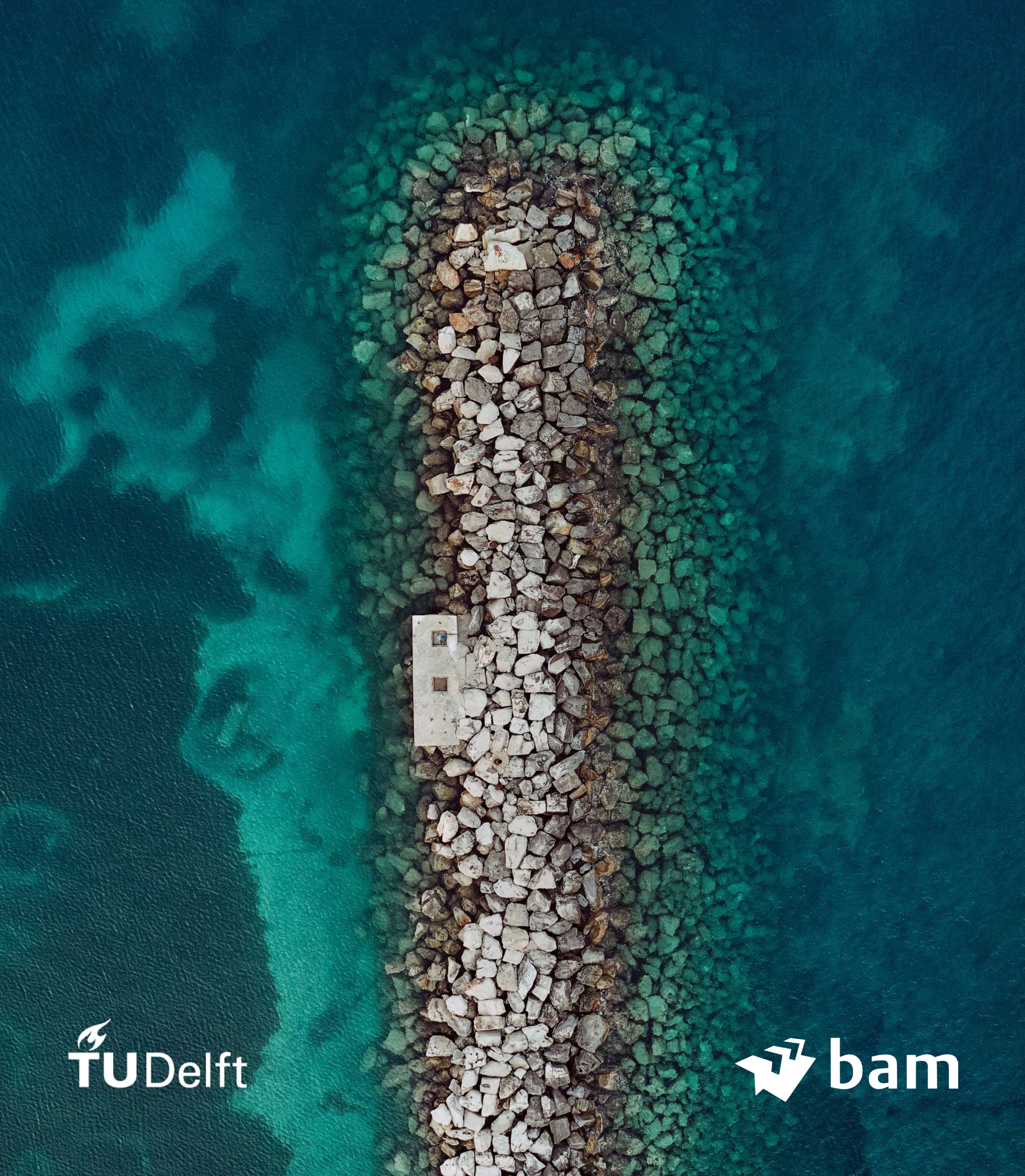
Improving the breakwater design process by using a design automation tool

S. Winkel



TU Delft

bam

Improving the breakwater design process by using a design automation tool

by

S. Winkel

In partial fulfilment of the requirements for the degree of

Master of Science
in Hydraulic Engineering

at the Delft University of Technology,
to be presented publicly on Thursday August 20, 2020 at 15:30h

Student number:	4300165	
Date:	August 13, 2020	
Thesis committee:	Dr.ir. B. Hofland	Delft University of Technology
	Dr.ing. M.Z. Voorendt	Delft University of Technology
	Dr.ir. G.A. van Nederveen	Delft University of Technology
	Ir. M.I. Kelkitli	BAM Infraconsult

An electronic version of this thesis is available at https://repository.tudelft.nl/

Cover photo by Jason Blackeye on Unsplash

“Dream big, start small. But most of all, start.”

— Simon Sinek

Preface

This report presents my research about the application of design automation in the breakwater design process and also concludes my time as a TU Delft student. Over the past years, I have become more and more interested in the field of digital construction, which was one of the main reasons to follow the IDM annotation besides my Hydraulic Engineering master. The challenge for me was therefore to find a topic that combined the technical knowledge from my master and the digital aspect from the IDM annotation. After brainstorming about ideas with friends and colleagues and seeing the potential of design automation tools, I came up with the topic of developing a design automation tool for breakwaters.

My research started with a literature study about design methods and the process of designing, i.e. how designers generate and select concepts. This is not the most typical literature study when following a technical master, but very interesting nevertheless, and a great way to expand my knowledge in other research fields. During this process, I was able to interview three designers, from whom I learned a lot about the practical design approach and how to apply design automation in practice. In this preface, I therefore want to thank them for their time, as without them this research would not have been possible. Based on these interviews I developed a new design automation tool in Python, which is open-source and can be downloaded from GitHub and PyPI! For this, I want to thank BAM, who made it possible to develop the tool with an open-source philosophy.

Conducting this research would not have been possible without the members of my committee. I would like to thank Bas Hofland for giving me the freedom to research my own topic, and keeping me focused on the relation between the design process and the development of the tool, which I sometimes lost. Also, many thanks to Mark Voorendt for all his help, especially in the beginning, with the design literature and design methods. But also, the countless times he read my report and provided me with his feedback. I would also like to thank Sander van Nederveen for his help with systems engineering and the IDM appendix. And last, but certainly not least, I want to thank Ikbal Kelkitli, who was always available at BAM for questions, and the last few months, unfortunately, via Skype. His help was indispensable for structuring the code, developing the tool, and writing this report. For the development of the tool, I also owe my thanks to Robert-Jan Lub, who helped me with structuring the code and the development of the tool in general.

Then I would also like to thank my friends from high school and all the new friends I made during my time at the TU Delft for all their support and the great times we had, from holidays to evenings in The Hague and Delft. But also, the countless walks we had during the past months, sometimes quite long ones, ending up in Kijkduin once... Let's make some new stories! Special shout-out to Mariska and Casper who were willing to read my thesis amidst a heatwave. And last but not least I would like to thank my parents, sister, and family for all their support during my graduation, and, of course, my entire time as a student.

To conclude the preface, this report marks the end of my time as a student of the TU Delft. During my time at the TU Delft I had the opportunity to study at a great university, was able to travel to Chile for a multi-disciplinary project, and to Bulgaria for a consultancy project. But now, the time has come to say goodbye to being a student and start the next phase, working on the digitisation of the construction industry. Enjoy reading this thesis!

S. Winkel
The Hague, August 2020

Table of Contents

List of Symbols

α	angle of the front slope of the breakwater	[°]
α_s	slip angle of a slice in Bishop's method	[°]
Δ	relative density	[–]
γ	volumetric weight of soil	[kN/m^3]
γ^*	influence factor for non-breaking waves for a storm wall on a slope or promenade	[–]
γ_b	influence factor for a berm	[–]
γ_f	influence factor for the permeability and roughness of or on the slope	[–]
γ_v	influence factor for a vertical wall on the slope	[–]
γ_β	influence factor for oblique wave attack	[–]
m	mass of the caisson	[kg]
μ	friction factor between the caisson and the foundation	[–]
ϕ	internal friction angle of the soil	[°]
$\xi_{m-1,0}$	breaker parameter computed with $s_{m-1,0}$	[–]
ξ_m	breaker parameter computed with s_m	[–]
c	cohesion of soil	[kPa]
C_r	reduction factor for a wide rubble mound crest	[–]
d	water depth above the foundation of a vertical breakwater	[m]
D_{n50}	nominal median diameter	[m]
F	factor of safety against slip circle failure, computed with Bishop's method	[–]
F_H	force due to the horizontal pressures	[kN]
F_U	force due to the uplift pressure	[kN]
g	acceleration due to gravity	[m/s^2]
G_c	armour crest width for a rubble mound structure	[m]
h	water depth	[m]
h'	water depth at which the armour of the foundation for a vertical breakwater is placed	[m]
H_D	design wave height in the Hudson formula	[m]
H_s	significant wave height	[m]
h_t	the depth of the toe below still-water level	[m]
$H_{2\%}$	wave height exceeded by 2% of the waves at the toe	[m]
H_{m0}	significant wave height from spectral analysis	[m]
k_D	experimentally determined coefficient, dependent on the material of the armour layer	[–]
L'	wave length at depth h'	[m]
$L_{m-1,0}$	wave length in deep water computed with the spectral wave period	[m]
M_p	moment due to the horizontal pressures	[kNm]

M_U	moment due to the uplift pressure	[kNm]
M_{50a}	median mass of the armour layer	[kg]
M_{50u}	median mass of the underlayer	[kg]
N	number of waves	[−]
N_{od}	damage number	[−]
N_s	stability number	[−]
P	notional permeability	[−]
p_{cap}	bearing capacity of the soil	[kPa]
p_e	pressure exerted by the caisson on the foundation	[kPa]
q	mean overtopping discharge per meter structure width	[m^3/s per m]
R_c	freeboard of the breakwater	[m]
S_d	damage level parameter	[−]
t	horizontal distance to the center of mass of the caisson	[m]
t_e	eccentricity of the force W_e	[m]
W_e	resultant vertical force of the caisson	[kN]

Summary

In the current breakwater design process, not all feasible concepts can be explored due to time constraints. However, designers are also influenced by breakwaters constructed near, or at the same location as their breakwater project. But also conservative assumptions influence the breakwater design process, for instance, regarding the wave height and water depth for which caisson and rubble mound breakwaters are both economical feasible. This results in a suboptimal design, which can result in losing a tender.

To make an optimal design more likely more concepts must be explored during the design process. Davila Delgado and Hofmeyer (2013) showed that a design automation tool can help to generate more concepts, and thus allows designers to explore more concepts during the tender. McKinsey&Company (2017) and Deloitte (2019) also concluded in their reports that digitisation, for instance, design automation, can improve the efficiency of the construction industry, and thus the design process. Design automation is defined as automating (part) of the design process to make it less time demanding (Bernal et al., 2015; Cederfeldt, 2007). Therefore, the objective of this research is to improve the breakwater design process with a design automation tool, which is able to quickly generate and verify rubble mound and caisson breakwaters during the design process.

The first phase of this research described the practical design approach because one must first understand the process to be automated to determine if design automation is needed. Therefore, three experienced breakwater designers from two Dutch companies have been interviewed to describe this practical design approach. In these interviews designers were first asked about how their design approach looked like, a visualisation of this practical approach is presented in Figure 1. Concepts are generated during the conceptual design phase, using an iterative design cycle. This design cycle was not explicitly mentioned by the interviewees. However, it was part of their description of this phase, indicating that experience has an important influence on the design process. Furthermore, the inclusion also confirmed that the iterative behaviour is a fundamental part of all design processes.

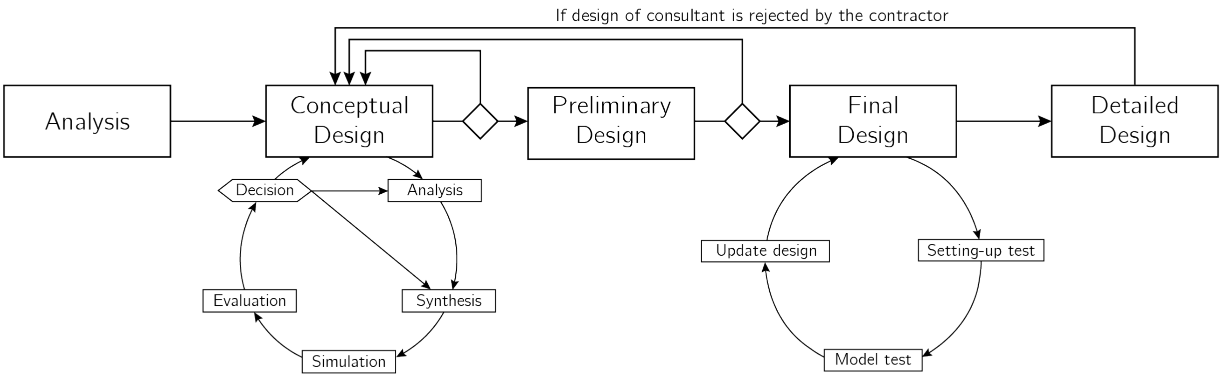


Figure 1: Graphical depiction of the breakwater design approach described by the interviewees

The second part of the interviews was about how concepts are generated and selected during the conceptual design phase. The interviews focused on this phase because several repetitive tasks are performed in this phase, for instance, the computations required to design a concept, as these tasks are best suited for automation. It appeared that in current design practice experience also plays an important role for the generation and selection of concepts. This is because designers reported using their experience or engineering judgement as the preferred method for the selection of the most promising concept.

Furthermore, hydraulic considerations, such as the wave height, are not the only considerations for generating and selecting concepts, also geotechnical conditions, the proximity of quarries, and availability of dry docks are important considerations. Moreover, interviewees frequently drew to explain their design process, further highlighting the importance of experience. To improve the breakwater design process a design automation tool must thus make as little decisions on its own as possible, this to give the designer the largest freedom as possible

and thus incorporating their experience in a combined semi-automatic design approach. An added benefit of the design automation tool is that it can prevent a bias towards certain concepts, as literature showed that if designers are made aware of their bias, other concepts will be explored.

In the second phase of this research, a new design automation tool was developed because existing tools did not fulfil the requirements derived from the interviews. Furthermore, two of the existing tools incorporated a probabilistic design approach, which is unwanted by the interviewed designers, since, according to the designers, this results in a too conservative design. But the interviewed designers also reported that there is a lack of data, which likely results in the designers making conservative estimates to compensate for the missing data. However, this research did not investigate if conservative estimates are made when data is missing, and therefore requires further research.

The developed tool is able to design conventional rubble mound, caisson, and vertically composite breakwaters. The structure of the tool is divided into three levels, where the lowest level, the core, consists of the individual functions and offers the largest freedom. The middle level is an implementation of the core into so-called design classes, which are classes that can be used to design a single breakwater. This level offers less freedom, compared to the lowest level, as several design choices had to be made so that a concept could quickly be generated. However, designing with a design class is significantly quicker than using the individual functions from the core to develop a custom script. The highest level, the design automation, is an implementation of the design classes to automatically design several breakwaters and breakwater types at once.

In the third, and last, phase of this research the tool was verified and validated by interviewing experienced breakwater designers. From a document and code inspection, it was concluded that the new tool fulfilled all requirements from the interviews. During the interviews for the validation the highest level of the tool has been presented, the design automation level. From this level, the graphical user interface was presented, as this requires almost no knowledge of Python. The designers stated that they would be able to use the developed tool in their design approach. One of the main advantages of the tool is that it was able to quickly design concepts, 0.14s per concept on average, and made it possible to explore the influence of parameters on the design and cost. This enabled them to explore more concepts and assess which breakwater type has the highest feasibility, and thus enabled them to generate a better design.

The developed tool is thus able to quickly design concepts and can be used in the current design approach. Since the tool can quickly generate concepts it is easier for designers to explore other breakwater types and assess the feasibility. The tool can thus be used to make designers aware of their biases and preferences towards certain breakwater types. Therefore, it can be concluded that the developed design automation tool can indeed improve the breakwater design process.

To further improve the tool more breakwater types must be added, for instance, a berm breakwater. This enables designers to explore even more breakwater types and will lead to improved results. Finally, it must be remarked that the design automation tool is developed for experienced breakwater designers since the tool does not determine the optimal breakwater. The resulting designs must always be critically assessed in the final design phase since the tool will always generate a design, even if the input is illogical or physically impossible.

1 Introduction

"Much of construction has evolved at a glacial pace", McKinsey&Company (2017, p.1) stated in their report about the construction industry in 2017. The statement was based on the fact that the construction industry has not been able to transform itself over the past two decades to increase productivity. Other industries, for instance, the automotive industry, have however been able to transform and increase productivity. Reasons identified by McKinsey&Company (2017) on why the construction industry is lagging are, amongst others, inefficient design processes and the lack of digitisation.

The aim of this research is to use digitisation to improve the design process, specifically the breakwater design process. Therefore, an example of the current breakwater design practice is given in Section 1.1. Hereafter, in Section 1.2, background information about different types of breakwaters and design automation is presented. Both sections serve as input for the problem statement, which is derived in Section 1.3, and the objective. The latter is part of the research approach, which is elaborated in Section 1.4. Finally, in the last section of this chapter, a reading guide is presented.

1.1 Motivation for this research

The motivation for this research is illustrated with an example from practice. A port authority was planning a port expansion to increase its capacity and allow larger vessels to enter the port. Part of the port expansion project was to construct two new breakwaters, in addition to the two existing caisson breakwaters. The port authority hired a consultant to make a reference design for the tender, the breakwaters in this reference design were rubble mound breakwaters with Xbloc as primary armour.

Multiple contractors participated in the tender and made an offer. However, during the evaluation of the offers, it appeared that all offers were about 30% more expensive than the estimate of the consultant. From the analysis it became apparent that the deviation was caused by an error in the computation of the core volume by the consultant. The port authority responded by hiring a new consultant and issuing a new tender.

BAM International participated in the new tender with a caisson breakwater, because they deemed Xbloc, or any other type of armour units, too risky due to the large wave height ($H_s \approx 8.0\ m$). However, BAM Infraconsult wanted to consider a concept with Xbloc, because they believed armour units were possible. In the end, Xbloc was not considered, because time had run out and the bid was made with a caisson breakwater. With the caisson breakwater BAM International was too expensive and decided to drop out of the tender. In the end, the tender was won with a Cubipod rubble mound breakwater, another type of armour unit, proving that a design with armour units was indeed possible.

This example illustrates that the breakwater design process is indeed inefficient, as McKinsey reported in 2017 for design processes in general. Firstly, an error in the computation of the core volume resulted in issuing a new tender. Secondly, not considering an alternative design during the tender resulted in losing the tender to a design similar to the alternative design.

1.2 Background Information

Expanding on the previous section where two different breakwater types, the rubble mound and caisson, have been introduced, a more elaborate description about the different types of breakwater is given in Subsection 1.2.1. Secondly, background information about, and the application of, design automation is presented in Subsection 1.2.2.

1.2.1 *Breakwater Types*

Section 1.1 already introduced a rubble mound and caisson breakwater, but more types of breakwaters can be distinguished. The different types can be divided into roughly two categories: the rubble mound and monolithic type breakwaters (Van den Bos and Verhagen, 2018). Rubble mound breakwaters consist of large heaps of loose elements with an armour layer of rock or concrete blocks, whereas monolithic breakwaters have a cross-section which acts as one block, for instance, a caisson. Figure 1.1 depicts the representative cross-sections for all breakwater types defined in the Rock Manual (CIRIA, CUR, CETMEF, 2007).

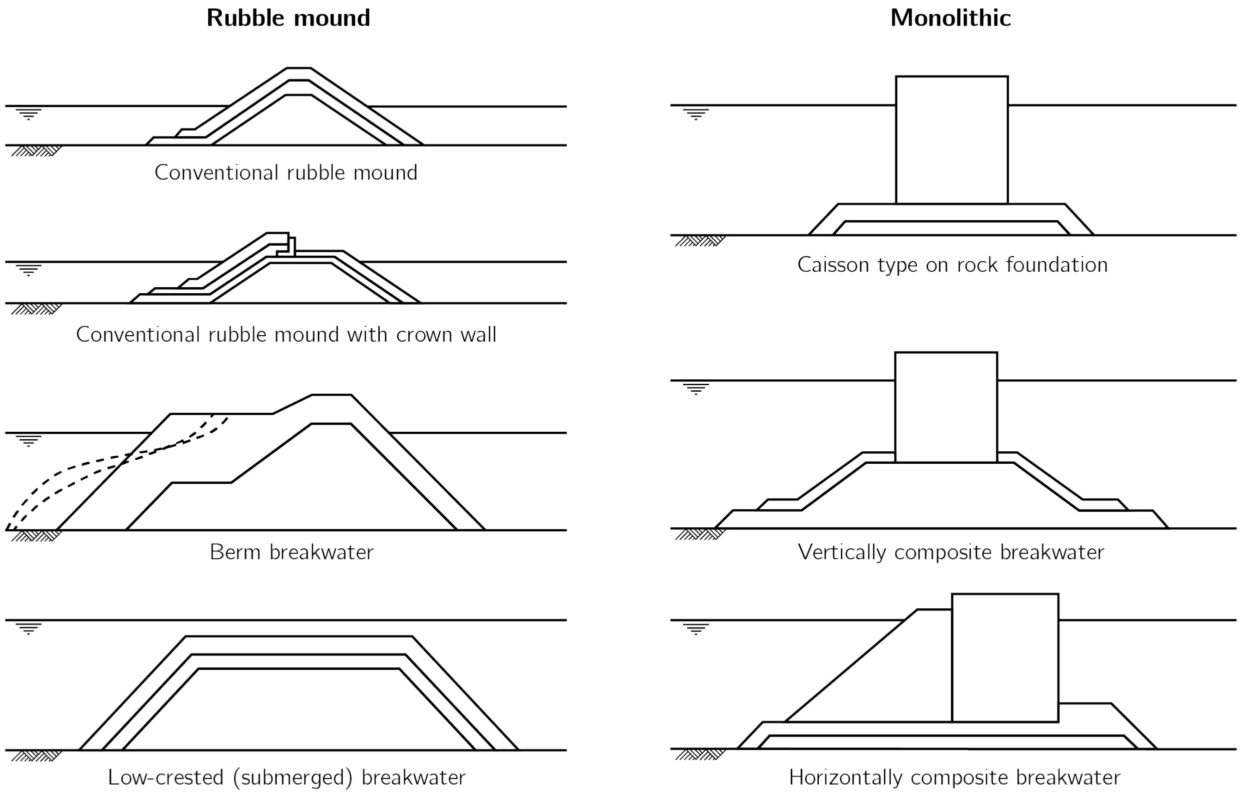


Figure 1.1: Typical cross sections of various types of breakwaters, with the rubble mound types on the left and the monolithic types on the right. Redrawn from CIRIA, CUR, CETMEF (2007, p.781)

Within the rubble mound breakwaters four types are distinguished:

- *Conventional rubble mound*
  This is the most common type of breakwater (Allsop et al., 2009). Depending on the wave height, the armour layer (top layer) is constructed with rock or concrete armour units. The main purpose of this structure is to provide shelter for other structures, for instance, jetties, berths, and beaches.

- *Conventional rubble mound with crown wall*
  Although similar to the conventional rubble mound breakwater, the type with a crown-wall is distinguished as a separate type since it allows for the construction of an access road on the crest. Especially for port operations, the road is an important feature because it allows for easy access to a quay or berth.

- *Berm breakwater*
  This type includes a berm in the structure. Depending on the stability of the armour layer, three types of berm breakwaters are classified:
  - *Non-reshaping statically stable*, where a few stones are allowed to move.
  - *Reshaping statically stable*, where during extreme events the profile is reshaped to a naturally stable profile within which individual stones are stable.
  - *Dynamically stable reshaping*, where during extreme events the profile is reshaped to a naturally stable s-profile in which the individual stones are still moving up and down the slope.
- *Low-crested (submerged) breakwater*
  Low-crested structures are mainly used if the wave height must be reduced but overtopping is allowed, or in case visibility is an important requirement. While the low-crested breakwater has been drawn submerged in Figure 1.1 it can also be (partly) emerged, for instance, during neap tide.

Within the monolithic breakwaters three different types are distinguished:

- *Caisson*
  A caisson filled with sand, or another fill material, is placed on a rock foundation. Because of the vertical wall this type is especially suited for harbours since a ship can easily dock against the caisson.
- *Vertically composite*
  This type is similar to the regular caisson since the only difference is that the rubble mound foundation occupies a larger portion of the depth. Therefore, it is harder to visually distinguish this type of breakwater from caisson breakwater as the difference is below the water. The definition from EurOtop (2018, p.194) is used in this research to distinguish this structure from the caisson breakwater. This definition is that the structure is classified as a vertically composite breakwater if the depth above the foundation divided through the water depth is smaller than 0.6.
- *Horizontally composite*
  The horizontal composite is a hybrid structure that combines aspects from the conventional rubble mound and caisson breakwater. On the seaside a rubble mound protection is made, whereas on the sheltered side a caisson is constructed.

A clear difference between the two categories is the water depth: the monolithic types are depicted in a larger water depth than the rubble mound types. This is because, generally, a rubble mound is more economical in water depths of less than 10 m, whereas the vertical breakwater is more economical in water depths of 15 m and deeper (CIRIA, CUR, CETMEF, 2007, p.811). Given a constant slope, the amount of material needed for a rubble mound increases significantly with increasing water depth due to the triangular cross-section. Contrary to the caisson breakwater, where, due to the rectangular cross-section, increasing water depth does not result in a significant increase in material. This makes the caisson breakwater often the more economical concept in larger water depths. In shallower water the monolithic types are, most often, not economically feasible due to the relatively large upfront investment required to set-up the caisson production. The exact water depth at which the monolithic type becomes more economical depends on the location of the project.

1.2.2 *Design Automation*

According to McKinsey&Company (2017), the productivity growth of the construction industry averaged 1% a year compared to 2.8% for the total world economy. It was estimated that a productivity boost of 50 to 60% is possible by, among others, rethinking design processes, reshaping regulations and using new digital technologies. Research about the implementation of these productivity boosters showed that companies are starting to adapt and transform processes, but implementation remains slow (Koeleman et al., 2019). Therefore, in 2019, Deloitte (2019) still identified rethinking design processes and incorporating digital technologies as potential tools to transform the construction industry.

To assess the use of digital technologies in The Netherlands, Bouwend Nederland (2019) carried out a benchmark on the digitisation of the Dutch construction industry. It was found that digitisation has the highest priority within construction companies. The main reason is the potential of digitisation to decrease costs and increase efficiency. Companies want to achieve this by optimising design and construction processes with the aid of digital tools as, for instance, 3D modelling and BIM (building information models). Again, automation and the design process are mentioned together and are therefore further studied. For a broader analysis of the digital tools used within the construction industry, one is referred to Appendix A.

Automating the design process is especially useful for generating different concepts in a relatively short amount of time, as Davila Delgado and Hofmeyer (2013) showed. In their experimental study, they made a design tool which was able to generate different structural solutions in a relatively short amount of time. It was concluded that the tool helped designers to quickly, and efficiently, consider several solutions. Furthermore, the automated process also helped the designer gain a greater insight into possible solutions, and thus, a better understanding of the design space.

One way to rethink design processes and use digital technologies is thus to use an automation tool within the design process. Terms mentioned to automate design processes are generative design, parametric design, and automated engineering. However, there is no unified definition among authors, and the terms are used interchangeably. These terms are therefore not used in this research. Instead, the broader term design automation is used, since it seems that this term is used more consistently. Design automation is defined as automating (part of the) design process to make the design process less time demanding (Bernal et al., 2015; Cederfeldt, 2007). Tasks best suited for design automation are the repetitive and routine tasks of the design process because these tasks handle explicit information and knowledge Cederfeldt (2007).

An example of a routine task within the breakwater design process is designing several concepts. In current practice this is not feasible since it requires time to develop concepts, as Section 1.1 illustrated. However, with the use of design automation, it becomes possible to develop several concepts in a relatively short amount of time, as Davila Delgado and Hofmeyer (2013) showed. Furthermore, automatically generating concepts has the added value that errors in computations are no longer made, assuming that, the tool is correctly built and verified.

1.3 Problem Statement

The example from practice, Section 1.1, already highlighted that disregarding a type of breakwater in an early design phase can result in a suboptimal design. Furthermore, in this specific example, the rubble mound breakwater type should not have been disregarded based on solely the wave height. This indicates that other factors may influence the generation and selection of concepts, for instance, choosing familiar solutions. In an experimental study, Ford and Gioia (2000) showed that individuals indeed have the tendency to select familiar solutions over unfamiliar ones. The unfamiliar concept for BAM International in the example from practice was a rubble mound breakwater with a wave height of about 8 m. However, considering unfamiliar concepts, and thus, other feasible breakwater types, is important to arrive at an optimal design, where the optimal design is defined as the cheapest design that fulfils the requirements of the client.

To summarise the problem:

- Preferences and conservative assumptions prevent the consideration of feasible concepts
- Not all concepts can be considered during the design process due to time constraints
- Decisions for generating and selecting concepts cannot be based on solely the wave height

This results in the following problem statement:

Ideally, the designer wants to consider all possible variants in the breakwater design process, but because of preferences, conservative assumptions, and time constraints, not all feasible concepts are explored. As a consequence, a suboptimal design is likely.

1.4 Objective, Scope, and Methodology

Generating and designing several concepts in a relatively short amount of time can be done by automating the design process, as was concluded at the end of Subsection 1.2.2. This was translated into the objective which is presented in Subsection 1.4.1. Hereafter, in Subsection 1.4.2 the scope is presented, which is followed by the methodology in the final subsection.

1.4.1 *Objective*

As time constraints are currently one of the main problems, the design automation tool should quickly generate, and design, different breakwater types, in such a way that the tool helps designers to consider several concepts in a short amount of time. Furthermore, the tool should be broadly applicable so that it can be used for all breakwater projects. This results in the following objective:

Improve the breakwater design process with a design automation tool, which is able to quickly generate and verify rubble mound and caisson breakwaters during the design process.

1.4.2 *Scope*

The research on the breakwater design process is performed within the following scope:

- Considered design methods must be related to engineering design
- Only decisions regarding the selection of concepts are considered in detail, these are:
  - Decisions to select which breakwater types are considered
  - How the most promising concept is selected from the generated concepts
- The considered breakwater types from Figure 1.1 are:
  - Conventional rubble mound breakwaters
  - Caisson type on a rock foundation
  - Vertically composite breakwaters
- For the design only one cross-section is considered

1.4.3 *Methodology*

The methodology used in this research to achieve the objective is a combination of a systems engineering and software development method. Although, it must be remarked that the software development method can be interpreted as an adaptation of systems engineering. A graphical depiction of the used methodology is presented in Figure 1.2.

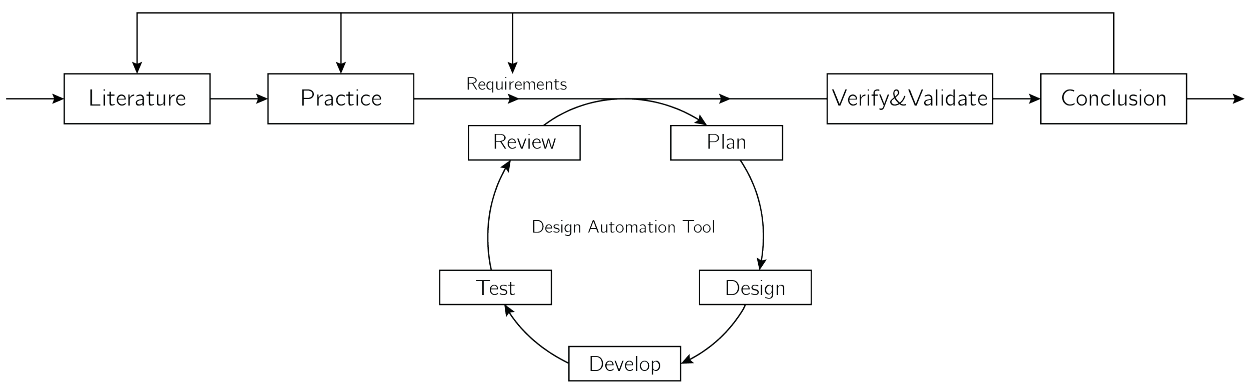


Figure 1.2: Visualisation of the methodology used in this research

From Figure 1.2 it can be seen that the used methodology is an iterative method with several feedback mechanisms. This is because the main goal of software development is to develop software in a process where requirements and solutions continuously change. Furthermore, the main purpose of systems engineering is to develop a system that fulfils the requirements and operational need of the client and continuously prove this. Due to this continuous process of verification and validation, a method must be iterative, so that changes can be incorporated in new versions. However, since this research is conducted within a limited time frame, continuously iterating and improving overall phases of Figure 1.2 is not possible. The steps conducted before, and after, the development phase are therefore depicted as sequential steps. This also means that the feedback from the conclusion is not performed in this research. However, the development of the design automation tool is performed iteratively.

The tasks performed in each phase are:

1. **Study of design literature**
   Before the practical approach is described in the next phase, a literature study about design methods and the process of designing is performed. This to understand the theory behind designing, and how designers make their choices.
2. **Study of current breakwater design practice**
   In this phase, experienced breakwater designers are interviewed about how they design a breakwater because as Cederfeldt (2007, p.4) said: *"To address the question of whether design automation is needed, one must first understand the process to be automated."*. This phase aims to answer the following questions to understand the practical design approach:
   - What approach is used in practice to design breakwaters?
   - How does the approach in practice compare to engineering design methods?
   - On what basis are breakwater concepts generated?
   - How is the most promising concept currently selected?
   - Does design automation have the potential to improve the breakwater design process?
3. **Set-up of the programme of requirements**
   The requirements that the design automation tool must fulfil so that the tool can be used in practice are derived in this intermediate step. The requirements are derived based on interviews with experienced breakwater designers.
4. **Developing a design automation tool**
   In this phase, several iterations are performed to develop a tool that fulfils all requirements. In the first iteration, existing tools are evaluated to investigate if an existing breakwater design automation tool can be used. The tasks in the second, and subsequent, iterations depend on the result of this evaluation, and is one of the following options:
   - An existing tool fulfils all requirements. In this case, the existing design automation tool is not further developed. Instead, the tool proceeds to the verification and validation (V&V) phase.
   - An existing tool does not yet fulfil all requirements but can be expanded. In this case, the existing design automation tool is further developed so that all requirements are fulfilled by the tool.
   - An existing tool does not yet fulfil all requirements, and cannot be expanded. In this case, a new design automation tool is developed.
5. **Verification and validation of the design automation tool**
   In this phase, the verification and validation of the developed design automation tool are performed. This is done by inspecting the documentation and code, testing the tool and, again, interviewing experienced breakwater designers. The last step is performed to validate if the tool fulfils the designers' operational need, which is formulated in the objective of this research, see Subsection 1.4.1.
6. **Conclusions and recommendations**
   This is the last phase and in this phase, the conclusions are drawn, or in other words, does the developed design automation tool meet the objective and can it be used in practice. Furthermore, recommendations are proposed for further research and development.

1.5 Reading Guide

In this introductory chapter, the problem statement and objective were introduced, together with the methodology to achieve the objective. The methodology (Figure 1.2) is divided into three parts, as can be seen in Figure 1.3. The first part, the analysis, consists of two chapters. Chapter 2 presents background information about engineering design methods and the process of generating and selecting concepts. This knowledge is used to understand the practical design approach, which is analysed by interviewing experienced breakwater designers. The results of the interviews are presented in Chapter 3. The main goal of this part is to understand the practical design approach and address the question of Cederfeldt (2007) whether design automation is needed.

Hereafter, in Part II, the design automation tool is developed in Chapter 4 by first listing the requirements that the tool must fulfil. These requirements are used to verify if existing tools can be used in this research. None of the tools fulfilled all requirements, and a new design automation tool was therefore developed in this chapter. Chapter 5 aims to verify and validate the developed tool, or in other words, does the tool fulfil all requirements, and does the developed tool fulfil the operational need of the designer.

Finally, in Part III, the conclusions of the research are presented. This is done by first discussing the developed design automation tool in Chapter 6. After which the conclusion of this research is presented in Chapter 7. Then, in the final chapter of this research, recommendations for further research and development are proposed.

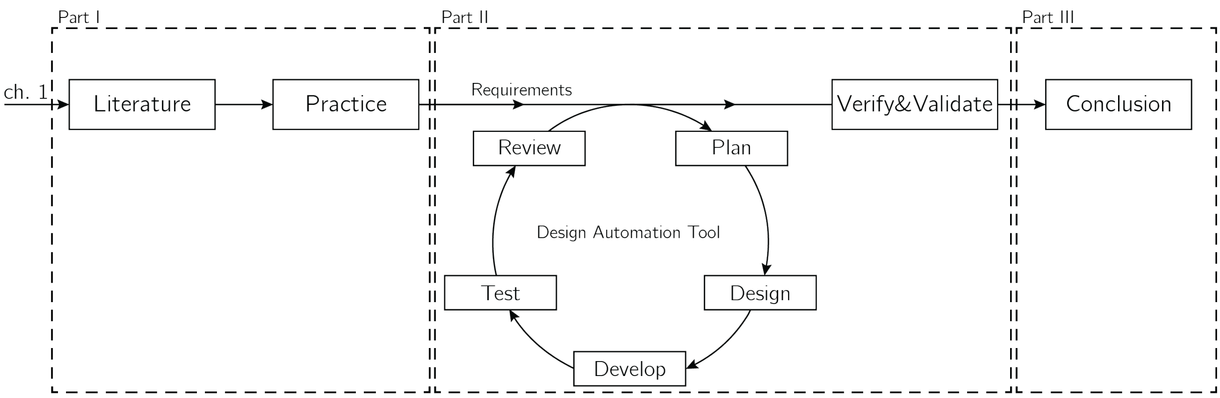
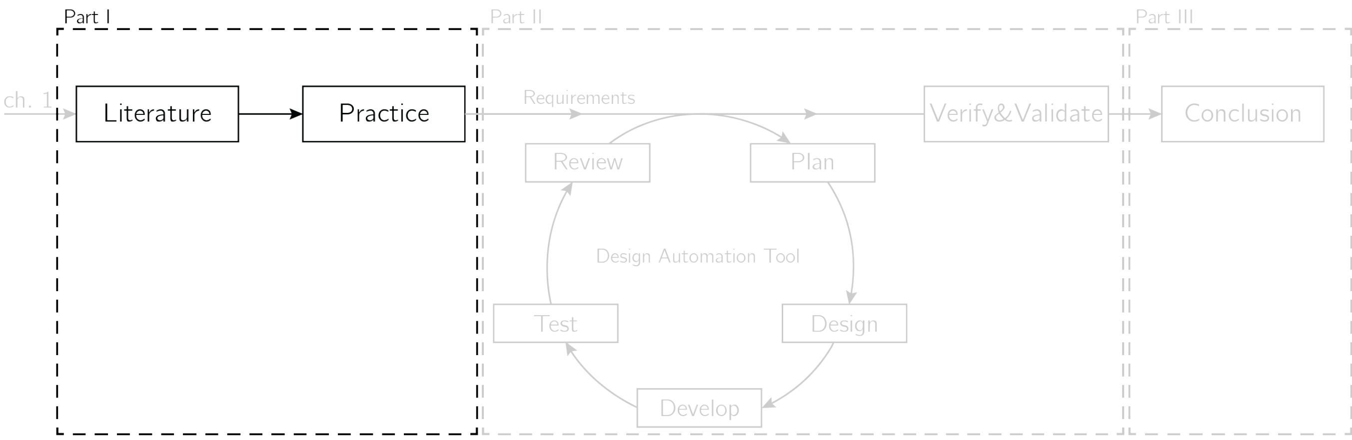


Figure 1.3: Visual representation of the reading guide

Breakwater Design
Theory and Practice



2

Breakwater Design in Theory

To determine if design automation has the potential to improve the breakwater design process, first, the practical design approach must be understood. However, to understand the practical design approach, a literature study about design methods and the generation and selection of concepts is first required. The goal of this chapter is therefore to gain insight in which design methods designers can use, and how designers currently generate and select concepts. First, four design methods, with an application for civil engineering projects, are described. Followed by Section 2.2, in which the generation and selection of concepts is studied. Finally, the chapter is concluded with some concluding remarks.

2.1 Engineering Design Methods

The process of designing a breakwater, or in fact any design process, is performed by following a structured way of working. This structured way of working is called a design method. Voorendt (2017) defined a design method as a general form of systematic processes to accomplish a problem-solving design. Sometimes this is also referred to as design methodology, but according to Eekels (1987) these terms are not interchangeable. He defines design methodology as the study about the methods applied during the design process. So, a design method is the structured way of working to solve a design problem and design methodology is the study about this structured way of working. Many design methods have been described over the years, all serving a different purpose and developed by different sectors. However, to understand the design method used to design a breakwater only methods applied in civil engineering are described.

One must keep in mind however that following a design method is no guarantee for success, or finding the best solution (Van Boeijen and Daalhuizen, 2010) since unlike natural science a design process is trivial. In other words, more solutions exist for the same problem, meaning that the solution is dependent on the designer itself. This is confirmed by Lawson and Dorst (2005) who found that the expertise level of the designer has a large influence on the design, and thus the design process, but more on the considerations of the designer in Section 2.2.

The following design methods are considered in this section:

- The basic engineering design cycle of Roozenburg and Eekels (1995), elaborated in Subsection 2.1.1. Although not specifically developed for engineering problems it is included in this research, because the method can be applied to all design problems due to its generic description. Furthermore, Roozenburg and Eekels describe their method as the most fundamental model of designing, as someone who claims to have solved a design problem has gone through this cycle at least once (Roozenburg and Eekels, 1995).
- The method introduced during the first-year Civil Engineering Bachelor course *Integraal Ontwerpen en Beheer*, *Integrated Design and Maintenance* in English, at the TU Delft. This method coupled the basic engineering design cycle with the design phases defined for a civil engineering project and is elaborated in Subsection 2.1.2.
- The Rock Manual is a technical manual used within Hydraulic Engineering for the construction of a structure with rock, or where rock is the primary material. Although the main focus of the manual is

the technical side of a project, the project phases are briefly mentioned. These project phases are further elaborated in Subsection 2.1.3.

- The systems engineering method, in Subsection 2.1.4, is a method where the main objective is the integration of the design into its environment. The main focus of the method is to define the needs and requirements of the client at an early stage and then fulfilling these by using an interdisciplinary approach. Systems engineering is included because according to the Royal BAM Group nv (2008) systems engineering is needed due to the inclusion of the design, exploitation and maintenance in contracts.

2.1.1 *The Basic Engineering Design Cycle*

The basic engineering design cycle by Roozenburg and Eekels (1995), depicted in Figure 2.1, is a cyclic method consisting of several steps with a decision moment at the end of the cycle. At this moment, a decision must be made to approve the design or start a new cycle with the gained experience from the previous cycle. The new iteration can start by repeating the steps of the previous iteration. Alternatively, it can be performed for a smaller part of the design, so that a more detailed design is made. Roozenburg and Eekels (1995) described their model as the most fundamental model, since someone who claims to have solved a design problem has gone through this cycle at least once. Furthermore, Van Boeijen and Daalhuizen (2010) added to this that it can be used in all phases of the design process, and to all engineering design problems.

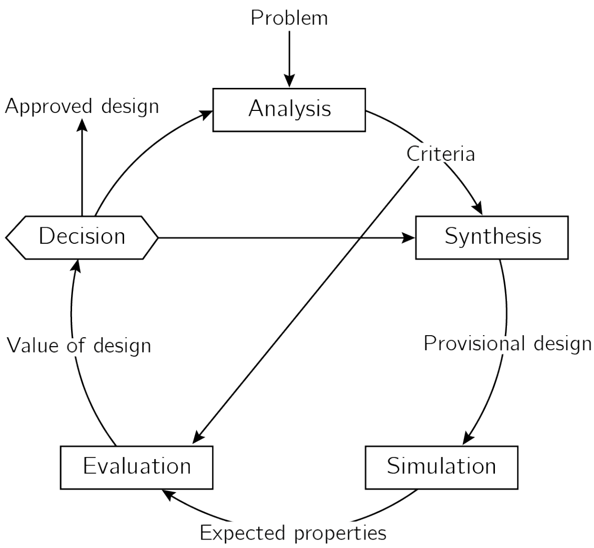


Figure 2.1: Basic engineering design cycle, redrawn from Roozenburg and Eekels (1995)

Analysis
In this step the problem is analysed, the main question which must be answered is: what should the function of the new product be? When it is clear what the function of the new product should be, work can start on making a list of criteria. This list is referred to as, either the *performance specifications* or *programme of requirements*. The performance can in most cases not be derived from the problem. Instead, it is formed during discussions with the client, designer and other stakeholders. Roozenburg and Eekels (1995) described the specifications as the perception the designer, client and stakeholders have of the problem.

Synthesis
In this step, a provisional design is generated. It is the least tangible step because the creativity of the designer is needed to translate the criteria into an idea, and the idea into a provisional design. Dorst and Cross (2001) observed that creative designs are generated by constantly exploring and iterating over the design space. Therefore, several design cycles have to be performed to arrive at the design that is eventually approved.

Simulation
Simulation is framing the behaviour and properties of the provisional design, created in the previous phase, by reasoning or testing. For example, a prototype of a mobile phone can be made and given to users to see how they interact with the new device. In the context of this research, the design process of breakwaters, calculations can be made to determine the dimensions of the breakwater.

Evaluation
The more detailed design of the previous step is evaluated in this step. The evaluation is performed by comparing the expected properties of the design with the desired properties of the specifications, which explains the link with the criteria. The feasibility of the design can be assessed with different tools, an example is a multi-criteria analysis in which the concept or concepts are scored based on a list of predetermined criteria. Another well-known method in civil engineering is a cost-benefit analysis which focuses on the cost and economic value of a concept.

Decision
Since the value of the design is now known a decision can be made on whether to approve the design, and elaborate the design proposal, or start a new iteration with the added knowledge of the previous iteration(s). If the decision is made to start a new iteration, two starting points are possible: return to the analysis and refine the list of requirements, or try to generate a better provisional design by returning to the synthesis.

2.1.2 *Civil Engineering Method*

During the first year Civil Engineering Bachelor course *Integraal Ontwerpen en Beheer* at the TU Delft, *Integrated Design and Maintenance* in English, students learn that the following design phases are used in civil engineering:

- Conceptual design
- Preliminary design
- Final design
- Detailed design

In the reader of *Integrated Design and Maintenance* by Hertogh and Bosch-Rekveldt (2013) these phases are combined with the basic engineering design cycle to arrive at a design method for civil engineering projects. A visualisation of this method is presented in Figure 2.2, and is from now on referred to as the *civil engineering method*. This method is however not derived by Hertogh and Bosch-Rekveldt (2013), instead, it were Soons and De Ridder who combined the terms of the basic engineering design cycle with the design phases (Voorendt, 2020).

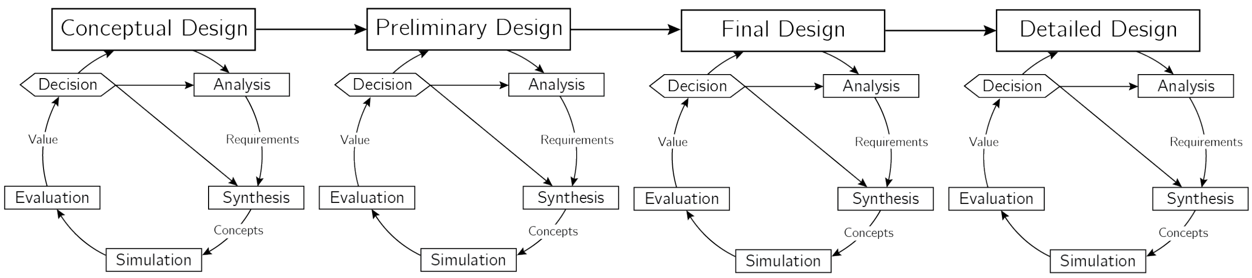


Figure 2.2: Engineering design method taught to civil engineering students, from Hertogh and Bosch-Rekveldt (2013)

To demarcate the different phases the accuracy of the cost estimate is used. However, it must be noted that it remains difficult to distinguish between the different phases in practice since the accuracy of the cost estimate is hard to determine and subject to estimations by experts. The phases are performed sequentially, and before the next design phase is started a decision is made on whether to proceed, or perform an additional iteration. This decision moment, also called a milestone, is similar to the decision in the basic engineering design cycle.

However, the description of the other steps is different. The first one is that in civil engineering multiple concepts are designed in the synthesis step, whereas Roozenburg and Eekels (1995) mention only one. After the synthesis the developed concepts are tested against the requirements. Or in the words of Hertogh and Bosch-Rekveldt (2013), it is verified if the concepts are part of the design space, which is limited by the minimal requirements and boundary conditions. Hereafter, the value of each concept is determined, often if the form of a cost-benefit analysis.

Another difference is that the phases are explicitly mentioned by Hertogh and Bosch-Rekveldt (2013), whereas the basic engineering design cycle does not explicitly incorporate these phases. The reason is that the basic engineering design cycle is mainly developed for product design (Roozenburg and Eekels, 1995). This is an important difference as designing a new product, for instance, a new phone, is designing for mass production whereas civil engineering design is, often, for one-off projects. However, it must be noted that Roozenburg and Eekels (1995) do stress the importance of repeatedly iterating over the design space.

Conceptual Design
The conceptual design is a broad design which gives insight into the dimensions and layout of the structure. Its main purpose is to explore different concepts with a strong emphasis on exploring original ideas by using the basic engineering design cycle. Furthermore, it must become clear how the functions of the design fit into the larger system. For example, how a certain type of breakwater fits into a harbour. At the end of this phase artist impressions showing the integration of the new system into the existing system are presented to the client. Furthermore, the costs are estimated with an accuracy of ± 40%.

Preliminary Design
During this phase, the form, layout and construction method are further refined. Furthermore, alternatives are explored for the different subcomponents of the conceptual design. The solution space is reduced by fixing design variables, this decreases the possibilities and complexity which simplifies the search for alternatives. At the end of this phase the cost estimate is updated, the accuracy should now be between +25% and -15%.

Final Design
During this phase, the preliminary design is further worked out with representatives of the client. Details and connections between subcomponents are designed by using the domain knowledge, and experience, of the designer. A more reliable cost estimate can be made. The accuracy of this estimate is in the order of ±10%. Furthermore, the main components of the design are verified against the requirements of the client.

Detailed Design
In the last design phase, the final design is specified in such a way that it can be built. Therefore, detailed drawings and specifications are made, these are used as input for the bill of quantities. These documents are then used to form the basis for contracts with suppliers and subcontractors. At the end of this phase the total cost of the project is known.

2.1.3 *Rock Manual Method*

Within the field of Hydraulic Engineering a well-known technical manual is the Rock Manual. This manual focuses on hydraulic engineering projects using rock as a construction material, all relevant formulas related to the application of rock are given in this manual. In addition, the manual includes a table describing all project phases, a visualisation of this table is presented in Figure 2.3.

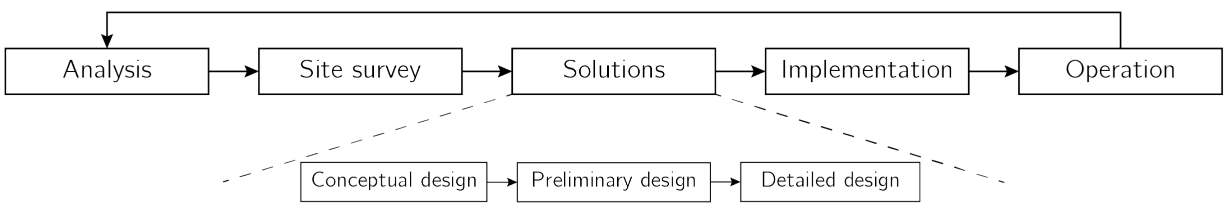


Figure 2.3: Project phases according to the Rock Manual (CIRIA, CUR, CETMEF, 2007, p.6)

While this method describes the entire project cycle, and not solely the design phases, a comparison with the civil engineering method of the previous subsection can still be made. If only the solution phase of the Rock Manual method is considered, it can be seen that the conceptual, preliminary and detailed design phases are present in both methods. Different from the previous two methods is the iterative behaviour, which is not explicitly included in the method. The phases are mentioned as sequential phases without feedback mechanisms or decision moments.

Analysis
The main goal in this phase is to understand the problem. From the problem analysis, the functional and performance criteria can be developed together with the constraints set by the client and local environment.

Site survey
In this phase all necessary site information must be gathered. This means that several practical questions related to the upcoming design activities are answered, for example:

- What kind of materials are available?
- What do the hydraulic conditions (waves, current, water level) look like?
- What does the bathymetry look like?
- What are the soil conditions?

The answer to these questions serve as the input for the solution phase and are used as input for the design or as constraints. The availability of material is an example of such a constraint. If, for instance, rock is not abundantly available a rubble mound breakwater with rock might not be feasible.

Solutions
When the site survey has been completed, the task of generating solutions is started. These are the design activities and consist of three phases, as can be seen in Figure 2.3.

- Conceptual design, in which types and layouts of the structure are developed. In this sub-phase, a project feasibility review is executed to assess the viability of the project.
- Preliminary design, in this sub-phase the developed alternatives of the conceptual design stage are developed, and the required analytical and modelling studies are performed. Furthermore, a cost estimate is made for each of the alternatives. Using this knowledge, the alternatives are compared, and a preferred alternative is selected.
- Detailed design, this is the final phase of the design. During this phase, the possible failure mechanisms are reviewed. Furthermore, the armourstone gradings, underlayers and filters are designed. The dimensions of the toe and crest of the structure are designed as well.

Implementation
When the detailed design is approved, work on the implementation can start. First, the armourstone production and control is started, whereafter the logistics and construction method are further elaborated. Finally, the construction starts and the detailed design is realised.

Operation
The final phase of a project, according to the Rock Manual, is the operational phase. In this phase the structure is monitored and if necessary maintenance is performed to keep the structure operational. At the end of the lifetime, the structure is decommissioned and the cycle starts again from the first phase.

2.1.4 *Systems Engineering*

The International Council on Systems Engineering (2019) defines Systems Engineering as: *"a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods."* The method includes the entire life cycle of the project from the analysis to the demolition of the system, similar to the Rock Manual method described in the previous subsection. One of the most commonly used system development models is called the V-Model. A visualisation of this model is presented in Figure 2.4.

The main goal of the method is to demonstrate to the client that the designed system is optimally developed and fulfils the needs of the client. This is done by using the process of verification and validation. Verification is checking if the system is correctly built, i.e. according to the specification. Whereas validation is checking if the realised system is able to do what the client wants, or asking the question: did we build the correct structure?

Because of this goal, the V-model is set up as an iterative method with many feedback mechanisms, which are depicted as arrows in Figure 2.4. By having all these feedback mechanisms, the method incorporates the possibility for corrective measures, without knowing in advance if they are necessary. The workflow through the method is from top left to top right while proceeding to the lower levels in between. On the left side, all design-related phases are depicted, while the right side depicts the implementation of the new system into the existing system. Or in terms of the breakwater design process, the construction phase of the project.

The system design process is set up in such a way that the system is developed from the highest level, i.e. System Level in Figure 2.4 towards the lowest level, the Part Level. These terms are somewhat abstractly defined, but consider the expansion of a harbour, similar to the example from Section 1.1. In this example, the harbour can be interpreted as the harbour system, the system level in Figure 2.4. To protect the ships against the sea in the expanded harbour a new breakwater was constructed, the subsystem level. The meaning of the lower levels depends on the type of breakwater. In this case, a rubble mound breakwater with armour units was chosen, so the armour units can be interpreted as the part level. However, it must be remarked that this interpretation depends on the project and designer(s), as the breakwater can also be seen as the system level.

Furthermore, the method constantly checks if the lower level can be integrated into the higher levels. In other words: besides a top-down approach there is also a bottom-up approach to validate and verify the solutions developed at the lower levels. If a solution does not satisfy the verification and validation criteria, a new iteration is performed, and a new solution is developed.

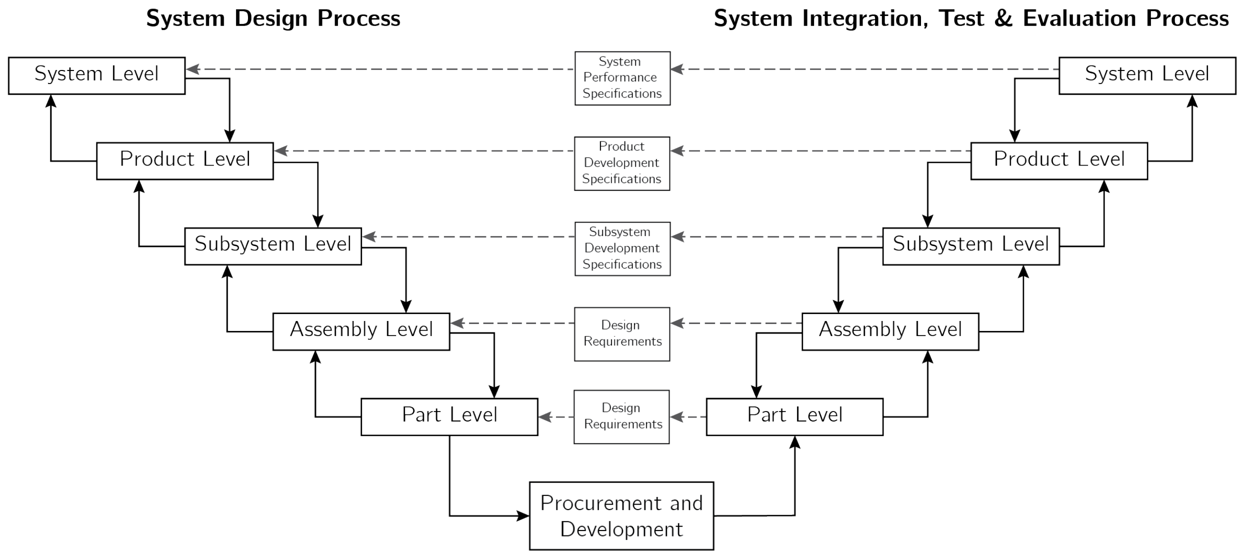


Figure 2.4: V-Model for system development, redrawn from Wason (2015)

2.1.5 *Similarities and differences between the methods*

Comparing the described methods, it appears that all methods are quite similar. They all describe a process where first the requirements of the client are analysed followed by developing several concepts. The most promising concept is then selected which is further developed and refined in subsequent phases.

What differs is the description of the iterative behaviour. Note that the term description is used here since the fact that the iterative behaviour is not explicitly mentioned does not mean that iterations are not performed. On the contrary, it is likely that the designer implicitly still performs these iterations and the steps mentioned in an iteration of the basic engineering design cycle. The reason behind this is that the steps of the basic engineering design cycle are intrinsically linked to the way humans gain knowledge, which follows the steps of observing, suspecting, expecting, testing and evaluating (De Groot, 1994). However, the interviews are needed to confirm this hypothesis. In Table 2.1 the steps from the basic engineering design cycle are presented with the steps and description from De Groot (1994).

From Table 2.1 it can be seen that although the specific terms are different, the described tasks are comparable. Both mention a process where the problem is first analysed, followed by phases where the problem is solved ending with an evaluation of the solution. Hence, Roozenburg and Eekels describing their method as a fundamental method. However, it must be noted that there are differences between the methods. Where the synthesis step in the basic engineering design cycle is aimed at the total picture, the method of De Groot is aimed at an aspect since the goal is to understand a small part of a system.

Table 2.1: Steps from the basic engineering design cycle compared to the steps and description from De Groot (1994)

Roozenburg and Eekels (1995)	**De Groot (1994)**	**Description of De Groot (1994)**
Analysis	Observe	Observing the current situation
Synthesis	Expect	Formulating an assumption based on the observations
	Suspect	Formulating a hypothesis that can be tested
Simulation	Test	Testing the hypothesis
Evaluation	Evaluate	Assessing the results of the test
Decision		

Confirming if the developed solution fulfils the requirements set by the client is also mentioned by all methods, despite using different terms. The systems engineering method incorporates this in the greatest detail with the verification and validation process, but also the decision moment in the basic engineering design cycle and civil engineering method can be interpreted as verifying if the requirements of the client are fulfilled.

So, while the terms of the systems engineering method might be more abstractly defined compared to the other methods, they are similar. Furthermore, when analysing the description of the phases from the other methods it can be seen that they all describe a process where a design is made from the large components to the smaller components. In other words, and terms of the systems engineering method, from system, to subsystem, to the part level. This description is especially clear in the civil engineered and rock manual method. These methods can thus be interpreted as an adaptation of systems engineering. Therefore, it can be concluded that while the described method looks different, and included different terms, the methods are in fact comparable.

2.2 Selection and Generation of Concepts

The most important tasks of the design process are generating and selecting concepts, and are present in each of the described methods. To draw an analogy with the basic engineering design cycle, Subsection 2.1.1, the synthesis step can be interpreted as generating concepts, and the selection of the most promising concept as the evaluation/decision step. This section, therefore, elaborates about how concepts are generated and selected. Lawson and Dorst (2005) already stated that the influence of the designer on these tasks is large. Hence, the influence of the designer on the design process is studied in the following two subsections, first about generating concepts, and secondly about how the most promising concept is selected. Finally, in Subsection 2.2.3, requirements specific to breakwater design are mentioned.

2.2.1 *Generating Concepts*

Dorst (2011) defined designing as the process of solving the following equation:

$$\underset{\text{(thing)}}{\textbf{WHAT}} \quad + \quad \underset{\text{(working principle)}}{\textbf{HOW}} \quad = \quad \underset{\text{(aspired)}}{\textbf{VALUE}}$$

In this equation the *value* a designer wants to improve or add to an existing system is, for instance, to increase the accessibility of a harbour. The accessibility of the harbour can be improved by lowering the waves in the harbour. Hence, *how* to achieve the *value* is already known. However, the *what* remains unknown. Dorst (2011) called this form of closed problem-solving Abduction-1 and is often what designers and engineers do. In this research the assumption has been made that the solution space is limited to constructing a breakwater, other structures or solutions to lower the waves in the harbour are therefore not considered.

Four methods for solving design problems have already been described in Section 2.1. Almost all methods prescribed some kind of cyclic behaviour to find the solution. However, these methods are not the answer to finding a solution, instead, they help the designer to structure their workflow. What yields an optimal design is the constant iteration over the analysis, synthesis and evaluation steps (Dorst and Cross, 2001). This is confirmed by an experimental study where engineering teams indeed generated ideas with a better value if ideas are continuously generated (Toh and Miller, 2015).

The danger in the process of continuously generating ideas is the diversity of the ideas. In an experimental study by Nijstad et al. (2002) it was found that if participants were exposed to less diverse ideas, they also generated more similar ideas. However, the opposite is true as well, because Nijstad et al. (2002) also showed that participants exposed to more diverse ideas tend to generate more diverse ideas.

Another factor influencing the generation of ideas is the experience of the designer. The experience level of the designer can positively influence the generation of ideas, but it can also negatively affect the generation. The positive is that an experienced designer responds to the problem intuitively and immediately performs the necessary design steps (Lawson and Dorst, 2005). But, if the designer has dealt with a similar situation before, the experience can influence the way the problem is analysed, as the solution of the previous situation comes to mind straight away (Dorst, 2011). This framing could result in selecting a familiar solution, which might not be the optimal solution for the current problem.

Ford and Gioia (2000) showed in an experimental study that people indeed often choose conventional and familiar solutions instead of creative ones. According to Rietzschel et al. (2010), this is due to an inadvertent bias against creative ideas. The main problem of selecting creative, and innovative, solutions is the inability of experts to accurately predict the effectiveness of a solution. However, if designers are made aware of their bias and given explicit instructions to consider creative and original ideas, while also regarding feasibility, designers indeed select more creative and original ideas that would otherwise have been dismissed (Toh and Miller, 2015).

From the literature, it can be concluded that if concepts are developed with an approach that stimulates the consideration of several diverse concepts with an open mind, a better final design is made. Therefore, it is beneficial for the design process to use a method that can help the designer with generating ideas. An example of such a method is a morphological chart, see Figure 2.5, which is an analytic and systematic method to generate different concepts (Van Boeijen and Daalhuizen, 2010). In this method, several solutions are developed for subfunctions. New concepts are then generated by combining the subfunctions.

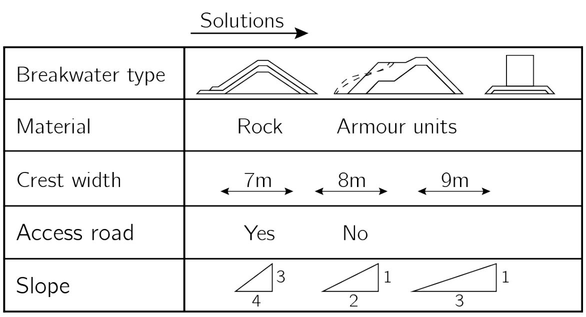


Figure 2.5: Example of a morphological chart for the generation of breakwater concepts. Note that the chart is not complete.

On the Afsluitdijk project, the morphological chart was used as a method to generate several concepts of the water level management system. The method enabled the design team to generate, and explore, a wide range of possible solutions for each subfunction because ideas could be generated on a lower level. Concepts were then generated by combining the possible solutions of the subfunctions. Eventually, the concepts were scored by using a multi-criteria analysis to identify the most promising concept. The design team reported that the method helped to create and consider several concepts in greater detail, and enabled them to make a better multi-criteria analysis (Labouchere, 2018).

2.2.2 *Selection of the Most Promising Concept*

Each of the described methods includes some sort of evaluation and decision moment, in these moments the value of the developed concepts is assessed, and the most promising concept is selected. As for the design process, and generation of ideas, several methods have been developed to help the designer, clients and stakeholders select the most promising concept. This subsection, therefore, presents three methods to support the selection process. These methods are: the multi-criteria analysis, value-cost diagram and a visual method.

Multi-criteria analysis
In a multi-criteria analysis, all developed concepts are scored based on a set of predefined requirements. These requirements are set by the client and stakeholders. The requirements included in the multi-criteria are often associated with a weight factor, this weight factor is often a point of discussion since it influences how important certain requirements are. To overcome the subjectivity associated with the weight factor Hertogh and Bosch-Rekveldt (2013) suggest to determine the weight factors by assessing the importance of each requirement against all other requirements, see Example 2.1 for an example of this process.

However, the weight of each requirement, and how a concept scores on the considered concept is still subject to expert judgement. Therefore, people responsible for scoring the concepts must be aware of a bias towards familiar solutions, as was also the case when generating concepts. Because of this Hertogh and Bosch-Rekveldt (2013) already noted that the results from a multi-criteria analysis can easily be manipulated.

Example 2.1

Consider the following four requirements: safety, minimise malfunctions, maintainability, ease of operation. To determine the weight factor of each requirement, all requirements are specified in a matrix, see Table 2.2. For each requirement in the leftmost column, the question must be raised if it is more important (1) or less important (0) than the requirement on the top row. In case that it is not possible to decide which requirement is more important, both are scored with a 1. The weight factor of each requirement can now be determined by the summation of the values on a row.

Table 2.2: Matrix to determine the weight factor for each requirement

		a	b	c	d	Total
Safety	**(a)**	x	1	1	1	3
Minimise malfunctions	**(b)**	0	x	1	0	1
Maintainability	**(c)**	0	1	x	0	1
Ease of operation	**(d)**	0	1	1	x	2

Value-Cost Diagram
The value cost diagram is a figure where the value is plotted against the cost of each concept. The value of a concept can, for instance, be determined with a multi-criteria analysis, excluding the cost of the concept as a requirement if it was included. Therefore, this method can be seen as a visualisation of a multi-criteria analysis where the cost was included as a requirement. In Figure 2.6, an example of a value cost diagram can be seen. In this figure four quadrants are presented:

- I, high value with a low cost, this is the quadrant where the optimal concepts are located.
- II, high value with a high cost, although these concepts have a high value, they are relatively expensive. Therefore, it should be tried to lower the cost, without lowering the value by too much.
- III, low value with a high cost, these concepts should not be developed any further.

- IV, low value with low cost, investigate if the value of the concept could be increased without increasing the cost.

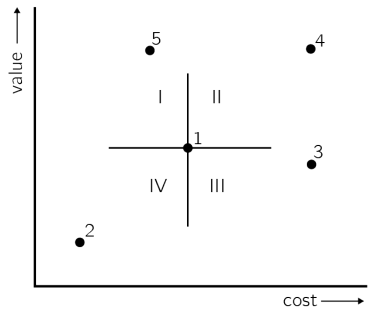


Figure 2.6: Example of a value-cost diagram

Visual Method

Another method to assess the concepts is a visual way, an example of such a method is the Design Explorer. The design explorer is an open-source tool to explore the design space (CORE studio, 2020). In this tool, users can import their design space from, for instance, Grasshopper or Dynamo. A 2D parallel coordinates plot is then generated, see Figure 2.7. In such a plot each concept is depicted by one line, with the parameters plotted as several y-axes. For each of the parameters a filter can be applied. An example of filtering with this method is given in Example 2.2.

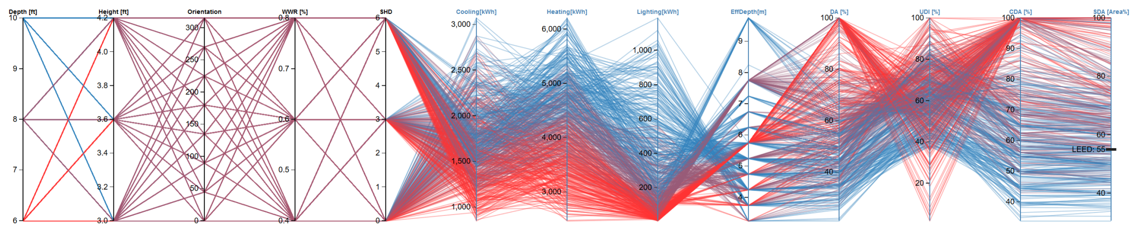


Figure 2.7: Example of the Design Explorer

Example 2.2

Consider the set of concepts from Figure 2.7. Certain concepts can be excluded from the design space by setting a filter, see Figure 2.8. In this case, only concepts having a cooling capacity between 2400 and 2600 kWh are included, concepts not fulfilling this requirement are excluded.

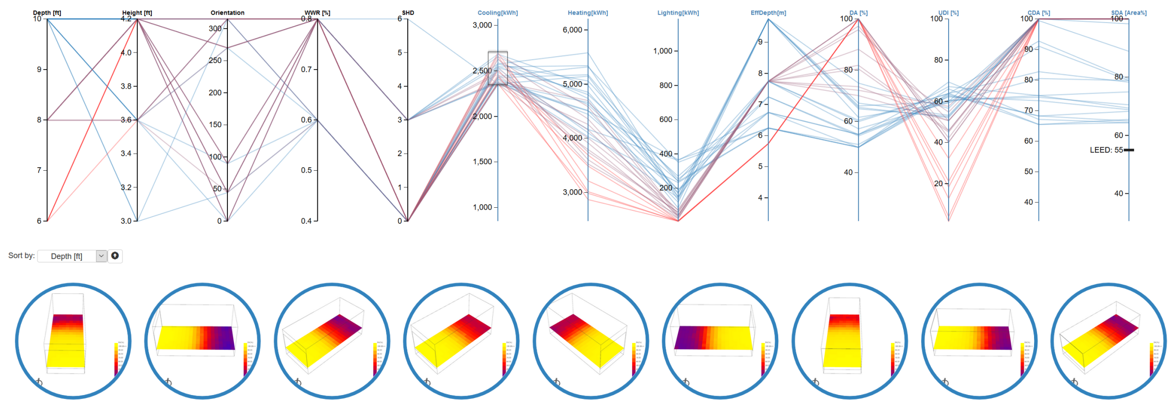


Figure 2.8: Example of the Design Explorer

2.2.3 *Requirements Specific for Breakwaters*

The freedom a breakwater designer has is more limited than the design problems described in the experimental studies described in Subsection 2.2.1. Dorst and Cross (2001), for instance, studied how designers generated concepts for a new litter disposal system of a train. The creativity and freedom needed to generate concepts for this problem are larger than for the design of a new breakwater since the amount of significantly different breakwater concepts is more limited and possible breakwater concepts are better defined (see Figure 1.1).

Because breakwater concepts are better defined, PIANC decided to set up Working Group 47 with the goal to provide guidance to designers on the choice of breakwater types based on functional and environmental requirements and optimal safety levels. However, simple rules for the choice of a breakwater are hard to formulate based on solely functional and environmental requirements since other aspects play an important role (MarCom WG 47, 2016). In Table 2.3 the most important functional requirements according to MarCom WG 47 (2016) are presented. Each breakwater type is scored on their performance on the requirement. Note that the score is relative and a + only indicates that the type is favourable in case only that specific requirement is considered. The definitions of the breakwater types from PIANC are comparable to the ones given in Section 1.2.1, with the difference that the caisson and vertically composite are both defined as a vertical breakwater in the PIANC report.

The requirement about the resistance to waves and currents, on the first row, has not been filled because it depends on the size of the rock, amour unit or stability of the caisson and not the type of breakwater. Nevertheless, since the resistance from waves and currents is an important functional requirement it is included. In Table 2.4 a similar table is presented for the environmental requirements.

Table 2.3: Most important functional requirements and performance of breakwater types according to MarCom WG 47 (2016)

Requirement	**Rubble mound**			**Composite**	
	Conventional	**Crown-wall**	**Berm**	**Vertical**	**Horizontal**
Resistance to damage from waves and currents					
Overtopping	+	+	+	-	+
Access roads on the breakwater crest	-	+	-	+	+
Provisions for berths	-	-	-	+	+
Limit wave reflection	+	+	+	-	+
Increase sediment bearing capacity to reduce sedimentation of the entrance	-	-	-	+	-
Limit wave transmission and cross sediment transport	+/- [1]	+/- [1]	+/- [1]	+	+
Water quality behind the breakwater	-/+ [2]	-/+ [2]	-/+ [2]	-	-
Limit scour of the seabed	+	+	+	-	+

+ indicates a better relative performance on the requirement, whereas - indicates a worse relative performance. Relative in this case signifies, compared to each other.
[1] *Depends on the permeability of the core, + if impermeable and - if permeable core*
[2] *Depends on the permeability of the core, - if impermeable and + if permeable core*

Table 2.4: Most important environmental requirements and performance of breakwater types according to MarCom WG 47 (2016)

Requirement	**Rubble mound**			**Composite**	
	Conventional	**Crown-wall**	**Berm**	**Vertical**	**Horizontal**
Water depth for economic feasibility	smaller	smaller	smaller	larger	larger
Max allowed wave height (better = larger construction window)	+/0 [1]	+	+	-	-
On site construction time (lower = better)	-	-	-	+	0
Allowance for weaker subsoils	+	+	+	-	-
Earthquake resistance	-	-	-	+	0
Geomorphology at the Site and Surroundings	+	+	+	-	-
Environmental (habitat for marine live)	+	+	+	-	0

[1] *Rock or cubes can be placed with higher waves (+), whereas armour units require lower wave heights (0)*

2.3 Concluding Remarks

In Subsection 2.1.5 it was already concluded that the described methods are quite similar, except for the terms used to describe the phases. Furthermore, it was concluded that all methods are an adaptation of systems engineering because designs are made by reasoning from the main functions to subfunctions. This decomposition of the design can also be seen in the morphological chart, Figure 2.5, since here, concepts are generated by developing several variants for each of the subfunctions. Moreover, the hypothesis was introduced that while iterations are not explicitly included in some methods, they are performed because of the fundamental nature. However, interviews with experienced designers are needed to check if this is indeed the case.

The early design phases are the most influential phases of the design process because during these phases the design is generated that is further refined in subsequent design phases. Since the creativity of the designer is required to generate a design, the conclusion can be drawn that the influence of the designer is the largest in the early design phases, as Rustell (2016) also stated. Furthermore, framing of a design problem can result in a suboptimal design, if the problem is incorrectly framed, as discussed in Subsection 2.2.1. This framing behaviour was also observed in the example from practice, described in Section 1.1, where the rubble mound breakwater was disregarded based on the large waves. In this case framing indeed resulted in a suboptimal design as the tender was lost to a consortium with a rubble mound breakwater, highlighting the large influence of early design phases.

Therefore, the main focus of the interviews, that will be conducted in Chapter 3, are these early design phases. And the decisions made in these early design phases, especially those regarding the generation and selection of concepts.

3

Breakwater Design in Practice

The main goal of this chapter is to understand the design approach used by experienced breakwater designers to determine if design automation has the potential to improve the breakwater design process. Therefore, interviews with experienced breakwater designers were conducted. However, before the interviews were conducted the method and structure of the interview that best fits the goal of this chapter has been determined. Since qualitative research, which interviews are, is not often used in engineering, an introduction into qualitative research with interviews is given in Section 3.1.

When the structure of the interview was determined, three interviews were conducted with experienced breakwater designers, who all have over 20 years of experience, see Table 3.1. The results of these interviews are presented in Section 3.2 and 3.3, where Section 3.2 mainly focused on the design process used in practice and Section 3.3 mainly focused on the considerations of designers when generating concepts, and selecting the most promising concept.

Hereafter, in Section 3.4, the breakwater database was used to investigate statements made during the interviews. The breakwater database is useful for this purpose since it consists of data about breakwaters constructed around the world, for instance, water depth and wave height. This data can therefore be used to gain a better understanding of current practice, as the data is a direct result of current design practice. Finally, with the aid of the interviews and main findings from the database, the main questions can be answered in Section 3.5. This question is, does design automation have the potential to improve the breakwater design process?

3.1 Structure of the Interview

To study the design approach used in practice a research method is needed which can capture the diversity of the design process. A quantitative method, mostly used in engineering, is not applicable because this method is used to represent numerical data (Daly et al., 2013). Therefore, a qualitative research method is used to describe the design method as used in practice.

Qualitative research is a scientific method to collect non-numerical data (Babbie, 2011, p.29). An example of a qualitative research method is an interview, which according to Jacob and Furgerson (2012) is the primary method to study human experience and actions. Since the process of designing is a human activity in which experience plays an important role (Lawson and Dorst, 2005), interviewing experienced designers is the best way to describe the design approach used in practice.

However, before conducting an interview, and even before formulating the questions, three considerations must be made: the type of interview, the structure of the questions and the type of question. These considerations are the topic of the next three subsections, after which the used interview structure and interviewed designers are presented in Subsection 3.1.4 and 3.1.5 respectively.

3.1.1 *Type of Interview*

Different methods exist for conducting interviews, each having its advantages and disadvantages. Babbie (2011) described the following three methods:

- Self-Administered Questionnaires, where the interviewees complete the survey on their own. A self-administered questionnaire can be distributed online or through the post. The advantage of this method is that it is cheaper than the other methods, and the possibility of anonymity which can encourage interviewees to give honest answers. This latter advantage is especially useful when dealing with sensitive topics. However, the disadvantage is that it is not possible to ask follow-up questions based on the answer of the interviewee.
- Interview Surveys, in which an interviewer asks the questions to the interviewee. By letting a person ask the questions the chance of misinterpreting a question is reduced and flexibility is increased because the interviewer can actively steer the interview. However, the interviewee may give answers which the interviewer wants to hear, as there is no anonymity.
- Telephone Survey in which an interviewer asks the questions over the telephone. The advantage is that this method is cheaper than interview surveys while still having a person asking the questions reducing the change of misinterpreting a question. However, this method is less personal compared to the interview surveys.

The goals of the interviews conducted during this research are to describe the design approach used in practice and figure out how experienced breakwater designers make their decisions. For the design approach, the desired outcome is one, or several, flowchart(s) depicting the practical design approach, similar to the ones described in Section 2.1. As the focus is on describing a process it is beneficial to have some flexibility, since this allows for asking specific questions about the described design approach. Therefore, the interview survey is chosen as the method for the interviews. Furthermore, during an interview survey, it is possible to ask questions to clarify certain steps of the design approach.

3.1.2 *Structure of the Questions*

The second decision that must be made is the structure of the question asked during the interview. There are three different ways questions can be structured (Onencan, 2013):

- Unstructured, in which no questions are determined beforehand. The interview and the interviewee have a conversation about the topic of interest.
- Structured, in which all questions are written out beforehand and the interviewer knows what he or she wants. There is no deviation from the script of questions.
- Semi-structured, some questions are determined beforehand but the interviewer has the flexibility to add more. The script with questions is guiding, but not leading the interview.

In the previous subsection flexibility was the main argument to choose for the interview surveys, as this allows to ask follow-up questions. It is therefore not possible to choose for the structured method, as this would nullify the advantage of the interview survey. The unstructured and semi-structured structures thus remain.

Not all questions are known before the start of the interview since the approach to be described is an unknown as well. Combined with the desire for flexibility during the interview means that the semi-structured method is the best solution for this case. Furthermore, McLaughlin (2006) stated that one of the aims of the semi-structured method is to describe events or processes, further emphasizing the benefits of this method. Moreover, Jacob and Furgerson (2012) advised inexperienced interviewer to use an interview protocol, which is a script containing the questions and structure of the interview. While it is not necessary to read the protocol word for word it is important to have since it will help to structure the interview. During an unstructured interview, it is not possible to have an interview protocol, as this contradicts the word unstructured and will make the interview structured. Therefore, the semi-structured approach is used, since this allows for some flexibility while being somewhat structured.

3.1.3 *Type of Questions*

Finally, the type of questions, two types of questions exist according to Babbie (2011):

- Open-ended questions, in which the interviewee is asked to provide his or her answer.
- Close-ended questions, where the interviewee is asked to select an answer from a list of answers defined by the interviewer.

The main requirement is flexibility, therefore open-ended questions are the best type of questions to use in this case. However, open-ended questions are more difficult to process as there is no, or almost no, uniformity (Babbie, 2011). Therefore, closed-ended questions are used to confirm the response of the interviewee. For example, a closed-ended question might be to confirm the design approach of the interviewee, which was derived based on his or her answers.

During the interview, the interviewer must always be aware of his own bias. Since the bias of the interviewer could steer the interviewee towards giving a certain response, an example of such a question is, *"Don't you agree that ..."*. This question implies that the interviewer agrees with the statements, and the interviewee might thus be inclined to give an affirmative answer.

3.1.4 *Structure to be Used*

The interview method used in this chapter is thus a semi-structured interview survey with open-ended questions. During the interview an interview protocol is used, this protocol, introduced in Subsection 3.1.2, helps to structure the interview and guarantees that questions are not forgotten. The interview is divided into four parts:

- Introduction, in which the interviewer introduces himself and describes the topic of the research. Furthermore, consent is asked to record the interview, as this makes it easier to post-process the interviews and allows the interviewer to focus on the answers without the need to write them down.
- Questions about the design process, in this part the interviewer wants to derive the design approach used by the interviewee. The goal is to make a flow chart of the design approach used in practice by the designer.
- Questions about generating and selecting concepts, this set of question will be about the considerations of the designer. Mainly focusing on how concepts are generated, i.e. how does the designer determine of which breakwater types (Figure 1.1) a concept is designed. Furthermore, when all concepts have been designed, how does the designer, or project team, select the most promising concept. What are the main requirements, and is a certain method used?
- Finally, the interview is concluded by thanking the interviewee for his or her time and asking if he has any final remarks.

The full interview protocol used during the interviews is included as Appendix B.

3.1.5 *Interviewees*

Three experience breakwater designers have been interviewed to understand the practical design approach. The experience, type of company and type of projects is presented for each designer in Table 3.1

Table 3.1: Description of the interviewed designers

Interviewee	**Experience**	**Nationality**	**Company**	**Type of projects**
A	20 years	Dutch	Engineering	Several projects, ranging from rubble mound to caisson breakwaters
B	30 years	Dutch	Contractor	Several projects, ranging from rubble mound to caisson breakwaters
C	18 years	Dutch	Contractor	Mainly conceptual designs of rubble mound breakwaters with armour units

3.2 Interview Results: Practical Design Approach

This section aims to understand the practical design approach, and investigate how this approach compares to the engineering design methods described in Section 2.1. The three interviewees all described a similar approach, although there are some small nuances between each of the described approaches. These nuances are the result of the different roles and different companies. A designer from an engineering consultancy firm describes a slightly different process compared to a designer of a contractor. Therefore, in Subsection 3.2.1 the general described approach is presented. Whereafter, Subsection 3.2.2 elaborates on the differences, and remarks made by the interviewees about the design approach. Finally, with the aid of the Subsection 3.2.1 and 3.2.2 the following questions are answered in Subsection 3.2.3:

- What approach is used in practice to design breakwaters?
- How does the approach in practice compare to engineering design methods?

3.2.1 *Described Design Approach*

In general, the interviewees all describe a similar approach. The practical design approach is described as a routine process with many repeating tasks, especially the computations performed in the early design phases. Furthermore, all interviewees describe a sequential process with several phases, although the terms used to describe the phases differ. However, the tasks described to be performed in each phase are in fact similar. Therefore, it is possible to define a general design approach, which is presented in Figure 3.1.

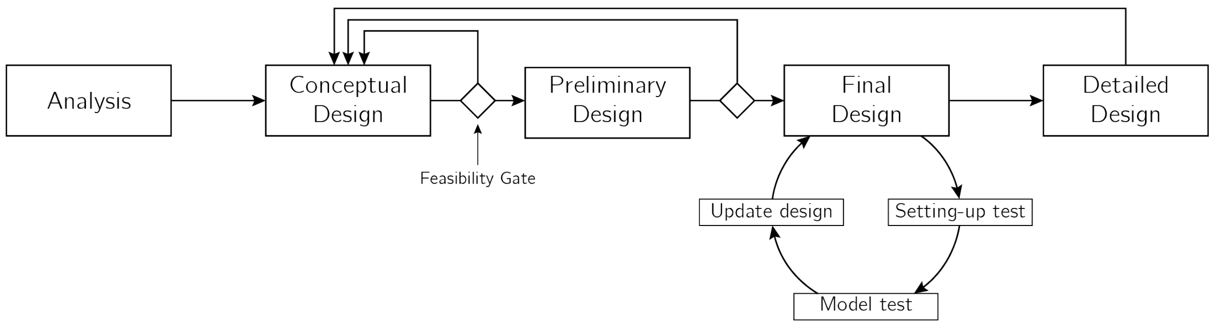


Figure 3.1: General design approach as described by the interviewees

Analysis
The first phase of the design process is: determining the boundary conditions, which are the hydraulic conditions such as wave height and water levels. Furthermore, the requirements and wishes of the client are analysed. These requirements are based on the function the breakwater should fulfil when completed, for instance, to protect a port. From this analysis, requirements on the maximum allowed overtopping and wave transmission are also determined.

Conceptual Design
During this phase, several breakwater concepts are developed. In general, the designers use the hydraulic and soil conditions derived in the previous phase to decide which type(s) of breakwater(s) will be developed. However, also the availability of material and equipment plays al large role in this decision. The precise process and considerations made by the designer is the topic of Subsection 3.3.1.

For each of the developed concepts a 2D design is made to estimate the required material, which is used to make a first, rough, cost estimation. A constant consideration during this phase is, how the breakwater is going to be constructed, as most costs are made during the construction. Interviewees B and C mentioned, independently from each other, the example that, if according to the computations a 2 m^3 armour unit is required it is sometimes better to construct the breakwater with 6 m^3 armour units to reduce the number of armour units, which reduces the construction time, and thus reduce the total cost.

At the end of the conceptual design phase the most promising concept must be identified, in Figure 3.1 depicted as the *feasibility gate*. The most common method to help the designers with their decision is experience and the multi-criteria analysis, but how the designers choose the most promising concept is the topic of Subsection 3.3.2.

Preliminary Design
In this phase, the concept from the previous phase is further engineered, for instance, the dimensions of the underlayers and the required rock class. The design in this phase is still a 2D cross-section. Then at the end of this phase a decision must be made on whether to proceed with the concept to the next phase or return to the conceptual design phase and develop new concepts.

Final Design
In the final design, the design is tested in a wave flume, although tests are not always performed. The decision whether to test in a wave flume mainly depends on the complexity and/or the size of the breakwater. In case the design is tested in a wave flume this is often only done for a 2D design since most of the uncertainty can be determined with a 2D test. This enables the designer to better assess the uncertainties and risks, and thus make a more reliable cost estimate. However, for larger projects where risks and uncertainties are larger, 3D tests are also performed to, for instance, test the round head. The test case used is an overload case in which all extreme conditions, design wave height are increased by 20%. Extreme water level, storm surge and sea-level rise are thus all combined into one storm. In Figure 3.1 the model test is represented by a cycle since the design is updated if the structure fails during the test.

It must be noted that the model tests are sometimes also done during the conceptual design phase. This was, for instance, the case for the Afsluitdijk project. The reason this was done is that the risks and uncertainties for such a project, 550 million EUR (Rijkswaterstaat, 2018), are large. In these cases, the increase in certainty, which decrease the cost, outweigh the added cost of carrying out model tests.

Detailed Design
Finally, the detailed design is made. In most projects, this is the phase a contractor starts his work because generally the contractor does not have to design the breakwater but receives a design from the consultant of the client. However, transitions and interfaces in the design still need to be worked out. An example given by Interviewee B is a lighthouse at the end of the breakwater. But he also remarks that there have been projects where the design of the consultant was rejected by the contractor. In this case, the contractor returns to the conceptual design and carries out his own design process.

3.2.2 *Remarks of the Interviewees about the Design Approach*

This subsection elaborates on nuances between the described approaches, and remarks made during the interviews. The first remark is about the described approach, all interviewees state that the approach from Figure 3.1 is a good graphical representation of the traditional design process. Traditional here meaning a process where a design is made by a consultant, and the contractor only joining for the detailed design process. This can also be seen in the description of the detailed design phase, at the end of the previous subsection, and is also the reason that there is an arrow from detailed towards the conceptual design phase. However, according to Interviewee A, the described design approach is less suitable for more complicated contracts as, for instance, DBFM[1] or Design&Construct contracts, because the contractor is involved from early on in this process.

In the conceptual design phase and subsequent phases, computations are made to determine the dimensions of the breakwater. All interviewees reported that these computations are done in a deterministic way and that no probabilistic method is used to determine the dimensions of the breakwater. Interviewee B even remarks: *"we don't make a probabilistic design. There is no one who does that."* He adds to this that often the required size of the armour is larger with a probabilistic method than a deterministic method. With the knowledge that the armour of the existing breakwater is smaller, and did not collapse, the interviewee concludes that using a probabilistic approach results in a more conservative design. The result is a more expensive bid, and thus a higher probability of losing the tender.

[1] DBFM is the abbreviation used in The Netherlands for the integrated contract form of Design, Build, Finance and Maintain

Besides this reason the following two reasons were also mentioned:

- For most projects, it is not possible to determine the correlation between the wave height and water depth, because often the time series data of the water level is missing.
- It is difficult to determine the uncertainty of some of the equations.

However, not using a probabilistic method does not imply that safety is not taken into account. On the contrary, the fact that the interviewees report to test their designs in a wave flume with an overload case means that the safety of the design is tested. Furthermore, the fact that the wave height and water depth are taken as fully correlated also introduces some safety. Although, this differs between projects as the additional safety depends on the correlation.

Then the role of Interviewee C, in Table 3.1 it can be seen that he is mainly focused on designing rubble mound breakwaters with armour units. Over the past decades, several different types of concrete armour units have been developed by different companies. Each of the developed armour units is a patented product, owned by the company. For example, Xbloc or XblocPlus, which are concrete armour units patented by Delta Marine Consultants, a subsidiary of BAM Infraconsult. This resulted in the role of Interviewee C. His role is slightly different compared to the role of a designer since it is his responsibility and objective to sell as much armour units as possible. To achieve his objective, he makes an alternative design for a certain project with his type of armour unit, which can be interpreted as an acquisition activity. With his alternative design, he then tries to convince the client, and contractor, that using his armour unit will result in the most optimal and cheapest design.

Another remark is that the differences between the conceptual and preliminary design phase are limited. Interviewee C even states that according to him there is no preliminary design phase in the breakwater design process. However, the preliminary design phase is included as this allows the practical approach to be scaled to larger projects.

3.2.3 *Evaluating the Practical Design Approach*

With the knowledge of the practical design approach, the two questions posed at the start of this section can now be answered.

What approach is used in practice to design breakwaters?
The approach used to design breakwaters in practice was presented in Figure 3.1. It can be seen that this approach does not include the iterative behaviour of the basic engineering design cycle, or the method of De Groot (1994) which was stated to be fundamental. But if the interviews are analysed more closely, and especially the questions about generating and selecting concepts, to be discussed in the next section, it can be seen to be implicitly mentioned by the interviewees. Since a process is described where first the requirements of the client are determined (analysis), whereafter concepts are generated in the conceptual design phase, the synthesis. For each of the concepts a hand computation is made (simulation), which is followed by selecting the most promising concept (evaluation) and a decision on whether to proceed to the next phase or start again (decision).

Therefore, it can be concluded that while the interviewees did not mention the iterative cycle to be part of their approach, it is included in their description of the tasks performed in this phase. It can thus also be concluded that the hypothesis, from Subsection 2.1.5, that if the cyclic behaviour is not explicitly mentioned it is likely to be performed implicitly, is indeed confirmed. Furthermore, this also highlights the statement of Roozenburg and Eekels (1995) that their method is a fundamental design method. The practical approach is therefore adapted to include the iterative behaviour. The adapted practical approach is presented in Figure 3.2.

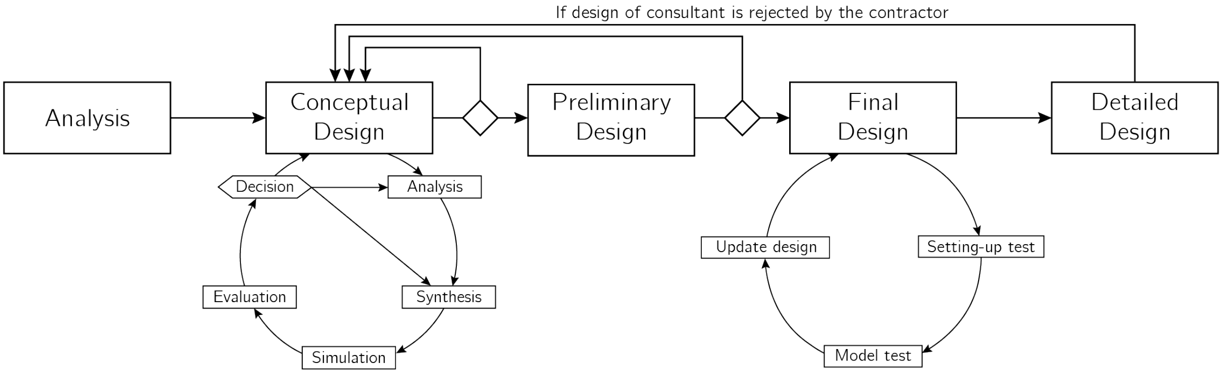


Figure 3.2: Adapted practical design approach to include the iterative behaviour

How does the approach in practice compare to engineering design methods?
At first glance, the practical approach shows many similarities with the Civil Engineering Method and the Rock Manual Method, depicted in Figure 2.2 and 2.3 respectively. The names of the design phases are similar, and the tasks performed in each phase are comparable. Similar to the Civil Engineering Method, the accuracy of the cost estimate was mentioned as a mark to demarcate the different phases.

Items from the systems engineering method can also be observed in the practical approach, mainly the reasoning from system to subsystem. Or in other words, the design process from the larger structure to the smaller components of the structure. This reasoning is observed in the description as concepts are developed during the conceptual design phase, after which the dimensions of the underlayer are determined. Or in other words, from the system, general design of the breakwater, to the subsystem, the individual layers.

The inclusion of the model tests in the practical method seems different at first. However, the model tests are used to verify the design and can thus also be interpreted as the verification of the systems engineering method, or the simulation step from the basic engineering design cycle. Model tests could not be explicitly included in these methods as the methods from Section 2.1 are developed to describe a more general design process, and are thus, as a consequence more abstract.

To summarise, the practical approach has elements of all design methods, since the iterative behaviour described by the basic engineering design cycle was implicitly described in the steps performed during the conceptual design phase. But the description of the tasks performed in the iterations more closely resembles the description of the Civil Engineering Method. Therefore, a similar conclusion, as the one at the end of Subsection 2.1.5, can be drawn namely, that all methods are comparable, and are adaptations of the systems engineering method.

3.3 Interview Results: Generating and Selecting Concepts

Generating concepts and selecting the most promising concept are the most important steps of the design process, as was concluded at the end of Chapter 2. The second part of the interviews therefore focused on how designers perform the necessary steps to, first generate concepts and eventually arrive at a single concept, which is then further developed in the subsequent design phases.

From the description of the phases, Subsection 3.2.1, it appeared that concepts are mostly generated and selected in the conceptual design phase. In this iterative phase concepts are generated in the *synthesis* step, which is the topic of Subsection 3.3.1, and selected in the *evaluation* step, which is discussed in Subsection 3.3.2. In Figure 3.3 these steps are highlighted in the practical design approach. With the knowledge of the process of generating and selecting concepts the following questions can be answered:

- On what basis are breakwater concepts generated?
- How is the most promising concept currently selected?

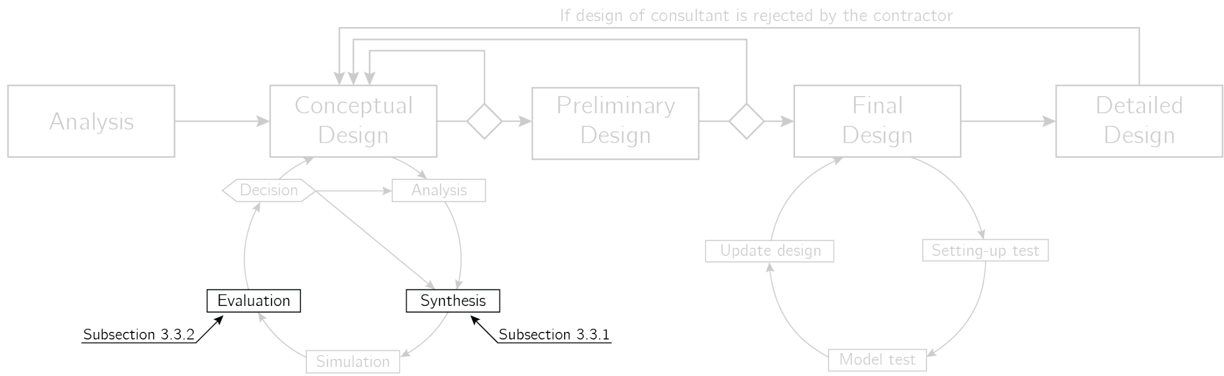


Figure 3.3: Practical design approach focused on the steps where concepts are generated (synthesis) and selected (evaluation)

3.3.1 *Generating Concept*

Generating concepts can be interpreted as the synthesis step from the basic engineering design cycle, Subsection 2.1.1. According to the description of Roozenburg and Eekels (1995) it is the least tangible step as the creativity of the designer is needed to translate the requirements into concepts.

Therefore, one of the first steps of generating concepts is to determine the functions of the breakwater. This is, asking the question: what is the main purpose of the breakwater? Must it protect a port, or is it also used for berths? Determining the function of a breakwater is discussed together with the client, from which the design lifetime with the allowed failure probability of the structure is also determined. Based on the design lifetime and allowed failure probability the design wave height, period and water depth are derived.

The determination of the design wave height is, however, subjected to subjectivity. During one of the interviews, it was mentioned that there was a project where the interviewee deemed the chosen size of concrete armour units too small for the project. According to the interviewee, this was because the designer had chosen a wave height that was too low. The interviewee based his assessment of the size of armour units, and thus the design wave height, on his experience as a designer and breakwaters close to the project. He found that breakwaters close to the project were all constructed with larger concrete armour units, strengthening his opinion about the design wave height. Due to the sensitivity of his opinion, telling the client that his analysis is wrong, it was decided, in what the interviewee calls a technical negotiation, to increase the size of the concrete armour units with a certain safety factor without changing the design wave height.

However, more factors influence this process than just the wave height, as was already concluded in Chapter 1. Another important consideration, mentioned by all interviewees, is the strength of the subsoil and seismic activity. Both are mentioned as having an important influence on the type of breakwater best suited for a project. For example, Interviewee B comments that when the soil consists of soft clay it is rarely a feasible option to construct a caisson breakwater since the bearing capacity of the soil is too low. However, he also mentions that there are projects where the soil was strengthened to support a caisson breakwater. Soil strengthening becomes an option when, for instance, the water depth is quite large which increases the price of a rubble mound breakwater due to the triangular cross-section. Although, soil strengthening might also become an option if the seismic activity is quite large, as a caisson breakwater can better resist seismic activity. Further highlighting that more feasible concepts must be explored, and not be disregarded based on assumptions.

One of the most, if not the most, important considerations for the type of breakwater, according to the interviewees, is the availability of material/equipment. When a caisson breakwater is the preferred type of breakwater, a dry dock is needed to construct the caissons. However, Interviewee B mentions that his company does not have the equipment to construct a caisson in, and must thus rent a dry dock. This results in a relatively large upfront investment compared to the rubble mound types, and might thus make the rubble mound breakwater the preferred alternative after all for his company. Whereas a company with the availability of dry dock might choose to develop caisson breakwater concepts since he does not have this large upfront investment. Interviewee B adds to this that especially companies from Spain, Portugal and Japan have a preference towards caisson breakwater as they have more experience and more equipment to construct this type of breakwater. For them, a rubble mound breakwater is the less attractive type of breakwater.

Where the availability of a dry dock is an important requirement for a caisson breakwater, the availability of rock is important for a rubble mound breakwater. For the armour layer rock is only used when the design wave height is below 3.0 m, for higher design waves concrete armour units are used. However, the underlayer is almost always constructed with rock. The availability of rock, and proximity of a rock quarry, is thus an important requirement when considering a rubble mound breakwater.

Lastly, the presence of an existing breakwater also influences a designer according to one of the interviewees. This bias can also be an explanation of the reason the caisson breakwater was chosen over a rubble mound breakwater in Section 1.1, due to which the tender was lost. When the location of this project is analysed, Figure 3.4, it can be seen that the existing breakwater is a caisson breakwater, indicating that framing of the problem might be the cause to not develop rubble mound concepts.

To summarise, the main considerations of designers when developing breakwater concepts are:

- Availability of material and equipment.
- Hydraulic conditions, the design wave height, period and water depth.
- Geotechnical conditions, the bearing capacity of the soil and seismic activity.

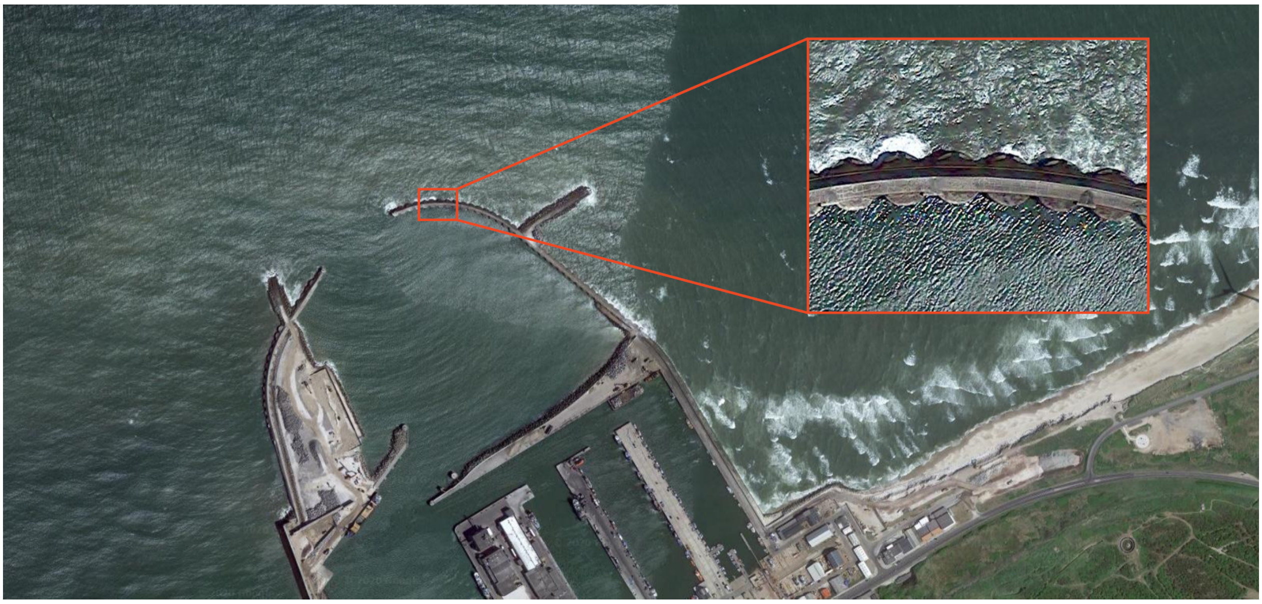


Figure 3.4: Satellite image of the existing breakwater of the project location from Section 1.1, retrieved from Google Maps

3.3.2 *Selection of the Most Promising Concept*

When several concepts have been generated, a decision must be made about which concept is the most promising concept and is further developed in the subsequent phases. Several methods to help the designer have been mentioned in Subsection 2.2.2, one of these was the multi-criteria analysis. The opinion about this method however seems to differ between the interviewees. Interviewee B states that a multi-criteria analysis must be done to select the most promising concept. Interviewee A on the other hand states that designers can easily steer the outcome of the multi-criteria analysis by changing scores or weights, making it a method open to subjectivity and opinions.

To overcome the subjectivity Interviewee A argues to always attempt to translate the performance on a requirement into a monetary value. With this statement the difference in opinion between the interviewees is gone since the first interviewee stated that the cost of a breakwater is always a requirement in the multi-criteria analysis. However, this does illustrate that there is some ambiguity about how, and if, the cost must be included as a requirement in the multi-criteria analysis.

Continuing on the requirements included in a multi-criteria analysis, the question was asked if the interviewees are familiar with the requirements derived by MarCom WG 47 (2016), see Subsection 2.2.3. All interviewees were familiar with the requirements, but their opinion about the requirements differ. Interviewee A describes the requirements as a good first approximation, but that they can never be used as a guideline. Where Interviewee B and C state that a lot of requirements are missing, and that the description of some of the requirements is unclear. Furthermore, Interviewee B adds that a caisson can, for instance, be made permeable, which increases the performance of a caisson breakwater on the requirement to limit the wave reflection.

When asked about the requirements which they include in a multi-criteria analysis, the following requirements were mentioned:

- Cost
- Risks, expert assessment of risks of the concept
- Construction time
- Geotechnical conditions
- Ecological impact of the breakwater

From this list of requirements, especially the risks are open to bias as each person perceives risks differently. Interviewee B provides an example to highlight how risks are perceived differently. In Chile there is a relatively large swell, they, therefore, deem concrete armour units with a complex placement pattern too risky and thus use cubes. However, the interviewee argues that it is not easier to place cubes and that there are projects where concrete armour units with a complex placement pattern were possible. While it is an option to make a bid with a Design&Construct contract, this option is never used, because model tests cannot be performed due to time constraint, and all risks are allocated to the contractor.

Finally, the interviewees were asked to rank several methods to select the most promising concept. In Table 3.2, the ranking from two of the interviewees is presented as they have quite the opposite opinion.

Table 3.2: Ranking of the methods to select the most promising concept from two of the interviewees

Method	**Description**	**Position**	
		Interviewee A	**Interviewee B**
Multi-criteria analysis	Assessing the value based on a set of predetermined requirements	5	2
Value-cost diagram	Diagram in which the value is plotted against the cost	3	4
Design Explorer	The visual method presented in Figure 2.7	4	1
Engineering judgement	Use the judgement of engineers	2	3
Own method	See below	1 [1]	[1,2]

[1] *Own method of interviewee A is experience, and own method of interviewee B is a combination of the multi-criteria analysis and engineering judgement*
[2] *Not ranked as the interviewee reported that it is a combination of two methods already in the ranking*

From the table, it can be seen that where Interviewee A ranks the multi-criteria analysis and design explorer as the worst methods, Interviewee B states that these are the best two methods. It must be noted however that Interviewee A did not include cost as a requirement in the multi-criteria analysis, but stated that the requirements must be monetised to compare them in a more subjective manner. Both interviewees do agree on the importance of engineering judgement, or their experience, as a method to select the most promising concept.

3.3.3 *Evaluating the Selection and Generation of Concepts*

With the knowledge about the generation concepts and the selection of the most promising concept the question, posed at the start of this section, can now be answered.

On what basis are breakwater concepts generated?
The decision about which types of breakwater concepts are developed in the conceptual design phase is more complicated than stated in the Rock Manual. When using hydraulic conditions, for instance, the water depth and design wave height, the chances of making a suboptimal design, and thus losing a tender increase. It is therefore important to consider other factors, such as the proximity of a quarry or dry dock. But the soil conditions play an important role in the selection of the type of breakwater to develop as well. Furthermore, also the company constructing the breakwater has a large influence on this selection, since it was stated by the interviewees that the availability of equipment and experience with a certain breakwater type also plays an important role.

However, interviewees also remarked that the existing breakwater at the project location can also influence the decisions for a certain type of breakwater. This can be interpreted as choosing for a familiar concept, which means that the designer is biased towards a certain solution. It might be that the bias is inadvertent, as Rietzschel et al. (2010) reported to be the case when designers consider unfamiliar or creative concepts, which might prevent the designer from generating more diverse concepts, as Nijstad et al. (2002) reported. Alternatively, it can also be caused by incorrectly framing the problem as Dorst (2011) reported.

Furthermore, it can be concluded that the experience of the designer has a profound impact on the design process, as Lawson and Dorst (2005) also stated. The conclusion is based on the fact that designers rely on rules of thumb, for instance, for a design wave height above 3m rock is often not an option. And the fact that examples of existing projects were often given during the interviews.

How is the most promising concept currently selected?
Several methods have been developed to help the designer select the most promising concept, as could be seen in Section 2.2.2. From Table 3.2 it can be concluded that, while there is no clear preferred method, there is a preference towards using experience and engineering judgement, confirming the point by Lawson and Dorst (2005) that experience plays an important role. The multi-criteria is also mentioned as an important method. But whether the cost must be included in the multi-criteria analysis, or included in the form of the value-cost diagram depends on the designer.

The requirements to be included in a multi-criteria analysis can be based on the requirements derived by MarCom WG 47 (2016), Table 2.3 and 2.4, but the interviewees agree that they must only be used as a guideline, but never copied directly. Requirements mentioned by interviewees to be included in a multi-criteria analysis were:

- Cost
- Risks, expert assessment of risks of the concept
- Construction time
- Geotechnical conditions
- Ecological impact of the breakwater

3.4 Analysis of the Breakwater Database

The breakwater database, developed by TU Delft and HR Wallingford (2019), consists of 522 breakwaters with data ranging from design wave height to contractor. The data in the database is a direct result of current breakwater design practice because the data describes actual breakwaters constructed around the world. The database can, therefore, be used to investigate some of the statements made by the interviewees and literature about the selection of breakwater. However, for many of the breakwaters data is missing on several parameters. It is therefore only possible to investigate the statements regarding the location and hydraulic conditions. Furthermore, this subsection will only focus on the rubble mound and caisson types, a more elaborate analysis of the database can be found in Appendix C.

First, the relation between the type of breakwater and water depth is investigated. In the Rock Manual a water depth of 15 m is given as the transition between a rubble mound and caisson breakwater (CIRIA, CUR, CETMEF, 2007), which is a good first estimate according to Interviewee B. However, the average water depth in which the caisson breakwaters from the database have been constructed is 24.63 m with a standard deviation[2] of 13.15 m. This is significantly larger than the 15 m of the Rock Manual, but it must be noted that the standard deviation is quite large compared to the mean, which indicates that more factors have an influence. In Table 3.3 the mean and standard deviation of the design wave height, water depth and length for the rubble mound and caisson breakwaters are presented.

Table 3.3: Mean (μ) and standard deviation (σ) of the wave height, depth and length for each rubble mound and caisson breakwater types. Data from TU Delft and HR Wallingford (2019).

	Hs [m]		**Depth [m]**		**Length [m]**	
	μ	σ	μ	σ	μ	σ
Rubble mound	5.61	2.07	12.17	8.89	2145	3785
Caisson	6.88	1.52	24.63	13.15	898	556

The length of the breakwater was mentioned during one of the interviews to be of influence, and that when the length increases a caisson breakwater becomes the better alternative. But with the data from Table 3.3 this conclusion cannot be drawn. On the contrary, the average length of caisson breakwaters in the database is shorter than the average length of rubble mound breakwaters. However, it must be noted that the standard deviations are large, for the rubble mound the standard deviation is even larger than the mean. Furthermore, the length of almost half of the breakwaters is not known. It is therefore difficult, if not impossible, to draw a meaningful conclusion.

It was also mentioned, in Subsection 3.3.1, that there is a preference towards certain type of breakwaters in certain countries. In Figure 3.5 the type of breakwater is plotted together with the location to investigate if such a preference can indeed be observed. From the figure, it can be concluded that, as mentioned, there is indeed a preference towards the caisson type in Japan and Spain. For Japan a likely explanation is the fact that caisson breakwaters are the preferred type when seismic activity is an important requirement, see Subsection 3.3.1 and Table 2.4. However, for Spain this is not the case, rubble mound and caisson breakwaters are constructed in similar hydraulic conditions, indicating that other factors indeed influence the selection of the breakwater type. It might be that the contractor who constructed these breakwaters has more experience with caissons, but for most breakwaters the contractor is not specified, making it impossible to investigate this relation.

3.5 Concluding Remarks

To conclude this chapter, and part, the final question is answered.

Does design automation have the potential to improve the breakwater design process?
It seems that in current design practice designers sometimes jump to conclusions when dealing with familiar situations, and choose familiar concepts over unfamiliar ones. However, this might not always be the optimal concept, as was the case for the example in Section 1.1. It was shown by Toh and Miller (2015); Nijstad et al. (2002) that if designers are made aware of their bias, unfamiliar solutions will be considered. One of the advantages of design automation is that concepts can be generated quicker, and it thus becomes easier to consider different concepts.

The main advantage of design automation is the fact that it can generate a large number of concepts in a short amount of time. This can help the designer in better identifying feasible breakwater types for the considered project. Furthermore, in the early design phases several repeating tasks are performed, for instance, the computations required to determine the dimensions. The design automation tool can be used to qualitatively assess if certain breakwater concepts are feasible when meeting with clients. Designers therefore no longer have to rely solely on the rules of thumb and can qualitatively assess the feasibility of concepts.

[2]To compute the standard deviation the assumption has been made that the data is normally distributed

For design automation to be applicable in a design process, a design process must have routine tasks handling explicit information (Cederfeldt, 2007). In the breakwater design process, it is possible to automatically generate several concepts, since the freedom of a breakwater designer is more limited compared to, for instance, a product designer. This is because the most common type of breakwaters are quite well defined, see Figure 1.1. It is, therefore, possible to generate concept using a morphological chart, making the breakwater design process even better suited for design automation.

To conclude, the breakwater design process can be automated because the process consists of several routine and repeating tasks, especially during the conceptual design phase. Furthermore, the fact that concepts can be generated, at least the most common breakwater types, with a morphological chart makes the conceptual design phase even more suitable for design automation. Added to this that if designers are made aware of their bias, unfamiliar solutions will be considered. The conclusion is drawn that design automation does have the potential to improve the breakwater design process.

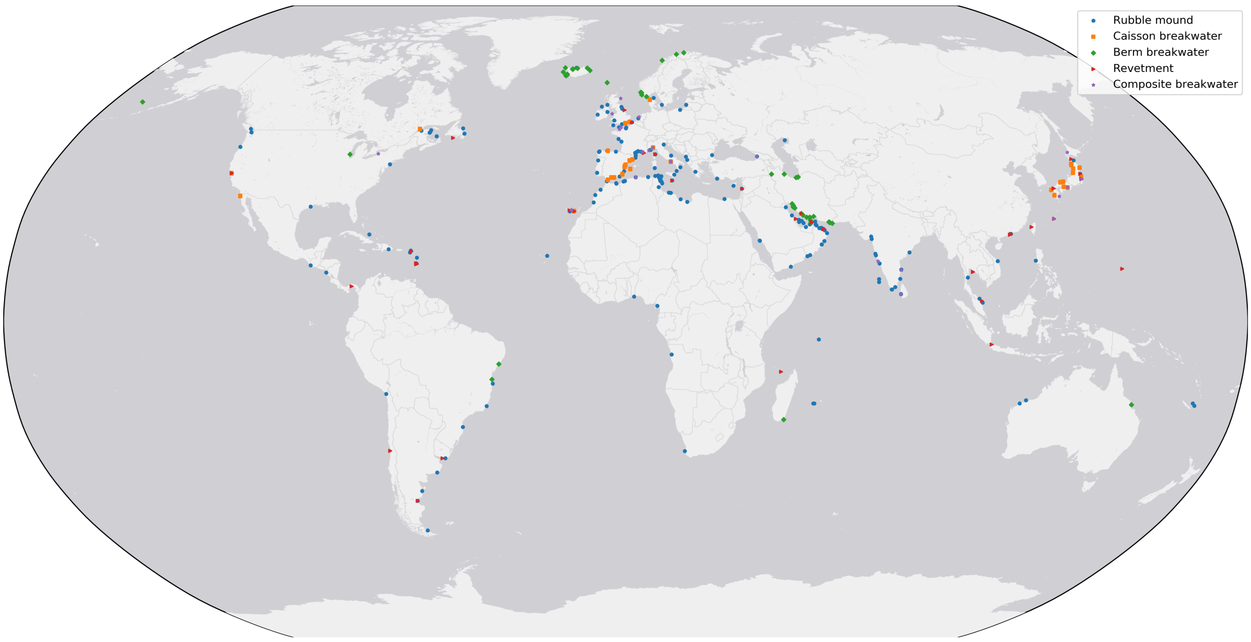
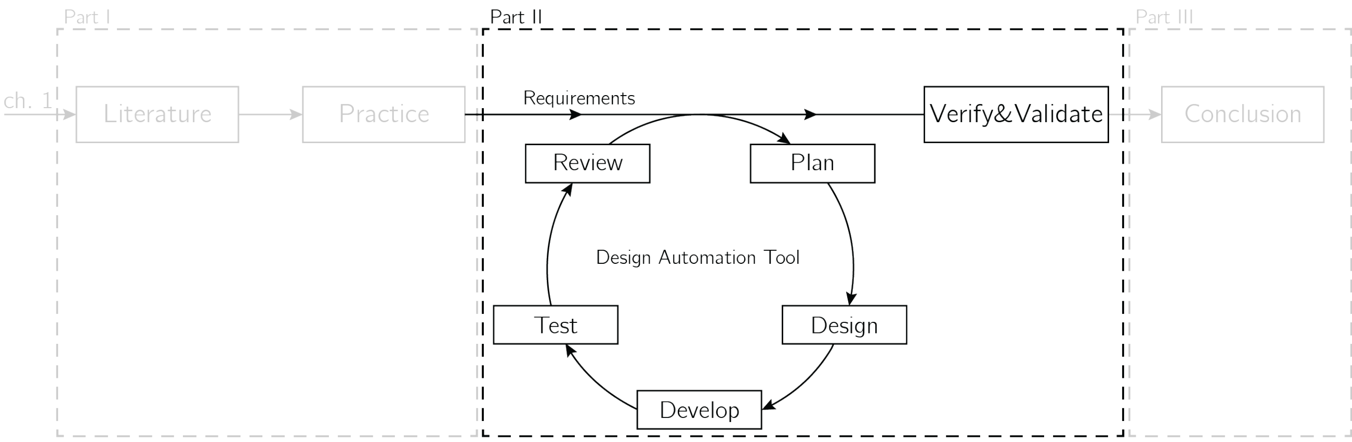


Figure 3.5: Location of different breakwater types. Data from TU Delft and HR Wallingford (2019).

II Design Automation



4

Developing a Design Automation Tool

At the end of the previous chapter, in Section 3.5, it was concluded that design automation does indeed have the potential to improve the breakwater design process. The conceptual design phase of the breakwater design approach (Figure 3.2) appeared to be especially suited for design automation since designers want to quickly explore feasible concepts. The topic of this chapter is therefore the automation of, a part of, the conceptual design phase.

The requirements that the tool must fulfil were derived in the first section of this chapter. These requirements are used in the subsequent section to verify existing breakwater design automation tools since several tools have been developed over the past years. The main goal of this verification was to check if an existing tool can be used in the breakwater design process, or be expanded in such a way that it can be used in the breakwater design process.

From the verification, it was concluded that none of the tools can be used, or expanded. A new design automation tool was therefore developed in Section 4.3. This section describes the included breakwater types, failure mechanisms and elaborates on the structure of the tool. Hereafter, in Subsection 4.4, three demonstrations of breakwater design with the tool are presented. Then, in the final section, some concluding remarks are presented.

4.1 Requirements of the Tool

This section derives the requirements for the design automation tool. These requirements were derived based on the interviews conducted with experienced breakwater designers, see Chapter 3, since the designers are the people who are going to use the tool, and are thus the most important stakeholder. The requirements were divided into the five categories defined by Royal BAM Group nv (2008), see Table 4.1

Table 4.1: Requirement categories as defined by Royal BAM Group nv (2008) with the corresponding subsection

Subsection	Category	Definition
4.1.1	Functional	Requirements about primary functions of the system
4.1.2	Aspect	Requirements about supporting functions of the system
4.1.3	Object	Requirements about objects of the system
4.1.4	Interface	Requirements about the interface with an external system
4.1.5	Process	Requirements about activities required to develop the system

4.1.1 *Functional Requirements*

F.01 The design automation tool shall include at least one rubble mound and one monolithic type breakwater, see Figure 1.1.

F.02 The design automation tool shall be able to design multiple concepts, variations, of one breakwater type to make it possible to investigate the influence of a parameter on the design.

F.03 The design automation tool shall be able to design a breakwater concept within 1 second so that designers can quickly explore different designs.

F.04 The design automation tool shall include construction cost so that designers are able to see an estimate of the cost.

F.05 The design automation shall include the availability of equipment/material so that the designer is able to make a better decision between several breakwater types.

F.06 The designer shall be able to choose the optimal concept. This means that the tool must not select the optimal concept from a set of concepts without designer interaction. This is because interviews showed that experience and engineering judgement are important selection methods. Furthermore, the definition of the optimal breakwater concepts, and the requirements to verify this, vary from project to project.

F.07 The design automation tool shall include hydraulic failure mechanisms to be able to make a design.

F.08 The design automation tool shall include geotechnical failure mechanisms to be able to investigate if the subsoil is strong enough to carry the breakwater.

4.1.2 *Aspect Requirements*

A.01 The design automation tool shall be easy to expand so that it is possible to further develop the tool in the future. Where easy to expand is defined as dividing the code into modules[1] and adhering to the Python style conventions, so that the tool is structured.

A.02 The design automation tool shall be open-source. This allows other developers to further develop the tool and allows for designers and developers to see how a design is generated by the tool.

4.1.3 *Object Requirements*

O.01 All parameters of the implemented equations not having a constant value shall be allowed to be changed.

O.02 The design automation tool shall include a documentation so that it is clear for users what features are available for usage

4.1.4 *Interface Requirements*

I.01 The design automation tool shall be able to be used in the breakwater design process, so that the tool can be used in practice by designers.

I.02 The design automation tool shall be able to load input from, and export data to, an Excel sheet.

4.1.5 *Process Requirements*

P.01 The design automation tool shall be developed in accordance with PEP8[2]. PEP8 elaborates on style conventions for writing Python code (van Rossum et al., 2013).

P.02 The design automation tool shall be developed in accordance with PEP287. This PEP describes the conventions to use when documenting Python code (Goodger, 2002).

4.2 Evaluation of Existing Design Automation Tools

This section investigates the possibility to use an existing design automation tool. Three tools, see Table 4.2, are discussed in the first three subsections. Finally, in Subsection 4.2.4 the tools are verified against the requirements listed in the previous section.

[1] In Python, a file (.py) is referred to as a module instead of a file

[2] PEP stands for Python Enhancement Proposals, and is a design document providing information to the Python community

Table 4.2: Existing design automation tools discussed in this section

Reference	Subsection	Title
Sijbesma (2019)	4.2.1	A parametric approach to a probabilistic design of rubble mound slope protection
De Smet (2015)	4.2.2	Excel-based design tool for the assessment of rubble-mound and caisson breakwaters
Laenen (2000)	4.2.3	Probabilistic model for comparing two breakwater types based on economic optimisation[1]

[1] *Title is translated from Dutch*

4.2.1 *Probabilistic Design Tool for Rubble Mound Breakwaters*

The design tool developed by Sijbesma (2019) was developed in Python, besides the input, which must be specified from an Excel sheet. The developed tool designs a rubble mound breakwater using a full probabilistic approach, meaning that the uncertainties are taken into account. The incorporated methods are a first-order reliability method (FORM) and a crude Monte Carlo. Since the tool is developed in Python it is possible to expand the tools with different modules, for instance, a caisson design module, or create a link with another software package. Therefore, the author of this tool was contacted and asked if the tool could further be developed in this research. However, unfortunately, no answer was received, and it was therefore assumed that the tool is not open-source.

The tool can design a rubble mound breakwater with rock as the armour layer and considers the following three failure mechanisms: overtopping, armour layer and toe stability. Within a short amount of time (3.5 minutes) about 29.000 configurations can deterministically be designed. Since it is hard, if not impossible, for a designer to effectively assess the configurations, an optimisation method is used. One of the optimisation parameters is the cost. The included costs are: the construction, limited to material or volumetric costs, and maintenance costs. The optimisation method used is a local optimisation function, this means that several configurations are suggested as solutions. It is therefore up to the designer to choose the optimal solution, but no selection method is provided to assist the designer.

One of Sijbesma's conclusions is that this full probabilistic approach results in a more expensive breakwater (Sijbesma, 2019, p.51), which was also stated by the interviewees in Subsection 3.2.2. As the interviewees were critical of a probabilistic approach because of the higher cost, using a probabilistic design approach is interpreted as a disadvantage, at least from a practical point of view.

4.2.2 *Excel Based Design Tool for Rubble Mound and Caisson Breakwaters*

The design tool developed by De Smet (2015) is developed in Excel with Visual Basic for Applications, or VBA for short. The developed tool can design a caisson breakwater and rubble mound breakwater with rock or armour units as armour layer. However, the decision for a rubble mound breakwater with armour units is automatically made by the tool, limiting the freedom of the designer. Furthermore, when armour units are used by the tool it is no longer possible to select the slope since the slope is automatically selected based on the chosen armour unit.

Based on the given input both breakwaters are designed, this is only done for the hydraulic stability, similar to the tool described in the previous subsection. In the end, two breakwater concepts are presented to the designer, one rubble mound and one caisson concept. It is then up to the designer to choose the optimal design, but no method is provided. What is included are the cost of the concepts, the tool includes the following costs: material, transport and construction cost. The transport cost includes the distance over which the materials are supplied to the construction site. However, it is unclear if the transport cost also includes the rent of a dry dock, or if such an investment can be incorporated in the cost estimation.

While the report of De Smet refers to an Excel file containing the model, this file was not found on the site of the University of Ghent. Furthermore, while it is possible to expand a VBA script, VBA itself is not open-source and mainly developed for automating Microsoft Office making a general-purpose programming language, such as Python or C more suitable and flexible towards the future.

4.2.3 *Probabilistic Design Tool for Rubble Mound and Caisson Breakwaters*

The design tool developed by Laenen (2000) is developed in Mathcad, a programme in which mathematical computations can be programmed. Different from the tools described in the previous subsections, the source code of this tool is publicly available as it is included in the appendix of the report. However, since Mathcad is not a general-purpose programming language it is less flexible to expand. Furthermore, Mathcad is proprietary software, meaning that a license must be bought to run the programme. Hence, while the developed tool might be open-source, a license for Mathcad is required to use the tool.

Two types of structures are included in this tool, a rubble mound breakwater with rock as armour layer and a caisson breakwater. The varying parameters of each structure are automatically optimised to determine the most economical concept. Both types are presented to the designer, and it is then up to the designer to choose the optimal concept for the considered project. It must be noted however that the presented concepts are always economically optimised. This automatic optimisation limits the freedom of the designer because it is not possible to turn it off. The following costs are included in the economic optimisation: construction cost and maintenance cost. The cost of transporting rocks from the quarry to the project site and the rent of a dry dock are included in the construction cost.

The tool incorporates a probabilistic method to compute the failure probability for the defined failure mechanisms. Where the tool discussed in Subsection 4.2.1 incorporated a FORM and crude Monte Carlo, this tool only incorporates the FORM. The failure mechanisms include the hydraulic failure mechanisms and three geotechnical failure mechanisms for the caisson breakwater. These are: occurrence of a shallow shear plane through the foundation, occurrence of a deep shear plane through the foundation and the occurrence of a shear plane in the subsoil. No geotechnical failure mechanisms were implemented for the rubble mound breakwater.

Finally, the input for this tool is not as flexible as the other tools. The crest of a rubble mound breakwater is, for instance, fixed to a value of 8.6 m since the assumption is made that the rubble mound breakwater is constructed from land, which requires a minimum crest width.

4.2.4 *Verification of the Existing Tools*

Based on the requirements from Section 4.1 the described tools can be evaluated, to investigate if one of the tools can be used. The verification of the existing tools is presented in Table 4.3. None of the tools fulfil requirement A.02 (the tool shall be open-source), as a consequence not all requirements could be verified as the code is not available for testing or usage. While this requirement already disqualifies all tools for expanding a verification against the other requirements is nevertheless performed. The assessment of whether a requirement was fulfilled by the design automation tool was performed by reading the reports, or in other words: a document inspection.

Table 4.3: Verification of the tools described in this section

Requirement	**Sijbesma (2019)**	**De Smet (2015)**	**Laenen (2000)**
F.01: include several breakwater types	✗	✓	✓
F.02: design multiple concepts	✓	✓	✓
F.03: design a concept within 1 second	✓	?	?
F.04: include construction cost	✓	✓	✓
F.05: include availability of equipment/material	✗	✓/✗	✓
F.06: designer selects optimal concept	✓	✓	✗
F.07: include hydraulic failure mechanisms	✓	✓	✓
F.08: include a geotechnical failure mechanism	✗	✗	✓/✗[2]
A.01: tool shall be easy to expand	✓	✓	✗
A.02: tool shall be open-source	✗	✗	✓[1]
O.01: possible to change values of parameters	?	?	?
O.02: include a documentation	?	?	?

[1] *The code is open-source, but the programme required to use the code is not*
[2] *Geotechnical failure mechanisms are included for the caisson, not for a rubble mound breakwater*

4.3 Developing an Alternative Design Automation Tool

Based on the verification of the existing tools, Subsection 4.2.4, it was concluded that none of the described tools fulfil all requirements, a new design automation tool had, therefore, to be developed. Since it was not feasible in the timespan of this research to develop a design automation tool with all breakwater types from Figure 1.1, and all failure mechanism, a further scope restriction was required. The scope of the tool is therefore presented in Subsection 4.3.1.

The implemented failure mechanisms for the conventional and vertical (composite) breakwater, the included breakwater types, are presented in Subsection 4.3.2 and 4.3.3. However, it must be noted that this is not a complete overview of all implemented equation. The method of Battjes and Groenendijk (2000) has, for instance, been implemented to compute $H_{2\%}$, so that it can be computed if it is not known. The reported equations are however a complete overview of the implemented failure mechanism. For a complete overview of all features, included functions, and equations one is referred to the online documentation. Then, in Subsection 4.3.4, the structure of the tool is explained. After which, in the final subsection, the limitations of the computational classes for each breakwater type, named the design classes, are discussed in Subsection 4.3.5.

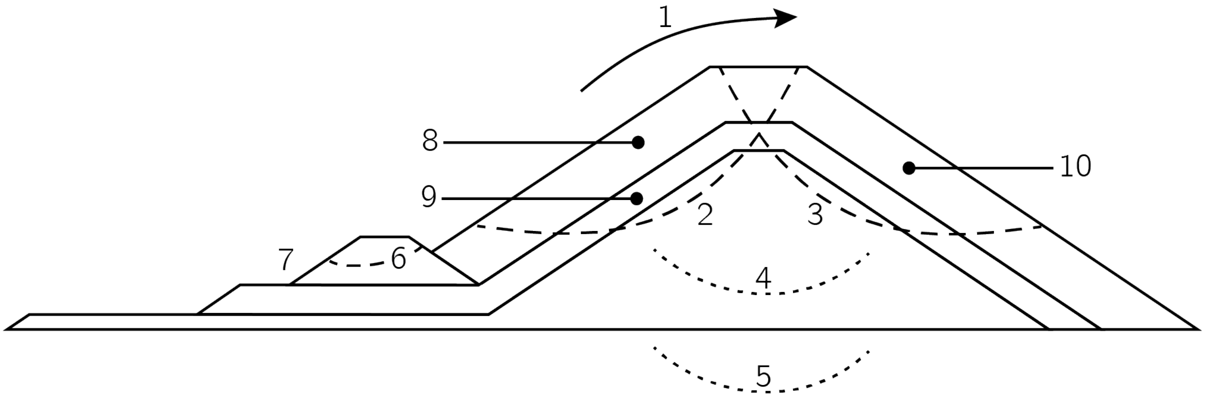
4.3.1 *Scope of the Tool*

In the total timespan of this research, it was not possible to develop a design automation tool with all breakwater types and failure mechanisms. Therefore, a further scope restriction was made. Note that this scope restriction is additional to the one from Subsection 1.4.2.

- The design automation tool is developed for the conceptual design phase of the breakwater design approach, see Figure 3.2
- The design automation tool is developed in Python
- Only one equation is included for each failure mechanism
- The following breakwater types from Figure 1.1 are implemented:
  - Conventional rubble mound breakwater with rock as armour layer
  - Conventional rubble mound breakwater with armour units as armour layer
  - Caisson type on a foundation
  - Vertically composite breakwater
- The assessment of the structural strength, for instance, the required thickness of the walls and reinforcing bars, is outside the development scope.
- Two types of armour units are fully implemented, Xbloc and XblocPlus. However, it must be possible to manually define another type of armour unit.
- Maintenance cost are not considered.

4.3.2 *Failure Mechanisms and Equations of the Conventional Rubble Mound Breakwater*

The failure mechanisms of the conventional rubble mound breakwater are depicted in Figure 4.1, with the implemented failure mechanisms in bold.



1. **Overtopping**
2. **Slip circle outer slope**
3. Slip circle inner slope
4. Settlement of the core
5. Settlement of the subsoil
6. Slip circle in the toe
7. **Toe erosion**
8. **Erosion or breakage of the armour**
9. **Filter instability**
10. Erosion of the armour of the inner slope

Figure 4.1: Failure mechanisms of a rubble mound breakwater, redrawn from Burcharth (1992, p.3). Failure mechanism in bold are implemented in the design automation tool

Overtopping

Overtopping is defined as the quantity of water passing over the crest of a structure per unit of time (Van den Bos and Verhagen, 2018). To compute the required crest height of the breakwater the mean value approach of the general equations for the average discharge on a slope are implemented (EurOtop, 2018, p.115). However, EurOtop (2018) advises to increase the average discharge by increasing the model constants with one standard deviation when designing. The increase of one standard deviation is therefore implemented as a default value, with the possibility to change this value.

$$\frac{q}{\sqrt{g \cdot H_{m0}^3}} = \frac{0.023}{\sqrt{\tan\alpha}} \gamma_b \cdot \xi_{m-1,0} \cdot \exp\left[-\left(2.7\frac{R_c}{\xi_{m-1,0} \cdot H_{m0} \cdot \gamma_b \cdot \gamma_f \cdot \gamma_\beta \cdot \gamma_v}\right)^{1.3}\right] \tag{4.1}$$

With a maximum of

$$\frac{q}{\sqrt{g \cdot H_{m0}^3}} = 0.09 \cdot \exp\left[-\left(1.5\frac{R_c}{H_{m0} \cdot \gamma_f \cdot \gamma_\beta \cdot \gamma^*}\right)^{1.3}\right] \tag{4.2}$$

In which q is the average overtopping discharge per unit width, H_{m0} is the significant wave height from spectral analysis and R_c is the freeboard of the breakwater. Furthermore, the influence factors for a berm, the roughness of the elements, oblique wave attack, a vertical wall on the slope and non-breaking waves for a storm wall on a slope are given by γ_b, γ_f, γ_β, γ_v and γ^* respectively. Note that the tool does not include a function to compute the influence factor for a berm or a vertical wall on a slope, since only the conventional rubble mound breakwater is implemented. However, it is possible to compute these influence factors and specify them as arguments to the overtopping function.

In case of wide crest, defined as wider than $3 \cdot D_{n50}$, the overtopping discharge can be reduced with a reduction factor. The equation to compute this reduction factor is given by (EurOtop, 2018, p.177).

$$C_r = 3.06 \exp\left(-1.5\frac{G_c}{H_{m0}}\right) \quad \text{with maximum } C_r = 1 \tag{4.3}$$

Where G_c is the width of the crest and C_r the reduction factor applied to the overtopping discharge.

Slip circle outer slope

The stability of the slope is evaluated with Bishop's method. With this method, the safety factor, F, which is the ratio of the strength over the load, can be determined. According to the equation the slope is stable if $F > 1$. However, Verruijt (2012) states that the result must always be handled with care as the method is not exact, and there is no guarantee that a slope will be stable with a computed F of 1.05.

The safety factor is determined by dividing the slip circle into several slices, and computing the strength and load of each slice. Summation of the individual strengths and loads results in the total strength and load, from which the safety factor can be determined. This process results in the following equation (Verruijt, 2012, p.274):

$$F = \frac{\sum \frac{c+(\gamma h - p)\tan\phi}{\cos\alpha_s(1+\tan\alpha_s\tan\phi/F)}}{\sum \gamma h \sin\alpha_s} \tag{4.4}$$

Where c is the cohesion of the soil, γ the specific weight of the soil, ϕ the internal friction angle of the soil, h the height of a slice and α_s the slip angle of a slice. From the equation, it can be seen that the factor of safety appears on both sides of the equation. The equation must therefore be solved iteratively.

Slip circles are generated by generating circles that intersect the lowest and highest point of the front slope, point (0,0) and Point2 in Figure 4.2. For each of the generated circles, the safety factor is computed, the circle returning the lowest safety factor is set as the governing slip circle.

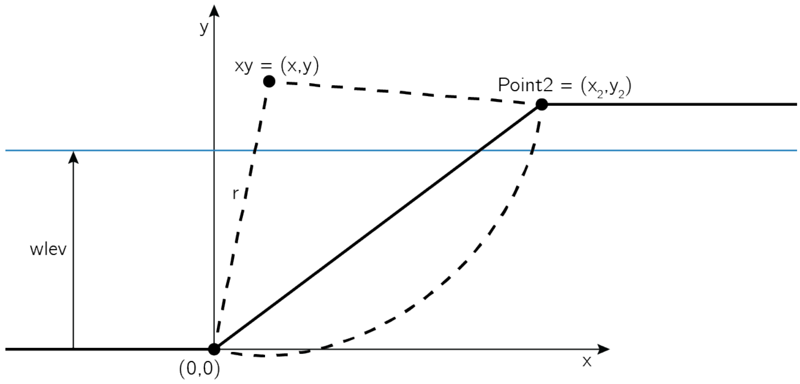


Figure 4.2: Implemented slip circle computation

Toe Erosion

At the seaside of the breakwater the armour layer is supported by a toe. The nominal diameter of the rocks in the toe can be smaller than the nominal diameter of the rocks in the armour layer due to the limited wave action at a depth larger than about one wave height. To compute the required nominal diameter of the toe structure the equation of van der Meer (1998, p.14) is implemented.

$$\frac{H_s}{\Delta D_{n50}} = \left(6.2\frac{h_t}{h} + 2\right) N_{od}^{0.15} \tag{4.5}$$

In which h_t/h is the relative water depth above the toe and N_{od} is the damage number.

Erosion or breakage of an armour layer with armour units

To determine the required diameter of the armour layer Hudson (1959) derived a general stability equation from experiments. The experiments were conducted with rock and Tetrapods as armour layer. For rock this equation it is often not used anymore, because the Hudson formula does not include the effect of the wave period, which does play a role outside the experimental ranges (Van den Bos and Verhagen, 2018). The Hudson formula is therefore only implemented for the armour units, and not for rock armour layers.

$$\frac{H_D}{\Delta D_{n50}} = \sqrt[3]{k_D \cot\alpha} \tag{4.6}$$

In the Hudson formula, K_D is the so-called *dustbin* factor, which includes all unknown factors. Whereas for armour units, K_D can be seen as the stability coefficient that depends on the type of armour unit (Van den Bos and Verhagen, 2018). Note that the Hudson formula includes the wave height as H_D, which is the design wave height. Since Hudson (1959) performed his experiments with regular waves he did not specify which wave height to use when dealing with irregular waves, as is the case when designing a breakwater. For the implementation of a breakwater with armour units, the significant wave height is used, in accordance with the Xbloc design guidelines (DMC, 2014).

Erosion or breakage of a rock armour layer

For the stability of a rock armour layer the equations from van der Meer (1988) are implemented, since these equations do include the effects of the wave period.

$$\frac{H_s}{\Delta D_{n50}} = c_{pl,d} P^{0.18} \left(\frac{S_d}{\sqrt{N}}\right)^{0.2} \xi_m^{-0.5} \quad \text{for plunging waves} \tag{4.7}$$

$$\frac{H_s}{\Delta D_{n50}} = c_{s,d} P^{-0.13} \left(\frac{S_d}{\sqrt{N}}\right)^{0.2} \sqrt{\cot\alpha}\, \xi_m^P \quad \text{for surging waves} \tag{4.8}$$

In which P is the notional permeability of the structure, S_d is the damage level parameter, N the number of waves and ξ_m the Iribarren number computed with the mean wave period. The model constants are given by $c_{pl,d}$ and $c_{s,d}$, and are equal to 6.2±0.4 and 1.0±0.08 respectively.

The experiments performed to derive these equations have been performed in deep water, and the wave heights are thus Rayleigh distributed. However, in shallow water the wave heights are no longer Rayleigh distributed and the ratio of $H_{2\%}/H_s$ is no longer equal to 1.4 (Battjes and Groenendijk, 2000). Van der Meer therefore also suggested equations for shallow water conditions, but these are not implemented. Instead, the adapted Van der Meer equation of Van Gent et al. (2003) are implemented. This is because these include the spectral wave period, $T_{m-1,0}$, which is the better wave period compared to the mean wave period, T_m, for predicting the stability of rock (Van den Bos and Verhagen, 2018).

$$\frac{H_s}{\Delta D_{n50}} = c_{pl,s} P^{0.18} \left(\frac{S_d}{\sqrt{N}}\right)^{0.2} \left(\frac{H_s}{H_{2\%}}\right) \xi_{m-1,0}^{-0.5} \quad \text{for plunging waves} \tag{4.9}$$

$$\frac{H_s}{\Delta D_{n50}} = c_{s,s} P^{-0.13} \left(\frac{S_d}{\sqrt{N}}\right)^{0.2} \left(\frac{H_s}{H_{2\%}}\right) \sqrt{\cot\alpha}\, \xi_{m-1,0}^P \quad \text{for surging waves} \tag{4.10}$$

In which $c_{pl,s}$ and $c_{s,s}$ are the model constant for the shallow water equation, equal to 8.4±0.7 and 1.3±0.15 respectively. Note that the Iribarren number in the shallow water equation is now computed with the spectral wave period, instead of the mean period.

Besides both equations, a general function is implemented that chooses between the deep and shallow water equation based on the input. This choice is based on the validity range, see Table 4.4, of both equations.

Table 4.4: Range of applicability of the Van der Meer equations for deep and shallow water conditions (CIRIA, CUR, CETMEF, 2007, p.579)

$h/H_{s\text{-toe}}$	**Very shallow water** ≈1.5 - ≈2	**Shallow water** < 3	**Deep water** > 3
Deep water equations			
Shallow water equations			

Filter Instability

To prevent the rocks of the underlayer from passing through the voids of the armour layer, and thus prevent the breakwater from collapsing, several rules have been implemented. The following rules of thumb are implemented for a rock armour layer (CIRIA, CUR, CETMEF, 2007), and an armour layer of Xbloc or XblocPlus (DMC, 2014).

$$M_{50u} = \frac{M_{50a}}{15} \text{ to } \frac{M_{50a}}{10} \text{ for Rock} \tag{4.11}$$

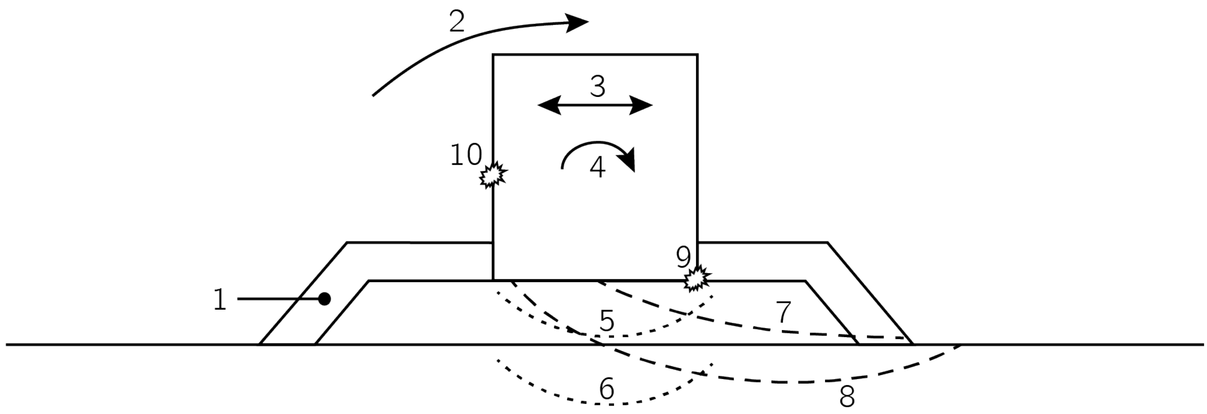
$$M_{50u} = \frac{M_{50a}}{15} \text{ to } \frac{M_{50a}}{6} \text{ for Xbloc} \tag{4.12}$$

$$M_{50u} = \frac{M_{50a}}{20} \text{ to } \frac{M_{50a}}{8} \text{ for XblocPlus} \tag{4.13}$$

Where M denotes the mass of the underlayer, subscript u, and the armour layer, subscript a.

4.3.3 *Failure Mechanisms and Equations of the Vertical (Composite) Breakwater*

The failure mechanisms of the vertical (composite) breakwater are depicted in Figure 4.1, with the implemented failure mechanisms in bold.



1. **Erosion or breakage of the armour**
2. **Overtopping**
3. **Sliding of the caisson over the foundation**
4. **Overturning of the caisson**
5. Settlement of the core
6. Settlement of the subsoil
7. Slip circle through the foundation
8. **Slip circle through the subsoil**
9. **Bearing capacity of the foundation**
10. Failure of the caisson structure

Figure 4.3: Failure modes of a caisson breakwater, redrawn from Van den Bos and Verhagen (2018, p.19) and MarCom WG 47 (2016, p.30). Failure mechanism in bold are implemented in the design automation tool

Erosion or Breakage of the Armour of the Foundation

The material of the armour of the foundation must be large enough to withstand the wave forces. To determine the required nominal diameter of this material the equation of Tanimoto is used (Goda, 2000, p.161).

$$D_{n50} = \frac{H_s}{\Delta N_s} \tag{4.14}$$

In which the stability number, N_s, is given by the following equation:

$$N_s = \max\left\{1.8, 1.3\frac{1-\kappa}{\kappa^{1/3}}\frac{h'}{H_s} + 1.8\exp\left[-1.5\frac{(1-\kappa)^2}{\kappa^{1/3}}\frac{h'}{H_s}\right]\right\} \tag{4.15}$$

Where h' denotes the water depth at which the armour is placed and κ is given by the following equation:

$$\kappa = \frac{4\pi h'/L'}{\sinh\left(4\pi h'/L'\right)}\kappa_2 \tag{4.16}$$

In which L' is the wave length at h', and κ_2 is a coefficient for oblique wave attack.

Overtopping

The method to compute the required crest height for a vertical (composite) breakwater is more extensive than for a rubble mound breakwater. This reason is that several equations are derived to compute the overtopping discharge of a vertical breakwater, or crest height, whereas only one equation has been derived for the rubble mound breakwater. To arrive at the correct equation several assessments must be made. These assessments have been included in a decision chart, which is presented in Figure 4.4

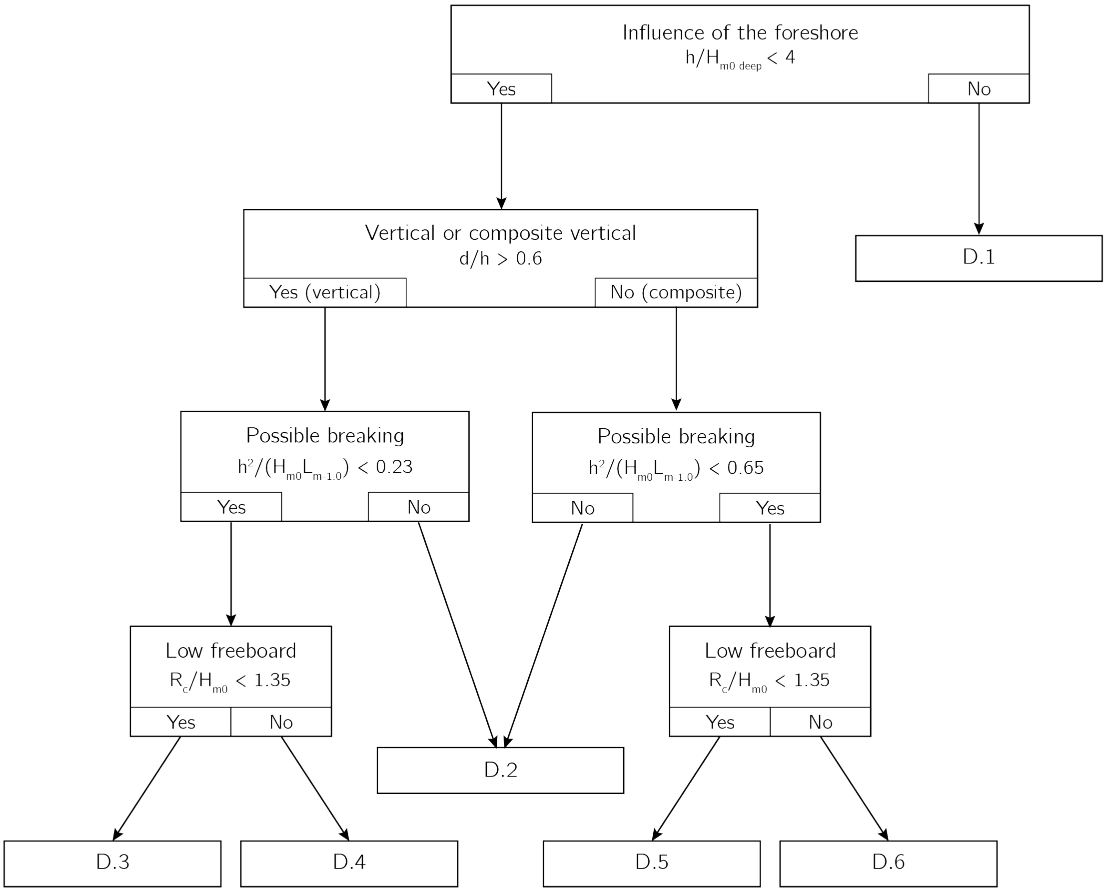


Figure 4.4: Decision chart for computing the overtopping discharge, redrawn from EurOtop (2018, p.194). The equation number corresponds to the equation number in Appendix D.

Sliding of the Caisson over the Foundation

To check if the mass of the caisson is sufficient to prevent sliding of the caisson over the foundation the safety factor against sliding is determined. This safety factor is determined with the following equation (Goda, 2000, p.146).

$$SF_{sliding} = \frac{\mu(Mg - F_U)}{F_H} \tag{4.17}$$

In which μ is the friction coefficient between the caisson and the foundation, and M is the mass of the caisson, note that this mass must be corrected for the submerged part. The forces F_U and F_H are the forces due to the uplift and horizontal pressure respectively. These forces and pressures are computed with the model of Goda-Takahashi, which includes a coefficient for impulsive pressures (Takahashi, 2002). The implemented equations from the Goda-Takahashi wave pressure model can be found in Appendix E.

The width of the caisson, and thus the mass, is computed by the tool and cannot be specified. This is implemented by allowing the user to choose the desired safety factor, which is set to 1.2 by default in accordance with the advice of Goda (2000). With this desired safety level, the required mass is computed, which results in a minimum required width to satisfy the desired safety factor.

Overturning of the Caisson

The mass of the caisson must be large enough to prevent the caisson from overturning. The safety factor against this failure mechanism is evaluated with the following equation (Goda, 2000, p.146).

$$SF_{overturning} = \frac{Mgt - M_U}{M_p} \tag{4.18}$$

In which M is again the mass of the caisson corrected for the submerged part, t is the horizontal distance from the heel (lower right) to the centre of mass. This horizontal distance is fixed in the design classes but can be varied on the core level. Note that the structure of the tool is explained in Subsection 4.3.4. The moments M_U and M_p are the moments due to the uplift and horizontal pressures respectively. These moments are also computed with the model of Goda-Takahashi, which can be found in Appendix E.

The method to compute the required width of the caisson due to sliding is also used to compute the required width due to overturning. This means that the user specifies the desired safety factor, again Goda (2000) advises to use 1.2. The desired safety factor is used to compute the required mass and the minimum required width.

Slip circle through the subsoil

This failure mechanism, a slip circle through the subsoil, occurs if the bearing capacity of the subsoil is exceeded. To determine if the bearing capacity of the soil is large enough the equation of Brinch Hansen (1970) is used. Note that the Bishop method is not implemented because that equation regards the stability of a slope, whereas in this case the soil foundation is assumed to be horizontal.

$$p_{cap} = i_c s_c c N_c + i_q s_q q N_q + i_\gamma s_\gamma \frac{1}{2} \gamma B N_\gamma \tag{4.19}$$

In which i and s are the inclination and shape factors respectively, and N are dimensionless constants. Note that compared to the original equation the depth, base and ground inclination factors have been omitted. The latter two because the assumption is made that the foundation is never constructed at an angle, and the depth factor because the foundation is not constructed in the soil but on top of the soil. Furthermore, c is the cohesion of the soil and q is the overload capacity.

Bearing Capacity of the Foundation

The caisson exerts an eccentric load on the foundation, because the point of application of the uplift pressure is not in the middle of the caisson. This eccentric loading could result in the caisson collapsing into the foundation. The bearing pressure of the rubble mound foundation must thus be large enough to withstand the pressure of the caisson. The pressure the caisson exerts on the foundation is computed with the following equation (Goda, 2000, p.147):

$$p_e = \begin{cases} \dfrac{2W_e}{3t_e} & : t_e \leq \dfrac{1}{3}B \\ \dfrac{2W_e}{B}\left(2 - 3\dfrac{t_e}{B}\right) & : t_e > \dfrac{1}{3}B \end{cases} \tag{4.20}$$

In which p_e is the exerted pressure of the caisson on the foundation, W_e the net downward force and t_e the eccentricity of this force. The computed pressure is checked against the maximum bearing capacity of the foundation, which must be specified by the user. Goda (2000) advises to keep the pressure of the caisson, p_e, below 400 to 500 kPa, although he also states that this limit is increasing in practice towards 600 kPa.

4.3.4 *Structure of the Tool*

The most important requirements during the development of the tool are both process requirements, P.01 and P.02. The process requirements prescribe the usage of two Python Enhancement Protocols (PEPs), these two PEPs are important for the development of the tool as these PEPs elaborate on style conventions when writing Python code, and maybe more important, how to document the code in docstrings. The main advantage of adhering to these conventions is that the readability of the code increases, making it easier for developers not familiar with the code to expand the tool.

In the requirements, Section 4.1, some requirements are contradictory to each other. These are requirement O.01, changing parameters, and F.03, quickly designing concepts. These requirements contradict each other because if the designer wants to change all values, and make all design choices, the required time to make a design increases. This is because the designer then needs to use all individual equations, increasing the required code, and thus the design time. The tool is therefore structured in such a way that both approaches, using individual equations and fast design, are accommodated for. The tool is therefore divided into three levels, where each level can be seen as a building block on top of the previous level. A visualisation of the structure is presented in Figure 4.5. The three levels are:

- **Level 1: The core**
  This is the lowest level of the tool and consists of several modules with the individual equations from Subsection 4.3.2 and 4.3.3. This level allows the greatest flexibility as it is merely an implementation of the equation in Python. At this level, no assumptions or design choices are made in Python.
- **Level 2: The design classes**
  The middle level consists of the, so-called, design classes. These classes can be used to quickly design a single breakwater, for instance, a rubble mound breakwater with rock as armour layer. The equations from the core required to design this structure are implemented in the design classes. This means that the design class automatically calls the corresponding equation from the core, without user interaction, which reduces the required lines of code, from a user point of view at least. However, by implementing the equations from the core into the design classes several design choices were made, limiting the freedom of the designer. The limitations and design choices made in the design classes are elaborated in Subsection 4.3.5.
- **Level 3: Design Automation**
  This is the highest level of the tool and consists of two main features:
  - The Configurations class where several types of breakwaters or several of the same type of breakwater are designed at the same time. This class allows parameters to be entered as varying parameters to design multiple breakwaters at once, and subsequently comparing them by using the design explorer or a multi-criteria analysis, or any other method as no default filter is applied.
  - An interactive design application that can be used to design a maximum of two breakwater types at the same time. Similar to the Configurations class several parameters can be set to varying parameters, for each of these parameters a slider is generated that can be used to vary the value of the parameter. Note that this interactive application does not use the Configurations class, but directly calls the design classes (level 2), see Figure 4.5. This was a conscious design choice and was made to speed up the interactive design application. Furthermore, this also makes sure that concepts are only generated when an explicit request is made by the designer, whereas the Configurations class generates all possible concepts in the specified design space.

The design classes, and design automation tools, limit the freedom of the designer since several design choices were made. These design choices are implicitly made by the designer when using one of these methods, it is therefore important that the designer always critically looks at the generated design(s). If the designer does not want to implicitly make these design choices or desires more freedom in the design of the cross-sections, he or she can use the equations from the core and design the entire breakwater. Alternatively, the designer can also use the functions from the core to develop his own design automation script. However, more lines of code are then required to design a breakwater, but on the other hand, design choices are made by the designer. In the end, it is up to the user to decide which approach best suits the design process, and project in question.

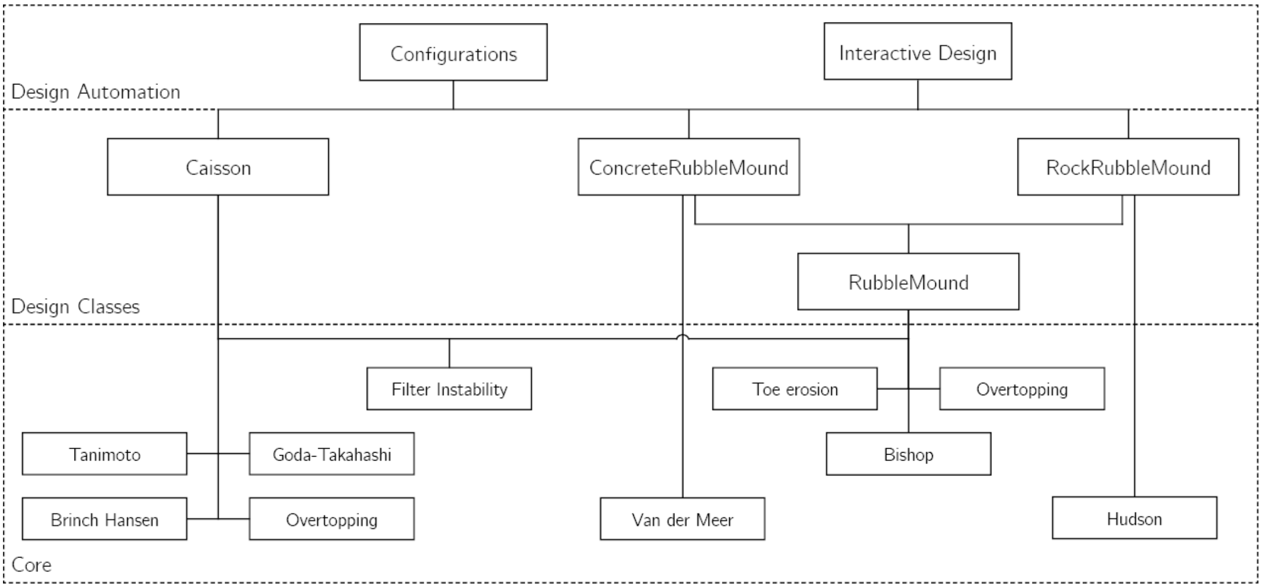


Figure 4.5: Structure of the tool with the three levels

4.3.5 *Design Choices Made in the Design Classes*

As a result of the implementation of the core in the design classes several design choices have been made, as was briefly mentioned in the previous subsection. This subsection elaborates on these choices and implementations, and thus the design limitations of the design classes. The design choices and implementations are:

- The design of the toe cannot be changed in the rubble mound design classes. In case the designer wants to use another configuration the functions from the core must be used.
- Due to the design of the armour layer underlayer and core the notional permeability in van der Meer (1988) and Van Gent et al. (2003) is set to a fixed value of 0.4. This value cannot be changed in the design classes since this requires an alternative design for the layers.
- To compute the required nominal diameter of the armour units the Hudson (1959) equation has been implemented. The significant wave height is currently used as the design wave height, which is in accordance with the Xbloc design guidelines. However, the design guidelines for other structures might specify another wave height.
- To compute the required the nominal diameter of rock a general function which automatically chooses between the equations of van der Meer (1988) and Van Gent et al. (2003) is implemented. This implementation results in two limitations:
  - The general function automatically selects the deep or shallow water equations based on the validity ranges, which makes it impossible to manually force the design class to use an equation. If the designer wants to use a specific equation, the breakwater must be designed from the core, where both equations can be used separately from each other.
  - There is a range where both equations are valid, as can be seen in Table 4.4. In the range where both equations are valid, the design class uses the maximum nominal diameter by default, this can, however, be changed to the minimum or average diameter.
- Currently only an armour layer out of rock, Xbloc or XblocPlus are fully supported. This means that only the filter rules for these materials have been implemented. While it is possible to define another armour unit, it is not possible to define a custom filter rule. In case another armour unit is used in the design classes, the filter rule must manually be set to rock, Xbloc or XblocPlus.
- In the caisson design classes, the caisson is always placed on top of a foundation. This foundation is always placed on top of the subsoil, it is not possible to use the caisson design class with (a part) of the

foundation in the subsoil.

- Several equations have been derived to compute the required freeboard of a caisson, see Appendix D. A general function is defined in the core that automatically selects the correct equation; this selection is based on the decision chart from Figure 4.4. In the design class, it is thus not possible to define which equation must be used. All equations from Appendix D are, however, available in the core as separate functions.
- The caisson design class is limited to designing symmetrical caissons. This is implemented by setting the value of t in equation 4.18 to 0.5, meaning that the centre of mass of the caisson is in the middle of the caisson.

Note that these design choices are also implicitly present on the design automation level, Figure 4.5, since the design classes are implemented in the Configurations class and interactive design function.

4.4 Demonstration of the Design Automation Tool

This section presents three examples of the developed design automation tool, called *breakwater*. In the first subsection, a design class is used to design one concept. Hereafter, in Subsection 4.4.2, an example with the Configurations class is presented to design multiple breakwater concepts at once. Then, in the final subsection, an example is given with the interactive design application.

4.4.1 *Designing One Breakwater*

With the developed tool it is possible to use one of the design classes, see Figure 4.5, to design one concept. In this example, a conventional rubble mound breakwater with armour units as armour layer is designed. The required code is presented in Code Snippet 4.1. Note that the input values are chosen and merely serve as an example. Furthermore, this example is also available on the GitHub repository as *example1.py*.

```python
import breakwater as bw

# compute the unknown wave heights
battjes = bw.BattjesGroenendijk(Hm0=4.4, h=15, slope_foreshore=(1,100))
H2_per = battjes.get_Hp(0.02)

# define a limit state with hydraulic parameters, and the allowed damage
ULS = bw.LimitState(
    h=15, Hs=4.5, Hm0=4.4, H2_per=H2_per, Tp=9.4, Tm=8.8, T_m_min_1=9.7,
    Nod=2, q=20, label='ULS')

# define a material for the armour layer
NEN = bw.RockGrading()

# define a concrete armour unit as armour layer
xbloc = bw.Xbloc()

# design a conventional rubble mound breakwater with xbloc as armour layer
CRM = bw.ConcreteRubbleMound(
    slope=(2,3), slope_foreshore=(1,100), B=5.5, rho_w=1025, LimitState=ULS,
    ArmourUnit=xbloc, Grading=NEN, Dn50_core=0.4)
```

Code Snippet 4.1: Code to design a rubble mound breakwater with armour units

With the code from Code Snippet 4.1 the breakwater is designed on the failure mechanisms presented in Subsection 4.3.2, also see Figure 4.5 for the specific integration of the failure mechanisms. When these specific values are used three variants are generated, see Figure 4.6, this is because the size of the armour units allows for two possible rock gradings for the underlayer. An additional variant is generated with an additional filter layer between the underlayer and core. The computed D_{n50} and class of these three variants are presented in Table 4.5. The computed freeboard, R_c, is 4.366 m and is the same for all variants.

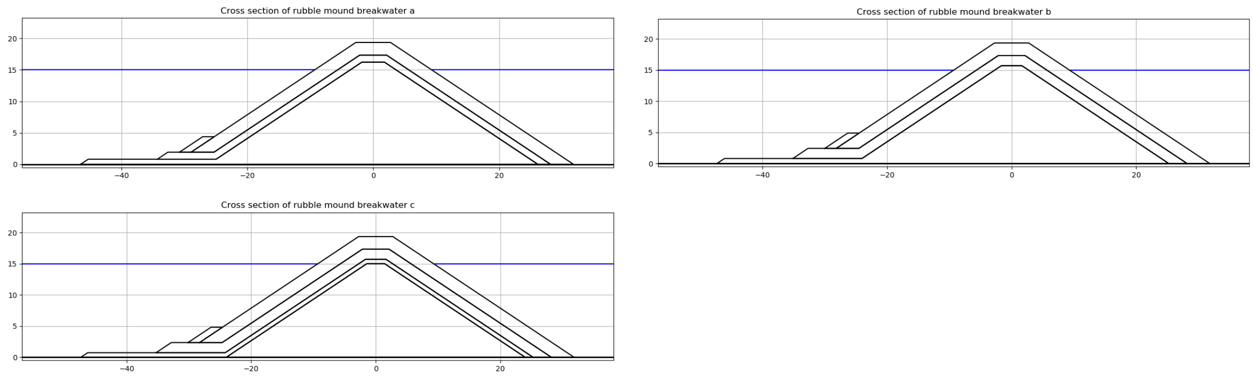


Figure 4.6: Cross section of the generated variants with the code from Code Snippet 4.1

Table 4.5: Computed values for the three variants

	Computed D_{n50} [m]			Class		
Variant	**a**	**b**	**c**	**a**	**b**	**c**
Armour	1.385	1.385	1.385	3 m^3	3 m^3	3 m^3
Underlayer	0.566	0.768	0.768	HMA 300/1000	HMA 1000/3000	HMA 1000/3000
Filter layer			0.415			LMA 60/300
Toe	0.6	0.6	0.6	HMA 300/1000	HMA 300/1000	HMA 300/1000

4.4.2 *Using the Configurations Class*

The Configurations class can be used to design multiple breakwater concepts at the same time, as was explained in Subsection 4.3.4 In Code Snippet 4.2 concepts for conventional rubble mound breakwaters with rock (RRM) and armour units (CRM) are generated. Furthermore, besides generating concepts for two breakwater types three parameters are specified as varying parameters, these are: the front slope, the crest width and the nominal diameter of the core. Each of these parameters is specified as a tuple[3] with a lower and upper bound, first and second value, and the number of equally spaced numbers to generate. Note that this example is also available on the GitHub repository as *example3.py*.

```python
import breakwater as bw

# compute wave heights not already known
battjes = bw.BattjesGroenendijk(Hm0=4.4, h=15, slope_foreshore=(1,100))
H2_per = battjes.get_Hp(0.02)

# define a limit state with hydraulic parameters, and the allowed damage
ULS = bw.LimitState(
    h=15, Hs=4.5, Hm0=4.4, H2_per=H2_per, Tp=9.4, Tm=8.8, T_m_min_1=9.7,
    Sd=5, Nod=2, q=20, label='ULS')

# define the materials for the armour layer
NEN = bw.RockGrading()
xbloc = bw.Xbloc()

# design multiple configurations of RRM and CRM
configs = bw.Configurations(
    structure=['RRM', 'CRM'], LimitState=ULS, rho_w=1025,
    slope_foreshore=(1,100), Grading=NEN, slope=((1,3), (3,4), 4), B=(5, 8, 4),
    Dn50_core=(0.2, 0.4, 3), N=2100, ArmourUnit=xbloc)
```

Code Snippet 4.2: Code to design with the Configurations class

[3]a tuple is a collection of ordered and unchangeable data. In Python tuples are specified with round brackets

The code of Code Snippet 4.2 generates 96 breakwater concepts, 4*4*3=48 concepts for each breakwater type. The values of each concept are stored inside the Configurations class. Since no automatic selection method is used it is up to the designer to determine the method he wants to use for choosing the optimal concept. The concepts can, for instance, be exported to the Design Explorer, see Subsection 2.2.2, of which the result is presented in Figure 4.7. Note that in the figure only a selection of parameters have been exported. Alternatively, the concepts can also be exported to Excel which generates an Excel sheet with all computed values and input.

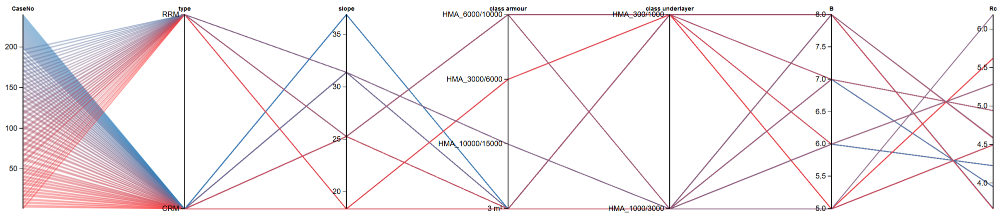


Figure 4.7: Visualisation of the concepts generated with Code Snippet 4.2 in the Design Explorer

The attentive reader might have spotted that more than 200 CaseNo (case numbers) are present in Figure 4.7, instead of the expected 96 concepts. However, this is because one concept can contain up to four variants, a variant is defined as two concepts having the same armour layer and crest height but a different underlayer and/or filter layer. That one concept can result in more variants was also observed in the previous subsection.

4.4.3 *Using the Interactive Design Application*

In this subsection, the interactive design application is used to design concept for two different breakwater types. The code required to start the interactive design application, the GUI[4], is presented in Code Snippet 4.3. Note that the same input and types of breakwaters are specified as in the example with the Configurations class from the previous subsection.

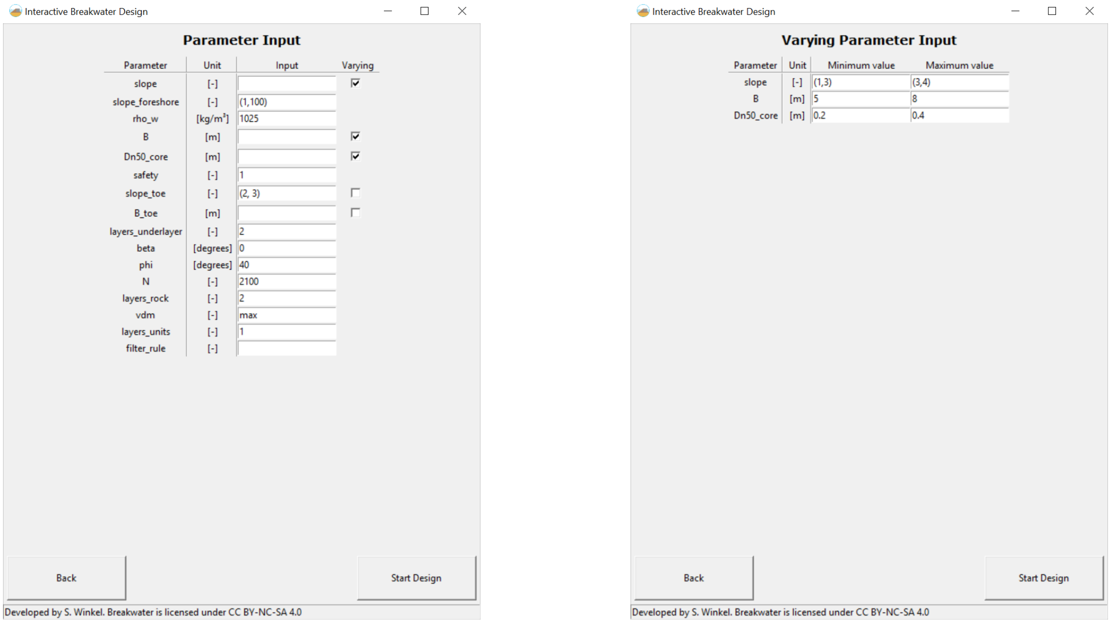
```python
import breakwater as bw

# compute wave heights not already known
battjes = bw.BattjesGroenendijk(Hm0=4.4, h=15, slope_foreshore=(1,100))
H2_per = battjes.get_Hp(0.02)

# define a limit state with hydraulic parameters, and the allowed damage
ULS = bw.LimitState(
    h=15, Hs=4.5, Hm0=4.4, H2_per=H2_per, Tp=9.4, Tm=8.8, T_m_min_1=9.7,
    Sd=5, Nod=2, q=20, label='ULS')

# define a material for the armour layer
xbloc = bw.Xbloc()
NEN = bw.RockGrading()

# define the soil
soil = bw.Soil(c=0, phi=30, gamma=17)
soil.saturated_weight(gamma_sat=21)

# start the interactive design application
bw.interactive_design(
    structure=['RRM', 'CRM'], LimitState=ULS, Grading=NEN, ArmourUnit=xbloc,
    Soil=soil)
```

Code Snippet 4.3: Code to start the interactive design application

The code from Code Snippet 4.3 starts the GUI. After the start screen, the parameters for the specified structures can be specified, see Figure 4.8a. Furthermore, by checking a checkbox, parameters can also be set as varying parameters. When the *Start Design* button is pressed and a checkbox is checked an additional input screen is

[4]Graphical User Interface

shown to the designer where the values of the varying parameters must be specified, see Figure 4.8b. Note that with the interactive design application only an upper and lower bound value are required.



(a) Screenshot of the input screen (b) Screenshot of the input of the varying parameters

Figure 4.8: Specifying the parameters with the GUI

When all required parameters have been specified the *Start Design* can, again, be pressed. With the press of a button a concept for each of the specified structures is generated, see Figure 4.9. New concepts can be generated by changing the values of the varying parameters with the sliders, and subsequently pressing the *Update Design* button. This generates a new concept with the old concept plotted behind the new one with grey dashed lines so that the designer can clearly see the differences and similarities.

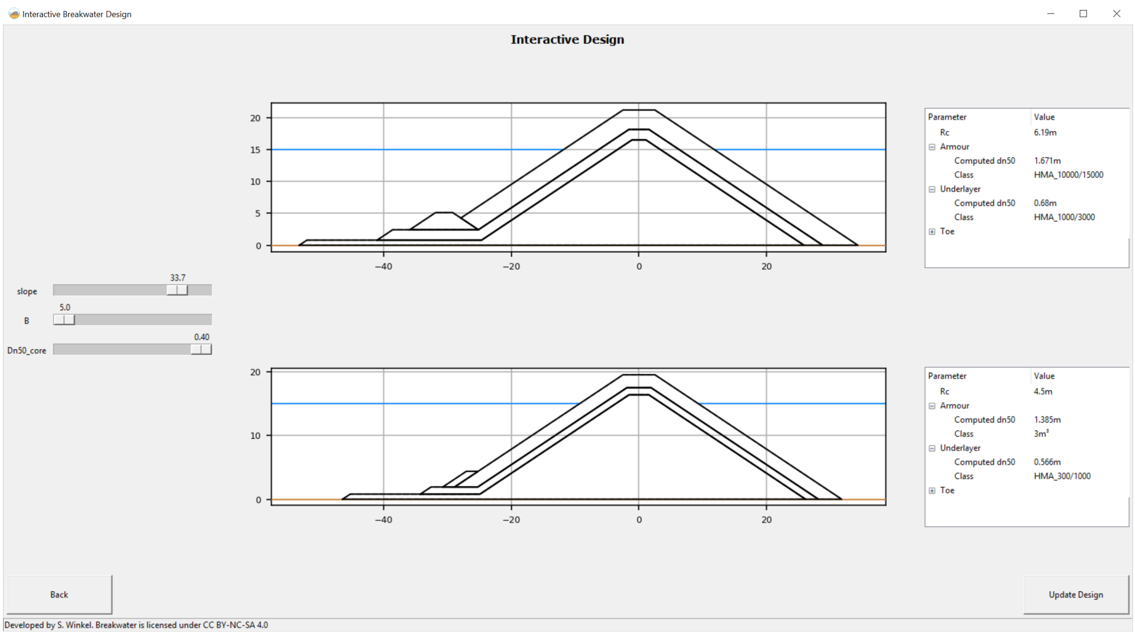


Figure 4.9: Screenshot of the GUI with a concept for a conventional rubble mound breakwater with rock (top) and armour units (bottom).

4.5 Concluding Remarks

From the analysis of existing breakwater design automation tools, it appeared that none of the tools fulfilled all requirements. More importantly, none of the described tools were open-source, and thus not available for usage. A new design automation tool that meets all requirements was therefore developed in Python. The tool incorporates three breakwater types, the conventional rubble mound and the (composite) vertical breakwater. Failure mechanisms due to hydraulic loading are implemented for both types, as well as one geotechnical failure mechanisms for each type. This is because the geotechnical stability was frequently mentioned during the interviews, Subsection 3.3, as an important consideration.

The design philosophy during the development was to limit the designer as little as possible. Therefore, no automatic optimisation algorithm was implemented. The tool does, thus, not automatically choose the optimal breakwater, neither does it design the optimal breakwater. Designs are only generated when the designer requests to make a design or specifies a range within concepts must be generated. This design philosophy was chosen because it appeared during the interviews, and literature study, that experience is an important factor during the design process. Because experience was also mentioned as an important method to select the most promising concept, Table 3.2, no predefined selection method was implemented. However, it is possible to export the generated breakwaters to the design explorer or make a multi-criteria analysis, the latter can be done in Excel or Python.

The design automation tool developed in this chapter was developed to be open-source and is part of the Python package called *breakwater*, of which the initial version was developed during this research. The source code is therefore freely available under the CC-BY-NC-SA 4.0 license[5] through the following sources:

- On the Python Packaging Index under the name *breakwater*: https://pypi.org/project/breakwater/
- On GitHub in the repository called *breakwater*: https://github.com/Sander-w/breakwater

A full documentation with all features is available on https://breakwater.readthedocs.io.

Figure 4.10: Logo of *breakwater*, the Python package developed for this research

[5]See https://creativecommons.org/licenses/by-nc-sa/4.0/ for a full version of the license

5

Verification and Validation of the Developed Tool

In this chapter, the verification and validation of the design automation tool, developed in Section 4.3, is discussed. This is done by inspection of the documentation and code, but also by demonstrating the tool to experienced breakwater designers. The design automation tool has been demonstrated to Interviewee A and B, see Table 3.1. Note that the tool has not been presented to Interviewee C as this designer is mainly focused on making conceptual designs for rubble mound breakwaters with armour units as armour layer.

First, in Section 5.1, the verification is performed. The verification aims to answer the question: has the design automation tool been developed in accordance with the requirements? Whereafter, in Section 5.2, the validation of the tool is discussed, which aims to answer the question: does the developed design automation tool fulfil the designer's operational needs? Then, in the final section of this chapter, the concluding remarks from the verification and validation are discussed.

Note that while this chapter describes the process of verification and validation as a sequential step in the development process, taking place after the tool has been developed, it was actually performed during the development of the design automation tool. This is because verification and validation must be performed continuously throughout the development phase (Wason, 2015).

5.1 Verification

The verification aims to answer the question if the developed design automation tool fulfils the requirements, as mentioned in the introduction of this chapter. The verification has therefore been divided into two subsections. First, in Subsection 5.1.1, the internal consistency of the tool is verified. This is, to verify if all the equations are correctly implemented, and indeed return the expected results. Then, in Subsection 5.1.2, the requirements from Section 4.1 are verified. Finally, in the last subsection, the conclusion of the verification is presented.

5.1.1 *Internal Consistency*

To verify if all implemented equations are correctly implemented, and indeed produce the expected results, test functions are written in Python. These test functions are written with the *unittest* module. The *unittest* module is part of the python standard library and is a unit testing framework. The module supports test automation, which means that functions and methods[1] can be called with constant values. The result of this call is evaluated against the expected result, which is the result of a hand computation or example. An example of a successful test can be seen in Code Snippet 5.1.

[1]In Python a function is an executable statement, when a function is part of a class is it referred to as a method instead of a function

Code Snippet 5.1: Result of a succesfull unittest

```
......
----------------------------------------------------------------------
Ran 6 tests in 0.001s

OK
```

In Code Snippet 5.1 it can be seen that each of the 6 tests was successful, meaning that all functions returned the expected results. As a result, it can thus be concluded that these functions have correctly been implemented. On the other hand, in Code Snippet 5.2, the result of a failed test is presented.

Code Snippet 5.2: Result of a failed unittest

```
...F..
======================================================================
FAIL: test_vandermeer_deep (__main__.TestRock)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "test_stability.py", line 25, in test_vandermeer_deep
    self.assertAlmostEqual(Dn50, 1.36, 2)
AssertionError: 1.2592191697321402 != 1.36 within 2 places (0.10078083026785989 difference)

----------------------------------------------------------------------
Ran 6 tests in 0.002s

FAILED (failures=1)
```

From this result it can be seen that one of the six test cases have failed, to be precise the test function of the deep water equation of the Van der Meer formula has failed. Furthermore, it can be seen that the test failed due to an *AssertionError*, caused by the fact that value from the called function is about 0.1 lower than the expected value. With the aid of this test case, the cause of the error is known, and one can thus immediately investigate the function with the deep water Van der Meer formula.

Throughout the development of the tool test cases have been written to verify if the equation was correctly implemented. This was done in the test and review step of the development of the design automation tool, see highlighted steps in Figure 5.1. From this figure, it can also be seen that the verification of the internal consistency has indeed continuously been verified. It must, therefore, be remarked that the failed test case of Code Snippet 5.2 merely served as an example. In fact, when all test cases are executed, none of them produce an error.

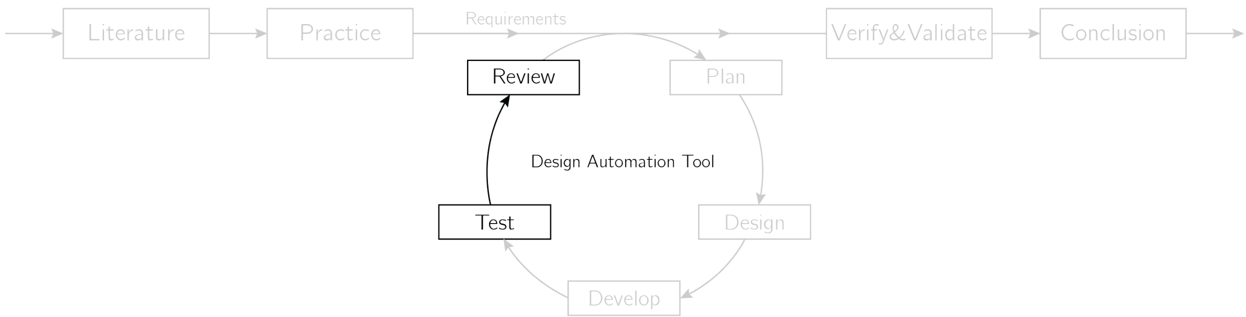


Figure 5.1: Methodology used in this research with the steps to verify the internal consistency highlighted

5.1.2 *Requirements*

F.01 **The design automation tool shall include at least one rubble mound and one monolithic type breakwater**
In Chapter 7 of the documentation, breakwater types, it can be seen that the design automation tool includes a conventional rubble mound breakwater with rock and armour units as armour layer, and a vertical (composite) breakwater.

F.02 **The design automation tool shall be able to design multiple concepts**
In Chapter 7 of the documentation, breakwater types, it can be seen that several parameters can be

changed in the design classes, although default values are specified for some parameters. Designing with different values results in another concept.

F.03 **The design automation tool shall be able to design a breakwater within 1 second**
Part of the python standard library is the *timeit* module, with this module a piece of python code can be timed for its duration. This module is used to verify if the design classes are able to design a breakwater within 1 second. The computation is performed 10,000 times for each design class, each time with the same set of parameters. The average time it took to compute a design was:

- RockRubbleMound: 0.093s
- ConcreteRubbleMound: 0.14s
- Caisson: 0.001s

Note: the rubble mound design classes are slower because iteratively computing multiple slip circles for Bishop's method is computationally expensive, and the fact that a slip circle analysis is not needed for the caisson design class.

F.04 **The design automation tool shall include construction cost**
In Chapter 7 of the documentation, breakwater types, it can be seen that all design classes have a method to add construction cost per cubic meter of material.

F.05 **The design automation shall include the availability of equipment/material**
In Chapter 7 of the documentation, breakwater types, it can be seen that it is possible to specify a transport cost argument in the cost method which makes it possible to include the proximity of a quarry. The caisson class has an investment method for adding the rent of a dry dock or any other required investment.

F.06 **The designer shall be able to choose the optimal concept**
From an inspection of the code, it can be concluded that concepts are not filtered with a selection method. Furthermore, no optimisation process, or criteria, are implemented in the code. It is possible to use the DesignExplorer or make a multi-criteria analysis, but this must be requested by the designer. The designer is therefore always able to choose the optimal concept.

F.07 **The design automation tool shall include hydraulic failure mechanisms**
In Chapter 8, 9 and 10 of the documentation it can be seen that the tool includes the following hydraulic failure mechanisms:

- Overtopping equation for the rubble mound breakwater and all equations from Figure 4.4 for monolithic breakwaters
- Erosion or breakage of the armour for rock and armour units
- Erosion of the toe of a rubble mound breakwater
- The wave pressure model of Goda-Takahashi
- Erosion or breakage of the armour of the foundation of a caisson breakwater

F.08 **The design automation tool shall include geotechnical failure mechanisms**
In Chapter 11 of the documentation, geotechnical stability, it can be seen that the tool includes the following geotechnical failure mechanisms:

- Brinch Hansen (1970) to compute the bearing capacity of the subsoil, implemented for vertical (composite) breakwaters
- Bishop's method to determine the factor of safety against a slip circle through the outer slope of the breakwater, implemented for the rubble mound breakwater

A.01 **The design automation tool shall be easy to expand**
The design automation tool consists of several modules that consist of functions and classes with the failure mechanisms from Subsection 4.3.2 and 4.3.3. New failure mechanisms can be added by adding a new module in the core and subsequently implementing the functions of the new module in the design

classes. Changes are then automatically made to the design automation features as these features call the design classes directly.

A.02 **The design automation tool shall be open-source**
The design automation tool can be downloaded from the GitHub repository, or installed from PyPI with the following command: pip install breakwater.

O.01 **All parameters of the implemented equations not having a constant value shall be allowed to be changed**
From the inspection of the functions and methods in the core, it can be concluded that all parameters can be specified with another value, apart from the gravitational acceleration, which is set to a fixed value of 9.81 and can thus not be changed. However, in the design classes, some values are fixed, for instance, the notional permeability in van der Meer (1988). This is due to design choices made in the design classes, see Subsection 4.3.5. However, in the core, all parameters can be changed so this requirement is fulfilled.

O.02 **The design automation tool shall include a documentation**
The documentation of the tool is available on https://breakwater.readthedocs.io.

I.01 **The design automation shall be able to be used in the breakwater design process**
The interactive design application of the design automation tool has been presented to the two interviewees. Both interviewees stated that they would be able to use the design automation tool in their design approach.

I.02 **The design automation tool shall be able to load input from, and export data to, an Excel sheet.**
In Chapter 12 of the documentation it can be seen that there is a function to create, and load, an Excel input file. Exporting data to Excel is also possible as the data in the Configurations class is stored in a DataFrame, which can be exported to Excel.

P.01 **The design automation tool shall be developed in accordance with PEP8**
From the inspection of the code on GitHub it can be seen that the code is developed in accordance with PEP8.

P.02 **The design automation tool shall be developed in accordance with PEP287**
From the inspection of the code on GitHub it can be seen that the docstrings in the code are in accordance with PEP287.

5.1.3 *Conclusion of the Verification*

The goal of this section was to verify the developed tool, and the verification question can therefore now be answered.

Has the design automation tool been developed in accordance with the requirements?
From the verification of the requirements, performed in Subsection 5.1.2, it can be concluded that the design automation tool fulfils all requirements listed in Section 4.1. Furthermore, all implemented equations have been verified by calling the corresponding function and comparing the outcome with examples from literature and design manuals. The conclusion can thus be drawn that the design automation tool has been developed in accordance with the requirements.

5.2 Validation: Opinion of Experienced Designers

The validation aims to answer the question if the developed design automation tool fulfils the designers operational need. The operational need can be inferred from the objective, the designer wants a design tool that can quickly design different types of breakwaters so that more concepts can be explored. To validate if the operational need is fulfilled by the developed design automation tool, interviews have been conducted with experienced breakwater designers, as was already mentioned in the introduction of this chapter.

The first subsection, therefore, elaborates on the structure of the interviews. Hereafter, in Subsection 5.2.2, general remarks from the interviewees about the tool are presented. This subsection is followed by a subsection about how the interviewed designers plan to use the tool in practice. During the interviews several suggestions have been made to further improve the tool, these suggestions are presented in Subsection 5.2.4. Finally, in the last subsection, the conclusion of the validation is presented.

5.2.1 *Structure of the Interviews for the Validation*

The structure of the interviews conducted for the validation is different from the structure of the interviews conducted in Chapter 2. The reason a different structure is used is because this time the interviewees are asked about their opinion. Furthermore, the interviewer wants to steer the conversation as little as possible so that the interviewee can give his honest opinion. It is thus important to prevent a bias in the questions. The question *are you able to use the tool in early design phases?* must thus be asked in two parts:

- First, what is your opinion about the design automation tool and try to infer based on the answers if the interviewee wants to use the tool
- Secondly the question must be asked in which phase the designer plans to use the tool, if the interviewee wants to use the tool

Therefore, the unstructured interview method is used instead of the semi-structured. This because the questions asked during the interview all depend on the opinion of the designer, and, can thus, not be determined beforehand. The type of interview and type of questions have, however, not changed compared to Chapter 2, meaning that also for these interviews an interview survey with open-ended questions is used.

During the interviews the interactive design application of the design automation tool was presented to the designers, see Figure 5.2, since using the application requires almost no knowledge of Python.

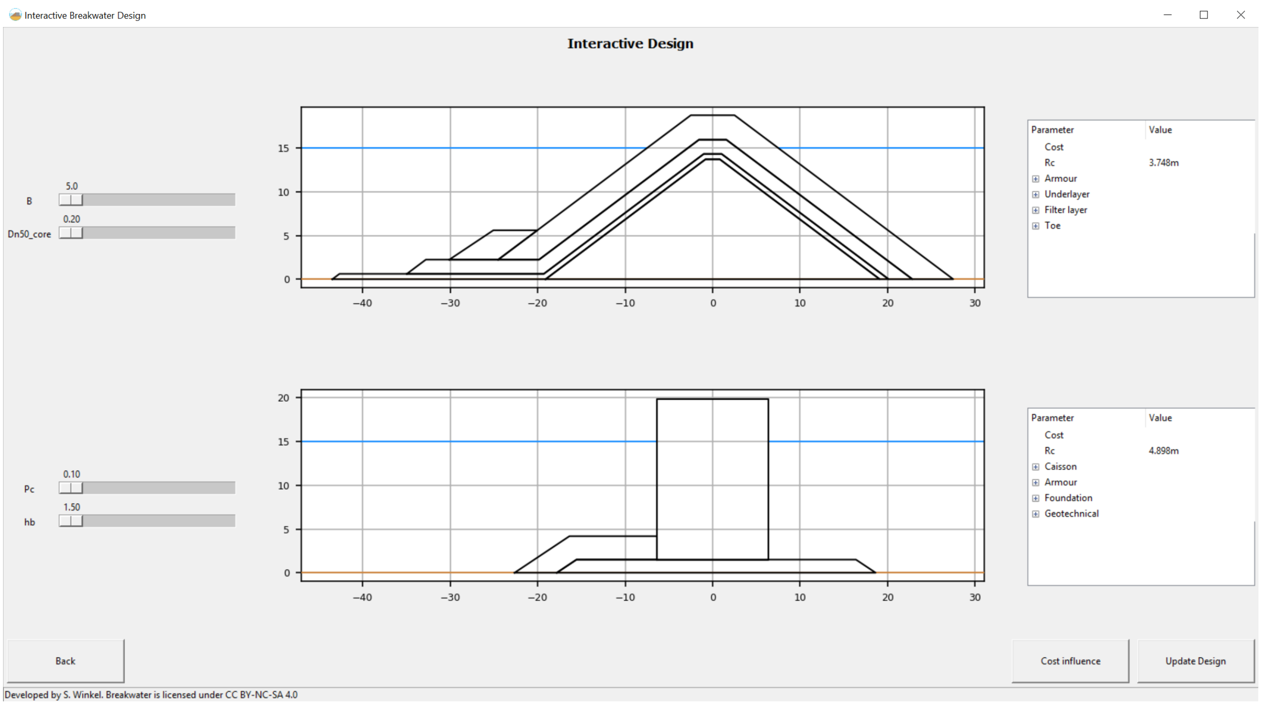


Figure 5.2: Screenshot of the interactive design application

5.2.2 *General Remarks*

Both interviewees stated that they see the added value of the design automation tool. Especially the speed at which it is possible to designs several different concepts, and breakwater types, is mentioned as a significant improvement over the Excel sheets. One of the main advantages according to Interviewee B is that it is not possible for users to change the code, which eliminates the main problem of Excel sheets, that each user can make changes to a sheet. Whereas, changes to the design automation tool can only be made by developers, who need to push their changes for approval to the GitHub repository. New versions only become available after a pull request[2] is approved.

Furthermore, a GUI is much clearer than an Excel sheet according to the interviewees. Both, therefore, state that the GUI is an important part of the design automation tool. Interviewee B also remarks that the usability of a tool significantly increases when a GUI is made. However, Interviewee A states that everything must be clear at first sight for a GUI to be good and usable. He states that this is currently not completely the case, because the definition of the parameters is not completely clear. It is therefore suggested to include a figure with the definition of each parameter. More improvements were suggested during the interviews, these are further discussed in Subsection 5.2.4.

5.2.3 *Using the Tool in the Practice*

When the interviewees were asked if they would use the tool, both confirmed that they would be able to use the tool in practice. They would use the tool early in the design process to quickly assess the feasibility of several variants and breakwater types. Interviewee A remarked that he would use the tool to assess the feasibility of a rubble mound and caisson breakwater. Based on the generated designs he would then decide whether to propose a design for a rubble mound, caisson breakwater, or both.

The tool is, however, less applicable for the later design phases. According to interviewee A, this is because later design stages are more specific, and thus require more detailed computations, which are not implemented in the current version of the design automation tool. Interviewee B also states that he would not use the tool in later design, because, generally, no large changes are made to the design.

Besides being able to quickly assess the feasibility of different breakwater types both interviewees also stated that the tool enabled them to assess the influence of certain parameters on the design and cost. The influence on the design can be assessed by changing the value of the parameters with the sliders, which results in a new design when the *update design* button is pressed. This feature is especially useful as the interviewees reported to be able to vary a large number of parameters. By pressing the *cost influence* button a plot is generated in which the influence of the parameter on the design is visualised, see Figure 5.3

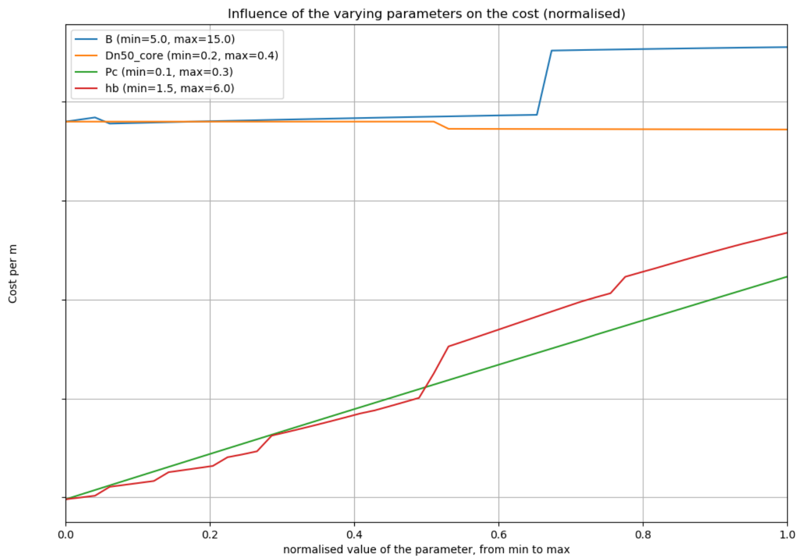


Figure 5.3: Example of a plot to visualise the influence of each of the varying parameters on the cost

However, the interviewees also reported the same disadvantage of the tool, which is that laymen are able to use the tool. These laymen are however not always able to critically assess the generated design for feasibility, or

[2] Pull requests inform other developers about changes made to the code and allow developers to verify and discuss changes before approving them

give realistic input. This because the tool will always generate a design, despite the input. It is, for instance, possible to design a breakwater in a water depth of 10 m and a significant wave height of 20 m. The interviewees thus remark that the same holds for this tool, as for any tool, rubbish in is rubbish out.

Finally, it was remarked by Interviewee A that he would like to overrule the tool on the computed values. As an example, he provides the case where he would like to specify the D_{n50} of the armour layer, and then let the tool compute the required values to satisfy the specified D_{n50}. The possibility to overrule computed values is thus a new requirement the tool must fulfil. However, while it is currently a disadvantage, as it limits the freedom of the designer, the interviewee also states that inability to overrule computed values is logical for a first version.

5.2.4 *Suggested Improvements*

During the interviews both interviewees frequently suggested improvements. The suggested improvements are:

- Possibility to specify the freeboard and compute the overtopping discharge, instead of only being able to specify overtopping discharge and computing the required freeboard
- Computing the wave transmission
- Add more geotechnical failure mechanisms
- Improve graphical representation of the structure by, for instance, adding the significant wave height as a sine wave and visualise the percentage of concrete and fill material
- Add a clear definition sketch of the parameters to the GUI
- Divide input screens into three screens instead of one. Screen one can then be the hydraulic input, screen two the geotechnical input, and screen three the safety factors
- Add the following parameters to the infobox (box at the right of the plot):
  - Volume of material per layer in m^3
  - Number of rows required for armour units
- Possibility to overrule all parameters, for instance, specify the grading to use for the armour layer.

5.2.5 *Conclusion of the Validation*

The goal of this section was to validate the developed tool by interviewing experienced breakwater designers. The main question of the validation step can therefore now be answered.

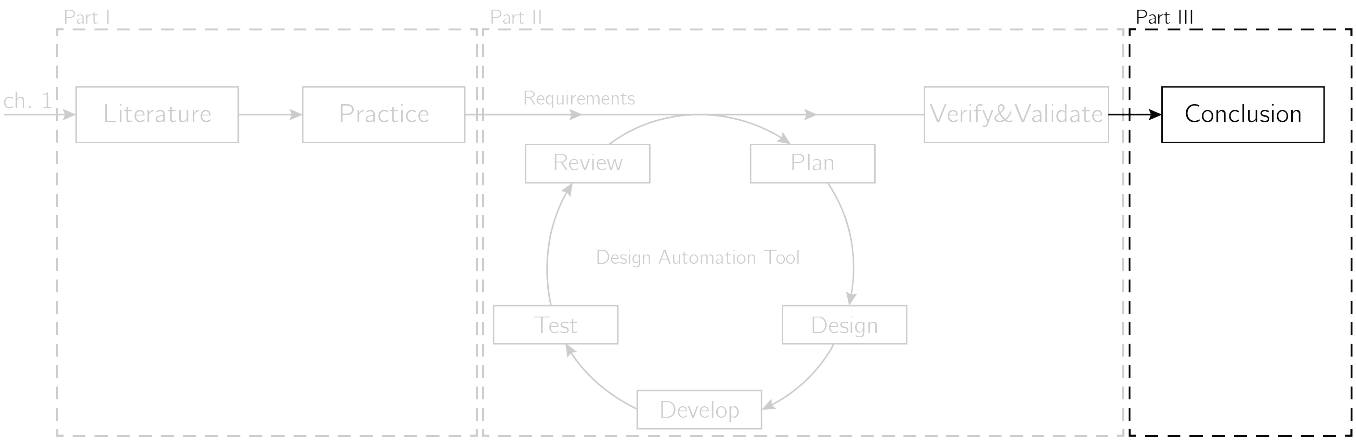
Does the developed design automation tool fulfil the designer's operational needs?
The operational need of the designer is to have a design automation tool that can quickly design several different concepts and breakwater types. From the interviews it can be concluded that the interviewed designers indeed see the added value of the developed tool, and also want to use the developed tool in their own design process. Both interviewees stated that they would use the tool to quickly design breakwaters to assess the feasibility of several breakwater types. With the tool designer are able to explore more feasible concepts in the same period. Furthermore, designers also reported being able to see the influence of parameters on the cost and design, which results in a better understanding of the design. It can therefore be concluded that the design automation tool does fulfil the operational need of the designer.

5.3 Concluding Remarks

It can be concluded that the developed design automation tool fulfils the requirements and the operational need of the designers. The tool is useful for assessing the feasibility of different breakwater types and can be used by designers in practice. Furthermore, the experience of the designer is involved in the process by not automatically selecting an optimal concept. Instead, the tool only generates concepts based on designer input and at the request of the designer. No preselection is made, and the tool presents all generated concepts to the designer, the way the concepts are filtered is up to the designer.

However, it must be remarked that the GUI of the design automation tool does not provide all design freedom due to design choices made in the design classes. Furthermore, the tool cannot be overruled on computed parameters, which is wanted by the designer to increase the freedom. This is a new requirement the tool must satisfy, but this only became apparent during the interviews conducted for the validation. It must be remarked here that the fact a new requirement was encountered highlights the importance of continuously iterating over the design space since each iteration will result in additional knowledge.

III Conclusion



6 Discussion

The validation suggests that the developed design automation tool can be used in the breakwater design process to quickly design several different breakwater types and assess their feasibility in early design phases. This chapter discusses the developed tool by first discussing the added value of the developed tool in the design process. Hereafter, in Section 6.2, the relation between the results of this research and literature are discussed. In the final section of this chapter, the limitations of this research are presented.

6.1 Added Value of the Design Automation Tool

The productivity growth of the construction industry is lagging behind the overall economy, as was described in Subsection 1.2.2. To increase productivity growth, the efficiency of the construction industry must increase by, for instance, rethinking design process and incorporating new digital tools. This enables designers to explore more feasible concepts, which results in a more optimal design. In current design practice not all feasible concepts can be explored because of time constraints, and concepts are disregarded based on assumptions, as was illustrated in Section 1.1.

The current breakwater design approach, Figure 3.2, is an iterative approach with sequential phases in which the design is further refined. During the conceptual design phase several concepts are explored, and the feasibility of different breakwater types is assessed. The interviewed designers describe that concepts are developed in an iterative process, making the process well suited for automation. However, both engineering judgement and experience of the designer are mentioned as important methods to select concepts, which cannot, or are hard to automate.

An important factor of influence on the practical design approach is the design culture. The design approach of designers is likely to be influenced by their country of origin, but also the type of company, and country of origin of the company. In this research only three designers have been interviewed, all from The Netherlands. This can result in a bias towards Dutch design culture, but this has not been investigated. Furthermore, there is a bias towards the design culture of contractors, which is likely to be focused on practical considerations such as constructability. This can also be observed in Chapter 3 by considering the comments about cost and constructability which were mainly made by Interviewee B.

To add value to the breakwater design approach, the tool must combine the strong points of automation and the designer. The strong points of automation are repeatedly performing the same task(s) without mistakes, assuming a correct implementation, and the fact that automation can perform computations quicker than designers. However, the disadvantage of automation is to assess and interpret the generated concepts[1], but this is a strong point of designers because the experience of designers enables them to quickly assess concepts, as was described in Section 2.2.

The importance of experience was also observed during the validation, Section 5.2, where the interviewees assessed the generated concepts within seconds. An example of this process is that one of the interviewees immediately assessed the feasibility of a caisson breakwater by assessing the width to height ratio of the caisson, and concluded with the following statement: *"the width seems to be about $\frac{2}{3}$ of the height, so it seems okay."*.

[1]Disregarding techniques as Machine Learning, because these require a large amount of data which is currently not available

Both designers stated that they would use the developed tool in their own design process for assessing the feasibility of different breakwater types. Furthermore, they state that the tool makes it possible to quickly assess, and visualise, the influence of different parameters on the design and cost.

An unexpected result was that probabilistic design approaches are not used in practice, whereas these are taught to students, and implemented in recently developed automation tools. The main reason is that for most projects it is not possible to determine the correlation between wave height and water depth and that the uncertainty of some equations are hard to determine. This lack of data causes designers to make conservative estimates for the missing data and the uncertainties, which results in a too expensive design when using probabilistic approaches. Another unexpected result, which became apparent during the development of the tool, is that more implicit design choices are made by designers than described during the interviews.

6.2 Relation of this Research with Literature

The results of the validation phase show similar results as the validation conducted by Cederfeldt (2007). He also concluded, from interviews, that evaluating more concepts was seen by designers as an advantage. The main advantage mentioned by designers is to be able to perform what-if scenarios, or in other words, see how certain parameters influence the cost and design. Davila Delgado and Hofmeyer (2013) also showed that a design automation tool capable of quickly designing concepts benefits the design process. These benefits of design automation have further been highlighted in this research.

Furthermore, it was shown in Section 1.1 that familiar solutions are more often selected, as was also shown by Ford and Gioia (2000) in their experimental study about decision making. The fact that the bias was inadvertent was shown by the interviews conducted in Section 5.2, where designers reported to use the tool for assessing the feasibility of different breakwater types. This confirms the conclusion of Rietzschel et al. (2010) that the bias towards familiar concepts is often inadvertent and can be overcome by presenting unfamiliar concepts, as Nijstad et al. (2002) reported.

To prevent this bias Toh and Miller (2015) and Dorst and Cross (2001) argued that the continuous exploration of concepts is important since this results in more diverse concepts, which, on its turn, leads to a more optimal design. For the design of a subsystem of the Afsluitdijk, concepts were continuously explored and systematically generated with a morphological chart. Labouchere (2018) reported that the design team got a better understanding of the design, and thus the influence of parameters. Interviewed designers acknowledged that they are better able to assess the feasibility of designs and the influence of parameters with the developed tool because more concepts can be explored.

The importance of continuously iterating can also be seen in the design methods of Roozenburg and Eekels (1995), Hertogh and Bosch-Rekveldt (2013), and Wason (2015), because all methods explicitly incorporate an iterative process in which concepts are generated. However, when the iterative behaviour is not explicitly included, it does not mean that iterations are not done by designers. This is because De Groot (1994) argued that this iterative behaviour is intrinsically linked to the way humans gain knowledge. That iterations can be performed implicitly was shown by the breakwater design approach. Although designers did not explicitly mention the iterations, it could be deduced from their description of the conceptual design phase. This also confirms the statement of Roozenburg and Eekels (1995) that their design method is fundamental.

Another implicit process in the breakwater design process are the design choices. During the development of the design classes, several design choices had to be made that were not mentioned during the interviews. Both the implicit performance of iterations and the implicit design choices prove the point of Lawson and Dorst (2005) that experience plays an important role in design processes because designers automatically perform the required actions. Moreover, engineering judgement and experience were also mentioned as important methods to assess the feasibility of concepts and for the selection of the preferred concept, which further highlights the point of Lawson and Dorst (2005).

Where the interviews and Lawson and Dorst (2005) indicate that it is important to include the experience of the designer in the process and let the designer choose the optimal concept, some existing models automatically optimise the concept. The tools of Sijbesma (2019) and Laenen (2000) both design a cost-optimal breakwater with a probabilistic design approach. The interviewed designers stated, however, that probabilistic approaches are not used in practice. This is because, according to the interviewees, a probabilistic design results in a design that is too conservative, and will thus be more expensive, which increases the probability of losing a tender. The fact that a probabilistic design is more expensive is also acknowledged by Sijbesma (2019). However, the exclusion of a probabilistic design approach does not mean that the uncertainties are not assessed. On the contrary, the breakwater design approach includes a cycle where model tests are performed in the final design phase to assess the safety of the design, or in other words, to investigate the uncertainties.

The tool developed by De Smet (2015) does not include an optimisation algorithm or method. Moreover, he also stated that it is up to the designer to decide if the designed concept is the optimal design or to design another concept. However, his tool is limited as certain choices are made automatically. For example, when choosing for a rubble mound breakwater with armour units as armour layer, the slope cannot be changed, limiting the applicability of the tool.

When more concepts are explored, the influence of parameters on the design and cost are better understood, as designers remarked in Section 5.2. With this increased understanding, a more optimal design is made, which increases the efficiency of designers. This proves the point of McKinsey&Company (2017) and Deloitte (2019) that the productivity and efficiency of the construction industry can be increased by rethinking design processes and using digital technologies.

To summarise, the findings of this research are largely in line with literature, apart from the application of probabilistic design approaches. During interviews, it became apparent that these design approaches are not, or hardly, used in practice. Furthermore, this research also showed that it is hard to study the way concepts are generated as designers frequently draw to explain their thought process. Additionally, more design choices had to be made during the development of the tool than explained by designers during the interviews.

6.3 Limitations of this Research

This research has the following limitations:

- All literature regarding the actions of designers, and how a designer thinks, are based on other research fields, for instance, product design or managerial decision processes. In this research, the assumption has been made that the decision process of product designers and managers is comparable to how breakwater designers consider different variants.
- Due to the limited development time, several design choices had to be made in the design classes, which limits the freedom of the designer. These limitations have been described in Subsection 4.3.5.
- The developed tool requires some basic knowledge about Python, which limits the use of the tool to designers who are familiar with Python. This also holds for the developed GUI because the GUI was not developed as a standalone tool.
- The practical design approach, described in Chapter 3, has been based on three interviews. Furthermore, two designers work for the same company, which makes it likely that their processes are similar.
- Only two interviews have been conducted for the validation of the tool, which is a limited number of interviews.
- The added benefit of the developed design automation tool has not been quantified.
- The developed design automation tool has not been verified and validated in a real design process.
- The verification of the implemented geotechnical equations is based on examples from the course Soil Mechanics (CTB2310) taught at the TU Delft. These examples are all performed with dry soil and a water level that is well below the soil.
- The tool was developed with the philosophy that the experience of the designer must be incorporated. This resulted in a tool that leaves certain decisions to the designer, which makes the tool not suitable for designers who have no background in hydraulic engineering.
- During the interviews, it became apparent that some of the design errors are the result of an incorrect translation from the 2D cross-section to a 3D design. Errors are often made due to an incorrect

computation of the average depth. It must be remarked that the developed tool only designs a 2D cros-section, and not a 3D breakwater. While it is possible to define several cross-sections and then use each cross-section to make a conceptual design for a 3D breakwater, it does not prevent the error in average depth computation since the designer still needs to code this part himself.

- Literature indicates that when it is easier to consider other and more diverse, concepts, this is also done. While the interviews of the validation suggest that designers will use the tool to consider all implemented structures, it has not been proven during a real design process.
- While it is possible to design several breakwater types with the developed tool, it does not mean that designers are actually going to consider these other breakwater types. This means that conservative assumptions might still be made, and that familiar solutions are chosen without the consideration of unfamiliar solutions. However, since designing is a human activity, this will always be an intrinsic part of a design process. On its own this is not a problem, as long as designers are aware of their bias and keep an open mind to unfamiliar solutions.
- The breakwater database was used in this research to analyse the hydraulic conditions in which breakwaters are constructed, but this research showed that these are not the only selection criteria. The other criteria, such as proximity of quarries, could however not be investigated due to a lack of data.
- The developed design automation tool does not include all breakwater types defined in Figure 1.1. The inadvertent bias towards breakwater types that have not been implemented is therefore still present when the design automation tool is used.

7
Conclusions

The objective of this research, formulated in Subsection 1.4.1, was to improve the breakwater design process with a design automation tool, which is able to quickly generate and verify rubble mound and caisson breakwaters during the design process.

However, before a design automation tool can be developed, one must first understand the practical design approach. Experienced breakwater designers were interviewed to gain a deeper understanding of the practical design approach. All interviewees described a design approach with sequential phases and a design process from a conceptual to a detailed design, which is reminiscent of the systems engineering approach. But the phases from the civil engineering and rock manual method were also mentioned by the interviewees. The practical breakwater design approach can thus be interpreted as a more specific adaptation of the systems engineering and civil engineering method. The practical approach is repeated in Figure 7.1.

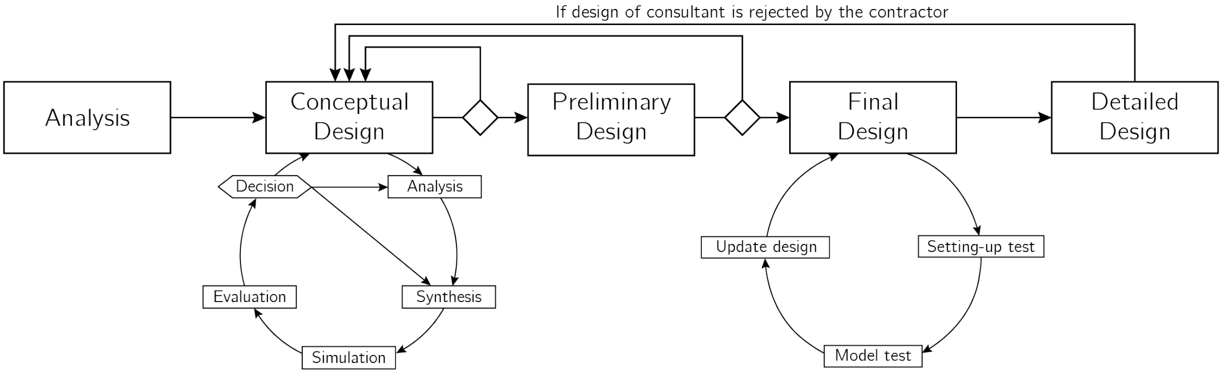


Figure 7.1: Practical design approach as described by the interviewed designers

The influence of the designer is largest in the early design phases, as the designer has the most freedom during these early phases. This is because in the later design phases, the design is further refined by designing subcomponents, or subsystems, to be in line with the terms from systems engineering. The general concept does, however, not change in these later design phases; a rubble mound breakwater will not be changed in a caisson breakwater. The early design phases, therefore, benefit the most from exploring more concepts, and the conceptual design phase was thus identified as the phase best suited for design automation.

During the interviews, experienced breakwater designers stated that geotechnical failure mechanisms are important considerations, especially for a caisson breakwater. Furthermore, experience is also an important factor when designers generate concepts since engineering judgement and experience were mentioned as important methods for selecting the optimal concept.

Because none of the existing tools fulfilled all requirements, a new design automation tool had to be developed. It must be noted that all tools were already ruled out for expansion or usage since none of the described tools were open-source. The new design automation tool is open-source, see the code on the GitHub repository and the documentation on breakwater.readthedocs.io. Furthermore, the tool also includes a geotechnical failure mechanism for the implemented breakwater types, the conventional rubble mound breakwater with rock or armour units as armour layer and the vertical (composite) breakwater. These breakwaters can quickly be designed with the developed tool, since the 'slowest' breakwater type, the rubble mound with armour units, still only takes, on average, 0.14s to generate and verify. It can, therefore, be concluded that the tool is able to quickly generate and verify rubble mound and caisson breakwaters.

The first part of the objective now remains: does the developed tool improve the breakwater design process? The breakwater design process is improved if more feasible concepts are explored. In the current design process, this is prevented by preferences, conservative assumptions, and time constraints. The question is, therefore: is the developed tool capable of making designers aware of their preference, and thus enable designers to consider more feasible concepts? Two design automation approaches were implemented in the developed tool. The first is the Configurations class, which generates multiple concepts at once, and the second, the interactive design application, which was used during the validation. Both interviewed designers stated that they would use the tool to investigate the feasibility of several breakwater types.

The design automation thus enables a design process where more concepts can be explored in the same period, compared to a design process without the tool, which is an advantage in a tender where time is limited. Both interviewees confirmed that the tool can be used in a design process during tenders or client meetings, and does indeed help in exploring more concepts. Furthermore, designers also reported that, when correctly verified, an automation tool results in fewer mistakes as it is harder to change the source code than Excel sheets. Moreover, the tool enables designers to quickly assess the influence of certain parameters on the design and cost. This results in a better understanding of the design space, and thus in a more optimal design.

Because of the importance of experience, it was decided to develop a tool that designs based on the input given by the designer and, thus, does not make decisions on its own. This enables a design process that combines the strong points of automation, speed, and accuracy, with the strong points of designers, their experience, and ability to immediately assess concepts. It is thus not a full, but a semi-automated design process. The automated steps are the synthesis, simulation, and evaluation steps of the conceptual design phase, see Figure 7.2.

The developed design automation tool allows for a quick exploration of feasible concepts, incorporates the experience of designers, and can be used in the early design phases. Especially the early design phases benefit from exploring more concepts and investigating the influence of parameters on the design and cost. It is therefore credible to conclude that the objective is achieved and that the developed design automation tool improves the breakwater design process.

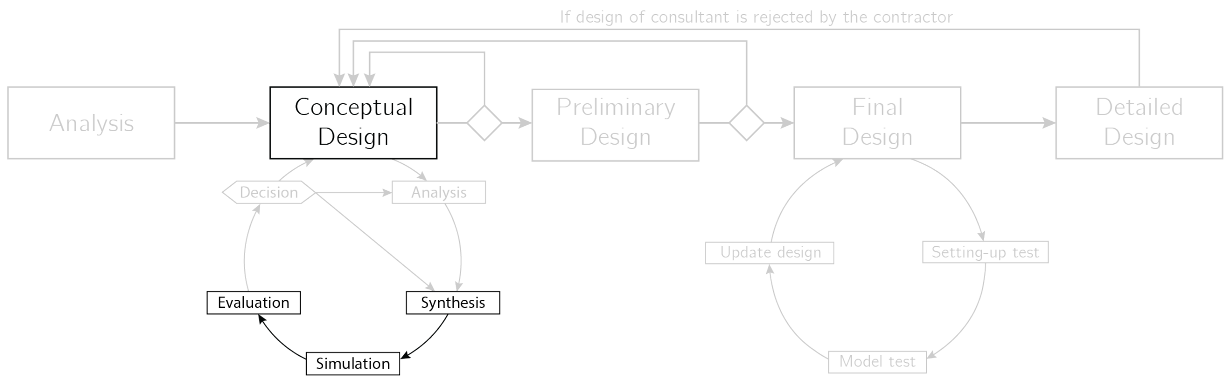


Figure 7.2: Part of the practical design approach improved by the developed design automation tool

8 Recommendations

In this chapter, recommendations are suggested to overcome the assumptions and limitations of this research. It is divided into four separate sections. In the first section, recommendations are suggested to improve the literature study and interviews of Part I. In this part, the breakwater database was used to analyse statements made during the interviews. However, this analysis can be improved. Recommendations regarding the breakwater database are therefore presented in Section 8.2. Hereafter, in Section 8.3, recommendations are suggested to improve the developed design automation tool, since several design choices and simplifications were made during the development. This is followed by the final section of this chapter in which recommendations are made to improve the verification and validation of the developed tool.

8.1 The Process of Designing Breakwaters

The main focus of this research, besides the development of a design automation tool, was the breakwater design approach. This was the main focus of Part I and was primarily performed by interviewing experienced breakwater designers and relating practice to the available design literature. To improve and further research the field of design processes, the following recommendations are suggested:

- The design literature related to generating and selecting concepts used in this research is mainly based on other design problems, for instance, product design. It should be verified if the described behaviour in this literature can indeed be observed in the breakwater design process, by observing designers during an actual design process.
- Literature about how engineers design and the decisions they make in this process is limited. Therefore, more research is required about how designers design, and the, sometimes, implicit decisions they make.
- During the interviews it became apparent that probabilistic approaches are currently not used in breakwater design practice. Lack of data has been mentioned as the primary reason, and, it must therefore be researched if this is the only missing link.
- The interviewed designers stated that using a probabilistic design results in a design that is too conservative and expensive. However, this statement was not researched with an experimental study. It is, therefore, interesting to conduct such research and further explore the mismatch between probabilistic design approaches and practical breakwater design.
- Only Dutch breakwater designers have been interviewed, which can result in a bias towards Dutch, or western European design cultures. However, this must further be investigated by interviewing designers from different countries.

8.2 The Breakwater Database

In this research, the breakwater database was used in Section 3.4 and Appendix C to analyse the statements made during the interviews. Furthermore, the database was used to investigate if relations can be found between certain parameters and breakwater types. The database is suited for this purpose as the data describing the breakwaters is a direct consequence of current design practice. To further improve this analysis the following recommendations are suggested:

- Improve the breakwater database by filling in the missing data. The database can then be used to better analyse the influence of, for instance, the contractor on the design.
- Add the proximity of quarries and dry docks to the breakwater database. This to verify if the proximity of a quarry and dry dock is indeed as important as mentioned by the interviewed designers.

8.3 Improving the Design Automation Tool

In this research, a design automation tool has been developed for the conceptual design of the conventional rubble mound and vertical(ly) (composite) breakwater. During the development, several design choices, assumptions, and simplifications have been made. However, to further develop the tool these choices must be minimised as much as possible. The following recommendations are suggested to further improve the developed tool:

- Improve the design classes by implementing more cross-sectional designs. Currently, only one cross-sectional design is implemented for each breakwater type.
- Improve the GUI so that it can be used as a separate tool and Python knowledge is no longer required to use the tool.
- Implement the improvements suggested during the validation interviews (also see Subsection 5.2.4):.
  - Possibility to specify the freeboard and compute the overtopping discharge, instead of only being able to specify overtopping discharge and computing the required freeboard
  - Computing the wave transmission
  - Add more geotechnical failure mechanisms
  - Improve graphical representation of the structure by, for instance, adding the significant wave height as a sine wave and visualise the percentage of concrete and fill material
  - Add a clear definition sketch of the parameters to the GUI
  - Divide input screens into three screens instead of one. Screen one can then be the hydraulic input, screen two the geotechnical input, and screen three the safety factors
  - Add the following parameters to the infobox (box at the right of the plot):
    * Volume of material per layer in m^3
    * Number of rows required for armour units
  - Possibility to overrule all parameters, for instance, specify the grading to use for the armour layer.
- Implement changes to the design classes so that fewer design choices have to be made, and the design freedom of the design classes is expanded. This limits the number of times designers need to use the functions from the core to design a breakwater. See Subsection 4.3.5 for the design choices of the design classes.
- Implement the missing breakwater types in the tool, since currently only the conventional rubble mound, vertical, and vertically composite breakwaters are implemented. This allows the designers to quickly explore more feasible concepts in the early design phases, improving the breakwater design process.
- During the interviews, it was mentioned that errors are also made in the translation of 2D cross-sections into a 3D design. This is a problem that can be overcome by expanding the current design automation tool with a 3D module. The module can then be used to define and design multiple cross-sections, which are then translated to a 3D design.
- Create a link between the developed design automation tool and SWAN so that the effects of the breakwater on the hydraulic conditions can quickly be investigated.

- The availability of equipment/material is currently implemented by allowing the user to specify a transport cost per cubic meter and the rent of a dry dock. This can further be improved by allowing the user to specify the location of quarries, or include a database of quarries from which the tool can determine the best quarry to use.
- The implementation of the following failure mechanism (also see Figure 4.1 and 4.3):
  - Rubble mound and Caisson: settlement of the subsoil
  - Rubble mound: slip circle through the inner slope
  - Rubble mound: slip circle through the toe
  - Rubble mound: erosion of the armour of the inner slope
  - Rubble mound: settlement of the core
  - Caisson: settlement of the core of the foundation
  - Caisson: slip circle through the foundation
  - Caisson: failure of the caisson structure
  - Caisson: the structural strength of the caisson
- Several experimental equations exist for the same failure mechanism. To incorporate the experience of designers with certain equations, other equations for the same failure mechanism must be implemented. It is then up to the designer to select the equations to be used for the design.
- Maintenance cost are currently not implemented in the tool

8.4 Improving the Verification and Validation of the Tool

This section provides recommendations to further improve the verification and validation of Chapter 5. These recommendations are:

- Verify the developed tool by using the designs of completed breakwater projects.
- Investigate if designers actually use the tool to explore concepts which would otherwise not have considered because of time constraints.
- In this research, the usefulness of the tool was assessed by interviewing experienced breakwater designers and asking their opinion. However, the designers did not use the tool in an actual design process. Therefore, the designers should be interviewed again, after using the tool in their design process.
- Only two designers have been interviewed for the verification and validation of the tool. Preferably, more experienced designers should be interviewed for the verification and validation of the tool.

References

ABN AMRO (2019). Over faalkosten in de bouwsector: Verspilde moeite. Technical report.

Allsop, N. W., Cork, R. S., and Verhagen, H. J. (2009). A database of major breakwaters around the world. *Coasts, Marine Structures and Breakwaters: Adapting to Change - Proceedings of the 9th International Conference*, 2:676–679.

Babbie, E. (2011). *The Basics of Social Research*. Wadsworth, Cengage Learning, Belmont, CA, 5th edition.

Battjes, J. A. and Groenendijk, H. W. (2000). Wave height distributions on shallow foreshores. *Coastal Engineering*, 40(3):161–182.

Beerda, E. (2020). Gezamenlijke software-ontwikkeling Heijmans, BAM en Ballast Nedam: 'Deze berekeningen moet toch iedereen uitvoeren'. https://www.cobouw.nl/bouwbreed/nieuws/2020/04/gezamelijke-app-heijmans-bam-en-ballast-nedam-deze-berekeningen-moet-toch-iedereen-uitvoeren-101283478. Retrieved on 02-05-2020.

Bernal, M., Haymaker, J. R., and Eastman, C. (2015). On the role of computational support for designers in action. *Design Studies*, 41:163–182.

Bouwend Nederland (2019). De National Benchmark Digitalisering in de Bouw. Technical report.

Brinch Hansen, J. (1970). A revised and extended formula for bearing capacity. *Bulletin of the Danish Geotechnical Institute*, (28):5–11.

Burcharth, H. (1992). Reliability evaluation of a structure at sea. *Proc. Of the Short Course on Design and Reliability of Coastal Structures*, pages 597–644.

Cederfeldt, M. (2007). *Planning Design Automation: A Structured Method and Supporting Tools*. Phd thesis, Chalmers University of Technology, Göteborg, Sweden.

CIRIA, CUR, CETMEF (2007). *The Rock Manual. The use of rock in hydraulic engineering (2nd edition)*. C683, CIRA, London, 2nd edition.

CORE studio (2020). Desgin explorer. http://core.thorntontomasetti.com/design-explorer/. Retrieved on 22-04-2020.

Daly, S., McGowan, A., and Papalambros, P. (2013). Using qualitative research methods in engineering design research. *Proceedings of the International Conference on Engineering Design, ICED*, 2 DS75-02:203–212.

Davila Delgado, J. M. and Hofmeyer, H. (2013). Automated generation of structural solutions based on spatial designs. *Automation in Construction*, 35:528–541.

De Groot, A. (1994). *Methodologie. Grondslagen van onderzoek en denken in de gedragswetenschappen*. Van Gorcum, Assen, 12th edition.

De Smet, L. (2015). *Excel-based design tool for the assessment of rubble-mound and caisson breakwaters*. Master thesis, Ghent University.

Deloitte (2019). Winning with connected construction Digital opportunities in engineering and construction. Technical report.

DMC (2014). Guidelines for xbloc concept desings. Technical report, DMC.

Dorst, K. (2011). The core of 'design thinking' and its application. *Design Studies*, 32(6):521–532.

Dorst, K. and Cross, N. (2001). Creativity in the design process: Co-evolution of problem-solution. *Design Studies*, 22(5):425–437.

Eekels, J. (1987). Ontwerpmethodologie: Mogelijkheden en Grenzen [Public Lecture].

EurOtop (2018). Manual on wave overtopping of sea defences and related structures. Technical report, EurOtop. Van der Meer, J.W., Allsop, N.W.H., Bruce, T., De Rouck, J., Kortenhaus, A., Pullen, T., Schüttrumpf, H., Troch, P. and Zanuttigh, www.overtopping-manual.com.

Ford, C. M. and Gioia, D. A. (2000). Factors Influencing Creativity in the Domain of Managerial Decision Making. *Journal of Management*, 26(4):705–732.

Goda, Y. (2000). *Random Seas and Design of Maritime Structures*, volume 15 of *Advanced Series on Ocean Engineering*. World Scientific Publishing Co. Pte. Ltd., PO Box 128, Farrer Road, Singapore, 2nd edition.

Goodger, D. (2002). Pep 287 – restructuredtext docstring format. https://www.python.org/dev/peps/pep-0287/. Retrieved on 29-05-2020.

Google (2020). Why open source? https://opensource.google/docs/why/. Retrieved on 02-05-2020.

Hertogh, M. and Bosch-Rekveldt, M. (2013). *Integraal Ontwerpen en Beheer*. Lecture notes, Delft University of Technology.

Hudson, R. Y. (1959). Laboratory investigation of rubble-mound breakwaters. *Journal of the Waterways and Harbors Division*, 85:610–659.

International Council on Systems Engineering (2019). Systems engineering definition. https://www.incose.org/about-systems-engineering/system-and-se-definition/systems-engineering-definition. Retrieved on 25-10-2019.

Jacob, S. A. and Furgerson, S. P. (2012). Writing Interview Protocols and Conducting Interviews: Tips for Students New to the Field of Qualitative Research. *The Qualitative Report*, 17:1–10.

Kirkpatrick, D. (1998). The second coming of apple through a magical fusion of man. *Fortune*, (23). Retrieved on 30-04-2020.

Koeleman, J., Ribeirinho, M. J., Rockhill, D., Sjödin, E., and Strube, G. (2019). Decoding digital transformation in construction | McKinsey. *McKinsey&Company*.

Koenen, I. (2020a). Bouw legt zich neer bij hardnekkig hoge faalkosten: 'onbegrijpelijk'. https://www.cobouw.nl/bouwbreed/nieuws/2019/04/bouw-legt-zich-neer-bij-hardnekkig-hoge-faalkosten-onbegrijpelijk-101271460?_ga=2.253077964.49083057.1587389212-1906355625.1587389212. Retrieved on 20-04-2020.

Koenen, I. (2020b). Waarom bijna niemand brood ziet in megaproject ViA15: een reconstructie. https://www.cobouw.nl/infra/nieuws/2020/02/waarom-zag-bijna-niemand-brood-in-megaproject-via15-een-reconstructie-101281695. Retrieved on 20-04-2020.

Koenis, C. (2018). Zeesluis ijmuiden ruim 2 jaar vertraagd, strop voor BAM en Volker. https://www.rtlz.nl/beurs/artikel/4269981/zeesluis-ijmuiden-ruim-2-jaar-vertraagd-strop-voor-bam-en-volker. Retrieved on 20-04-2020.

Labouchere, P. (2018). Systematic and Creative Design for Design and Build Tenders. *Internal Memorandum*, pages 1–10.

Laenen, K. (2000). *Probabilistisch model voor het vergelijken van twee golfbrekertype op basis van economische optimalisatie*. Master thesis, Delft University of Technology.

Lawson, B. and Dorst, K. (2005). *Computational and Cognitive Models of Creative Design VI*, chapter Acquiring design expertise, pages 211–230. KeyCentre, University of Sydney.

MarCom WG 47 (2016). Criteria for the selection of breakwater types and their related optimum safety levels. Technical Report 126, PIANC, Boulevard du Roi Albert II 20, Brussels, Belgium.

McKinsey&Company (2017). Reinventing Construction: A Route To Higher Productivity. Technical report.

McLaughlin, B. (2006). Types of Interviews.

Nijstad, B. A., Stroebe, W., and Lodewijkx, H. F. (2002). Cognitive stimulation and interference in groups: Exposure effects in an idea generation task. *Journal of Experimental Social Psychology*, 38(6):535–544.

Onencan, A. (2013). Interviewing Techniques [Lecture Slides]. Brightspace of EPA7030 Interviewing Techniques (2019/20 Q3). Retrieved on 31-10-2019.

Ramón, G. S., Grassa, J. M., and Grau, J. I. (2010). Breakwater development in Spain. The last ten years. *Coasts, Marine Structures and Breakwaters: Adapting to Change - Proceedings of the 9th International Conference*, 1:43–61.

Rietzschel, E. F., Nijstad, B. A., and Stroebe, W. (2010). The selection of creative ideas after individual idea generation: Choosing between creativity and impact. *British Journal of Psychology*, 101(1):47–68.

Rijkswaterstaat (2018). Rijkswaterstaat gunt project afsluitdijk aan consortium levvel. https://www.rijkswaterstaat.nl/nieuws/2018/03/rijkswaterstaat-gunt-project-afsluitdijk-aan-consortium-levvel.aspx. Retrieved on 22-04-2020.

Rijkswaterstaat (2019). Toekomstige Opgave Rijkswaterstaat: Perspectief op de uitdagingen en verbetermogelijkheden in de GWW-sector. Technical report.

Roozenburg, N. and Eekels, J. (1995). *Product Design: Fundamentals and Methods*. Wiley series in product development: planning, designing, QREngineering. John Wiley & Sons Ltd., Baffins Lane, Chichester, England.

Royal BAM Group nv (2008). SE-wijzer Handleiding Systems Engineering voor BAM Infra. Technical report, Royal BAM Group nv, Bunnik.

Rustell, M. (2016). *Knowledge extraction and the development of a decision support system for the conceptual design of liquefied natural gas terminals under risk and uncertainty*. Phd thesis, University of Surrey.

Sigurdarson, S., Smarason, O., and Viggosson, G. (2006). Berm Breakwaters. *Per Bruun, Port Engineering*, pages 1–28.

Sijbesma, F. B. (2019). *A parametric approach to a probabilistic design of rubble mound slope protection*. Master thesis, Delft University of Technology.

Speckle (2020). What is speckle. https://speckle.systems/docs/essentials/introduction/what. Retrieved on 30-04-2020.

Takahashi, S. (2002). Design of vertical breakwaters. Technical Report 34, Port and Airport Research Institute, Japan.

Tamimi, A. K., Ali, T., Anoohi, R., Rajput, A., and Alkamali, K. (2019). Evaluations of 3D Concrete Printing Produced in the Environment of United Arab Emirates. *International Journal of Civil and Environmental Engineering*, 13(12):705–709.

Toh, C. A. and Miller, S. R. (2015). How engineering teams select design concepts: A view through the lens of creativity. *Design Studies*, 38:111–138.

TU Delft (2020). Integral design and management (idm) annotation. https://www.tudelft.nl/studenten/faculteiten/citg-studentenportal/onderwijs/master/annotations/integral-design-and-management-idm-annotation/. Retrieved on 03-05-2020.

TU Delft and HR Wallingford (2019). International breakwater directory. http://kennisbank-waterbouw.tudelft.nl/breakwaters/. Accessed on 13-11-2019.

Van Boeijen, A. and Daalhuizen, J. (2010). *The Delft Design Guide*. BIS Publisher.

Van den Bos, J. and Verhagen, H. (2018). *Breakwater Design*. Lecture notes, Delft University of Technology.

van der Meer, J. (1988). *Rock slopes and gravel beaches under wave attack*. Phd thesis, Delft University of Technology, Delft, The Netherlands.

van der Meer, J. (1998). Geometrical design of coastal structures. In *Seawalls, dikes and revetments*, chapter 9, pages 161–175. Balkema, Rotterdam, The Netherlands.

Van Gent, M. R., Smale, A. J., and Kuiper, C. (2003). Stability of rock slopes with shallow foreshores. *Coastal Structures 2003 - Proceedings of the Conference*, 40733:100–112.

van Nieuwenhuizen-Wijbenga, C. (2019). Structuurvisie ZuidasDok [letter of government]. Retrieved from https://www.tweedekamer.nl/kamerstukken/verslagen/detail?id=2019D31568. Retrieved on 20-04-2020.

van Rossum, G., Warsaw, B., and Coghlan, N. (2013). Pep 8 – style guide for python code. https://www.python.org/dev/peps/pep-0008/. Retrieved on 29-05-2020.

van 't Land, W., Bastiaens, G., van Wolfswinkel, J., and Heideman, P. (2018). Samenwerking door digitale ontwerpmethoden. *Cement*, 70(7):48–52.

Veerman, M. (2020). The new cad: Code aided design. Presentation during IABSE Connect Seminar 'digital techniques and the end of the structural engineer'.

Verruijt, A. (2012). *Soil Mechanics*. Delft University of Technology, Delft, The Netherlands.

Voorendt, M. (2017). *Design principles of multifunctional flood defences*. Phd thesis, Delft University of Technology.

Voorendt, M. (2020). personal communication.

Wason, C. S. (2015). *System Engineering Analysis, Design, and Development*. Wiley series in system engineering and management. John Wiley & Sons, Inc., Hoboken, New Jersey, 2nd edition.

Appendices

A

State of the Construction Industry

Chapter 1 started with the claim from McKinsey&Company (2017) that the construction industry evolved at a glacial pace compared to other industries. This claim was further substantiated in Subsection 1.2.2 by the fact that the productivity growth of the construction industry was lagging other industries. In Section A.1 the conclusions and recommendations of these consultancy firms are summarised. Hereafter the current state, and future, of the digitisation of the construction industry is discussed. This Section is followed by Section A.3, in which the education of the digital construction is discussed.

A.1 Consultancy view on the Construction Industry

Over the past decade, several major consultancy firms have analysed the construction industry to investigate why productivity is lagging behind other industries. The main reasons for the lagging productivity are (McKinsey&Company, 2017; Deloitte, 2019):

- The increasing complexity of projects. This complexity can be explained twofold, first, the technical complexity of projects is increasing and secondly the site complexity.
- Misalignment of contractual structures and incentives.
- The construction industry is highly fragmented, which prevents chain optimisation.
- An underinvestment in digitalisation and innovation.
- Inadequate and inefficient design processes

These main reasons, apart from the last two, are about external factors and industry dynamics. Therefore, they are not easily changed and require a willingness to change across the industry. Over the past two years, the willingness to change the industry is growing in The Netherlands, partly due to large cost overruns on recent projects. Examples are the construction of a sea lock in IJmuiden, which is 200 million euros over budget (Koenis, 2018). While, the situation for the Zuidasdok project is even grimmer with an estimated cost overrun of 0.7 to 1 billion euros, according to a letter from the Minister of Infrastructure and Water Management (van Nieuwenhuizen-Wijbenga, 2019).

The major cost overruns on large infrastructure projects in The Netherlands resulted in several consortia dropping out of tenders. For the ViA15 project, this resulted in only one valid bid, an unwanted situation for Rijkswaterstaat, the public works authority, because of a lack of competition between contractors. In an article from Koenen (2020b) on Cobouw a tender specialist of VolkerWessels cited the high risks of the project as the reason for dropping out, almost all risks were allocated to the contractors. Furthermore, he stated that public contractors have a monopoly position and that the market can only indicate a problem with the risk allocation of the project by dropping out.

As a result, Rijkswaterstaat commissioned McKinsey to investigate the Dutch civil sector. The report concluded that the risk allocation of current large infrastructure projects is indeed a problem for contractors and that the risks can be lowered by changing the tender process (Rijkswaterstaat, 2019). The recommendation is to change the current tender process to a two-phase tender process, where the tender is split between a design phase and a construction phase. The cost estimation of the construction phase will happen after the design phase has been completed when more information is known and risks can thus better be estimated (Rijkswaterstaat, 2019).

However, the Rijkswaterstaat (2019) also mentions a lack of innovation and productivity improvement of the construction industry. This is the recurrent conclusion of the major consultancy firms is that rethinking processes and innovation either goes to slow, or doesn't even happen at all. This is an improvement companies can make on their own, it is then about the vision of the company and how they see the future of the construction industry. McKinsey&Company (2017) cited that the disruption is slow because of a lack of R&D processes, and investment in R&D. One of the cited reasons is that the margins are small, and this is indeed true. However, as Deloitte (2019) pointed out, digitalisation of the industry can also help to improve the margins and even generate new income sources. This can help to reduce the failure cost on projects, and thus the chance of cost overruns.

Nonetheless, ABN AMRO (2019) concluded in their report that despite 90% of the companies being aware of the failure cost, over 25% state that it is not a priority in their company to reduce the failure cost. Which is remarkable as more than half of the companies have failure cost above 5% a year (ABN AMRO, 2019). Indicating that the failure cost are an accepted inefficiency. Petran van Heel, from ABN AMRO, states in an interview that the ability to learn is hard to find. He adds to this that it is inconceivable that so much money is lost, as a large number of mistakes are due to miscommunication (Koenen, 2020a). Again, amongst other reasons, a lack of innovation is mentioned (ABN AMRO, 2019).

A.2 Digitisation in Practice

Two of the main reasons mentioned in the previous section are an underinvestment in digitalisation and innovation and inefficient design processes. In Subsection 1.2.2 it was already highlighted that companies do identify the potential of digitalisation, mainly to increase the efficiency and reduce the cost (Bouwend Nederland, 2019). Therefore, this section first discusses the current state of digitalisation in Subsection A.2.1. Followed by Subsection A.2.2 with the future of the construction industry.

A.2.1 *Current State*

During a symposium of the International Association for Bridge and Structural Engineering (IABSE) Veerman (2020) gave the following statement: *"Over the past decades we have had evolution, not revolution."* One of the examples he gave was that CAD software for the generation of technical drawings is now widely adopted, but that the process of making technical drawings has not changed.

However, there are processes currently changing, for example, 3D printing of concrete. This can be seen as a true revolution as it changes the way a concrete component is constructed. The revolution is that 3D printing of concrete allows for more complex shapes of concrete since a wooden formwork is no longer needed, which also saves material. Furthermore, Tamimi et al. (2019) showed that during a demo project in Dubai a house could be constructed in 19 days, but that currently the main problem is the concrete mix. The revolution can even go further by rethinking the design process. When linking the tools used by the different disciplines, all involved parties immediately have all the necessary data. van 't Land et al. (2018) showed that with the computational design tools the construction can be optimized, resulting in a strong reduction of material use and thus a more sustainable construction.

In recent years thing have been changing, the added benefits of digitalisation become more clear for several companies. This was also concluded by Bouwend Nederland (2019) in their benchmark study of the Dutch construction industry. 85% of the interviewees acknowledged that digitalisation has the potential to reduce failure costs and risks. However, the interviewees also mention that one of the main problems preventing digitalisation is that there is no clear return on investment. The return on investment is indeed hard to quantify, as each project is unique which makes it difficult to compare one project to another. This might be the explanation for the lack of investment in digitalisation and innovation mentioned by McKinsey&Company (2017); Deloitte (2019).

However, as so many persons from inside and outside the construction industry acknowledged the added benefits, it is strange to conclude that adaptation is lagging and that the real disruption has still not occurred. After all these years of ploughing on, the lack of investment cannot be the only explanation. Maybe this quote by Steve Jobs, given to Kirkpatrick (1998) in Fortune Magazine, helps to give another perspective: *"Innovation has nothing to do with how many R&D dollars you have. When Apple came up with the Mac, IBM was spending at least 100 times more on R&D. It's not about money. It's about the people you have, how you're led, and how much you get it."*

On the positive, there are plenty of initiatives and companies where innovation and digitalisation is stimulated. One of these examples is Speckle, which is an open-source data platform for the AEC[1] industry (Speckle, 2020). Its main purpose is to enable the user to transfer data from one platform to another platform, for instance from Unity to Revit. Since it is an open-source initiative it means that anyone can contribute to the project, or use it in their workflow.

The benchmark of Bouwend Nederland (2019) also included a matrix in their assessment of the maturity of digitalisation, this matrix is translated and presented in Figure A.1. The laggards have not yet, or are barely, started with the digitalisation, whereas the innovators are the most advanced. The orange bar indicated the average adoption among the companies. From the matrix, it can clearly be seen that the majority started with the digitalisation, on assigning responsibility and planning most of the companies are even quite advanced. However, it can also be seen that the actual adoption is not as far advanced, as the average adoption is also the average between the laggards and the innovators.

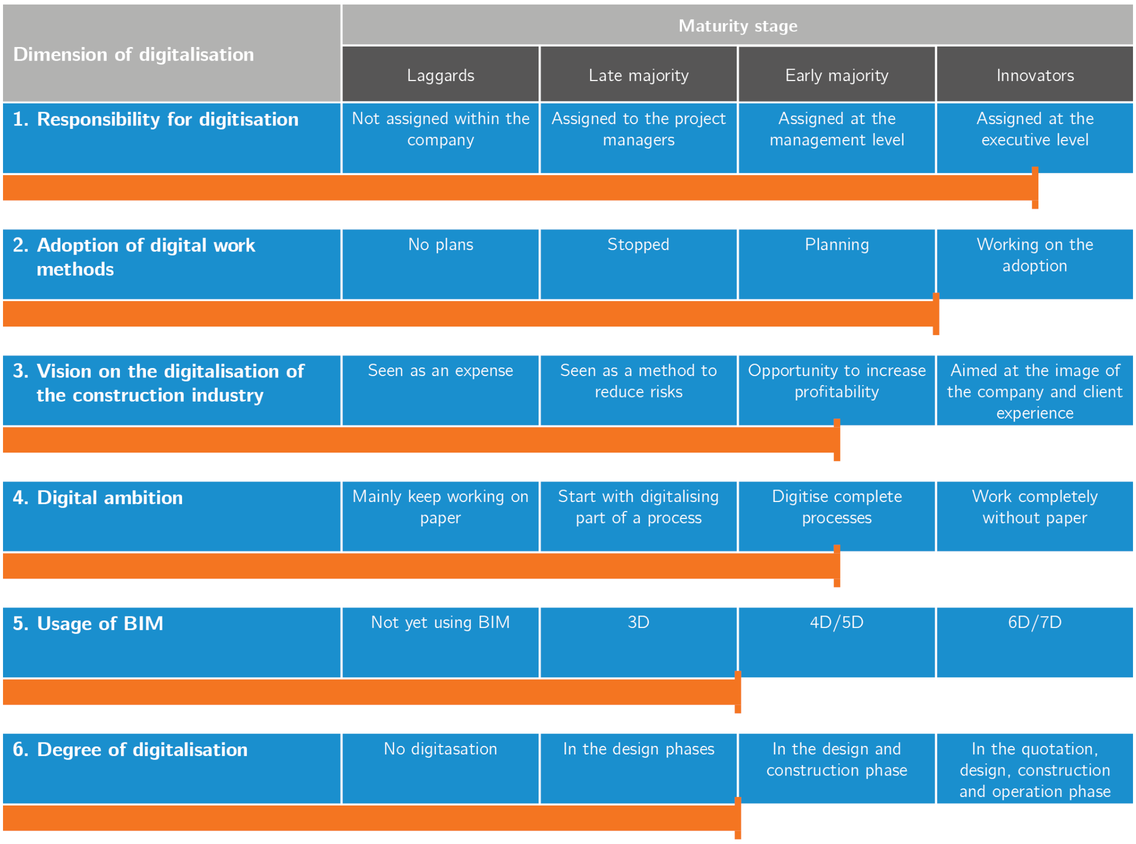


Figure A.1: Matrix depicting the different dimensions of digitalisation in relation to the maturity and adaptation of the technology. Matrix copied and translated from Bouwend Nederland (2019, p. 25)

A.2.2 *Future*

The coming years the construction industry will see major changes, the designers and project managers in the offices are the first to see this change, after which the construction site will follow. This subsection will, however, mainly focus on the digitalisation of the offices.

[1]AEC stands for Architecture, Engineering, and Construction and is an abbreviation used in the construction industry

First, there will be a change from computer-aided design to code aided design. Computer-aided Design, CAD, is the process of translating the technical drawings from paper to digital versions, and was the point about evolution and not revolution the speaker at the IABSE conference tried to deliver. Code aided design is a design process where design automation plays an important role. In such a process the designer works together with computer programs to design new structures. In this, the process the computer will not take over the designing tasks, but merely show possible configurations, in the end, it is up to the designer to combine or generate new ideas. However, it will take a lot of effort to develop these algorithms. Furthermore, there is a lot of scepticism towards these tools, as designers state that these algorithms can never replace the creativity of designers as they can merely do what they are programmed to do.

What is more feasible, and is what I think is the future of the construction industry, at least from the code aided design point of view, are smaller algorithms capable of automating the repeating and routine tasks. Especially to replace the many Excel sheet computations that are now used in the offices. One of the main advantages is that the algorithms are more flexible and easier to expand. Furthermore, the algorithms can handle large amounts of data more efficiently.

The largest advantage for code aided design is, however, the possibility to create a link between several programmes, by using the API[2]. By linking several programmes together the amount of time required to perform certain tasks is reduced. Furthermore, as fewer numbers need to be copied by hand, the amount of errors is also reduced, which reduces the overall failure costs. And as there is a central place where the numbers are stored, everyone has the same, and correct, number or information.

To develop the required algorithms more industry-wide initiatives, as Speckle, are needed. However, these projects are open source and one might argue that it will be impossible to get a return on investment. This is indeed a large disadvantage, and will likely prevent new developments. A solution is the strategy of Google and Facebook, both companies have opened a part of their platform to enable developers to build applications based on their platform. Furthermore, Google also states that one of the reasons is to increase the creativity, as a more diverse pool of people are working on a project (Google, 2020). Furthermore, Google argues that by open-sourcing some software a larger return on investment is made because more people are working on a project.

However, Google (2020) also mentions that not everything must be open-source, for instance, if the software will reveal company secrets or advantages. This is also the main reason Heijmans do not share their dike design tool. But Heijmans is prepared to share an older, outdated, version of the programme in an initiative called the *Digital Engineering Community*. The Digital Engineering Community is a multi-company initiative to develop new applications for civil engineering (Beerda, 2020). The main advantage is that the investment costs are spread over several companies, reducing the investment cost for each company. Furthermore, it also has the added benefit that more companies are able to use the application, advancing the construction industry.

Digital twins are also gaining attention over the past year, a digital twin is a link between the virtual BIM environment and the building itself. By fitting the building with several sensors the state of the building can be monitored in real-time to, for instance, see when maintenance must be performed. With a digital twin, the utilisation of a building can also be monitored, this enables the operator to more efficiently light or heat the building, which results in a more sustainable building. If this technology is compared with the usage of BIM in Figure A.1 it can be seen that the digital twin is not even mentioned by the innovators. although it must be noted that the digital twin is also sometimes explained as a 7D BIM model.

The final development to be discussed is Virtual Reality (VR). Over the past year construction companies started to use VR as a tool during the design phase. VR enables designers to experience their design, and see how their design translates to the 3D environment. Designers are able to literally walk through their building or drive over the designer highway. Which gives designers the ability to identify problems in their design, and thus correct them before construction has started. This gamification of designs is called serious gaming and is also used to, for instance, verify if an oil refinery can safely be operated from the designed operations room. Furthermore, VR can also be used as a communication tool as stakeholders can experience their design, and see with their own eyes how a design is integrated into their neighbourhood.

[2]API is the abbreviation used in computer science for Application Programming Interface. An API defines the requests that can be made to a programme, and how to make them.

A.3 The Integral Design and Management Annotation

Civil Engineering students at the TU Delft are able to follow the Integral Design and Management (IDM) annotation. The annotation is meant for students who want to combine their technical knowledge with engineering management skills, such as digital design&construction (TU Delft, 2020). This section elaborates on the relation between the IDM annotation and this research. First, in Subsection A.3.1, the relation between the annotation and this research are described. Hereafter, the advantages and disadvantages of the annotation are discussed.

A.3.1 *Relation between the Annotation and this Research*

This research aims to combine the knowledge from several research fields to improve the breakwater design process, which in this research is done by developing a design automation tool. Combining several research fields, in particular the technical with digitalisation and design practice is precisely the goal the IDM annotation tries to achieve.

During the IDM annotation, several courses can be followed related to project management and the systems engineering method, which was described in Subsection 2.1.4 as a design method. Furthermore, in Subsection 2.1.5 it was concluded that all described methods are similar, and can be interpreted as more specific adaptations of the systems engineering method. Furthermore, as the development of the design tool can be seen as a design process on its own, elements of the systems engineering method can be seen throughout this research. For example, Chapter 5, where the verification and validation process of the design automation tool was carried out. But also during the development of the process a systems engineering approach was used, since experienced breakwaters were involved during the development by interviewing them.

Another strong link with the IDM annotation is the development of the design automation tool. As a part of the IDM annotation, several courses can be followed about the digitalisation of the construction industry. Most of these courses are, however, mainly focused on the management side of the construction industry. This is 3D and 4D modelling, and life cycle management with BIM software. How to develop a design automation tool is not part of the courses offered by the IDM annotation. The only course at the TU Delft teaching a type of design automation is *Parametric Design and Engineering*, but this course focuses on design automation tool for buildings.

A.3.2 *Usefulness of the Annotation*

Following the IDM annotation, especially the compulsory course about systems engineering proved to be an advantage while conducting this research. This was mainly because systems engineering is a fundamental design method and elements can be seen in other methods, and in the description of the practical design approaches. Furthermore, the systematic approach was fundamental during the development of the design automation tool. First and foremost in the extensive inclusion of the designers, as requirements were derived from the interviews. Secondly, by providing a framework for testing the tool, i.e. the process of verification and validation.

However, what is missing in the annotation is the focus on design automation, or parametric design, as it is most often called in practice. The only course of the Civil Engineering master programme teaching a type of design automation is *Parametric Design and Engineering*, but this course focuses on design automation for buildings. What is missing is a type of course in which the general applications of design automation can be explored, for instance during the second year course Bouwplaats in which students learn Python. Teaching Python by letting the students develop their own design automation tool for a civil engineering structure makes the why more tangible since students know-how, in this case, Python can be used within civil engineering. This important step must be taken as more and more companies are focusing on code aided design or coding links between different programmes.

Civil engineering education should, however, not change into a computer science education. What the main point is that it is important to be able to explain problems to a programmer, who has studied computer science, and to understand the problems a programmer might face when coding. This multi-disciplinary approach is important to improve the construction industry, and finally digitalise the construction industry, to lower failure cost and improve the sustainability of the industry. Furthermore, the multi-disciplinary approach is also a fundamental part of the systems engineering method and the annotation. Bridging the gap between programmers and civil engineers is, at least in my opinion, essential to successfully digitise the construction industry.

B

Interview Protocol

In this appendix the full interview protocol used during the interviews conducted in Chapter 3 is presented. The method and structure of the interview was explained in Section 3.1.

B.1 Introduction

My research is about obtaining more insight in how a breakwater is designed in practice, i.e. the design method used in practice. This will also be the topic of the first part of the interview. What I want to try, together, is to create a flow chart representing the design method you use when designing a breakwater.

During the second part of the interview I will ask questions about how variants, or concepts, are created. Specifically, when are key decisions made during the design process. This to compare the theoretical design methods described in literature to what is being done in practice.

Before we start I would like to ask if I can record this interview and use the outcome of this interview in my thesis? Furthermore, I would like to inform you that you are not obligated to answer the questions and that you can withdraw from the interview at any moment. Finally, do you have any questions about me or the research before we start?

B.2 Questions - Design Process

- How does the process of designing a breakwater look like?
  - Which processes are repeated? Or tasks that can be seen as a loop?
  - What are the key moments in this process?
- What would you identify as problems in the design process?
  - If not already answered: Why are these problems?
  - What is the current way of dealing with this problem?
  - Is this problem still present in latter design phases?
- Has a decision ever been made on common sense or intuition?
  - Could you describe the situation?
  - Reflecting back on the decision, was it your opinion about the decision?

B.3 Questions - Considering and Selecting Variants

- When is the decision for a certain type of breakwater made?
  - On what basis are these variants selected?

  - How dependent is the variant on external factors?
  - Interesting projects on the decision of variant, where different variants have been considered?
- How many variants are being considered?
  - In what phase of the design process are these variants being considered?
  - Where in the process is the decision made to go for one variant?
  - What is, in your experience, the main reasons of dismissing variants?
- Have you ever had the idea that considering more variants could be beneficial for the design process?
  - If so, which variants do you want to consider?
  - During your time as a designer has the design ever changed?
- Is the design made in a deterministic or probabilistic approach?
  - If deterministic: is a safety factor taken into account during the design?

B.4 Final Remarks

Thanks for answering all of my question. This is it from my side, but do you have any final remarks about the interview or the research? Maybe tips on how to improve the interview?

I would like to inform you that the results of this interview will be part of my thesis, and that I can provide you with a copy of this report if wanted. Finally, I would like to thank you for making time for me in your schedule.

C

Breakwater Database

The full analysis of the breakwater database, of which the results have been used in Subsection 3.4, is presented in this appendix. The breakwater database, developed by TU Delft and HR Wallingford (2019), consists of 522 breakwaters with data ranging from design wave height to contractor. In Section C.1 a description on the data in the database is given, hereafter the following analyses are performed:

- The relation between the wave height, depth, length and breakwater type in Section C.2
- The relation between the location and type of breakwater in Section C.3

C.1 Data Description

The breakwater database is prepared by HR Wallingford UK and the TU Delft. The first version of the database became available in 2009, but this version already relied on earlier work from the late 1970s (Allsop et al., 2009). Most of the breakwaters in the database are constructed between 1961 and 2010, although some old breakwaters, for example the Plymouth breakwater from 1812 is also included in the database. The version of the database used for this research has been accessed on the 13th of November 2019. However, development probably stopped around 2012 since a breakwater project with a finish date in that year is reported as: *ongoing*. Besides start and finish date the database includes more parameters, in table C.1 all parameters in the database are presented.

Table C.1: Parameters included in the breakwater database

Breakwater description		
Harbour/location	Country	Breakwater description
Description of works	Length	Max depth
Owner		
Construction		
Construction start date	Completion date	Cost
Contractor	Consultant	Hydraulic laboratory
Design		
Breakwater structural type	Design wave	Period (T_z and T_p)
Primary armour	Front slope(s)	Rear armour
Rear slope(s)	Crest elevation	Crest width

It can be seen that the database includes a parameter, breakwater structural type, which classifies the breakwater into one of the following categories:

- Rubble mound breakwaters
- Composite breakwaters
- Revetments

- Berm breakwaters
- Caisson breakwaters

However, definitions of the types given in the database are not completely aligned with the ones given in Subsection 1.2.1. Where this research makes a distinction between the horizontally and vertically composite breakwaters, the database does not. Vertically composite breakwaters are classified as caisson breakwaters, and horizontally composite breakwaters as composite breakwaters. Secondly, the rubble mound breakwater with crown wall is classified as a rubble mound. Furthermore, the database also classifies structures as revetments, which is not included in this research. A revetment is defined as a facing of stone, or concrete, constructed to protect a scarp or embankment (Van den Bos and Verhagen, 2018, p. 244). The difference between a breakwater and a revetment is thus, that a revetment does not have water on both sides of the structure, but water on one side and land on the other.

From table C.1 it can be seen that many parameters are included in the database. However, a lot of datapoints are missing. In table C.2 an overview of the available datapoints for each of the design parameters is presented. The maximum water depth and length have also been included in the table, since both have been mentioned during the interviews as important criteria when considering breakwater types.

Table C.2: Number of datapoint for the design parameters

	Type	Hs	Tz	Tp	Slope		Crest		Depth	Length
					Front	Rear	Height	Width		
Rubble mound	234	162	13	127	175	46	40	13	160	102
Composite	26	4	0	1	8	8	14	4	9	3
Revetment	52	42	0	32	43	-	1	-	40	15
Berm	60	58	24	28	16	13	15	12	45	10
Caisson	41	20	1	4	-	-	25	24	27	20
	413	286	38	192	242	67	95	53	281	150
Unclassified	**109**									
Total	**522**									

Most of the breakwaters in the database are classified as rubble mound, which is due to the fact that rubble mound breakwaters are the most common type of breakwater Allsop et al. (2009). Hence, the database is likely not biased towards the rubble mound type but a fair representation of all breakwaters. However, Allsop et al. (2009) does mention that the database is biased towards solutions with concrete armour units, since rock armoured breakwaters are not often discussed in literature. Examining the number of rubble mound breakwaters with rock as primary armour, it appears that only 19 of 234 breakwaters have rock as primary armour. Hence, the database is biased towards concrete armour units.

C.2 Wave Height, Water Depth and Length

Choosing between different types of breakwaters, especially between rubble mound and caissons, is often based on the wave height, water depth and length. These parameters have a strong influence on the choice for a certain type of breakwater, as these criteria were all mentioned by designers when asked what considerations they make during the conceptual design phase, also see Section 3.3. The considerations and criteria are:

- It is generally more economical to choose a caisson instead of a rubble mound breakwater if the water depth is larger than 15 m (CIRIA, CUR, CETMEF, 2007)
- In case of a large length and water depth a caisson is chosen over a rubble mound breakwater
- Rock is chosen over armour units if the significant wave height is smaller than 3.0 m

C.2.1 *Statistical Description*

In table C.3 the mean and standard deviation of the wave height, depth and length are presented. Note that to compute the standard deviation the assumption is made that the data is normally distributed. Furthermore, as less than 4 datapoint are available for the wave height and length of a composite breakwater, the mean and standard deviation for these structures is not computed.

Table C.3: Mean (μ) and standard deviation (σ) of the wave height, depth and length for each breakwater type. Data from TU Delft and HR Wallingford (2019).

	Hs [m]		**depth [m]**		**length [m]**	
	μ	σ	μ	σ	μ	σ
Rubble mound	5.61	2.07	12.17	8.89	2145	3785
Composite			23.67	9.89		
Revetment	5.08	1.70	8.45	4.91	2411	3422
Berm	4.53	1.55	11.49	6.05	2272	4630
Caisson	6.88	1.52	24.63	13.15	898	556

The first conclusion drawn from the data is that a breakwater with caissons is indeed chosen if the water depth is larger. However, the mean depth for which a caisson breakwater is chosen, at least for the breakwaters in the database, is deeper than the Rock Manual stated. Furthermore, the standard deviations of the depth for both indicate that there is a relatively wide range of water depths where both types of structures are feasible.

The second conclusion is that caissons are constructed in conditions with a higher mean wave height. However, as caissons are generally constructed in larger water depths it was to be expected that the mean wave height would also be larger, since larger waves occur in deeper water. For completeness the Spearman rank-order coefficient was computed to assess the correlation between the wave height and wave height. The correlation coefficient was found to be 0.5, which proves that there is indeed a positive correlation between the wave height and depth.

The wave height is used as a selection criteria for deciding between a rubble mound breakwater with armour units or rock as primary armour. While only 19 of the 234 breakwaters have rock as primary armour units, as mentioned the previous section, an analysis is nevertheless made to assess the use of the practical rule of thumb. This results in the following data: for rock a mean wave height of 3.39m±1.05m, and for armour units a mean wave height of 5.67m±1.92m. In a design process where not all options can be explored, the practical rule of thumb that rock is chosen if H_s is smaller than 3.0m is thus a reasonable and conservative estimate.

Lastly, the length of the breakwater appeared to influence the choice between breakwater types. From the interview it was found that if the length of the breakwater is relatively long, a caisson breakwater is the preferred alternative. However, choosing a caisson breakwater if the length of the breakwater is relatively long is in contradiction with the data from the database, since the mean length of the rubble mound breakwater is twice as long as the caisson. Although, it must be noted that the standard deviations are quite large, in fact they are larger than the mean of the length indicating that there is a large uncertainty. It is therefore, difficult, if not impossible, to draw any conclusion.

C.2.2 *Relation Between the Wave Height and Water Depth*

In figure C.1 the wave height and depth are plotted together. Breakwaters missing either the wave height or depth have been excluded from the plot. From the figure it can be seen that most caissons are constructed at locations where both the water depth and wave height are larger (H_s > 6.0 m and depth > 15.0 m). However, the transition between the rubble mound and caissons is not so well defined as the Rock Manual stated and that in reality the transition between the different types of structures is larger. Berm breakwaters are mainly constructed in shallower water depth with lower waves, as could already be seen in table C.3.

Reflecting back on the example from current design practice, elaborated in Section 1.1, it can be concluded that it was a strange decision to disregard rubble mound breakwaters based on the wave height. In figure C.1 it can be seen that with a wave height of 8.0m, and a water depth of 11.5m, the project was in the range where both types are feasible. However, with these parameters the preference, at least in the database, is still towards a rubble mound concept.

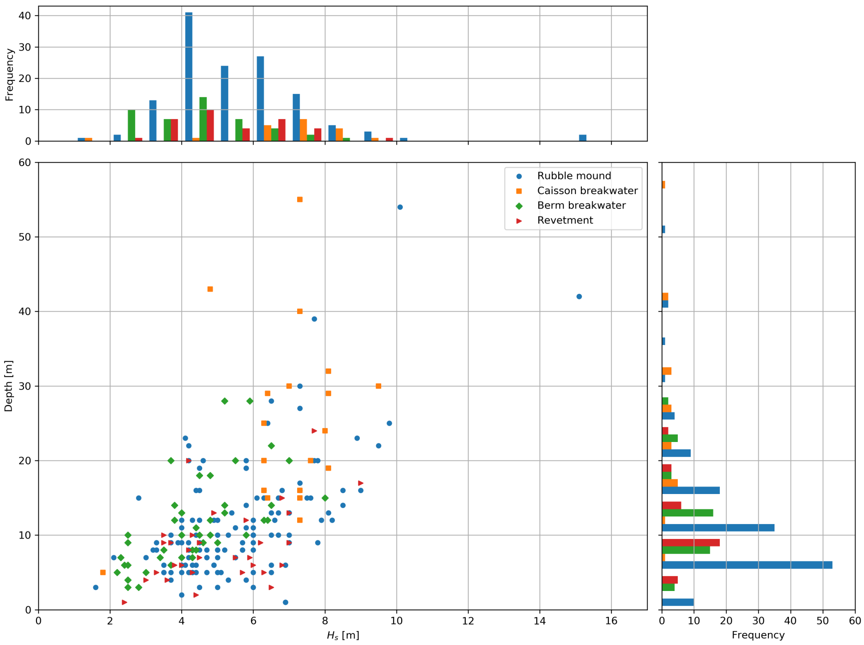


Figure C.1: Significant wave height versus water depth for different breakwater types. Data from TU Delft and HR Wallingford (2019).

In order to better investigate the relation between the wave height and water depth, the bivariate normal distribution is plotted. This distribution is only plotted for the rubble mound and caisson breakwater, as the other types of breakwaters are not within the scope of this research, see Subsection 1.4.2. The earlier computed correlation between the wave height and water depth of 0.5 is included in the computation of the bivariate. From C.2 it can be seen that there is a relatively large overlap between the different types of structures, which further highlights the need to consider both types of structures.

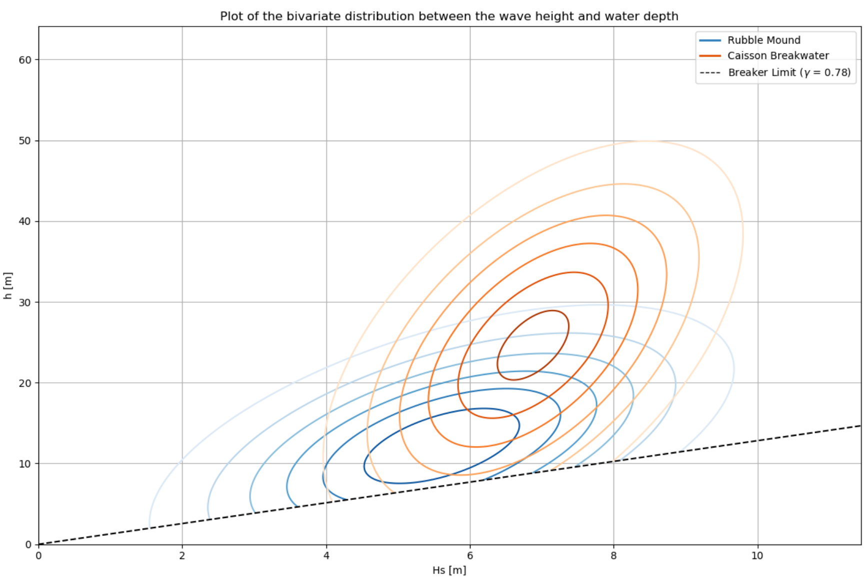


Figure C.2: Bivariate normal distribution between the wave height and water depth for a rubble mound and caisson breakwater. Data from TU Delft and HR Wallingford (2019).

C.3 Locations and Type of Breakwater

Because during the interviews, Section 3.3, it appeared that, at least, according to the interviewees there is a relation between breakwater type and location this relation is investigated. In figure C.3 the location of the breakwater is therefore plotted together with the type of breakwater.

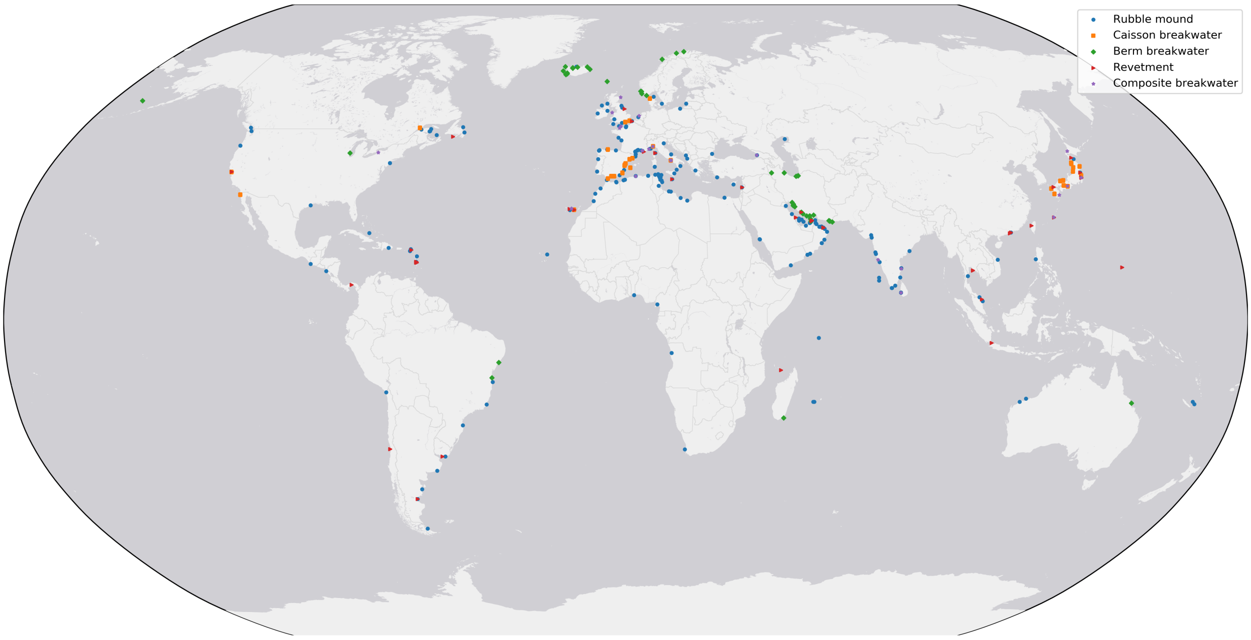


Figure C.3: Location of different breakwater types. Data from TU Delft and HR Wallingford (2019).

It can be seen that the caisson breakwaters are indeed mainly constructed in Japan and Spain. Besides the preference for caisson breakwater, the preference of the Nordic countries for berm breakwaters can also clearly be seen. This can be explained by the fact that rock is abundantly available in these countries, which makes construction out of rock often the most economical option.

However, also Iran has a clear preference for berm breakwaters, almost all Iranian breakwaters that are in the database are berm breakwaters. Whereas, no berm breakwaters have been constructed on the other side of the Persian Gulf. This can be due to the fact that rock is more abundantly available in Iran, or the political sanctions against Iran which prevented companies from doing business with Iran. But analysing the exact reason is outside the scope of this research. However, It is noteworthy that Sigurdur Sigurdarson, the developer of the Icelandic breakwater, visited Tehran in 1998 to give a presentation on the latest developments in berm breakwaters. This was right around the time the first berm breakwater was constructed in Iran, which was in 1996 (Sigurdarson et al., 2006).

While most differences are between countries, there are also differences within countries. When zooming in on, for instance, Spain it can be seen that both rubble mound and caisson breakwaters are constructed along the coast. On this map, figure C.4, the port of Barcelona is an interesting case since both types of structures are used to protect the port.

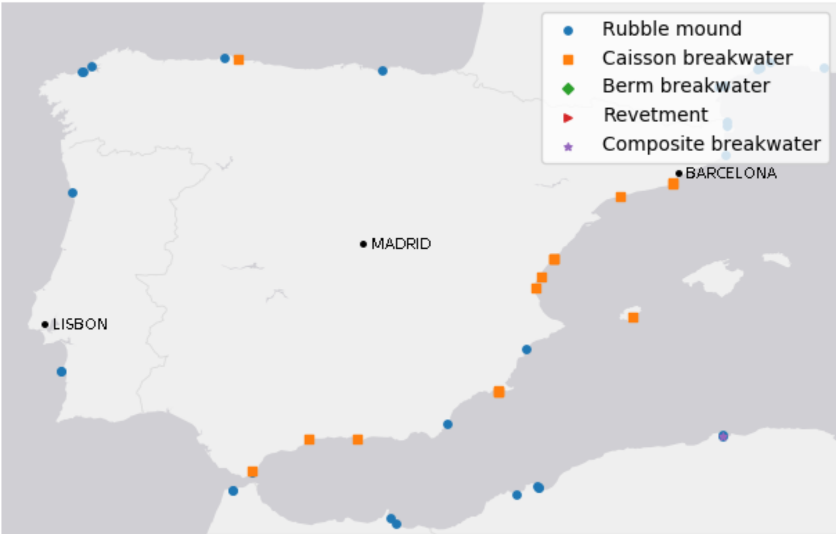


Figure C.4: Locations of different breakwater types in Spain. Data from TU Delft and HR Wallingford (2019).

In Figure C.5 a satellite image of the port is presented, with a zoomed in version of the breakwater of interest on the right hand side. The breakwater of interest is divided into three stretches, of which the design data is summarised in table C.4. Based on the findings from the previous section, that caisson breakwaters are more often constructed in larger water depths, the breakwater in Barcelona is a counterintuitive variant as the breakwater in the deepest water is a rubble mound breakwater.

A possible explanation is that the second stretch is used to moor ships, as this is a functional requirement a caisson breakwater performs better on compared to a rubble mound breakwater. Also see table 2.3 in Subsection 2.2.3. So it might be that the rubble mound was the better solution, i.e. more economical, but that the client specified a requirement for berth provisions which resulted in a combined breakwater.

Figure C.5: Satellite image of the port of Barcelona, image retrieved from ESA

Table C.4: Design parameters for the Southern breakwater of the port of Barcelona (Ramón et al., 2010; TU Delft and HR Wallingford, 2019)

Type	**Location**	**Hs**	**Maximum depth**	**Length**
rubble mound	Stretch 1	7.3	23	2000
Caisson	Stretch 2	7.6	21.5	1700
rubble mound	Stretch 3	7.3	27	1100

D

Overtopping Equations for a Vertical (Composite) Breakwater

This appendix presents the equations implemented in the design automation tool for the overtopping failure mechanism of the vertical (composite) breakwater. The equation numbers correspond to the equation numbers from the decision chart, depicted in Figure 4.4. In Section D.1 the definitions of the parameters are presented, whereafter in, Section D.2, the equations are presented.

D.1 Definition of Parameters

In Figure D.1 the definition of the water depth, h, water depth on top of the foundation d and freeboard, R_c is depicted.

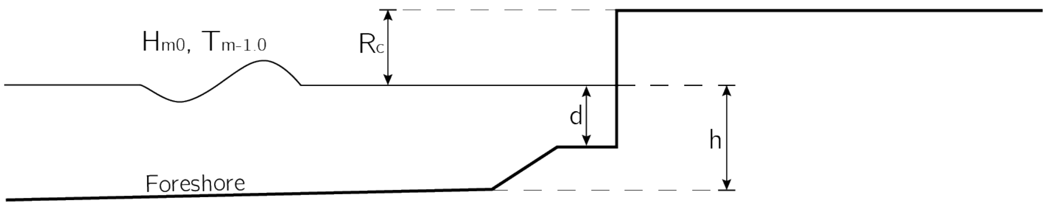


Figure D.1: Definition of the parameters used in the overtopping equations for a vertical (composite) breakwater

A vertical breakwater is classified as a vertical composite breakwater if the relative water depth, $\frac{d}{h}$ is smaller than 0.6. This definition is in accordance with EurOtop (2018) and was also introduced in Subsection 1.2.1. The definition of the other parameters used in the equations are:

q	overtopping discharge
g	acceleration due to gravity
H_{m0}	significant wave height from spectral analysis
$s_{m-1,0}$	fictitious wave steepness based on the spectral wave period, $T_{m-1,0}$

D.2 Equations

In this section all implemented equations from Figure 4.4 are presented, these equations are copied from EurOtop (2018). Note that all equations are included in the core of tool for use. However, the tool also includes a function to automatically determine and use the correct equation.

D.2.1 *Vertical (Composite) Breakwater in Deep Water*

The first step in the decision chart is to determine if the foreshore influences the waves. If the foreshore does not influence the waves, equation D.1 is used.

$$\frac{q}{\sqrt{g \cdot H_{m0}^3}} = 0.047 \cdot \exp\left[-\left(2.35\frac{R_c}{H_{m0}}\right)^{1.3}\right] \tag{D.1}$$

D.2.2 *Vertical (Composite) Breakwater without Breaking Waves*

If the foreshore does influence the waves, but there is no possibility of breaking waves the following equation is used:

$$\frac{q}{\sqrt{gH_{m0}^3}} = 0.05\exp\left(-2.78\frac{R_c}{H_{m0}}\right) \tag{D.2}$$

D.2.3 *Vertical Breakwater with Low freeboard*

If the breakwater is classified as a vertical breakwater with a low freeboard, the following equation is used:

$$\frac{q}{\sqrt{gH_{m0}^3}} = 0.011\left(\frac{H_{m0}}{hs_{m-1,0}}\right)^{0.5}\exp\left(-2.2\frac{R_c}{H_{m0}}\right) \tag{D.3}$$

D.2.4 *Vertical Breakwater without Low freeboard*

When the breakwater is classified as a vertical breakwater without a low freeboard, referred to in the documentation as normal, the following equation is used:

$$\frac{q}{\sqrt{gH_{m0}^3}} = 0.0014\left(\frac{H_{m0}}{hs_{m-1,0}}\right)^{0.5}\left(\frac{R_c}{H_{m0}}\right)^{-3} \tag{D.4}$$

D.2.5 *Vertical Composite Breakwater with Low freeboard*

If the foundation of the breakwater is higher, the vertical breakwater is classified as a vertical composite breakwater. When the breakwater also has a low freeboard the following equation must be used:

$$\frac{q}{\sqrt{gH_{m0}^3}} = 1.3\left(\frac{d}{h}\right)^{0.5}0.011\left(\frac{H_{m0}}{hs_{m-1,0}}\right)^{0.5}\exp\left(-2.2\frac{R_c}{H_{m0}}\right) \tag{D.5}$$

D.2.6 *Vertical Composite Breakwater without Low freeboard*

When the breakwater is classified as a vertical composite breakwater without a low freeboard, which is referred to as normal in the documentation, the following equation is used:

$$\frac{q}{\sqrt{gH_{m0}^3}} = 1.3\left(\frac{d}{h}\right)^{0.5}0.0014\left(\frac{H_{m0}}{hs_{m-1,0}}\right)^{0.5}\left(\frac{R_c}{H_{m0}}\right)^{-3} \tag{D.6}$$

E

Goda-Takahashi Wave Pressure Model

In this appendix the equations implemented in the design automation tool for the wave pressure model of Goda and Takahashi are presented. This model was used to compute the forces and moments required to compute the following failure mechanisms from Subsection 4.3.3:

- Sliding of the caisson over the foundation
- Overturning of the caisson
- Bearing capacity of the foundation
- Slip circle through the subsoil

The appendix is divided into two sections. In the first section the definitions of the parameters are presented, whereafter, in Section E.2, the equations are presented.

E.1 Definition of the Parameters

In Figure E.1 the definitions of the used parameters and pressures are presented.

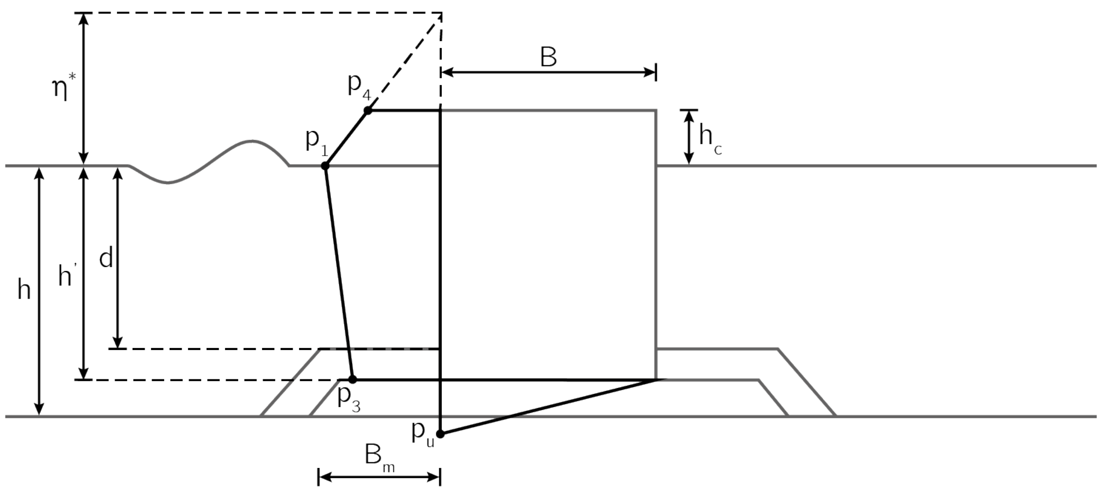


Figure E.1: Definition of the used parameters in the extended Goda formula

E.2 Equations

This section includes all equations of the Goda-Takahashi wave pressure model that have been implemented in the design automation tool. The equations are copied from Goda (2000) and Takahashi (2002).

E.2.1 *Forces*

The horizontal (P) and vertical (U) force on the caisson due to the wave pressures are given by the following equations:

$$P = \frac{1}{2}(p_1 + p_3)\, h' + \frac{1}{2}(p_1 + p_4)\, h_c^* \tag{E.1}$$

$$U = \frac{1}{2} p_u B \tag{E.2}$$

In which h_c^* is the height to which the wave pressure is exerted on the caisson, or in other words: the minimum value of h_c and η^*.

E.2.2 *Moments*

The moment due to the horizontal pressure, M_p, and uplift pressure, M_U, are given by the following equations. Note that the moments are computed from the heel of the caisson, or in other words: the lower right of the caisson.

$$M_p = \frac{1}{6}(2p_1 + p_3)\, h'^2 + \frac{1}{2}(p_1 + p_4)\, h' h_c^* + \frac{1}{6}(p_1 + 2p_4)\, h_c^{*2} \tag{E.3}$$

$$M_U = \frac{2}{3} U B \tag{E.4}$$

E.2.3 *Pressures*

The horizontal pressures due to the waves are given by the following equations:

$$p_1 = 0.5(1 + \cos\beta)\left(\lambda_1 \alpha_1 + \lambda_2 \alpha^* \cos^2\beta\right) \rho g H_{\max} \tag{E.5}$$

$$p_3 = \alpha_3 p_1 \tag{E.6}$$

$$p_4 = \alpha_4 p_1 \tag{E.7}$$

In which β is the incident wave angle, λ_1, λ_2 and λ_3 are modification factors dependent on the type of structure and α^* is the impulsive wave pressure coefficient, see Subsection E.2.5. The equations to compute the wave pressure coefficients, α_1, α_3 and α_4, are presented in the next subsection.

The uplift pressure is given by the following equation:

$$p_u = \frac{1}{2}(1 + \cos\beta)\lambda_3 \alpha_1 \alpha_3 \rho g H_{\max} \tag{E.8}$$

E.2.4 *Wave Coefficients*

The wave pressure coefficients are given by the following equations:

$$\alpha_1 = 0.6 + \frac{1}{2}\left[\frac{4\pi h/L}{\sinh(4\pi h/L)}\right]^2 \tag{E.9}$$

$$\alpha_2 = \min\left\{\frac{h_b - d}{3h_b}\left(\frac{H_{\max}}{d}\right)^2, \frac{2d}{H_{\max}}\right\} \tag{E.10}$$

$$\alpha_3 = 1 - \frac{h'}{h}\left[1 - \frac{1}{\cosh(2\pi h/L)}\right] \tag{E.11}$$

E.2.5 *Impulsive Wave Coefficients*

The impulsive wave pressure coefficient is given by the following equation:

$$\alpha^* = \max\{\alpha_2, \alpha_I\} \tag{E.12}$$

Where α_2 is given by equation E.10 and α_I by the following equation:

$$\alpha_I = \alpha_{I0}\alpha_{I1} \tag{E.13}$$

In which α_{I0} represents the effect of the wave height on the foundation.

$$\alpha_{I0} = \begin{cases} \dfrac{H}{d} & : H \leq 2d \\ 2 & : H > 2d \end{cases} \tag{E.14}$$

And α_{I1} represents the shape of the foundation. This term is computed with the following equations:

$$\alpha_{I1} = \begin{cases} \dfrac{\cos\delta_2}{\cosh\delta_2} & : \delta_2 \leq 0 \\ \dfrac{1}{\cosh\delta_1\sqrt{\cosh\delta_2}} & : \delta_2 > 0 \end{cases} \tag{E.15}$$

$$\delta_1 = \begin{cases} 20\delta_{11} & : \delta_{11} \leq 0 \\ 15\delta_{11} & : \delta_{11} > 0 \end{cases} \qquad \delta_2 = \begin{cases} 4.9\delta_{22} & : \delta_{22} \leq 0 \\ 3\delta_{22} & : \delta_{22} > 0 \end{cases} \tag{E.16}$$

$$\delta_{11} = 0.93\left(\frac{B_M}{L} - 0.12\right) + 0.36\left(\frac{h-d}{h} - 0.6\right) \tag{E.17}$$

$$\delta_{22} = -0.36\left(\frac{B_M}{L} - 0.12\right) + 0.93\left(\frac{h-d}{h} - 0.6\right) \tag{E.18}$$