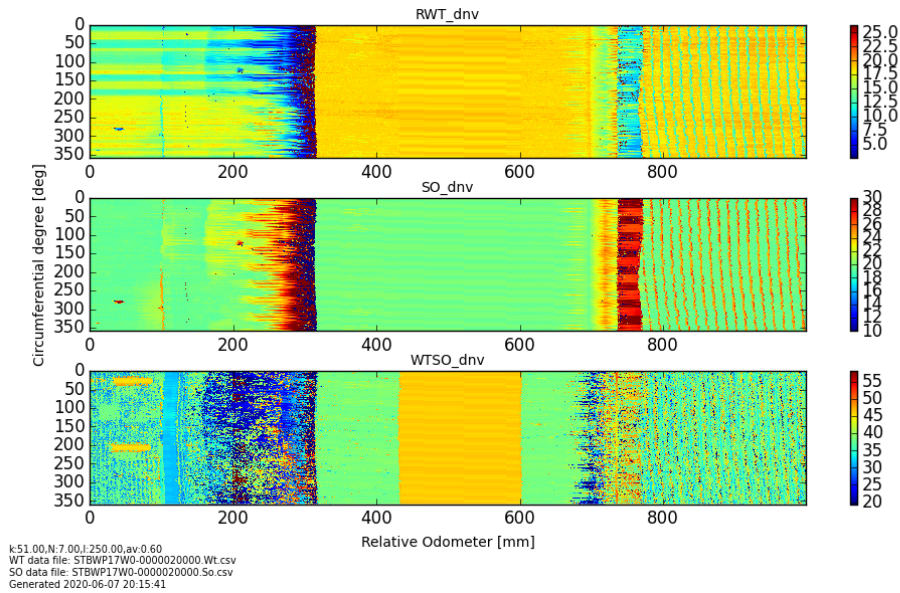


Figure 5.14: Contour from rbpGen of the section between 20000 and 21000 mm.

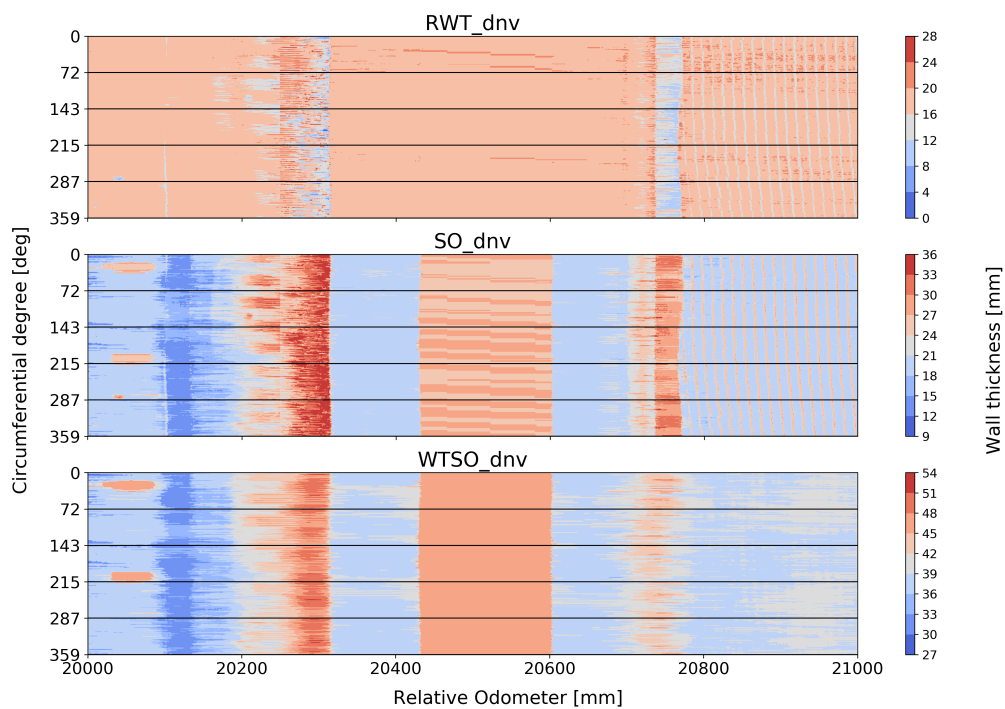
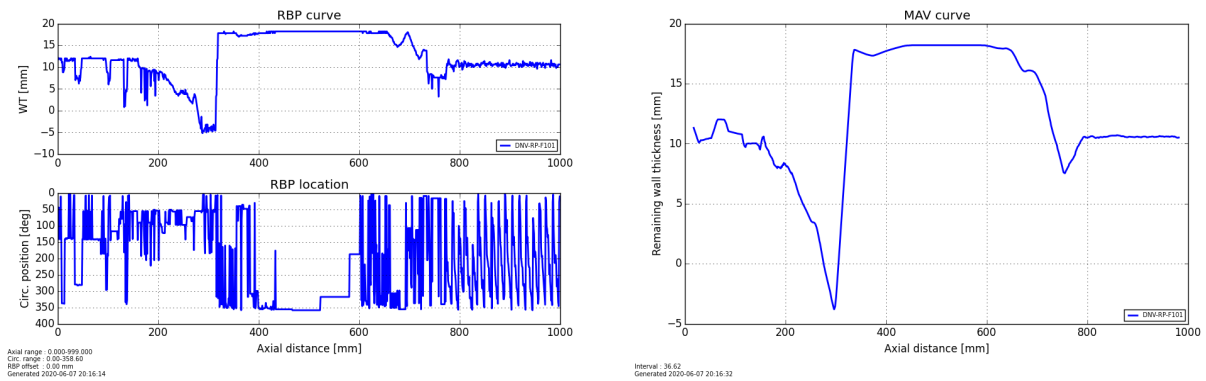


Figure 5.15: Contour from the new program, of the section between 20000 and 21000 mm.

As for the previous case, the patterns seem to be captured by both contours, though the plot from rbpGen in Figure 5.14 seem more subjected to noise compared to the new implementation in Figure 5.15. This may, however, partly be caused by the different colour palettes chosen, as rbpGen has a more varied colour span.

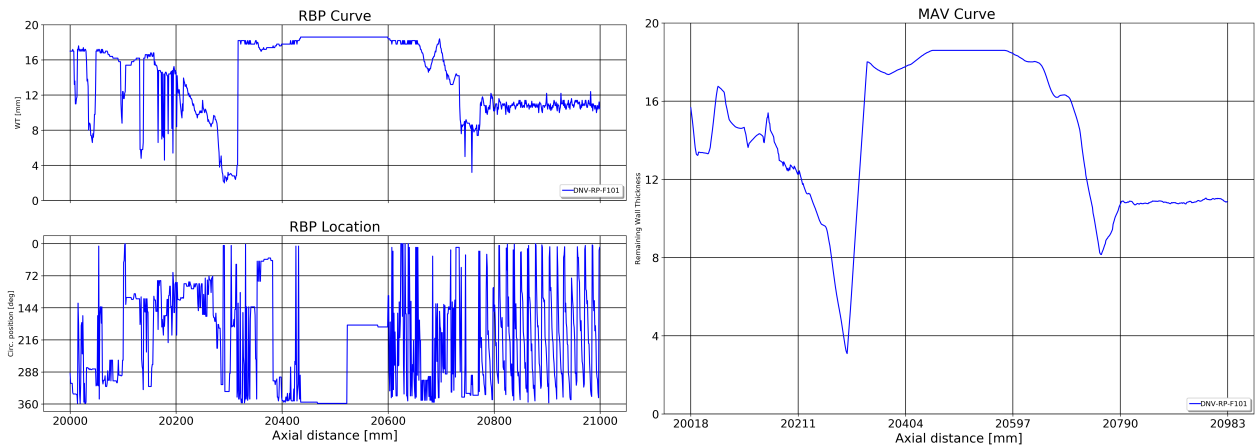
The RBP- and MAV-plots from both the programs are plotted and are shown in the figures below.



(a) RBP-results.

(b) MAV-results.

Figure 5.16: Results of the section between 20000 and 21000 mm using rbpGen.



(a) RBP-results.

(b) MAV-results.

Figure 5.17: Results of the section between 20000 and 21000 mm using the new program.

These results seem more comparable, and both the RBP- and MAV-plots share the overall shape. Nevertheless, the extreme point in the RBP and in the MAV created by rbpGen (Figure 5.16) has a negative WT-value, which is unfeasible for further calculations. The shape and location of the dip are also present in Figure 5.17 made by the new program, but here the values are positive.

5.1.5 Real test file: Section 3

The results presented here are from the third section tested, which lies within the odometer interval [40000, 41000] mm. The wall thickness of this analysis is set to 17.6 mm, as this seem to be the dominating value within this area. First, the contours are plotted:

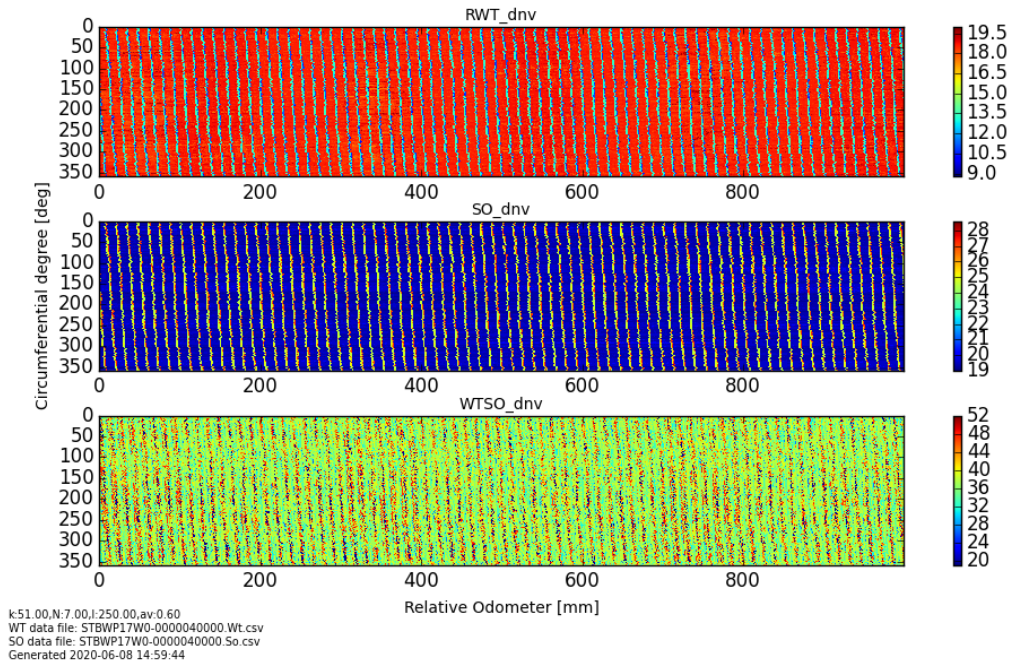


Figure 5.18: Contour from rbpGen of the section between 40000 and 41000 mm.

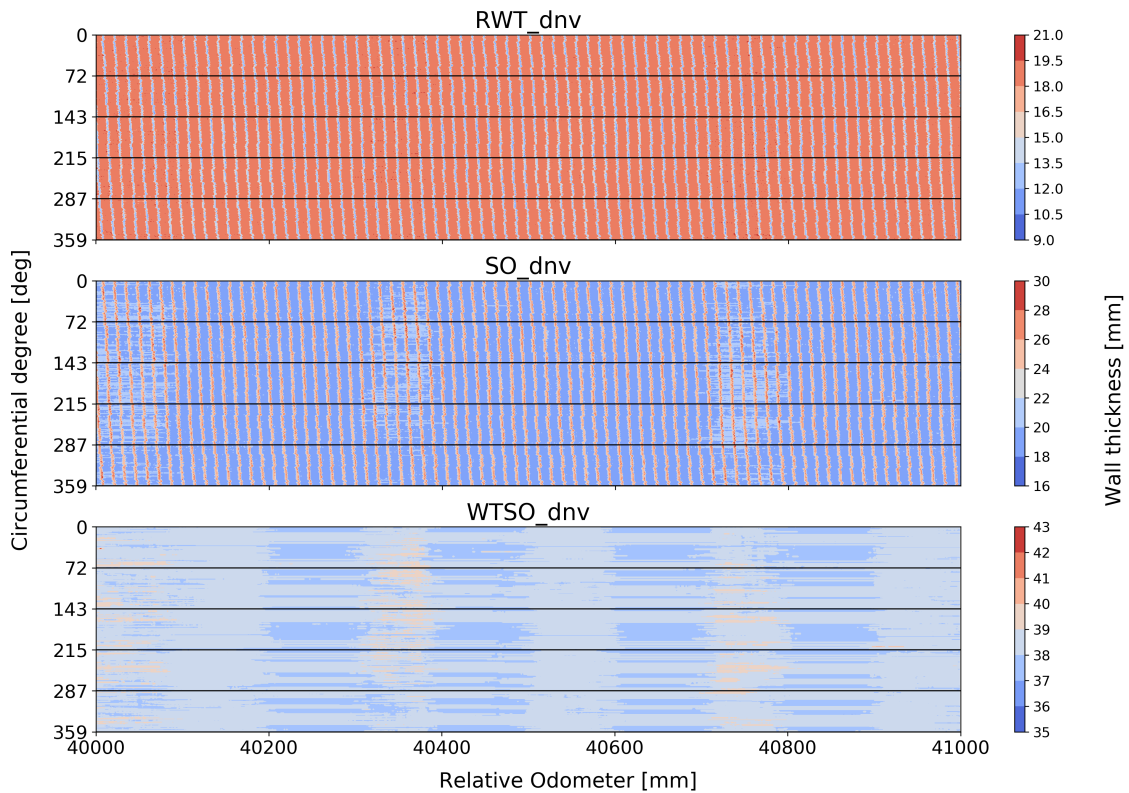
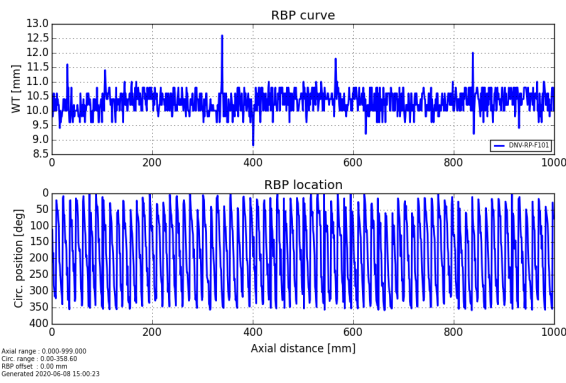
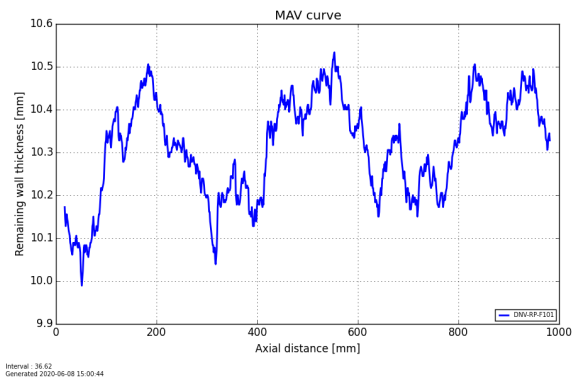


Figure 5.19: Contour from the new program, of the section between 40000 and 41000 mm.

The RBP- and MAV-plots are plotted for this section as well and are shown in the figures below.

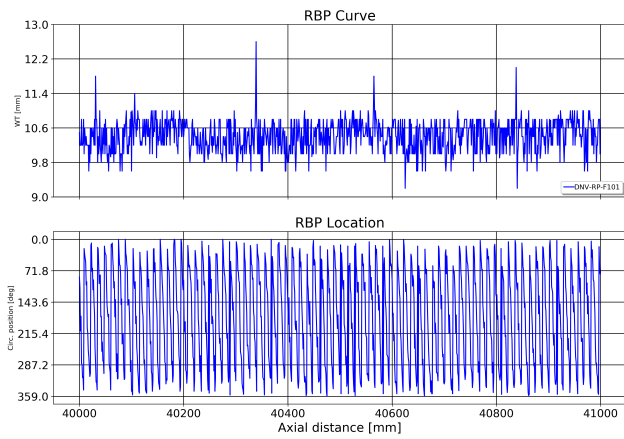


(a) RBP-results.

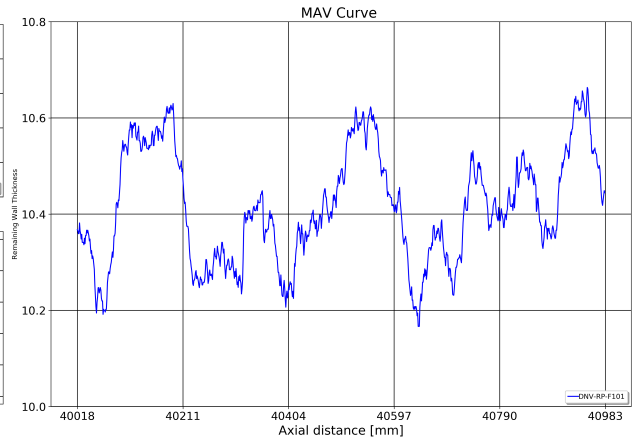


(b) MAV-results.

Figure 5.20: Results of the section between 40000 and 41000 mm using rbpGen.



(a) RBP-results.



(b) MAV-results.

Figure 5.21: Results of the section between 40000 and 41000 mm using the new program.

The results for this section are close to each other and seem to correspond well. It is worth noticing that the results from rbpGen in Figure 5.20 have slightly lower mean compared to the results in Figure 5.21 generated from the new program. This suggests that the new program has slightly lower conservatism on its results.

5.2 An overview analysis of the pipeline dataset

Using the new program implementation, a real case dataset is to be investigated. The dataset consists of 100 sets of WT- and SO-data, each representing a length of 1000 mm along the pipeline. This means that the total length of this pipeline section is 100 m, with a resolution of 1 mm. Now, the feature of limiting the minimum remaining wall thickness to the lowest observed value is used as well, which makes the datasets possible to analyse further.

The entire dataset has been filtered by the new program. The filtering is done section-wise, with section lengths of 1000 mm. The quality plots and RBP-plots will be continuous over the length of the dataset. For the MAV-plot, the results are plotted against the Odometer values, even though they really are discontinuous, lacking a few elements at the ends of each section. This leads to a lack of data at the ends of the MAV-plot. This is, however, not considered a major problem as the values and trends should still be valid. The averaging is dependent on the wall thickness; it is set to a constant value of 25 mm throughout the entire filtering process for simplicity. The resulting quality plots, along with the RBP- and MAV-plots are found in figures 5.22, 5.23 and 5.24 below.

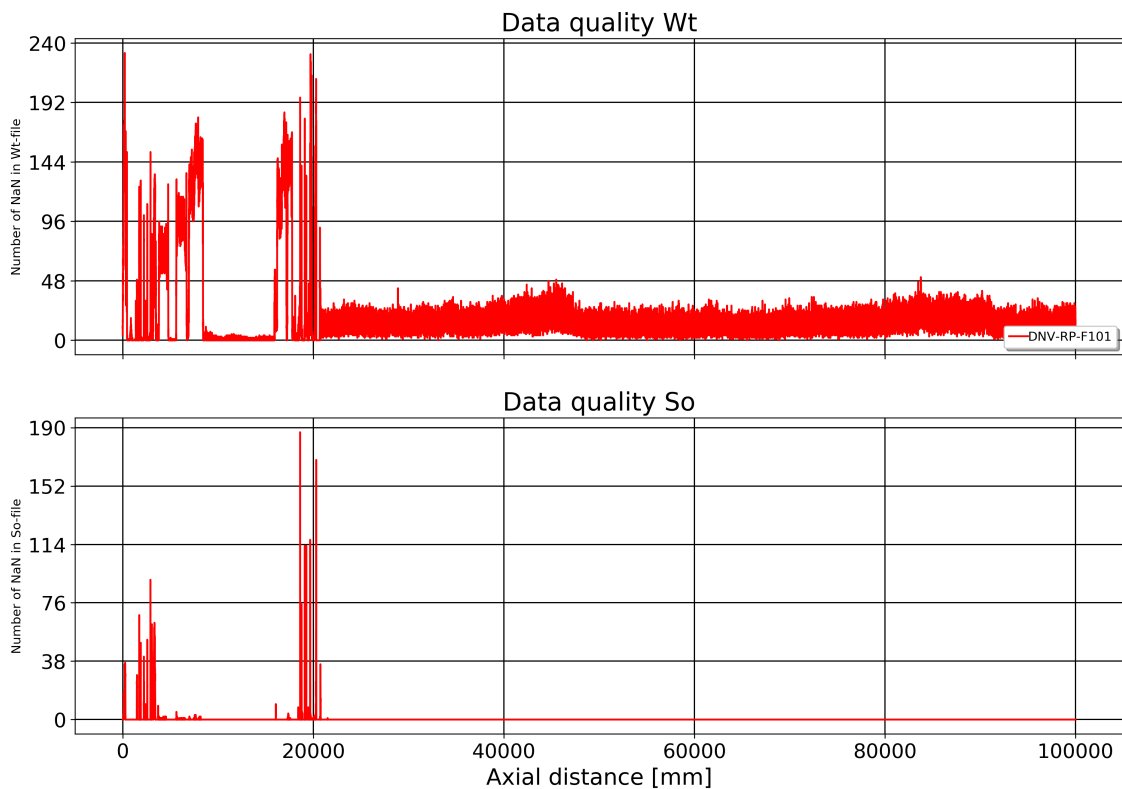


Figure 5.22: Quality plot of the entire dataset. Y-axis denotes the number of missing data per mm Odometer value.

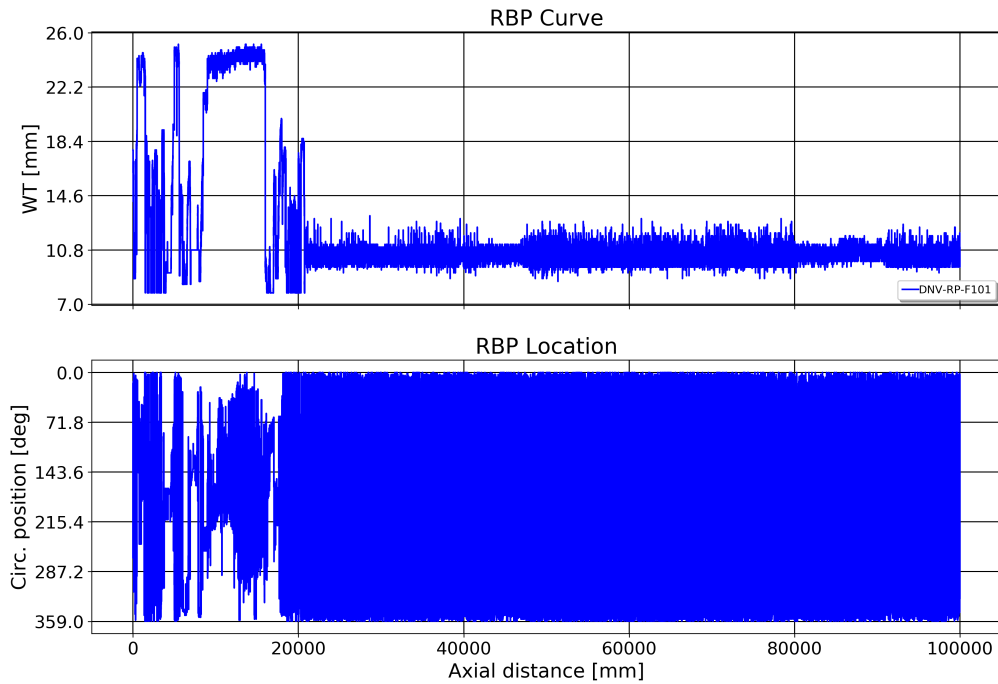


Figure 5.23: Overview of the RBP for the entire dataset.

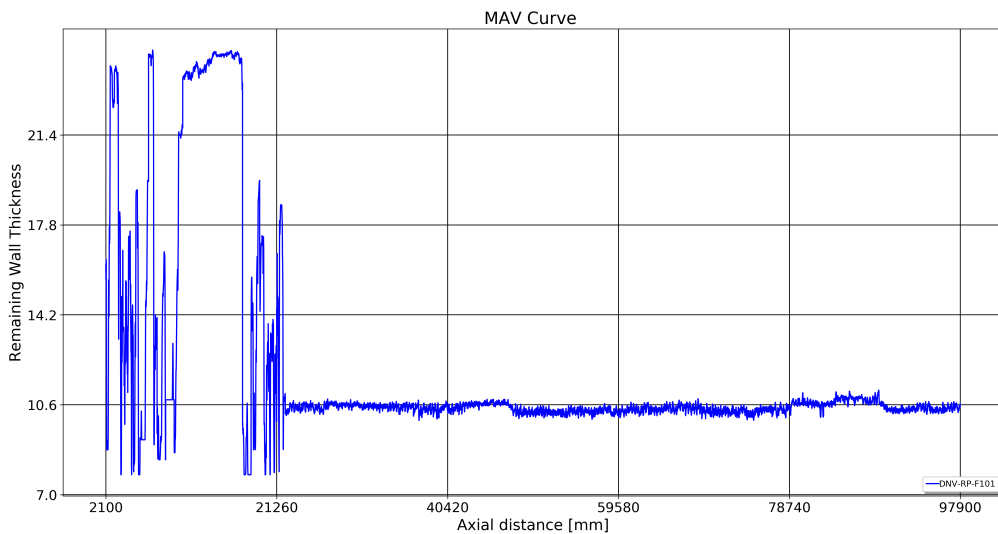


Figure 5.24: Overview of the MAV of the entire dataset. The MAV is made using a constant averaging value of $k = 0.5\sqrt{D \cdot 25}$ mm. The axial distance cannot be taken as absolute values, but it gives an approximate location.

The results from the quality overview in Figure 5.22 show that the dataset has problems until about 23000 odometer value, from there on the dataset has decent quality. The condition on the predicted wall thickness limits the defect depth downwards, so that the heavy fluctuations that can be seen in Figure 5.15, e.g., have been limited as well.

5.2.1 Defect Listing results

The defect listing program was run on the entire dataset. The defects are defined as rectangular areas with a constant depth, which is a coarse simplification. A challenge was that when using the program on the corroded sections from an odometer value of 25000 and onward, all the sections from there and out were each classified as one defect. This is due to the defect shape that can be observed in the contours from Figure 5.19, e.g., where the defects are going in oblique stripes, non-parallel to the edges of the plot. The Defect Lister iterates and defines overlapping defects as one defect, which makes it model the entire section as corroded. The contour plot of the entire dataset length can be found in Figure B.1 in the Appendix.

In total, the program has found 99 defects along the length, though many of these are entire sections that are classified as one defect. Different section formulation will give some variation in this result.

In Figure 5.25 an overview of the grouping of the different defects is shown. The listing is performed by dividing the entire dataset into sections with length equal to 2000 mm, which in part creates the section pattern seen in the figure.

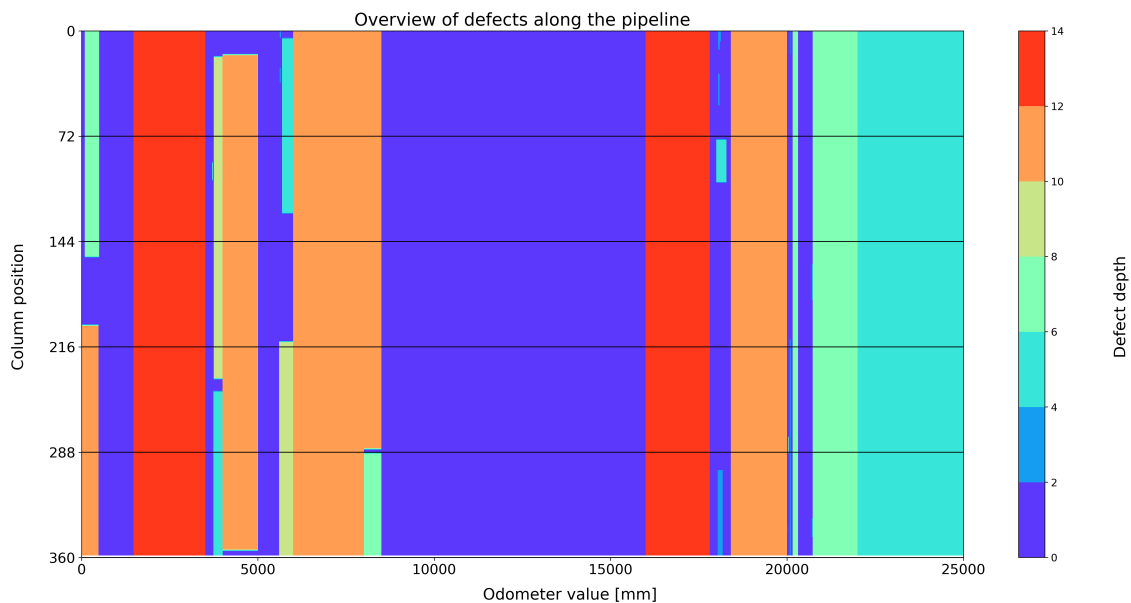


Figure 5.25: Grouping of defects from the defect listing in the odometer interval $[0, 25000]$ mm.

As can be seen, many of the sections are classified as one continuous defect, but smaller patches of defects can also be seen. With this section specification, the most severe defect can be found between 1482 mm and 1999 mm, which has a depth value of 13.2 mm and a remaining wall thickness of 7.80 mm. The point with the most severe defects can also be seen in the MAV-curve from Figure 5.24, where it can be seen to coincide with the heavy fluctuations in remaining thickness in the first part of the plot.

The most severe defects along the length are listed in Table 5.1, where the defects with a depth over 5mm are listed.

Table 5.1: Overview of the most severe defects along the pipeline, i.e. with depth larger than 5 mm. RWT is the remaining wall thickness.

Defect Maximum Depth [mm]	RWT [mm]	From x = [mm]	To x = [mm]
7.44	13.6	95	498
13.2	7.80	1482	1999
10.5	10.5	0	477
12.6	7.82	2000	3511
8.54	11.9	3752	3999
5.24	15.2	3741	3999
10.9	10.2	4000	4998
8.90	12.2	5606	5999
11.3	10.3	6000	7999
10.2	12.9	8000	8501
7.85	15.3	8000	8500
12.9	7.82	16000	17808
11.3	8.00	18413	19999
7.00	10.4	20163	20317
7.61	9.84	20735	21999
6.96	10.49	20034	20048
5.12	12.5	82000	83999

5.3 Reliability analysis using generated defect list

A reliability analysis as described in Section 4.4 is executed. The input is the defect list containing all the defects, including the most severe shown in Table 5.1. There are 99 defects in total, and each is described by seven random parameters. The computational time will increase quickly with an increased number of cycles, so the number of cycles that can be run is limited for these cases. For increasing the computational efficiency so that more cycles can be run, the number of defects is reduced, so that only the most severe defects are analysed.

5.3.1 MCS initial state, 10000 cycles, 99 defects

The analysis is run with 10000 cycles in each iteration, and ten iterations, with the current state of the pipeline analysed. The results from this run gave no observations of failure, i.e. that with 10^5 cycles run, there were no modes where the system failed. This indicates that the current annual failure probability is smaller than 10^{-4} (for this pipeline segment length) and that the system has sufficient capacity at the present state. There is no reason to show plots of zero probability of failure, but an overview of the parameters deciding the limit states is shown for the initial case in Figure 5.26 and 5.27 below.

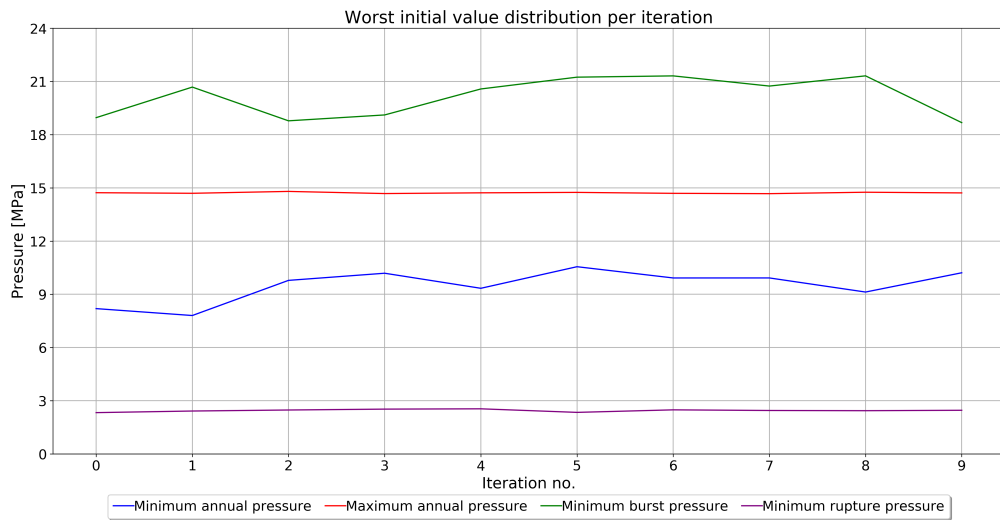


Figure 5.26: The worst pressure parameters in each iteration (not necessarily appearing together).

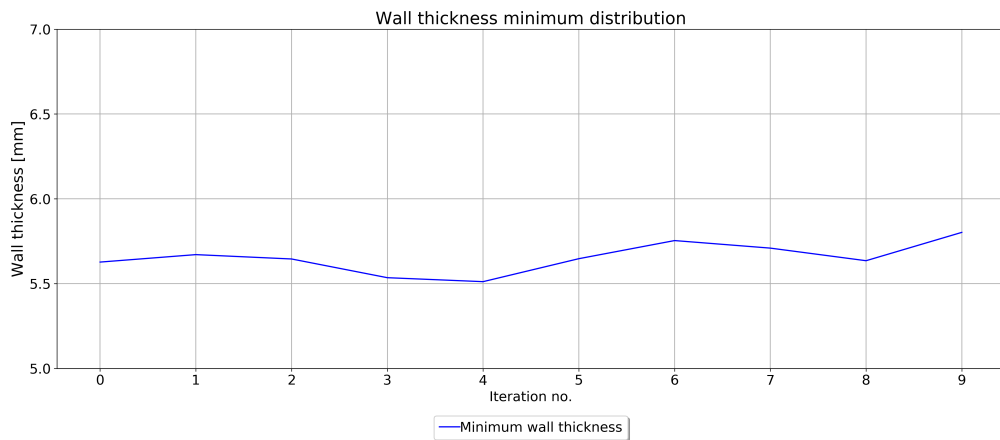


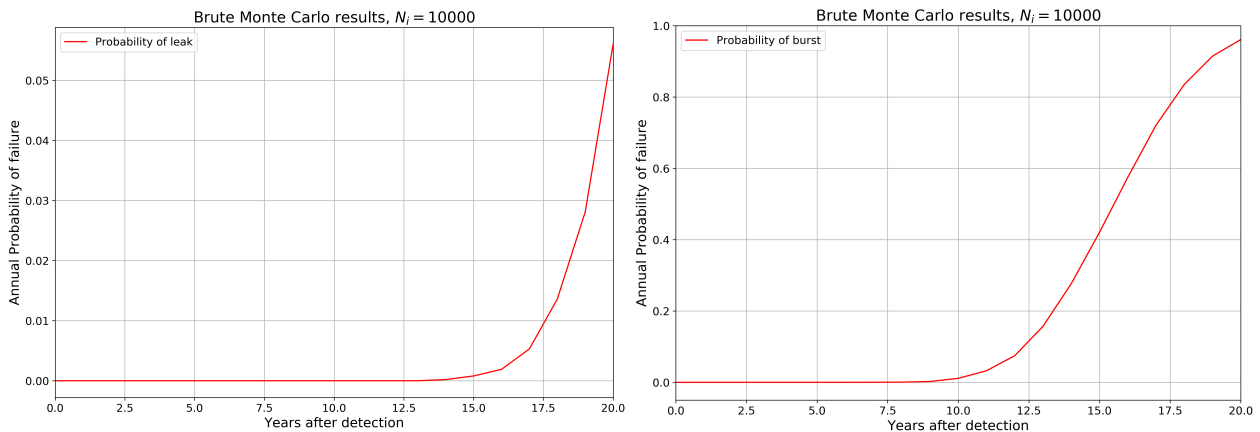
Figure 5.27: Minimum wall thickness for each iteration.

It can be seen that the minimum rupture pressure capacity is significantly lower than both the

minimum and the maximum annual pressure. The minimum burst capacity is, however, well above the annual maximum pressure for all 10^5 cycles, so the pipeline section has sufficient capacity. The minimum wall thickness is well above zero, indicating that leakage is not a problem for the initial stage.

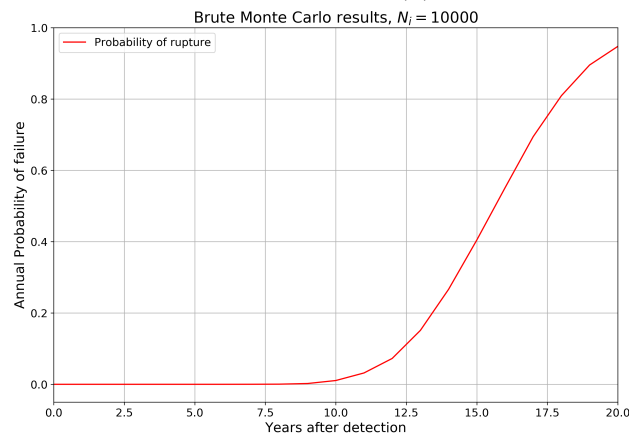
5.3.2 Time dependent MCS: 10000 cycles, 15 defects

The defects with defect depth less than 6 mm was removed to speed up the analysis, giving a total number of 15 defects. The analysis was run with 10000 cycles in each iteration, and with time equal to the next 20 years. The resulting plots can be seen in Figure 5.28 below:



(a) Time-dependent probability of small leak.

(b) Time-dependent probability of burst.



(c) Time-dependent probability of rupture.

Figure 5.28: Resulting plots for MCS-run with 10000 cycles, 9 defects.

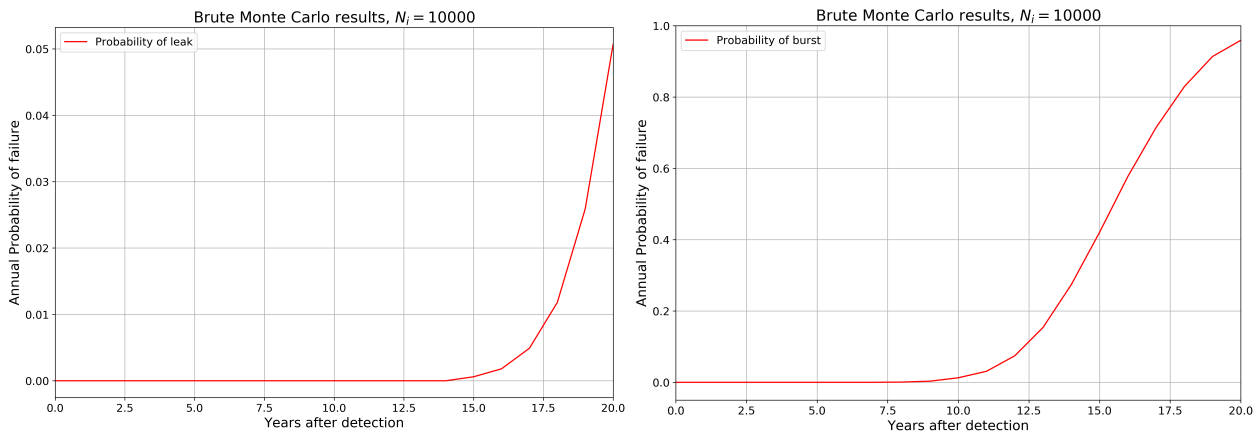
The minimum wall thickness is 7.8 mm, meaning that it would take 19.5 years before a small leak happens with a corrosion rate of 0.4 mm/yr, which is a conservative value provided by CSA (2007). This means that the corrosion rate should be a valid estimate. We can see that the probability of a small leak occurring is negligible until 15 years, and after that, the probability rises exponentially. The probability of a burst is larger, but it remains low until about 9 years when it starts to increase rapidly.

The probability of a large leak is equal to the difference between Figure 5.28b and 5.28c. These plots are almost equal, meaning that the probability of large leak is almost negligible compared to the probability of rupture. This indicates that in case of a burst, plastic instability phenomena are likely. However, this may partly be because of the way the defects are defined in this analysis, as rectangular areas with a constant depth. This analysis has also only considered the 15 largest

defects, which are expected to dominate the failure probabilities.

5.3.3 Time-dependent MCS: 10000 cycles, 60 defects

The analysis was run again with a larger number of defects included, to see if the probability of failure is influenced by taking into account more defects with lower depth. The largest defects as the ones analysed in the previous section are still included. The boundary on depth is now set to 3 mm, leading to a defect list of 60 defects. All other parameters are the same as the previous analysis. The resulting plots can be found in Figure 5.29 below.



(a) Time-dependent probability of small leak.

(b) Time-dependent probability of burst.



(c) Time-dependent probability of rupture.

Figure 5.29: Resulting plots for MCS-run with 10000 cycles, 60 defects.

The results are almost equal to the ones seen in Figure 5.28, indicating that the smaller defects have little influence on the total probability of failure. The differences are of an order of 10^{-4} , which can be blamed on statistical variations, since the number of cycles is relatively low.

5.3.4 Time-dependent MCS: 100 000 cycles, 4 defects

An analysis was also attempted with 100 000 cycles analysing the four defects with remaining wall thickness less than 9 mm. The first five years after detection were analysed. The number of defects is kept low to have a feasible computational time.

No failure was observed with this parameter setting, meaning that the probability of failure for the first five years is probably below 10^{-4} . The resulting plot is shown in Figure B.3 for completeness. It is placed in the appendix since it gives no additional information.

6 General Discussion

This section is split into three parts, where one part concerns itself about the program development for the data filtering algorithm, and the results found in Section 5.1. The second part looks at the defect formulation and the condition of the dataset. The last part is discussion regarding the reliability analysis conducted.

6.1 Discussion regarding the data treatment

The program developed in this project was in part an attempt to improve the rbpGen-program. The filtering methods were studied in detail, and some inaccuracies and fields where the algorithm could be improved were identified. The results from the new program were compared with results produced by rbpGen for verification. These results seem equal for the fabricated datasets, and the new algorithm seems more stable than the old one when dealing with missing data. However, there are still issues when handling these datasets. By introducing the condition that the wall thickness cannot be predicted lower than the lowest observed value, the issues concerned with the filtering have been masked. However, whether this masking results in a less conservative capacity than using the feature list provided by the inspection company directly is unknown, and has not been investigated in the project. Compared to the old algorithm, the results obtained seem stable and of the same order, and for the real-case dataset examined the lower-bound specification seem to work well for producing consistent results.

Finding out exactly what causes the instability has not been achieved during this project. The wall thickness formulation after the data filtering is set as $RWT = WTSO - SO$, meaning that some numerical instability causes the SO to become larger while the $WTSO$ becomes smaller for certain elements. A prime suspect is the filtering of SO -data, which is done as part of the filtering scheme for creating the RWT dataset. The instability seems to be a phenomenon that happens when multiple data elements are missing over a section, and hence ensuring data quality before filtering might be a way to mitigate against this. In conclusion, setting the resulting values to the minimum observed thickness should be a conservative way to deal with the instability. However, if this value is the result of a misreading, it may be overly conservative.

Disregarding the instability present in both programs, the new algorithm for data filtering seem to provide better results. This may be caused by an error found in the filtering algorithm, where the lower bound of the reference values was set to zero. This has been improved in the new program, which might be the reason behind the small improvements in results observed in Section 5.1. The three cases may be a too narrow basis to conclude with anything, but the results do indicate that the program output is correct and slightly less conservative than the old program.

6.2 About the defect listing algorithm and the real case dataset

The defect listing algorithm was developed as an interesting alternative to the MAV. The original thought was that the MAV might identify multiple pits randomly spread out as a single long defect, which would be unreasonable. For the dataset investigated in this project at least, this was not an issue. The pipeline had long corroded sections, and the metal loss was in most cases spread out over large parts of the circumference. So, for this project, it is believed that listing defects from the MAV would create similar results, though there was not enough time to investigate this method further.

The defect listing applied in this project estimates all defects as rectangular metal losses and does not take the complex shape into account. This is considered a conservative approach, but may in many cases be overly conservative. Automation of complex shape assessment is, however, more complicated, and requires a more advanced approach than used in this project.

For this dataset, there are multiple problems present, especially in the first 20 000 mm of the pipeline.

The amount of defects in those areas causes entire sections to be classified as defects. Smaller defects may be obtained when using different formulations for wall thickness over each section; the nominal thickness was set to the mean of the closest sections here. Different section lengths can also give different results. However, as can be seen by comparing the defect contour in Figure 5.25 to the MAV in Figure 5.24, most defects can be considered discovered and classified correctly. Using the MAV would likely give the same results, so using the defect listing algorithm is not necessary for this dataset.

6.3 About the Reliability analysis

The analyses predict that a rupture is much more probable than a large leak in case of a burst. This is probably due to the way the defect shapes are defined, as long rectangular areas growing in both depth and in length. A different defect formulation may give other results in this regard. For example, if pit growth within a large corroded area was taken into account, the probability of large leak would likely increase.

When looking at the time-dependence of the probability of failure, it is evident that the pipeline section has sufficient capacity to withstand the annual maximum pressure for some time forward. No failures were observed running 100000 cycles for the next five years, indicating that the probability of capacity problems during these years can be considered at least lower than once every 1000 years. Corrosion growth over several (more than five) years is required before the failure probability becomes significant. These results should be considered heavily dependent on the corrosion rate.

The reliability analysis conducted in this project is for illustration of methodology. The part of the pipeline considered is 100m long and, usually, such reliability studies are conducted for longer reference lengths. This means that the integrity of the entire pipeline cannot be ensured by analysing only this dataset, but it may give an indication of the conditions present. It is also important to keep in mind that there are a number of sources of error for this analysis. Some of these errors include:

- The starting point is the defect list, which is conservatively formulated. The sources of error from this are stated in the section above. It should give more conservatism than necessary.
- The defect formulation is based on filtered data, which are filtered section-wise. Different section-characteristics may give different results. Since the MAV's are not used, this should not influence the results too greatly.
- Spatial effects are ignored so that defects that really should have correlated characteristics are assumed independent. Two defects interacting to lower the capacity is also ignored. This may be non-conservative, depending on the conditions in the pipeline.
- The possibility of defects growing together is ignored. This is a non-conservative assumption.
- The defects are formulated section-wise, which may split up defects. This can be mitigated against by using different formulations.
- The reliability analysis assumes the conditions constant over time, and that the conditions are described by the distributions specified in Section 4.4. This is probably the largest source of error, as most of the parameters are estimated from literature examples or from standards.

Also, the depth growth rate is taken from the literature, while the length growth rate is set randomly. This may ignore the actual conditions of the pipeline, as corrosion depends on a variety of factors. Some of these are described in Section 3.1.2, and it can be estimated roughly by analysing previous datasets from the same pipeline. Still, this is also assuming constant conditions, which may be wrong in some cases.

A limitation with the program is that the analyses can only be executed for a limited number of cycles. The number of random variables is large, and by treating the pipeline as a series system, one has to organise the data structures accordingly. The solution selected in this project is probably not the best solution with regards to computational efficiency, and the code runs quite slowly. So the system reliability was only checked for 10^4 cycles, but with a different number of defects. As previously mentioned in Section 2.3.2 DNV-GL suggests, as a rule of thumb, $\frac{100}{p_f}$ number of simulations for crude Monte Carlo simulation. This means that the analyses with 10^4 simulations should be able to capture a failure probability larger than 10^{-2} , or once every 100 years with the formulation used in this analysis. Whether this is sufficiently accurate will depend on the consequences of failure, which are not considered further here.

Finally, the probability of failure for the entire pipeline of 5km cannot be derived from this analysis. For that to be possible, it must be assumed that this 100-meter section can represent the entire pipeline. Then there are two possibilities; either by assuming that all sections can be considered equal and sum up the failure probabilities as a series system, or by fitting a defect distribution to this section, and extrapolate the tail for the entire pipeline predicting the worst defect of the system. Both approaches have their limitations, and both will depend strongly on how the datasets are treated in the initial analysis. Since this project has focused more on the dataset treatment, the further extrapolation to the entire pipeline was considered of little interest and has hence not been attempted.

7 Conclusion and Recommendations

7.1 Regarding methodology and results

The results from the data filtering can be considered of good quality, even though the filtering algorithm has some limitations. The inclusion of a lower bound on the wall thickness can be considered a conservative approach, and the program in total should provide more accurate results compared to rbpGen. The results from the comparison are promising, and the work from this project should be a good starting point for making a better implementation than rbpGen using Python 3. It should be noted that more comparison and verification should be needed before the program can be used, in addition to increasing the user-friendliness and usability.

The defect listing results are of the same order as the MAV-approach and can be considered an unnecessary implementation for the program. It does have some value in controlling the defect formulation and could be useful in analysing a dataset with different corrosion, but on the dataset analysed here, it was not worth the effort in developing it. The results were useful in further analysis, but these results could have been easier obtained using the MAV.

The reliability analysis used to calculate system failure probabilities can take different failure modes into account and can handle a large number of random parameters. The drawback is the computational efficiency, which is low when many cycles or many defects are considered. The failure probabilities are dominated by the defects with largest depths, which means that the MCS can be focused around those with little loss of accuracy.

The results from the reliability analysis are concerned with large uncertainties, especially regarding the defect growth rates. However, they indicate that this part of the pipeline has sufficient capacity to survive the annual maximum pressure for some time forward and that the burst modes are the most critical for this pipeline. The entire pipeline has not been considered in this project, so no conclusions can be made in that regard.

7.2 Further work

Regarding the filtering methods, some additional work is required before it is as easy to use as rbpGen. The codes should be cleaned up and made to work with, e.g., a GUI, which is more user-friendly compared to the approach used in this project, which was development-oriented. Some more verification is needed, as the number of comparison studies in this project is quite low. The instability in filtering should also be investigated further, finding the reason behind negative results for poor datasets. For the results in this project, the filtering should have been executed with different section specification to see differences in results. However, there was not enough time available for this to be studied.

The major problem concerned with filtering in this project was the missing data treatment, as the datasets provided occasionally had large amounts of missing data. It would be of interest to use the missing data treatment used in this project (or develop a more advanced approach) in combination with one of the image denoising tools from Scipy. Because most focus went into the missing data treatment, the only filter applied in this project was the DNV-filter.

The listing of defects should be applied on the MAV as well, to compare it to the defect listing method used in this project. Further, complex-shaped defects should be taken into account, as the defect definition in this project is a simplification with a high degree of conservatism. It is also possible to differentiate between external/internal defects using the SO-data, but this has not been explored here.

The Monte Carlo analysis has been set up using distributions and parameters found in the literature; these could be more carefully selected. The spatial correlation could be taken into account, making

the different defects more dependent on each other, and the limit states could also be formulated so that neighbouring defects may have a joint influence on the capacity. An improvement on the linear growth model used in this analysis could also be adopted; more advanced growth models could be used, especially if historical inspection data is included. Parts of the literature review in this project are reviewing some of these more advanced methods, but they could not be included in the model due to their complexity and the information available.

The model built in this project is the simplest form of Monte Carlo simulation and is not an especially fast implementation of it either. Streamlining the algorithm is probably possible, and also using importance sampling to reduce the number of cycles needed for estimating the reliability would speed up the analyses.

For estimating the failure probability of the pipeline system, the entire pipeline should be taken into account, as the part analysed in this project is insufficient as reference length. This can be done, e.g., by estimating the corrosion on the remainder of the pipeline using statistical methods or by applying the rest of the inspection data to a similar approach as the one used in this project.

References

- Aker Solutions AS (July 2017a). *Theory and verification of corrPipe (Internal Report)*.
— (July 2017b). *Theory and verification of rbpGen (Internal Report)*.
- Al-Amin, M., W. Zhou, et al. (Sept. 24, 2012). “Bayesian Model for Calibration of ILI Tools.” In: *Volume 2: Pipeline Integrity Management*. 2012 9th International Pipeline Conference. Calgary, Alberta, Canada: American Society of Mechanical Engineers, pp. 201–208. URL: <https://asmedigitalcollection.asme.org/IPC/proceedings/IPC2012/45134/201/266557> (visited on 05/19/2020).
- Al-Amin, M. and Wenxing Zhou (Sept. 2, 2014). “Evaluating the system reliability of corroding pipelines based on inspection data.” In: *Structure and Infrastructure Engineering* 10.9, pp. 1161–1175. URL: <http://www.tandfonline.com/doi/abs/10.1080/15732479.2013.793725> (visited on 05/06/2020).
- ASME (2012). *Manual for Determining the Remaining Strength of Corroded Pipelines*.
- Breitung Karl (Mar. 1, 1984). “Asymptotic Approximations for Multinormal Integrals.” In: *Journal of Engineering Mechanics* 110.3, pp. 357–366. URL: [https://ascelibrary.org/doi/abs/10.1061/\(ASCE\)0733-9399\(1984\)110:3\(357\)](https://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-9399(1984)110:3(357)) (visited on 11/18/2019).
- Camp, Hans-Joachim de la (2006). “Demonstration of Pipe and Pipeline Integrity by Means of Intelligent Pigging Systems.” In: Pipeline Technology 2006 Conference.
- Cortesi, David (2019). *Pyinstaller Manual*.
- CSA (2007). *Oil and gas pipeline systems, CSA standard Z662-07*.
- Der Kiureghian, Armen (2005). “First- and second-order reliability methods.” In: *Engineering design reliability handbook* 14 (CRC Press).
- Der Kiureghian, Armen, Hong-Zong Lin, and Shyh-Jiann Hwang (1987). “Second-Order Reliability Approximations.” In: *Journal Engineering Mechanics* 113. (Visited on 10/29/2019).
- DNV-GL (2019a). *DNVGL-RP-F101 Corroded pipelines*. URL: <https://oilgas.standards.dnvgl.com/download/dnvgl-rp-f101-corroded-pipelines> (visited on 03/22/2020).
— (2019b). *DNVGL-ST-F101 Submarine pipeline systems*.
- Galambos, Theodore V (1981). “Load and Resistance Factor Design.” In: *ENGINEERING JOURNAL*, p. 9.
- Gong, Changqing and Wenxing Zhou (Nov. 2, 2017). “First-order reliability method-based system reliability analyses of corroding pipelines considering multiple defects and failure modes.” In: *Structure and Infrastructure Engineering* 13.11, pp. 1451–1461. URL: <https://doi.org/10.1080/15732479.2017.1285330> (visited on 02/01/2020).
- Hall, Matt (Apr. 27, 2012). “Smooth operator: Smoothing seismic interpretations and attributes.” In: *The Leading Edge*. Publisher: Society of Exploration Geophysicists. URL: <https://library.seg.org/doi/abs/10.1190/1.2431821> (visited on 05/25/2020).
- Hennig, Thomas et al. (2012). “UT ILI Riverbottom feature assessment of challenging datasets Thomas Hennig,” in: 7th Pipeline Technology Conference. Lingen, Germany.
- Hernández-Rodríguez, M.A.L. et al. (Sept. 2007). “Corrosive wear failure analysis in a natural gas pipeline.” In: *Wear* 263.1, pp. 567–571. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0043164807004590> (visited on 04/14/2020).
- Hunter, J. D. (2007). “Matplotlib: A 2D graphics environment.” In: *Computing in Science & Engineering* 9.3, pp. 90–95.
- Jones, Eric, Travis Oliphant, and Pearu Peterson (2020). *Scipy: Open source scientific tools for Python*. URL: <http://www.scipy.org/>.
- Katsuki, Satoshi and Dan M. Frangopol (1994). *Hyperspace Division Method for Structural Reliability*.
- Kiefner, Dr John F and Kolin M Kolovich (2007). “CALCULATION OF A CORROSION RATE USING MONTE CARLO SIMULATION.” In: p. 7.
- Kiefner, J. F., W. A. Maxey, et al. (Jan. 1, 1973). “Failure Stress Levels of Flaws in Pressurized Cylinders.” In: *Progress in Flaw Growth and Fracture Toughness Testing*. Ed. by J. G. Kaufman et al. 100 Barr Harbor Drive, PO Box C700, West Conshohocken, PA 19428-2959: ASTM

- International, pp. 461–481. URL: <http://www.astm.org/doiLink.cgi?STP49657S> (visited on 05/29/2020).
- Kiefner, J. F. and P. H. Vieth (Dec. 22, 1989). *A modified criterion for evaluating the remaining strength of corroded pipe*. PR-3-805. Battelle Columbus Div., OH (USA). URL: <https://www.osti.gov/biblio/7181509> (visited on 05/01/2020).
- Leis, B. N. and D. R. Stephens (Jan. 1, 1997). “An Alternative Approach to Assess the Integrity of Corroded Line Pipe - Part II: Alternative Criterion.” In: The Seventh International Offshore and Polar Engineering Conference. International Society of Offshore and Polar Engineers. URL: <https://www.onepetro.org/conference-paper/ISOPE-I-97-491> (visited on 05/29/2020).
- Liu, Zhenhui, Stig Olav Kvarme, and Odd Einar Lindøe (June 17, 2018). “A Comparison Study on Processing ILI Data With Different Filtering Methods.” In: *Volume 5: Pipelines, Risers, and Subsea Systems*. ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering. Madrid, Spain: American Society of Mechanical Engineers, V005T04A042. URL: <https://asmedigitalcollection.asme.org/OMAE/proceedings/OMAE2018/51241/Madrid,%20Spain/287395> (visited on 03/22/2020).
- Maes, Marc A., Michael H. Faber, and Markus R. Dann (Jan. 1, 2009). “Hierarchical Modeling of Pipeline Defect Growth Subject to ILI Uncertainty.” In: *Volume 2: Structures, Safety and Reliability*. ASME 2009 28th International Conference on Ocean, Offshore and Arctic Engineering. Honolulu, Hawaii, USA: ASMEDC, pp. 375–384. URL: <https://asmedigitalcollection.asme.org/OMAE/proceedings/OMAE2009/43420/375/338558> (visited on 05/15/2020).
- Mahmoodian, M. and A. Alani (Feb. 2014). “Modeling Deterioration in Concrete Pipes as a Stochastic Gamma Process for Time-Dependent Reliability Analysis.” In: *Journal of Pipeline Systems Engineering and Practice* 5.1, p. 04013008. URL: <http://ascelibrary.org/doi/10.1061/%28ASCE%29PS.1949-1204.0000145> (visited on 05/30/2020).
- McKinney, Wes (2010). “Data Structures for Statistical Computing in Python.” In: *Proceedings of the 9th Python in Science Conference*. Conference Name: Proceedings of the 9th Python in Science Conference, pp. 56–61. URL: <http://conference.scipy.org/proceedings/scipy2010/mckinney.html> (visited on 05/22/2020).
- Melchers, Robert E. (1999). *Structural Reliability Analysis and Prediction*. Second Edition. John Wiley & Sons.
- Moan, Torgeir (1994). *Design of Offshore Structures*. Vol. 1. (Visited on 09/26/2019).
- NDT Global (2020). *Ultrasonic Corrosion Inspection*. NDT Global. Library Catalog: www.ndt-global.com. URL: <https://www.ndt-global.com/about/our-technology/ut-corrosion-inspection-technology> (visited on 06/01/2020).
- Nessim, Maher et al. (Dec. 1, 2009). “Target Reliability Levels for Design and Assessment of Onshore Natural Gas Pipelines.” In: *Journal of Pressure Vessel Technology* 131.6. Publisher: American Society of Mechanical Engineers Digital Collection. URL: <https://asmedigitalcollection.asme.org/pressurevesseltech/article/131/6/061701/444117/Target-Reliability-Levels-for-Design-and> (visited on 04/02/2020).
- Nie, Jinsuo and Bruce R. Ellingwood (2000). “Directional methods for structural reliability analysis.” In: *Structural Safety* 22, pp. 233–249.
- Paul, Subir (2012). “Modeling to Study the Effect of Environmental Parameters on Corrosion of Mild Steel in Seawater Using Neural Network.” In: *ISRN Metallurgy* 2012, pp. 1–6. URL: <https://www.hindawi.com/archive/2012/487351/> (visited on 04/13/2020).
- Pedregosa, Fabian et al. (2011). “Scikit-learn: Machine learning in Python.” In: *Journal of machine learning research* 12, pp. 2825–2830.
- PM International Suppliers (2020). *API 5L X65*. URL: <https://www.api5lx.com/api5lx-grades/api-5l-x65.php> (visited on 06/03/2020).
- Qin, H., W. Zhou, and S. Zhang (Dec. 1, 2015). “Bayesian inferences of generation and growth of corrosion defects on energy pipelines based on imperfect inspection data.” In: *Reliability Engi-*

- neering & System Safety* 144, pp. 334–342. URL: <http://www.sciencedirect.com/science/article/pii/S0951832015002495> (visited on 03/18/2020).
- Ronold, Knut O. (Apr. 28, 2016). *DNVGL-RP-C211 STRUCTURAL RELIABILITY ANALYSIS*. (Visited on 09/26/2019).
- Skrede, Sindre O. (2019). *Methods for Computation of Reliability of Marine Structures*.
- Stroock, Daniel W. (2005). *An introduction to Markov processes*. Graduate texts in mathematics 230. OCLC: ocm57170822. Berlin ; New York: Springer. 171 pp.
- The SciPy community (2019). *Multi-dimensional image processing (scipy.ndimage)*. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.median_filter.html (visited on 05/24/2020).
- The Spyder Website Contributors (2018). *Spyder- The Scientific Python Development Environment*. URL: <https://www.spyder-ide.org/>.
- Thoft-Christensen, Palle and Michael J. Baker (1982). *Structural Reliability Theory and Its Applications*. Springer-Verlag.
- Thompson, Phil (2007). *PyQt v4 - Python Bindings for Qt v4*. URL: <https://web.archive.org/web/20080429185425/http://www.riverbankcomputing.co.uk/static/Docs/PyQt4/pyqt4ref.html>.
- Timashev, S. A. et al. (2008). “Markov Description of Corrosion Defects Growth and Its Application to Reliability Based Inspection and Maintenance of Pipelines.” In: *2008 7th International Pipeline Conference, Volume 4*. 2008 7th International Pipeline Conference. Calgary, Alberta, Canada: ASME, pp. 525–533. URL: <http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1641189> (visited on 05/15/2020).
- Vanaei, H. R., A. Eslami, and A. Egbewande (Jan. 1, 2017). “A review on pipeline corrosion, in-line inspection (ILI), and corrosion growth rate models.” In: *International Journal of Pressure Vessels and Piping* 149, pp. 43–54. URL: <http://www.sciencedirect.com/science/article/pii/S0308016116302150> (visited on 03/18/2020).
- Walt, Stefan van der, S. Chris Colbert, and Gael Varoquaux (Mar. 2011). “The NumPy Array: A Structure for Efficient Numerical Computation.” In: *Computing in Science Engineering* 13.2. Conference Name: Computing in Science Engineering, pp. 22–30.
- Zhou, Wenxing (Oct. 1, 2010). “System reliability of corroding pipelines.” In: *International Journal of Pressure Vessels and Piping* 87.10, pp. 587–595. URL: <http://www.sciencedirect.com/science/article/pii/S030801611000116X> (visited on 01/22/2020).
- (Dec. 1, 2011). “Reliability Evaluation of Corroding Pipelines Considering Multiple Failure Modes and Time-Dependent Internal Pressure.” In: *Journal of Infrastructure Systems* 17.4, pp. 216–224. URL: <https://ascelibrary.org/doi/full/10.1061/%28ASCE%29IS.1943-555X.0000063> (visited on 01/31/2020).

A Flowcharts

A.1 Flowchart of the entire program

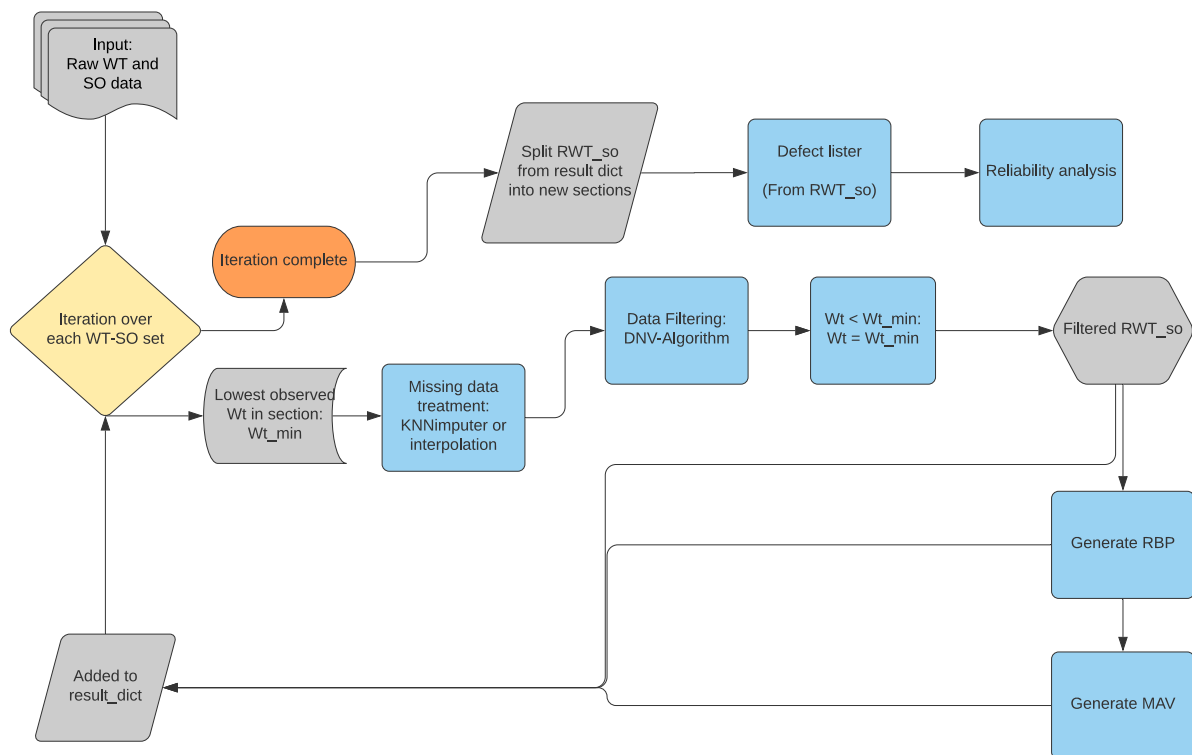


Figure A.1: Flowchart of the entire program.

A.2 Flowchart of the defect listing algorithm

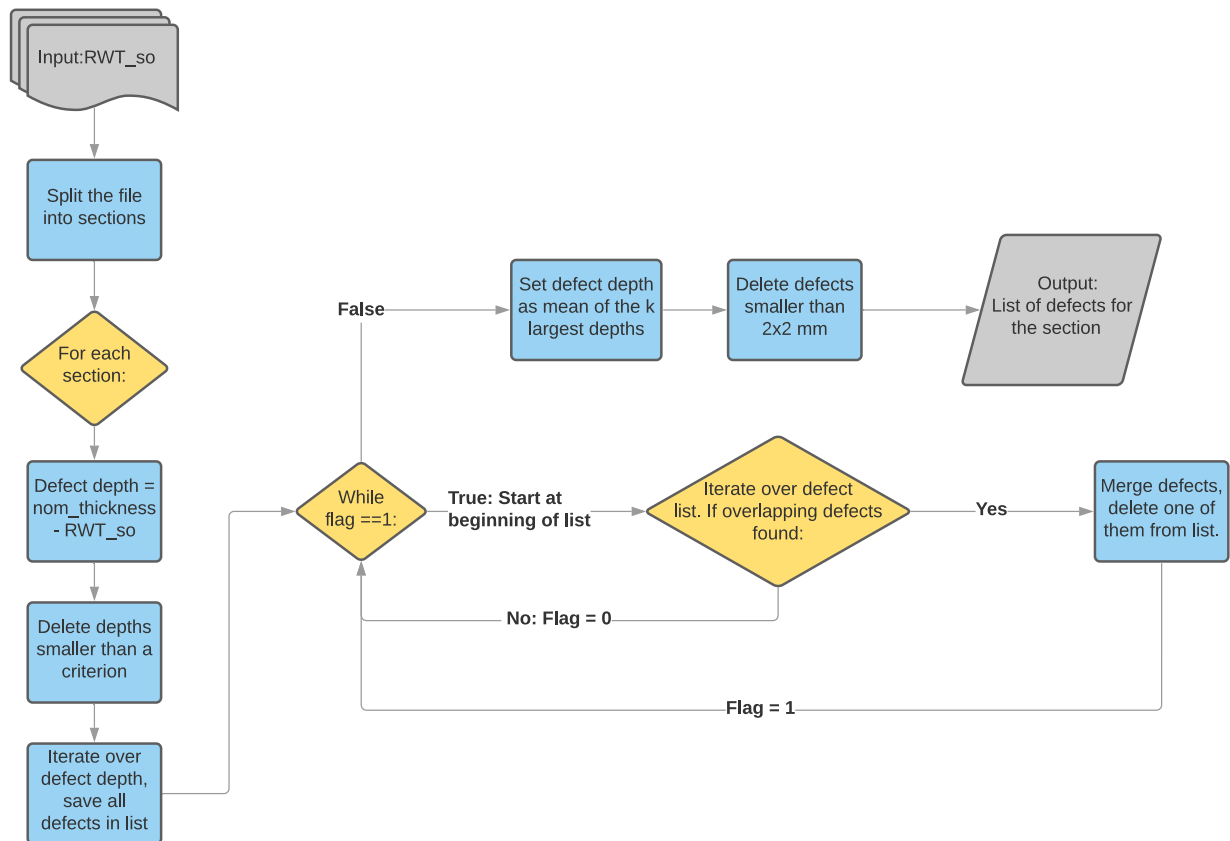


Figure A.2: Flowchart of the defect listing algorithm.

A.3 Flowchart of the MCS-program

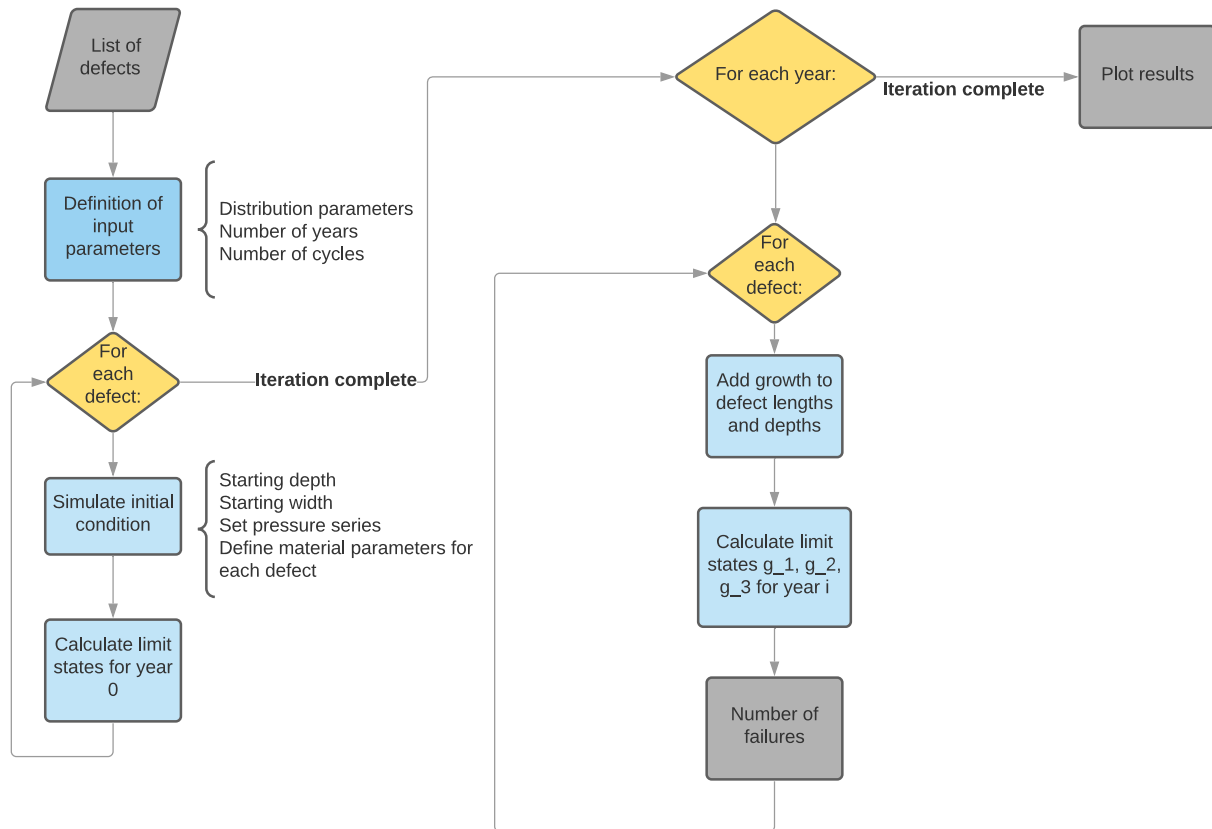


Figure A.3: Flowchart of the Monte Carlo Simulation program

B Plots

B.1 Contour plots

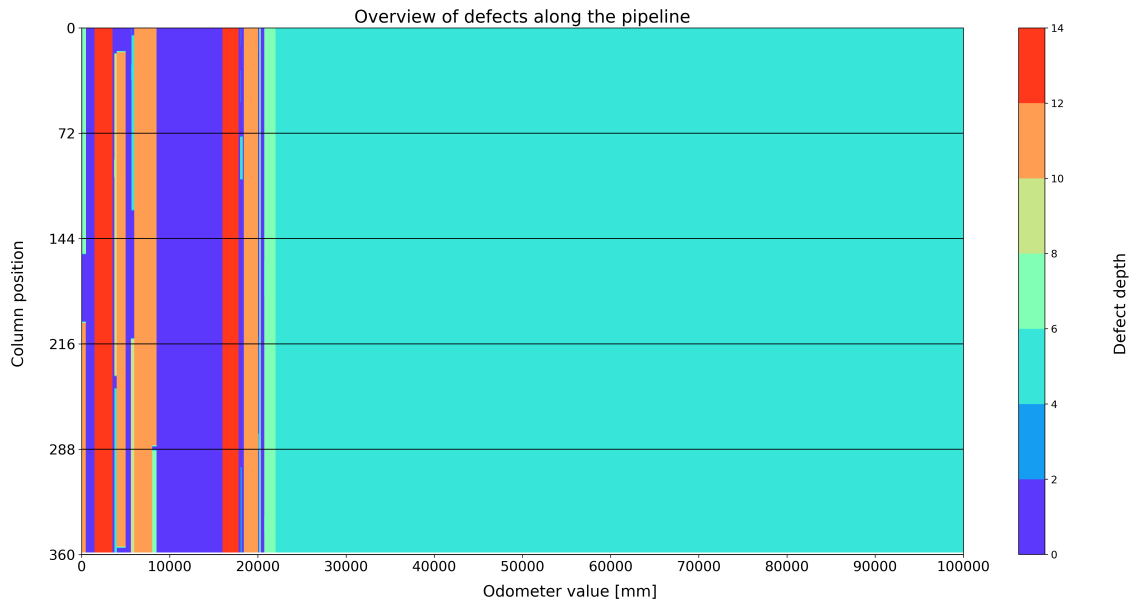


Figure B.1: Grouping of defects from the defect listing in the odometer interval of $[0, 100000]$ mm.

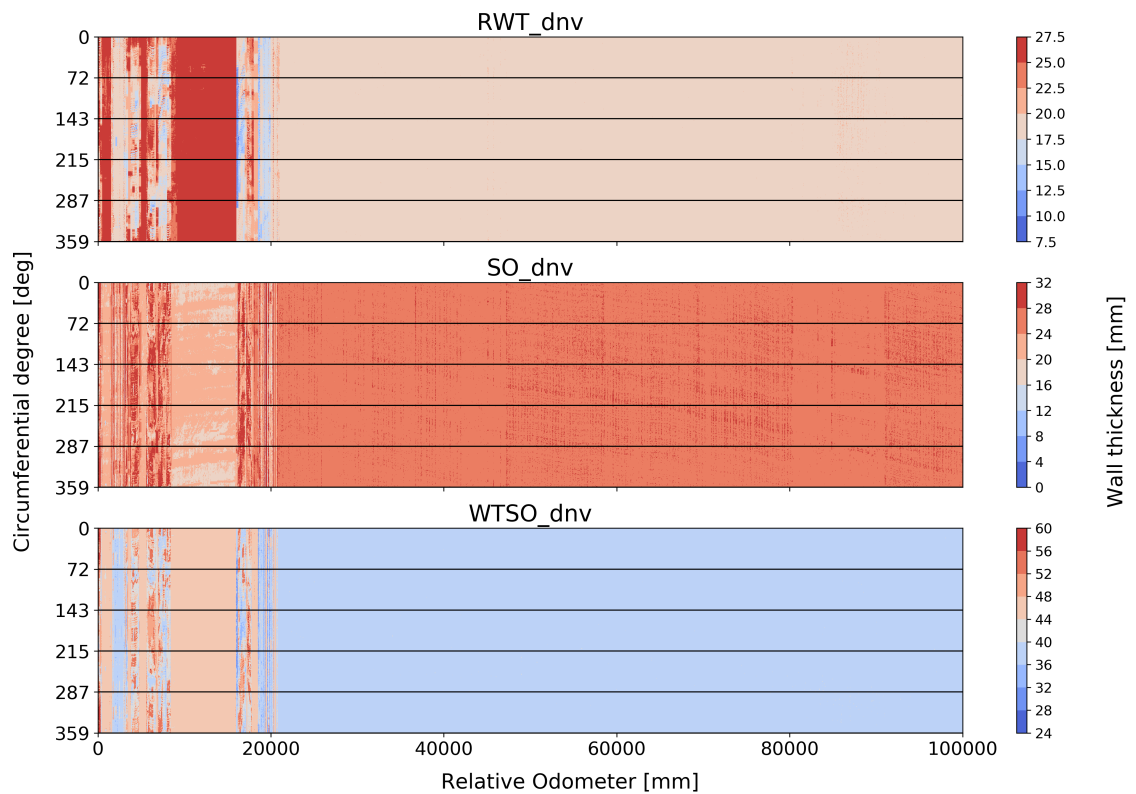
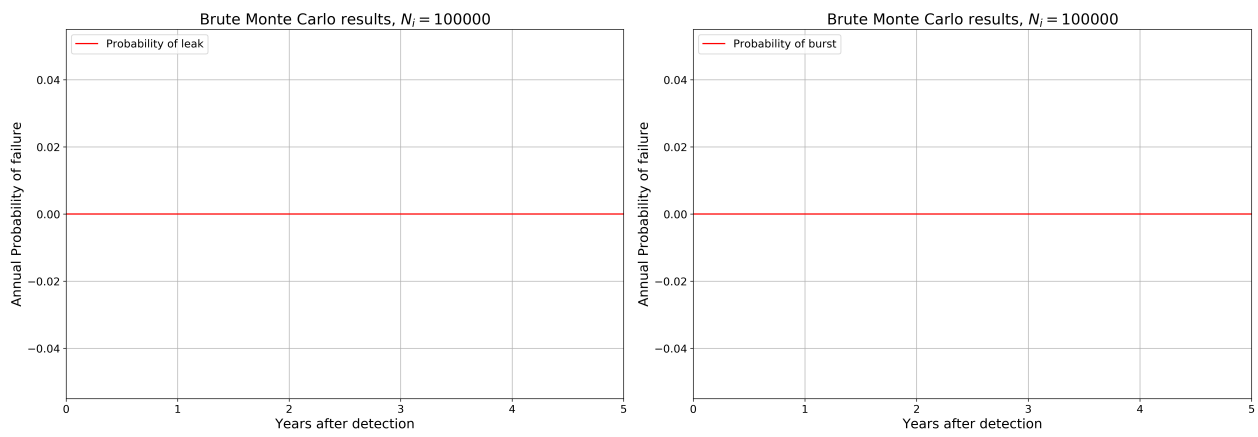


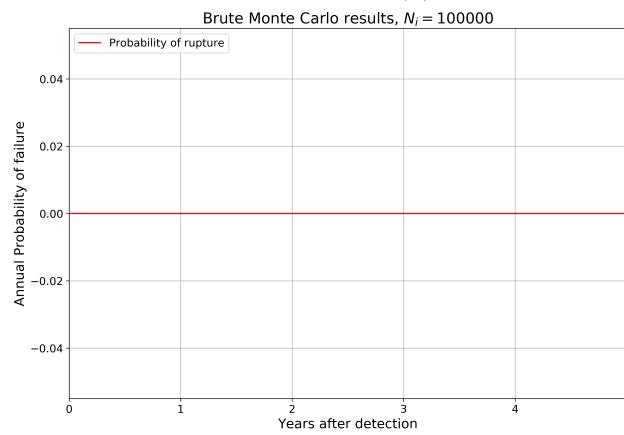
Figure B.2: Contour plots of the entire dataset.

B.2 Results from MCS with zero probability of failure



(a) Time-dependent probability of small leak.

(b) Time-dependent probability of burst.



(c) Time-dependent probability of rupture.

Figure B.3: Resulting plots for MCS-run with 100 000 cycles, 5 defects.

