Support Structure Optimization

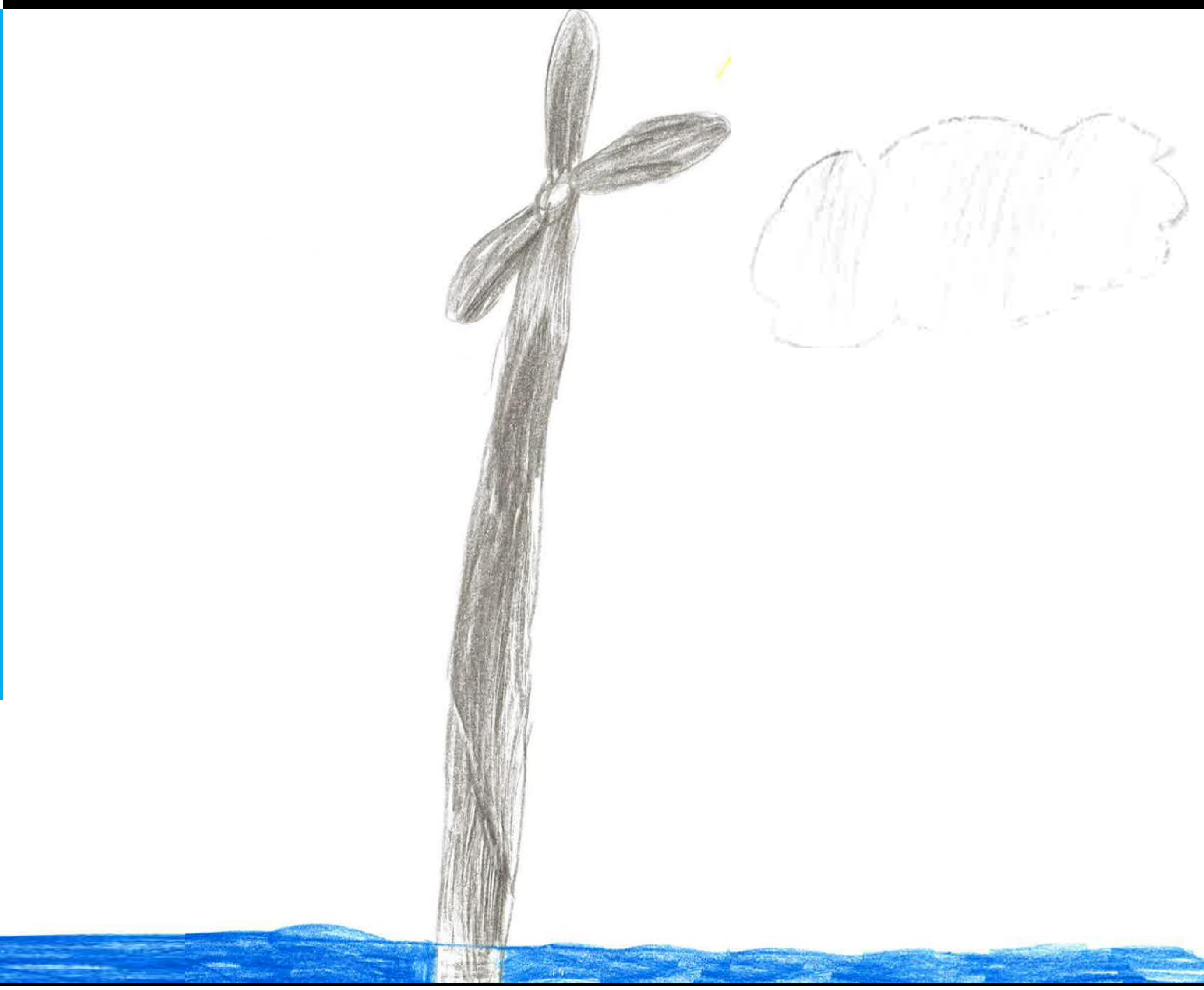
On the use of load estimations for time efficient optimization of monopile support structures of offshore wind turbines

Nico Maljaars

Report no : 2017.021
Coach : Dr.ir. M. Langelaar
Professor : Prof.dr.ir. A. Van Keulen
Specialisation : Engineering Mechanics
Date : May 10, 2017

Department of
Precision- and Microsystems Engineering

Master of Science Thesis



Support Structure Optimization

On the use of load estimations for time efficient optimization of monopile support structures of offshore wind turbines

Master of Science Thesis

For the degree of Master of Science in Mechanical Engineering at Delft University of Technology

Nico Maljaars

May 10, 2017

Supervisors:	Dr.ir. M. Langelaar	TU Delft
	Dr.ir. S.N. Voormeeren	Siemens Wind Power
	Ir. M. van der Meulen	Siemens Wind Power
Thesis committee:	Prof.dr.ir. A. Van Keulen	TU Delft
	Dr.ir. R.A.J. van Ostayen	TU Delft
	Dr.ir. M.B. Zaaijer	TU Delft

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

TU Delft Delft University of Technology

SIEMENS

"Who has gathered the wind in his fists?"
— *Agur (∼ 950 BC)*

Abstract

Over the years, the installed capacity of offshore wind turbines is increasing rapidly. However, the Levelized Costs Of Energy (LCOE) is still higher than the LCOE of traditional energy production methods like nuclear power or energy from coals or gas. This research focuses on a further decrease of the LCOE, by minimizing the mass of a monopile support structure of a wind turbine. This is done in a so called integrated way: Optimizing the tower and the foundation together.

The design variables used in this research are the wall thickness and the diameter of every ± 3 meter section. These can even be cylindrical or conical. To simplify the problem, a parametrization of the designs is used, which reduces the design variables from around 180 to 28. This is checked with existing designs.

Due to the interaction between mostly the first eigenfrequency and eigenmode, the diameter and the waves, it is expected that several local optima exist. Therefore, the proposed optimization strategy is a Particle Swarm Optimization which can be used for a global search for an initial position for a gradient based optimization to find a local optimum, which is possibly the global optimum. In this research the focus is on the Particle Swarm Optimization.

The constraints of the optimization are Fatigue, Buckling, the maximum deflection of the monopile, the angle of the conical parts and the D/t-ratio of the monopile. These are used in the initial design of support structures, so that the optimized designs are realistic. To take the constraints into account, the objective is taken as the mass extended by the penalized constraints.

To reduce the optimization time, the evaluations of the objective function are done by using load estimations instead of extensive load calculations. Several methods are compared on a theoretical basis: Response Surface Methodology, Radial Basis Functions, Kriging, Support Vector Regression, Multi-adaptive Regression Splines and Non-Uniform Regression B-Splines. The performance of a selection of methods is checked on the problem, to come up with reliable estimation methods. To improve the accuracy of the estimations, interaction of Particle Swarm Optimization and the estimators is proposed via estimator updating.

During this research, an optimization tool for monopile support structures is developed. This tool is able to use calculations or estimations of the loads. In order to study the behaviour of the proposed optimization approach and to compare it with the traditional design approach, several case studies are formulated based on a realistic design problem. These are optimized with the optimization tool.

Using a constant tower diameter, the optimization tool is able to reduce the mass of the support structure with 13%. Using the tower diameter also as design variable in the optimization gives a further reduction of the mass with 4%. Several test runs are done, to check whether a global optimum is found or not.

Acknowledgements

As some people know, I like nice quotes. To thank my Siemens supervisors, I found one as well: ‘Succes is a science; if you have the conditions, you get the result’. A part of this project consists of swarm intelligence, simply expressed in the saying: ‘Two heads are better than one’. During my project ‘Three heads are better than one’ was better. Sven and Michiel, you both were the conditions to finish this project by the many ‘weekly updates’ we have had. Michiel, thanks for your assistance by teaching me the fundamentals of Object oriented programming and explaining me FUEL, PILS and STIFT. Sven, thank you for often asking me: ’What do you want to do?’ and ’Do you like it, what you’re doing?’.

I would like to thank Matthijs Langelaar from Delft University for his supervision. I really appreciated the way in which you guided me through this process: You gave me the freedom to do the things I wanted to do, while sometimes pointing me in a better direction by subtle hints, like ‘Ambitieus...’.

‘Many heads are better than one’ fits well to the sixth floor of the Siemens building in The Hague. Thanks to all of you for the nice discussions about both the technical and the sales side of turbines, but also the many other interesting topics during life. Colleague BOBs, thank you for all the ‘potjes’ we have played and off course all forms of other assistance and distraction! Stay thinking about your children’s children...:)

Friends and family, thank you for being who you are! Especially, I want to thank my father and my niece Hannah. My father for the many times you have brought me to Middelburg at Monday mornings and Hannah for reducing the complexity of the design process of an offshore wind turbine, as shown on the front page of this thesis.

Pedor, Seti, Arjoan, Geit, Willy and Corrie, thanks for accepting me as room mate! As I know you have suffered from the last weeks before my graduation in the lack of meals prepared by myself. I will compensate for this...:)

Last but not least, I would like to thank Marit. You joined the project in the final stage, but that stage was giving the highest work load as well. I hope you haven’t suffered that much from the balance between my thesis, you and the rest of the world.

During this project, many aspects of my master track Precision and Microsystems engineering came along. As the track was studying processes on very small scale, wind turbines are some orders of magnitude larger. But for both, often the same principles were valid. Years ago, someone wrote down: ‘As knowledge increases, wonder deepens.’ My wish is that this project will result in glory to God, the creator of wind and waves, heaven and earth.

Den Haag, April 2017 Nico Maljaars

Table of Contents

Acknowledgements **v**

1 Introduction **3**

1-1 Offshore wind energy . . . 3

1-2 Past research on OWT support structures . . . 6

1-3 Novel aspects of this thesis . . . 8

1-4 Thesis objective and tasks . . . 9

1-5 Thesis outline . . . 9

2 Optimization problem **11**

2-1 Introduction design problem . . . 11

2-2 Defining optimization problem . . . 12

2-2-1 Conflicts of interest . . . 12

2-2-2 Objective function . . . 13

2-2-3 Design variables . . . 14

2-2-4 Parametrization . . . 14

2-2-5 Constraints . . . 15

2-3 Optimization algorithm . . . 19

2-3-1 First characterisation of the objective function . . . 19

2-3-2 Algorithm . . . 20

2-3-3 Penalization of the constraints . . . 21

2-3-4 Algorithm dependent parameters . . . 24

2-4 Overview computational steps . . . 27

2-4-1 Set-up model . . . 27

2-4-2 Calculations . . . 29

2-4-3 Analysis . . . 30

2-4-4 Computational time . . . 30

3 Theory of Approximation Methods **33**
3-1 Terminology estimation methods . . . 33
3-2 Explanation of several methods . . . 35
3-2-1 Polynomial fit (Response Surface Methodology (RSM)) . . . 35
3-2-2 Radial Basis Functions (RBF) . . . 36
3-2-3 Kriging (KRI) . . . 38
3-2-4 Support Vector Regression (SVR) . . . 41
3-2-5 Multi-Adaptive Regression Spline (MRS) . . . 42
3-2-6 Non-uniform Rational B(asis)-splines (NRB) . . . 43
3-2-7 Neural networks . . . 44
3-2-8 Methods compared . . . 45
3-3 Multivariate output . . . 47

4 Load estimation **49**
4-1 Quantities to be estimated . . . 49
4-2 Defining inputs of the estimation . . . 51
4-2-1 Assessment of an estimation . . . 51
4-2-2 Creating training, check and test data . . . 52
4-2-3 Possible inputs of the estimation . . . 53
4-2-4 Comparison of several input combinations . . . 54
4-2-5 Comparison of several eigenmode input options . . . 56
4-3 Estimator choice . . . 57
4-3-1 First selection estimators . . . 58
4-3-2 Methodology for choosing free parameters in estimators . . . 58
4-3-3 Comparison estimators . . . 59
4-3-4 Improving vector estimations by Proper Orthogonal Decomposition . . . 60
4-4 Estimator updating . . . 60

5 Optimization results **65**
5-1 Introduction case studies . . . 65
5-2 Case study 1: FDB_ . . . 66
5-3 Case study 2: FD_P . . . 68
5-4 Case study 3: FDB_LP . . . 70
5-5 Case study 4: FDB_LSP . . . 71

6 Conclusions **77**
6-1 Load estimations . . . 77
6-2 Optimization . . . 78
6-3 Optimization and Estimation . . . 79
6-4 Bigger picture . . . 79

7 Recommendations **81**
7-1 Optimization . . . 81
7-2 Estimation methods . . . 82
7-3 Optimization and estimation . . . 83

A Introduction Optimization methods **85**
A-1 Unconstrained optimization . . . 85
A-1-1 Local optimization algorithms/Gradient based optimizers . . . 85
A-1-2 Global optimization algorithms . . . 86
A-2 Constrained optimization . . . 89
A-2-1 Penalty functions . . . 90

B Common terms in statistics **91**

C Non-uniform Regression B-splines algorithm **95**

D Damage Equivalent Load (DEL) **97**

E DEL derivatives **101**

Bibliography **105**

List of Abbreviations

1P-frequency	Frequency of the rotor
3P-frequency	Blade passing (tower) frequency
AEP	Annual Energy Production
ALS	Accidental Limit State
BHawC	Bonus energy Horizontal-axis wind turbine Code
DEL	Damage Equivalent Load
FLS	Fatigue Limit State
GBO	Gradient Based Optimization
KRI	KRIging
LAT	Lowest Astronomical Tide
LCOE	Levelized Costs Of Energy
MRS	Multi-adaptive Regression Splines
MSL	Mean Sea Level
NRB	Non-Uniform Rational Basis Spline
OWT	Offshore Wind Turbine
PPD	Pile Penetration Depth
PSD	Power Spectral Density
PSO	Particle Swarm Optimization
RBF	Radial Basis Function
RNA	Rotor Nacelle Assembly
RSM	Response Surface Modelling
SCF	Stress Concentration Factor
SCOE	Social Costs Of Energy
SLS	Serviceability Limit State
SUS	Support Structure
SVM	Support Vector Machine
SVR	Support Vector Regression
SWL	Still Water Level
SWP	Siemens Wind Power
ULS	Ultimate Limit State

Chapter 1

Introduction

These days most people are aware of the problems that can arise when the energy consumption and energy production on planet earth stay at the same level as it is now. Mainly two sorts of problems can be distinguished: Global warming by the emission of for example carbon dioxide and the exhaust of non-sustainable energy sources like oil, coals and gas. Last years, renewable energy sources in general have gained popularity, although (world) politics and natural disasters have influenced this in a postive and negative way. Wind energy is one of the most important renewables.
This introduction chapter gives first a short overview of offshore wind energy and the design process of offshore wind turbines. Then an overview of optimization projects on wind turbines is given and it is shown how this optimization project distinguishes itself from previous projects. From this an objective and the tasks belonging to this objective are derived. Finally an overview of the chapters is given.

1-1 Offshore wind energy

Using the power that is available by flowing air has a long history. Chinese and Egyptian people were first in adding a sail to their vessels around 4000-3400 BC[45]. The initial idea of a wind mill was described by Hero of Alexandria in the first century. He used a rotor with blades, to pump air in an organ. It was in many ways less advanced than a state-of-the-art windturbine, but the concept of a horizontal axis windmill with blades was there! When electricity was getting more and more popular, the idea came up to create electricity from wind. From the second half of the 20^{th} century, onshore wind turbines became interesting for commercial purposes. Offshore wind turbines were first commissioned around the start of the 21^{st} century and the cumulative installed capacity is still increasing, as shown in Figure 1-1 for Europe.

Foundation types

For offshore wind turbines, several foundation types are used or proposed. An overview of different types is shown in Figure 1-2a. In the initial years of offshore wind turbines, it was expected that monopiles were possible in water depths of at maximum 15 meter. But for several projects in the North Sea, monopiles are already planned for water depths of around 40 meter with significantly larger turbines! In 2016, 88% of the installed foundations was still the monopile foundation as shown in Figure 1-2b.

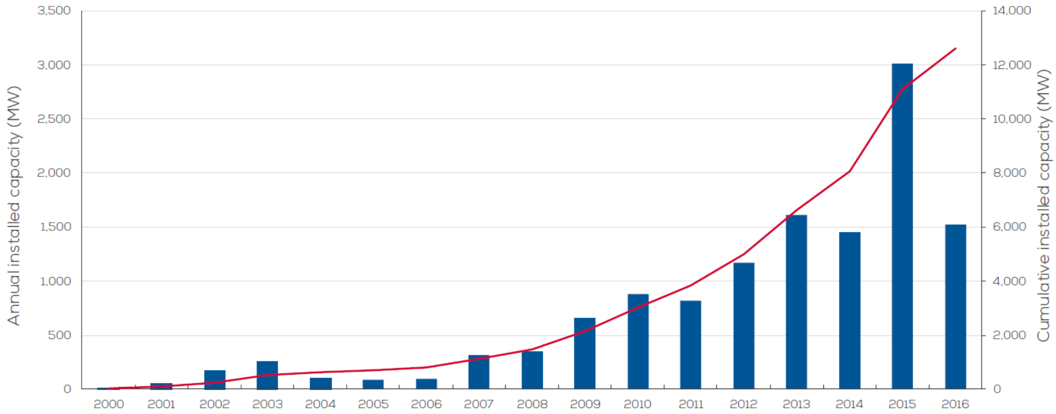
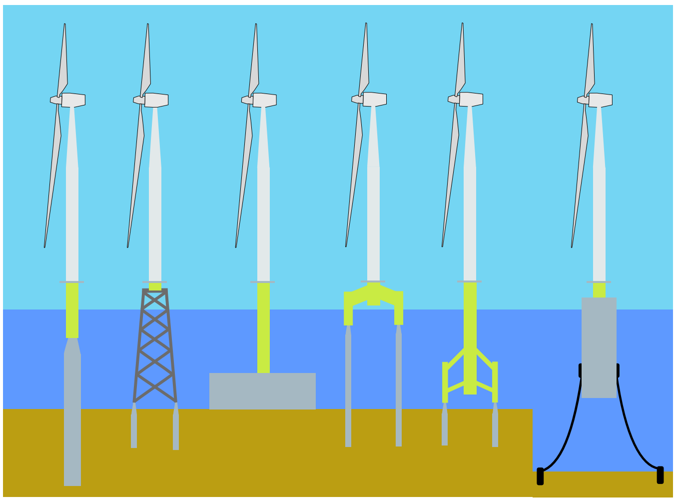
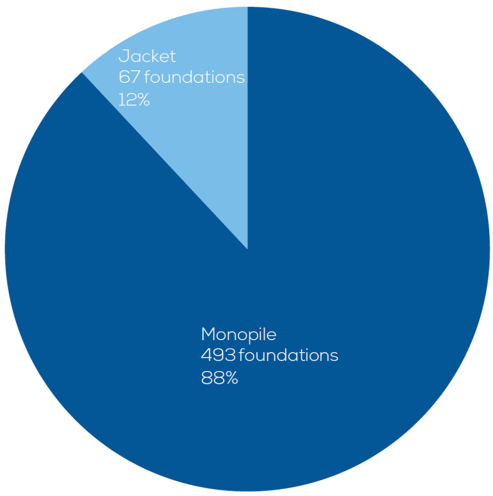


Figure 1-1: Installed offshore capacity per year and the cumulative installed capacity in Europe.[13]



(a) Foundation types. From left to right:
Monopile, jacket, gravity-based, tripile, tripod and floating.



(b) Installed foundations in 2016 (Europe).[13]

Figure 1-2: Overview foundation types used for offshore wind.

Design process OWTs

For onshore wind turbines, there is little interaction between the design of the foundation and the design of the turbine. This is because the wind loads on the blades, which are design-driving, are almost independent of the foundation design. For this reason, a design of an onshore support structure is done by two different parties: The first one is the wind turbine designer, which is responsible for the tower. The second one is the foundation designer, which is responsible for the foundation. When offshore turbines are designed, this design strategy is also used. So an optimized tower (Figure 1-3) from the wind turbine designer and an optimized transition piece and a monopile from the foundation designer together, form possibly an optimal design.

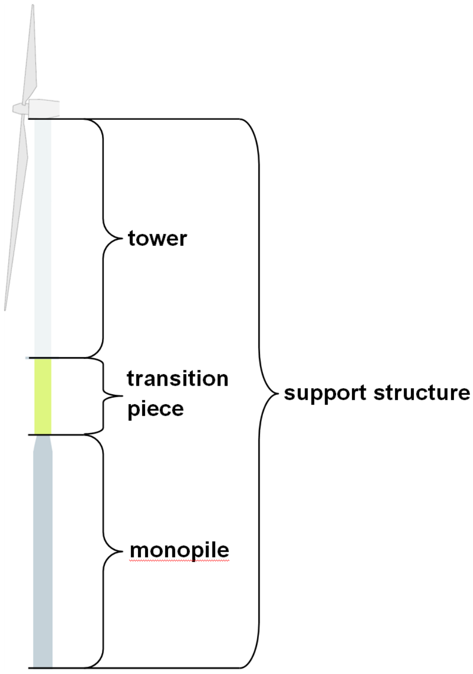


Figure 1-3: Definition of a support structure.

During this design process, some iterations with information about loads and displacement are done(Figure 1-4). The number of iterations is ideally unlimited until there is convergence in the designs. Although

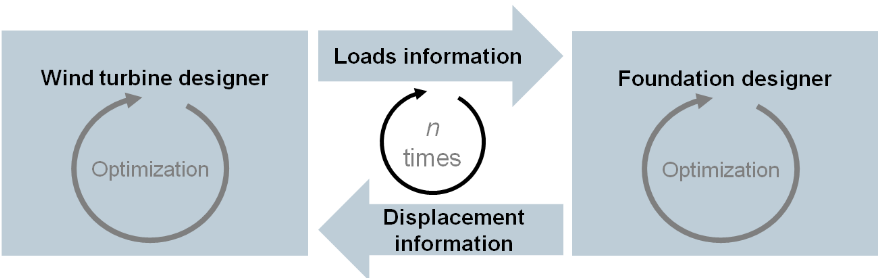


Figure 1-4: Overview of the former design process.

in practice, this is mostly limited by time. And so the number of iterations is in most cases 2 or 3. Besides some practical aspects during the design and looking to the ideal case, an important question has risen: Is this way of designing an offshore wind turbine, leading to an optimal (light-weight) design? For simplicity the weight of the design is assumed to have a linear relation with costs. Research pointed out that integrated design of a tower, transition piece and the monopile (together called support structure) can lead to significant material savings and in this way, cheaper turbines[11][12].

The design considerations of offshore wind support structures include 4 limit states:

- **Fatigue Limit States (FLS)** result from cyclic loading of the structure.
- **Ultimate Limit States (ULS)** are caused by extreme forces on the structure by extreme waves and gusts.
- **Accidental Limit States (ALS)** corresponds to accidental events or failure.
- **Serviceability Limit States (SLS)** requires the functionality of the structure under routine conditions.

This research uses just FLS and ULS because these are generally accepted to be design-driving.

1-2 Past research on OWT support structures

Because of the need for sustainable energy resources, research is done in many fields related to wind energy. Most research is done in the field of wind farm layout design (mostly based on the wakes of the turbines int he farm), the conversion from wind into electricity, structural optimization of the blades and the distribution of fatigue in a wind farm by optimizing the turbine controller (driven by LCOE via maintenance). Optimization of the final goal, minimization of the LCOE[1], is really scarce because cost models are difficult to make and to rely on [15][26][51].

Some studies focused on the optimization of the tower (sometimes the foundation included as well). Muskulus and Schaffhirt[26] have formulated the most challenging issues of design and optimization of a support structure:

- Many **Nonlinearities** due to wakes, waves, soil
- A **Complex environment** caused by the irregular nature of the conditions (causing the need for long simulation times), scour, marine growth and sea ice.
- **Fatigue as design-driving**, so simulations representing the full lifetime are needed.
- **Specialized analysis software** is needed to combine aerodynamic, hydrodynamic and structural analysis.
- **Tightly coupled and strongly interrelated systems** make optimization a complex task. Optimizing each system separately will probably result in a suboptimal configuration. Besides this, installation, operation and maintenance is difficult to take into account.
- **Many design variables and constraints** are possible for the optimization of multi-member support structures, such as jackets.

Now, the most important scientific contributions in the area of support structure optimization are briefly reviewed:

- Malaawi and Negm[28] propose 5 methods (Light-weight design, high stiffness, high stiffness/mass-ratio, frequency-placement criterion, maximum frequency criterion) for the optimization of the structure. Maximizing the systems' first natural frequency by a non-linear mathematical programming model (Powell's technique) using an interior penalty on the objective, gave the best results. This research was done in 2000 and performed on a 100 kW turbine.

[1] Although the final goal is to maximize sustainability, minimization of the LCOE is a common way to reach this.

- Uys et al.[49] show the optimization of a 1 MW turbine based on 3 tubular sections with a height of 15 meter each. Production costs are minimized and buckling is taken as a constraint. The design variables are the mean wall thickness of each 15 meter segment and a certain number of ring stiffeners to prevent buckling. When using the mean wall thickness, it is assumed that the dynamic behaviour and the resulting forces are independent of the distribution of this wall thickness. Besides this, only one static load case is used, even so Chantharasenawong et al. uses two static load cases[3].
- Perelmuter and Yurchenko [30] perform an optimization on the steel conical shell tower of an onshore turbine. The weight is used as objective function. The top, bottom and one extra diameter, the wall thickness for every section of the tower and the length of the tower are used as design variables. In a straightforward, analytical way the stresses for two load cases are calculated, which are used (beside some geometrical constraints) as a constraint. The algorithm that is used to come up with the solution of their non-linear programming task is called the improved gradient method.
- Pasamontes et al.[29] have done an optimization study on the jacket of the OC4 project. A genetic algorithm is used to minimize the weight (fitness defined as $F_i = m_{\max} - m_i$) with first, the diameter and wall thickness as design variables (16 design variables). In the end, the number of bays of the jacket has become a design variable as well (20 design variables). They used design dependent ULS and FLS constraints on each joint in the structure, both based on 1 load case of 30 seconds which is extrapolated to the full lifetime of the structure. For the jacket, the ULS case was design-driving. 300 generations with 15 individuals for the first case and 30 individuals for the second case, are needed to come up with a solution.
- Although simulations in the time domain are most accurate, they are too time-consuming to use in a fatigue assessment during an optimization. For this reason frequency domain calculations are used by several researchers. For example: Thiry et al. [44] developed a methodology to optimize monopile steel structures (5 MW turbine) with a genetic algorithm. The objective is to minimize the weight of the support structure, while the constraints are implemented by means of penalties in the fitness function. Constraints are taken both for FLS and ULS. FLS is calculated based on structure independent damage from wind and structure dependent damage from waves (calculated in frequency domain by linearly combining the PSD of the environment and the PSD of the support structure). ULS is calculated as a result from the wind load on the rotor, the pressure on the structure and a wave load (described by $H_w = 10$ m and $T_w = 14$ s). Soil is not taken into account as the structure is clamped above mudline. This study shows a weight reduction of 21%.

Integrated optimization (tower, transition piece and monopile)

For some years, research into integrated optimization of support structures is done. Both Godfroy[11] and Haghi[12] did an optimization on the SWT-3.6-107 turbine. They used a constant overall geometry by fixing the diameters of all plates. Several case studies were done by optimizing the wall-thickness. These showed a mass reduction of around 25% [11] when using constraints on buckling and natural frequency. A reduction of 12% [12] was found when using constraints on natural frequency, buckling and fatigue. For the latter case, fatigue and the first natural frequency were design-driving.

According to Haghi, higher mass reductions could be obtained with an integrated optimization compared to a strategy where the tower and monopile are optimized separately. This is explained by the fact that the tower mass in the integrated optimization increased, while the total weight decreased, compared to the separate optimized components. It is important to mention that these optimization loops used one load calculation for all considered designs. The successive optimization algorithms used, were built-in MATLAB-functions performed by the command '*fmincon*'. First the Interior point algorithm was used, followed by the SQP algorithm and third, the active-set algorithm was used. The interior point algorithm caused the largest decrease in mass of around 90% of the total mass decrease, while the SQP algorithm was just for sharpening this result by another 9.5% of the total decrease.

Most times the active set algorithm just concluded that the point found was an optimum. The gradients were found by using (forward) finite differences. Haghi recommends to use an own written algorithm for a higher flexibility during coding.

Calculation procedures

Most of these optimization studies used time domain simulations to calculate time series of deflections, loads and stresses. These were used for calculating the maximum stresses for buckling and the time series of stresses for the fatigue life state of the OWT. Because the calculation of these time series requires large computational effort, just a very small number of load cases were taken into account or the loads were kept constant during the optimization. Van der Tempel[50] and Ziegler [55] propose to use another method for the calculation of fatigue life of OWTs using a frequency domain analysis combined with Dirliks method[5] to come up with Damage Equivalent Loads. This gives much shorter calculation times compared to time domain calculations while a relatively high accuracy is achieved. Due to the relatively short calculation times, Seidel et al.[37] recommend this methodology to use in the initial design processes.

1-3 Novel aspects of this thesis

Next, it is discussed on forehand how this research distinguishes itself from previous optimization studies.

Realistic state-of-the-art design

Previous optimization studies did generally solve conceptual problems[44]. As a consequence, these studies did not result in practically applicable designs. Besides this, previous studies were done on relatively small turbines. As state-of-the-art turbines in 2017 have a capacity of around $7 \sim 8$ MW, masses, eigenfrequencies and hydrodynamic loads are completely different and so are the challenges.

In this thesis, a tool is described which is developed to be used for the initial design of state-of-the-art turbines.

Realistic constraints

Previous research projects used mostly not-realistic or a non-complete set of constraints. Non-realistic constraints were due to neglecting the FLS[49]. A non-complete set of contraints was used when a small number of sea states was interpolated to the full life-span for a fatigue analysis[29]. In this thesis, the aim is to use all constraints which are possibly design-driving and therefore used in the initial design, to result in a representative design of a turbine in the end.

Design variables

Previous, more realistic studies only used a limited number of design variables, to simplify the optimization problem. Other studies did so, to keep realistic loads[12]. For example, when using design independent loads, the eigenfrequency of a structure should not change that much. This was reached by fixing the TW and MP diameters.

Wind turbine designers are also used to keep some important parameters constant. For example the diameter of the tower is fixed to standardise the internals of the tower.

In this optimization, TW and MP diameters are both design variables. With the case studies, the influence of fixing the TW diameter is assessed.

1-4 Thesis objective and tasks

As previous sections have shown, there is still a gap between research and application in the wind industry. This can also be derived from Muskulus and Schaffhirt when they state[26]: "*... a convincing demonstration of the potential for cost reduction with automatic wind turbine design and optimization methods is lacking.*".

In this thesis an approach is proposed which can be used in practice by designers of offshore wind turbines. This is shown with a case study. To let the approach be useful, it is important that it has capabilities to run an optimization in about 15 hours (overnight). From this follows the objective of this thesis:

Develop an efficient approach for integrated optimization of monopile based Offshore Wind Turbine support structures, which can be used in engineering practice.

From this objective, four main tasks are formulated:

1. Develop an **optimization strategy** for integrated optimization of a support structure.
2. Investigate **estimation methods** and develop an accurate way for the estimation of loads.
3. **Combine Optimizer and Estimator** into an efficient optimization routine.
4. Show the performance of this optimization approach on **a representative case study**.

1-5 Thesis outline

This chapter has introduced some background concerning wind energy, wind turbines and the optimization of the design of a support structure of a wind turbine.

In Chapter 2, the optimization problem is introduced. After some explanation is given concerning the characteristics of this problem, an optimization strategy is proposed (task 1). This includes the objective and the algorithm to solve this optimization. The different computational methods that are used in this thesis for the evaluation of the objective are explained, after which the chapter finishes with an overview of the computational times for this evaluation. From this overview follows the need for a cheaper evaluation of the objective.

Therefore, Chapter 3 is introducing several estimation methods (task 2a), which can possibly be used for the estimation of loads. These methods are explained so far, that an informed decision can be made on the choice of the estimator.

Chapter 4 explains first which quantity needs to be estimated. After that, step by step, the inputs of the estimator are defined. This is followed by the choice of the estimators (task 2b), based on the implementation as explained in Chapter 3. Initial results showed a risk of combining estimator and optimizer. Therefore a method is introduced to further improve the accuracy of the used estimations.

When the estimators are defined, they are implemented into the optimization (task 3). To show the performance of the combination of estimator and optimizer, in Chapter 5 first some case studies are defined. The results of the optimization of these case-studies are shown and further explained (task 4).

The most important conclusions that can be drawn, are given in Chapter 6. This thesis concludes with Chapter 7 were several recommendations are given concerning modelling, optimization of monopile support structures and estimations.

Chapter 2

Optimization problem

The final goal of this thesis is to come up with an efficient approach for integrated optimization. To show the context of the optimization, the design problem on which the approach is tested, is introduced first. After this, the optimization problem is defined, by explaining the objective, the design variables and the constraints. This is followed by the explanation of the optimization algorithm. This chapter concludes with an overview of the computational steps.

2-1 Introduction design problem

The OWT optimization is performed for wind turbines that are planned to be built in the North Sea. The farm consists a number of 7.0 MW Siemens OWTs. The water depth is around 40 metres and the wind turbines are placed on monopiles. The hub height is about 100 metres above sea level Because the soil is sand and so relatively strong, a penetration depth of the monopile into the soil of 30 metres is needed. Adding up these lengths gives a support structure length of around 170 meters.

As mentioned in Section 1-2, minimization of the LCOE is the final goal. Because of the absence of a direct formulation of the LCOE based on the geometry of the monopile, previous optimization studies used the weight as objective to minimize. Because in engineering practise, this objective is also commonly used, in this optimization minimization of the weight is done.

The weight of the new design is compared to the weight of a representative wind turbine designed for that position (further referred to as 'current design' or 'current support structure'). This design is made by a tower designer and a foundation designer in the 'old' way as described in Section 1-1. For confidentiality reasons, masses, dimensions and frequencies are all normalized in this thesis. In the normalization the weight of the current design is set to 1.

As mentioned in Section 1-2, optimization of support structures has to deal with many non-linear effects. This is why it is expected that there will be many local optima when the weight of the support structure is minimized.

*

Removing transition piece The current support structure has three parts which are assembled offshore: a monopile, a transition piece and a tower. The connection between TP and TW is done with a flange, while the connection between TP and MP is a grouted connection. Nowadays, flanged connections between TP and MP, as it is done between TP and TW, are getting more common. For this reason, in this optimization, a flanged connection is assumed. When weights are compared with the current design, for a fair comparison, the weight of the structure without flanges and skirt is taken.

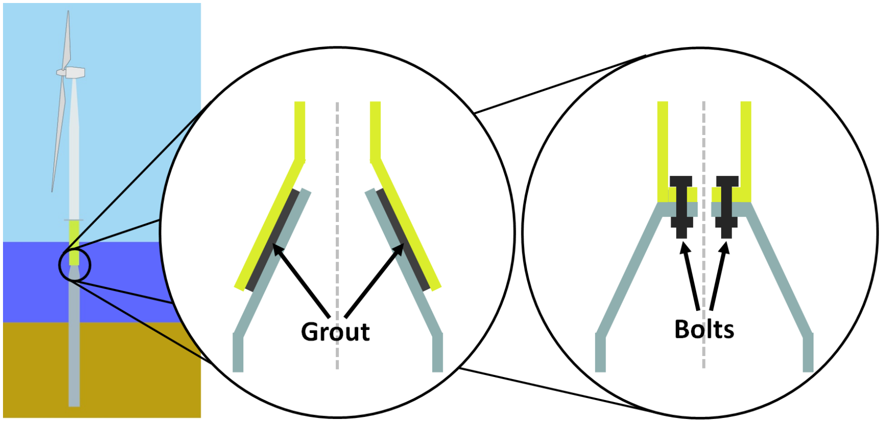


Figure 2-1: Cross-section of a grouted connection at the left and a flanged one at the right.

2-2 Defining optimization problem

Before starting up an optimization, it is important that the problem is well defined. But before objective and constraints are defined, some conflicts of interest are discussed to give more understanding of the problem.

2-2-1 Conflicts of interest

Most optimization problems show some conflicts of interest. These conflicts can transform the optimization into a challenge. To give some more insights in direct and in indirect conflicts of interest belonging to this optimization, the most important ones are mentioned below.

Stiffness versus weight Within the optimization, stiffness and weight are both dependent on the wall thickness and the diameters. So low values for these parameters, give a low weight for the support structure, which gives a good value for the objective function, when the mass is minimized. For the stiffness this is more complicated. Because there is a ULS constraint on the maximum deflection and rotation of the monopile, a certain minimum stiffness is needed. For the constraint on buckling the same statement can be done, that a minimum stiffness is needed to prevent buckling under a certain load.

Stresses versus weight To a certain extent, it is preferable to have small instead of high stresses. So having a load, a large diameter and wall thickness creates a large cross sectional area, which causes small stresses. This results in a conflict of interest between stresses and weight.

Stiffness versus loads The stiffness influences the loads in both the FLS and the ULS case. For the FLS case this is caused by a typical increasing equivalent spectral energy for a decreasing first natural eigenfrequency. A relative larger amplitude around SWL for the first modeshape, for an unchanged first natural frequency, causes higher loads for the FLS case as well [37]. The stiffness influences the loads in the ULS case also. A flexible support structure is deflected more by a large wave while a stiffer support structure can withstand such a wave better.

Weight versus eigenfrequency A simple formula to calculate an eigenfrequency is given by:

$$\omega = \sqrt{\frac{k}{m}}. \tag{2-1}$$

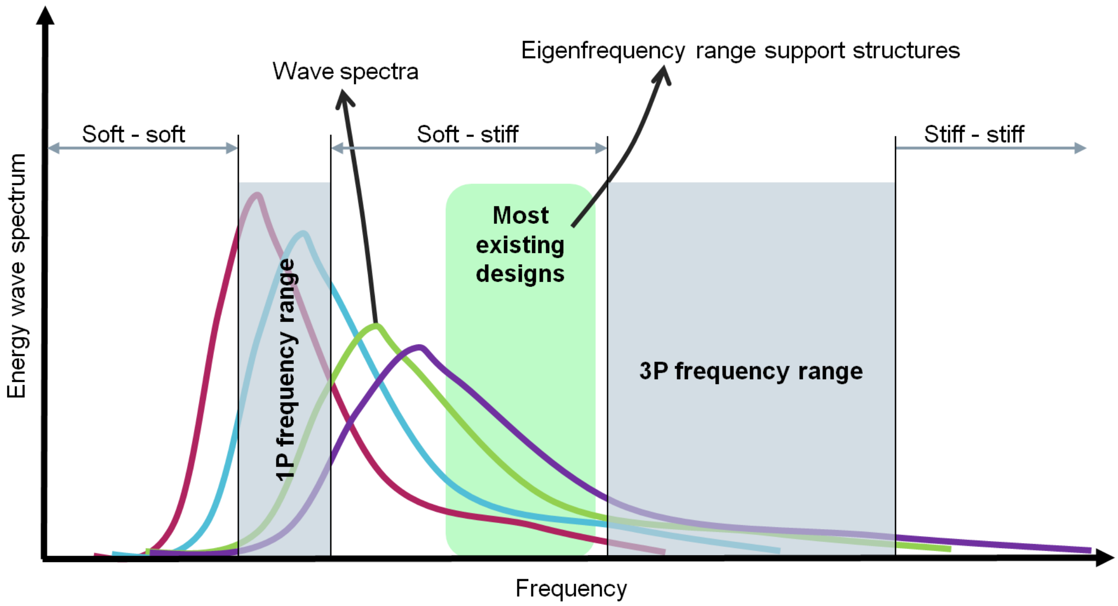


Figure 2-2: Energy-frequency spectrum of waves, with indicated the 1P and 3P frequency ranges and the eigenfrequency range of current OWTs with a monopile foundation.

It is directly clear that a higher mass causes a lower eigenfrequency, while a higher stiffness causes a higher eigenfrequency. In the wind industry eigenfrequencies are mostly linked to the rotational frequency of the rotor. This is shown in Figure 2-2.

The 1P frequency is the rotational frequency of the rotor. The 3P frequency is the blade-tower passing frequency for an OWT with three blades. When the eigenfrequency is within one of these areas, the support structure is often excited in the neighbourhood of this frequency during its lifetime. This causes many vibrations, which can be harmful for the fatigue life of the structure. Besides this, typical wave spectra contain most energy in waves with a frequency around the 1P frequency. So placing the eigenfrequency of the structure close to these high energetic wave frequencies, causes many wave induced vibrations. This can be harmful for the fatigue life of the support structure as well.

2-2-2 Objective function

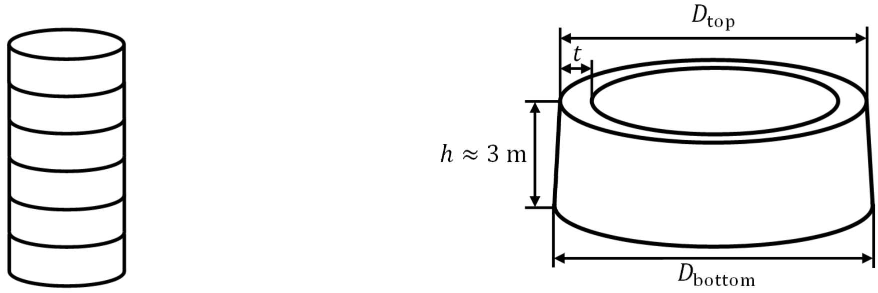
When keeping in mind that fatigue is generally design-driving, an optimization strategy is to remove material until the damage on every place along the support structure exceeds 1 during its lifetime (when the fatigue damage is smaller than 1). This is the same way as when the optimization is done manually. Although, when looking to the conflicts of interest as given in Section 2-2-1, it should be realized that this does not by definition lead to the best optimum in terms of weight. The final goal is to minimize the weight of the support structure m_{sus}. Therefore, this is taken directly as objective function:

$$f(\boldsymbol{x}) = m_{\text{sus}}(\boldsymbol{x}) = \sum_{i=1}^{n_{\text{plates}}(\boldsymbol{x})} m_i(\boldsymbol{x}). \tag{2-2}$$

with m_i the mass of plate i and n_{plates} as the number of plates.

2-2-3 Design variables

A support structure is an assembly of all different sections welded to each other as it is shown in Figure 2-3a. A section is a circular plate of steel with a certain top diameter, bottom diameter and a constant wall thickness, as shown in Figure 2-3b. So for this case study (Section 2-1), this gives about 170 design variables (assuming a constant tower length).



(a) Multiple sections. **(b)** 1 section with its design variables.

Figure 2-3: Section definition as used in this thesis.

2-2-4 Parametrization

While defining an optimization problem, a reduction of the design space is in most cases also a reduction of the problems complexity. Although careful handling is needed to prevent that the results become dependent on the design space reduction.

For the support structure some things are important to couple or parametrize when looking to the current design, to the physics, to its environment and to its fabrication:

- The top of a plate and the bottom of the next plate need to coincide with each other for welding.
- Current production processes are arranged to handle a constant tower diameter with a conical top part and a constant monopile diameter with a conical top part to connect to the tower diameter. This is visible in Figure 2-4 as well.
- The top diameter of the tower is fixed and thus it is an input of the optimization.
- Just below the top, the wall thickness typically decreases in a more or less quadratic manner.
- Because the wind loads exerted on the support structure itself are small compared to the wave loads on the support structure and wind loads on the rotor, the moment line over the height looks quite linear.
- The door in the tower bottom adds a stress concentration factor (SCF) for the fatigue analysis. For this reason the plates around the interface level have generally a deviating thickness compared to overall development of the wall thickness.
- For both sides of the conical part between tower and monopile, the transition from straight to conical causes design-driving SCFs. A linear development of the wall thickness of the conical part is for this reason conservative.
- The soil can consist of different layers (sand or clay) with their own properties which can influence the shape of the monopile below mudline due to 'flexible' layers.
- Below mudline, the bending moment and the shear force are transferred to the soil, in such a way that they are almost zero at pile tip

Based on this, a parametrization of the design variables is made. This is shown in Figure 2-4 and gives a reduction to 28 possibly free variables. In the parametrization, the wall thickness's are not defined per section, but per part. Every part has a wall thickness at the top and at the bottom. The wall thickness for one section is the mean of a linear interpolation for part 1 till 6 and 8 till 10, and a cubic interpolation for part 1 and part 7 (where 3 thickness's are defined) of the wall thickness over one section. Note that for the straight parts, the wall thickness at the bottom of part n is equal to the wall thickness at the top of part $n+1$.

To verify this parametrization, it is compared with existing tower and foundation designs by doing a Least-Squares fit of the parametrization. In Figure 2-5 the results are shown for the current design. It is clear that the parametrization is able to come up with a comparable shape of a tower of an OWT. For a foundation this can be done as well, although these results cannot be compared directly due to the TP removal (as explained in Section 2-1).

Design space

To decrease the size of the design space certain bounds can be used. All common support structures have a minimum diameter of 4 meters and a maximum diameter of 10 m for both tower and monopile. This is driven by the production and the installation. That is why these diameters are used as bounds. The lengths of the tower parts are chosen in such a way that the height of the towertop can be in between 100 and 140 m above Still Water Level(SWL). The pile penetration depth (PPD, length of the monopile below mudline) is in between 25 and 50 meter.[1] The wall thickness of the lowest plate is fixed a priori at 90 mm needed for driving the monopile into the seabed.

2-2-5 Constraints

Before a wind farm can be build, it has to be certified. For this purpose, the designs have to be checked with guidelines. DNV-GL provides a guideline which requires checks on 4 limit states: Ultimate Limit State(ULS), Fatigue Limit State (FLS), Accidental Limit State(ALS) and Serviceability Limit State(SLS)[7]. Experience within SWP gives that ULS and FLS are generally design-driving. This aligns with previous optimization studies (Section 1-2), which uses among other things, Fatigue or Buckling or both as a constraint. Next, all constraints that can be used for the assessment of a design during the optimization are shortly discussed. At the end of this chapter, the computational implementation is discussed. For ULS, Buckling and Maximum deflection are discussed.

Fatigue

The FLS assessment requires that the fatigue damage should not exceed the admissible damage. This can be calculated by the Palmgren-Miners rule, which states that the cumulative damage D of the stress levels r during lifetime had to stay below a constant $i_{\max}$:

$$D = \sum_{r=1}^{l} D_r = \sum_{r=1}^{l} \frac{n_r}{N_{r,\max}} < i_{\max}, \tag{2-3}$$

with l the number of stress levels, n_r the number of cycles, $N_{r,\max}$ the maximum number of cycles and D_r the damage for stress level r. $N_{r,\max}$ can be derived from a Wohler-curve. An example of this is given in Figure 2-6. The maximum damage $i_{\max}$ is 1 for the tower and 0.9 for the monopile, due to some extra damage caused by pile-driving.

Maximum deflection

A support structure can show large deflections when subjected to extreme load cases (extreme wave, broken turbine). These large deflections can lead to the exceeding of the ultimate resistance of the soil, which can lead to plastic deformation of the soil. This can possibly result in an inclination or even a collapse of the support structure. For this reason, maximum deflection and rotation of the top and the tip of the MP are used as constraint.

[1] These numbers are site-specific. In this case the soil conditions result in a relatively short monopile (Section 2-1). For weaker soil-conditions, it is possible that the upper bound has to be increased.

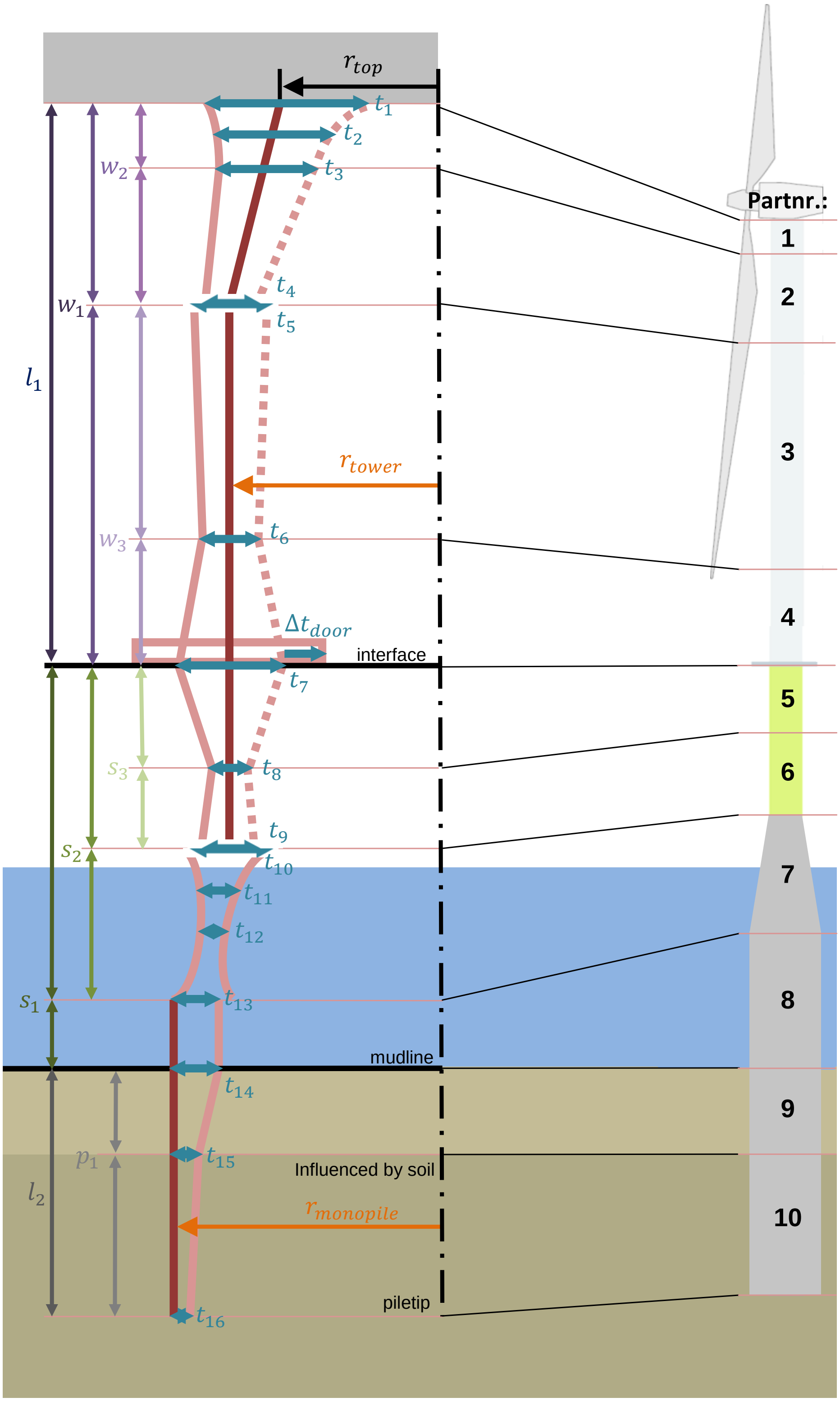


Figure 2-4: Shape of a support structure used for the parametrization. Blue indicates wall thickness variables($t_1, \ldots, t_{16}, t_{door}$). Orange indicates radius/diameter variables($r_{tower}, r_{monopile}$). l_1 and l_2 give length to the tower and monopile. w, s and p are ratings between two lengths. Black indicates fixed heights and so, even r_{top} is fixed.

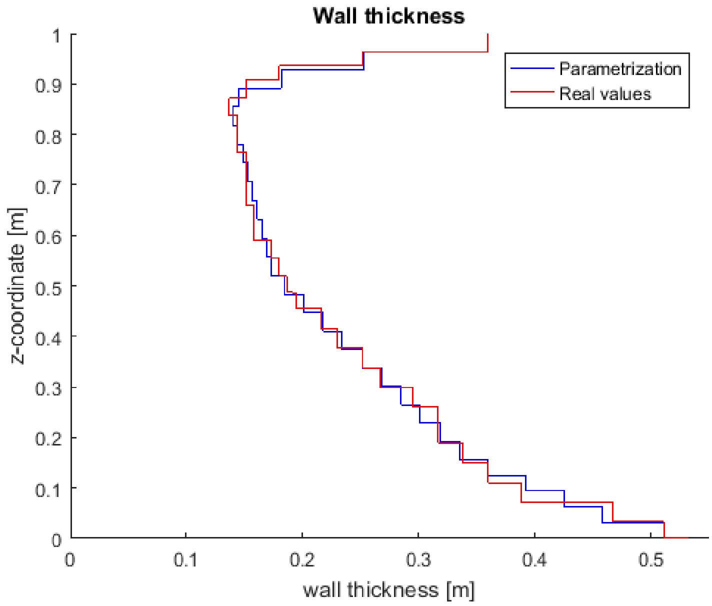


Figure 2-5: Normalized thickness over height found by a Least-Square Error fit on the current support structure design.

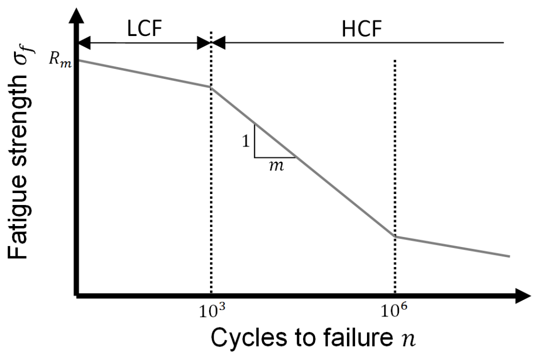


Figure 2-6: A typical Wohler(SN)-curve, with the indicated regions of Low Cycle Fatigue (LCF) and High Cycle Fatigue (HCF).

Buckling

Slender structures have a high probability to buckle under high loads. Buckling should be prevented because post-buckling behaviour in most cases has a lower stiffness than pre-buckling behaviour. This instability can cause a collapse of the OWT. From scaling perspective two sorts of buckling exists. Applied on an OWT they are:

1. **Global buckling**
   This is sometimes also called column buckling and shown in Figure 2-7. It is not known as a failure mode on an OWT, as the bending loads resulting from wind and waves lead to significantly higher normal stresses in the structure than the normal loads resulting from gravity. Due to these stresses local buckling is often more critical and therefore global buckling is not taken into account during the optimization.

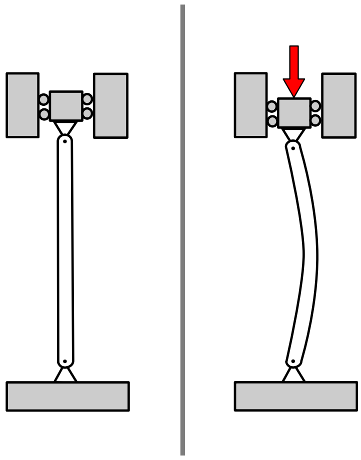


Figure 2-7: A column exhibiting the characteristic deformation of buckling under a centric axial load[42]

.

2. **Local buckling**
   This is buckling on much smaller scale. For example, when the deflections of the tubular cross section are showing large, non-symmetric deformations. Although the buckling effect is local, it can cause a collapse of a full turbine (as shown in Figure 2-8). For this reason it is taken into account in the optimization.

First natural frequency

In several optimization studies, the first natural frequency is taken as a constraint. It was used to prevent high energy adsorption by the support structure when its first eigenfrequency shifted too close to the wave frequency spectrum, 1P frequency or 3P frequency (Figure 2-2). Because they kept loads on the structure constant, the effect of eigenfrequencies close to the fore-mentioned frequencies was not represented in the loads, which could cause unrealistic results. In this optimization study, the wave loads depend on the design (explained in Section 2-4). Because the 1P frequency range is overlapping the wave frequency range with a relatively high amount of energy, this is not used as a separate constraint. The 3P frequency is, for this size of turbines, often in between the first and second well-separated eigenfrequencies and so generally not critical, but just checked in the end. For these reasons no constraint on one of the eigenfrequencies of the support structure is used.

Geometrical constraints

Besides constraints that resulted from loads during its lifetime, also some constraints can be formulated concerning the geometry of the support structure:

Figure 2-8: A buckled wind turbine, caused by local buckling of the tower[Unknown].

1. **Wall thickness monopile**
   During pile-driving, high forces are exerted on the monopile. The magnitudes of these forces depend mainly on the soil properties. To prevent buckling during driving, a maximum diameter-wall thickness ratio of 120 is assumed to be able to prevent this.

2. **Angle conical parts**
   In the production process some restrictions are made on the angle of the conical parts. The conical parts are located at the top of the SUS and in between the interface and mudline if they are present. A minimum and a maximum angle of these parts are inputs of the optimization. In the current design process these are taken as 1° and 3°. In this research a maximum angle of 6° and no minimum angle is used, to check whether broadening of this constraint can give evident improvements of the design. This is checked in the end.

2-3 Optimization algorithm

This section explains an optimization strategy, by first having a closer look to the objective function. After this an algorithm is proposed with a formulation to include design constraints.

2-3-1 First characterisation of the objective function

As written in Section 1-2, Section 2-2-1 and based on expectations and experiences of the engineers at SWP, it is expected that several local optima exist. Other local optima are caused by the discrete number of sections with a minimum and maximum length. When changing the length of one of the parts of the support structure, when a plate is added or removed, there can be a discontinuity in the response surface. Due to these local optima, gradient-based solvers will generally lead to a local optimum, dependent on the starting point chosen.

Besides this, a property of the objective surface is known before the optimization starts, which can be used during the optimization: The mass has a minimum when the design parameters, representing a thickness or a diameter, are equal to zero. This results in a known gradient for these parameters if a design is feasible.

2-3-2 Algorithm

Based on this knowledge of the objective function, an optimization strategy has to be chosen. This strategy should have the ability to do a global optimization. A short introduction into optimization and an overview of several optimization methods and algorithms is given in Appendix A. The optimization strategy used in this thesis, is split up in two parts. The output of the first optimization algorithm is used as initial position for the second optimization algorithm as shown in Figure 2-9. So the first algorithm has to come up with an initial position that has a high chance to result in a global optimum after a local search by the Gradient Based Optimization (GBO) algorithm.

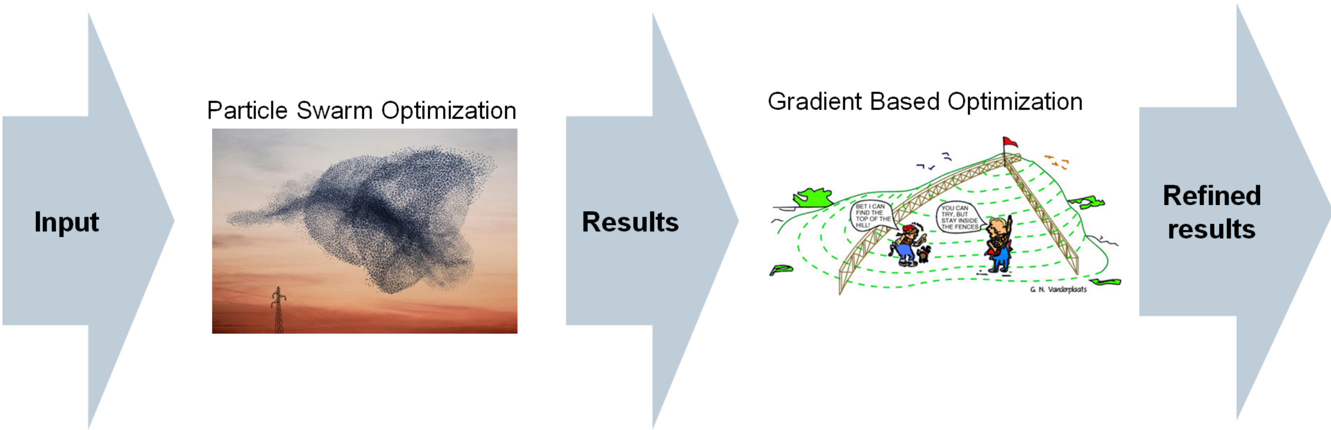


Figure 2-9: Overview of the two parts used by the algorithm (pictures by N. Castillon[2] and G. Vanderplaats[52]).

Initial algorithm (global search)
Jones [14] makes a clear statement concerning global optimization:

"A well known theorem by Torn and Zilinskas (1987)[46] *tells us that, in order to converge to the global optimum for a general continuous function, the sequence of iterates must be dense. Of course, this is a rather trivial method of convergence; in essence, it says that, in order to guarantee convergence to the global optimum, we must converge to every point in the domain. The practical lesson of the theorem is that any globally convergent method must have a feature that forces it to pay attention to parts of the space that have been relatively unexplored and, from time to time, to go back and sample in these regions."*

Within the group of direct search methods, for global optimization, most attractive are the algorithms which work according to Jones statement and claim to have a large chance to result in a global optimum (or give a good starting position for GBO). Now 3 algorithms are left: Simulated Annealing (SA) and, from the group of biologically inspired methods, Genetic Algorithm(GA) and Particle Swarm Optimization(PSO). Some properties of the algorithms can play an important role in the optimization:

- **Exploration full domain**: To find possibly a global optimum, evaluations spread over the full domain are needed.
- **Creativity/Randomness**: Having calculated certain positions, it can be needed to make large changes in the design to escape from a local optimum.
- **Insight in algorithm**: After the optimization is done, it can give good insights when there is a way to evaluate how the algorithm has come up with the optimum.
- **Add knowledge**: When knowledge about the optimum is available on forehand, it improves the chances to come up with a global optimum if these can be added.

Table 2-1: Comparison of three global optimization algorithms

Algorithm:	Exploration full domain	Creativity/ Randomness	Insight algorithm	Add knowledge	Parallel computing
SA	$-$	$+-$	$+-$	$+-$	$+-$
GA	$+-$	$+$	$-$	$-$	$+$
PSO	$+$	$+-$	$+$	$+$	$+$

– **Parallel computing**: Time is an important factor in optimization. A straightforward way to decrease the time an optimization takes is to use parallel computing.

In Table 2-1 the algorithms are compared based on these properties. When comparing these three possibilities, PSO seems to be most attractive for this study. This is caused by its global searching nature and good capabilities to add information to the algorithm based on available knowledge (as explained in Section 2-3-1).

Particle Swarm Optimization PSO is one of the swarm intelligence algorithms. It is based on a flock of birds or a shoal of fish, collecting food to survive. During the optimization a population, consisting a number of individuals n_i, is evolving over time, which is expressed in a number of generations p, with $p = 1$ till $p = n_g$. Every time step (generation) p the position of every individual i is updated with:

$$\boldsymbol{x}_{p+1,i} = \boldsymbol{x}_{p,i} + \boldsymbol{v}_{p+1,i}\Delta t. \tag{2-4}$$

$\boldsymbol{v}$ is the velocity of an individual:

$$\begin{aligned}\boldsymbol{v}_{p+1,i} = A_v\boldsymbol{v}_{p,i} + r_l B_v\frac{(\boldsymbol{x}_{b,l} - \boldsymbol{x}_{p,i})}{\Delta t} + r_{\text{ind}}C_v\frac{(\boldsymbol{x}_{b,i} - \boldsymbol{x}_{p,i})}{\Delta t} \\ + r_g D_v\frac{(\boldsymbol{x}_{b,g} - \boldsymbol{x}_{p,i})}{\Delta t} + r_g E_v\frac{(\boldsymbol{x}_o - \boldsymbol{x}_{p,i})}{\Delta t} + r_g F_v\frac{(\boldsymbol{x}_{b,\text{nb}} - \boldsymbol{x}_{p,i})}{\Delta t} + r_l G_v\boldsymbol{v}_{b,l}.\end{aligned} \tag{2-5}$$

In this equation, b stands for best position, nb for neighbour, o for origin, l for local (current) generation, and g for global, so for all 1 till p generations. r_l, r_g and r_o are randomly chosen numbers in between a r_{min} and 1 to reflect arbitrary behaviour of an individual. A_v, B_v, C_v, D_v, E_v, F_v and G_v, are constants to give weight to the different parts.

For the best individual of generation p, $\boldsymbol{v}_{p+1}$ is calculated in a different way:

$$\boldsymbol{v}_{p+1,i} = A_v\boldsymbol{v}_{p,i} + r_g I_v\frac{(\boldsymbol{x}_{b,g} - \boldsymbol{x}_{p,i})}{\Delta t} + r_g J_v\frac{(\boldsymbol{x}_o - \boldsymbol{x}_{p,i})}{\Delta t}. \tag{2-6}$$

I_v and J_v are again constants to indicate the importance of the different parts.

To prevent that particles cross the outer bounds of the design space, they bounce back with respect to the outer bounds. To prevent cycling, the bounce is handled as a partially inelastic collision.

Second algorithm
After the PSO has been done, a straightforward GBO is started. It uses the steps as shown in Table 2-2 to come up with a better design. Because the GBO is just able to find the local optimum, dependent on the starting point, the length ratios, w_1, w_2, w_3, s_1, s_2 and p_1 are fixed. In this way, the GBO becomes more simple.

2-3-3 Penalization of the constraints

First some notes are made on the combination of PSO, global optimization, the objective and the constraints:

Table 2-2: Overview of the steps during the GBO

#		Name	Description
0		Start	Get starting point from PSO
1		Gradient	Calculate gradient using Finite Differences
2:		Line search:	
	a	Initial step	Equation A-3
	b	Reduction	Till maximum step size < 10% design space
	c	Decrease/increase step	Using the Golden section ratio
	d	Repeat (2c)	Till $f(\boldsymbol{x}_n) - f(\boldsymbol{x}_{n-1}) < \Delta f_{\min}$
3		Repeat (1) and (2)	Till gradient ≈ 0 or $n_{iterations} > n_{\max}$

- For PSO, it is needed to have a single objective or fitness without constraints.
- According to the statement of Jones, evaluations through the full design space are required to be globally convergent.
- When using PSO, the objective surface has to exist over the full design space.

From these points, it is proposed to use the exterior penalty function method (Section A-2-1) to implement the constraints into the optimization.

As mentioned in Section A-2-1, it is important to scale the penalty functions in such a way, that the extra weight that is given to that design due to constraint violations is comparable with the real mass of the design. The way the penalty functions are formulated is not unique. The chosen implementation is based on efficient coding and is done in the following way:

- **Fatigue** can be checked by looking to the damage D or the utilisation ratio u_r. The relation between them is the SN-curve as shown in Figure 2-6. For the penalty calculation, the utilisation ratio is taken, because this gives a smoother penalty surface:

$$p_{f,\text{side}} = \frac{\sum_{u_{r,\text{side}} > u_{r,\text{side,max}}} (u_{r,\text{side}} - u_{r,\text{side,max}})}{n_{u_{r,\text{side}}}}, \tag{2-7}$$

$$p_f = p_{f,\text{inside}} + p_{f,\text{outside}}, \tag{2-8}$$

  with $n_{u_{r,\text{side}}}$ the number of fatigue checkpoints. For clarity, we will use the word 'damage' instead of 'utilisation ratio' in the remainder of this thesis.

- **Maximum deflection**. To demonstrate the ultimate stability of the support structure, 3 checks have to be done. A stability check where the convergence of the non-linear solver is checked with a high safety factor. A Characteristic deflection check where the deflection should not cross a certain maximum deflection. And a check on the derivative of the characteristic deflection to verify the stability of the deflection to the PPD. In the optimization, one check is used. This characteristic deflection check is for most design cases leading. It has 4 values which should not cross their maximum $d_{i,\max}$: the deflection and rotation at mudline and pile tip (d_1 till d_4):

$$p_d = \frac{1}{4}\left(\sum_{i=1}^{4} \max\left(0, \frac{d_i - d_{i,\max}}{d_{i,\max}}\right)\right). \tag{2-9}$$

  If the iterative solver for calculation of the non-linear deflection is not converged to a solution, $p_d = 1$

- **Buckling** is formulated in such a way, that the structure is expected to buckle if the buckling ratio b exceeds 1 (further explained in Section 2-4-3). The penalty function is formulated as follows:

$$p_b = \frac{\sum_{b>b_{\max}}(b - b_{\max})}{n_b}, \tag{2-10}$$

with n_b the number of buckling checkpoints.

- The **diameter-wall thickness** ratio can be calculated with:

$$r_{\mathrm{tD}} = \frac{D}{t}, \tag{2-11}$$

so then a penalty function can be formulated as:

$$p_t = \frac{\sum_{r_{\mathrm{tD}}>r_{\mathrm{tD,max}}} \frac{r_{\mathrm{tD}} - r_{\mathrm{tD,max}}}{r_{\mathrm{tD,max}}}}{n_{r_{\mathrm{tD}}}}, \tag{2-12}$$

with $n_{r_{\mathrm{tD}}}$ the number of wall thickness ratio checkpoints which is equal to the number of sections of the monopile.

- **Angle conical parts** A penalty function for this can be calculated with:

$$p_{\theta,i} = \max(0, \frac{\alpha_i - \alpha_{\max}}{\alpha_{\max}}), \tag{2-13}$$

$$p_\theta = p_{\theta,\mathrm{TW}} + p_{\theta,\mathrm{MP}}, \tag{2-14}$$

with α_i the angle of the conical part i and $\alpha_{\max}$ the maximum angle.

Final note It is important to see that for a constraint violation, all of these penalty functions give usually a value between 0 and 1. In this way, the effect of a constraint violation is easy to control. Just when a design is far from the feasible domain the penalties can become larger than 1. This happens for example, when the iterative solver on maximum deflection does not converge or when the fatigue damage or the buckling ratio over the full height exceeds its maximum.

Extended objective function

Combining the original objective function (Equation 2-2) and the penalty functions for the constraints (Equation 2-7 till Equation 2-13), the objective function becomes:

$$\hat{f} = m_{\mathrm{sus}} + m_{\mathrm{sus,c}} A_c(B_c\bar{p}_f + C_c\bar{p}_d + D_c\bar{p}_b + E_c\bar{p}_t + F_c\bar{p}_\theta + G_c\bar{p}_h). \tag{2-15}$$

In this, $m_{\mathrm{sus},c}$ is the mass of the support structure designed in the traditional way, as described in Section 1-1, with which the optimized structure is compared. This is used instead of m_{sus} because this is preferable when a derivative to $\hat{f}$ is calculated. In this study, $\bar{p}_i = (1 + p_i)^2 - 1 = p_i^2 + 2p_i$ is used to get a sharp transition in between feasible and infeasible designs (as show in Figure 2-10). Although this function will result in a more difficult gradient based optimization part due to a discontinuous derivative, this formulations is chosen because the accent will be on PSO. PSO has an advantage of this because the split between the feasible domain and the infeasible domain is more clear.

Why the optimum itself does not depend on the magnitudes of the penalties

In general in a multi-disciplinary optimization, the objective consists of several physical domains which are scaled by the weights of the penalties. This causes the final result of the optimization to be dependent on the weights of the penalties. In the formulation of the objective function as it is used in this thesis, the final goal of the optimization is to minimize the mass, while the design had to be feasible at the optimum design position. Because it has to be feasible at its final position, the penalties at that position should be equal to zero, which results in a design which is not directly dependent on the magnitude of the penalties.

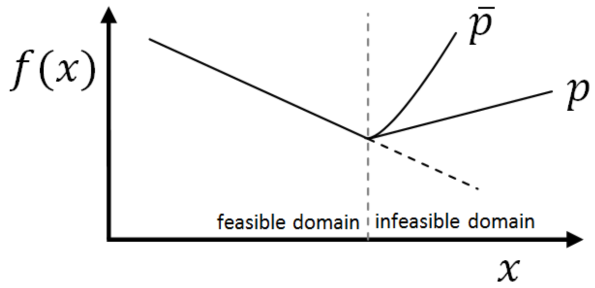


Figure 2-10: Difference in using p and $\bar{p}$ as penalty in the objective function.

Further characterization of the extended objective function
Using the extended objective function, two other sources of local optima arise compared to the objective function without the penalties. These are not due to the physical processes, but due to the modelling and the formulation of the penalties:

- A discrete number of sections is used per part. This number is calculated by rounding up $\frac{l_{\text{part}}}{l_{\text{max,section}}}$. In this way, while increasing or decreasing the length, points were the buckling of or the fatigue ration is calculated are added or removed which can cause discontinuities in the objective surface.
- When the iterative solver for the deflection cannot find an optimum, as mentioned $p_d = 1$. This causes at that point a discontinuity in p_d as function from the design parameters.
- As mentioned before, the penalties are added to the objective with $\bar{p}_i$. This causes the first derivative to be discontinuous.

2-3-4 Algorithm dependent parameters

In the algorithm, two groups of parameters need to be defined: The first group of parameters consists the number of generations and the number of individuals. The second group consists of the weights of the constraints, while the third group consists of the weights of the different fly directions. The latter two are both dependent on the generation the PSO is in. There is no unambiguous way for defining these parameters, therefore an iterative procedure was adopted. The approach is briefly explained in this section and some considerations of this iterative process are discussed.

Weights constraints ($A_c, ..., G_c$)
Factor A_c in Equation 2-15 is chosen in such a way that at the start, when a constraint is violated, the penalty is relatively small. Because the final design should not have any constraint violations, A_c is increasing during the PSO, as shown in Figure 2-11. In this way a violation of some constraint in the end causes a high penalty in the objective.

The other factors in Equation 2-15 are chosen in such a way, that the constraints on the conical parts, the MP thickness ratio and the maximum deflection, are less important than the constraints on fatigue and buckling. This is done because feasible designs for fatigue and buckling are more difficult to find than a feasible design for the other constraints. To the end they made equal, which is shown in Figure 2-12.

Weights fly directions ($A_v, ..., J_v$)
The weights of the fly directions as shown in Equation 2-5 and Equation 2-6, are very important and cause the algorithm to succeed or to fail in finding an optimum. The problem with these weights is that a fundamental reason how to choose them does not exist, because they are problem dependent. For example on the number of local optima and the smoothness of the objective surface. So the only way to choose for them is by trial and error, although some general statements can be made:

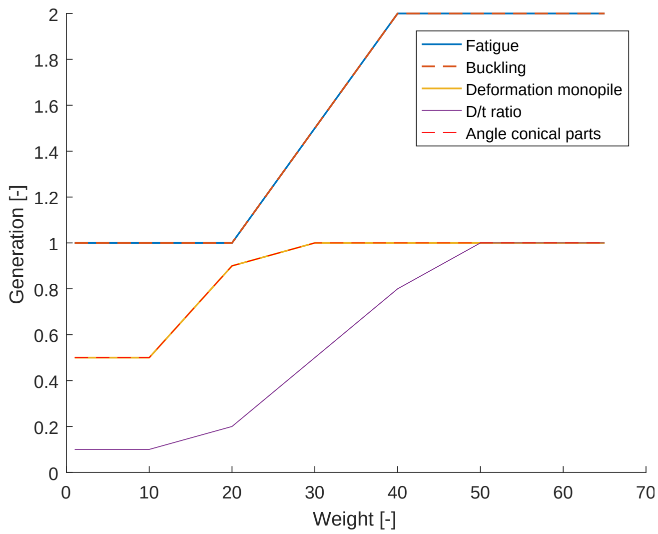


Figure 2-11: Chosen general weight for the constraints A_c.

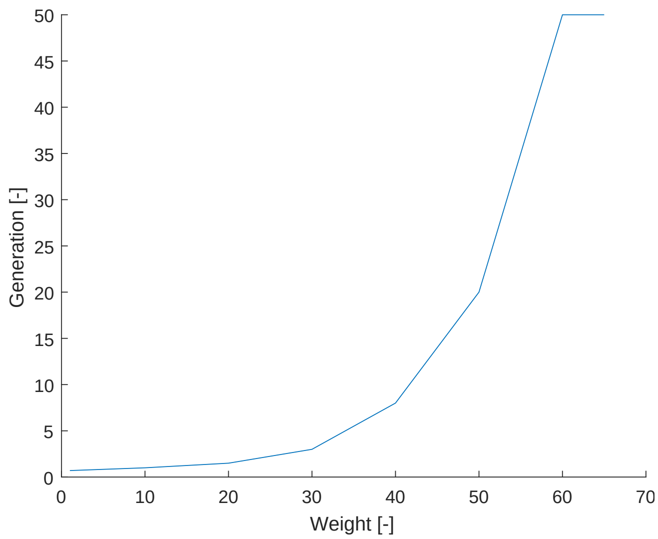


Figure 2-12: Chosen weights for the different constraints.

Local/global search (A_v and Δt) With Δt the step size between generations can be tuned. From start Δt is relatively high, so that an exploration of the design space is done. During the optimization it decreases so that it is below one at the end. This forces the individuals to converge in the direction they are pointed to. This can be controlled with A_v as well. Then Δt can be set to 1 for more insight in the progress, as explained in Chapter A Section A-1-2.

Neighbour/global focus (D_v and $x_{b,g}$ vs F_v and $x_{b,nb}$) In the beginning of the optimization it can be beneficially to do several more locally oriented searches. To do this, every individual uses the best individual from a part $\frac{1}{p_{\text{ind}}}$ of the total number of individuals which are the closest. In this way, depending on the spread of individuals over the design space, up to p_{ind} more local searches are done. Because the weight of the global best individual is increasing, these local searches go over to one global search.

Random behaviour (r_l, r_g, r_{ind}) Individuals have a certain freedom to make own decisions. This is represented in the random behaviour factors. As shown in Figure 2-13 the random behaviour factor (Rbhv) is decreasing to the end of the optimization.

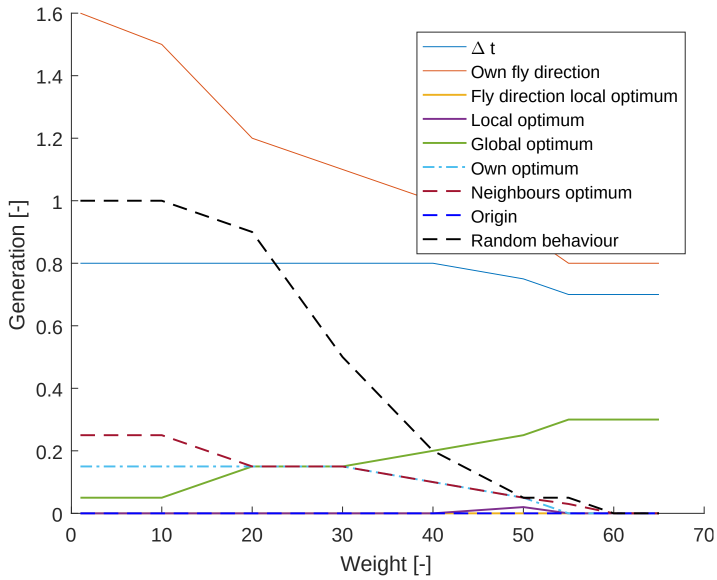


Figure 2-13: Chosen weights for the fly directions.

The weights of the flying direction of the local best individual, the position of the local best individual and the origin are set equal to zero. Investigation were pointing out that these did not necessarily improve the convergence.

Local best individual The local best individual is given other weights than the other individuals of its generation as shown in equation Equation 2-6. This is done to control its behaviour when it is in a global optimum (so when it just found a new optimum). In the start of the optimization (when the individuals are far away from each other), this individual takes a small step after finding an optimum, so that there is a chance to find a new optimum. To the end of the optimization in general step sizes become smaller. Because individuals are close to each other, it is more likely that an other individual finds the next optimum. To prevent too fast convergence at that moment, more weight is given to the local bests fly direction, so that it makes a relatively large step, which prevents unnecessary fast convergence of individuals.

Number of individuals n_i and number of generations n_g
The PSO has to converge to an optimum to the end of the optimization. Although several options exist to stop the PSO, it is chosen to use a maximum number of generations. Another common

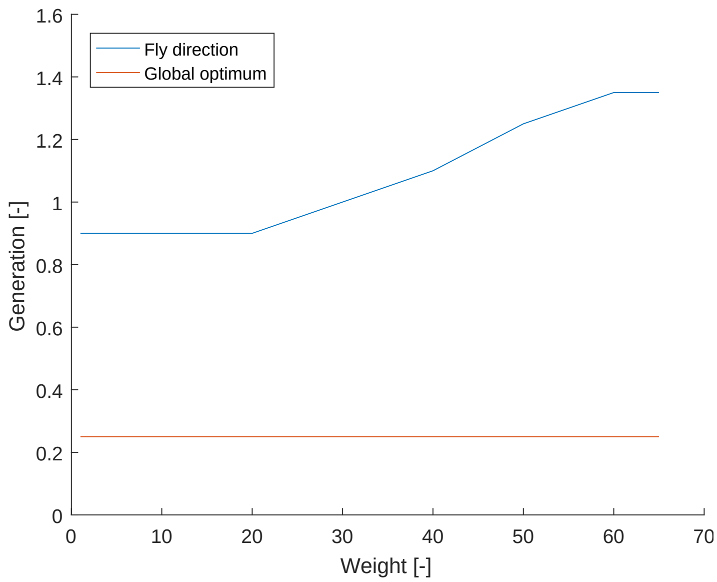


Figure 2-14: Chosen weights for the fly directions for the local best individual.

option is to stop if the objective f has not changed more than a certain criterion during a number of generations $n_{\text{criterium}}$: $f_{best}(x_n) - f_{best}(x_{n-n_{criterion}}) < \Delta f_{Min}$. Because GBO uses this criterion also and can do this more effectively, it is not used for PSO. n_g depends on the weights of the fly directions. After the weights were defined, test runs were done until the influence of GBO on the final result was become small. This means that the PSO found a position close to an optimum. In this optimization 65 generations are used. The number of individuals has to be set before the optimization starts. The more individuals are used, the larger the chance on a global optimum. This optimization study uses initially 90 individuals, but investigates the effect of the number of individuals for most optimization cases.

2-4 Overview computational steps

Figure 2-15 gives an overview for all computational steps that are needed to evaluate the objective. Now all important steps in Figure 2-15 are discussed. For brevity, just the basic idea is explained. Sometimes some references are given, where the interested reader can find background information. The computational steps can be split up in 3 parts which are discussed now consecutively: the set-up of the model, the calculations and the analysis.

2-4-1 Set-up model

Dimensions

From the design variables $\boldsymbol{x}$, all dimensions of the support structure are calculated. This is done with the parametrization as explained in Section 2-2-4. The dimensions are used for the construction of the structural model, the analytical analysis of fatigue and buckling and for evaluating the geometrical constraints.

PY-curve

With the diameter below mudline, PY-curves can be constructed for a certain number of heights. A PY-curve gives a relation between the deflection of the pile in the soil and the pressure on the pile. The method that is described in the guideline [7] for the construction of the curves, was first published by Matlock [24]. Some corrections are applied for specific soil types as clay and sand.

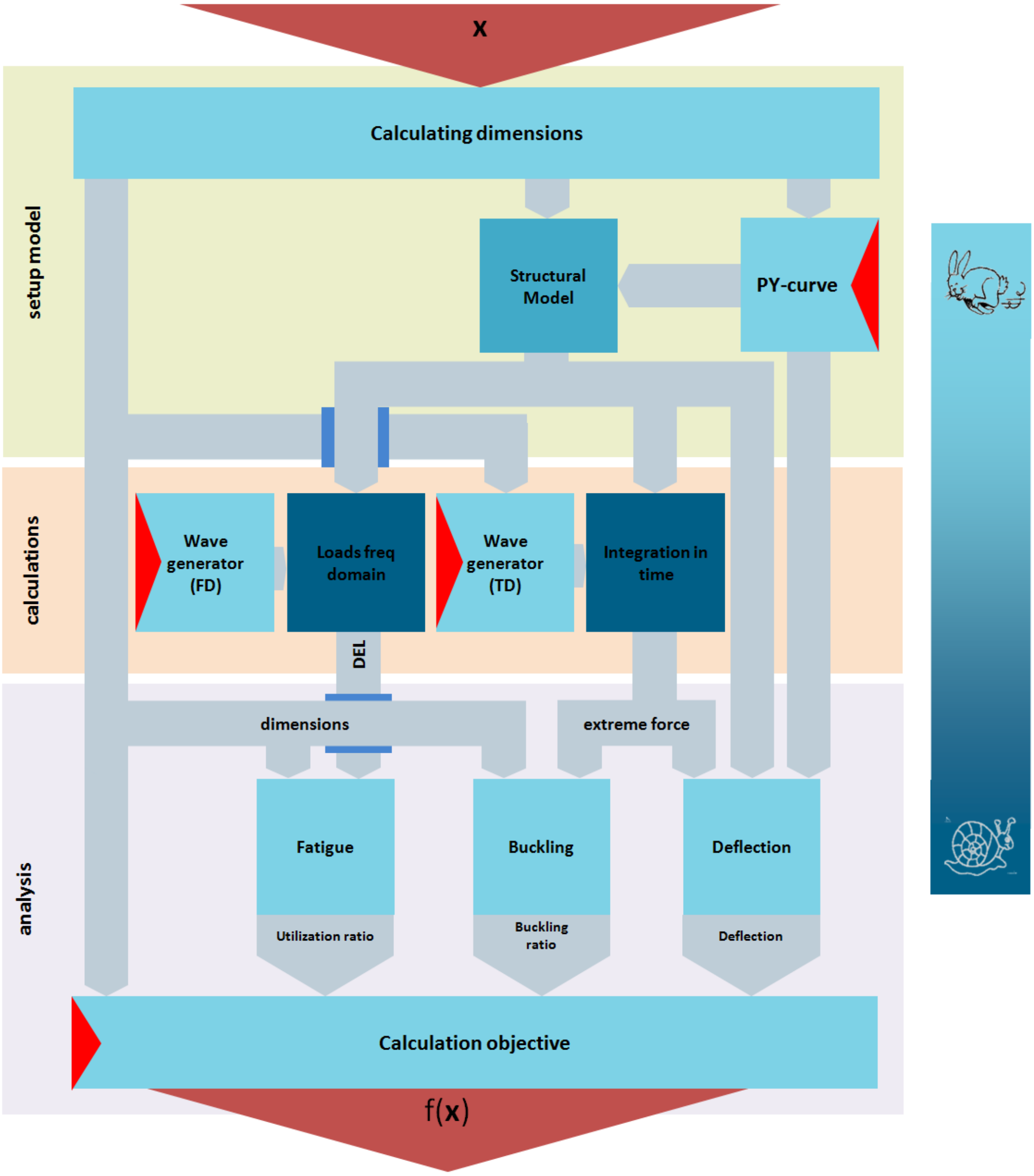


Figure 2-15: Overview of the computational steps to calculate the objective $f(x)$. An indication of the calculation time is given by the intensity of the blue colour, as shown on the right. The red triangles indicate where important (site specific) input is given into the optimization

Structural model
The structural model constructs the numerical model from the support structure. It uses Timoshenko-beam elements to build a FEM-model of the support structure. Besides this, it adds some additional masses:

- A concentrated mass, representing the RNA.
- Several concentrated masses which result from smaller components (flanges, boat landings).
- It adds spring elements, derived from the PY-curves, to represent the soil. The soil damping is kept constant and added via the structural (modal-)damping.

These additional masses are kept constant during the optimization. When its needed, the height is made design dependent.

It gives as a result the structural matrices from the assembly. These are used for the load calculations in both time and frequency domain. Besides this, the structural matrices are used for finding a non-linear static solution for the deflection test.

With the structural model, the eigenmodes and eigenfrequencies are calculated which are used for the load in frequency domain.

2-4-2 Calculations

In this phase, some calculations are done in a numerical way.

Wave generator
To calculate wave loads on a monopile-founded support structure, Morison equation is widely used. It assumes that the movement of the structure does not influence the wave loads. Wave loads are needed for both the ULS and the FLS case. The ULS loads are calculated in the time-domain, while the FLS loads are calculated in the frequency domain.

Time domain Every iteration a new wave load is created which depends on the dimensions of the structure. So the surface of the sea and the geometry of the structure are used as input, to come up with a time series of a load distribution as output. These are used to calculate the response of the structure in time.

Frequency domain To describe the waves in the frequency domain, a JONSWAP (Joint North Sea Wave Observation Project) spectrum is used. Within this spectrum every sea state can be described by a peak height H_s and a certain period of the wave T_s. Via measurements this is related to a certain occurrence with a certain wind speed. The latter is used to relate aerodynamic damping to the weather circumstances. With these energy density spectra of the waves, the load frequency spectra (over the height) can be derived as well. When linearising the drag force, Morisons equation can be used in the frequency domain. To take diffraction into account, MacCamy-Fuchs diffraction correction is applied.

Seidel [35][36] has described an accurate way to reduce the number of load cases that are necessary to be evaluated. This is done by looking to the relative spectral energy at the first eigenfrequency.

Loads in frequency domain
From the structural properties, the frequency response of the structure can be calculated. This can be combined with the wave loads in the frequency domain, to obtain a load response spectrum of the structure. From this, the fatigue loads can be derived using Dirliks method[5] (more detailed explanation of this method is given in Appendix D). This is an equivalent of rain-flow counting in the time-domain. This results in Damage Equivalent Loads (DELs). The damage these DELs cause for a pre-scribed number of cycles is equivalent to the damage resulting from all different load cases.

Wind and wave loads can be superimposed using a representative misalignment[37][22]. The influence of most design variables on the wind loads is assumed as small. An exception is made for the length of the tower. Because the wind loads behave very linear, a simple scaling is used to predict the wind loads of a tower with the length l based on the calculated wind loads from a tower with length $l_{\min}$. The moment line above SWL is linearly extrapolated, in such a way that the maximum bending moment is increased with $\frac{l}{l_{\min}}$. The shear force for the length above $l_{\min}$ is kept equal to the shear force at $l_{\min}$. These scaling is based on rules of thumb used within SWP CoC.

Loads in time domain

With the extreme loads from the wave generator in time domain, the response of the structure can be calculated. This is done with a Newmark time-integration scheme. It calculates time series for both deformations and loads in the structure. To these extreme loads by an extreme wave, wind loads are added up.

As for the loads in the frequency domain, the wind loads are assumed as constant except for the length of the tower. Linear extrapolation for the length of the tower above $l_{\min}$ is used for both bending moment and shear force to take increased wind loads into account due to a taller tower.

2-4-3 Analysis

In this phase, the numerical calculated results are used for some (analytical) analysis.

Fatigue

When the DELs are known, the fatigue life of the structure is checked according to the internal Design Brief of SWP[54]. This is according to the guidelines provided by the certification body DNV-GL To get a representative design, a design independent Stress Concentration Factor (SCF) for the door is used. This is taken equal to the SCF as used in the initial design.

Buckling

With the extreme forces in the structure known, an analytical buckling check on the structure is done according to the guidelines provided by DNV-GL[6] and EN[43]. To prevent overly conservative designs, it is assumed that the maximum length between flanges is 30 meter.

Deflection

As explained in Section 2-3-3, the maximum deflection and rotation of pile head and pile tip are criteria. These are calculated by Newton-Raphson iterations to find the non-linear static solution for the extreme load resulting from the time-integration. The non-linearity is resulting from the non-linear PY-curves. During the time integration these were linearized.

Calculation objective

As last step, the geometrical constraints are evaluated. Then the objective is calculated with Equation 2-15. Penalties are calculated with Equation 2-7 till Equation 2-13.

2-4-4 Computational time

The computational time for the evaluation of one individual is shown in Table 2-3.

Dependent on the way of lumping sea states during the calculation of the DEL, the time it takes to evaluate one individual is in between 0.8 and 7 minutes. Early investigations showed that it is needed to use around 120 individuals in 65 generations to find an optimum for a relatively easy OWT optimization run (diameter and length of the tower are fixed). Then a full optimization takes about $120 \cdot 65 \cdot 0.8\text{min/iteration} \approx 6240$ minutes which is about 104 hours and about 4.3 days (with a fast calculation of the DELs). During this time, the algorithm is about 83 hours in the calculation phase

Table 2-3: Indication of the computational times, when the optimization is done using a single core with a clock speed of 2.6 GHz.

Step:	Domain:	Calculation time Minimum:	Maximum:
Calculating DELs[2]	Frequency	$\pm$ 30 s	$\pm$ 420 s
Calculating F_{extr} [3]	Time	$\pm$ 10 s	$\pm$ 10 s
Other calculations and evaluations	-	$\pm$ 10 s +	$\pm$ 10 s +
Total	-	$\pm$ 50 s	$\pm$ 440 s

and 21 hours in the set-up and analysis phase. Besides this, the quality of the found optimum by PSO is up to a certain level dependent on the number of individuals and the number of generations that are used. So using more individuals and generations can lead to a better optimum.

Summary

In this chapter an optimization strategy is given. PSO is proposed to be used as initial optimization algorithm, followed by a gradient based optimizer to find a local optimum. A disadvantage of PSO is that the result, to a certain extent, depends on the number of individuals and generations. An overview of the computational steps is given, with an indication of the computational time an optimization takes when using load calculations. To do a more time and computationally efficient optimization, the following chapters are about the estimation of loads. Next chapter introduces the theory of several estimation methods, while Chapter 4 uses these theories for load estimations.

[2]Depends on the chosen way of lumping, definition of the aerodynamic damping

[3]This is for one cases with zero initial conditions, which is assumed to give representative loads for the initial design. To improve this, the mean of several simulations with the extreme wave added to a random sea state has to be taken.

Chapter 3

Theory of Approximation Methods

While running simulations in engineering practice, calculations leading to precise and accurate results are most useful. For numerically computed results this can be improved, to a certain limit, by refining the discretization. Drawback of this is a higher computation time. If a computation has to be done once, this is not necessarily a problem. When it has to be done several times, this can become a problem. An example of this is optimization, when more design iteration can lead to better results. In such cases it can be beneficial to prevent long calculations, by estimating their output. While using results of an estimator it is important to be aware of losses in accuracy and precision.
To align with literature and explain some terms used in this thesis, this chapter gives first an introduction into some terminology used for estimation methods. Then several methods are introduced. Based on these theories, the general properties, the computation and the inputs of the methods are compared. The chapter concludes with a short review on multivariate output.

3-1 Terminology estimation methods

For models that try to reproduce values of expensive calculations, several names exists in literature. For example:

- Metamodels
- Response surface models[1]
- Estimation methods
- Surrogate models
- Emulators
- Auxiliary models

In this thesis, the term ‘Estimation methods’ is chosen. The combination of outputs of the estimation methods, is referred to as the ’Response Surface’. Besides these, some other important terms are used:

- **Training points**: These are the input points while constructing the estimator. Every point consists of data on which the estimation is based and data that had to be estimated.

[1] It is important to distinguish this name from the estimation method called ’Response Surface Methodology’. This method is explained in Section 3-2 in more detail.

- **Check points**: These are used as a second set of input points while constructing the estimator. Free input parameters needed for the construction of the estimator are chosen in such a way that the cumulative error between the calculated values on the check points and the estimated values on these points is minimal. So the estimator itself is not based on these points.
- **Test points**: These points are used to give a quantitative indication of the accuracy of the estimator. The accuracy is expressed in terms of the performance or the correlation coefficient. These are calculated by comparing the calculated values and the estimated values on the position of the training points.
- **Interpolating**: Two sorts of estimation methods exist. Methods that pass through all training points and methods that do not. In literature several names to classify this exist: Interpolating and non-interpolating[14], strictly and non-strictly interpolating[18] and exact and non-exact interpolating[40]. This thesis uses the names of Jones: Interpolating and non-interpolating. An important property from non-interpolating methods for optimization is that adding a new training point will not necessarily lead to a more accurate surface. Jones [14] gives an elaborate explanation of this.
- **Shape preserving**: A shape preserving estimator will predict the shape of the response surface very accurately, but does not necessarily go through the data points.
- **Overfitting**: This occurs when the model has too many free parameters and it fits the data at too fine a scale. A result of this is that the estimator will predict noise which is superimposed to the underlying behaviour[9].
- **Basis-functions**: In several estimation methods, basis-functions are used. These functions are spread around a point to share information of that point with its surroundings. Several basis-functions exists. In Table 3-1 the most common basis-functions are shown. Note that in all functions one parameter θ or a vector with parameters $\boldsymbol{\theta}$ is added. These are generally chosen while constructing the estimator. When it is used, the basis-function is called a parametric basis-function, when it is fixed the basis-function is called a fixed basis-function. Note that θ is influencing the behaviour much more for the Gaussian, Multi-quadratic and Inverse Multi-quadratic functions than for the other functions. For the Kriging basis-function a vector with parameters $\boldsymbol{p}$ is added as well. These need to be defined before the construction of the estimator.

Table 3-1: Overview of several Basis-functions

Name	Function	Notes
Linear	$\psi(r,\theta) = \theta r$	
Quadratic	$\psi(r,\theta) = \theta r^2$	
Cubic	$\psi(r,\theta) = \theta r^3$	Minimizes [2] $\int \frac{\partial^2 \hat{y}}{\partial x^2}^2 dx$
Thin plate spline	$\psi(r,\theta) = \theta r^2 \ln r$	
Gaussian	$\psi(r,\theta) = e^{-\frac{r^2}{2\theta^2}}$	Intuitive statistical motivation
Kriging	$\psi(\boldsymbol{r},\boldsymbol{\theta},\boldsymbol{p}) = e^{-\sum_{i=1}^{n} \theta_l \lvert r_l \rvert^{p_l}}$	Is related to the Kriging estimation method
Multi-quadratic	$\psi(r,\theta) = \sqrt{r^2+\theta^2}$	
Inverse Multi-quadratic	$\psi(r,\theta) = \frac{1}{\sqrt{r^2+\theta^2}}$	

- **Curse of dimensionality**: When increasing the dimensions of a problem, generally the complexity of the problem increases as well. This can cause problems during optimization, but even for constructing an estimator method[19]. Wang and Shan[53] identify this as the most prominent problem for Metamodel-Based Design Optimization.

[2] Compared to all other interpolating functions for which this integral exists and is finite.[14]

Table 3-2: Notation of input and outputs of estimations as used in this thesis.

Description:	Univariate:	Multivariate:
Input:	$\boldsymbol{x}^{(i)} = \begin{bmatrix} x_1^{(i)} & x_2^{(i)} & ... & x_k^{(i)} \end{bmatrix}^T$	$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}^{(1)} & \boldsymbol{x}^{(2)} & ... & \boldsymbol{x}^{(n)} \end{bmatrix}$
Calculated function:	f	$\boldsymbol{f}(\boldsymbol{X}) = \begin{bmatrix} f(\boldsymbol{x}^{(1)}) & f(\boldsymbol{x}^{(2)}) & ... & f(\boldsymbol{x}^{(n)}) \end{bmatrix}^T$
Calculated output:	y	$\boldsymbol{y}(\boldsymbol{X}) = \begin{bmatrix} y(\boldsymbol{x}^{(1)}) & y(\boldsymbol{x}^{(2)}) & ... & y(\boldsymbol{x}^{(n)}) \end{bmatrix}^T$
Estimated function:	$\hat{f}$	$\hat{\boldsymbol{f}}(\boldsymbol{X}) = \begin{bmatrix} \hat{f}(\boldsymbol{x}^{(1)}) & \hat{f}(\boldsymbol{x}^{(2)}) & ... & \hat{f}(\boldsymbol{x}^{(n)}) \end{bmatrix}^T$
Estimated output:	$\hat{y}$	$\hat{\boldsymbol{y}}(\boldsymbol{X}) = \begin{bmatrix} \hat{y}(\boldsymbol{x}^{(1)}) & \hat{y}(\boldsymbol{x}^{(2)}) & ... & \hat{y}(\boldsymbol{x}^{(n)}) \end{bmatrix}^T$

3-2 Explanation of several methods

Classification of the estimation methods

Based on the underlying background, Turner[48] defines three classes of estimation methods:

- Geometric models: are used originally to fit a curve through a data set and to visualize geometric models in 2D or 3D (in for example CAD packages).
  - (Non-linear) Response Surface Models
  - Multi-adaptive Regression Splines
  - Non-uniform Rational B-splines
- Stochastic models: assume a stochastic component in the data.
  - Radial Basis Functions
  - Kriging
- Heuristic models: try to find underlying patterns in the data.
  - Support Vector Machines
  - Neural networks

Several studies have compared estimation methods. Turner [48] gives a good overview of many these comparisons. In the end it can be concluded that the performance of a method is quite case specific and therefore no approximation methods can be excluded on forehand. Now the theoretical backgrounds of the estimation methods are explained. Multi-adaptive Regression Splines, Non-uniform Regression B-Splines, Support Vector Machines and Neural Networks are not explained in full detail, but far enough to understand why they are not used in this project. For the explanation of the methods the notation as shown in Table 3-2 is used. Note that $\boldsymbol{x}^{(i)}$ for $i = 1, ..., n$ is representing n training points, while $\boldsymbol{x}$ is representing a new position in k dimensions.

3-2-1 Polynomial fit (Response Surface Methodology (RSM))

When all inputs of a certain function f are known, the output y of this function can be expressed as:

$$y = f(x_1, x_2, ..., x_k), \tag{3-1}$$

in which x_1 till x_k are the inputs. For some region of interest, the output can be assumed as linear in the variables (based on a Taylor series expansion) with:

$$\hat{f} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k. \tag{3-2}$$

When more curvature is present in the function f, this can be extended to higher orders, although orders higher than 5 should preferably not be used to prevent overfitting. For clarity, from now on the Response Surface Methodology is shown in vector notation:

$$\hat{f} = \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{w} + b. \tag{3-3}$$

In which $\boldsymbol{p}(\boldsymbol{x})$ is the vector filled with inputs resulting from the polynomial order and cross-terms chosen, $\boldsymbol{w}$ is the vector which gives weights to the input parameters and b is taken as a constant. In general b is taken as the mean of all training input .

Myers[27] defined a few fundamental assumptions for this method:

1. A complicated or unknown function $y = f(x_1, x_2, ..., x_k)$ exists of which the input variables are **quantitative** and **continuous**.
2. The function f can be approximated in the region of interest by a **low-order polynomial**.
3. The **independent variables** x_1, x_2, ..., x_k are controlled in the observational process and **measured with negligible error**.

Knowing this, when the dependencies are left away, the function f shows:

$$f = \hat{f} + \varepsilon = \boldsymbol{p}^T\boldsymbol{w} + b + \varepsilon. \tag{3-4}$$

When $\hat{\boldsymbol{y}}$ is a vector with n estimations of the vector $\boldsymbol{y}$:

$$\boldsymbol{y} = \hat{\boldsymbol{y}} + \boldsymbol{\varepsilon} = \boldsymbol{P}^T\boldsymbol{w} + b\mathbf{1} + \boldsymbol{\varepsilon} = \boldsymbol{P}^T\boldsymbol{w} + \boldsymbol{\varepsilon}. \tag{3-5}$$

In this equation $\boldsymbol{P}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{p}_1(\boldsymbol{x}) & \boldsymbol{p}_2(\boldsymbol{x}) & ... & \boldsymbol{p}_n(\boldsymbol{x}) \end{bmatrix}^T$ and the mean b of all output data points y is assumed to be equal to 0. Minimization of $\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}$ for $\boldsymbol{w}$ gives:

$$\boldsymbol{w} = (\boldsymbol{P}^T\boldsymbol{P})^{-1}\boldsymbol{P}^T\boldsymbol{y}. \tag{3-6}$$

Note that the number of known data points n, has to be higher than the order of the polynomial including its cross-terms, to prevent free variables. This is mostly a problem when there is a large number of inputs, when using most cross terms of a higher order polynomial.

Non-linear Response Surface Methodology (nRSM)

Instead of the polynomial functions with a degree ≤ 1 used for Response Surface Methodology, other functions can be used also. An important point when creating an estimator with non-linear Response Surface Methodology is that it is needed to know the non-linear function on forehand, including their offset or phase-shift. Besides this, the optimization problem to find the coefficients in $\boldsymbol{w}$ is not linear any more.

3-2-2 Radial Basis Functions (RBF)

A complicated function with unknown properties which represents some physical effects, can often be expressed as a weighted superposition of several other functions, with known properties. When having a certain value y on a point $\boldsymbol{x}$ in a space with dimension k, in the direct neighbourhood of the sampling point two things can be assumed when there are no discontinuities:

- The value will be close to the known value y.
- The influence of the sampling point on its neighbourhood decreases with the distance r from that point.

As estimator, this method is called Radial Basis Functions. Mathematically this can be expressed as:

$$\hat{y}(\boldsymbol{x}) = \sum_{i=1}^{n} \psi(f(\boldsymbol{x}, \boldsymbol{x}^{(i)}))w_i = \boldsymbol{\psi}^T(\boldsymbol{x}, \boldsymbol{X})\boldsymbol{w}, \tag{3-7}$$

in which $f(\boldsymbol{x}, \boldsymbol{x}^{(i)})$ is a distance, in RBF generally taken as the Euclidean distance $r = ||\boldsymbol{x} - \boldsymbol{x}^{(i)}||$, n the number of training points so that $\boldsymbol{x}^{(i)}$ represents the position of training point i and where $\boldsymbol{X}$ represents the positions of all training points. The function ψ is called the basis-function and has two variables: The shape of the basis-function and the free parameter(s) in it.

When a basis-function and its parameters are chosen, the next step is to come up with the unknown $\boldsymbol{w}$. A training data set containing n data points with the belonging values $\boldsymbol{y}$ can be put in Equation 3-7:

$$\boldsymbol{\Psi}(\boldsymbol{X})\boldsymbol{w} = \boldsymbol{y}. \tag{3-8}$$

This equation shows the linear relationship between $\boldsymbol{\Psi}(\boldsymbol{X})$ and $\boldsymbol{y}$. In literature, $\boldsymbol{\Psi}(\boldsymbol{X})$ is better known as the Gram-matrix and is defined as:

$$\psi_{i,j}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \psi(||\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}||). \tag{3-9}$$

This system of equations can be solved for $\boldsymbol{w}$:

$$\boldsymbol{w} = \boldsymbol{\Psi}(\boldsymbol{X})^{-1}\boldsymbol{y}. \tag{3-10}$$

Calculation of $\boldsymbol{\Psi}^{-1}$ is needed. Whether this is possible of not, depends on the used basis-functions and training points. For example when two points are close to each other, it is possible that $\boldsymbol{\Psi}$ becomes ill-conditioned, which causes failure of the method during the Cholesky factorization[9].

When $\boldsymbol{w}$ is known, a new estimation is done by:

$$\hat{y}(\boldsymbol{x}) = \boldsymbol{\psi}(\boldsymbol{x})^T\boldsymbol{w}, \tag{3-11}$$

using the same basis-functions in $\boldsymbol{\psi}$ as they were used for the calculation of $\boldsymbol{w}$. Instead of estimating the full amplitude of y, it is also possible to use a constant or a low order polynomial as basis ($\hat{\mu}$). Then $\boldsymbol{w}$ can be calculated with:

$$\boldsymbol{w} = \boldsymbol{\Psi}(\boldsymbol{X})^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu}) \tag{3-12}$$

where $\mathbf{1}$ is a $1 \times n$ column vector, so a vector with a size equal to the size of $\boldsymbol{y}$. Now a new estimation can be done via:

$$\hat{y}(\boldsymbol{x}) = \hat{\mu} + \boldsymbol{\psi}(\boldsymbol{x})^T\boldsymbol{w}. \tag{3-13}$$

In Equation 3-13, for a new estimation, just $\boldsymbol{\psi}(\boldsymbol{x})$ has to be calculated. $\hat{\mu}$ and $\boldsymbol{w}$ are calculated during the construction of the estimator.

Radial Basis Functions and Noisy data

When having noisy data, the need to interpolate between data has gone. A method to distinguish between the underlying response and noise is by the introduction of the regularization parameter λ [31]. This is added to the main diagonal of the Gram-matrix. This results in:

$$\boldsymbol{w} = (\boldsymbol{\Psi} + \lambda\boldsymbol{I})^{-1}\boldsymbol{y}. \tag{3-14}$$

When the variance of the response data is known, λ should be set to this variance[9]. But since this is generally not known, a good option is to add it to the list of parameters that need to be estimated independent of building the estimator. With $\lambda \neq 0$, by definition $\boldsymbol{\psi}$ for the input of a training point is not a column or row from $\boldsymbol{\Psi}$ any more. This causes the estimator to not be interpolating.

3-2-3 Kriging (KRI)

Kriging is a method which is originally used for mining. The spatial distribution of a certain natural resource can be modelled via a probability as if the samples were taken from a random process. This concept was first introduced by Danie Krige [21] and further developed as method by the French mathematician and founder of Geostatistics, Matheron. The standard work for Kriging as estimation method is from Sacks et al[34].

For the explanation of Kriging, two derivations exists. Sacks et al. uses the most common derivation which shows the predictor to be the Best Linear Unbiased predictor (BLUP). In this thesis, Kriging is explained via a more intuitive approach which is used by Jones [14]. In the end, an analogy with the Radial Basis Functions estimation method is shown.

Assume a set of training data points $\boldsymbol{X}$ with a response $\boldsymbol{y}$, which is assumed to be the result of a stochastic process with μ as mean of $\boldsymbol{y}^{(i)}$.

Because the response is said to be result of a stochastic process, the correlation between the result of two training points can be expressed as function of their inputs. For example, when using the Kriging distribution function (looks like the Gaussian distribution function):

$$\mathrm{cor}(Y(\boldsymbol{x}^{(i)}), Y(\boldsymbol{x}^{(j)})) = \psi(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = e^{-\sum_{l=1}^{n} \theta_l |x_l^{(i)} - x_l^{(j)}|^{p_l}}. \tag{3-15}$$

So every dimension has its own distribution with parameters θ_l and p_l (the meaning of these parameters is explained later on). When $\boldsymbol{x}^{(i)} = \boldsymbol{x}^{(j)}$ the correlation is 1 and the output will be equal as well. And when $||\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}|| \to \infty$ the correlation will drop to zero and the output will be fully independent of each other. Defining the correlation matrix $\boldsymbol{\Psi}$ as the correlation between all sample points, it can be written as:

$$\boldsymbol{\Psi}(\boldsymbol{y}, \boldsymbol{y}) = \begin{bmatrix} \mathrm{cor}(f(\boldsymbol{x}^{(1)}), f(\boldsymbol{x}^{(1)})) & \mathrm{cor}(f(\boldsymbol{x}^{(1)}), f(\boldsymbol{x}^{(2)})) & ... & \mathrm{cor}(f(\boldsymbol{x}^{(1)}), f(\boldsymbol{x}^{(n)})) \\ \mathrm{cor}(f(\boldsymbol{x}^{(2)}), f(\boldsymbol{x}^{(1)})) & \mathrm{cor}(f(\boldsymbol{x}^{(2)}), f(\boldsymbol{x}^{(2)})) & ... & \mathrm{cor}(f(\boldsymbol{x}^{(2)}), f(\boldsymbol{x}^{(n)})) \\ ... & ... & ... & ... \\ \mathrm{cor}(f(\boldsymbol{x}^{(n)}), f(\boldsymbol{x}^{(1)})) & \mathrm{cor}(f(\boldsymbol{x}^{(n)}), f(\boldsymbol{x}^{(2)})) & ... & \mathrm{cor}(f(\boldsymbol{x}^{(n)}), f(\boldsymbol{x}^{(n)})) \end{bmatrix}. \tag{3-16}$$

From Equation B-14 in Appendix B can be derived that:

$$\mathrm{Cov}(\boldsymbol{y}, \boldsymbol{y}) = \sigma^2 \boldsymbol{\Psi}. \tag{3-17}$$

Given $\boldsymbol{X}$ and $\boldsymbol{y}$, the unknowns in Equation 3-17 are μ, σ, $\boldsymbol{p}$ and $\boldsymbol{\theta}$. $\boldsymbol{\theta}$ and $\boldsymbol{p}$ are vectors with a length equal to the number of dimensions in the input $\boldsymbol{x}^{(i)}$. To estimate these it is assumed that $\boldsymbol{y}$ is the results of a deterministic process and so no errors are in this data.

When assuming a Gaussian/normal distribution with mean μ and standard deviation σ, the likelihood of a certain model can be expressed with:

$$L(\boldsymbol{y}|\mu, \sigma) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} e^{-\frac{\sum_{l=1}^{n} (y_i - \mu)^2}{2\sigma^2}}, \tag{3-18}$$

using the sampled data gives the so called likelihood function:

$$L = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}} |\boldsymbol{\Psi}|^{\frac{1}{2}}} e^{-\frac{(\boldsymbol{y}-\boldsymbol{1}\mu)^T \boldsymbol{\Psi}^{-1} (\boldsymbol{y}-\boldsymbol{1}\mu)}{2\sigma^2}}. \tag{3-19}$$

This likelihood function can be maximized to come up with a model that is as likely as possible. To simplify this maximization, the natural logarithm is taken. This is also called, the ln-likelihood function:

$$\ln L = -\frac{n}{2} \ln(2\pi\sigma^2) - \frac{1}{2} \ln(\sigma^2) - \frac{1}{2} \ln |\boldsymbol{\Psi}| - \frac{(\boldsymbol{y} - \boldsymbol{1}\mu)^T \boldsymbol{\Psi}^{-1} (\boldsymbol{y} - \boldsymbol{1}\mu)}{2\sigma^2}. \tag{3-20}$$

To find the maximum, the derivatives of σ and μ can be set equal to 0, which gives the maximum likelihood estimations (MLE) for μ and σ in terms of $\mathbf{\Psi}$ and $\boldsymbol{y}$. Substitution of these in Equation 3-20 and leaving out constant terms, gives the concentrated ln-likelihood function:

$$\ln(L) \approx -\frac{n}{2}\ln(\hat{\sigma}^2) - \frac{1}{2}\ln|\mathbf{\Psi}|. \tag{3-21}$$

This function needs to be maximized for $\boldsymbol{\theta}$ and $\boldsymbol{p}$(when using the Kriging basis-function) which are parameters in $\mathbf{\Psi}$. $\boldsymbol{p}$ can cause the surface to be very smooth for a relatively high value of p (for example $p = 2$) or can have sharp peaks for a relatively low value of p (for example $p = 0.5$) this is shown in figure Figure 3-1. According to literature[18], $\boldsymbol{p}_l$ can be set to 2 if the surface and its derivatives

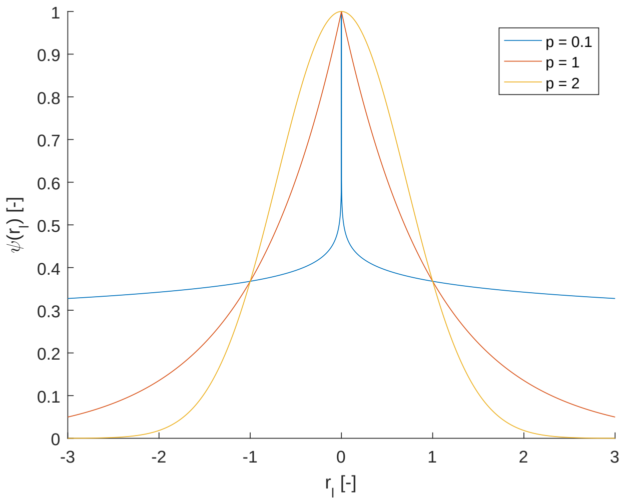


Figure 3-1: The Kriging basis-function ψ as function of r_l for several values for parameter p[9].

are continuous. This has to be done before the optimization of the likelihood starts, because $\boldsymbol{p}$ is influencing the general shape of the problem. As Equation 3-21 cannot be differentiated directly, numerical optimization techniques are generally used to find $\boldsymbol{\theta}$.

When σ, μ, $\boldsymbol{\theta}$ and $\boldsymbol{p}$, and so the estimator is defined, the goal is to estimate a new value $\hat{y}$ at a new position $\boldsymbol{x}$. This prediction is first added to the training data:

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y}^T & \hat{y} \end{bmatrix}^T, \tag{3-22}$$

$$\hat{\boldsymbol{\psi}} = \begin{bmatrix} \mathrm{cor}(f(\boldsymbol{x}^{(1)}), \hat{f}(\boldsymbol{x})) \\ \dots \\ \mathrm{cor}(f(\boldsymbol{x}^{(n)}), \hat{f}(\boldsymbol{x})) \end{bmatrix} = \begin{bmatrix} \psi_1 \\ \dots \\ \psi_n \end{bmatrix}, \tag{3-23}$$

$$\hat{\mathbf{\Psi}} = \begin{bmatrix} \mathbf{\Psi} & \hat{\boldsymbol{\psi}} \\ \hat{\boldsymbol{\psi}}^T & 1 \end{bmatrix}. \tag{3-24}$$

Now the ln-likelihood of the augmented data is:

$$\ln L = -\frac{n}{2}\ln(2\pi\hat{\sigma}^2) - \frac{n}{2}\ln(\hat{\sigma}^2) - \frac{1}{2}\ln|\hat{\mathbf{\Psi}}| - \frac{(\hat{\boldsymbol{y}} - \mathbf{1}\hat{\mu})^T \hat{\mathbf{\Psi}}^{-1}(\hat{\boldsymbol{y}} - \mathbf{1}\hat{\mu})}{2\hat{\sigma}^2}. \tag{3-25}$$

For the maximization of the ln-likelihood with $\hat{y}$ as variable, just the last term is contributing. So it is enough to use just this term for the determination of $\hat{y}$. Substituting $\hat{\boldsymbol{y}}$ and the explicit expression of $\hat{\mathbf{\Psi}}^{-1}$, while removing terms without $\hat{y}$ gives:

$$\ln(L) \approx -\left(\frac{1}{2\hat{\sigma}^2(1 - \boldsymbol{\psi}^T\mathbf{\Psi}^{-1}\boldsymbol{\psi})}\right)(\hat{y} - \hat{\mu})^2 + \left(\frac{\boldsymbol{\psi}^T\mathbf{\Psi}^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu})}{\hat{\sigma}^2(1 - \boldsymbol{\psi}^T\mathbf{\Psi}^{-1}\boldsymbol{\psi})}\right)(\hat{y} - \hat{\mu}). \tag{3-26}$$

For maximizing this function, it can be differentiated to the new estimation $\hat{y}$. Setting this equal to zero, an explicit expression for $\hat{y}(\boldsymbol{x})$ can be derived:

$$\hat{y}(\boldsymbol{x}) = \hat{\mu} + \boldsymbol{\psi}(\boldsymbol{x})^T \boldsymbol{\Psi}(\boldsymbol{X})^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu}). \tag{3-27}$$

In Equation 3-27, for a new estimation, just $\boldsymbol{\psi}(\boldsymbol{x})$ has to be calculated. $\hat{\mu}$ and $\boldsymbol{\Psi}(\boldsymbol{X})^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu})$ are calculated during the construction of the estimator together with the parameters $\boldsymbol{\theta}$ and $\boldsymbol{p}$ in the basis function.

Estimate of the standard error

When looking to Equation 3-26, a value for the new estimated point $\hat{y}$ is more likely when the second derivative of the likelihood has a high extreme value around $\hat{y}$. The absolute value of the second derivative of the ln-likelihood is:

$$\frac{\partial \ln(L)}{\partial \hat{y}} = \frac{1}{\hat{\sigma}(1 - \boldsymbol{\psi}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\psi})}. \tag{3-28}$$

When this has to have a large extreme value around $\hat{y}$ to be a good estimation, the closer the denumerator goes to zero, the better the prediction is. So the size of the error in $\hat{y}$ can be related to:

$$\text{indication}_{\text{error}} = \hat{\sigma}(1 - \boldsymbol{\psi}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\psi}). \tag{3-29}$$

The expression for the mean squared error for a predictor (s^2 or σ^2) is:

$$s^2 = \hat{\sigma}\left(1 - \boldsymbol{\psi}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\psi} + \frac{(1 - \boldsymbol{\psi}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\psi})^2}{\mathbf{1}^T \boldsymbol{\Psi}^{-1} \mathbf{1}}\right). \tag{3-30}$$

Note that Equation 3-29 and Equation 3-30 are very close to each other. The added part in Equation 3-30 can be interpreted as the uncertainty in μ because it has to be estimated from data[14].

Co-kriging and Noisy data

As is explained in the previous section, close data points can cause problems during building the estimator, even it can cause very local behaviour while this is not expected. This can be prevented by the use of the regularization parameter introduced by Poggio [31]. It is implemented in the same way as for RBF (Section 3-2-2). It is again added to the main diagonal of the correlation-matrix. This results in:

$$\boldsymbol{w} = (\boldsymbol{\Psi} + \lambda \boldsymbol{I})^{-1} \boldsymbol{y}, \tag{3-31}$$

where λ is again representing the variance (σ^2) in the response data. This is estimated during building the estimator ($\hat{\sigma}$), so this value can be used. Another option is to add it to the list of parameters that need to be estimated independent of building the estimator. With $\lambda \neq 0$, by definition $\boldsymbol{\psi}$ is not a column or row from $\boldsymbol{\Psi}$ any more This causes the estimator to not be interpolating.

Kriging basisfunctions

For Kriging, in some literature the basis function especially founded for Kriging is used [8][14]. This basis-function looks like a Gaussian distribution. Some literature gives also room to other basis-functions. Kleijnen[18] uses linear, exponential and Gaussian basis-functions. Simpson [39] is mentioning the possibility, but uses Kriging basis-functions. Koehler[20] comes up with the cubic(extended version), exponential, Gaussian and Matérn basis-function. The standard work from Sacks [34] restricts itself not to a specific basis-function but gives 'special interests' to those with the standard Kriging form.

Kriging and Normalization
Some literature suggests that it is not necessary to normalize the input parameters of Kriging relative to each other[14]. During the construction of the Kriging estimator the likelihood is maximized for the parameter θ in every dimension. In this way, the dimensions are already scaled relative to each other.
Other literature advises to scale the data always to ranges between zero and one, in order to have comparable values for $\boldsymbol{\theta}$ from problem to problem. Generally spoken θ_l will always be in between $\theta_{l,\min} \approx 10^{-3}$ and $\theta_{l,\max} \approx 10^2$[9]. This results in a more simple maximization problem, because bounds for θ_l can be defined.

3-2-4 Support Vector Regression (SVR)

Support Vector Regression(SVR) is a particular implementation of Support Vector Machines(SVM). SVM is about classification of data, while SVR is about function prediction.

The basic set-up of a linear estimation done with SVR is:

$$\hat{f} = \mu + \boldsymbol{w}^T \boldsymbol{x}. \tag{3-32}$$

Now it is assumed that the complexity of the estimator depends on the norm of $|\boldsymbol{w}|^2$. Besides this, to have some relation with the inputs y, a maximum error of ε is admissible at the training points which the estimator is based on. Then the solution of $\boldsymbol{w}$ becomes an optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w}} \frac{1}{2}|\boldsymbol{w}|^2 \quad \text{s.t.} & \\ & y_i - \boldsymbol{w}^T \boldsymbol{x}^{(i)} - \mu \leq \varepsilon \\ & \boldsymbol{w}^T \boldsymbol{x}^{(i)} + \mu - y_i \leq \varepsilon \end{aligned} \tag{3-33}$$

with $i = 1...n$ (n is the number of training points). Using this, the assumption is done that there exist a $\boldsymbol{w}$ that creates a function that is able to fit all y_i within ε. If this is not the case, the optimization will not be able to find a feasible solution. To give the optimization a possibility to handle these outliers, slack variables are introduced: ξ^+ and ξ^- for respectively underestimating $f(\boldsymbol{x}) - \varepsilon$ or overestimating $f(\boldsymbol{x}) + \varepsilon$. Now the optimization problem becomes:

$$\begin{aligned} \min_{\boldsymbol{w}} \frac{1}{2}|\boldsymbol{w}|^2 + C\frac{1}{n}\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-) \quad \text{s.t.} & \\ & y_i - \boldsymbol{w}^T \boldsymbol{x}^{(i)} - \mu \leq \varepsilon + \xi_i^+ \\ & \boldsymbol{w}^T \boldsymbol{x}^{(i)} + \mu - y_i \leq \varepsilon + \xi_i^- \\ & \xi_i^+ \geq 0 \\ & \xi_i^- \geq 0 \end{aligned} \tag{3-34}$$

The constant C controls the trade-off between a small $|\boldsymbol{w}|^2$ (so a simple estimator) and the violation of the bounds ε. The method of tolerating an error is called the ε-insensitive loss function and gives the possibility to use a sparse or reduced basis set of vectors to use in regression. This problem can also be written as an unconstrained minimization problem by means of Lagrange multipliers (η_i^+, η_i^-, α_i^+, α_i^-), with the Lagrangian:

$$\begin{aligned} L = \frac{1}{2}|\boldsymbol{w}|^2 + C\frac{1}{n}\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-) - \sum_{i=1}^{n}(\eta_i^+\xi_i^+) - \sum_{i=1}^{n}(\eta_i^-\xi_i^-) & \\ - \sum_{i=1}^{n}\alpha_i^+(\varepsilon + \xi_i^+ - y_i + \boldsymbol{w}^T\boldsymbol{x}^{(i)} + \mu) - \sum_{i=1}^{n}\alpha_i^-(\varepsilon + \xi_i^- + y_i - \boldsymbol{w}^T\boldsymbol{x}^{(i)} - \mu). & \end{aligned} \tag{3-35}$$

This should be minimized for $\boldsymbol{w}$, μ, ξ_i^+ and ξ_i^- (primal variables) and maximized for η_i^+, η_i^-, α_i^+ and α_i^- (dual variables). A combination of minimization and maximization means that the optimum is a saddle point at which the derivatives with respect to the primal variables are equal to zero. Rewriting the partial derivative with respect to $\boldsymbol{w}$ gives:

$$\boldsymbol{w} = \sum_{i=1}^{n}(\alpha_i^+ - \alpha_i^-)\boldsymbol{x}^{(i)}. \tag{3-36}$$

Using this in Equation 3-32 and still assuming a linear regression relation, this gives:

$$\hat{f} = \mu + \boldsymbol{w}^T\boldsymbol{x} = \mu + \sum_{i=1}^{n}(\alpha_i^+ - \alpha_i^-){\boldsymbol{x}^{(i)}}^T\boldsymbol{x}. \tag{3-37}$$

To extend this, a projection of $\boldsymbol{x}$ to another space $\boldsymbol{\phi}$ can be assumed. Assume that ${\boldsymbol{x}^{(i)}}^T\boldsymbol{x}^{(i)} = \boldsymbol{\phi}_i^T\boldsymbol{\phi}_i = \psi_i$. Then Equation 3-32 becomes:

$$\hat{f} = \mu + \sum_{i=1}^{n}(\alpha_i^+ - \alpha_i^-)\psi_i. \tag{3-38}$$

For $\boldsymbol{\psi}_i(\boldsymbol{x}, \boldsymbol{x}^{(i)})$, also other functions can be used, when they meet certain requirements[9].These basis-function ψ are called Mercer kernels. Linear, (in)homogeneous polynomials, Gaussian and Kriging basis-functions are possible. To find the values α_i^+ and α_i^-, the maximization of Equation 3-35 needs to be done. Substituting explicit solutions from the primal variables in Equation 3-35 causes L to be just dependent on α_i^+ and α_i^-.

$$L = \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)\boldsymbol{\Psi}(\boldsymbol{x}^{(i)}, \boldsymbol{x}_j) - \varepsilon\frac{1}{2}\sum_{i=1}^{n}(\alpha_i^+ - \alpha_i^-) + \sum_{i=1}^{n}y_i(\alpha_i^+ - \alpha_i^-), \tag{3-39}$$

To find α_i^+ and α_i^-, the following optimization problem should be solved:

$$\begin{aligned} \max_{\alpha_i^+, \alpha_i^-} L \quad & \text{s.t.} \\ & \sum_{i=1}^{n}(\alpha_i^+ - \alpha_i^-) = 0, \\ & \alpha_i^+ \in [0, \frac{C}{n}], \\ & \alpha_i^- \in [0, \frac{C}{n}]. \end{aligned} \tag{3-40}$$

C and ε are free to choose. ε indicates how far new estimations on training points may deviate from the calculated values at these training points without any consequences. This gives a region around the estimation, which is sometimes referred to as a 'ε-tube'. C can be seen as the penalty on new estimations which are placed out of the ε-tube. This penalty is used in the optimization as counterpart of making $\boldsymbol{w}$ as flat as possible. So to say, ε can be used to eliminate the influence of noise within a certain bandwidth, while C pushes the estimator to place all training-points within the ε-tube.

3-2-5 Multi-Adaptive Regression Spline (MRS)

MRS is based on coupling local, usually relatively simple, first or second order RSMs. The local functions in corresponding regions overlap. In this way more continuous derivatives are available when comparing this method with the Adaptive Regression Splines method (not dealt with in this thesis).

The construction of this estimation model consists basically two phases: A forward stepwise phase to create a certain number of local functions till a criterium is reached. Followed by a backwards stepwise phase to remove functions in regions which are of low importance. The interested reader is referred to the paper of Friedman [10] which is the basis for many other papers dealing with Multi-adaptive Regression Splines.

3-2-6 Non-uniform Rational B(asis)-splines (NRB)

In literature this method is known as NURBS. It is mostly used for geometric representations in imaging, for instance for medical or modelling purposes. Turner and Crawford [47] has proposed to use this method for metamodeling as well. The basic set-up for a multi dimensional output $\hat{\boldsymbol{f}}$ based on an input vector $\boldsymbol{x}$ in a normalized domain, done with NRB is:

$$\hat{\boldsymbol{f}}(\boldsymbol{x}) = \frac{\sum_{i=1}^{n_{\mathrm{CP}}} \boldsymbol{b}_i w_i N_{i,k}(\boldsymbol{x})}{\sum_{i=1}^{n_{\mathrm{CP}}} w_i N_{i,k}(\boldsymbol{x})}. \tag{3-41}$$

This is valid for $\boldsymbol{x}_{\min} \leq \boldsymbol{x} \leq \boldsymbol{x}_{\max}$. In this equation n_{CP} is the number of control points, $\boldsymbol{b}_i$ is the value at the position of the i^{th} control point, w_i its weight and $N_{i,k}$ is the B-spline basis-function[3] for the i^{th} control point for curve order k. They are explained shortly:

B-spline basis-function: $N_{i,k}$ In NRB, the basis-function acts as a mapping from the true value on a normalized spline. The basis-functions for a one dimensional input are defined by the following recursive functions:

$$N_{i,k}(x) = \left(\frac{x - u_i}{u_{i+k-1} - u_i}\right) N_{i,k-1}(x) + \left(\frac{u_{i+k} - x}{u_{i+k} - u_{i+1}}\right) N_{i+1,k-1}(x) \tag{3-42}$$

$$N_{i,1}(x) = \begin{cases} 1 & \text{if } u_i \leq x < x_{i+1} \\ 0 & \text{otherwise.} \end{cases} \tag{3-43}$$

In these the knot points u_i are indicating which control points does have influence on the final result. This depends on the order of the B-splines.

Control points: Control points is a term just used in NRB and should be compared with the training points where the estimator is based on, although the control points are not necessarily on the same positions as the training points used as input for the construction of the estimator. On the position $\boldsymbol{x}_i$ of control point i, $\boldsymbol{b}_i$ can be calculated with:

$$\boldsymbol{N}(\boldsymbol{t}_j)\boldsymbol{b}_i = \boldsymbol{f}(\boldsymbol{t}_j), \tag{3-44}$$

with $\boldsymbol{t}_j$ representing the coordinates of the training points in the neighbourhood of $\boldsymbol{x}_i$ and $\boldsymbol{f}(\boldsymbol{t}_j)$ the calculated values on the position of these training points. The basisfunction matrix $\boldsymbol{N}(\boldsymbol{x}_i, \boldsymbol{t}_j)$ can be calculated with Equation 3-42 and Equation 3-43.

Weights: w_i The weight of a certain control point is influenced by the number of training points in its neighbourhood. So the closer the data is to a certain point, the higher the reliability of that control point is. The weights are calculated with:

$$w_i = w_{\min} + (w_{\max} - w_{\min})\boldsymbol{r}_i^T \boldsymbol{R}^{-1} \boldsymbol{r}_i, \tag{3-45}$$

$$\boldsymbol{r}_{i,j} = \psi(\boldsymbol{t}_j, \boldsymbol{x}_i) = e^{-\theta|\boldsymbol{t}_j - \boldsymbol{x}_i|^p}, \tag{3-46}$$

$$\boldsymbol{R}_{j,k} = \psi(\boldsymbol{t}_j, \boldsymbol{t}_k) = e^{-\theta|\boldsymbol{t}_j - \boldsymbol{t}_k|^p}. \tag{3-47}$$

[3] Note that this basis-function cannot be chosen!

Turner [48] found

$$\theta = \ln(w_{\min}), \tag{3-48}$$

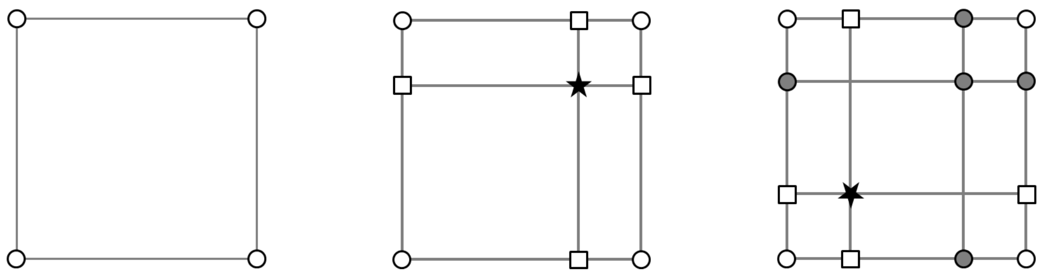
$$p = \frac{\ln(\ln(C))}{ln(\frac{1}{n_C})}, \tag{3-49}$$

with $w_{\min} = 0.1$, $w_{\max} = 1.0$ and $C = 2$ to give satisfying results in his examples.

Fitting of a NRBs estimator

Setting up a new NRB estimator is sometimes called 'fitting'. When fitting a NRB estimator, it is important to give attention to two important things:

1. Curse of dimensionality While fitting a NRB estimator, it is necessary to maintain at least a rectangular mesh with hyper rectangles of control points. This is shown in Figure 3-2 for 2 dimensions. Although, it is easier to use hyper squares [47]. This results in many control points when building the estimator ("curse of dimensionality").



(a) Start with control points on bounds domain.

(b) Control point added plus some other to create a rectangular mesh.

(c) Extra control point added to maintain a square mesh.

Figure 3-2: Iterative control point addition scheme for a 2D-input problem[47]. ○ indicates the control points with which the construction starts. ★ is indicating the added control point. □ represents the added positions to maintain a rectangular mesh.

2. Data sets In Appendix C the algorithm for building a NRB estimator is shown. It shows that during an iteration within the algorithm, a stopping criterium (correlation coefficient, RMS error) has to be checked. To do this, a data set with calculated or tested values has to be available. Based on the errors the estimator gives (compared to this dataset), new control points are added as described above.

3-2-7 Neural networks

Neural networks can be used as another approximation technique. It is able to approximate complex models very well, but they have some disadvantages[4]:

- Needs many positions to use as training data set
- Uses a black box approach
- Needs a computationally expensive training process

Based on these points, it is decided to leave Neural Networks out in the comparison of approximation techniques.

3-2-8 Methods compared

Now the different estimation methods that are explained above are compared to each other. A further analysis is done on the differences between RBF and Kriging. Afterwards the methods are first compared by some important properties and then based on the computational methods.

Overview methods
In Table 3-3 an overview is given for several properties of the methods that are explained in this section.

Table 3-3: Overview of the properties of the estimation methods.

Method	Valid region	Interpolating data	Noise handling	Non-linearities	Time to build	Error prediction	Multi-variate output
RSM	Global	$-$	$+$	$-$	$++$	$\varepsilon(\boldsymbol{X})$	$+-$
RBF	Global	$+$	$+-$	$+$	$+$	$\varepsilon_{RMS}(\boldsymbol{x})$	$-$
KRI	Global	$+$	$+-$	$+$	$-$	$\varepsilon_{RMS}(\boldsymbol{x})$	$-$
SVR	Global	$+-$	$+$	$+-$	$+-$	$\varepsilon(\boldsymbol{X})$	$+-$
MRS	Loc $\rightarrow$ Glob	$-$	$+-$	$+-$	$-$	$\varepsilon(\boldsymbol{X})$	$+-$
NRB	Loc $\rightarrow$ Glob	$-$	$+-$	$+-$	$-$	$\varepsilon(\boldsymbol{X})$	$+$

Basically, every estimation for a certain scalar value, based on a vector input is a combination of the product of two vectors with a low-order polynomial part or a constant b:

$$\hat{y} = \hat{f}(\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{y}) = \boldsymbol{\psi}(\boldsymbol{x}, \boldsymbol{X})^T \boldsymbol{w}(\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{y}) + b(\boldsymbol{x}, \boldsymbol{X}, \boldsymbol{y}). \tag{3-50}$$

In this, $\boldsymbol{x}$ is the new input vector, $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n]$ are the input vectors from the known training data points and $\boldsymbol{y} = [y_1, y_2, ..., y_n]^T$ are the outputs at the training data points. Every estimation method, except NRB, has an own set-up of the constant weight vector $\boldsymbol{w}$ and the non-constant input-vector $\boldsymbol{\psi}(\boldsymbol{x})$. When neglecting b, Table 3-4 shows an overview of the computational parts of all methods except MRS (because the formulation is split up in a lot of small estimators, they are left away).

Table 3-4: Overview of the computational parts of the estimation methods.

Method	$\boldsymbol{w}$	$\boldsymbol{\psi}(\boldsymbol{x}, \boldsymbol{x}_1, ...\boldsymbol{x}_n)$	$\boldsymbol{\Phi}(\boldsymbol{x}_1, \boldsymbol{x}_2, ...\boldsymbol{x}_n)$ or α_i	$\psi(\boldsymbol{c})$
RSM	$\boldsymbol{\Phi}^{+}\boldsymbol{y}$	$\boldsymbol{\psi}(\boldsymbol{x})$	$\left[\begin{array}{cccc} \psi(\boldsymbol{x}_1) & \psi(\boldsymbol{x}_2) & ... & \psi(\boldsymbol{x}_n) \end{array}\right]^T$	$\left[\begin{array}{cccccc} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \boldsymbol{c}_1^2 & \boldsymbol{c}_2^2 & \boldsymbol{c}_1\boldsymbol{c}_2 & ... \end{array}\right]$
RBF	$\boldsymbol{\Phi}^{-1}\boldsymbol{y}$	$\psi_i(r(\boldsymbol{x}, \boldsymbol{x}^{(i)}), d)$	$\psi_{i,j}(r(\boldsymbol{x}_j, \boldsymbol{x}^{(i)}), d)$	Basis functions(1D)[4](Table 3-1)
KRI	$\boldsymbol{\Phi}^{-1}\boldsymbol{y}$	$\psi_i(\boldsymbol{r}(\boldsymbol{x}, \boldsymbol{x}^{(i)}), \boldsymbol{d})$	$\psi_{i,j}(\boldsymbol{r}(\boldsymbol{x}_j, \boldsymbol{x}^{(i)}), \boldsymbol{d})$	Basis functions(nD)[5](Table 3-1)
SVR	$\alpha_i^{+} - \alpha_i^{-}$	$\psi_i(\boldsymbol{x}, \boldsymbol{x}^{(i)})$	Equation 3-39	Mercer kernels
NRB	$\boldsymbol{N}^{-1}\boldsymbol{p}$	$\frac{w_i N_{i,k}(\boldsymbol{x})}{\sum_{j=1}^{n_C} w_j N_{j,k}(\boldsymbol{x})}$		$N_{i,k}$:Equation 3-42 and Equation 3-43

From Table 3-4 it can become clear that RBF and Kriging have several things in common. The next paragraph is explaining the differences in more detail.

RBF and Kriging RBF (Section 3-2-2) and Kriging (Section 3-2-3) have both a different derivation. But in the end they have quite some things in common. In Table 3-5 an overview is given from the most important properties of both methods.

[4]For every dimension the same free parameter is used, so non-parametric basis-functions are used
[5]For every dimension a specific parameter is used, so parametric basis-functions are used

Table 3-5: Comparing Radial Basis Function and Kriging.

Method	Radial Basis Function	Kriging
Function	$\hat{y}(\boldsymbol{x}) = \hat{\mu} + \boldsymbol{\psi}(\boldsymbol{x})^T \boldsymbol{\Psi}(\boldsymbol{X})^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu})$	$\hat{y}(\boldsymbol{x}) = \hat{\mu} + \boldsymbol{\psi}(\boldsymbol{x})^T \boldsymbol{\Psi}(\boldsymbol{X})^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu})$
$\hat{\mu}$	Low order polynomial	Low order polynomial
$\boldsymbol{\psi}(\boldsymbol{x})$	$\psi_i(\vert\vert\boldsymbol{x}^{(i)} - \boldsymbol{x}\vert\vert)$	$\psi_i(\vert\boldsymbol{x}^{(i)} - \boldsymbol{x}\vert)$
Kriging basis-function	$e^{-\theta\vert\vert\boldsymbol{x}^{(i)} - \boldsymbol{x}\vert\vert^p}$	$e^{-\sum_{l=1}^{n} \theta_l \vert x_{i,l} - x_{i,l}\vert^{p_l}}$
$\Psi_{i,j}(\boldsymbol{x}^{(i)}, \boldsymbol{x}_j)$	$\psi(\vert\vert\boldsymbol{x}^{(i)} - \boldsymbol{x}_j\vert\vert)$	$\psi(\vert\boldsymbol{x}^{(i)} - \boldsymbol{x}_j\vert)$

Table 3-6: Parameters estimation methods

Method	Parameters			
	A priori		In estimator	
	Discrete	Continue	Discrete	Continue
RSM	Order			
RBF	Order, basis-function			d
KRI	Order, basis-function	p [7]		$\boldsymbol{d}$
SVR	Kernel	C, η,		$\alpha_i^+ - \alpha_i^-$
MRS	Self interactions, $n_{\text{functions}}$	$P_{\text{new var}}$, ϵ	Order	
NRB	$\text{Order}_{\text{max}}$		Order	$\boldsymbol{x}_{\text{KN}}$, $\boldsymbol{x}_{\text{CP}}$

When looking to the calculation of $\boldsymbol{\psi}(\boldsymbol{x})$ and the example on the Kriging-basis-function, it can be concluded that Radial Basis Functions uses the Euclidean distance (l^2-norm) in a metric space as input from the basis-function without any relative scaling of dimensions, while Kriging uses the Rectilinear-distance[6](l^1-norm) while scaling the variables relative to each other with θ_l and p_l. For this reason, it should not be necessary to compare RBF and Kriging with each other as method, because Kriging should always be able to reproduce the results of RBF. The question should be how the dimensions can be scaled relatively to each other, which dimensions influence the results of the estimator most and if it is worth to do an optimization within Kriging compared to directly calculating the estimator with RBF. One thing is important to note for Kriging: The optimization problem to find $\boldsymbol{\theta}$ can have local optima, which can be difficult in the optimization when looking for a global optimum.

Besides this, a result from the comparison is that the expression for the approximation of the ln-likelihood (Equation 3-26) used in Kriging can be applied on RBF also. From this follows that Equation 3-28 till Equation 3-30 can be used for RBF as well, which results in the same error prediction for RBF as used in Kriging.

Parameters to choose, to calculate or to optimize to

In Table 3-6 an overview of the parameters that are needed to define an estimator is given. The parameters can be split up in two groups: Parameters that had to be set before the construction of an estimation starts and parameters that are tuned during the construction of the estimator. For both groups, discrete and continuous parameters are possible. The continuous parameters that have to be set before the construction of the estimator starts, are most difficult to decide on. Using this in a continuous optimization, requires repeatedly construction of the estimator, which increases the overall construction time heavily.

[6]Sometimes also called Manhattan-distance or taxicab-distance

3-3 Multivariate output

The estimators introduced in Section 3-2, are based on the prediction of a scalar output. Only NRB[8] offers the potential to fit an entire vector. For engineering applications, the property that has to be estimated is in several cases multivariate. This can be solved for univariate estimators, by making an estimator for each variable in the multivariate output separately, as shown in Figure 3-3. This is a

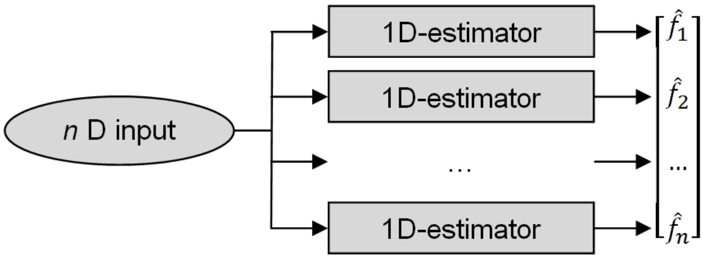


Figure 3-3: Multivariate output estimation bases on univariate estimators.

good approach, when the entries in the vector are independent of each other. But for related entries, the estimator will be more complicated than needed. For example when the result of the estimation is a load distribution over height, the load on node $n+1$ and on node $n-1$ are in general in the same order of magnitude than the load on node n.

As a solution for this it is proposed to estimate linear independent vectors which can be superimposed to each other. This can be done by a decomposition method as Proper Orthogonal Decomposition for example.

Proper Orthogonal Decomposition

Liang et al.[23] give a clear overview from several derivations of Proper Orthogonal Decomposition (POD). Because it is not in the scope of this thesis to study this extensively, a short, practical summary is given derived from this overview:

Assumed are m vectors with n entries, representing a response of a system:

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}_1 & \boldsymbol{z}_2 & \dots & \boldsymbol{z}_m \end{bmatrix}. \tag{3-51}$$

When subtracting the mean $\boldsymbol{\mu}$ from the response vectors($\boldsymbol{x}^{(i)} = \boldsymbol{z}_i - \boldsymbol{\mu}$), the covariance matrix can be defined as:

$$\boldsymbol{C} = \frac{1}{m}\sum_{i=1}^{m} \mathrm{E}\left[(\boldsymbol{z}_i - \boldsymbol{\mu})(\boldsymbol{z}_i - \boldsymbol{\mu})^T\right] = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}^{(i)T}\boldsymbol{x}^{(i)} = \boldsymbol{X}\boldsymbol{X}^T. \tag{3-52}$$

By solving the eigenvalue problem

$$\boldsymbol{C}\boldsymbol{\phi}_j = \lambda_j \boldsymbol{\phi}_j, \tag{3-53}$$

for $j = 1 : n$, one can find the Proper Orthogonal Modes (POMs) $\boldsymbol{\phi}_j$ and the Proper Orthogonal Values (POVs) λ_j of the system. If $\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & \dots & \boldsymbol{\phi}_m \end{bmatrix}$ every response can be projected on the space or mapped from the space spanned by these orthogonal modes with:

$$\hat{\boldsymbol{x}} = \boldsymbol{\Phi}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}), \tag{3-54}$$

$$\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{\Phi}\hat{\boldsymbol{x}}. \tag{3-55}$$

[7]For continuous functions a straightforward choice is $p = 2$ because this gives continuous derivatives

[8]According to the experts opinion provided by Prof. T. Simpson, Professor of Mechanical & Industrial Engineering, Engineering Design and Architecture at Pennsylvania State University.

Chapter 4

Load estimation

Chapter 2 has given an introduction into the optimization problem. Then all iterations are done with full load calculations, the optimization problem turns out to be rather computationally extensive. For this reason, in this chapter a method is proposed using fast load estimations instead of extensive load calculations. The background of the compared estimation methods is given in the previous chapter. First, it is explained which quantities are to be estimated and why this is done. After this, a review and some tests on possible inputs of the load estimation is given. Based on a comparison of several estimators, choices for the methods are made. Then a comparison of methods for multivariate output estimations is done. The chapter concludes with a proposal of interaction between the optimizer and the estimator during the optimization.

4-1 Quantities to be estimated

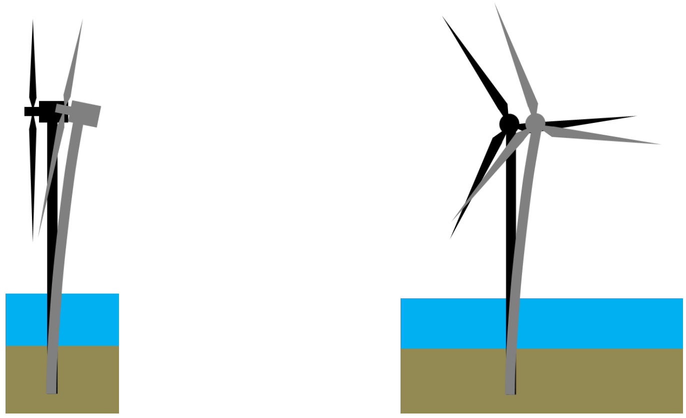
When building an approximation, the first question is: What has to be approximated and in which way can this be done? This section gives some options and it explains shortly why it is chosen to approximate the loads with an estimation and not by means of a derivative.

Estimating damage and buckling ratio
As explained in Section 2-4-3, damage and buckling ratio are both calculated in an analytical way by combining a numerical calculated load with geometrical properties. These geometrical properties are represented in this optimization by 28 design variables as shown in Figure 2-4. The combination of the load with local wall thickness's and local stress concentration factors causes much more local effects in the damage and buckling ratio than in the load itself. As described in Section 2-4-4, the algorithm is in the calculation phase for approximately 90% of the computational time. In this phase, no discontinuities other than the discretization in space and time (frequency) are present.

For these reasons, this chapter explains a method to estimate the results of the calculation phase: Fatigue loads, expressed as Damage Equivalent Loads, and Extreme loads.

Damage Equivalent Loads (DEL) DELs consist of the bending moment and the shear force, for both production life-time and idling life-time in the fore-aft direction (shown in Figure 4-1a) and the side-side direction (shown in Figure 4-1b). This gives 8 vectors in total.

As explained in Section 1-2, the evaluation of the DELs is done by a frequency domain analysis. To evaluate the accuracy of the frequency domain method, it can be compared to a time domain analysis



(a) Fore-aft direction. **(b)** Side-side direction.

Figure 4-1: Definitions of directions of a wind turbine as used in the wind industry.

done with the Siemens in-house developed tool BHawC[1]. For the wind farm used as design problem for this research, the error in the DELs was about $0 \sim 5\%$. So for the estimator to come up with a representative DEL for the initial design, the error in the estimation should be smaller than 5%.

Extreme Load The extreme load can be expressed as an extreme bending moment and an extreme shear force per evaluated sea state. For the initial design the time series of the extreme load are calculated with constant structural and aerodynamic properties. A direct comparison between the results as used for the initial design and the results calculated with BHawC for the final design, cannot be done. This is, because the extreme loads are dependent on several parameters, as for example the probability of the extreme wave and the extreme gust and the number of simulations used for the average extreme load, which are not defined in the initial design phase. For this reason, the estimation of the extreme load does not need a very high accuracy. The results of the estimations are considered as representative when they have an error smaller than 10% when compared to the calculated extreme loads.

The nodes for different support structures are not localized at the same position. To handle with this, the estimations are done on a standard grid. A mapping is used to switch between the grids, as shown in Figure 4-2. By using a sufficient number of nodes in this grid, it is assumed to be able to represent all possible load distributions. For the support structures with a free tower length or free PPD, the heights of the grid are ratios from the actual length.

Estimating loads with a derivative

Using a Taylor series expansion, it is possible to approximate a function when its value and its derivatives on a certain point are known. Then the derivatives are needed:

Damage Equivalent Loads A derivative can be found for the DEL. As derived in Appendix E the first derivative to 1 parameter takes 4 times more computational effort than a new calculation itself. This is caused by the chain rule for calculating the derivative of the transfer function (Equation E-29). Implementing this will lead to a reduced calculation time if the number of approximations that can be used around the calculated point is larger than 5. Using this, it is important to know for which space around the calculated point, the derivative is valid.

[1]A simulation tool, using a non-linear (aeroelastic) finite element formulation.

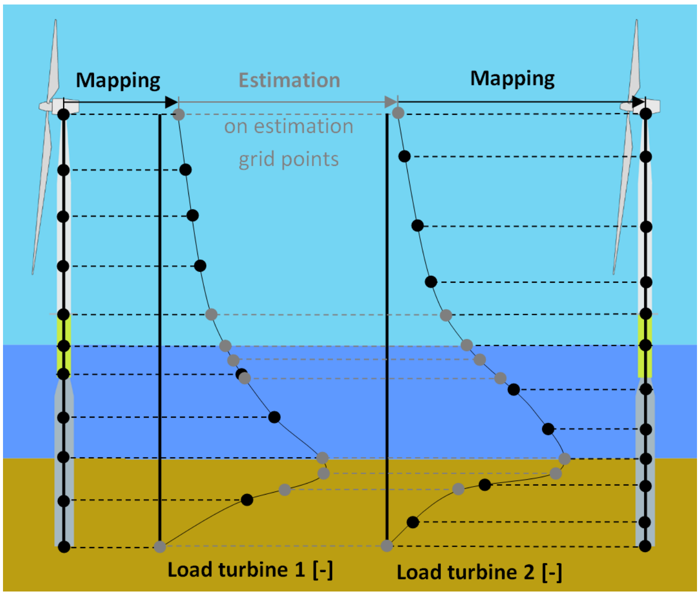


Figure 4-2: Example set-up for the mapping of a load. It is first mapped from the discretization of support structure 1 (black) to a grid used for the estimation. Then the estimation is done (grey), where after the load is mapped back to the discretization of support structure 2 (black).

Extreme loads The extreme load is a result of a time-domain simulation. Methods exist to come up with the derivative of a response over time to a variable. Using these will increase the computational time and will result again in the question to what extent the derivative is applicable.

From the above mentioned points, it is concluded that it is a better way to estimate the damage equivalent loads and the extreme loads directly, so without using any derivatives.

4-2 Defining inputs of the estimation

The inputs for the estimator are very important, because they should allow the estimator to distinguish and characterize global and local effects. First several options for the inputs are given, then some methods are given to indicate the quality of an estimator. Finally, this is used to come up with the inputs used in the estimations.

All comparisons are done on designs with all the design variables (as shown in Figure 2-4) free. In that way, the relative difference between the designs is the largest and so the variability within the estimations and the difficulty of the estimations.

4-2-1 Assessment of an estimation

The overall goal of the estimation method is to estimate the loads, represented by a vector, as accurate (shape-preserving) and precise (interpolating) as possible. The quality of one estimation, the proportional error of vector i, $q^{(i)}$, can be expressed by a weighted mean of the error over the height:

$$q^{(i)} = \frac{1}{n_p^{(i)} \sum_{k=1}^{n_p} w_k} \sum_{j=1}^{n_p} w_j \frac{\left(\hat{f}_j^{(i)} - f_j^{(i)}\right)}{f_j^{(i)}}. \tag{4-1}$$

In this equation, $n_p^{(i)}$ is the number of entries in that vector, $\hat{f}_j^{(i)}$ is the estimated and $f_j^{(i)}$ is the computed value for the j^{th} entry of the vector and w_j is the weight of the j^{th} value.

For the quality of a load estimation, the weight is taken proportional to the magnitude of the computed value, with a minimum of a half times the maximum weight. This is done to emphasize the accurate estimation of the maximum loads around mudline, without fully neglecting the loads at the top and the bottom of the structure.

During measurements on a physical test set-up, due to uncontrollable variables, repeatedly testing will show different errors with a random nature. When repeatedly using a computer model with the same input, due to the deterministic nature of the simulation, the outcome will give every time the same value. For this reason, most statistical error estimations on results of computer experiments give a bad representation for the quality of a simulation. Just the R^2-test is an exception on this[40].

The quality of a series of estimations can be represented by the correlation coefficient R^2 (Appendix B) per entry of the vector or by a performance profile. A simple example of a performance profile is shown in Figure 4-3. If the number of test points is sufficient and if the positions of both the test

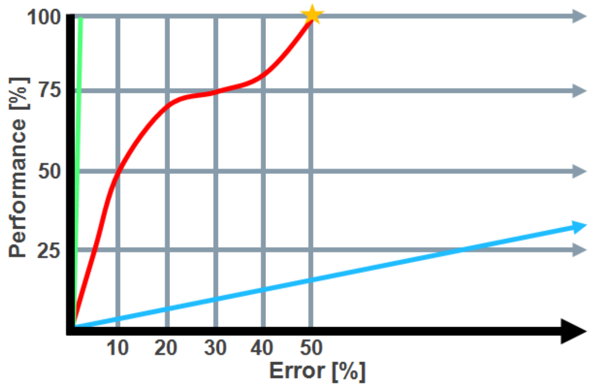


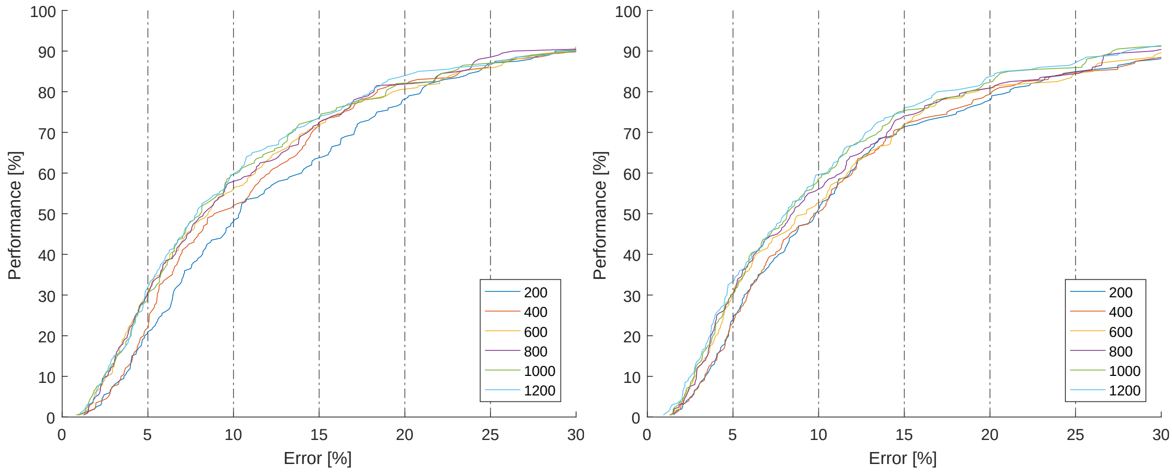
Figure 4-3: Simple example of a performance profile (red line). The green line represents a good performance, while the blue line shows a bad performance. An example: From the graph it can be seen that for the red line, 50 % of the estimations have an error smaller than 10%, that 75% have an error smaller than 30 % and that all estimations have a proportional error smaller than $q_{\text{min}} = 50\%$. The orange star indicates the minimum performance q_{min}.

points and training points are taken from a random set, then for a new estimation it can be assumed that the error is equal or smaller than the minimum performance q_{min}.

As mentioned in Section 4-2-4, the errors in the load calculations for the initial design are around 0-10%. During the optimization it is important that all constraint assessments give an indication of the feasibility from the design. But it is not needed that all results meet this criterion. Just the loads for the end result should be smaller or equal to the error in the calculated result, to come up with a realistic final design. This is checked for the end results of the different optimization cases.

4-2-2 Creating training, check and test data

For the selection of the inputs, many points are calculated. To come up with representative locations of these points over the full design space, Latin-Hyper-Cube sampling[25] is used for the design parameters. It is assumed that this will give a comparable, evenly distributed sample plan in the input domain. When the number of these points is taken large enough and when they are random, the performance profiles are assumed to give a proper representation from the performance of all possible designs. The number of check points, to find C, and test points are both set equal to 200, which is assumed to give a good representation of the performance. The number of training points can be set in two ways: Based on a minimum accuracy or based on the convergence of the accuracy. The performance of different numbers of training points for FLS (production bending moment) and ULS (bending moment) is shown in Figure 4-4. When looking to the improvement of the performance



(a) Performance for the estimation of the production bending moment(FLS).

(b) Performance for the estimation of the extreme bending moment(ULS).

Figure 4-4: Performance with different numbers of training points in the FDB_-case.

profile, it can be concluded that adding more points will improve the estimator, although the improvement itself converges. In other words: More points improve the performance, but not in proportion with the time needed to calculate these points. For this reason, it is assumed that using 800 points gives a sufficient performance of the estimator considering the required calculation time.

4-2-3 Possible inputs of the estimation

DEL For the wave induced DELs, the following inputs are important because they are the only design dependent parts used in the calculation (Appendix D Equation **??**):

- For the structural response part:
  - Eigenfrequencies structure: ω_i and ω_i^2.
  - Eigenmodes structure: ϕ_i.
  - Modal mass structure: μ_i.
- For the loads part:
  - Diameter below SWL: D (used for the linearized drag force) and D^2 (inertia force).

Extreme load For the Extreme loads resulting from an extreme wave, the following inputs are important because they are the only design dependent parts used in a time-domain simulation:

- For the structural response part:
  - Eigenfrequencies structure: ω_i^2.
  - Eigenmodes structure: ϕ_i.
  - Modal mass structure: μ_i.

- For the loads part:
  - Diameter below SWL: D (drag force) and D^2 (inertia force).

Modal information Four important notes concerning the modal information are made before discussing the selection of inputs:

1. For both the DEL and extreme load input, the combination of eigenfrequency and modal mass is also representing the modal stiffness μ_i, although it can be added separately as well.
2. For the extreme load, the constant aerodynamic damping does not cause interaction between different modes, when Rayleigh damping is assumed.
3. Nearly all energy in the frequency spectrum for waves is present below 0.25 Hz. A typical eigenfrequency of the first mode is below 0.25 Hz as well. The typical eigenfrequency of the second mode is a number of times higher and therefore the first eigenmode dominates in the contribution of the eigenmodes to the DEL. For the time series of the deflection, the modes with a low eigenfrequency are most dominant as well.
4. An eigenmode can be expressed in several ways, depending on the discretization in space. It is important to realize that for wave loads the part just at and below the Still Water Level (SWL)[2] is most important because there the amplitude of the wave force is maximum.

4-2-4 Comparison of several input combinations

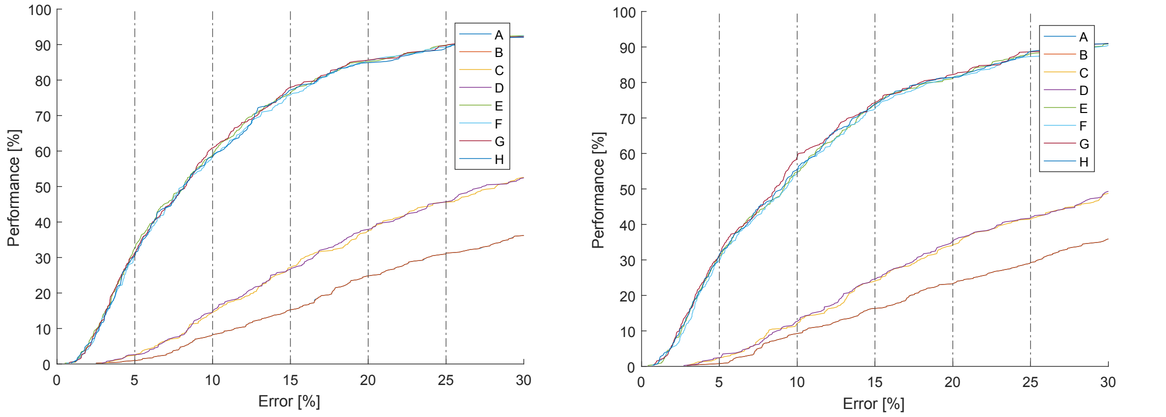
A high number of inputs can possibly produce better results, but makes the construction of an estimator more difficult. To select which inputs are used, it is chosen to compare them, based on one estimator which is able to take non-linear effects into account, but which does not need any continuous input parameters. So the Radial Basis Functions-method is used. It is assumed that the effect of different input parameters is the same for every estimation method. Now this is done for DELs and the Extreme Loads. For DELs just the bending moment in fore-aft direction for the production and idling case is shown because these are dominant in the total fatigue damage. The other forces show comparable behaviour. For the extreme load both bending moment and shear force are discussed. Initially the first eigenmode ϕ_1 is represented with the deflection (mass normalized) of the top, at SWL, 5, 10 and 15 meter below SWL, at mudline and at the pile tip In Section 4-2-5 the approach of using the eigenmode as input is discussed in more detail.

From Section 4-2, the following combinations of parameters are derived:

A: ω_1.
B: ω_1^2.
C: $[\omega_1\ D]$.
D: $[\omega_1\ D^2]$.
E: $[\omega_1\ D\ \phi_1]$.
F: $[\omega_1\ D\ \phi_1\ \mu_1]$.
G: $[\omega_1\ D\ \phi_1\ \gamma_1]$.
H: $[\omega_1\ D\ \phi_1\ \mu_1\ \gamma_1]$.

Damage Equivalent Loads In Figure 4-5 the performance over the height, for the estimation of DELs, is given for several input parameter combinations. It shows that using inputs related to structural properties in combination with inputs related to the loads, give the best performance. The negligible small difference between line A and B, shows that there is no significant difference in using ω_1 instead of ω_1^2. Likewise, the small difference between C and D, shows that there is no significant difference

[2] All calculations are done neglecting tidal effects.



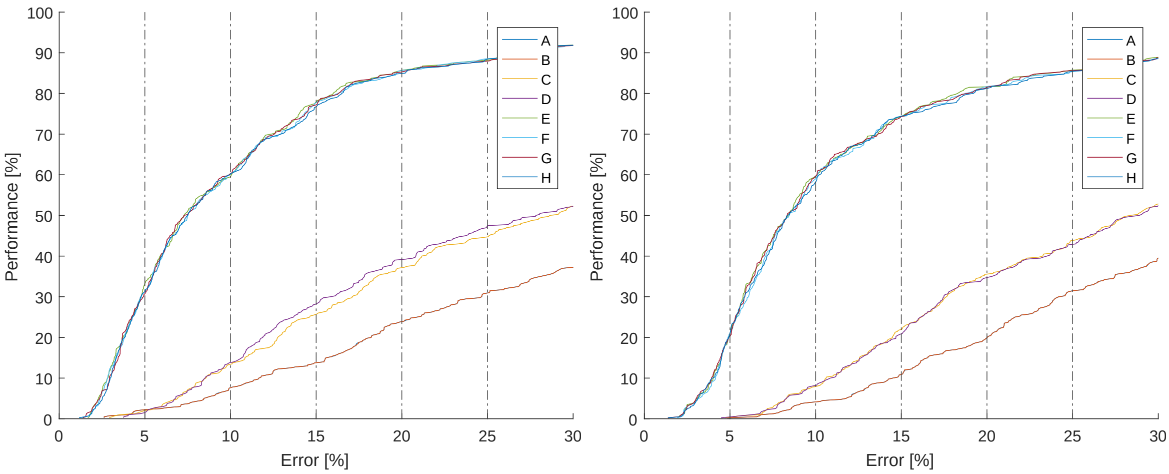
(a) Performance for the bending moment during production.

(b) Performance for the bending moment during idling.

Figure 4-5: Performance of several combinations of input parameters for the Damage Equivalent Loads in the FDB_-case.

in using D or D^2. The difference in the options for the structural inputs, when using at least the eigenfrequency and eigenmode, is small. The difference in calculation time for the construction of the estimator for E, F, G and H is comparable and is negligible compared to the calculation of the training points. When zooming in on both the correlation (not shown for brevity) and the performance, G is a better than E, F and H. For this reason it is chosen to use the input combination G for the estimation of DELs.

Extreme Loads In Figure 4-6, the results for several input combinations are shown for estimating the extreme loads. It shows, as for the estimation of DELs, that both inputs related to structural



(a) Performance for the bending moment.

(b) Performance for the shear force.

Figure 4-6: Performance of several combinations of input parameters for the extreme loads

properties as well as inputs related to the properties of the loads are needed to come up with reliable estimations. The same considerations as for the DEL can be taken, to choose for input parameter combination G.

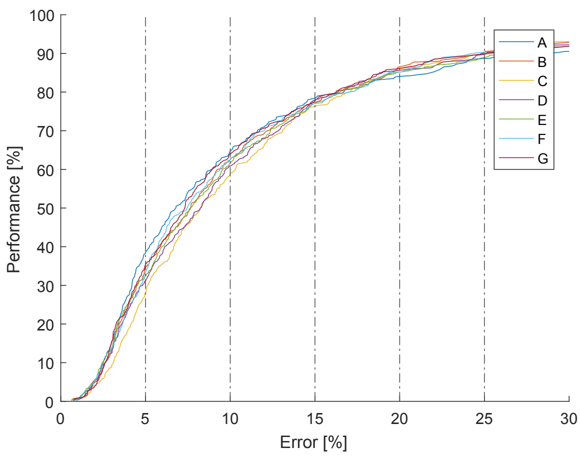
4-2-5 Comparison of several eigenmode input options

As mentioned in Section 4-2, the way of using the eigenmode ϕ_1 as input is not unique. For all loads, it is important to note that a high deflection at SWL and below SWL enables the waves to transfer more energy into the structure compared to a structure with a low deflection at the same heights. Generally the wave amplitude is the highest at SWL and decreasing to mudline, so a more dense grid just below SWL is preferred. It is chosen to use two grids: SWL + $\begin{bmatrix} 0 & 2 & 4 & 6 \end{bmatrix}$ meter (grid I) and SWL + $\begin{bmatrix} 0 & 5 & 10 & 15 \end{bmatrix}$ meter (grid II). Although there are many other possibilities, these are chosen in such a way that grid I puts more emphasis on the surface itself while grid II takes deeper deflections into account. Six input combinations are defined:

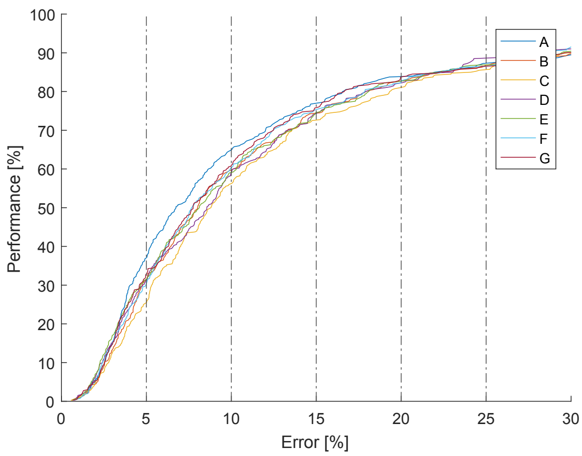
A: Deflections at 4 points distributed over the height.
B: Deflections at 64 points (almost equal to the number of elements used to model the support structure) distributed over the height.
C: Deflections at Grid II, mudline and pile tip
D: Deflections at towertop, Grid I, mudline and pile tip
E: Deflections at towertop, Grid II, mudline and pile tip
F: Deflections at towertop, halfway towertop and SWL, Grid I, mudline, halfway mudline and pile tip and the pile tip itself.
G: Deflections at towertop, halfway towertop and SWL, Grid II, mudline, halfway mudline and pile tip and the pile tip itself.

These input combinations are tested for both the DELs and the extreme loads separately. The eigenmodes are used in the chosen input combination G with $[\omega_1 \; D \; \phi_1 \; \gamma_1]$.

Damage Equivalent Loads In Figure 4-7 the results for the defined options of eigenmode inputs are shown for estimating the damage equivalent loads. It shows a small dependency on the way of



(a) Performance for the bending moment for the production case.



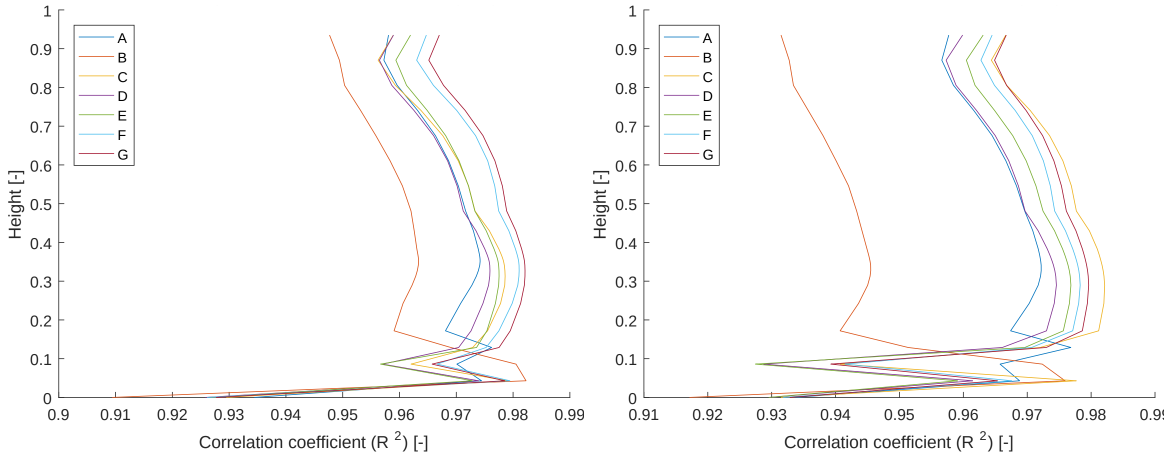
(b) Performance for the bending moment for the idling case.

Figure 4-7: Performance for the estimation of DELs, for eigenmode implementations A-G.

implementing the eigenmode. The following conclusions are drawn:

- Comparing combination D and E and combination F and G, using the deflections at grid I is better than using the deflections at grid II.
- Comparing combination C and E, using the deflection at towertop gives better results than without using the deflection at towertop.

- When looking to the performance of combination B, it can be concluded that just using more nodes does not necessarily give better results.



(a) Correlation coefficient for the bending moment for production.

(b) Correlation coefficient for the bending moment for idling.

Figure 4-8: Correlation coefficients for the estimation of DELs, for eigenmode implementations A-G.

From Figure 4-7, it can be concluded that input combination A gives the best results for the estimation. This is remarkable, because it is the input combination with the smallest reference to fixed places on the support structure. When looking to the correlation coefficients over the height in Figure 4-8, for the heights between 0.05 and 0.015 A gives almost the highest correlation, while for the other heights F and G are performing better. The absolute value of the DEL for the part just below mudline is the highest and gives therefore the highest weights in the performance as well. This causes the high performance for a despite the relatively low correlation coefficient. Based on the performance for case with the tower length, tower diameter and PPD fixed, it is chosen to use eigenmode implementation F in the optimization.

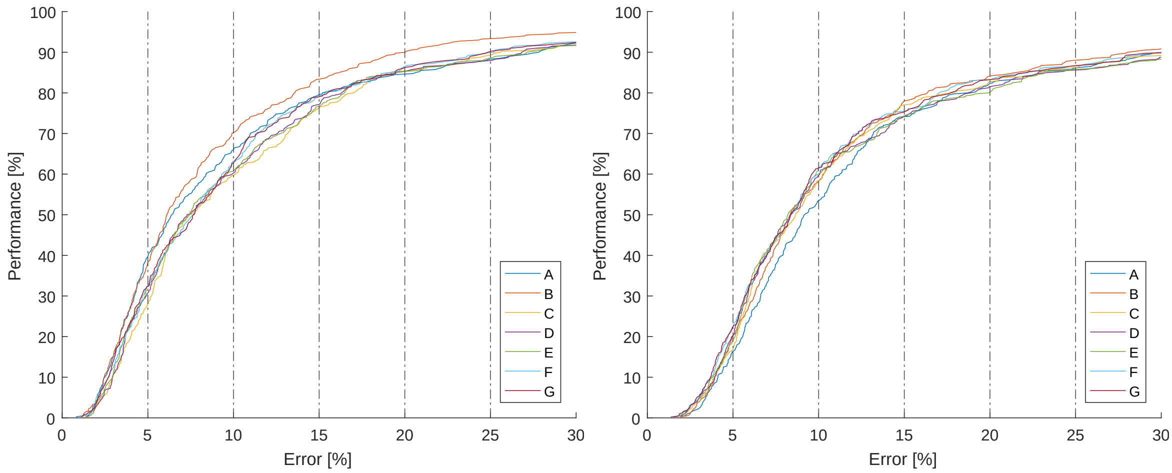
Extreme Loads In Figure 4-9 the results of several input combinations are shown for estimating the extreme loads. The following conclusions can be drawn:

- For the bending moment, the parameter combination B is clearly outperforming the other combinations. For the shear force, for errors smaller than 14%, B is not better than the other input combinations
- For the shear force, combination A gives clearly worse results than the other combinations.
- The other results are too close to each other to base conclusions on.

For an easy implementation and because there is no eigenmode implementation which performs best on all points (neither for the case with the fixed length and size of the tower and the fixed PPD), it is chosen to use the same eigenmode implementation as for the DELs.

4-3 Estimator choice

As explained in Section 4-1, 2 sorts of vectors need to be estimated. It is possible to use a different estimation method for every entry in all vectors. In this research it is chosen to use one estimation method for the DELs and one estimation method for the extreme loads, independent of each other.



(a) Performance for the bending moment. **(b)** Performance for the shear force.

Figure 4-9: Performance for eigenmode implementations A-G for estimating the extreme load.

The choice for the estimation methods is based on the most important direction of the load, according to Section 4-2-4.

4-3-1 First selection estimators

In Table 3-6 it is shown which parameters are needed to be chosen before the construction of the estimator can start. This table is most important when constructing an estimator for an unknown response surface. It is shown that Kriging, SVR, MRS and NRB have continuous parameters which are needed to be defined a priori. These parameters are dependent on the non-linearity of the surface. Investigations show that it is difficult to find values for these parameters and for this reason it is difficult to come up with good estimation models for Kriging, SVR and MRS. Therefore, SVR and MRS are not used in the comparison of estimation methods. Kriging has one continuous parameter p. According to literature[18], this can be set to $p = 2$ if the surface and its derivatives are continuous. This is expected to be valid for the load surfaces. For the construction of NRB, it is mentioned in Section 3-2-6 that a rectangular mesh is required. Because the relation of design variables and the inputs of the estimator is not analytical and not unique[3], there is no straightforward way to create this rectangular mesh. Moreover, for a rectangular grid in higher dimensions, many training points are required ('curse of dimensionality'). Lastly, adding new control points is based on a known response surface or a number of a priori calculated points. This causes a need for more training points or checkpoints, in comparison with the other estimators. For these reasons, NRB is left out the comparison as well.

First, a methodology is proposed to give the estimators a possibility to optimize themselves, after this, the estimators are compared to each other using this construction strategy.

4-3-2 Methodology for choosing free parameters in estimators

In Section 3-2-8 it is explained that all estimators need discrete input parameters $p_1 ... p_n$ defined a priori, to use during the construction of the estimator. The choice for the discrete parameters is done

[3]As explained in Section 4-2, the inputs into the estimator are not the same as the design variables used in the optimization. The steps in between are the set-up of a structural model (including the calculation of the eigensolution) which represents the designs.

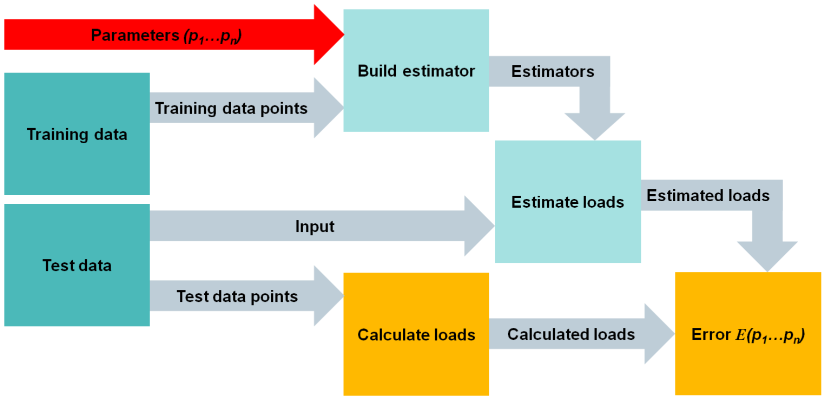


Figure 4-10: Overview for calculating the Error E as function from the input parameters $(p_1...p_n)$ from the estimator.

according to the scheme shown in Figure 4-10. The Error E is calculated with:

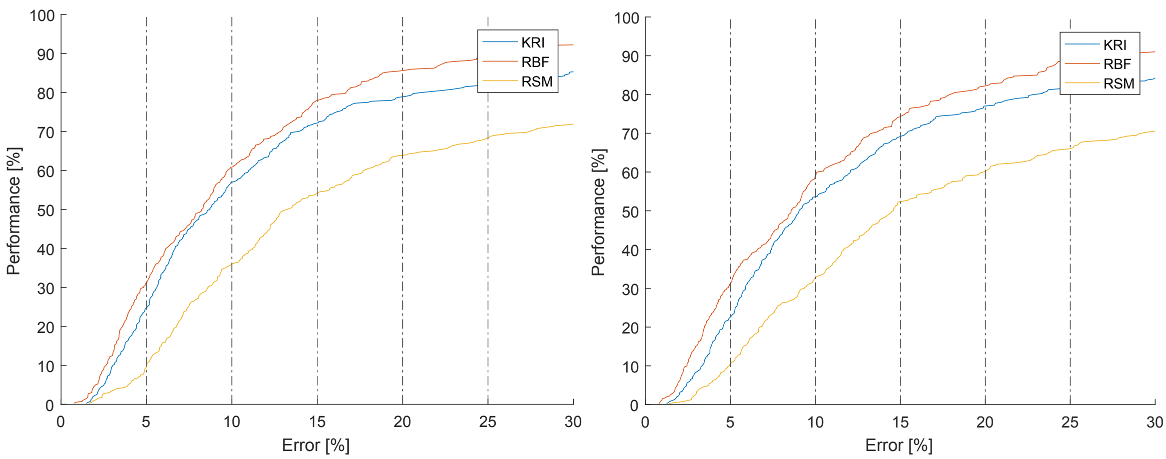
$$E(p_1...p_n) = \sum_{i_{td}=1}^{n_{td}} \varepsilon_{i_{td}}(p_1...p_n). \tag{4-2}$$

In this equation n_{td} denotes the number of test data points and $\varepsilon_{i_{td}}$ is the difference between the calculated value and the estimated value. After calculating the errors of all parameter combinations, the best combination of input parameters, with the smallest value for E is chosen. Note that RSM, RBF and Kriging can just create univariate estimators, so this optimization is used for every entry of the load vectors.

4-3-3 Comparison estimators

Estimator choice for FLS

In Figure 4-11 the results are shown for estimating the DELs. It is clear that RBF performs best for



(a) Performance for the bending moment during production.

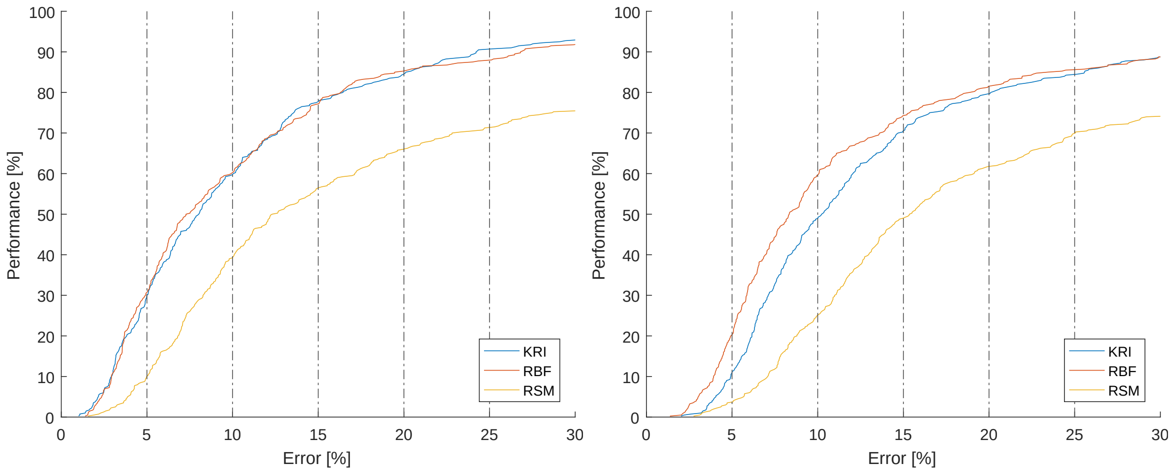
(b) Performance for the bending moment during idling.

Figure 4-11: Performance of the different estimators for estimating DELs.

both production and idling cases. As explained in Section 3-2-8, it can be concluded that Kriging is not able to find a global optimum. It is clear that the performance of the RSM is much worse than the performance from the non-linear estimators RBF and Kriging. This is an indication of the non-linearity of the surface.

Estimator choice for ULS

In Figure 4-12 the results of estimating the extreme loads are shown. For the bending moment, the



(a) Performance for the bending moment.

(b) Performance for the shear force.

Figure 4-12: Performance of the different estimators for estimating extreme loads.

results for RBF and Kriging are comparable. For the shear force, again RBF is outperforming the other two estimators.

4-3-4 Improving vector estimations by Proper Orthogonal Decomposition

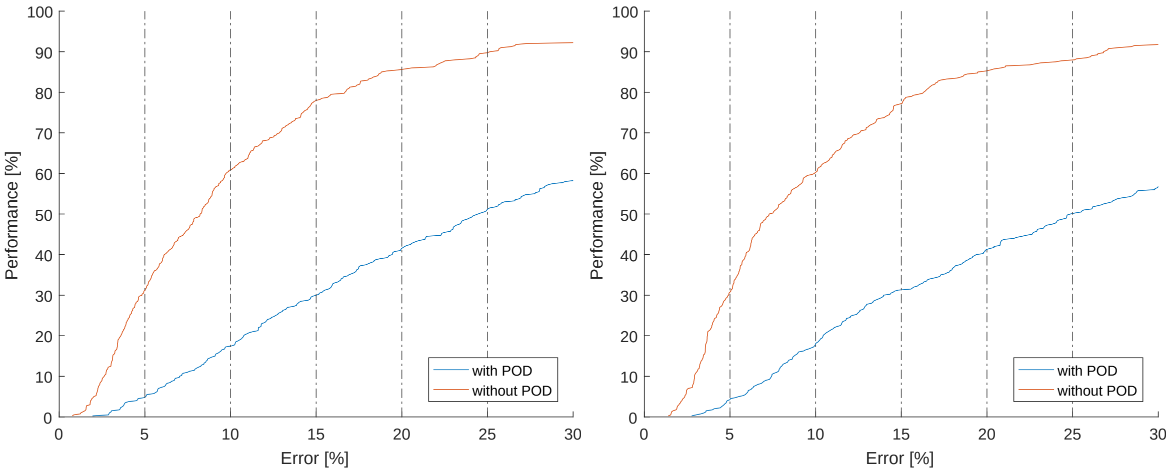
As discussed in Section 3-3, the estimation of multivariate output is possible in several ways. Until now, it is done by the construction of several 1D-estimators. The inputs are known and the estimation method is known, so the next step is to compare this with estimating the loads in a subspace created with POD. The results for this are shown in Figure 4-13. For brevity, just the results for the damage equivalent bending moment for production and the extreme bending moment are given. The results for the other damage equivalent loads and extreme loads are comparable.

From Figure 4-13a it is clear that the use of POD for estimating DELs does not improve the results. The same behaviour can be seen in the extreme loads.

From this it can also be derived that the input parameters have a more local effect on the loads, than global effects. Although a common shape in the loads can be easily recognized (for this reason there is a large distinction in the POVs belonging to the first POMs), it is difficult to estimate a magnitude of this shape based on the input parameters. More local effects are added up to this via subsequent modes with magnitudes that are as difficult to estimate.

4-4 Estimator updating

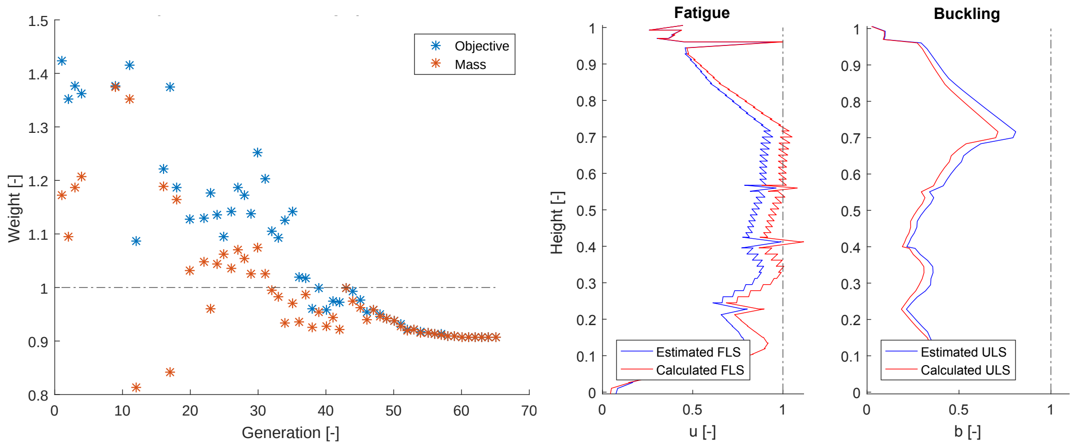
From the figures follows that the estimations for the loads are not accurate enough to estimate all possible designs. Off course, in the optimization, just the end result has to be estimated in an accurate way. In Figure 4-14a a preview of an optimization run is given. In this optimization all constraints



(a) Performance for the damage equivalent bending moment during production.

(b) Performance for the extreme bending moment.

Figure 4-13: Performance with and without using Proper Orthogonal Decomposition.



(a) History of the objective and the mass.

(b) Comparison of the utilization ratio and the buckling ratio with the estimated and calculated loads.

are used and the tower length, tower diameter and pile penetration depth are free. From this figure, it seems that the optimization results in a promising design. It shows a decrease in mass of about 9%. In Figure 4-14b the utilization and buckling ratio for the calculated and the estimated loads are compared. When using same weighting as for calculating the performance, the utilization ratio shows a weighted mean error of about 11%. The extreme load shows a weighted mean error of about 13 %. This shows that the optimization algorithm has found an optimum on a place where there is a relatively large error between the estimated and the calculated loads.

To prevent this phenomena during the optimization, the interaction between PSO and the estimators is changed in the following way: After every generation in the PSO, the training data for the construction of the estimator is extended with a calculated result on the place of the best individual. This is shown in Figure 4-15.

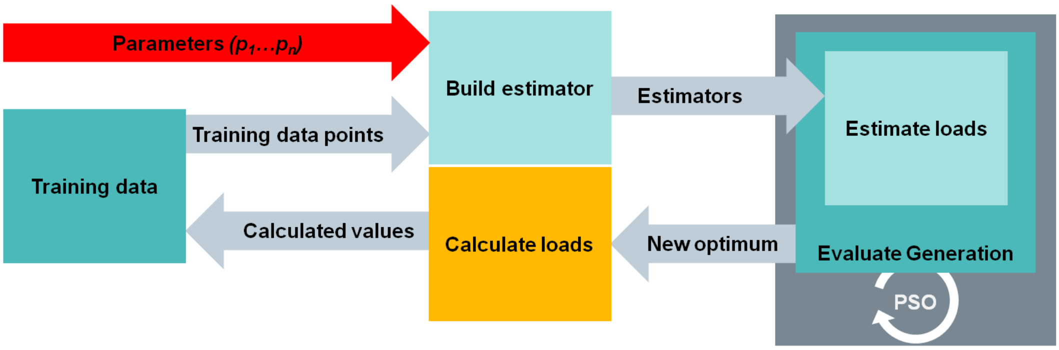


Figure 4-15: Overview of the interaction between PSO and the estimator. p_1 till p_n are constant.

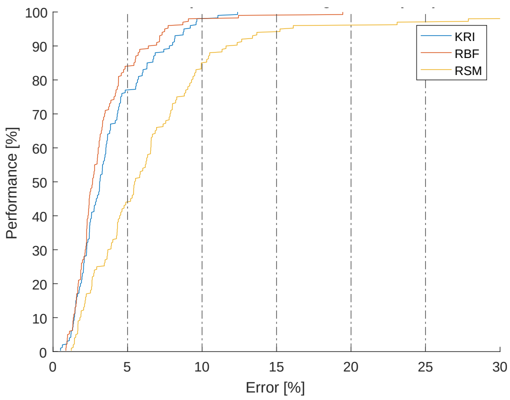
It is assumed that the added point will not change the characteristics of the response surface itself. Therefore the previously defined parameters p_1 till p_n can be re-used. Because during this research, PSO is responsible for the definition of the most important dimensions, during Gradient Based Optimization(GBO) the estimators are not updated. Because GBO is generally using just a few iterations and does little adjustments to the design, the estimations will stay close to the estimation in the final step of PSO and therefore the errors in the estimation will stay small. This is verified with the weighted error of both the utilization and the buckling ratios.

Summary

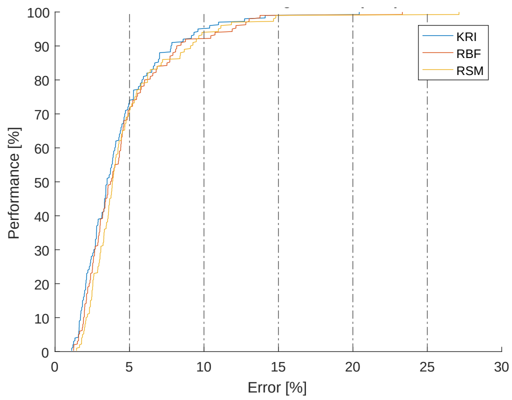
This chapter started with explaining the choice to estimate the loads. After looking to different input options and eigenmode implementations it is chosen to use as input for both the DEL and the extreme load, $\boldsymbol{x} = \begin{bmatrix} \omega_1 & D & \phi_1 & \gamma_1 \end{bmatrix}$. For ϕ_1 the deflections at towertop, halfway towertop and SWL, a grid around SWL (SWL + $\begin{bmatrix} 0 & 2 & 4 & 6 \end{bmatrix}$ meter), mudline, halfway mudline and pile tip and the pile tip itself are used. Using this input, RBF gives in all cases the best performance. For the estimation of the loads for structures with a free tower length and a free PPD, for the DELs, about 30% of the designs can be estimated with errors smaller than the maximum error, for the extreme loads this is about 60%.

The estimation of the loads with less degrees of freedom gives better results. This is shown in Figure 4-16. For the DELs, RBF gives the best results, although Kriging comes close to it. For the extreme loads, the three methods have a comparable performance. It is remarkable that even RSM comes close to the other estimators!

Investigations showed that even a relatively high performance, as shown in Figure 4-16, can cause problems in the optimization. For this reason, an estimator updating method is proposed. This can be used the optimization for continuously improvement of the accuracy of the estimations.



(a) Performance for the damage equivalent bending moment during production.



(b) Performance for the extreme bending moment.

Figure 4-16: Performance for several estimators for the estimations of loads of structures with fixed tower length and fixed PPD.

Chapter 5

Optimization results

As formulated in the objective of this thesis, this research has as goal to "develop an efficient approach for the optimization". Previous chapters have explained the optimization and the estimation of loads. To show the performance, qualities and drawbacks of the combined optimization and load estimation, it is used on 4 case studies. These are defined first, whereafter they are optimized using the proposed optimization stategy in combination with the estimator updating. At the end, a summary of the results is shown from which some conclusions are drawn.

5-1 Introduction case studies

The case studies that are used in this thesis serve several purposes. At one hand they have to show the value of the proposed optimization strategies and at the other hand they had to show options for further reducing the mass of the support structure. To do this, several combinations of constraints and free variables. In Table 5-1 it is shown which constraints are used and which variables are fixed for the different case studies. From now on, referring to a case is done with the abbreviation as given in the table. The constraints and free variables are used or fixed for the following reasons:

FDB_ This is the case with all common constraints, while having maximum freedom in design during the optimization. The results of this case are most interesting for the research in OWT optimization, to see if large mass reductions are possible by introducing possibly large deviations of the current

Table 5-1: Overview of the case-studies which are used in this thesis. It is indicated whether constraints are used and if parameters are fixed. The abbreviations are chosen in such a way that the used constraints (before the '_') and the fixed variables (after the '_') can be derived.

	Constraints			Fixed variables		
Abbreviation case:	Fatigue	Deflection	Buckling	Length tower	Diameter tower	Penetration depth monopile
FDB_	Yes	Yes	Yes	Free	Free	Free
FD_P	Yes	Yes	No	Free	Free	Fix
FDB_LSP	Yes	Yes	Yes	Fix	Fix	Fix
FDB_LP	Yes	Yes	Yes	Fix	Free	Fix

standard design of an OWT. Besides, this case is used to conclude that the tower length can be fixed on forehand to simplify the optimization problem.

FD_P The FDB_-case is used to fix the length of the tower on forehand. To reinforce and explain this conclusion further this case is used.

FDB_LSP This is the case which is closest to the current design process. Therefore, the length of the tower, the diameter of the monopile and the PPD are defined a priori. So a fair comparison can be made by looking to these results with respect to the current support structure.

FDB_LP The diameter is kept constant in the current design process. In this optimization case, it is free. By comparing the results of this case with the FDB_LSP-case, it can become clear how far the design can be improved when this diameter can change.

5-2 Case study 1: FDB_

Results

In Figure 5-1 the history of the mass is shown for the FDB_-case optimized with 90 individuals in PSO. The final result shows a mass reduction of about 3.34%. The GBO did three iterations resulting

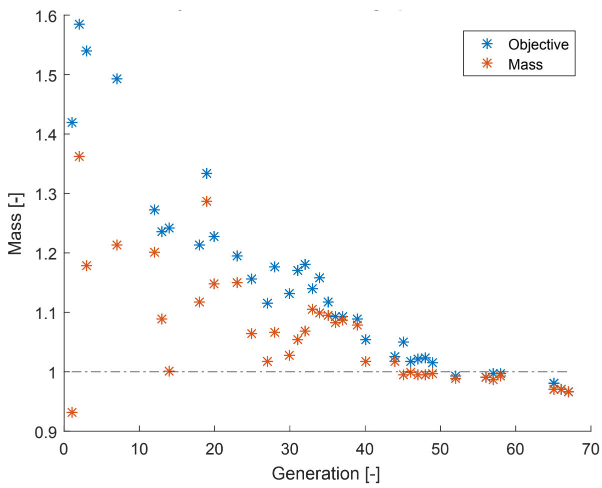


Figure 5-1: History of the objective and the mass for the FDB_-case.

in a decrease of the mass of about 1%. At that position, the GBO stopped because the last decrease in mass was smaller than the criteria set.

In Figure 5-2 the resulting design is shown. When looking to the constraint ratios, the buckling ratio and the D/t ratio both stay far below 1. The angle of the conical parts is feasible as well. The fatigue ratio is going to 1 on several places over the height. From the maximum deflection constraint, the constraint ratios for the rotations at mudline and at pile tip where both almost one as well. From this, it can be concluded that fatigue in combination with the maximum deflection constraint where design-driven.

Computing FLS and ULS without estimator

In Figure 5-3 the difference is shown in the utilization and buckling ratio for the calculated and the estimated loads. The fatigue ratio shows a weighted mean error of about 0 %. The buckling ratio shows a weighted mean error of about 3 %. This error can have two reasons (these are valid for the errors in the results of the other cases as well):

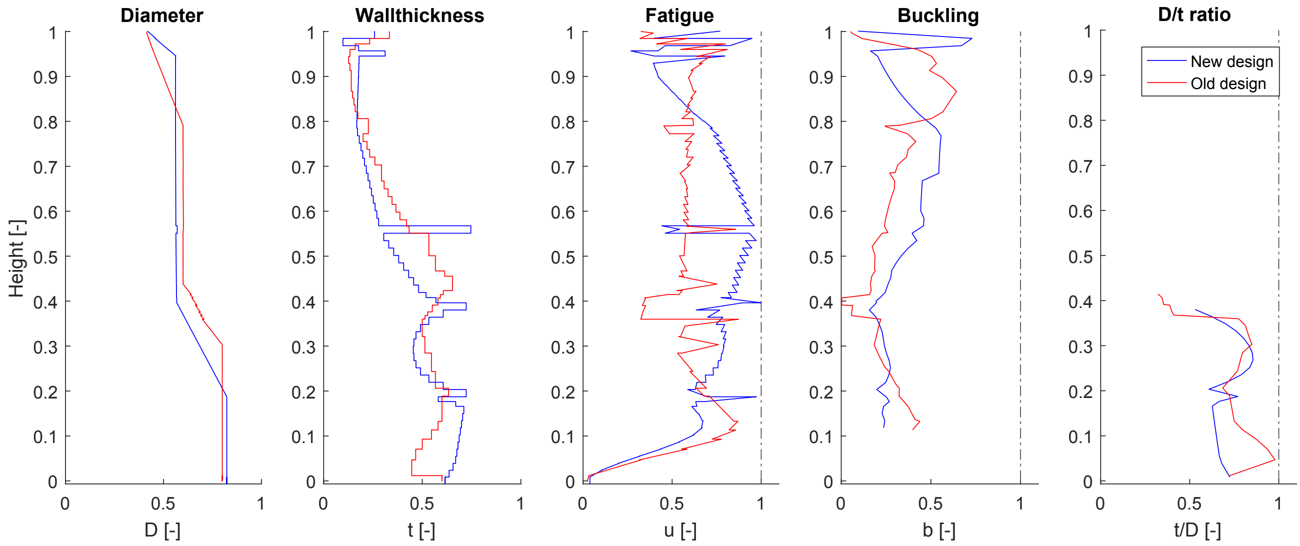


Figure 5-2: Final design for the FDB_-case.

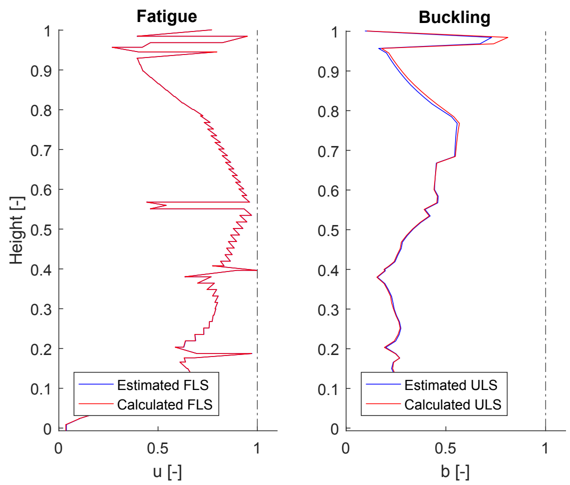


Figure 5-3: Comparison of utilization and buckling ratio for estimated and calculated FLS and ULS for the FDB_-case. The fatigue ratio has a weighted error of about 0% and the buckling ratio a weighted error of about 3%.

- Because in the GBO phase, no error updating is applied, as mentioned in Section 4-4, the final result is not part of the training data points used for the construction of the estimator.
- As shown in Figure 4-2, there is a mapping back and from the estimation grid. This can introduce errors as well.

Because the error stays below the bounds as specified in Section 4-1, the assumption that the GBO, due to its local nature, will have small influence on the loads is justified. Therefore, estimator updating in the GBO phase is not needed. Because the following optimization runs show comparable results, for brevity just the weighted error of those cases is mentioned, without presenting the results in a figure.

Tower length

In Figure 5-4 the history of the length of the tower and the PPD are shown for all individuals as well as for the global best individual. The swarm is converging to a tower length of about 1 which

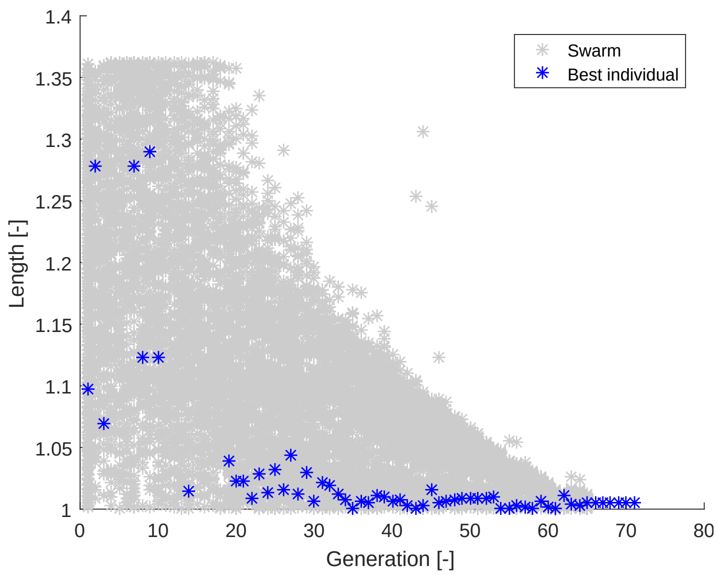
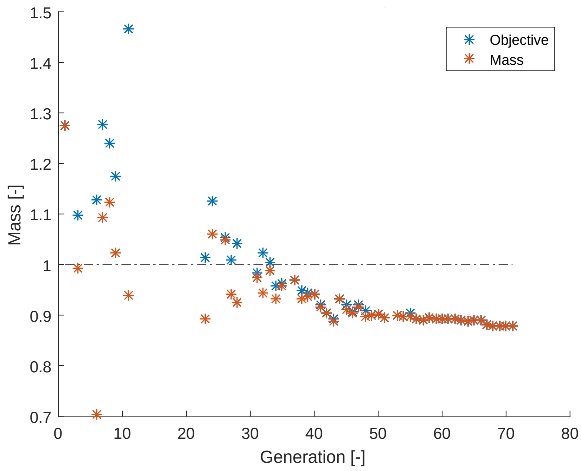


Figure 5-4: History of the length of the tower. It is normalized with respect to the minimum length.

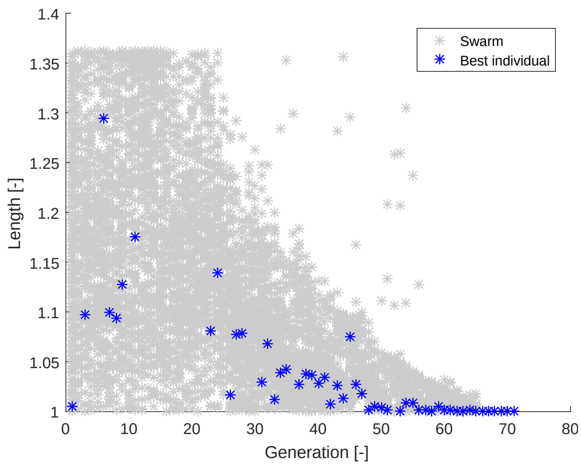
is the minimum value for the length. First investigations without the wind load scaling, as explained in Section 2-4-2, gave for this optimization tall towers as result. This resulted in soft-soft designs for the support structure, as explained in Figure 2-2. The use of the more conservative wind load scaling gives, in all optimization runs for the FDB_-case, a length of the tower about equal to the minimum tower length. From this it can be concluded that a shorter tower is always preferable for comparable monopile founded support structures, when the benefit from a higher RNA, a higher Annual Energy Production (AEP), is neglected. Therefore, to simplify the optimization, the length of the tower can be fixed before the optimization starts.

5-3 Case study 2: FD_P

In Figure 5-5 and Figure 5-6 the results are shown for the FD_P case. The PSO used 120 individuals. The mass saving compared to the current design is about 11%, although this design is not necessarily realistic because the buckling constraint is not taken into account. The utilization ratio shows a weighted mean error of about 2%, so the design is feasible for the fatigue constraint.



(a) History of the objective and mass.



(b) History of the length of the tower.

Figure 5-5: History of some properties during the optimization of the FD_P-case.

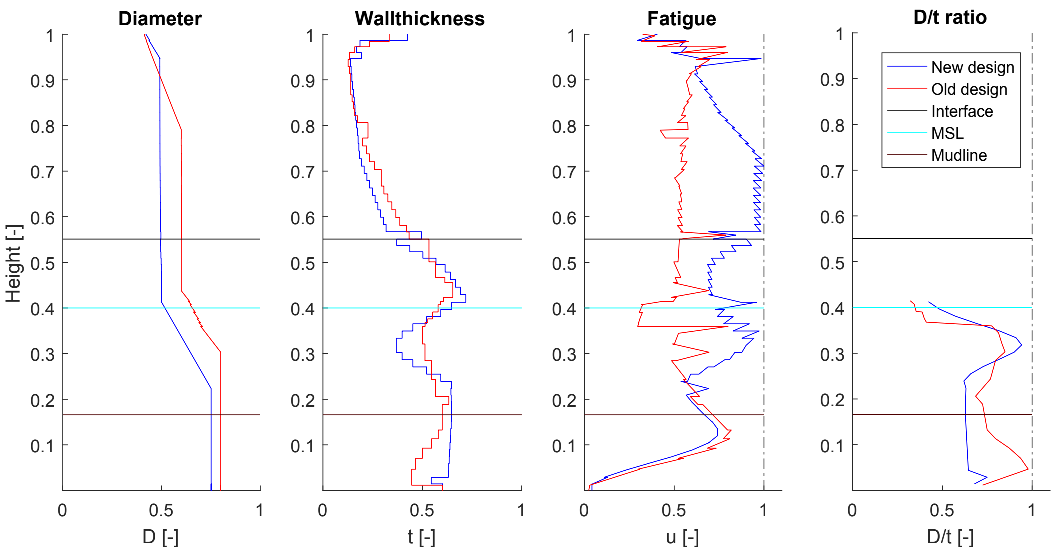


Figure 5-6: Final design for the FD_P-case.

In Figure 5-5b, from generation 1 till 20, no convergence to a specific length is visible. This is caused by the weights of the fly directions as explained in Section 2-3-4, which forces the exploration of the full design domain. From generation 20, the optimization starts converging.

Tower length

Initial investigations without wind load scaling done without buckling constraint showed tall soft-soft designs as for the FDB_-case. While using wind load scaling, the length of the tower is decreasing during the optimization till it reaches the minimum tower length, as shown in Figure 5-5b. This minimum tower length is equal to the length of the tower of the current design. From this, it can be concluded that the length of the tower is not driven by buckling. Besides this, the same statements can be made concerning fixing the tower length as for the FDB_-case.

5-4 Case study 3: FDB_LP

In Figure 5-7 and Figure 5-8 the results are shown for the FDB_LP case. The results show a decrease

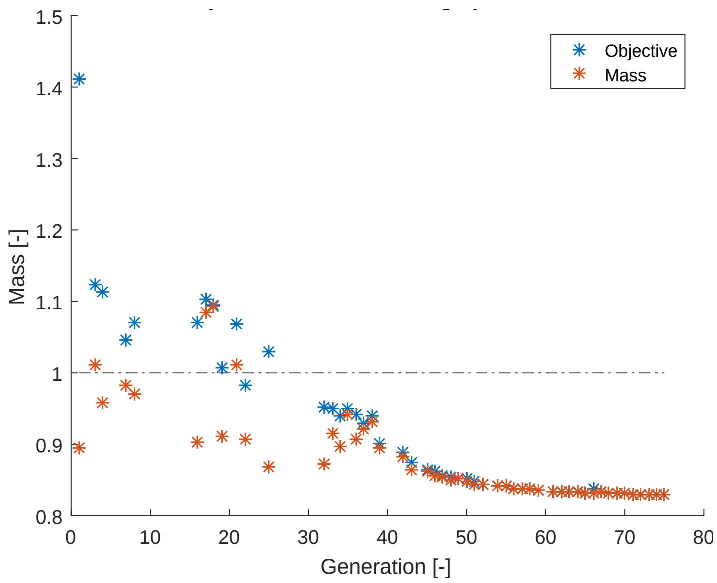


Figure 5-7: History of the objective and mass for the FDB_LP-case.

in the mass of about 17.0% compared to the current design. GBO made 9 steps, although these steps all result in a very small mass decrement of about 0.2%. As visible in Figure 5-8, the final design has no constraint violations for fatigue and buckling. Fatigue and the maximum rotation of the monopile are, as for the FDB_-case, driving the design because they have normalized ratios of about 1.

In Figure 5-7, it is visible that during the optimization, the mass and objective are not always decreasing. This is caused for two reasons:

- As mentioned in Section 2-3-4, the weight of the constraints is increasing during the optimization. For this reason, the best design position in generation n can have a smaller objective than the best design positions in generation $n+1$.
- When a new optimum is found, that design position is calculated and added to the estimator. Even though the design seems to be feasible when evaluating it using the estimator, the next generation it can become infeasible due to the updated estimator.

When comparing the current design and the design found by the optimizer, some things can become clear:

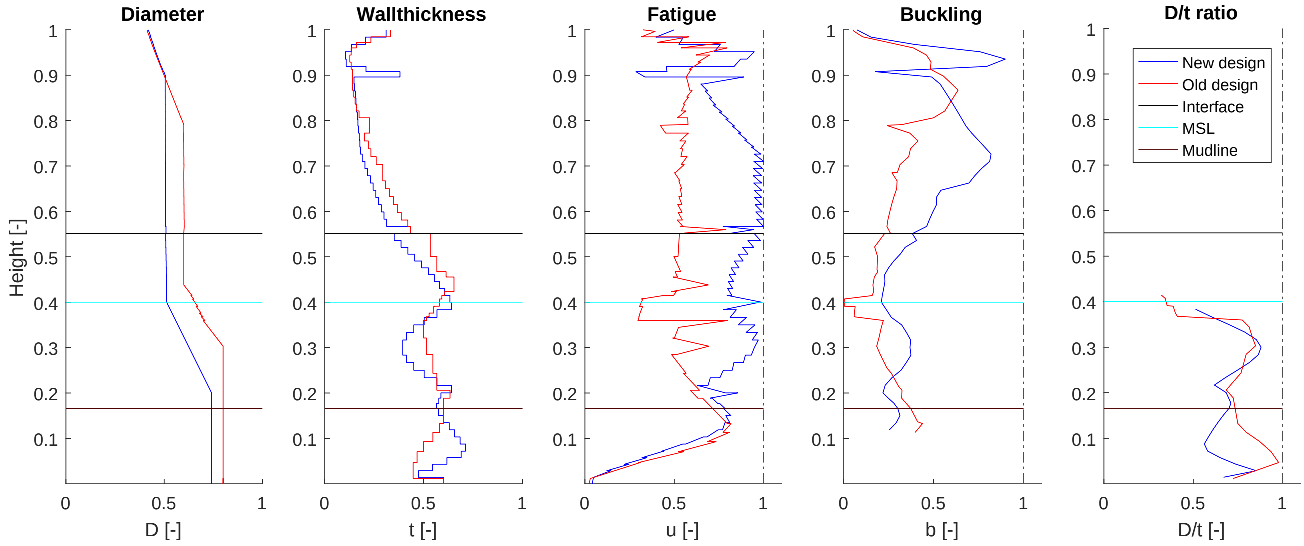


Figure 5-8: Final design for the FDB_LP-case.

- The diameter of the tower found by the optimizer is much smaller than the current tower design.
- The diameter of the monopile found by the optimizer is much smaller than the monopile diameter in the current design.
- The conical part starts at a lower level (about equal to SWL) than the conical part of the current design.
- The wall thickness of the monopile of the optimized structure around mudline is comparable with the wall thickness of the current design, although it stays increasing below mudline.
- The t/D-ratio just above pile tip is not driving the wall thickness of the monopile any more.

The utilization ratio based on load estimations shows a weighted mean error of about 0%. The buckling ratio, a weighted mean error of about 0%.

As mentioned in Section 2-2-5, the eigenfrequency constraints for the 1P and 3P frequency and the angle of the conical parts are not used. When checking these constraints, the first eigenfrequency of the structure is above the 1P frequency region. As expected, the 3P frequency region is in between the first and the second eigenfrequency of the structure. The angle of the conical part is around 2°. From this, it can be concluded that the relaxation on these constraints have not given any problems.

5-5 Case study 4: FDB_LSP

This case is done to compare it directly with the current design. Therefore, the diameter and the length of the tower as well as the PPD are fixed from the beginning.

Results

In Figure 5-9 and Figure 5-10 the results are shown for the FDB_LSP case. It takes a long time, to find new optima after the first two generations. This shows a clear difference between PSO and GBO: GBO is always be able to find a better position if the gradient on a certain position, which is not an optimum, is known. For PSO, this is not necessarily true because of the randomness in the behaviour and the emphasis on the own fly direction of the individuals. From the start of the optimization, these are relatively high to explore the full design space. After they are decreasing, the individuals are forced to converge and generally new optima are found. This is clearly visible in this case.

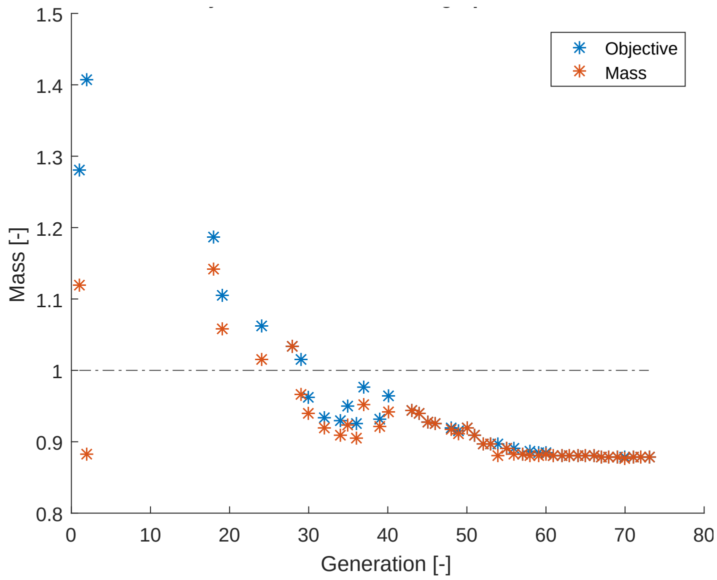


Figure 5-9: History of the objective and mass for the FDB_LSP-case.

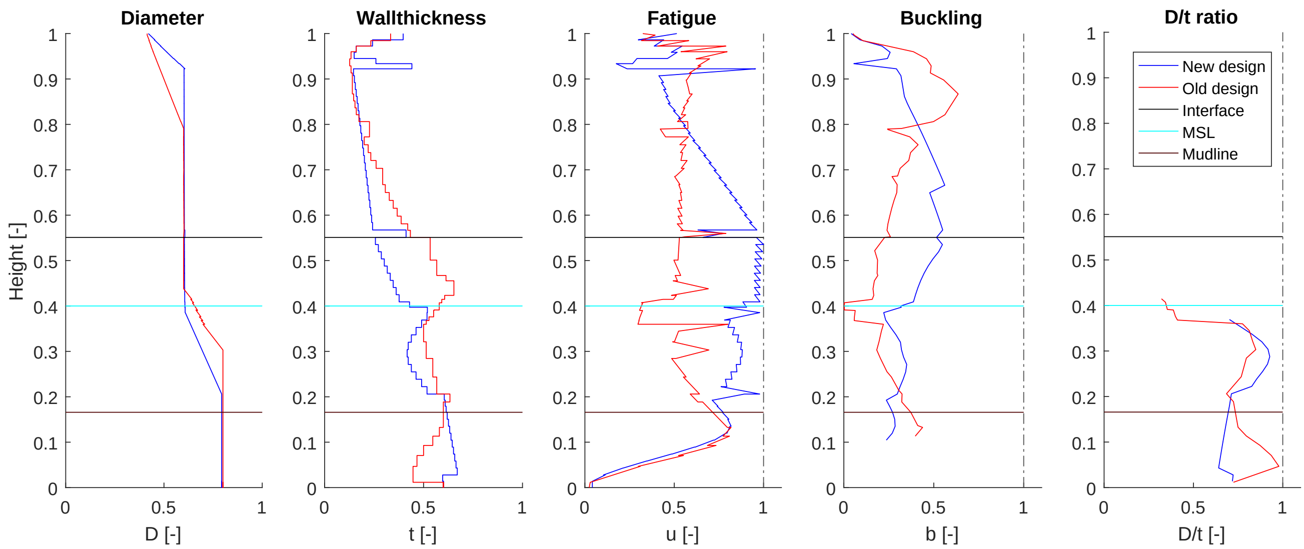


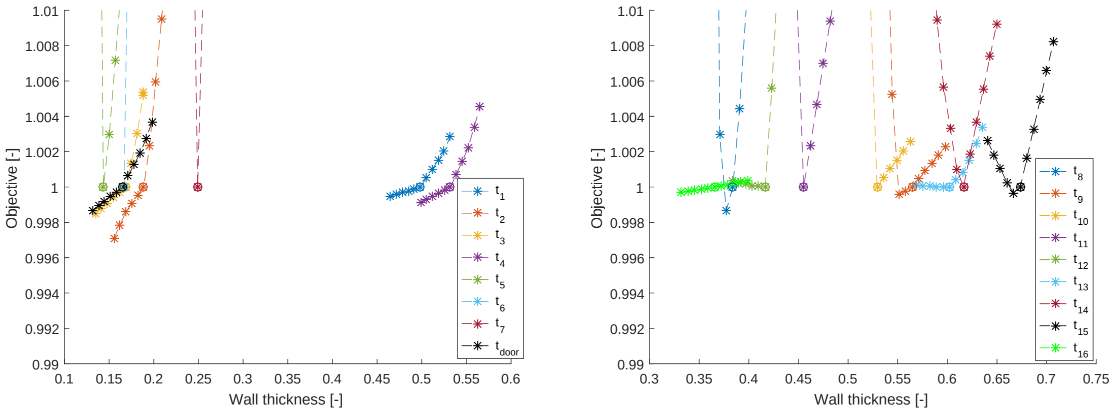
Figure 5-10: Final design for the FDB_LSP-case.

The utilization ratio shows a weighted mean error of about 3%. The buckling ratio shows a weighted mean error of about 0%. It is remarkable that in this design and even in the designs found for the other optimization cases, the wall thickness below mudline is not rapidly decreasing as it is for the current design.

The designs found for this case and the design found for the FDB_LP-case are both feasible designs in the FDB_- and the FD_P-case. So it can be concluded that the FDB_- and the FD_P-case were not able to find a global optimum. Because of the free PPD and tower length in the FDB_-case and the free tower length in the FD_P-case, it can have a larger variability in the eigenfrequency and as result from this, in the loads as well. Therefore, the FDB_-case and the FD_P-case are expected to have more local optima than the FDB_LP- and the FDB_LSP-case have, which substantiate the choice for fixing the tower length.

Sensitivities to design variables

For the result as shown in Figure 5-10, the sensitivities to some design variables are calculated. These are shown in Figure 5-11. From this, it can become clear that the stopping criterion stops the GBO relatively early. The actual exit condition of the GBO is the improvement of the objective during the last gradient-base step. Refining this criterion will possibly lead to a further improved design. When



(a) Wall thickness's of the tower.

(b) Wall thickness's of the monopile.

Figure 5-11: Objective around the found positions of several design variables for the FDB_LSP-case. They are obtained by calculating the objective with load estimations for small deviations (marked with a star) from the found position (marked with an encircled star). This is done for every variable separately.

looking to the results of all cases, the wall thickness's of the conical part at the top of the turbine show an unexpected development over the height. Combining this with the found sensitivities, it can be concluded that the optimizer did not find an optimum for these variables.

Summary

In this summary, some conclusions and remarks are shown regarding all optimization cases.

Decrease of mass
In Table 5-2, the results for all optimization cases are shown. As explained before, the FDB_-case and the FD_P-case were not able to find a global optimum. For these cases, GBO has a relatively large contribution of about 1% to the mass decrease in the final result. For the FDB_LP-case and the FDB_LSP-case this is respectively 0.2 and 0.3%. For the case which is the closest to the current design process, a mass reduction of about 12% was found. When the diameter of the tower was a design variable as well, a reduction of about 17% was obtained.

Table 5-2: Summary optimizations

	PSO			GBO	
	n_{ind} [-]	n_{gen} [-]	w_{end} [-]	n_{steps} [-]	w_{end} [-]
FDB_	90	65	0.981	3	0.970
FD_P	120	65	0.890	6	0.879
FDB_LP	150	65	0.832	9	0.830
FDB_LSP	150	65	0.881	8	0.878

Reduction of wave loads
When looking to the optimized designs, it is the question: Why is it possible to decrease the mass while they are still feasible. A part of the reduced mass is in the reduction of the loads. Some important properties that influence the wave loads on the structure are given in Table 5-3 (the diameter is also visible in the figures with the results). The eigenfrequency, the diameter at SWL and the amplitude of the first (mass normalized) modeshape, have decreased for the optimized designs. For the eigenfrequency, this means that it is closer to the high energetic part of the wave spectrum, which increases the loads. A decreased diameter causes decreased wave loads, as Morisons equation tells that the diameter is linearly related to the drag forces and quadratic in the inertia forces. For the current design, below mudline, the wall thickness is decreasing rapidly to the pile tip. Therefore, the D/t-ratio is driving the design of the monopile. For the optimized designs, the wall thickness decreases much less below mudline. For the FDB_LSP-case, it is even increasing. This will cause a stiffer monopile, which results in a decreased modal amplitude. A decreased modal amplitude at SWL will decrease the wave loads, as less wave energy is transferred to the structure.

Table 5-3: Some properties of the resulting designs compared to the current support structure.

	ω_1 [-]	D_{SWL} [-]	ϕ_{SWL} [-][1]
Current design	1	1	1
FDB_LP	0.852	0.898	0.805
FDB_LSP	0.757	0.753	0.761

Fixed variables
As explained, to reduce the design space, the tower length can be fixed. In the current design process, the PPD is fixed in an early stage as well. For this reason this variable was fixed in the FDB_LP- and the FDB_LSP-cases as well. In Figure 5-12, the history of the PPD for two optimization runs is shown. This shows two different PPD's, which means that the PPD is design specific. For this reason it is not a good choice to fix the PPD on forehand.

[1] Mass normalized eigenmode.

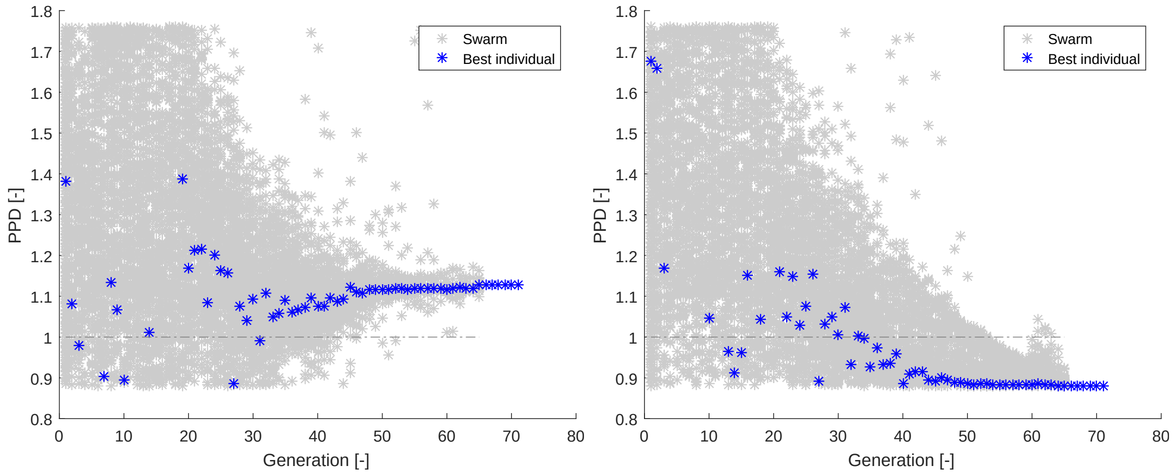


Figure 5-12: History of the PPD for two optimization runs of the FDB_-case. They are normalized with respect to the PPD of the current design.

Convergence

In Figure 5-13 the results of several optimization runs are given. From this figure, it can be concluded that even with the highest number of individuals the PSO does not find a global optimum.

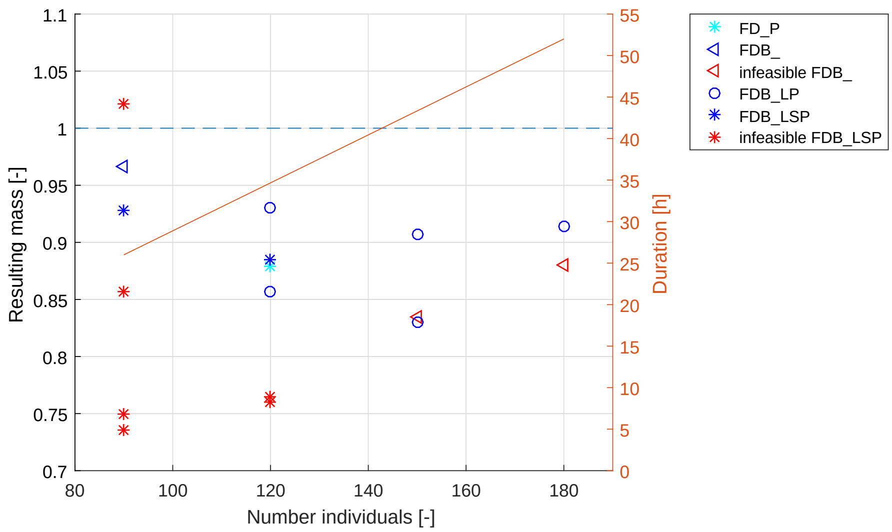


Figure 5-13: Overview of all results obtained with the proposed optimization approach. On the right axis, the computational time of an optimization run is shown.

Estimator updating

As shown in Figure 5-13, for the FDB_- and the FDB_-LSP-case, some optimized designs were infeasible. A design is marked as infeasible if is has an error in the estimated loads larger than the criteria as given in Section 4-1. When using estimator updating, this error was not always reduced. It is out of the scope of this research to study this phenomena, although first investigations showed that for the infeasible designs, one of the structural inputs of the estimators was close to bounds of the domain. Updating of the estimator close to the bounds has just local effects. Therefore, the next iteration in the optimization, the new design was just shifting closer to the bound. Where these step were repeated.

Chapter 6

Conclusions

This chapter shows the most important conclusions of this research. They are split up in three parts, according to the tasks defined in Chapter 1. First some conclusions are made regarding the load estimations. After this, some conclusions are drawn regarding the optimization. Then the combination of optimization and estimation is discussed. The chapter is finished with an overall conclusion by placing the results of the optimization into the perspective of offshore wind industry.

6-1 Load estimations

2. Investigate estimation methods and develop an accurate way for estimation of loads.

Accuracy of the load estimations

From Chapter 4 follows that it is possible to compute an estimator for a fast estimation of both DELs and extreme loads. In the estimator the input vector, $\boldsymbol{x} = \left[\begin{array}{cccc} \omega_1 & D & \boldsymbol{\phi}_1 & \gamma_1 \end{array} \right]$ is used, within the eigenmode $\boldsymbol{\phi}_1$, the deflections at towertop, halfway towertop and LAT, Grid I, mudline, halfway mudline and pile tip and the pile tip itself. Because the optimization found results with mean errors larger than the criteria as given in Section 4-1, estimator updating was used.

Estimator updating

For updating the estimator, the input parameters p_1 till p_n were taken equal to the parameters that were already found. The results of the updated estimator confirm the validity of this assumption. The updating process itself shows a large difference in the construction of the estimator. While updating of the RBF estimator requires just one matrix multiplication, updating of the Kriging estimator requires a new optimization run which takes much more time, despite the fact that a reasonable starting point can be chosen.

An advantage of estimator updating is that the initial estimator has less impact on the end result. As explained in Section 4-2-2, in this research the number of training points was fixed at 800. With the estimator updating this number becomes less critical. So for future use, a smaller number of training points is apparently sufficient.

6-2 Optimization

1. Develop an optimization strategy for integrated optimization of a support structure.
4. Show the performance of this optimization approach on a representative case study.

Influence of constraints and free parameters on optimization

Design-driving constraints Fatigue and the maximum rotation constraint were in all cases design-driving. Buckling and the angle of the conical parts were not found to be design-driving. Although the D/t-ratio was design-driving for the current support structure, it was not design-driving in the optimized cases. This was caused by the relatively large wall thickness of the monopile, from fatigue perspective, for the optimized designs. The larger wall thickness is possibly a cause of the decreased modal amplitude on SWL, which reduces the wave loads for both DELs and extreme loads.

Free tower length Before the wind load scaling was applied, optimization runs on the FDB_-case resulted in designs with a tower length higher than the current tower length. The eigenfrequencies of these structures were shifted to the soft-soft region. Via this way, relatively low DELs were found which resulted in feasible designs, with very low masses. Using the wind load scaling, the results for the FDB_- and the FD_P-case give a tower length of about equal to the current tower length. This is shown in Section 5-2. Therefore it can be concluded that the wind loads are driving the length of the tower. From these cases, it can be concluded that the optimization will always lead to the shortest tower length as possible within the bounds. For this reason, the tower length can be fixed before the optimization starts, which reduces the complexity of the optimization.

Free tower diameter The current design process uses a constant tower diameter to standardize processes. When comparing the results from the FDB_LSP- and the FDB_LP-case, it can be concluded that leaving this diameter free, can lead to a further decrease in mass of about 5% in this case study.

Loads in optimization

As explained in Section 2-2-1, several structural properties have influence on the loads. The results in the previous chapter show that during the optimization, these properties are adjusted in such a way that the loads decreases. In this way, the mass can decrease. Therefore, during the optimization, it is important to take the loads into account.

Optimization algorithm

a. Optimization strategy The strategy by using PSO on an objective extended with penalty functions, shows the ability to reduce the mass of the support structure. Because of the results without wind load scaling did give soft-soft designs, it can be concluded that the algorithm is able to escape from local optima.

b. Global optimum During the optimization project several optimization runs are done on the same cases. The initial positions of the designs in PSO are taken from a set of samples calculated with Latin Hypercube Sampling for an even distribution over the design space[25]. Comparing the results of the FDB_-, the FDB_LP- and the FDB_LSP-case, it can be concluded that for the FDB_- and the FDB_LP-case, for the used number of individuals, the global optimum is not always found.

Tool for initial design

This research has resulted in a tool for the initial design of support structures for wind turbines based on monopile foundations. The time an optimization takes, is dependent on the number of individuals chosen and will decrease when doing multiple optimization with the updated load estimator. When using a reasonable number of 120 individuals on one 2.6 GHz core, it will take about 35 hours. This is too long for an overnight run.

6-3 Optimization and Estimation

3. Combine Optimizer and Estimator into an efficient optimization routine.

Efficiency of using load estimations during optimization

For the FDB_LSP- and the FDB_LP-case, during PSO, 9750 design positions are evaluated. When using load calculations, this needs the same number of calculations. As shown in Section 4-2-2, 800 load calculations are done to construct the estimator, before the optimization starts. Besides these load calculations, for updating the estimator, a maximum number of design positions equal to the number of generations is calculated as well. This gives a maximum of 865 load calculations for 5850 design positions. For PSO using more generations with more individuals will increase the probability of finding the global optimum. This points out that the use of an estimator during PSO, is a very efficient combination, especially with the improved accuracy of the estimations when estimator updating is applied.

Improvement optimization time

From time perspective, during the optimization, the evaluation of the objective for one design position takes about 30 seconds. As shown in Table 2-3, using load calculations, takes about 50 to 440 seconds dependent on the use of sea states and aerodynamic damping. From this can be derived that the estimations decrease the optimization time by at minimum a half.

6-4 Bigger picture

For confidentiality reasons no precise numbers of the weights of the support structures used as case study are given. But some number can be give as an indication. As explained before, the optimization shows mass savings of around 17%. The weight of the support structure of a 7 MW turbine for the described position is typically in the order of 10^6 kg. With a steel price of $1 \sim 3$ Euro per kg steel and assuming a linear relation in between these, cost savings in the order of $170 \sim 510$ kEuro are possible per turbine.

Chapter 7

Recommendations

During this project, some drawbacks of methods, lacks in knowledge or challenges are found to be important, but too time consuming to solve as part of this research. In this chapter they are shared and can be used for possible follow-up projects. They are split up in three sections according to the set-up of this thesis: The optimization, the estimation methods and the combination of optimization and load estimation.

7-1 Optimization

Repeatedly building the structural model

During the optimization, every iteration, a new structural model is assembled. In Table 2-3 this belongs to 'Other calculations and evaluations'. As explained in Section 2-4-1, several element matrices stay constant during the optimization. In this optimization they are redefined every time, while this can be done more effectively by using constant parts of the structural matrices.

Wind loads scaling

In this research, the wind loads for the fatigue assessment are scaled as a function of the height. This method is based on practical knowledge from engineers within SWP. As mentioned in Section 6-2, the wind load and so the assumption concerning load scaling, is driving the design. For this reason, it is needed to do a verification of the wind load scaling after the optimization but it would be even better to take the wind loads into account during the optimization with more realistic wind load calculations. In these wind load calculations, 1P and 3P frequency effects are possibly taken into account, so that all important constraints are used in a reliable way.

Modelling of costs of a support structure

In this study, the length of the tower was a variable as well. It was assumed that a taller support structure, gives the same Annual Energy Production (AEP) as the support structure with the minimum length. This is not correct because a higher height of the RNA will give better wind conditions. In this optimization, this can be implemented by giving this a negative penalty, but this can give strange comparisons. For example, when a slightly too high fatigue damage is compensated by a higher AEP production. When using a cost model of the support structure these considerations can be taken into account. For example, a high fatigue damage can be translated into a decreased lifetime and a higher AEP in increased earnings during its lifetime.

Optimization based on estimation models

In the field of estimation methods, research is already done into the direct combination of estimators and optimization algorithms. A requirement for this is that the design variables, which have to be optimized, are an input of the estimation model and the output should represent directly the objective. In this thesis, there is a complicated, non-recursive function in between the design variables and the inputs of the approximation model. Furthermore, the output of the estimation model is not directly the objective function, as explained in Section 4-1. Therefore theories concerning optimization based on estimation models are not applicable. To improve the calculation time of one optimization run, an option for further research is to explore if it is possible to estimate the objective or the penalties directly.

Balance between PSO and GBO

Generally spoken GBO is computationally more efficient than PSO. For this reason there is a certain balance between PSO and GBO. In this research the emphasis is on the use of PSO, in the attempt to escape from local optima. For this reason no investigations are done to find a more efficient interaction between PSO and GBO. To get better calculation times, this can lead to a valuable improvement.

Verification results

As explained in Section 1-2, the FLS assessment is done based on frequency domain calculations. These frequency domain calculations are computationally less involved than time domain calculations but less accurate and therefore used for the initial design. To validate the fatigue life of the support structure, load calculations in time domain has to be used. This is a good way to verify the results of the optimization as well.

Decreasing the calculation time of the design tool

a. Decrease design space During this research, the bounds of the design space were taken very wide. For example, as mentioned in Section 2-2-4, the bounds on the tower and monopile diameter were taken as 4 and 10 meter. This gives the algorithm the possibility to come up with ingenious designs, but makes the optimization more difficult due to the large variability in the possible designs. For further use as optimization tool, the bounds can be taken much narrower. This will give the possibility to reduce the number of individuals used in PSO and will decrease the calculation time.

b. Parallel computing During this project all computations are done with MATLAB using just a single core (2.6 GHz). The chosen optimization algorithm, PSO, shows very good capabilities for parallel computing, because all individuals in a certain generation are independent of each other. From time perspective, this will have a significant influence. Because the algorithm is most of the time in the phase of evaluating the individuals, the computational time will be decreased by approximately a half when using two cores in parallel.

Extend this optimization project to full a wind farm

Current state-of-the-art turbines have mostly a unique, optimized monopile, while the tower is in most cases just designed for 1 till 3 design positions. With the optimization tool developed during this thesis, possibilities arise for the unique design of more combinations of tower and monopile in a wind farm. An advantage of using load estimations is that these are just dependent on the sea states and not on soil conditions. Therefore it is possible to use these estimators, also the updated ones, for the full wind farm, which is quite efficient.

7-2 Estimation methods

Relationship RBF and Kriging

When looking to the mathematical formulation of RBF and Kriging, the basis is the same. The difference is in the relative scaling of the dimensions. This has some important consequences:

- When comparing Kriging and RBF, it is needed to realize that Kriging should always be able to have the same or an increased performance compared to RBF. When this is not the case, during the construction of the Kriging-estimator, a local optimum is found.
- Choosing between RBF and Kriging can be done by comparing the performance as it is done in this research. Another way to choose between them is by looking to the available computational power.
- Due to the similarities, the error prediction (or likelihood of the estimated value) used in Kriging can be used for RBF as well, as explained in Section 3-2-8.
- For Kriging, the Kriging basis-function is not necessarily giving the best results. In this research, the Kriging basis-function was never chosen, as other basis-function gave better results.
- For future research it is good to deal with Kriging as an extended version of RBF instead of dealing with it as an estimation method on its own. In this way, Kriging can benefit from the developments of RBF and the other way around.

NURBS

In this research, NURBS was known as the only estimation method which was able to do multi-variate output estimations. Nowadays, NURBS used estimation method is still under development. Practical applications of this estimation model are lacking. This study shows some problems with the practical implementation of NURBS which will not arise when using this method just on scientific problems. For example, because most scientific problems are cheap to evaluate, positioning a new control point can be done just by looking at the error when evaluating many points.

As mentioned in Section 3-2-6, the NURBS method needs a rectangular grid to construct the estimator. When the known training points are not on a rectangular grid, an estimation, for example an interpolation, needs to be done to evaluate the values at control points. This can lead to inaccurate results. In other scientific fields, unstructured grids are becoming more common. Possibly this can be used for the creation of NURBS models as well.

7-3 Optimization and estimation

Infeasible designs

The optimization algorithm found positions in the design space, where the estimator was not able to do reliable estimations. Even with estimator updating this could not be solved. As mentioned, a possible reason of this is that these positions are close to the bounds of the estimation inputs. Further research is needed to verify this.

Optimization based on load calculations

It is possible that load estimations deceive the optimizer. For example, when the loads for a part of the domain are underestimated due to inaccurate load estimations. This can be checked by comparing optimization runs with and without load estimators.

Using the estimators for gradients

In this research a gradient based optimizer is used after the use of PSO. The gradients are found by finite differences. This is done because the evaluation of the objective function is relatively cheap because of the use of estimators. Improving the calculations of the gradients can be done by using information which can be derived from the estimators. These can give direct derivatives of the loads to the input parameters.

Updating estimator

As explained in Section 4-4, the estimator is updated after the evaluation of every generation within PSO. This is based on the assumption that the results from the estimation has to fit the results from the calculation exactly. The computation of the DELs and extreme loads used in this research for the initial design, have a slight deviation from the more accurate results used for the final design. Therefore, a better criterion to update the estimator, is when the estimated value on a certain position is out of the trust region around the calculated value. This can be based on the likelihood of the estimated value (when using RBF or Kriging) or the error when comparing the estimated with the calculated value.

Appendix A

Introduction Optimization methods

Optimization is a topic with ongoing research in several fields like structural optimization, shape optimization and topology optimization. Due to continuously improving computational power, optimization gets available in more and more fields of research and industries.
This appendix is just to introduce some basic concepts of optimization that are used in this work. Because this thesis has as goal to apply optimization on wind turbines, it does not directly contribute to the theoretical field of optimization.
The appendix starts with the general expression of an unconstrained optimization problem, followed by algorithms that are able to find a solution for this by local or global optimization. Then constraints are added to the unconstrained problem so that a constrained problem appears. To come up with an unconstrained optimization problem penalty functions are introduced.

A-1 Unconstrained optimization

The most basic expression of an unconstrained minimization problem is:

General formulation of an unconstrained minimization problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{s.t.} \quad \boldsymbol{x}_{min} \leq \boldsymbol{x} \leq \boldsymbol{x}_{max}$$

Generally spoken, unconstrained problems are more simple to solve then constrained optimization problems. Several methods to come up with a solution of the problem exist. They can be split up in Local optimization algorithms (or Gradient Based Optimizers) and Global optimization algorithms.

A-1-1 Local optimization algorithms/Gradient based optimizers

These can be further itemized in 3 categories: 0^{th}-order methods, 1^{th}-order methods and 2^{th}-order methods. The difference between the categories is how they use information about the gradient. The

0^{th}-order methods uses no derivatives, the 1^{th}-order methods uses the first derivative, while the 2^{th}-order methods uses the first and the second derivative. In general, using a gradient simplifies the search for an optimum, although the optimum found is not necessarily a global optimum. This is because the algorithm starts with a limited number of designs and it does not allow an increased objective, which clashes with the statement of Jones as given in Section 2-3-2.

0^{th}-order methods (direct-search methods)

Cyclic coordinate search/Powell's conjugate directions These methods uses line searches in one direction, which changes per iteration step. In Cyclic coordinate search the direction is just formed by a dimension defined by one design variable. For Powell's conjugate directions, the search direction is spanned by the previous two search directions including their optimum.

Nelder-Mead simplex method In this method steps are made to an optimum by starting with $n+1$ points and their objectives, in a n-dimensional design space. During every iteration, the worst point is replaced by a reflection through the centroid of the other points.

1^{st}-order methods (descent methods)

When the derivative on a certain point is known, it is easy to choose a direction to look into when the goal is to decrease the objective. Methods which use this information are called 1^{st} order methods or descent methods.

When the objective is a result of a simulation and the derivative is not exactly known or very difficult to calculate, a way to estimate the derivative at that position is to use the finite difference method:

$$\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = \lim_{h \to \infty} \frac{f(\boldsymbol{x}+h) - f(\boldsymbol{x})}{h}. \tag{A-1}$$

If the design space is larger than 1-dimensional, the objective change (assuming linear behaviour) by a step α is:

$$\Delta f_{\boldsymbol{x}+\alpha \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}} = \alpha \sum_i \frac{\partial f(\boldsymbol{x})}{\partial x_i}. \tag{A-2}$$

So when a certain change in the objective (Δf_w) is wanted, this can be reached by making a step α in the direction of the objective:

$$\alpha = \frac{\Delta f_w}{\sum_i \frac{\partial f(\boldsymbol{x})}{\partial x_i}}. \tag{A-3}$$

When searching in one direction, one of the options to find an optimum on that line is to use Golden ratio stepping.

Golden ratio stepping The idea of the golden ratio is thinking the other way around. Based on a Fibonacci series, the golden section ratio ϕ can be calculated as $\phi = \frac{\sqrt{5}-1}{2} \approx 0.618$. Using the objectives at 0, ϕ, $\phi^2 = 1 - \phi$ and 1, where 0 and 1 are the outer bounds, and assuming a fully convex domain, it is easy to step in the directions of the minimum.

A-1-2 Global optimization algorithms

Some methods use the principles which can be derived from Jones statement as given in Section 2-3-2, to find a possible global optimum:

Random methods

The first category uses random numbers in several ways:

Full random method A full random method picks recursively a random position in the design space and calculates the objective on this place. The design position is saved when the new objective is smaller than the old one.

Random walk method A derivation of full random method is the random walk method. Instead of choosing random a new position, a unit direction vector in a random direction is taken. When the new position has a better objective than the old one, a step is made. The step size is decreased during the 'walk'.

Simulated annealing This algorithm is inspired by the cooling down of a metal to arrive at the arrangement of minimum internal energy. Steps are taken as in the random walk method. But instead of neglecting all designs with a higher objective as the old one, a 'bad' design is accepted if

$$r_{accept} < P = e^{\frac{f(\boldsymbol{x}_{old}) - f(\boldsymbol{x}_{new})}{T}}, \tag{A-4}$$

with r_{accept} a random number chosen, $f(\boldsymbol{x}_{old})$ the objective at the old position, $f(\boldsymbol{x}_{new})$ the objective at the new position and T the temperature (analogy with the annealing process). T is decreased during the optimization.

Random methods concluded Because 'bad' designs are sometimes accepted during the optimization and due to the random (big) stepsizes, these algorithms have possibilities to find a global optimum. But the randomness causes these algorithms to be inefficient. This can be a problem when high computational effort is needed to come up with the objective.

Biologically inspired methods

A biologically inspired method is based on a natural phenomena, which is translated into an algorithm. In these group of methods, the design is judged by looking to its fitness function. This is the same as the objective function in non-biologically inspired methods, although often the fitness should be maximized, while, in general, an objective function is minimized. Several methods exist which basically can be split up in 2 categories: Genetic algorithms and Swarm intelligence algorithms:

Genetic algorithms In 1859, Charles Darwin presented his most famous work: "On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life." In his work he describes a theory about the evolution of races and species. His theory describes the micro and macro evolution of a population in which the most fittest individuals can stay. Translating this into an optimization method gives the genetic algorithm. Instead of a description of the design domain in a continuous way, the design is represented by a genetic representation with strings containing binary or discrete numbers or genes.

During the optimization a certain population is 'simulated' containing a certain number of individuals n_{ind}, during a certain number of generations n_p. When a new generation is created some phenomena are visible:

- **Reproduction**: Exact copies of the old generation go into the new generation.
- **Cross-over**: Two or more individuals create one or more new individuals by combining the genes of all parents.
- **Mutation**: In the new generation some genes changes randomly.

The following things are needed to come up with new designs which can contain an optimum:

1. Create initial generation, by taking random strings of a certain length, dependent on the number of design parameters.
2. Follow the population in time for a certain number of cycles n_p:

(a) Create a new generation (based on fittest individuals) by:
   – Reproduction,
   – Cross-over,
   – Mutation.

(b) Calculate the fitness for every individual.

(c) Order the individuals based on fitness.

(d) Repeat (a) till (c) until the number of generation is equal to n_p.

3. Translate the strings into a design.

Swarm intelligence algorithms In Dutch, a well-known saying is ’Twee weten meer dan een’, sometimes translated in ‘Two heads are better than one’. It says that the combined knowledge of two people together, can reach further than their knowledge on its own[1]. The group of biologically inspired methods are based on this principle as well. Several comparable methods are developed, while each is given its own name [38]:

– Particle swarm optimization
– Ants Colony Optimization Algorithm
– Cat swarm optimization
– Artificial Bee Colony Algorithm
– Bees algorithm
– Imperialist competitive Algorithm
– Competitive Optimization Algorithm
– Intelligent water drops Algorithm
– Gravitational search Algorithm
– Cuttlefish optimization Algorithm
– Cuckoo Search
– Bat Algorithm
– Flower pollination Algorithm
– Etcetera...

Most recent algorithms (all except Particle Swarm Optimization and Ants colony Optimization) started to attract criticism in the research community. This is due to the fact that finding a new name for a slightly different algorithm, does not by definition change the basic principles of the algorithm[1][41]. It is not in the scope of this research to judge about this. What they clearly have in common is that they all use a population during a number of ’timesteps’ or generations. The difference between these algorithms is that they all use another strategy to create a new generation. Most important is that they create the new generation based on one or more previous generations or a part of it. It is not possible to say on forehand which of these swarm intelligence algorithms fits the best for an optimization problem, because this is depends on the nature of the problem and the shape of the fitness-surface. Besides this, the strategies for the different algorithms can be adjusted easily and extended with parts of the strategy of other algorithms. For these reasons, one of these strategies is chosen to explain in more detail.

The particle swarm optimization is the most basic one and one of the best-developed techniques [32]. Besides this, it has many extensions and applications [32][17]. PSO is inspired by flocks of birds or shoals of fish. During their life, they collect food to survive. The individuals with the easiest/best access to food are most likely the most fit individuals. During a joint flight, individuals have own

[1]Although experts opinions does not point in one direction as the titles of these books explain: "Extraordinary Popular Delusions and the Madness of Crowds." by Charles Mackay (1841) vs. "The Wisdom of Crowds" by James Surowiecki (2004).

knowledge, based on own history, and they have shared knowledge in the form of a group norm or standard[16].

Every time step, the position of each individual can be updated with

$$\boldsymbol{x}_{n+1,i} = \boldsymbol{x}_{n,i} + \boldsymbol{v}_{n+1,i}\Delta t, \tag{A-5}$$

with Δt the size of the time step (which can be any factor, dependent on the chosen $\boldsymbol{v}$) and $\boldsymbol{v}$ as the velocity of the individual. In its most basic form:

$$\boldsymbol{v}_{n+1,i} = A\boldsymbol{v}_{n,i} + r_1 B\frac{(\boldsymbol{x}_{best,global} - x_{n,i})}{\Delta t} + r_2 C\frac{(\boldsymbol{x}_{best,i} - x_{n,i})}{\Delta t}. \tag{A-6}$$

A, B and C are constants to give weight to the different parts, r_1 and r_2 are randomly chosen numbers to reflect the arbitrary will of an individual, $\boldsymbol{x}_{best,global}$ is the position of the best individual since the start of the optimization (although some researchers use the position of the best individual of the current generation[52]) and $\boldsymbol{x}_{best,i}$ is the best position individual i has ever been. In Equation A-6, $\boldsymbol{v}_{n,i}$ can be seen as the inertia an individual has during flight. This inertia prevents the individual to go in a straight line to the best position the swarm knows. $\boldsymbol{x}_{best,global}$ is the 'shared knowledge' of the group while $\boldsymbol{x}_{best,i}$ can be seen as individual knowledge. It is important to notice that this formulation of the optimization problem has no need to scale dimensions relative to each other because of the uncoupled 'time'-stepping.

With the constants A, B and C, the behaviour of the swarm can be changed. For example, a relatively high value for A gives a high accent on own direction. In this way the path the individual follows is difficult to disturb by other 'attractive' positions.

A, B and C are chosen manually based on the knowledge of the problem. When A, B and C are time dependent, the behaviour of the swarm can change during flight. In this way the goal of the flying swarm can be changed. When choosing a high value for A during the start of the optimization, a (brute force) global search is done, which becomes more intelligent when some promising positions are located. When A is decreased when going to the end of the optimization, the optimization gets a more local character. In this way the algorithm can be forced to converge towards the end of the optimization. It is important to realize that the use of Δt can be misleading when changing this together with A, B and C. It is more easy to use $\Delta t = 1$ during the optimization while just changing the parameters A, B and C.

So on, by using multiple generations the best position to find 'food', or in other words, the position with the highest fitness can be explored.

This method can be extended easily by adding up other places that can have influence on $\boldsymbol{v}_{n+1,i}$ such as neighbours, best individual of a generation, fly directions, origin, etc.

As mentioned, a certain generation has a 'shared knowledge'. This is not just in the way of an objective at the position of each particle, but also in the derivatives of the objective (in a finite difference way) if they are close to each other.

A-2 Constrained optimization

Most optimization problems have constraints. Constraints are additional equations which should be satisfied, which makes the optimization problem not fully free and in general more complicated. These additional equations can be formulated as an equality constraint, marked with $h(\boldsymbol{x})$, or an inequality constraint, marked by $g(\boldsymbol{x})$. Now the optimization problem can be formulated in the following way:

General formulation of a constrained problem

$$
\begin{aligned}
\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad & \text{s.t.} \\
& \boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0} \\
& \boldsymbol{h}(\boldsymbol{x}) = \mathbf{0} \\
& \boldsymbol{x}_{min} \leq \boldsymbol{x} \leq \boldsymbol{x}_{max}
\end{aligned}
$$

Note that the outer bounds of the design variables are not handled as constraints.

A-2-1 Penalty functions

To prevent using explicit formulated constraints in the optimization, an option is to use penalization functions. Via this group of functions, constraints can be added to the objective which gives a pseudo-objective:

$$\hat{f}(\boldsymbol{x}) = f(\boldsymbol{x}) + A f_{penalty}(B\boldsymbol{g}(\boldsymbol{x}) + C\boldsymbol{h}(\boldsymbol{x})) \tag{A-7}$$

In this function $\boldsymbol{g}$ and $\boldsymbol{h}$ are generally normalized, $f_{penalty}$ is a multiplier which brings the second term to a comparable order of magnitude than the first term, B and C are mostly constant values which give weight to the constraints in comparison with each other, while A is in general dependent on the progress of the optimization. Now this pseudo-objective $\hat{f}(\boldsymbol{x})$ gives a new optimization problem which is now an unconstrained problem:

General formulation of a constrained problem using a penalty function

$$
\begin{aligned}
\min_{\boldsymbol{x}} \hat{f}(\boldsymbol{x}) \quad & \text{s.t.} \\
& \boldsymbol{x}_{min} \leq \boldsymbol{x} \leq \boldsymbol{x}_{max}
\end{aligned}
$$

It is possible to formulate these penalization in several ways. These can generally be split up in two classes dependent on the feasibility of the design positions that are evaluated (and the feasibility of their optimum): With an interior optimum (barrier functions/interior penalty function methods) or with an exterior optimum (penalty function). Based on this several extended methods exists, for example the Augmented Langrange Multiplier Method, the Variable penalty function method and the Quadratic extended interior penalty function method. The main problem with the basic penalty functions (interior and exterior) in combination with gradient based optimization is that they cause numerical ill-conditioning due to definition dependent (non-)existing derivatives[52].

Appendix B

Common terms in statistics

In this appendix, some common terms that are used in statistics are explained. This is done to give the interested reader some insights as well as to align with literature.

Normalization of data

Input data can exist in different domains. This can result in different orders of magnitude for different dimensions. To give the estimators better chance to make use of all dimensions in a comparable way, a normalization function f can be used. Two common normalization functions are a projection on $[0, 1]$ and using mean and standard deviation:

1. Projection on $[0, 1]$ A data set $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & ... & x_n \end{bmatrix}$ can be projected into another dimension with

$$f(x) = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{B-1}$$

where x_{min} and x_{max} are respectively the minimum and maximum of the data.

2. Using mean μ **and standard deviation** σ Another option for the projection is

$$f(x) = \frac{x - \mu}{\sigma}, \tag{B-2}$$

where μ and σ are respectively the mean and standard deviation of the data.

Note that using a normalization is not necessarily improving an estimation! If some data has a larger influence on the output, it can be beneficially to have higher values for this data. The specific parameter in Kriging which optimizes a parameter for every basis-function can be seen as a normalization as well to get comparable values for all dimensions.

Maximum Likelihood Estimation

For a data set that consists n data points, if the errors ε_1 till ε_n are randomly distributed according to a normal distribution with the standard deviation σ, the probability is[9]:

$$P = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}n}} \prod_{i=1}^{n} \{ e^{-\frac{1}{2}\left(\frac{y^{(i)} - \hat{y}(\boldsymbol{x}, \boldsymbol{w})}{\sigma}\right)^2} \varepsilon \}. \tag{B-3}$$

Maximizing this or minimizing the negative of its natural logarithm for a constant σ and ε gives:

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n} \left[y^{(i)} - \hat{y}(\boldsymbol{x}, \boldsymbol{w}) \right]^2, \tag{B-4}$$

which is known as the least squares criterion.

Variance
The variance in a data set is defined as (corrected sample variance):

$$\mathrm{var}(\boldsymbol{y}) = n_t \sum_{i=0}^{n_t} x_i^2 - \left(\sum_{i=0}^{n_t} x_i \right)^2 \tag{B-5}$$

$$= \frac{1}{n_t - 1} \sum_{i=1}^{n_t} (x_i - \mu)^2 \tag{B-6}$$

$$= \frac{1}{n_t - 1} \sum_{i=1}^{n_t} (x_i^2) - \mu^2. \tag{B-7}$$

Standard deviation
Following from the variance, the standard deviation is defined as (corrected sample standard deviation):

$$\sigma(\boldsymbol{y}) = \sqrt{\mathrm{var}(\boldsymbol{y})}. \tag{B-8}$$

Covariance
The covariance between two data sets $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ is defined as:

$$\mathrm{cov}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = n_t \sum_{i=0}^{n_t} y^{(i)} \hat{y}^{(i)} - \sum_{i=0}^{n_t} y^{(i)} \sum_{i=0}^{n_t} \hat{y}^{(i)} \tag{B-9}$$

$$= \frac{1}{n - 1} \sum_{i=1}^{n_t} (y^{(i)} - \mu_y)(\hat{y}^{(i)} - \mu_{\hat{y}}) \tag{B-10}$$

$$= \frac{1}{n - 1} \sum_{i=1}^{n_t} (y^{(i)} \hat{y}^{(i)}) - n_t \mu_y \mu_{\hat{y}}. \tag{B-11}$$

Or, in another way:

$$\mathrm{cov}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = E[(\boldsymbol{x} - \mu_x)(\boldsymbol{y} - mu_y)] \tag{B-12}$$

$$= E[\boldsymbol{xy}] - \mu_x \mu_y. \tag{B-13}$$

Now the correlation can be derived as:

$$\mathrm{cor}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{\mathrm{cov}(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\sigma_x \sigma_y}. \tag{B-14}$$

Correlation coefficient
A correlation coefficient is a number to show how several datasets are related to each other. It tries to give a quantitative number to that. It can be calculated by:

$$r^2 = \left(\frac{\mathrm{cov}(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\sqrt{\mathrm{var}(\boldsymbol{y}) \mathrm{var}(\hat{\boldsymbol{y}})}} \right)^2. \tag{B-15}$$

Root Mean Square Error(RMSQ)
The Root Mean Square Error is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=0}^{n_t} (y^{(i)} - \hat{y}^{(i)})^2}{n_t}}. \tag{B-16}$$

Cross-validation

Splitting the data in q roughly equal subsets, then removing each of these subsets in turn and fitting the model to the remaining, aggregated, $q-1$ subsets. More formally, if a mapping $1,...,n \rightarrow 1,...,q$ describes the allocation of the n training points to one of the q subsets and $\hat{y}^{-\zeta(i)}(\boldsymbol{x})$ is the value (at $\boldsymbol{x}$) of the predictor obtained by removing the subset $\zeta(i)$, the cross-validation measure, which is employed here as an estimate of the prediction error is

$$\varepsilon = \frac{1}{n}\sum_{i=1}^{n} L\left(y^{(i)}, \hat{y}^{-\zeta(i)}(\boldsymbol{x}^{(i)}, \boldsymbol{w})\right). \tag{B-17}$$

When the loss function $L(a,b)$ is defined as the squared error, this becomes:

$$\varepsilon_{cv}(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n} \left[y^{(i)} - \hat{y}^{-\zeta(i)}\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right)\right]^2. \tag{B-18}$$

Appendix C

Non-uniform Regression B-splines algorithm

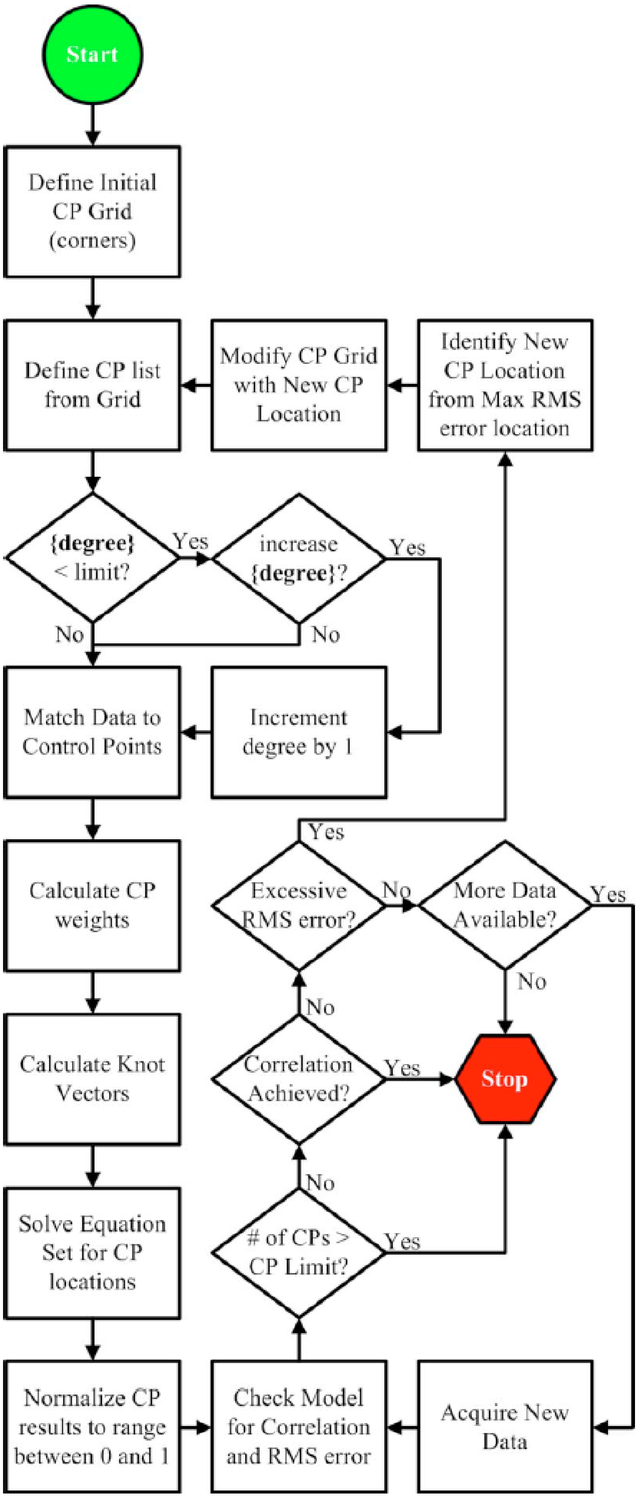


Figure C-1: The basic fitting algorithm used in HyPerMaps to define a HyPerModel iteratively adds control points to the control net to reduce the maximum error in the metamodel. The model is refined until a stopping criterion is achieved[47].

Appendix D

Damage Equivalent Load (DEL)

The following information has to be known to come up with the damage equivalent load of the support structure as result of several environmental conditions[35][55]:

- Frequency response structure
  - Modal analysis
  - Modal damping ratios
  - Mechanical transferfunctions (mode dependent)
  - Hydrodynamic transferfunction
- Loads in frequency domain

This frequency domain information can be combined to get the response spectrum of the structure dependent on the load spectrum. To derive fatigue damage from this, Dirliks method can be used. Next, a brief overview of this method is given[5]:

Application of Dirliks method
Dirliks method is comparable with rainflow counting in time-domain. It combines fatigue damage at different load levels and with a different number of cycles. This method is needed because the loads in frequency domain $S_{l,k}$ dependent of the frequency f show a broad-band spectrum. The steps needed are:

1. **Calculate spectral moments:**

$$m_n = \int_0^\infty f^n S_{l,k}(f) df \approx \sum_{j=1}^{n_f} \Delta_{f_j} f_j^n S_{l,k}(f_j), \tag{D-1}$$

$$\Delta_{f_j} = f_{j,2} - f_{j,1}, \tag{D-2}$$

$$f_j = \frac{f_{j,2} + f_{j,1}}{2}. \tag{D-3}$$

with n_f as the number of frequency bins, f the frequency ans $S_{l,k}$ the power spectral density. The number of frequency bins is strongly correlated to the total calculation time from this method.

2. **Calculate irregularity factor:**

$$\gamma = \frac{m_2}{\sqrt{m_0 m_4}}. \tag{D-4}$$

3. **Calculate Dirliks constants:**

$$\alpha = \frac{m_1}{m_0}\sqrt{\frac{m_2}{m_4}}, \tag{D-5}$$

$$D_1 = \frac{2(\alpha - \gamma^2)}{1+\gamma^2}, \tag{D-6}$$

$$R = \frac{\gamma - \alpha - D_1^2}{1 - \gamma - D_1 + D_1^2}, \tag{D-7}$$

$$D_2 = \frac{1 - \gamma - D_1 + D_1^2}{1 - R}, \tag{D-8}$$

$$D_3 = 1 - D_1 - D_2, \tag{D-9}$$

$$Q = \frac{1.25(\gamma - D_3 - D_2 R)}{D_1}. \tag{D-10}$$

4. **Calculate Probability Density Function (PDF) (core Dirliks method):**

$$p(L) = \frac{1}{2\sqrt{m_0}}\left[A_1 + A_2 + A_3\right], \tag{D-11}$$

$$A_1 = \frac{D_1}{Q} e^{\frac{-Z}{Q}}, \tag{D-12}$$

$$A_2 = \frac{D_2 Z}{R^2} e^{-\frac{Z^2}{2R^2}}, \tag{D-13}$$

$$A_3 = D_3 Z e^{-\frac{Z^2}{2}}, \tag{D-14}$$

$$Z = \frac{L}{2\sqrt{m_0}}. \tag{D-15}$$

L is the magnitude of a load level belonging to a certain load bin.

5. **Calculate DEL:**

   (a) **Calculate expected number of cycles**

$$E[n] = T\sqrt{\frac{m_4}{m_2}}. \tag{D-16}$$

   T is the expected (design) lifetime from the structure.

   (b) **Calculate damage D**

$$\Delta L_k = L_{k,2} - L_{k,1}, \tag{D-17}$$

$$D = E[n]\sum_{k=1}^{n_l} p(L_k)\Delta L_k L_k^m. \tag{D-18}$$

(D-19)

   n_l is the number of bins for the load levels.

   (c) **Calculate DEL**

$$\text{DEL} = \left(\frac{D}{n_r}\right)^{\frac{1}{m}}. \tag{D-20}$$

n_r is an on forehand defined number of cycles which is equal for all sea states and m is the slope in the SN-curve (Figure 2-6).

6. **Combine DEL of all sea states:**

$$\mathrm{DEL}_{\mathrm{total}} = \sum_{\text{all } H_s, T_p\text{-combinations}} \mathrm{DEL}. \tag{D-21}$$

Appendix E

DEL derivatives

The final goal is to calculate:

$$\frac{\partial \text{DEL}_{\text{total}}}{\partial \boldsymbol{x}} = \sum_{\text{all } H_s, T_p\text{-combinations}} \frac{\partial \text{DEL}_h}{\partial \boldsymbol{x}} = \sum_{h:\text{all } H_s, T_p\text{-combinations}} \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{x}}, \tag{E-1}$$

$$\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{x}} = \frac{\partial \text{DEL}}{\partial \boldsymbol{x}} = \frac{\partial \text{DEL}_k}{\partial x_l} = \frac{\partial d_k}{\partial x_l}, \tag{E-2}$$

with $\boldsymbol{d} \equiv \text{DEL}$ as the vector with DELs as result of all sea states so that d_k is the k^{th} entry in that vector. $\boldsymbol{x}$ is a vector with all design parameters as be described in Section 2-2-4. In this Appendix, the same symbols are used as in Appendix D.

Derivative Dirliks method

Now the derivative to Dirliks method is given. Because the purpose is just to show the mathematical derivation, no explanation to the equations is given.

$$d_k = \left(\frac{E[n_k] E[L_k]^m}{n_r} \right)^{\frac{1}{m}} = \left(\frac{1}{n_r} \right)^{\frac{1}{m}} E[n_k]^{\frac{1}{m}} E[L_k], \tag{E-3}$$

$$\frac{\partial d_k}{\partial x_l} = \left(\frac{1}{n_r} \right)^{\frac{1}{m}} \left(\frac{1}{m} E[L_k] E[n_k]^{-\frac{m-1}{m}} \frac{\partial E[n_k]}{\partial x_l} + E[n_k]^{\frac{1}{m}} \frac{\partial E[L_k]}{\partial x_l} \right), \tag{E-4}$$

$$\frac{\partial E[n_k]}{\partial x_l} = \frac{T}{2\sqrt{m_2 m_4}} \left(\frac{\partial m_4}{\partial x_l} - \frac{m_4}{m_2} \frac{\partial m_2}{\partial x_l} \right) \tag{E-5}$$

$$\frac{\partial E[L_k]}{\partial x_l} \approx \frac{1}{m} \left[\sum_{k=1}^{n_l} \Delta_{L_k} p(L_k) L_k^m \right]^{-\frac{m-1}{m}} \left(\sum_{k=1}^{n_l} \Delta_{L_k} \frac{\partial p(L_k)}{\partial x_l} L_k^m \right), \tag{E-6}$$

$$\frac{\partial m_n}{\partial x_l} \approx \sum_{j=1}^{n_f} \Delta_{f_j} f_{j,m} \frac{\partial S_{l,k}(f_{j,m})}{\partial x_l}. \tag{E-7}$$

The derivatives to the probability density function is:

$$\frac{\partial p(L_k)}{\partial x_l} = -\frac{1}{4m_0\sqrt{m_0}}\frac{\partial m_0}{\partial x_l}\left[A_1 + A_2 + A_3\right] + \frac{1}{2\sqrt{m_0}}\left[\frac{\partial A_1}{\partial x_l} + \frac{\partial A_2}{\partial x_l} + \frac{\partial A_3}{\partial x_l}\right], \tag{E-8}$$

$$\frac{\partial A_1}{\partial x_m} = \frac{1}{Q}\left(\frac{\partial D_1}{\partial x_m} - \frac{D_1}{Q}\frac{\partial Q}{\partial x_m} + \frac{D_1 Z}{Q}\left(\frac{1}{Q}\frac{\partial Z}{\partial x_m} - \frac{Z}{Q^2}\frac{\partial Q}{\partial x_m}\right)\right)e^{\frac{-Z}{Q}}, \tag{E-9}$$

$$\frac{\partial A_2}{\partial x_m} = \frac{1}{R^2}\left(Z\frac{\partial D_2}{\partial x_m} - 2\frac{D_2 Z}{R}\frac{\partial R}{\partial x_m} + D_2\frac{\partial Z}{\partial x_m} + \frac{D_2 Z^3}{2R^2}\left(\frac{Z}{R^2}\frac{\partial Z}{\partial x_m} + \frac{Z^2}{R^3}\frac{\partial R}{\partial x_m}\right)\right)e^{\frac{-Z^2}{2R^2}}, \tag{E-10}$$

$$\frac{\partial A_3}{\partial x_m} = \left(Z\frac{\partial D_3}{\partial x_m} + D_3\left(1 + \frac{Z^4}{2}\right)\frac{\partial Z}{\partial x_m}\right)e^{-\frac{Z^2}{2}}, \tag{E-11}$$

$$\frac{\partial Z}{\partial x_m} = \frac{L}{2m_0\sqrt{m_0}}\frac{\partial m_0}{\partial x_m}. \tag{E-12}$$

The derivatives to the irregularity factor and the Dirliks constants are:

$$\frac{\partial \alpha}{\partial x_m} = -\frac{m_1}{{m_0}^2}\sqrt{\frac{m_2}{m_4}}\frac{\partial m_0}{\partial x_m} + \frac{1}{m_0}\sqrt{\frac{m_2}{m_4}}\frac{\partial m_1}{\partial x_m} + \frac{1}{2}\frac{m_1}{m_0}\sqrt{\frac{1}{m_2 m_4}}\frac{\partial m_2}{\partial x_m} - \frac{1}{2}\frac{m_1}{m_0 m_4}\sqrt{\frac{m_2}{m_4}}\frac{\partial m_4}{\partial x_m}, \tag{E-13}$$

$$\frac{\partial \gamma}{\partial x_m} = \frac{1}{\sqrt{m_0 m_4}}\frac{\partial m_2}{\partial x_m} - \frac{1}{2}\frac{m_2}{m_0\sqrt{m_0 m_4}}\frac{\partial m_0}{\partial x_m} - \frac{1}{2}\frac{m_2}{m_4\sqrt{m_0 m_4}}\frac{\partial m_4}{\partial x_m}, \tag{E-14}$$

$$\frac{\partial D_1}{\partial x_m} = 2\left(\frac{\partial \alpha}{\partial x_m} - 2\gamma\frac{\partial \gamma}{\partial x_m}\right)\frac{1}{1+\gamma^2} - \left(2\gamma\frac{\partial \gamma}{\partial x_m}\right)\frac{2(\alpha - \gamma^2)}{(1+\gamma^2)^2}, \tag{E-15}$$

$$\frac{\partial D_2}{\partial x_m} = \left(-\frac{\partial \gamma}{\partial x_m} - \frac{\partial D_1}{\partial x_m} + 2D_1\frac{\partial D_1}{\partial x_m}\right)\frac{1}{1-R} + \left(\frac{\partial R}{\partial x_m}\right)\frac{1 - \gamma - D_1 + D_1^2}{(1-R)^2}, \tag{E-16}$$

$$\frac{\partial D_3}{\partial x_m} = -\frac{\partial D_1}{\partial x_m} - \frac{\partial D_2}{\partial x_m}, \tag{E-17}$$

$$\frac{\partial Q}{\partial x_m} = 1.25\left(\frac{\partial \gamma}{\partial x_m} - \frac{\partial D_3}{\partial x_m} - D_2\frac{\partial R}{\partial x_m} - R\frac{\partial D_2}{\partial x_m}\right)\frac{1}{D_1} - \left(\frac{\partial D_1}{\partial x_m}\right)\frac{1.25\left(\gamma - D_3 - D_2 R\right)}{D_1^2}, \tag{E-18}$$

$$\begin{aligned}\frac{\partial R}{\partial x_m} = &\left(\frac{\partial \gamma}{\partial x_m} - \frac{\partial \alpha}{\partial x_m} - 2D_1\frac{\partial D_1}{\partial x_m}\right)\frac{1}{1 - \gamma - D_1 + D_1^2} \\ &+ \left(\frac{\partial \gamma}{\partial x_m} + \frac{\partial D_1}{\partial x_m} - 2D_1\frac{\partial D_1}{\partial x_m}\right)\frac{\gamma - \alpha - D_1^2}{(1 - \gamma - D_1 + D_1^2)^2}.\end{aligned} \tag{E-19}$$

Derivative of the Power spectrum S_k^{ss}

If l indicates the DOF for which the PSD is written and ss indicates a certain combination of H_s and T_p:

$$S_k^{ss}(\omega) = \sum_{l=1}^{n}\left|H_{k,l}^{f\to f}\right|^2 S_l^{ss}(\omega). \tag{E-20}$$

Next step is to calculate $\frac{\partial S}{\partial x_m}$:

$$\frac{\partial S_k^{ss}}{\partial x_m} = \sum_{l=1}^{n}\left|H_{k,l}^{f\to f}\right|^2\frac{\partial S_l^{ss}(\omega)}{\partial x_m} + 2S_l^{ss}(\omega)\left|H_{k,l}^{f\to f}\right|\frac{\partial\left|H_{k,l}^{f\to f}\right|}{\partial x_m}. \tag{E-21}$$

Note that the first term within the sum, $\left|H_{k,l}^{f\to f}\right|^2 \frac{\partial S_l^{ss}(\omega)}{\partial x_m}$ is not equal to zero just for the dimensions x_m in the design space which influences the diameters below SWL. So for all other dimensions:

$$\frac{\partial S_k^{ss}}{\partial x_m} = \sum_{l=1}^{n} 2S_l^{ss}(\omega) \left|H_{k,l}^{f\to f}\right| \frac{\partial \left|H_{k,l}^{f\to f}\right|}{\partial x_m}. \tag{E-22}$$

In this equations, $H_{k,l}$ are the structural transfer functions and S_l the power spectra of node l.

The derivatives of the structural properties to the design variables can be found in literature, for example in [33].

Summary: Equations to come up with derivative to DEL

Derivative to DEL

$$\frac{\partial d_k}{\partial x_l} = \left(\frac{1}{n_r}\right)^{\frac{1}{m}} \left(\frac{1}{m} E[L_k] E[n_k]^{-\frac{m-1}{m}} \frac{\partial E[n_k]}{\partial x_l} + E[n_k]^{\frac{1}{m}} \frac{\partial E[L_k]}{\partial x_l}\right) \tag{E-23}$$

Derivatives to expected load and number of cycles

$$\frac{\partial E[n_k]}{\partial x_l} = \frac{T}{2\sqrt{m_2 m_4}} \left(\frac{\partial m_4}{\partial x_l} - \frac{m_4}{m_2} \frac{\partial m_2}{\partial x_l}\right) \tag{E-24}$$

$$\frac{\partial E[L_k]}{\partial x_l} \approx \frac{1}{m} \left[\sum_{k=1}^{n_{bins,L}} \Delta_{L_k} p(L_k) L_k^m\right]^{-\frac{m-1}{m}} \left(\sum_{k=1}^{n_{bins,L}} \Delta_{L_k} \frac{\partial p(L_k)}{\partial x_l} L_k^m\right) \tag{E-25}$$

Derivative to load probability density function

$$\frac{\partial p(L_k)}{\partial x_l} = -\frac{1}{4 m_0 \sqrt{m_0}} \left[A_1 + A_2 + A_3\right] + \frac{1}{2\sqrt{m_0}} \left[\frac{\partial A_1}{\partial x_l} + \frac{\partial A_2}{\partial x_l} + \frac{\partial A_3}{\partial x_l}\right] \tag{E-26}$$

with $\frac{\partial A_1 a}{\partial x_m}$ as described in **??** till E-19.

Derivative to spectral moment

$$\frac{\partial m_n}{\partial x_l} \approx \sum_{j=1}^{n_{bins,f}} \Delta_{f_j} f_{j,m} \frac{\partial S_{l,k}(f_{j,m})}{\partial x_l} \tag{E-27}$$

Derivative to power spectrum

$$\frac{\partial S_k^{ss}}{\partial x_m} = \sum_{l=1}^{n} \left|H_{k,l}^{f\to f}\right|^2 \frac{\partial S_l^{ss}(\omega)}{\partial x_m} + 2 S_l^{ss}(\omega) \left|H_{k,l}^{f\to f}\right| \frac{\partial \left|H_{k,l}^{f\to f}\right|}{\partial x_m} \tag{E-28}$$

Derivative to transfer function

$$\frac{\partial \left|\boldsymbol{H}^{f\to f}\right|}{\partial x_m} = \left(\frac{\partial \boldsymbol{K}}{\partial x_m} \boldsymbol{\Phi} \left|\boldsymbol{A}\right| + 2\boldsymbol{K} \frac{\partial \boldsymbol{\Phi}}{\partial x_m} \left|\boldsymbol{A}\right| + \boldsymbol{K}\boldsymbol{\Phi} \frac{\partial \left|\boldsymbol{A}\right|}{\partial x_m}\right) \boldsymbol{\Phi}^T \tag{E-29}$$

Derivative to damping factor

$$\frac{\partial \left|\boldsymbol{A}\right|}{\partial x_m} \equiv \frac{\partial \left|A_{ii}\right|}{\partial x_m} = \frac{4\lambda_i}{\mu_i \omega^2} \frac{\lambda_i^2 \frac{\partial \lambda_i}{\partial x_m} + (1+2\zeta_i) \frac{\partial \lambda_i}{\partial x_m}}{\left(\lambda_i^4 + 2(1+2\zeta_i)\lambda_i^2 + 1\right) \sqrt{\lambda_i^4 + 2(1+2\zeta_i)\lambda_i^2 + 1}} \tag{E-30}$$

$$\frac{\partial \lambda_i}{\partial x_m} = \frac{1}{\omega} \frac{\partial \omega_i}{\partial x_m} \tag{E-31}$$

Derivative to eigenfrequency

$$\frac{\partial \omega_i^2}{\partial x_m} = \frac{\boldsymbol{x}_i^T \left(\frac{\partial \boldsymbol{K}}{\partial x_m} - \omega_i^2 \frac{\partial \boldsymbol{M}}{\partial x_m}\right) \boldsymbol{x}_i}{\mu_i} \tag{E-32}$$

Derivative to eigenmodes

$$\tilde{\boldsymbol{x}}_i = \left(\boldsymbol{K} - \omega_i^2 \boldsymbol{M}\right)^+ \left(-\frac{\partial \boldsymbol{K}}{\partial x_m} + \omega_i^2 \frac{\partial \boldsymbol{M}}{\partial x_m} + \frac{\partial \omega_i^2}{\partial p} \boldsymbol{M}\right) \boldsymbol{x}_i \tag{E-33}$$

$$\frac{\partial \boldsymbol{x}_i}{\partial x_m} = \tilde{\boldsymbol{x}}_i - \frac{\boldsymbol{x}_i}{\mu_i} \left(\frac{1}{2} \boldsymbol{x}_i^T \frac{\partial \boldsymbol{M}}{\partial x_m} \boldsymbol{x}_i + \boldsymbol{x}_i^T \boldsymbol{M} \tilde{\boldsymbol{x}}_i\right) \tag{E-34}$$

Bibliography

[1] A. Brownlee and J. R. Woodward. Why we fell out of love with algorithms inspired by nature, June 2015.

[2] N. Castillon. Bird ballet: Swarming video by neels castillon, 2013.

[3] C. Chantharasenawong, P. Jongpradist, and S. Laoharatchapruek. Preliminary design of 1.5-mw modular wind turbine tower. In *The 2nd TSME International Conference on Me-chanical Engineering*, 2011.

[4] S. M. Clarke, J. H. Griebsch, and T. W. Simpson. Analysis of support vector regression for approximation of complex engineering analyses. *Journal of mechanical design*, 127(6):1077–1087, 2005.

[5] T. Dirlik. *Application of computers in fatigue analysis.* PhD thesis, University of Warwick, 1985.

[6] DNV. Buckling strength of shells, 2013.

[7] DNVGL. Support structures for wind turbines. Technical report, DNV GL AS, 2016.

[8] A. Forrester, A. J Keane, and N.W. Bressloff. Design and analysis of" noisy" computer experiments. *AIAA journal*, 44(10):2331–2339, 2006.

[9] A. Forrester, A. Sobester, and A. Keane. *Engineering design via surrogate modelling: a practical guide.* John Wiley & Sons, 2008.

[10] J.H. Friedman. Multivariate adaptive regression splines. *The annuals of statistics*, pages 1–67, 1991.

[11] P. Godfroy. Integrated design methodolohy for a monopile support structure for offshore wind turbines, using numerical optimization. Master's thesis, Delft University of Technology, 2010.

[12] R. Haghi. Integrated design and optimization of an offshore wind turbine monopile support structure. Master's thesis, Delft University of Technology, 2011.

[13] A. Ho and A. Mbistrova. The european offshore wind industry, 2017.

[14] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.

[15] M. J. Kaiser and B. Snyder. *Offshore wind energy cost modeling: installation and decommissioning*, volume 85. Springer Science & Business Media, 2012.

[16] J. Kennedy and R. Eberhart. Particle swarm optimisation. *IEEE*, 1995.

[17] J. Kennedy, J.F. Kennedy, R. C. Eberhart, and Y. Shi. *Swarm intelligence*. Morgan Kaufmann, 2001.

[18] J.P.C. Kleijnen. Kriging metamodeling in simulation: a review. *European Journal of Operational Research*, 192(3):707–716, 2009.

[19] P.N. Koch, T.W. Simpson, J.K. Allen, and F. Mistree. Statistical approximations for multidisciplinary design optimization: the problem of size. *Journal of Aircraft*, 36(1):275–286, 1999.

[20] J.R. Koehler and A.B. Owen. 9 computer experiments. *Handbook of statistics*, 13:261–308, 1996.

[21] D. G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6):119–139, 1951.

[22] M. J. Kühn. *Dynamics and design optimisation of offshore wind energy conversion systems*. DUWIND, Delft University Wind Energy Research Institute, 2001.

[23] Y.C. Liang, H.P. Lee, S.P. Lim, W.Z. Lin, K.H. Lee, and C.G. Wu. Proper orthogonal decomposition and its applicationsŮpart i: Theory. *Journal of Sound and vibration*, 252(3):527–544, 2002.

[24] H. Matlock. Correlations for design of laterally loaded piles in soft clay. *Offshore Technology in Civil EngineeringŠs Hall of Fame Papers from the Early Years*, pages 77–94, 1970.

[25] M.D. McKay, R.J. Beckman, and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[26] M. Muskulus and S. Schafhirt. Design optimization of wind turbine support structures-a review. *Journal of Ocean and Wind Energy*, 1(1):12–22, 2014.

[27] R.H. Myers. *Response Surface Methodology*. Allyn and Bacon, 1971.

[28] H. M. Negm and K. Y. Maalawi. Structural design optimization of wind turbine towers. *Computers & Structures*, 74(6):649–666, 2000.

[29] L. B. Pasamontes, F. G. Torres, D. Zwick, S. Schafhirt, and M. Muskulus. Support structure optimization for offshore wind turbines with a genetic algorithm. In *ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering*, pages V09BT09A033–V09BT09A033. American Society of Mechanical Engineers, 2014.

[30] A. Perelmuter and V. Yurchenko. Parametric optimization of steel shell towers of high-power wind turbines. *Procedia Engineering*, 57:895–905, 2013.

[31] F. Poggio, T.and Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990.

[32] R. Poli. Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications*, 2008:3, 2008.

[33] D. Rixen. Lecture notes engineering dynamics. Lectures given by Paulo Tiso in 2014-2015, September 2008.

[34] J. Sacks, W. J. Welch, T. J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical science*, pages 409–423, 1989.

[35] M. Seidel. Wave induced fatigue loads. *Stahlbau*, 83(8):535–541, 2014.

[36] M. Seidel. Wave induced fatigue loads on monopiles-new approaches for lumping of scatter tables and site specific interpolation of fatigue loads. In *Conference Proceedings IWEC*, 2014.

[37] M. Seidel, S.N. Voormeeren, and J.B. van der Steen. State-of-the-art design processes for offshore wind turbine support structures. *Stahlbau*, 85(9):583–590, 2016.

[38] Y. Sharafi, M.A. Khanesar, and M. Teshnehlab. Cooa: Competitive optimization algorithm. *Swarm and Evolutionary Computation*, April 2016.

[39] T. Simpson, F. Mistree, J. Korte, and T. Mauery. Comparison of response surface and kriging models for multidisciplinary design optimization. In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, page 4755, 1998.

[40] T. W. Simpson, J.D. Poplinski, P. N. Koch, and J. K. Allen. Metamodels for computer-based engineering design: survey and recommendations. *Engineering with computers*, 17(2):129–150, 2001.

[41] K. Sörensen. MetaheuristicsŮthe metaphor exposed. *International Transactions in Operational Research*, 22(1):3–18, 2015.

[42] Spindustrious. Buckled column, 2005.

[43] Dansk standard. Design of steel structures - part 1-6: Strength and stability of shell structures. Technical report, Danish Standards, 2007.

[44] A. Thiry, P. Rigo, L. Buldgen, G. Raboni, and F. Bair. Optimization of monopile offshore wind structures. 2011.

[45] W. Tong. *Wind power generation and wind turbine design.* WIT press, 2010.

[46] A. Torn and A. Zilinskas. *Global Optimization.* 1987.

[47] C. J. Turner and R. H. Crawford. N-dimensional nonuniform rational b-splines for metamodeling. *Journal of Computing and Information Science in Engineering*, 9(3):031002, 2009.

[48] C.J. Turner. *HyPerModels: Hyperdimensional Performance Models.* PhD thesis, 2005.

[49] P.E. Uys, J. Farkas, K. Jarmai, and F. Van Tonder. Optimisation of a steel tower for a wind turbine structure. *Engineering structures*, 29(7):1337–1342, 2007.

[50] J. Van der Tempel. *Design of support structures for Offshore Wind Turbines.* PhD thesis, Delft University of Technology, 2006.

[51] S. Van der Wurff. Internship report: Cost optimization of jacket foundation design. Master's thesis, Delft University of Technology, 2014.

[52] G. N. Vanderplaats. *Multidiscipline Design Optimisation.* Vanderplaats Research & Development, Inc., 2007.

[53] G. G. Wang and S. Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical design*, 129(4):370–380, 2007.

[54] F. Wenneker. Design brief fls assessment (fatigue). Technical report, Siemens Wind Power A/S, 2017.

[55] L. Ziegler. Probalistic estimation of fatigue loads on monopile-based offshore wind turbines - application to sensitivity assessment & clustering optimization for support structure cost reduction -. Master's thesis, Delft University of Technology and Norwegian University of Science and Technology, 2015.