



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

PETTERI HUUHKA
AUTOMATION OF CONSTRUCTION PLANNING OF A
REINFORCED CONCRETE RETAINING WALL
Master's thesis

Examiner: D.Sc (Tech.) Kristo Mela

Examiner and topic of the thesis
were approved on 26 March 2018

ABSTRACT

PETTERI HUUHKA: Automation of construction planning of a reinforced concrete retaining wall

Tampere University of Technology

Master of Science Thesis, 57 pages

December 2018

Master's Degree Programme in Science and Engineering

Major: Structural engineering

Examiner: D.Sc (Tech.) Kristo Mela

Keywords: Design automation, parametric design, algorithm-aided design, reinforced concrete structures

The work of designing structures can currently be executed completely by using computers. Much manual work involving multiple pieces of software is still required. This thesis explores the methods and prerequisites involved in the automation of the planning process as a whole.

The construction planning process consists of multiple stages. The stages include the gathering of reference data, structural design and the creation of production information. The stages are comprised out of many tasks of various types. Data transfers between different pieces of software, data processing, mathematical calculations, building information modelling, drawing production and the writing of reports are all tasks that require different kinds of actions.

In order to automate the planning process, a good understanding of all the tasks that are involved in the process is necessary. The planning process must thus be thoroughly evaluated. The evaluation of the process is however only one part of the preparation of automation. The variations in the structure in different projects, and the resulting differences in the process must also be explored. An automated construction planning system must be applicable to several situations. The system is therefore required to cope with at least the most common variations in the design of the structure.

The thesis presents methods for the automation of the planning process. Some of the methods provide a framework for data management and task coordination, but do not provide direct approaches for the automation of individual tasks. Methods that pertain to the automation of the separate tasks of the planning process are also explored.

The thesis includes a case study, which involves the creation of an automated construction planning system for an example structure. The example structure is a reinforced concrete cantilever retaining wall. The focus of the case study is in combining the building information modelling and structural design. The automated system was created using the Rhinoceros 3D and Grasshopper programmes. The possibility of expanding the automated system to include drawing and report production is also explored at a conceptual level. The pieces of software considered for the expansion are Tekla Structures and TeXworks.

TIIVISTELMÄ

PETTERI HUUHKA: Teräsbetonisen kulmatukimuurin rakennesuunnittelun automatisointi

Tampereen teknillinen yliopisto

Diplomityö, 57 sivua

Joulukuu 2018

Rakennustekniikan diplomi-insinöörin tutkinto-ohjelma

Pääaine: Rakennustekniikka

Tarkastaja: TKT Kristo Mela

Avainsanat: Suunnittelun automatisointi, parametrinen suunnittelu, algoritmiavusteinen suunnittelu, teräsbetonirakenteet

Rakenteiden suunnittelun voi suorittaa nykyaikana kokonaan tietokonetta käyttämällä. Tästä huolimatta paljon manuaalista työtä tarvitaan rakennussuunnitelmien laatimiseen, monia eri ohjelmistoja käyttäen. Tässä tutkimuksessa selvitetään edellytyksiä ja metodeja automatisoida koko rakennesuunnitteluprosessi.

Rakennesuunnittelun prosessi koostuu useasta vaiheesta. Vaiheiksi voidaan luokitella lähtötietojen keruu, rakenteen voimasuureiden laskenta ja mitoitus sekä rakennusdokumenttien laatiminen. Vaiheet koostuvat useista eri tyyppisistä tehtävistä. Datan siirto ohjelmistojen välillä, datan prosessointi, matemaattiset laskelmat, rakenteen tietomallinnus, piirustusten luonti ja raporttien kirjoittaminen ovat tyyppisiä toimenpiteitä.

Suunnittelun automatisoinnissa tärkeää on tunnistaa automatisoitavat tehtävät. Automatisoitavaksi tarkoitettujen rakenteiden suunnitteluprosessi tarvitsee siten tutkia huolellisesti. Rakennesuunnitelmien tuottamiseen vaadittavien toimien selvittäminen on kuitenkin vain yksi osa automatisaation valmistelusta. Sen lisäksi tarvitsee saavuttaa myös käsitys rakenteiden vaihteluista eri rakennusprojektien välillä. Automatisoidun rakennesuunnittelusysteemin pitää olla sovellettavissa useampaan suunnitteluprojektiin. Systeemin pitää siis pystyä käsittelemään vähintään yleisimmät variaatiot rakenteiden mitoituksessa ja ulkonäössä.

Tutkimuksessa esitellään muutamia mahdollisia menetelmiä rakennesuunnittelun automatisointiin. Jotkin metodeista ovat sellaisia, jotka tarjoavat kehyksen datan käsittelyyn ja tehtävien koordinointiin, mutta eivät yksittäisten tehtävien automatisointiin. Lisäksi tutkitaan metodeja, jotka ovat soveltuvia myös erillisten tehtävien ja tehtäväryhmien automatisointiin.

Työn osana suoritetaan tapaustutkimus, jossa luodaan automatisoitu suunnittelusysteemi valitulle esimerkkirakenteelle. Esimerkkirakenne on teräsbetoninen kulmatukimuri. Tapaustutkimuksessa pääpaino on rakenteen tietomallinnuksen ja rakenneanalyysin ja mitoituksen yhdistämisessä. Automaattisen systeemin luomiseen käytettiin Rhinoceros 3D ja Grasshopper-ohjelmistoja. Systeemin laajentamista piirustusten ja raporttien tuottamiseen Tekla Structures ja MikTex ohjelmilla on myös alustavasti käsitelty.

PREFACE

This Master's thesis was written between March and November of 2018 at the Tampere University of Technology, Finland. The examiner of the thesis was D.Sc Kristo Mela from the faculty of Business and Built Environment. I am grateful for Dr. Mela for his guidance during the creation of this thesis.

I would like to thank Ramboll Finland Oy and the head of the Bridges-department Harri Koskinen for developing and initiating my thesis of this topic. Special thanks also to Eero Särkkä and Markus Rynänen for their input and the vast amount of sage advice that they offered. I would also extend my gratitude to all my colleagues in the Bridges-unit and all of Ramboll for their useful input.

I am also most grateful to my parents, family and friends for their steadfast support during the completion of this work.

In Tampere, Finland, on November 19, 2018

Petteri Huuhka

CONTENTS

1.	INTRODUCTION	1
1.1	Background and objectives of thesis.....	1
1.2	Scope.....	2
2.	INITIAL STEPS OF AUTOMATION.....	4
2.1	The construction planning process.....	4
2.1.1	Reference data	5
2.1.2	Structural design	5
2.1.3	Production information.....	10
3.	AUTOMATION METHODS.....	13
3.1	Master model	17
3.2	Knowledge-based engineering.....	18
3.3	Algorithm-aided design	20
3.3.1	Design software	21
3.4	Optimisation.....	22
3.4.1	Optimisation of retaining wall.....	25
3.5	Robotic process automation.....	26
3.6	Machine learning	27
3.7	Summary	28
4.	CASE STUDY: CREATION OF AN AUTOMATED SYSTEM FOR CON- STRUCTION PLANNING.....	30
4.1	Utilised software	30
4.2	The operation of the automated system	32
4.2.1	Reference data	32
4.2.2	Initial data and design choices.....	32
4.2.3	Geometry generation	37
4.2.4	Structural design	37
4.2.5	Production information.....	42
4.3	An overview of the system.....	42
4.4	Testing the automated system	44
4.5	Areas for further development	51
5.	CONCLUSIONS	53
	REFERENCES	55

LIST OF FIGURES

Figure 1.	<i>A basic reinforced concrete cantilever retaining wall [11].</i>	2
Figure 2.	<i>The structural system of a common retaining wall.</i>	6
Figure 3.	<i>The construction planning process chart, reference data.</i>	11
Figure 4.	<i>The construction planning process chart, structural design.</i>	12
Figure 5.	<i>The construction planning process chart, production information.</i>	12
Figure 6.	<i>The automation potential of a task [4].</i>	14
Figure 7.	<i>The dimensions of a simple retaining wall.</i>	16
Figure 8.	<i>Representation of a master model-system [32].</i>	17
Figure 9.	<i>Representation of the KBE-based MOKA methodology [25].</i>	19
Figure 10.	<i>Graphic algorithm-aided design.</i>	22
Figure 11.	<i>Depiction of constraints. Adapted from figures 1-2, 3-3 [3] and 1.2 [5].</i>	24
Figure 12.	<i>Process chart of the automated system, Grasshopper. Areas which are not included are marked in italicised blue print.</i>	33
Figure 13.	<i>Process chart of the automated system, Production information. Areas which are not included are marked in italicised blue print.</i>	34
Figure 14.	<i>Dimensions of the retaining wall.</i>	35
Figure 15.	<i>Example reinforcement of a retaining wall.</i>	35
Figure 16.	<i>Loads that can be assigned to the retaining wall.</i>	36
Figure 17.	<i>A calculations component in Grasshopper.</i>	37
Figure 18.	<i>The calculation of load combinations.</i>	38
Figure 19.	<i>The design of reinforcements.</i>	38
Figure 20.	<i>A plane view of the retaining wall.</i>	39
Figure 21.	<i>The Grasshopper system.</i>	43
Figure 22.	<i>The ground surface meshes.</i>	44
Figure 23.	<i>An overview of the generated retaining wall.</i>	50
Figure 24.	<i>The initial and optimised cross-sections of the wall.</i>	50
Figure 25.	<i>The plane view of the initial structure.</i>	51
Figure 26.	<i>The plane view of the optimised structure.</i>	51

LIST OF TABLES

Table 1.	<i>A summary of automation methods</i>	28
Table 2.	<i>Used cost information</i>	40
Table 3.	<i>Design values, Loads and dimensions</i>	45
Table 4.	<i>Geological reference data</i>	46
Table 5.	<i>The optimised dimensions</i>	47
Table 6.	<i>Dimensional ranges.....</i>	47
Table 7.	<i>The reinforcements of the structure in 5 zones.....</i>	48
Table 8.	<i>Utilisation rates.....</i>	49
Table 9.	<i>Initial and optimised values for the cost function</i>	49

LIST OF SYMBOLS AND ABBREVIATIONS

AAD	Algorithm-aided design
AI	Artificial intelligence
BIM	Building Information Modelling
CAD	Computer Aided Design
EC	Eurocode
FEM	Finite Element Method
GUI	Graphic User Interface
KBE	Knowledge-based engineering
RPA	Robotic process automation
UR	Utilisation rate
$A_{s,req}$	required reinforcement cross-sectional area
$A_{s,req}$	required reinforcement cross-sectional area
A_s	cross-sectional area of the chosen reinforcement
e_{eb}	concrete cost for the edge beam
e_{fb}	concrete cost for the footing slab
e_s	reinforcement steel cost
e_w	concrete cost for the wall
f	objective function
f_{yd}	design yield strength of the reinforcement steel
g	set of inequality constraints
h	set of equality constraints
H_d	design value for horizontal forces
$M_{dst;d}$	sum of the destabilising moments
M_{Ed}	design bending moment
m_s	reinforcement steel mass
$M_{stb;d}$	sum of stabilising moments
N_d	design value for vertical forces
p	penalty function
R_d	design bearing resistance of the ground
R_d	design value of the shear strength of the ground
$R_{p;d}$	design value of horizontal forces that resist the sliding
V_d	design value for the shear force
V_{eb}	concrete volume for the edge beam
V_f	concrete volume for the footing slab
w_k	calculated design crack width
w_{max}	maximum allowed crack width
V_{Rd}	design capacity for shear
V_w	concrete volume for the wall
\mathbf{x}	set of design variables
\mathbf{x}^L	set of the lower boundary values of the variables
\mathbf{x}^U	set of the upper boundary values of the variables
X_d	design value of a action effect
X_{Rd}	structural design capacity
z	internal lever arm of the moment
ω	mechanical reinforcement ratio of the structure
ω_d	maximum acceptable mechanical reinforcement ratio

1. INTRODUCTION

1.1 Background and objectives of thesis

Construction planning is the process of designing a structure and producing the production information to manage the execution of the construction work. The process includes for instance structural calculations and the creation of drawings. Computers have been used for a long time in the various stages of construction planning. As the power and efficiency of computers has increased, the design process has become effectively completely computerised. Regular office programmes that enable data and text processing, as well as spreadsheet calculations have been utilised extensively. Specialised pieces of software for structural analysis and engineering calculation have also existed for some time. Drafting tools that utilise Computer Aided Design (CAD), have dominated drawing production. All such software have however mostly functioned with little interactivity with each other, which has fractured the planning process. The process also continues to require much manual human work, from generating content to processing and transferring various pieces of data. A number of the programmes provide tools to automate some of their operations, but automated systems that are able to automate the entire process from start to finish have not been in common use.

Technological advances enables automating the design process to a ever greater degree. Building information modelling (BIM) and Algorithm-Aided Design (AAD), while not entirely new technologies, are increasing their prominence. A building information model is a three-dimensional (3D) computer construct which holds much parametric data on the modelled objects, such as the material information for instance [27]. This data enables the programme to know the nature of the objects, such as differentiating between a concrete wall and a concrete slab, and handle them accordingly. Tekla Structures, a BIM programme, can for example produce dimension and reinforcement drawings automatically for concrete slab elements. Coupled with algorithmic tools that can generate the objects and plenty of the parametric data into the BIM programmes the production of models and drawings has become faster. AAD can be used for structural calculations as well, but that is not a wide spread practise yet.

The aim of this study is to examine the methods and possibilities for the computerised automation for the entire process of construction planning. These methods can pertain to the overall automatic coordination of the construction planning. More specific methods for the automation of individual tasks or task groups are also studied. The methods are examined as a literary study.

In order to evaluate the automation methods, a thorough understanding of the process to be

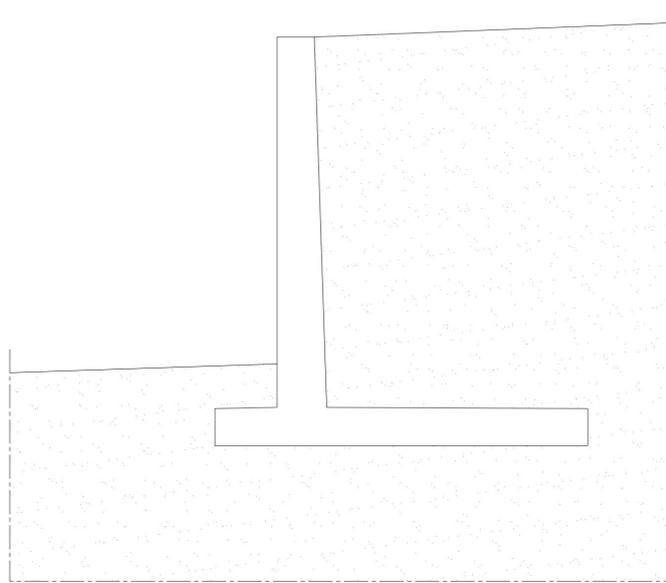


Figure 1. A basic reinforced concrete cantilever retaining wall [11].

automated is required. The construction planning processes of all structures share some similarities, but can differ greatly on a more detailed level. To enable the exploration of automation in practise, an example structure is selected. The example structure in this thesis is a reinforced concrete cantilever retaining wall. Figure 1 displays a simple version of such wall. The retaining wall was chosen as the example structure due to the relative simplicity of its designing procedure. This enables the focus on automation rather than the intricacies of the design. In order to identify methods for automation, the entire design process for a retaining wall is scrutinised in great detail. The designing process is mainly directed by the building code and other official instructions. The process examination is therefore presented based on such documents.

In this thesis a case study is also performed. The case study consists of the implementation and testing of an automated construction planning procedure for a retaining wall. A set of programmes deemed best suited for the purposes of automation are chosen for the implementation of the system. An automated design template is subsequently constructed. One of the main objectives of the case study is to examine methods to integrate modelling and structural design. The system will thus need to produce a 3D model of the structure, which also fulfils all design requirements. The possibilities to extend the system to drawing and report production is examined at a preliminary level.

1.2 Scope

The construction planning processes for different types of structures is potentially similar only on a very general level. On a more detailed level many aspects such as the calculations, loadings and drawings can differ significantly depending on the structure. Only the planning process of a retaining wall is examined in any detail in this thesis. Some general

principles of the examination can be utilised for other types of structures. The process evaluation will still have to be conducted for different structures separately.

The examination of the planning process is limited to the involvement of a structural engineer. Other work that is needed before the construction planning can begin, for example preliminary planning, geological surveys and engineering, are not included. Retaining walls, for instance, are also often built in connexion with roads, which means that the road design determines the placing of the wall, and thus has to be conducted beforehand.

Only a few aspects of the actual performing of the structural design for the retaining wall, or the use of the algorithm-aided tools are explained in detail. These are matters which are elaborated extensively in other works, so only an overview is provided in this thesis. A short introduction into parametric design and AAD is given in chapter 3. References and citations reference studies are of course made in the text when appropriate.

The geometry of the retaining wall, the conditions on individual construction sites and many other aspects vary between projects. When constructing the automated design template in the case study, an attempt will be made to take all of these variations into account. The specific structural calculations and drawing types for all possible situations are however not be constructed to completion. The template is made in a manner that it allows for correctly produced building plans in at least one type of situation. Yet the inclusion of provisions for all the variations that the wall can have, even if they do not produce finished or correct results, is important. In that way calculations for example, that are initially omitted can be incorporated later. It means however that this first version automation template will not work in all conceivable situations. The specific limitations and boundaries for the use of the automated design are listed in chapter 4, Case study: creation of an automated system for construction planning.

2. INITIAL STEPS OF AUTOMATION

Automating any work requires determining the goals of the automation and the specific tasks to be automated [20]. The goals can be increasing the efficiency of the design process or devising methods to transfer tasks to a computer. Construction planning is work that can already be performed completely by using computers. Despite this computerisation, much manual human involvement is required. The efficiency of the planning process is therefore increased by transferring the manual work to the computer. Reducing the amount of manual work aims also at an increasing quality by eliminating human errors.

Construction planning is a process of comprised of a series of tasks. These tasks include, among others, the acquisition of data, performing calculations and writing reports. In order to identify the tasks to be automated, the planning process needs to be thoroughly examined. One purpose of this examination is to obtain a good understanding of all the tasks that comprise the process. Another purpose is to evaluate what type of actions completing the task requires. The performance of the task can involve for instance producing text or laying out a drawing. Different types of actions may require different skills from humans, such as interpreting images. Different types of actions can therefore need different methods to automate them as well.

The design of retaining walls may be often similar, but is rarely completely identical. The conditions of individual construction sites and the wishes of a client regarding the appearance and the accessories of a wall, for example, cause variance in the design process. A need to change the design during the process may also arise. The differing conditions and desires pertaining to the wall may necessitate changes in the design process, and must therefore be considered. It would, after all, not be feasible to create an automated system that is only usable for one project. Sandberg et al. recommend studying past planning projects to obtain an understanding of the most commonly occurring variations [25].

2.1 The construction planning process

This section depicts the examination of the construction planning process, that is needed when automation is considered. The focus is on a reinforced concrete cantilever retaining wall, the chosen example structure of this thesis. The planning process of a wall naturally shares some traits with the planning processes of other types of structures. The design process of any structure can be divided into three principle stages, the acquisition of reference data, structural design and producing the production information. Required data and specific tasks within those stages may of course vary significantly.

2.1.1 Reference data

The construction planning of any structure commonly requires at least some data from external sources. The sources of the information can include for example surveyors, municipal authorities and utility companies. Such data is called reference data, and it is always specific to a certain construction location. Some types of reference data affect design decisions and calculations. The acquisition of the data is therefore the first stage of construction planning.

The appearance of the wall is mostly a matter of choice. It depends on the aesthetic sense of the client, the designer or both. Any accessories installed to the wall, such as lighting columns can affect the structural design of the wall. At the very least, such items must be presented in the drawings.

Geological and geographical reference data are among the most important when designing a retaining wall. Geological data pertains to the mechanical properties of the soil and other relevant factors below the ground surface. Such factors include the position of the bedrock surface and the level of the ground water. Geographical data of interest includes data on the topography of the ground, both in its initial state and the final designed state. Geological and geographical information is often available as 2D or 3D models and numerical data.

Possible items in the vicinity of the retaining wall must also be located. Other structures both above and below the ground, utilities, data cables are some examples of things that can exist near the retaining wall. Such things can affect the placing of the wall, affect the loading or impose restrictions on the dimensions the wall can have.

2.1.2 Structural design

Structural design encompasses structural analysis as well as the structural calculations. The structural analysis can be performed utilising the principles of statics. Another method of analysis is finite element analysis or some other numerical method. Statical analysis is however the more common method for evaluating retaining walls, and is therefore the only one examined hereafter.

Structural design is guided by regulations which present procedures for the structural calculations and requirements for the safety of the structures. The currently prevalent procedure for structural design in Finland is the eurocode (EC) system, a ten-part European set of standards. Some aspects, such as certain coefficients and rules, may be determined nationally, and are given in the national annexes to the eurocodes. The relevant eurocodes in designing a retaining wall are eurocode 0 (EC0), eurocode 1 (EC1), eurocode 2 (EC2) and eurocode 7 (EC7) [6, 7, 8, 9]. The standards EC0, EC1, EC2 and EC7 pertain to basic rules, loading, concrete structures and geotechnical design, respectively. The Finnish Transport Agency, a national authority, also publishes a set of eurocode implementation instructions. These instructions are titled NCCI 1 to NCCI 7, with the numbering corresponding to

that of the eurocodes. The instructions provide clarifications and additional information on the national implementation of the eurocodes, especially in regards to infrastructure [18][16][17].

Structural design begins with choosing the structural model. The model contains the structural members with their assigned dimensions, their joints and the supports of the structure. For a cantilever wall the model for the concrete structure is always the same. It consists of a footing slab and a wall that are joined together with a rigid joint, as depicted in figure 2. Occasionally an edge beam is also included on top of the wall, which is also depicted in figure 2. The support for the concrete structure depends of the

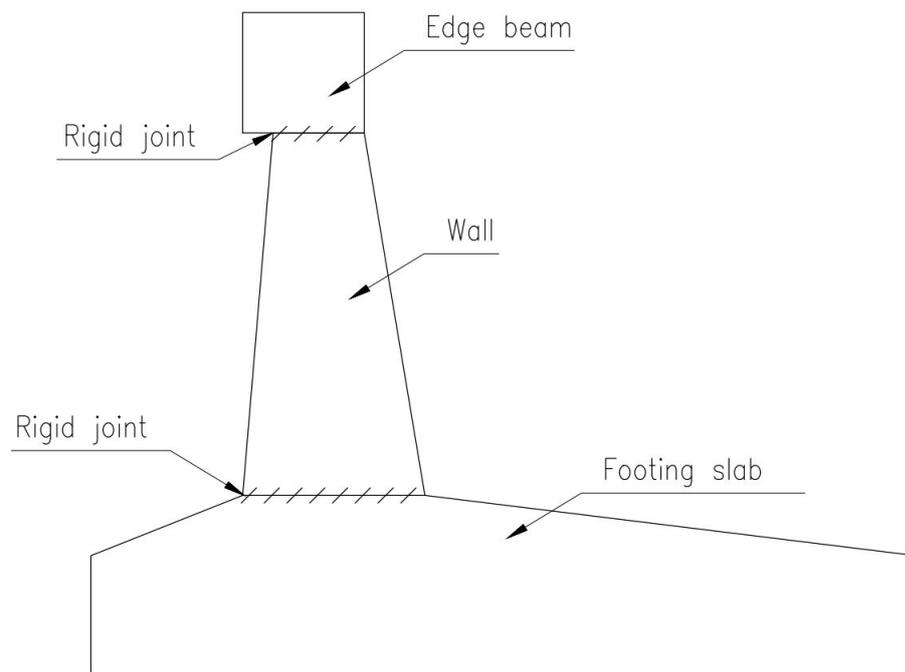


Figure 2. The structural system of a common retaining wall.

chosen foundation type. A retaining wall can be founded directly on the soil on a natural foundation bed. Another possible foundation types are a pile foundation or anchoring the wall to the bedrock with a steel anchor. The choice of foundation type is made based on a preliminary assessment of the geological properties of the site. Each foundation type involves slightly different structural calculations. Only the calculations pertaining to a natural foundation are examined in detail in this thesis.

The next task is to determine the loads imposed on the structure. The loads need also be classified, since the classification affects the manner in which the loads are combined. There are three classifications, permanent loads, imposed loads and accidental loads. Loads are also divided into favourable and unfavourable loads. A favourable load is beneficial to the stability or capacity of a structure, while an unfavourable load is not. Whether a load is favourable or not depends on the design situation [7]. The permanent loads of a retaining wall are the self-weight of the concrete structure, earth pressure and other relevant

permanent loads [11]. Self-weight and earth pressure are calculated from geometric and geographical data, other loads need to be assigned. Assigned loads that may act on a retaining wall are presented in the list below.

- Permanent surface load behind the wall
- Imposed surface load behind the wall
- A line load behind the wall
- A line load on top of the wall
- A horizontal load
- Permanent surface load in front of the wall
- Imposed surface load in front of the wall

A structure and its foundation is verified for the ultimate limit state (ULS) and serviceability limit state (SLS). In addition the accidental design situation may also have to be taken into account. The difference between these limit states is that the ULS concerns situations at the point in which any part of the structure fails and collapses. The ULS concerns also deformations and displacements that are great enough to equal collapse, rendering the structure unsafe to use. The SLS concerns situations which are detrimental to the appearance or usage of the structure, but do not result in structural failure. Examples of such situations are the cracking or deformation of a concrete surface and or the vibration of a floor surface. Load combinations are formed for the different limit states and design situations according to eurocode 0 [6].

After the action effects are ascertained for each structural member in all relevant design situations, the structural calculations can be performed. The ULS can be divided into three areas, which are EQU, STR and GEO. The EQU limit state is used to examine whether the structure retains equilibrium. For a retaining wall the EQU state is calculated by verifying that moments that act to stabilise the structure are larger than the moments that act to destabilise it. Equation (1) describes the design criterion [9].

$$\frac{M_{dst;d}}{M_{stb;d}} \leq 1 \quad (1)$$

where

$M_{dst;d}$ is the sum of the destabilising moments
 $M_{stb;d}$ is the sum of stabilising moments

The STR limit states pertains to situations in which the structure experiences internal failure or excessive deformations. The STR limit states apply to the reinforced concrete structure. The wall and footing slab are designed to withstand a bending moment and the shear force. The bending strength of a concrete structure is determined by the amount of

reinforcement that is used. Equation (2) depicts the required reinforcement for the bending moment.

$$A_{s,req} = \frac{M_{Ed}}{z \cdot f_{yd}} \quad (2)$$

where

- $A_{s,req}$ is the required reinforcement cross-sectional area
- M_{Ed} is the design bending moment
- z is the internal lever arm of the moment
- f_{yd} is the design yield strength of the reinforcement steel

The total cross-sectional area of the reinforcement bars assigned to the structure must exceed the required cross-sectional area. The design criterion can therefore be expressed in the form presented in Equation (3).

$$\frac{A_{s,req}}{A_s} \leq 1 \quad (3)$$

where

- $A_{s,req}$ is the required reinforcement cross-sectional area
- A_s is the cross-sectional area of the chosen reinforcement

A concrete structure must be designed in such a fashion, that in the event of failure, the reinforcement will yield before the concrete fails under compression. Then the fracture will be ductile instead of brittle. The fracture type is evaluated through the mechanical reinforcement ratio. The resulting design criterion is depicted in Equation (4).

$$\frac{\omega}{\omega_d} \leq 1 \quad (4)$$

where

- ω is the mechanical reinforcement ratio of the structure
- ω_d is the maximum acceptable mechanical reinforcement ratio

The structure must also withstand shear forces. The design criterion for shear is presented in Equation (5).

$$\frac{V_d}{V_{Rd}} \leq 1 \quad (5)$$

where

- V_d is the design value for the shear force
- V_{Rd} is the design capacity for shear

Additional design criteria for pile founded and anchored situations would be verifying the concrete structure for a punching force with pile and anchored foundations. In the relevant situation the load bearing capacities of the piles and the anchors also need to be examined.

The GEO limit state pertains to the load bearing capacities of the ground. Equation (6) depicts the design criterion for the vertical load bearing capacity of the ground [9]. In literature N_d is also marked as V_d , but N is used in this thesis to avoid confusion with the shear force.

$$\frac{N_d}{R_d} \leq 1 \quad (6)$$

where

N_d is the design value for vertical forces
 R_d is the design bearing resistance of the ground

The structure sliding on the ground must also be prevented by ensuring that the shear strength of the ground is sufficient. Equation (7) [9] depicts the design criterion for the sliding capacity.

$$\frac{H_d}{R_d + R_{p;d}} \leq 1 \quad (7)$$

where

H_d is the design value for horizontal forces
 R_d is the design value of the shear strength of the ground
 $R_{p;d}$ is the design value of horizontal forces that resist the sliding

With a retaining wall the SLS case that needs to be examined is the cracking of the concrete structure. Equation (8) presents the design criterion for cracking.

$$\frac{w_k}{w_{\max}} \leq 1 \quad (8)$$

where

w_k is the calculated design crack width
 w_{\max} is the maximum allowed crack width

A design is acceptable if it satisfies the all the design criteria presented in equations (1)-(8), with the exception of Equation (2). In a situation where the assigned external forces and soil properties remain constant, the choices of dimensions and reinforcements determine whether the design is within the acceptable range. The first choice of dimensions and reinforcements may not fall within the acceptable range. The result values of the design criteria equations may also be well below 1, leading to a superfluously strong structure. The dimensions and reinforcements may therefore be necessary to adjust multiple times to obtain a satisfactory result. Structural design is thus an iterative process.

2.1.3 Production information

The production information is a set of documents to direct the construction of a structure. It consists of drawings, reports and lists. The required content of the production information in Finland is elaborated in instructions given by the Finnish Transport Agency [1, 29]. These instructions are technically valid only for projects commissioned by the Finnish Transport Agency, but can commonly be used in other projects as well. This section depicts common items comprising the production information based on the aforementioned instructions. A brief explanation about these items is also given.

Construction drawings are used to convey relevant information about the structure to the construction workers. The set of construction drawings for concrete structures contains at least three types of drawings. Those are a general drawing, dimension drawing and a reinforcement drawing. A general drawing is meant to convey an overview of the structure and its immediate surroundings. The general drawing includes a longitudinal section, one or more cross-sections and a plane view. Some additional information, such as texts, are usually also included. A dimension drawing depicts the measurements of the structure. The purpose is to provide all dimensional information needed to enable the execution of the construction work in accordance with the plans. The locations of accessories and other equipment are given. Material information is given for all structural members presented. The reinforcement drawing depicts the steel reinforcement of the structure in detail. The diameters, locations, shapes and total lengths and spacings of the bars are portrayed. The shapes of bars range from straight to even quite complex forms, into which the bars can be bent.

Producing 3D building information models of structures is common. Building information models contain geometric, material and other attribute data pertaining to the structure. The construction drawings can be produced from the model.

Written reports include at minimum a work specification. A report detailing the structural design is also common. A work specification relays the requirements for the construction as well as information that can not be suitably presented in the drawings. Structural design reports present the structural analysis and calculations performed on the structure. The report is used in verifying that the calculations are correct.

A bill of quantities itemises all the materials and objects that are needed to construct the structure. It also includes quantity estimates for aspects connected to construction, such as excavations, landfills and demolitions or relocations of existing structures. The bill of quantities can be used as the basis of a cost estimate. The two can also be combined by assigning unit costs to the quantities. Cost estimation software also include the costs of construction activities. Quantities and costs can be calculated using a number of different software and web-based programmes.

A separate list of reinforcements bars, a bar bending schedule, is customarily provided. It acts in part as a bill of quantities for reinforcement bars. It also itemises the reinforcement

bars by diameter and shape. The list is convenient in the manufacture of the all the differently shaped bars.

The plan usually includes also a document listing. This listing itemises all the separate documents, usually at least the specification and drawings, that are included in the finished construction plan. This list depicts the names of the documents, their labels, dates of completion, and for drawings, the view scales used. The listing can be printed into the general drawing, as its own document or both.

The construction planning process is condensed as a process chart in three parts in figures 3-5. It is included to give an overview of the process.

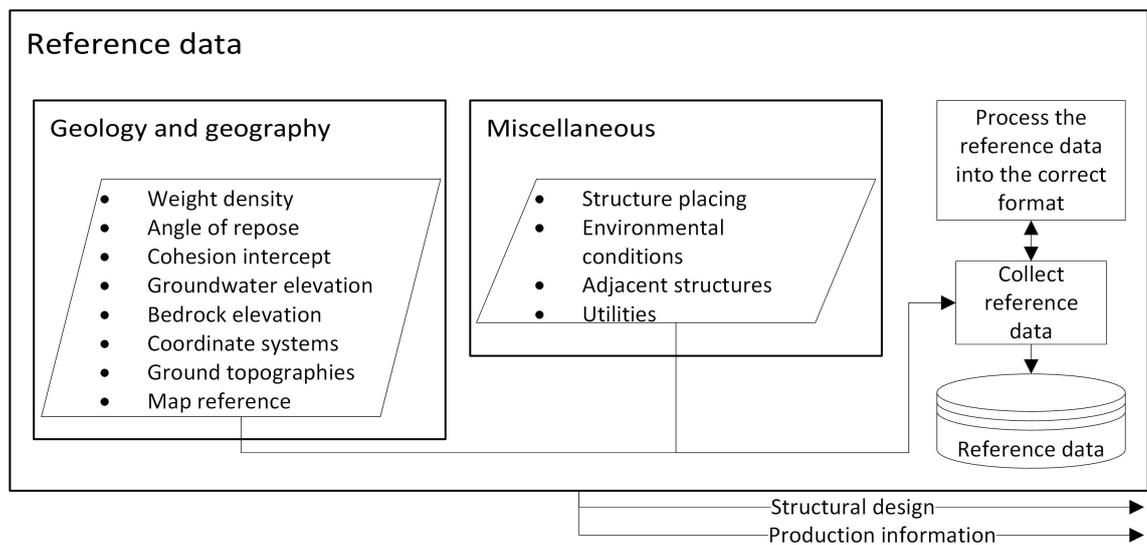


Figure 3. The construction planning process chart, reference data.

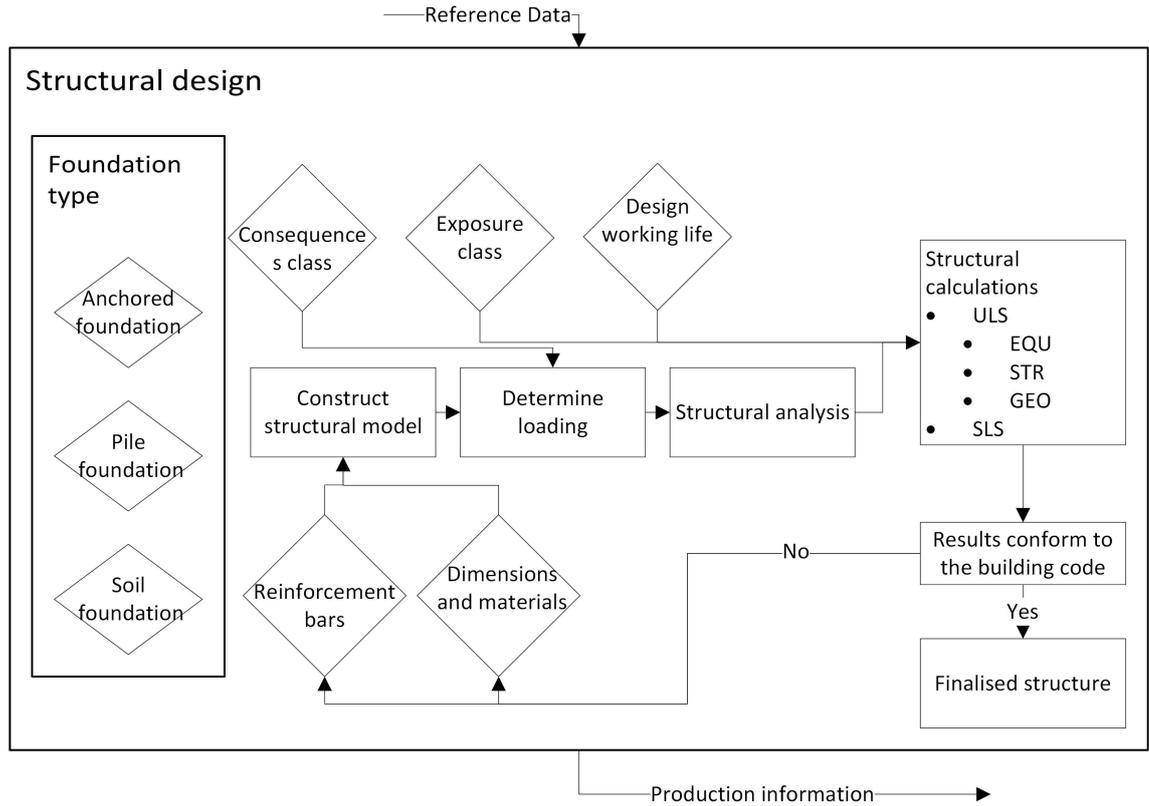


Figure 4. The construction planning process chart, structural design.

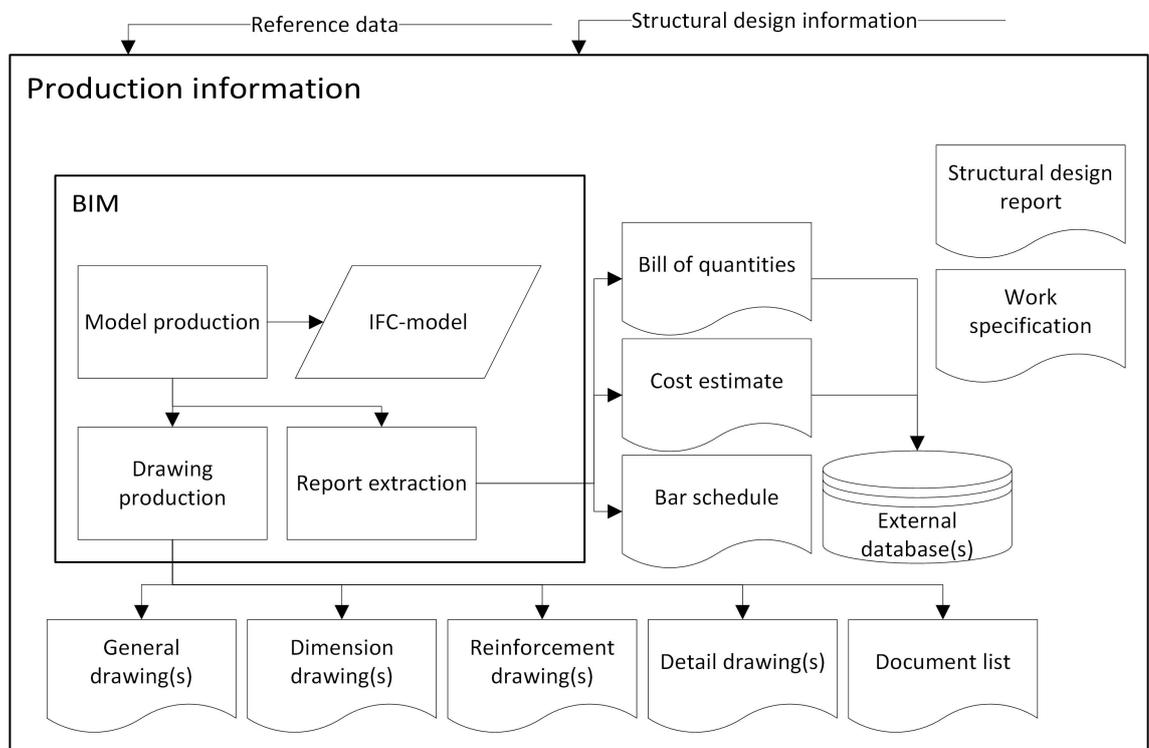


Figure 5. The construction planning process chart, production information.

3. AUTOMATION METHODS

The considerations that are required prior to automation are examined in chapter 2. A more detailed depiction is given to the designing process of a retaining wall, to the level of individual tasks. It is those tasks that are in focus when an automated system is constructed. The tasks that comprise the planning process are of various types, including for instance calculation and text production. Many automation methods need therefore to be considered, since few are versatile enough to be applicable to all tasks. As construction planning is performed by using computers, these methods pertain mainly to automating computer work. Construction planning is also performed by utilising various pieces of specialised software, such as tools for finite element method (FEM) analysis and BIM-programmes. The features and capabilities of those pieces of software are difficult and impractical to replicate into a completely original programme. That would require much time and programming expertise. Therefore the possibilities to automate the usage of the software, as well as methods to connect them into one unified system are examined. Some of the researched automation methods may concern more specialized tasks such as calculations. The main points are therefore both the usefulness of these certain methods, and also their ability to function as part of a unified system. The automation approaches presented were selected mainly on grounds of the authors previous experience and the availability of research material.

This chapter explores the theoretical basis and general features of the automation methods. Another point of interest is naturally their practical operation and implementation. That aspect is highly dependent on the software chosen for the execution of the automated system. Some pieces of software may not support all the explored automation methods. Also the effectiveness of the automation may vary between different software. Some methods also require special software of their own.

Potential automation methods are considered for all the types of tasks presented in chapter 2. That does however not mean that all the tasks are practical candidates for automation. Figure 6 offers a graphic representation of the potential of any individual task to be automated. According to the figure the tasks that are routine and manual have the greatest potential to be automated. More cognitive and non-routine tasks are understandably more difficult to automate. An assessment on whether the benefits of automation merit the efforts has to be therefore made for each individual task. This approach is also recommended by authors writing on the subject of automation [15][33][25]. In this thesis this assessment is therefore made in chapter 4, where the case study examines the practical creation of an automated system. The details of each automation method that affect the evaluation of the potential of each task to be automated is explored in this chapter.

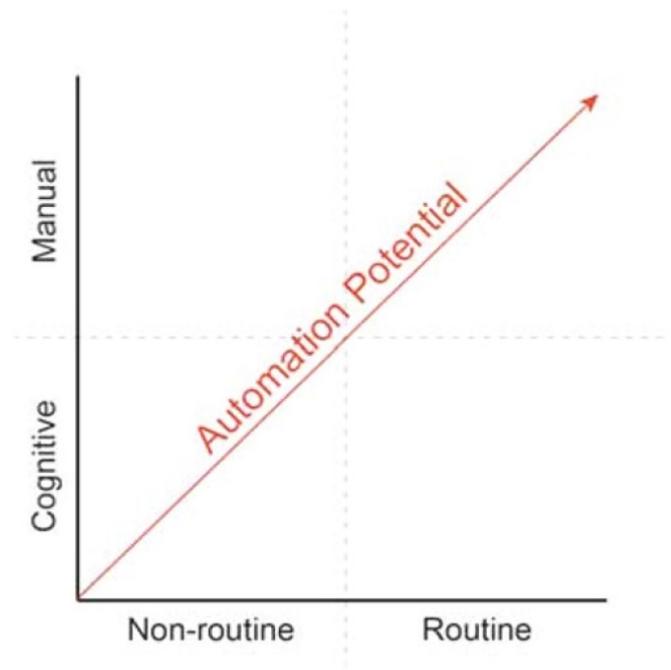


Figure 6. *The automation potential of a task [4].*

Tyson has suggested some basic guidelines for automation [32]. According to Tyson, the main criteria for an automated system are interactivity, transparency and the ability to interface. An automated system will need to be programmed to make choices independently. Occasionally it may emerge, that the choices of the system are not good or practical, even if they conform to the rules the system is given. In an interactive system the designer is then able to direct the design work by superseding the automatic choice. The interactive control of the designer should be integrated into the automation, as a system that would have to be reprogrammed for adjustments is not very practical. In this manner the designer may also be able to compensate for aspects the automation system cannot cope with. Such situations can arise in circumstances that the system has not been programmed to handle, either due to an oversight or the excessive difficulty of the programming. A built in capability for the designer to override the automation system can therefore expand the scope of applicability.

An automated system needs to be constructed in a transparent manner. Transparency in this context means that functions of the automated system can be discerned and identified with relative ease. This serves multiple purposes. Firstly it makes the inspection of the automated system more simple. That enables the easier correction of potential errors. It also makes the familiarisation to the system easier for designers who have not used it before. Transparency also aids to the interactivity of the system. A system that provides many tools for making modifications will nevertheless be impractical if their usage and effects are not apparent.

The current design process consists of disparate pieces of software that have little to no interaction with each other. As a result reference data and any data produced in the different

stages of the planning process have been manually inserted to all the various systems used in the planning. Having to insert the same pieces of data into various different software is however rather redundant, and carries the risk of errors.

Tyson suggests that the automated system to include a centralised database in order to address the issue. The database should include all the reference data as well as all the information produced during the process. Such information is comprised of selected dimensions, material information and calculation results, among others. A database, in one form or another, enables the same information to be available in all the stages of the design process. A data management system can utilise the database to coordinate the information between different software.

When automating computer work a natural starting point is to consider programming. A program is the sequence of instructions that a computer requires in order to perform a task, and programming is the act of producing those instructions. Programming is performed with a number of programming languages, of which Python is one [31]. A programming language is a formal textual language with a defined syntax, and they are used to convey the tasks a programmer desires to achieve to the computer. Every programme that exist on a computer is created in this manner. It would therefore be possible to write a computer programme that would perform all the desired tasks of an automated construction planning system. This would however require much time and programming expertise, as well as necessitate the replication of many features commercial software already possess. For these reasons it is not considered a very practical solution for design automation.

Writing original programme script is not the only method that utilises programming. Existing pieces of engineering software may include the possibility of some form of programming within their own structure. This in-programme programming is intended to enable the extension of the functionality of a piece of software beyond built-in features. Examples of software with programming capabilities include Microsoft Excel and PTC MathCad.

One application of built-in programming features is for structural calculations. Structural calculations, and statical structural analysis are ultimately rather straight forward mathematics. By using structural calculation programmes a calculations template can be created. This template is automated to a rather high extent. In a comprehensively automated template the designer has only a limited number actions to perform. Those include:

- Choosing of structural dimensions (presented in figure 7)
- Choosing material properties
- Determining reinforcements
- Determining external loads

The choices listed above have to be adjusted in the iterative process of structural design.

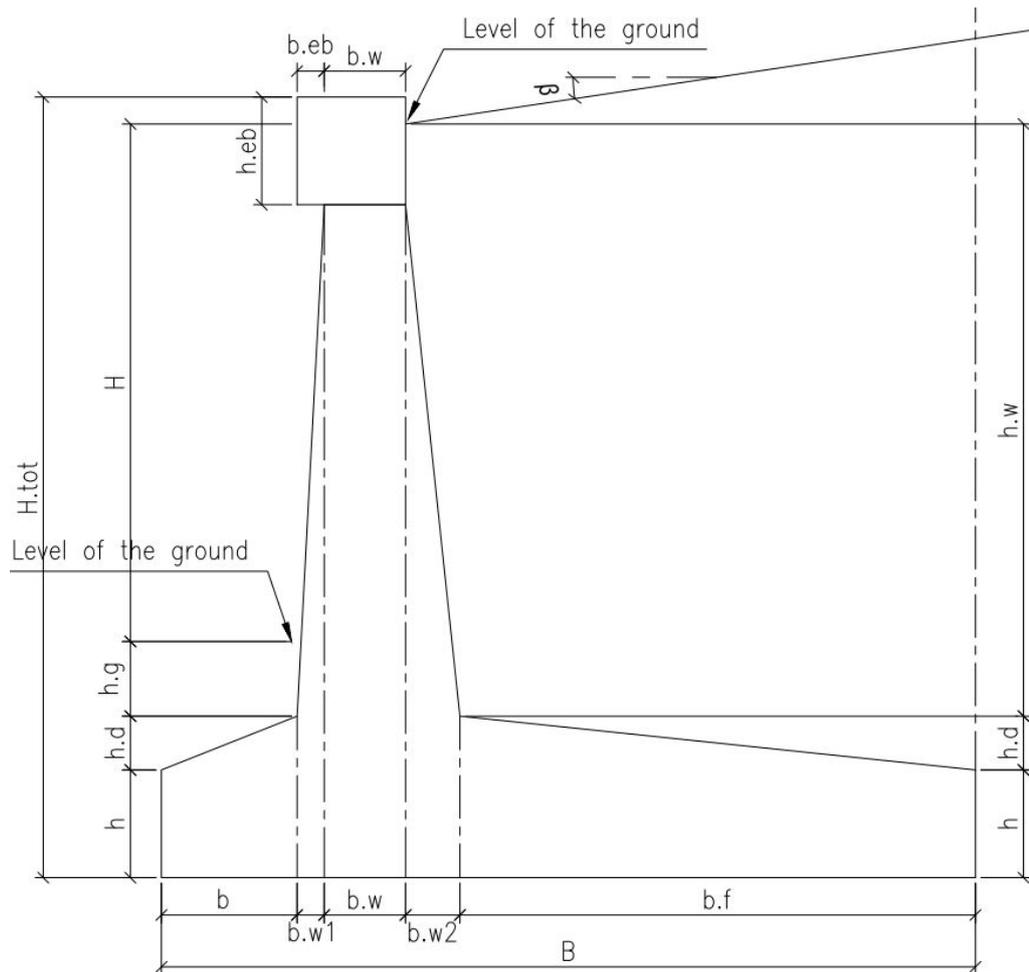


Figure 7. The dimensions of a simple retaining wall.

Challenges arise from the building codes, such as the eurocodes. Many expressions, equations, coefficients and factors are provided in the eurocodes. Many of the aforementioned aspects are also conditional. An aspect chosen by the designer, the design situation or a previously calculated value can determine conditional aspects. For example, the value of a coefficient x may be either y , z or w , depending on the value of a previously determined factor q . Such conditional variables abound in the eurocodes. On rare occasions a necessary value may also be obtainable only through the reading of graphs, diagrams or nomograms. Such conditional aspects have to be taken into account in order for a calculation template to function properly. Even small changes can potentially cause alterations into a myriad of values down the line. They also have to be handled in an automatic manner. Otherwise the designer would have to examine all the calculations in a template for all design iterations and manually assign appropriate coefficients. Such procedure would naturally be rather arduous, and contrary to the purpose of automation.

One aspect in which built-in programming of software is already utilised is structural calculations. A comprehensive template requires only relatively few inputs, and is therefore fast and simple to use. The actions that remain for the designer to perform are the insertion

of information about the dimensions of the structure as well as various material properties. The necessary dimensions for the cross-section of a wall are depicted in figure 7. External loads, if any apply, need also to be inserted. A designer is also required to determine the reinforcements for the structure. Some of the reinforcements are determined by the action effects in the wall. Otherwise the minimum reinforcement requirements given in the standards are used.

3.1 Master model

A master model is a early method to implement design automation. Master model approaches have been studied as early as 1976 [25]. A master model is described as a 3D representation of the structure as well as a central database for all the design information [32]. The model is therefore rather similar to a building information model. Unlike many current BIM programmes, a master model must have integrated connexions to software that processes the other areas of the planning process. An illustration of the principles of a master model are presented for example by Tyson [32]. This image is represented in figure 8.

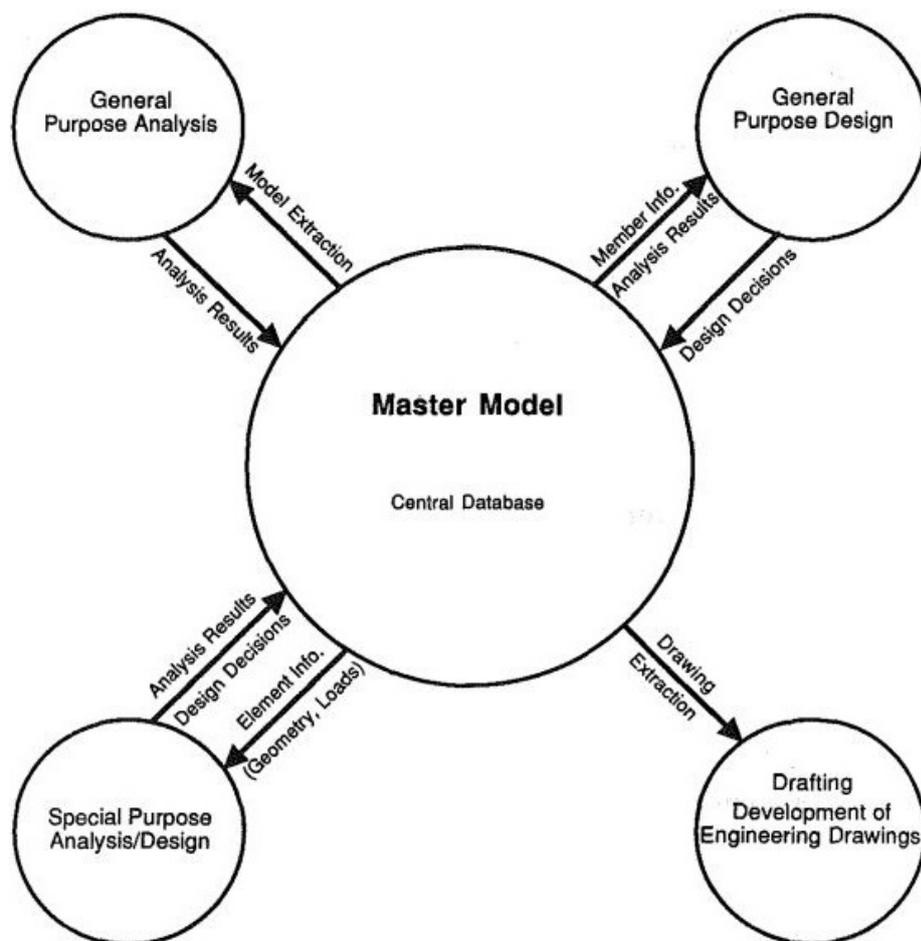


Figure 8. Representation of a master model-system [32].

The master model database would include all the design decisions made, such as the dimensions of the structure, the loads and materials among others. In the model Tyson presents, and which is featured in figure 8, the master model has general purpose analysis and design for complex structure systems, such as hall frames. The results of these processes are synchronised with the master model. The special analysis exists for the analysis of more simple and independent structural members that are not necessary to include in the complex analysis. The drafting system extracts the relevant portions of the structure for drawing development. The form of the master model system is somewhat dependent on the structure that it is used on. In the study of Tyson this is a steel framed building. For other structure types of the system would have to be adjusted accordingly.

The master model approach does not in it self necessarily address the automation of individual tasks. It still addresses one of the main points of automation, which is data management. It has been remarked before, that one of the key tasks of an automated system is to manage information in a manner that any data entered or generated by the system will be subsequently transferred to every part of the system. Tyson [32] and Sandberg et al. [25] describe a master model as a coordinating link between different clients. These clients can be people from the earliest stages to the construction phase itself. Also the different software used in the construction planning process can be viewed as clients of the master model. Any changes in one client part of the master model will be automatically propagated to all other client parts which are affected. The effect the changes have on the operation of the clients is then evaluated either automatically or manually.

3.2 Knowledge-based engineering

Knowledge-based engineering (KBE) is a method for automating design work by computerising engineering knowledge [25]. The main application of KBE is to automate repetitive and routine design work [34]. The method is not exclusive in the construction industry, and has been utilised in the manufacturing field as well. According to La Rocca, the origins of KBE were mainly in aerospace and automotive industries [24]. The term has subsequently passed to more general use within a variety of engineering fields. The KBE method has been used in connexion to construction as well. The application of KBE has been studied for example in relation to prefabricated timber houses [26].

The definition of knowledge-based engineering is to acquire and formalise human engineering knowledge [25]. The formalised knowledge is then utilised in design automation systems that run on computers. One of the principle purpose of a KBE system is also to store the obtained engineering knowledge for further use in other projects. It is after all important that the same engineering knowledge would not have to be gathered multiple times. A KBE system will also make any such information more easily accessible to other engineers [34]. The types of knowledge to be included in the KBE system are for example rules for geometric dependencies the requirements of the building code. Other possible knowledge types are rules-of-thumb, common sense and tradition from an engineering

point of view [26].

At least two methods exist for the formalisation of the engineering knowledge. These are rule based and frame based [24]. The rule based systems operate by conditional rules given to various situations. The rules have an if-then structure and can be viewed in essence as conditional programming. The other method for formalising knowledge is a frame system, which consists of frames that contain many slots. The slots are akin to cells in a spreadsheet and contain various pieces data, rules or references to other frames for the data. A frame can for instance contain the attribute information of individual product parts, or in construction, structural members. With the interdependencies and amalgamations of frames, complex data structures can be created. Frame based systems are thus, according to La Rocca, more sophisticated [24].

Knowledge-based engineering can be developed by specialised software applications [24]. These software systems provide a framework upon which the KBE system can be built. In most instances the actual KBE system will have to be created by using programming. Multiple programming languages that are created specifically for KBE exist [24]. The utilisation of KBE will therefore require programming proficiency. The system also needs other applications, such as geometric displays for viewing the models. This makes KBE similar to a master model system.

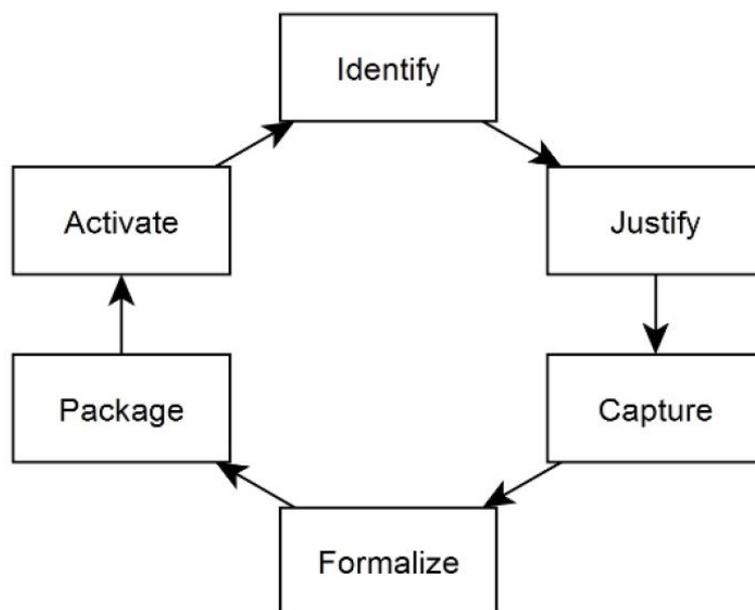


Figure 9. Representation of the KBE-based MOKA methodology [25].

Figure 9 displays a well established method for the creation of a KBE system. The method is called MOKA, which stands for “Methodology and software tools Oriented to Knowledge-Based Engineering Applications.” [34, p.9] Sandberg et al. provide a concise explanation of the method [25]. The method begins with the identification of objectives and the extent and a preliminary technical specification. The identification is then justified,

usually commercially, and possible risks are considered. The necessary knowledge is next captured and subsequently formalised in to a usable format. Packaging means producing the programming code for the application. The system is initiated for use in activation. MOKA is not the only methodology for KBE system development [34].

The necessity for software development restricts the usability of KBE systems [34]. Simple design tasks may be more effective to automate with less demanding systems. The importance of a clearly defined design process and available design knowledge are also emphasised.

3.3 Algorithm-aided design

An algorithm is defined as a series of detailed executable commands or tasks. An algorithm is always created to achieve a certain outcome, and it defines a finite process. As such, an algorithm does not relate only to computers, but can be applied to thought processes as well [28]. An algorithm is a term that is found in various contexts throughout computer science. Algorithms are also in use in various mathematical and computer applications. Algorithm-aided design (AAD) is a broad term that can refer to any type of designing work that involves algorithms, akin to computer aided design (CAD). The terminology is however not entirely consistent, and individual variations in the meanings of the terms exist [14]. For example CAD can mean any design performed by computers, or merely 2D drafting. The definition used for AAD is presented by Humppi and Österlund [14] as a script based method of creating 3D models. This script can be written in a textual form, making it akin to traditional programming. The form of AAD explored in this thesis, however, involves graphic scripting.

Graphic AAD is in its essence visual programming. Like traditional programming, visual programming languages are formal, only with a visual syntax [22]. According to Preidel et al. [22] visual programming has advantages over traditional programming. When a script is arranged into graphic elements, they are easier for a human to construe. Being easier to understand, visual programming languages are simpler to use. It also lowers the the threshold for a person to utilise programming. Visual programming can however lack the ability to perform certain kinds of program structures, such as loops [22].

Thus far AAD has been used mostly to model structures in BIM [14]. Many programmes that utilise AAD have indeed been created as extensions to modelling software. As such, many of the algorithms built in the programmes involve the processing of geometric data, such as points and surfaces. Yet they possess diverse tools and preset algorithms for mathematical calculations as well. Many extensions also exist to AAD software itself, which enable FEM-calculations, for instance. That makes algorithm-aided structural design possible.

Parametric design is a concept that is related to AAD and BIM. Parametric design is a concept that can be divided into two areas, which are parametric information modelling and

parametric structural design. The former refers of course to BIM. Many BIM programmes, such as Tekla Structures utilise parametric modelling, in which the various modelled objects are interdependent of each other. Parametric structural design encompasses principally the calculations needed to verify structural integrity. They can be used both separately and together.

In parametric design the parameters are divided into two groups, parametric properties and parametric variables. A parametric property is a term used of the attributes of a structure which has dependencies with other parametric properties. Most of the aspects connected to a retaining wall are parametric properties, all the dimensions, for instance [10].

A parametric variable is a value or attribute assigned to a structure, or a part of it, that does not have an effect on other aspects. In other words, parametric variables are completely independent on any of the other aspects of the structure, and the other parts of the structure are independent of them. For example, if the retaining wall were to be painted, the colour of the paint would be a parametric variable. It has a value that can be decided, but it has neither an effect on, nor does it derive from anything other than perhaps the aesthetic sense of a designer. The paint type itself would however be a parametric property, as it is determined by the material of a surface and estimated weather conditions, and might be taken into consideration when assessing the durability of the structure. Other examples might include the name or names of the project, which feature in the reports.

The capability to manage parametric data is a key aspect of BIM. AAD programmes are used to generate the building information models into the BIM software, which necessitates the handling of the parametric data in the AAD system as well.

3.3.1 Design software

Design software that utilise algorithm-aided design are among the main points of interest in this thesis. Pieces of software have been developed to create building information models through AAD [14]. For this reason the programmes explored in this thesis have all been developed in connexion to BIM software. Some examples of AAD software are listed below, with the BIM programmes to which they are connected.

- Grasshopper, parent software Rhinoceros 3D
- Dynamo, parent software Autodesk Revit
- Generative Components, parent software Microstation

All the AAD software listed above have graphic user interfaces (GUI). This graphic user interface is rather similar between different software. Figure 10 pictures the GUI of the AAD programme Grasshopper. A graphic interface makes using AAD tools rather easy to master. It does not involve programming experience or great mathematical skill. The AAD

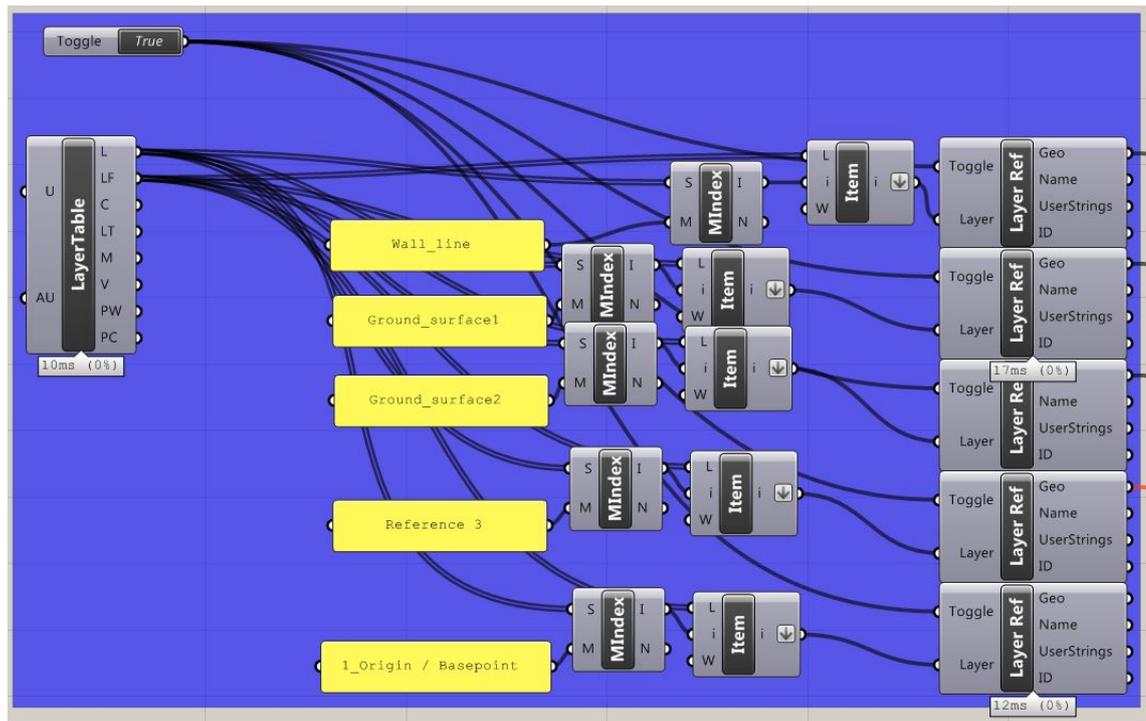


Figure 10. Graphic algorithm-aided design.

programmes have a set of predefined algorithm components. These blocks have inputs and outputs that are joined with lines, which represent a transfer of data, as can be seen in figure 10. An individual component usually performs a very specific function, such as creating points at predetermined distances along a curve. The functionalities of these predefined components range from the very simple, such as addition, to more complex operations. The components can also be combined to achieve some objective that cannot be done with the predefined algorithms. The segmented manner in which the combined algorithm is created can make it slightly convoluted.

AAD software has been developed to aid in the generation of 3D models. It is possible however to apply AAD to a more expansive use. The capabilities of AAD software depend on the components available. The library of these components can be, and is, expanded by further software development [22]. Grasshopper has, for example, an extension called Karamba, which enables FEM analysis within the software. These extensions and the diverse tools that they provide enable structural design and data processing to be performed within the AAD software. New components for interoperability with other pieces of software can also be developed.

3.4 Optimisation

Structural analysis is described in chapter 2 as an iterative process. The designer chooses the structural properties, and adjusts them between iterations until an appropriate solution is found. While a designer may aspire to find a result as close to an optimum solution

as possible, this procedure can not be called optimisation. The reason for that is that the modifications are not made methodically, but are rather determined by the personal preferences and the instinct of the designer. This procedure progresses essentially through trial and error [3]. The end point of the iterative process is also often determined by the very subjective criterion of “good enough”.

Optimisation is a distinctive term that refers to a systematic computerized search for an optimum solution. This optimum solution that is searched for is the maximum or minimum value of some aspect that can be expressed mathematically. Belegundu and Chandrupatla present the general form of the equation of a minimisation problem [5]. The equation of the general form is reproduced in Equation (9).

$$\begin{aligned}
 & \text{minimize} && f(\mathbf{x}) \\
 & \text{subject to} && g_i(\mathbf{x}) \leq 0 && i = 1, \dots, m \\
 & && \text{and} && h_j(\mathbf{x}) = 0 && j = 1, \dots, l \\
 & && \text{and} && \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U
 \end{aligned} \tag{9}$$

where

- f is the objective function
- \mathbf{x} is the set of design variables
- g is the set of inequality constraints
- h is the set of equality constraints
- \mathbf{x}^L is the set of the lower boundary values of the variables
- \mathbf{x}^U is the set of the upper boundary values of the variables

The objective function f represents some computable property that is to be optimised. In structural design this can be one of a number of structural aspects, such as the total volume and the total weight. In practical terms the favoured target for optimisation however is the cost. The objective function is even often called simply the cost function [3][5]. Yet the costs can be also calculated with varying precision. The precision depends on what aspects are considered when the expenses are computed. The costs of individual building materials are usually tabulated, but various auxiliary costs of the construction work may be included as well.

The vector \mathbf{x} denotes all the design variables affect the value of the objective function. In a concrete structure the variables to be optimised could be the dimensions and the reinforcements. Additional variables may be determined, if deemed relevant. The design variables for a structure are often discrete. They obtain values that are integers or part of a defined set. The number of reinforcement bars is always an integer, for example. The dimensions of a structure are inherently continuous, but values that are fractions of millimetres could not be accepted. Dimensions can therefore be assigned only discrete values as well.

A combination of design variables is created by assigning a value to each member of the vector \mathbf{x} . The solution of the objective function is the value the function produces with

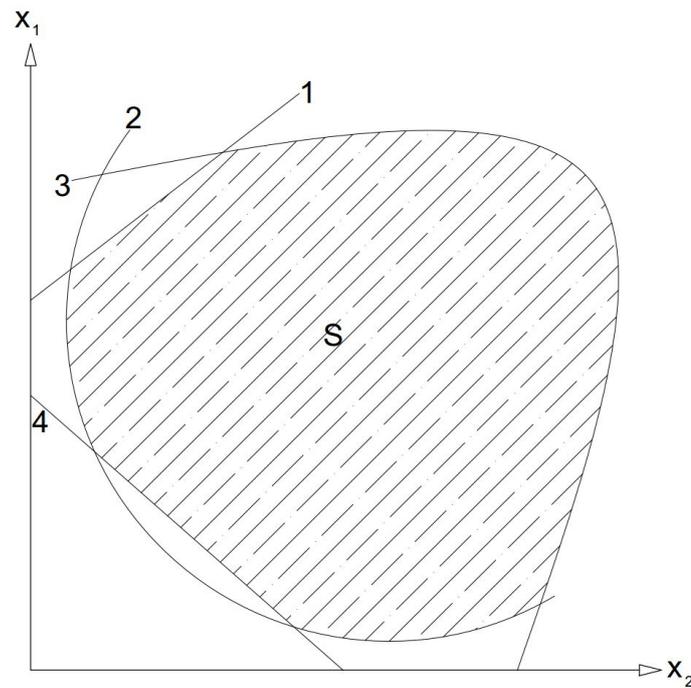


Figure 11. Depiction of constraints. Adapted from figures 1-2, 3-3 [3] and 1.2 [5].

one combination of design variables. The number of variable combinations, and thus the number of solutions may be very high. Yet not all those solutions will be acceptable. The constraints define the feasible set of acceptable solutions. The effect of the constraints is illustrated in figure 11, which shows a feasible set S bound by 4 constraint curves in a situation with 2 design variables. The principle constraints in structural design are those imposed by the design requirements, as explained in chapter 2. The constraints that derive from the building codes are mostly inequality constraints. Others may be required to be determined depending on the situation.

An iterative optimisation process proceeds in a systematic, rule based manner. The purpose is not to calculate the value of the objective function with every possible combination of the design variables, and then choose the lowest or highest result. Due to the potentially high number of those combinations, the computational time may be very high. The optimisation process includes therefore rules designed to guide the solution towards the optimum, thereby limiting the number of iterations. The process usually begins with some initial guess, which is a combination of design variables chosen either arbitrarily or based on deduction or experimentation [3]. The rules of the optimisation process will then define the size and direction of the iterative step. The step therefore determines, how much and in which direction the values of the design variables are altered between iterations.

The rules by which the optimisation process proceeds are defined by the used optimisation method. Special numerical methods have been developed for the search of the optimum solution. These methods are called either directly optimisation methods, or optimisation algorithms [3]. Examples include genetic algorithms and the quasi-Newton method.

Optimisation algorithms are mathematical constructs, though many draw inspiration from real-life phenomena. Genetic algorithms, for instance, are derived from actual evolutionary mechanisms of living creatures. Many algorithms have been developed to address multiple different situations, since not all methods are suitable for all types of problems. Further development is also ongoing. A suitable optimisation algorithm should determine the step and direction of the iteration in such a manner that the result converges on the optimum solution [3]. The algorithm includes some stopping criterion as well. This criterion determines when the algorithm interprets it has reached the optimum solution, and the end of the process.

The use of optimisation in automation is to remove the human involvement in the iterative search for an acceptable design solution. In traditional structural design a designer must perform the iterations and check the results after each one personally. This can be rather time consuming even with a comprehensive template as discussed earlier. Searching for the solution through optimisation transfers this time consumption from a human to a computer. Another benefit is the reduction of construction costs. The use of optimisation with a cost function will introduce other tasks however. Cost information is not constant, but changes over time due to economic fluctuation. In order to maintain the accuracy of the optimisation results, the cost data must be kept up to date.

3.4.1 Optimisation of retaining wall

This subsection explores some aspects of optimisation pertaining to cantilever retaining walls. Optimisation specifically as it pertains to cantilever retaining walls has been studied by Gandomi et al.[12] as well as Pei and Xia [21], among others. These studies concentrate on simplified representations of retaining walls. The structural systems of geometry and reinforcement are also not identical to the type of retaining wall that is explored in this thesis. Some salient points can however be extracted.

Pei and Xia use a cost function which includes the costs of concrete and steel [21]. The used cost function include any labour costs. Pei and Xia do subsequently recommend having a more extensive cost function, which also takes aspects such as labour costs, and other auxiliary costs into account. Gandomi et al. test two objective functions, one for cost and one for total weight [12]. The cost function does include labour costs as well. This suggest that calculating the mere expenses of materials may not be the best choice for an objective function. Between the two studies 6 optimisation methods were employed. Gandomi et al. utilised differential evolution, evolutionary strategy and biography based optimisation. Pei and Xia used a genetic algorithm, particle swarm optimisation and simulated annealing. Out of the methods used by Gandomi, biography based optimisation was ranked best, while the genetic algorithm and particle swarm optimisation were found suitable by Pei and Xia. A wider study would have to be conducted to determine the a more globally preferable optimisation method. A point of interest is however that Pei et.al found the Simulated annealing system to be unsuitable for their problem [21].

3.5 Robotic process automation

A robot is a term perhaps more closely associated with physical, mechanical robots, such as those used by industry. The term robot is however applied to some types of software as well. Robotic Process Automation (RPA) is a method which involves specialised pieces of software [33]. As the name suggests, the explicit purpose of RPA is to automate work processes performed on a computer. The approach adopted by the RPA method is to emulate the manner in which a human performs computer work [33]. This means that the software performs computer functions through the user interface similarly to a human [4]. In this manner the RPA-software can use various other pieces of software or internet based services, for example. It can also be used to connect the processes of the other programmes together [33].

The use of RPA has hitherto been considered mainly for business purposes, as evidenced by the articles pertaining to it. The matter has been discussed by van der Aalst et al. [33], Lacity and Willcocks [15], among others. RPA has been utilised thus far mainly to automate computerised support functions in a business environment [15]. The industries in which RPA has been adopted include banking, telecommunications and healthcare providers [4] [15]. No research was found to indicate that the RPA technology has previously been considered for use in the construction industry. The processes that the articles reference as being automated through RPA are rather simple and repetitive. The tasks that are transferred to RPA-software recur also numerous times during a given time. Usually the tasks are of a nature, that they may require hundreds of employees to perform [15].

The purpose of RPA is the automation of computer work. The work involved in construction planning may differ much from those industries in which RPA has been utilised. Yet the automation of construction planning in a large part the automation of computer processes, as has been discussed before. A main aspect of automation in the design process also involves connecting various pieces of software into a system that can function uniformly.

RPA-software functions in accordance with the instructions that it has received. These instructions are given in a different manner than traditional programming. One of the benefits of RPA-software therefore is that its use does not require programming experience. Automating the computer processes is therefore conducted by the personnel that has most experience in the task.

It is suggested by van der Aalst et al. [33] that the versatility and applicability can be expected to increase. The computer science principles of Artificial Intelligence and machine learning are referenced as possible approaches to develop RPA further. The objective of the development is to increase the ability of the software to adapt to changing circumstances. Artificial intelligence and machine learning are the topics of the next section. Development in the method of robotic process automation may change its utility for structural planning.

3.6 Machine learning

Artificial intelligence (AI) is a field of study that researches computer systems that emulate human thinking [13]. It can be viewed as a subset of computer science, but it involves multiple other scientific fields as well. Machine learning is one branch of the overall field of AI, but it is the focus of this study [23]. Michalski et al. define three focus points of machine learning [19]. These areas are:

- Task-oriented studies
- Cognitive simulation
- Theoretical analysis

Cognitive simulation studies methods to investigate and simulate the manner in which a human learns. Theoretical analysis strives to search for a wide variety of possible task independent learning methods. Task-oriented studies examine methods to apply machine learning to specific activities [19]. The task-oriented approach is of interest in this study.

Traditional programming requires that the task to be performed is explicitly defined. Only then can the programme script be formulated. A programmer must also attempt to foresee the possible variations the programmed process may experience. Machine learning is a method in which the computer formulates the programme for the desired task automatically [2]. This is necessary if a problem is difficult to define, or a method for finding a solution is difficult to envision. In such situations the machine learning method can be used to devise the program automatically [2].

This applicability of machine learning is widespread, from facial recognition to enabling the operation of autonomous vehicles [30]. Generating natural language texts is also an area to which machine learning is applicable [35]. Machine learning is evaluated for use in the production information-stage.

A machine learning system will need much data to function [2]. This data serves as a reference point for the system to learn from. An example is given by Alpaydin [2]. In the example, a spam email filter is examined. The filter needs to extract and remove unwanted messages from the email stream. This would be difficult to do with traditional programming, since the variability in the texts is so large. A machine learning system however can examine multiple, perhaps even thousands of messages that have previously been identified as spam. The system would then identify common traits and markers of the unwanted messages and create a procedure to filter them from the other messages.

A similar approach can be used with written reports. It would be difficult, though not impossible, for a designer to account for all the various changes to the text in different situations. Yet an engineering company may well have possession of many versions of such reports from previous projects. A machine learning system could be used to identify the various sections of the reports, and how they change in different situations. A programme

can then be created to generate the reports. A large enough database of past projects can improve the accuracy [2]. A similar procedure can be utilised for drawings, where the machine learning system would use multiple past drawings to examine their features.

3.7 Summary

This section offers a brief summary of the key automation principles and all the automation methods presented in this chapter. An automation method is meant to perform tasks with limited need for human involvement. It is also necessary to manage the data, which means its transfer and storage between all the stages of the automated process.

Table 1 displays the automation methods presented in this chapter. The table also includes the aspects of construction planning for which these methods are suitable.

Table 1. A summary of automation methods

	Programming	Master Model	KBE	AAD	Optimisation	RPA	Machine learning
Data management	*	*	*	*		*	
Data processing	*	*	*	*			
Structural analysis	*	*	*	*	*		
Structural calculations	*	*	*	*	*		
Model generation	*	*	*	*			
Drawing production	*	*	*				*
Report creation	*	*					*

Programming is the baseline of automation methods. Programming in one form or another is after all the method by which all computer applications are constructed. Thus programming can be seen as a method that is able to perform any task. Some form of programming is also present in all the other automation methods that were examined.

The master model is based on a central data model which stores and allocates all the relevant construction data. The model is however more of a method to coordinate the data flow and the different design aspects from analysis to drafting. The master model does not directly automate the performance of the different tasks.

The KBE method details a procedure of identifying, gathering and implementing knowledge for structural design processes. The purpose of the method is also to enable the reuse

of the information. The KBE is similar to the master model approach in that it combines and coordinates between different design aspects. Implementing KBE requires proficiency in the special programming languages, which demands expertise.

AAD is a method developed primarily for script based geometric modelling. This scripting can be performed with visual programming languages that makes it easier to adopt. The applicability of AAD extends to data processing and structural design.

The RPA software performs computer tasks in a similar manner to a human. It can utilise the same user interface as a computer user. The technology has been employed for example in data centres, to make data entries into various databases. RPA requires special software to operate, but the system instructions can be produced without programming abilities. Its applicability to construction planning tasks has not been researched, as far as the author knows.

Machine learning, an AI field, can be utilised in situations that are otherwise difficult to programme. Machine learning has been previously utilised in pattern recognition and text generation. The method may be useful in construction drawing generation and report generation.

4. CASE STUDY: CREATION OF AN AUTOMATED SYSTEM FOR CONSTRUCTION PLANNING

The case study consists of the creation of an automated construction planning system for a reinforced concrete cantilever retaining wall. The purpose of the automated system is to automate a number of the tasks involved in the construction planning process which are presented in chapter 2. The automation involves some of the automation methods discussed in chapter 3. The automated system incorporates tasks of the planning process from all the three stages, reference data processing, structural design and the creation of the production information. Priority was given to the automated creation of the 3D-geometry of a retaining wall and the structural calculations. This was seen as a sufficient base from which the automated system can subsequently be expanded and augmented. This chapter provides an overview of the function of the system, and some of the main points that had to be considered. The functionality of the system was also tested with by performing the design of an example structure. The test is described in detail later in this chapter.

4.1 Utilised software

The automated system is created with existing commercially available software. The utilised programmes are listed below, and this chapter provides a brief introduction to each of them. The pieces of software were chosen primarily for the relevant functionalities they possess in relation to construction planning. Another consideration was their familiarity to the author.

- Rhinoceros 5 3D
- Grasshopper, a Rhinoceros plug-in
- Tekla Structures
- TeXworks
- Microsoft Excel

Rhinoceros 5 3D is a 3D modelling programme. It has the ability and tools to read various model files as well as generate, modify and display them. For the purposes of the automated system, however, the main interest of Rhinoceros is as the host of the Grasshopper-programme. Grasshopper is a plug-in of Rhinoceros, that does not have the capacity to operate independently. Grasshopper itself also has numerous additional parts which extend its capabilities. Grasshopper is a programme that utilises the principles of algorithm-aided design. It has a graphic interface and operates on the principles of visual programming, that were explained in chapter 3. Being developed to aid in 3D modelling,

Grasshopper has numerous existing components for the generation and processing of various types of geometrical data. The programme has also mathematical tools that can be used for engineering calculations and an additional part that facilitates FEM analysis. Grasshopper contains an integrated tool for optimisation called Galapagos as well. The software has also facilitates the use of traditional programming, for example with the Python programming language. That enables a designer to extend the functionality of the programme beyond that provided by the existing components.

Rhinoceros 3D is a modelling programme that can also produce drawings. It is however a more generalised modelling software, which is why building information modelling is performed with other programmes. Tekla Structures is specifically a BIM programme, and possesses the appropriate features for such work. The software has utilities for creating and modifying 3D-geometries for various types of structures. Tekla can be used to generate the drawings of the modelled structures and to generate various reports about them. Grasshopper possesses a direct link to Tekla. The geometries generated in Grasshopper can therefore be directly transferred into Tekla Structures. The Tekla-link components enable also adding all the parametric data which is involved in BIM to the structural members.

The TeXworks software utilises the Latex typesetting language. The Latex system is a code-based method for generating texts. The code can be created with designated programmes such as TeXworks, or it can be produced as plain text in a simple text file. The Latex typesetting system includes a separate document class file, which dictates the general appearance of the text. Additional rules are set by a user with the scripted commands in the text. The system then produces the layout of the text. Grasshopper possesses components for the production and processing of texts as well. The resulting Latex-code can be exported from Grasshopper into a format TeXworks can use.

Microsoft Excel is a common office programme. It is a spreadsheet software that has extensive capabilities for the processing of numerical and textual data. It can also be used for mathematical calculations. Grasshopper possesses components that can directly read and write Excel-documents.

The principle software for the automated system is Grasshopper. It is where the model generation and structural design of the retaining wall is performed. Additional programmes are required to encompass other stages of the planning process. As the software descriptions indicate, however, these pieces of software can be linked to the Grasshopper template. In this respect the produced automated system bears a noteworthy similarity with the Master model and KBE approaches described in chapter 3. The structural model is generated by Grasshopper and is therefore contained within the file. The system also acts as a database for design data and coordinates the flow of information between different programmes, as a Master model does. One of the main principles applied to the automated system is the retention of data. According to the principle, any data inserted in, or generated by, the automated system should automatically manage its use. In essence, any external data once inserted, should subsequently be transferred, processed and utilised wholly within

the automated system. The same applies to any data that is generated by the system. The Grasshopper template serves in this capacity. The KBE method centres on the acquisition, formalisation and implementation of engineering knowledge. The same is done in the Grasshopper system, in which the requirements of the eurocodes are formalised with the visual programming language. The system can include in the same fashion also other knowledge, such as the customs and traditions retaining wall design.

4.2 The operation of the automated system

A chart of the function of the automated system is presented in figures 12 and 13. The figures show in a simplified manner all the tasks required to complete the construction planning of a retaining wall in a professional setting. Figure 12 depicts the portion of the process that is realised mainly with Grasshopper. The latter part of the process, presented in figure 13, pertains to the production information created with Tekla Structures, TeXworks and Excel. As stated before, however, the automated system does not currently complete all these tasks, as the focus is modelling and structural design. In order to depict the differences between the current and complete forms of the automated system, the tasks that are not included in the system are marked in bold italic printed in blue. This section provides a general description of the manner in which the system performs the different tasks of the planning process.

4.2.1 Reference data

Geological reference data is generated by a geological engineer, not the designer of the retaining wall. The data is collected into a template file created with Microsoft Excel. In this manner the data file for the geological data is separate from the main system created with Grasshopper. This is important, as the geological engineer may not have access to the Grasshopper software or the skill to use it. Excel however is a common office programme that is familiar to many people. The data entered into an Excel file can also be directly imported into the Grasshopper system without the involvement of the structural designer.

The other possible external reference data are reference models. Reference models, which can be 2D CAD-models or various types of 3D data need simply to be saved in the parent directory of the project. From there they are imported into the automated system by Grasshopper. The minimum number of reference models that the automated system requires to function are the wall alignment and the ground surfaces. The wall alignment is a simple 2D line, which the wall follows. The ground surfaces are 3D models detailing the shape of the ground both in front and behind the wall.

4.2.2 Initial data and design choices

The designer of a retaining wall is required to make many design decisions and determine much initial data. All such actions are confined within the Grasshopper software. The most

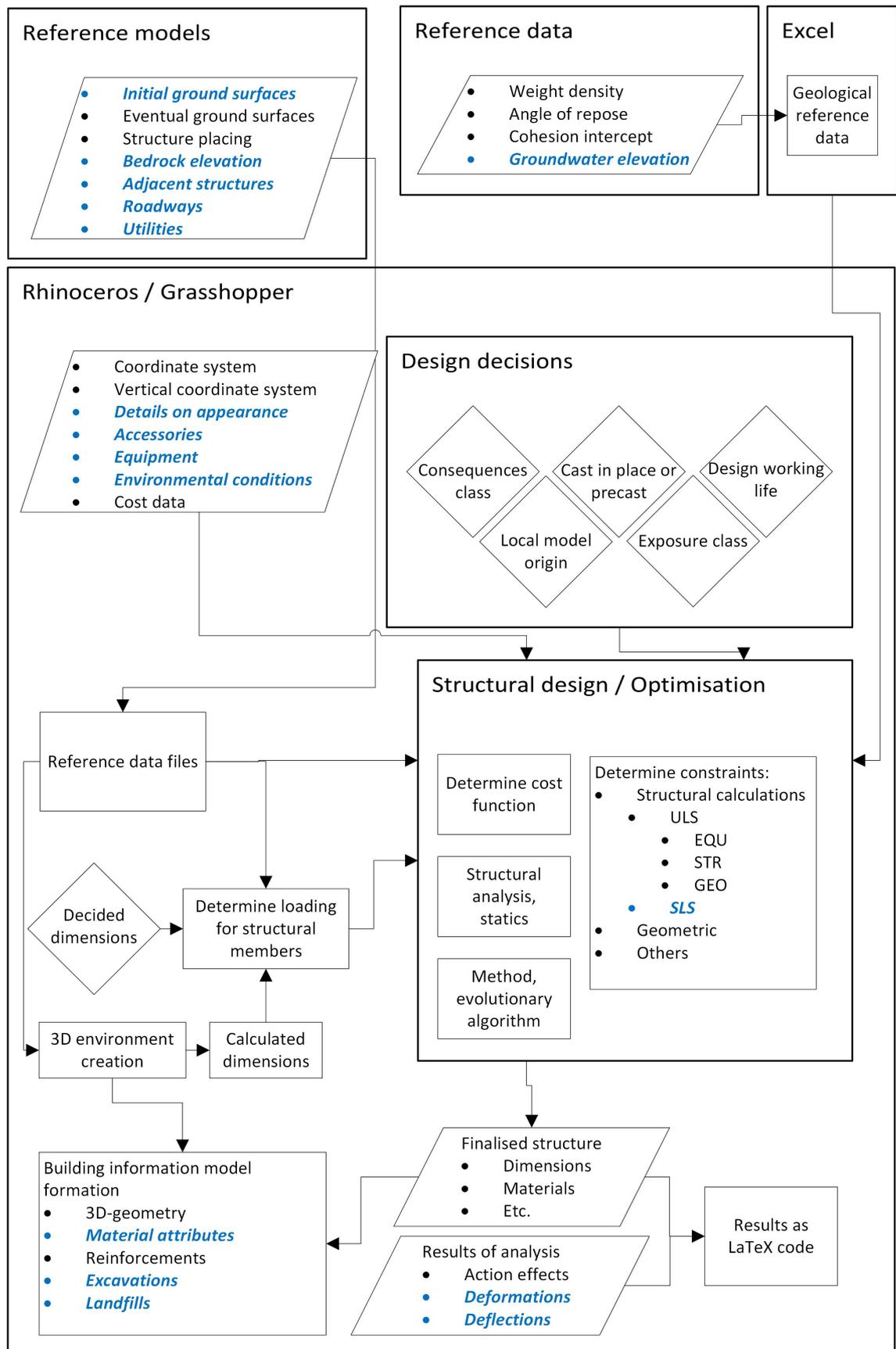


Figure 12. Process chart of the automated system, Grasshopper. Areas which are not included are marked in italicised blue print.

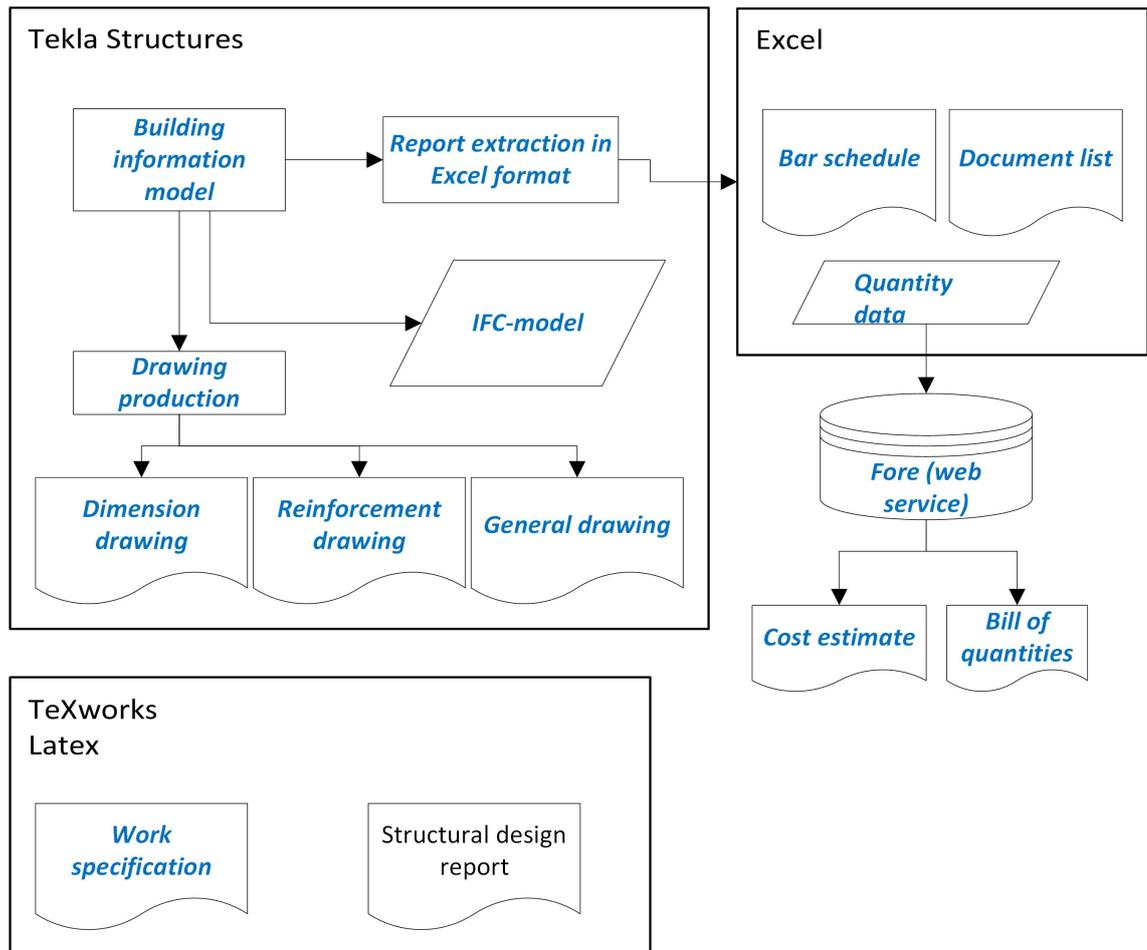


Figure 13. Process chart of the automated system, Production information. Areas which are not included are marked in italicised blue print.

relevant design decisions are given in figure 12. In addition to those, a designer is required to determine the loading, dimensions and reinforcements for the structure. Some of these choices can be automated through optimisation. That part of the process is discussed in more detail later.

All the dimensions that determine the shape of the wall are presented in figure 14. Dimensions beginning with lower case letters are for the designer to choose. Dimensions beginning with capital letters are auxiliary dimensions that the system calculates. The dimension H is calculated from the model data, the upper and lower ground surfaces, that have to be imported into the system. Figure 14 presents the most complex wall cross-section shape that the automated system can generate. More simple variations can be created by assigning some dimensions as 0. The simplest cross-section would be a wall consisting of two perfectly rectangular wall and footing slabs.

The reinforcement of the wall and the footing slab is relatively standardised and usually follows a quite similar form. This form is presented as an example reinforcement in figure 15. As figure 15 shows, the reinforcement consists of transverse bars in a cross-section plane that follow the surfaces of the wall and the footing slab. Longitudinal bars that

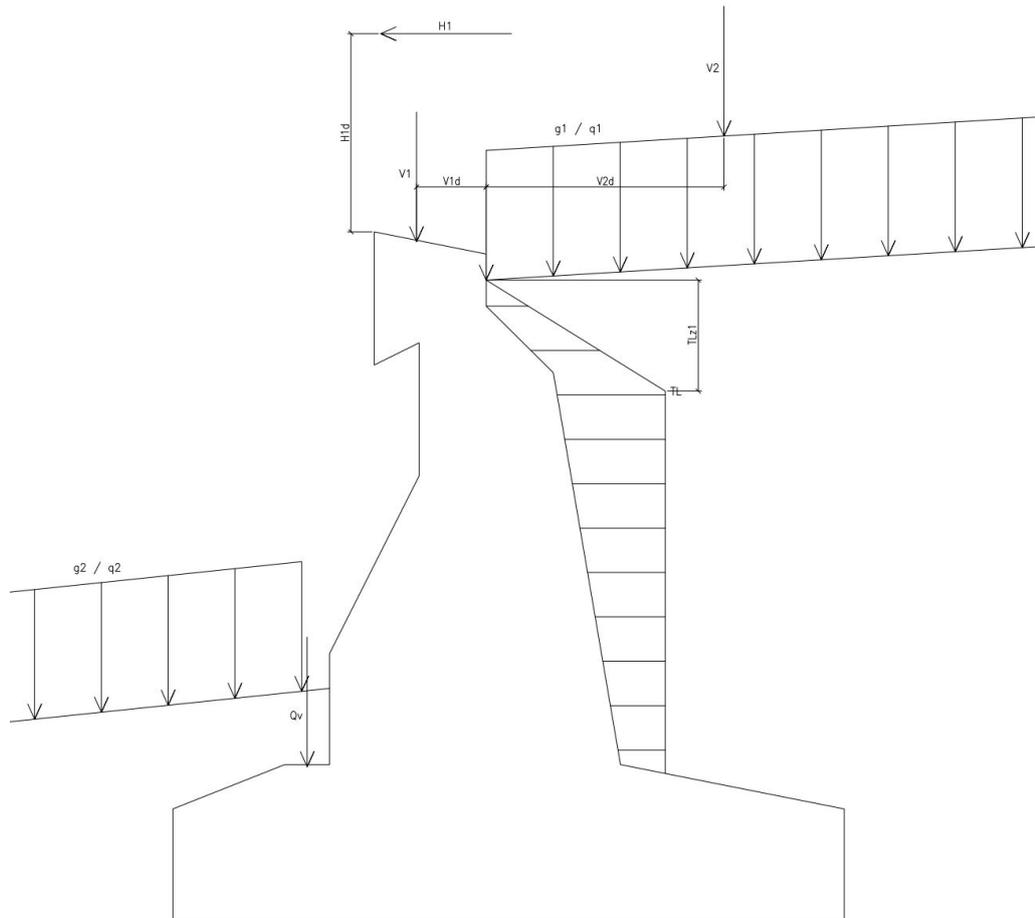


Figure 16. Loads that can be assigned to the retaining wall.

follow the length of the wall are also inserted. The structural calculations pertaining to the reinforcements focus on determining the diameters of the reinforcement bars and the number of bars per metre. The required anchoring lengths for the reinforcement bars must also be calculated, especially for the vertical bars of the wall, which join the wall and the footing slab.

Some of the loading for the retaining wall can be calculated, such as the earth pressure. Many loads must still be manually assigned to the structure. These loads are depicted in figure 16.

The height of the wall may vary along the length of the wall resulting in very different magnitudes for the action effects. The required amount of reinforcement varies therefore as well. If the reinforcement were uniform, it would have to be determined by the most severe action effects. This can lead to unnecessarily strong reinforcements in other parts of the wall, leading to higher expenses. This is addressed by dividing the wall lengthwise into sections, that allow for an alternating reinforcement. The number of sections is set to 5, regardless of the length of the wall. This number is deemed sufficient to allow the required variability, but still remain easily manageable at the construction site.

The reinforcement of edge beams does not usually derive from any structural calculations. A standard reinforcement is therefore used. The template for the standard reinforcement is in the NCCI 7 instructions [17].

4.2.3 Geometry generation

The 3D geometry of the retaining wall is determined by the reference models and the assigned dimensions. The generation of the geometry requires at minimum the final states of the lower and upper ground surfaces, as well as the alignment of the wall as a 2D line. This alignment line is broken into equally long sections. By default the cross-sections are generated at intervals of 1 metre, though the actual interval is adjustable by the user. A cross-section subsequently generated in all the end points of those sections. The generated cross-sections follow the shape of the ground surfaces. The 3D geometry of the structure can be created based on these cross-sections. The geometry is created in 3 separate parts. Those parts are the edge-beam, which is an optional portion atop the wall, the wall itself and the footing slab. This arrangement conforms to one possible method to create a BIM-model of a retaining wall

4.2.4 Structural design

The structural design of the retaining wall is also performed within Grasshopper, using the mathematical capabilities of the software. This can be done with existing mathematical components, such as addition, subtraction or division. Grasshopper includes also a component in which even complex equations can be written. An example of this component is illustrated in figure 17. The figure presents the statical calculations determining the shear force and moment caused by the earth pressure. Other actions are similarly computed with the calculation equations written to suit the situation. Figure 18 depicts the same principle being applied to calculate the load combinations. The component allows also for conditional clauses of the if-then structure. Thus a carefully formulated equation within the component can be applied to many structural variations.

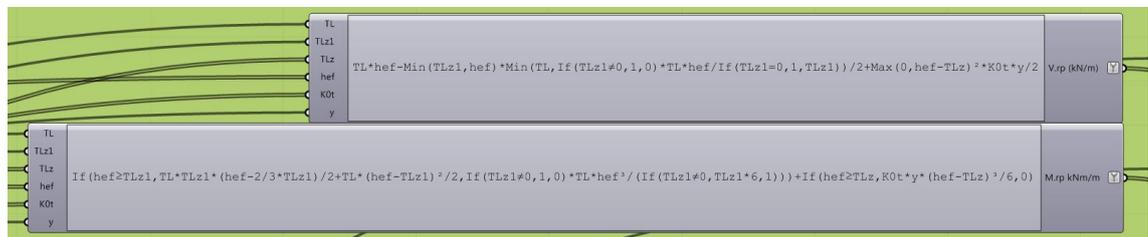


Figure 17. A calculations component in Grasshopper.

The calculation can consist also of a combination of many components. This is displayed in figure 19, in which the utilisation rate for the reinforcement of the footing slab is computed. The component is also capable to compute the values for all the separate cross-sections simultaneously. The calculations can therefore be performed for each of the cross-sections

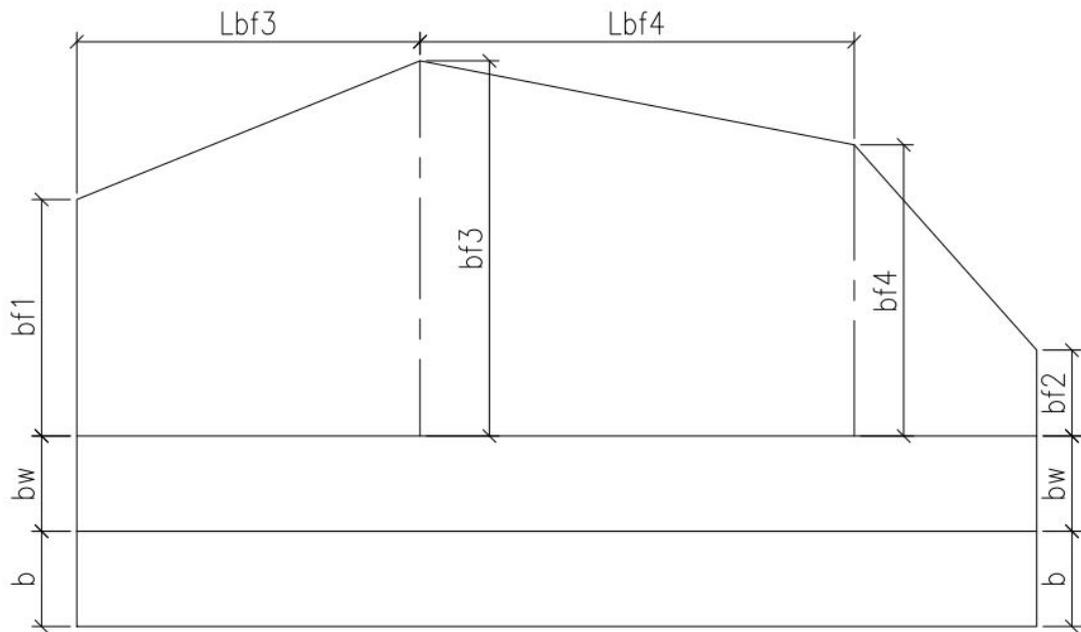


Figure 20. A plane view of the retaining wall.

presented in figure 20. The measurements $bf1$ and $bf2$ tell the length of the footing slab at the ends of the wall. The dimensions $bf3$ and $bf4$ are the lengths of the footing slab at two intermediate points along the length of the wall. $Lbf3$ and $Lbf4$ determine the distances of the intermediate points from the end of the wall and from each other, respectively. The optimised dimensions are presented in figure 14 highlighted with rectangles around the dimension mark. Figure 20 presents the optimised dimension that are not illustrated in figure 14. They are also listed below.

- bw , the thickness of the wall
- $bw2$, the width of the bevelling on the back of the wall
- h , the thickness of the footing slab
- $bf1$ and $bf2$, the lengths of the footing slab at the ends of the wall
- $bf3$ and $bf4$, the lengths of the footing slab at points intermediate points along the length of the wall.
- $Lbf3$ and $Lbf4$, the lengths determining the locations of the intermediate points mentioned above

The other dimensions, while also possessing some structural significance, pertain most of all to the appearance of the wall. Their values are therefore chosen by the designer in any event, and are subsequently not suitable or relevant for optimisation. All the variables pertaining to the steel reinforcement, except the reinforcement of the edge beam, is suitable

Table 2. Used cost information

Item	Unit cost
Concrete, edge beam	150 €/m ³
Concrete, wall	140 €/m ³
Concrete, footing slab	110 €/m ³
Reinforcement steel	1,50 €/kg

for optimisation. The optimisation system chooses therefore the reinforcements depicted in 15 for the 5 sections.

All of the optimisation variables in the automated system are discrete. The values for dimensions, for example, can only be chosen at specific intervals, with a 5mm interval being the default. The total area of reinforcement steel, which is the most relevant aspect in reinforcement design, is the product of the amount and diameter of the reinforcement bars. Both of those variables only get integer values.

The objective function of the optimisation problem is a cost function. The cost function consists of the sum of the construction materials and their respective costs. The mathematical form of the cost function is presented in Equation (10). For concrete the costs are given in relation to the volume of the concrete structure. The costs for the concrete can vary depending on the structural part for which it is intended. The concrete structure is therefore examined in 3 portions. The reinforcement steel is priced according to the total weight of the reinforcement bars. The unit costs used in the automated system are presented in table 2. These costs are estimates extracted from a web-based cost estimation service, Fore. The cost information is entered into the system manually, and must therefore also be updated manually at regular intervals.

$$e_{eb} \cdot V_{eb} + e_w \cdot V_w + e_f \cdot V_f + e_s \cdot m_s + p(\mathbf{UR}) \quad (10)$$

where

- e_{eb} is the concrete cost for the edge beam
- V_{eb} is the concrete volume for the edge beam
- e_w is the concrete cost for the wall
- V_w is the concrete volume for the wall
- e_{fb} is the concrete cost for the footing slab
- V_f is the concrete volume for the footing slab
- e_s is the reinforcement steel cost
- V_f is the reinforcement steel mass
- p is the penalty function
- \mathbf{UR} is the set of penalty function variables

The cost function is mostly rather simple multiplication and summation. The last portion of Equation (10), the penalty function, is included for the purpose of optimisation. The Galapagos optimisation tool has no direct methods to assign constraints. The design

constraints that derive from the building code, as well as other relevant constraints, are therefore included in the form of a penalty function. The purpose of the penalty function is to either drastically increase value of the objective function in minimisation problems, when the design is not within the acceptable range. In maximisation problems the penalty function decreases the value of the objective function. In this manner the penalty function guides the optimisation towards acceptable solutions.

The constraints of the structural optimisation problem derive mainly from the design criteria imposed by the eurocodes. The design criteria of a retaining wall are explained in chapter 2. For the penalty function they are presented in the form displayed in Equation (11).

$$\frac{X_d}{X_{Rd}} = ur \leq 1 \quad (11)$$

X_d is the design value of a action effect
 X_{Rd} is the structural design capacity
 ur is the utilisation rate

As can be seen from Equation (11), any utilisation rate less or equal to 1 is acceptable. For the penalty function the utilisation rates are transformed into the form presented in Equation (12). The modified utilisation rate UR as presented in Equation (12) will obtain the value 0 whenever the utilisation rate is within the acceptable range. When the utilisation rate is beyond the acceptable range, more than 1, the value increases in a linear fashion. The increase is very rapid, however, due to the multiplier 1000000. In this manner the minimisation problem will be strongly guided towards an acceptable solution.

$$UR = \max(0, ur - 1) * 1000000 \quad (12)$$

ur is the utilisation rate of an individual design situation
 UR is the modified utilisation rate

The complete penalty function is the sum of all the modified utilisation rates. The function is presented in Equation (13). With the modifications presented in Equation (12), the value of the combined penalty function will be 0 when all design constraints are fulfilled and will therefore not affect the cost function.

$$p(\mathbf{UR}) = \sum_{n=1}^i UR_i \quad (13)$$

p is the penalty function
 UR_i is the modified utilisation rate of design situation i

4.2.5 Production information

The BIM model of a retaining wall was not created during the case study. The 3D model of the concrete structure generated in Grasshopper can however be directly transferred into Tekla Structures with an existing component. This component generates the geometry and all the relevant material information as a BIM model in Tekla. Reinforcements can be transferred in a similar manner, using a specific component.

Report generation was studied only on a conceptual level in the case study. The adopted focus was to test the transfer of the various pieces of data imported in, or generated with, Grasshopper into the work specification and calculations report. TeXworks, which runs the Latex-code was utilised. The Latex file operates on commands, which can be created in Grasshopper and exported as a text file. A text file would contain Latex commands accompanied by their respective values. The commands inserted in the proper locations in the text would cause the imported values to appear in the printed text. Only a few commands were imported into a text file. No report template was created either, other than that test, that the text file transferred the commands as intended.

4.3 An overview of the system

The whole system created in Grasshopper is presented in figure 21. For clarity not all the lines connecting the Grasshopper components are visible in the picture. Some points of interest concerning the function of the system can be discussed however. Despite the extensive visual size of the system, the user need only be concerned with a limited number of inputs. Those inputs are grouped in the coloured areas that are grouped all next to each other at the top of the picture. The geometry is created after the inputs, and everything below the horizontal line is involved with the structural design of the retaining wall. The coloured areas represent groupings that ideally are ordered in such a way that they can be copied to other projects than retaining wall design. The Tekla and Latex connexions, which have not yet been properly created, are marked in the picture as texts.

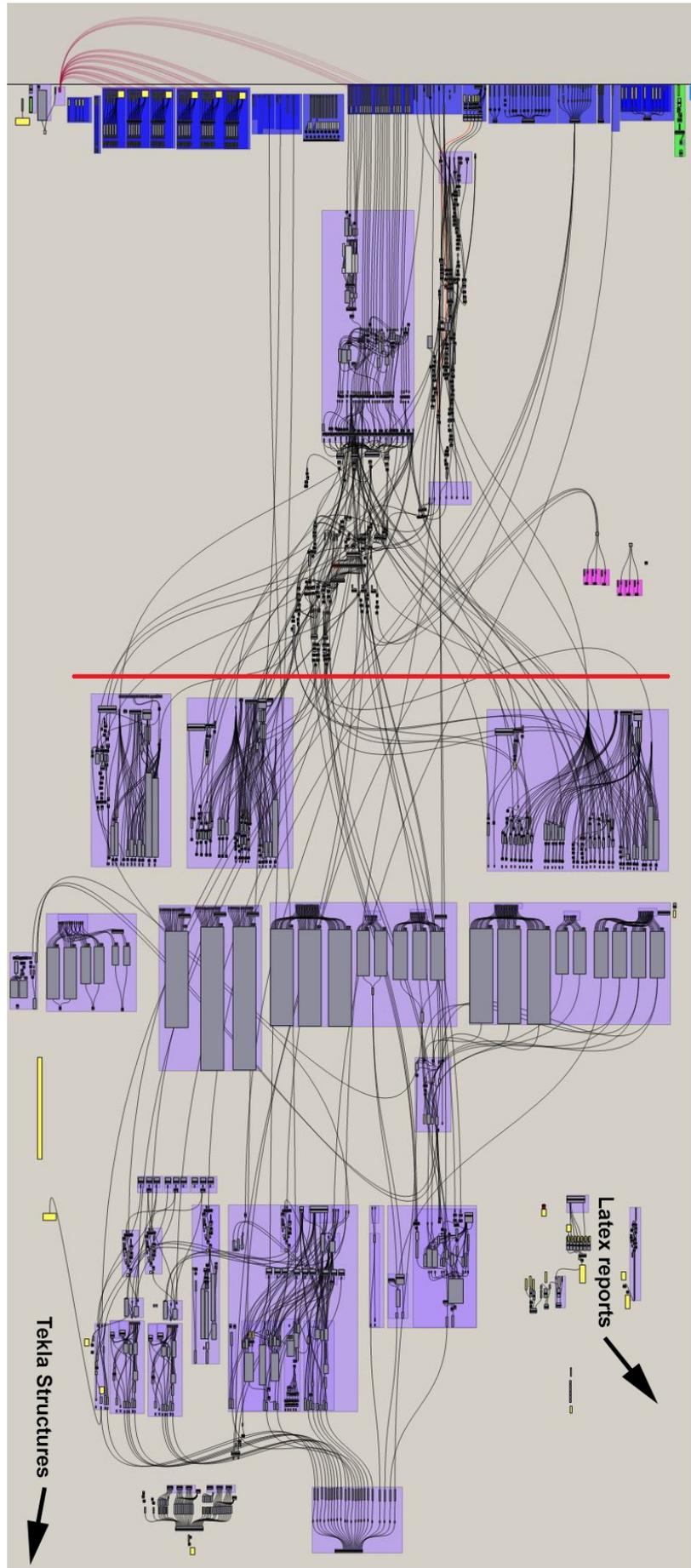


Figure 21. The Grasshopper system.

4.4 Testing the automated system

The system was tested by creating the construction plans for an example structure to the extent currently possible. A set of reference data purposely created for the testing of the automated model. This reference data was created to simulate the reference data of actual, real-life construction projects. It is therefore formatted according to customary reference data for such projects. The reference data consisted of two 3D mesh objects that represent the surfaces of the ground in their final states. They are portrayed in figure 22. The alignment of the wall in the proper coordinates is also required. The alignment is in the form of a 2D line. A set of geological reference data was also determined. The geological data does not correspond to any actual location, but is fabricated for this thesis, as are the reference models. The geological data is presented in table 4.

A set of initial data which consisted of the design decisions, loads and dimensions that are not part of the optimisation. These initially chosen values are presented in table 3. These values are constant throughout the design process.

The optimisation tool Galapagos was also tested. The test consisted of an initial design and a optimised design that could be compared. The initial design was created by manually choosing the values for the optimised dimensions and the reinforcements. This initial design was deliberately chosen to be in the acceptable range, fulfilling all design criteria. As the optimum result must also be within the acceptable range, a clearer comparison between the total cost of a merely adequate solution and an optimised solution can be obtained.

As the operation of the system has already been discussed previously, this section focuses on presenting the results. Tables 3 and 4 present the constant initial values. Table 5 presents the dimensions to be optimised. Both the initial values for the dimensions and the values obtained through optimisation are presented in the table. The ranges of values for the dimensions were also determined. The ranges are displayed in table 6. The ranges are mostly invented for the purposes of testing. The exception are the measurements for the footing slab. Due to the limited area that the reference models of the ground surfaces cover, too large values for the dimensions bf1-bf4 would place the structure beyond their area.

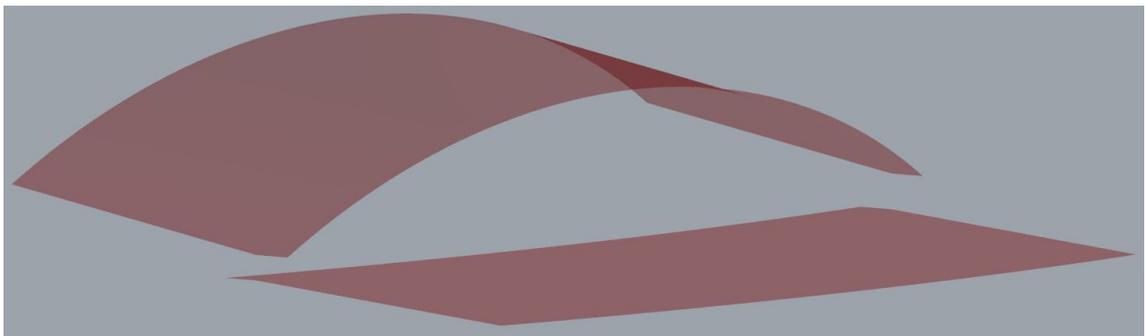


Figure 22. The ground surface meshes.

Table 3. Design values, Loads and dimensions

Consequence class	CC2
Exposure class	X0
Design working life	50 years
Concrete grade	C30/37
Construction class	2
Tolerance class	1
Reinforcement steel grade	B500B
Minimum bar distance	50mm
Compaction TL	16kN/m ²
Compaction depth TLz1	0,5m
Permanent area load g1	0kN/m ²
Imposed area load q1	20kN/m ²
Horizontal load H1	0kN
Vertical load V1	1kN/m
V1 type	Permanent
Vi distance from edge beam (outward)	1m
Vertical load V2	0kN/m ²
Permanent area load g2	0kN/m ²
Imposed area load q2	0kN/m ²
Vertical load Qv	0kN
D_{\min}	100mm
$h_{gb_{\min}}$	100mm
$h_{gf_{\min}}$	200mm
he	500mm
be	400mm
hev	0mm
hv	20mm
hb	500mm
hw1	0mm
hw3	0mm
bw1	0mm
hd	100mm
b	500mm
bp	0mm

Table 4. Geological reference data

Geological data	
Landfill in front of the Wall	
Effective weight in front of the wall	25 kN/m ³
The effective angle of repose in front of the wall	38 °
The effective cohesion in front of the wall	0
Landfill at the back of the Wall	
Effective weight at the back of the wall	25 kN/m ³
The effective angle of repose at the back of the wall	38 °
The effective cohesion at the back of the wall	0
Soil underneath the Wall	
Effective weight beneath of the wall	21 kN/m ³
The effective angle of repose beneath of the wall	35 °
The effective cohesion beneath of the wall	0

The system depends on measuring the earth volume behind and in front of the wall from these surfaces, and does not function if the structure reaches beyond them.

Aside from the dimensions, the system can optimise the reinforcements. The reinforcements can be chosen separately for the 5 zones along the length of the wall. Table 7 displays the reinforcements chosen for each of the surfaces of the structure in all the zones. The values n_1 to n_5 are the numbers of reinforcement bars per metre for all 5 zones. Equally the values A_1 to A_5 indicate the corresponding cross-sectional areas for the reinforcement. In table 7 both the initial and optimised values are depicted. As the reinforcement of the edge beam is standardised, and is not part of the optimisation, it is not included.

The results of the structural design are displayed in table 8. The table lists the utilisation rates of the various design situations both in the initial and the optimised situation. The example retaining wall is 15 metres long and consists of 16 separate and differently sized cross-sections. The system calculates the utilisation rates in all the design situations for all cross-sections. The values displayed in table 8 are the maximum values of all the 16 cross-sectional utilisation rates calculated. It can be seen from the table, that the optimised solution does obtain higher utilisation rates in some of the design situations, but not all. One utilisation rate is a full 100%. This utilisation rate does not derive directly from the eurocodes, however, as it is the value for the bar distance. The minimum acceptable is a user defined value.

The initial and optimised values for the cost function are presented in table 9. The expression of the cost function is presented in Equation (10). As all of utilisation rates in both the initial and optimised situation are under 100%, the value of the penalty function is

Table 5. *The optimised dimensions*

Dimensions	Initial values (mm)	Optimised values (mm)
bw	300	275
bw2	0	125
h	800	700
bf1	2000	2345
bf2	3000	1970
bf3	6500	0
Lbf3	5000	0
bf4	6000	0
Lbf4	4000	0

Table 6. *Dimensional ranges*

Dimensions	Lower limit (mm)	Upper limit (mm)
bw	250	300
bw2	0	1000
h	200	2000
bf1	1000	6500
bf2	1000	6500
bf3	0	6500
Lbf3	0	9000
bf4	0	6500
Lbf4	0	8000

Table 7. The reinforcements of the structure in 5 zones

d mm	n ₁ pcs	A ₁ mm ²	n ₂ pcs	A ₂ mm ²	n ₃ pcs	A ₃ mm ²	n ₄ pcs	A ₄ mm ²	n ₅ pcs	A ₅ mm ²
Wall reinforcement - bending, initial values										
25	12	5890.5	12	5890.5	12	5890.5	12	5890.5	12	5890.5
Wall reinforcement - bending, optimised values										
16	5	1005.3	9	1809.6	11	2211.7	11	2211.7	9	1809.6
Wall reinforcement - shear, initial values										
12	8	904.8	8	904.8	8	904.8	8	904.8	8	904.8
Wall reinforcement - shear, optimised values										
12	5	565.5	5	565.5	5	565.5	5	565.5	5	565.5
Wall reinforcement - horizontal, initial values										
20	8	2513.3	8	2513.3	8	2513.3	8	2513.3	8	2513.3
Wall reinforcement - horizontal, optimised values										
12	3	339.3	3	339.3	4	452.4	4	452.4	3	339.3
Footing slab reinforcement - lower surface, initial values										
32	19	15280.7	19	15280.7	19	15280.7	19	15280.7	19	15280.7
Footing slab reinforcement - lower surface, optimised values										
20	3	942.5	6	1885.0	3	942.5	3	942.5	4	1256.6
Footing slab reinforcement - upper surface, initial values										
32	19	15280.7	19	15280.7	19	15280.7	19	15280.7	19	15280.7
Footing slab reinforcement - upper surface, optimised values										
20	6	1885.0	5	1570.8	11	3455.8	6	1885.0	6	1885.0
Footing slab reinforcement - longitudinal bars, initial values										
16	18	3619.1	18	3619.1	18	3619.1	18	3619.1	18	3619.1
Footing slab reinforcement - longitudinal bars, optimised values										
10	6	471.2	6	471.2	9	706.9	6	471.2	6	471.2

in both cases 0. The difference between the cost function values is rather large, especially considering that both designs fulfil the design criteria of the eurocodes. The optimisation with the evolutionary algorithm solver of Galapagos had a duration of approximately 4,5 hours. Due to the duration of the optimisation, only one proper test was conducted. Further study on the optimisation may improve its efficiency.

Figures 23-26 provide a visual representation of the initial and optimised states of the structure. Figure 23 displays an overview of the structure that the system generated from

Table 8. Utilisation rates

Design situation	Initial values (%)	Optimised values (%)
GEO limit state		
The eccentricity of loading on the footing slab	45,44	71,56
The vertical load bearing capacity of the ground	17,95	44,63
The load bearing capacity of the ground for slide	94,91	99,95
EQU limit state		
SRT limit state		
Wall vertical reinforcement		
Bending	94,67	99,33
Required mechanical reinforcement ratio	80,9	21,62
Maximum mechanical reinforcement ratio	85,46	21,77
Shear	16,59	25,02
Bar distance	60,0	55,0
Minimum reinforcement	5,64	48,69
Minimum shear reinforcement	36,71	86,56
Wall horizontal reinforcement		
Footing lower surface reinforcement		
Bending	75,43	99,56
Required mechanical reinforcement ratio	51,42	3,93
Maximum mechanical reinforcement ratio	68,18	9,68
Bar distance	95,0	30,0
Footing upper surface reinforcement		
Bending	83,68	99,9
Required mechanical reinforcement ratio	57,05	9,56
Maximum mechanical reinforcement ratio	68,18	17,74
Bar distance	90,0	45,0
Footing longitudinal reinforcement		
Footing longitudinal reinforcement, bar distance	84,44	97,78
Footing longitudinal reinforcement, bar distance	67,5	100,0

Table 9. Initial and optimised values for the cost function

	Initial value (€)	Optimised value (€)
Cost function value	61890	11920

the inputs. The different colours in the image denote the 3 separate parts that the concrete structure is divided in. Figure 24 portrays an arbitrary cross-section of the wall. The figure includes both the initial design and the optimised design of the same cross-section. The largest difference in the design between the initial and optimised situations is in the footing slab. This difference is illustrated in figures 25 and 26. Figure 25 offers the plane view of the initial structure, while figure 26 depicts the optimised structure. The figures show clearly, that the optimised footing slab is much smaller than in the user defined initial design. This in part explains the great difference in the value of the cost function.

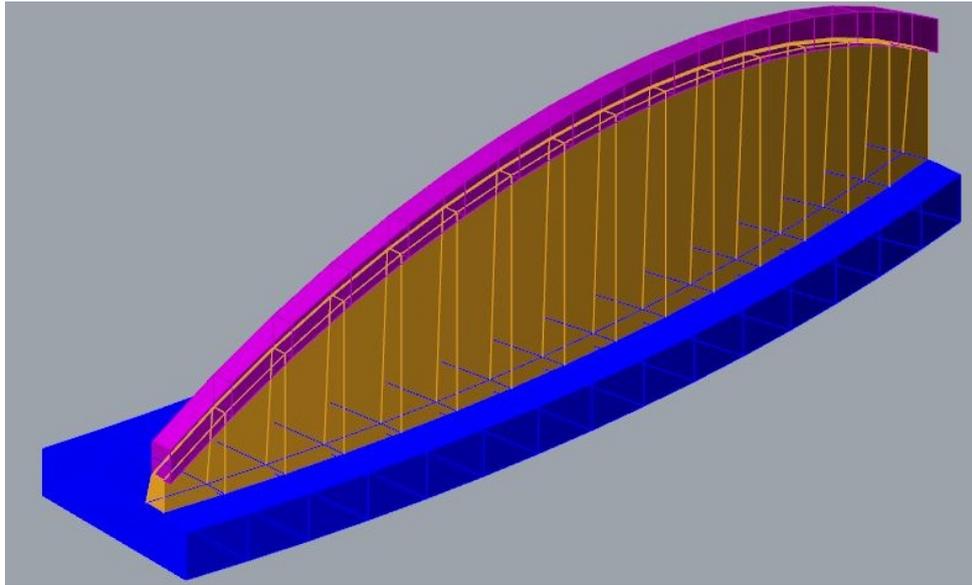


Figure 23. An overview of the generated retaining wall.

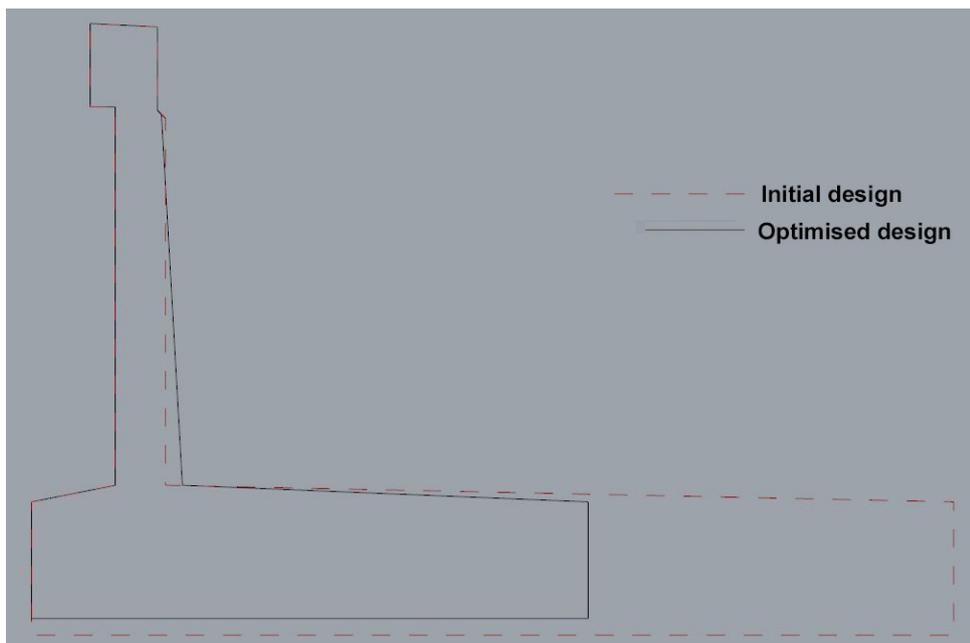


Figure 24. The initial and optimised cross-sections of the wall.

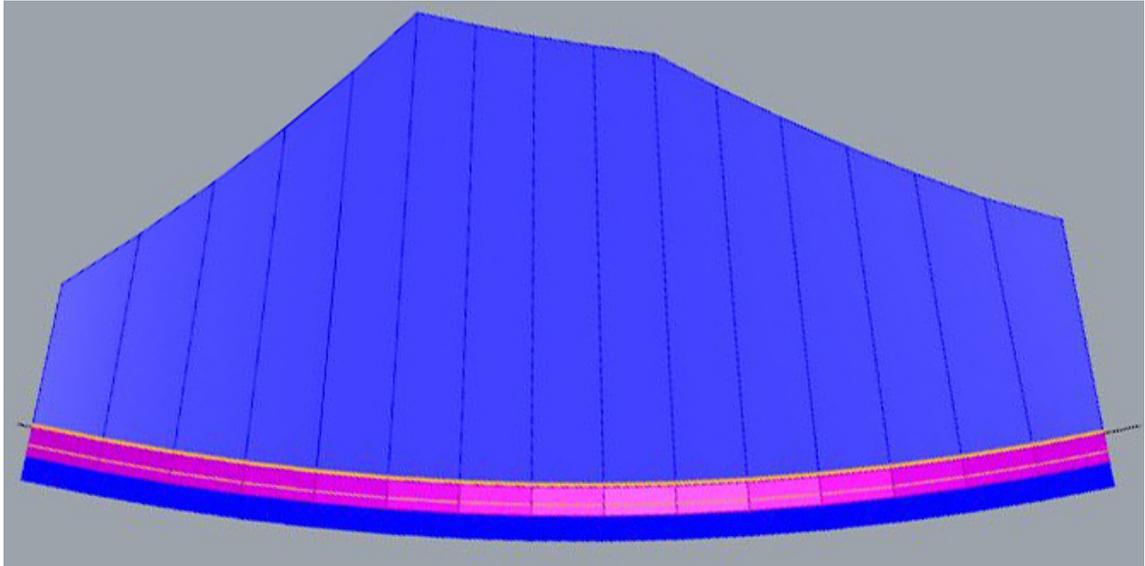


Figure 25. The plane view of the initial structure.

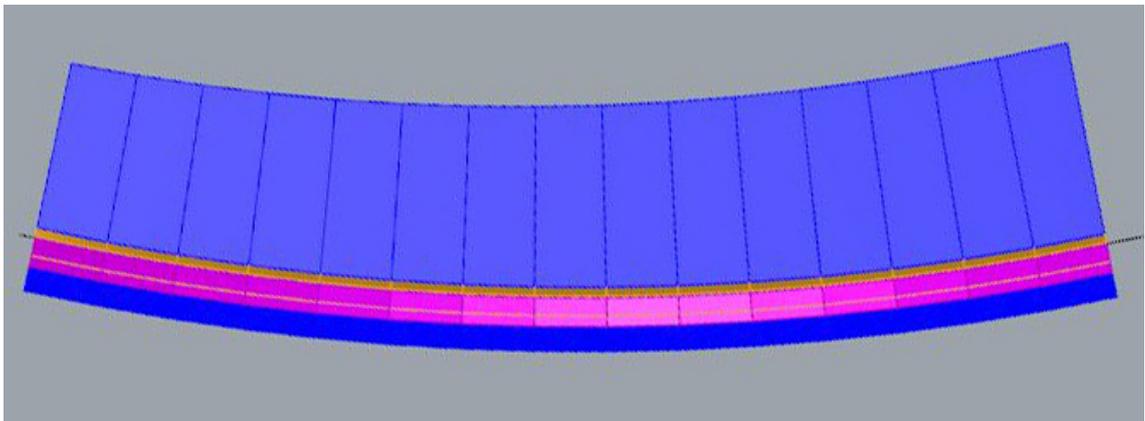


Figure 26. The plane view of the optimised structure.

4.5 Areas for further development

One of the most important areas is improving the connexion that the Grasshopper system has with Tekla Structures, the BIM programme. The 3D concrete structure is easily transferable to Tekla with the existing link between the two pieces of software. But the concrete is not the only significant geometry that needs to be exported to Tekla. The Grasshopper system calculates currently the necessary reinforcements for the structure in 5 zones. The system does not however create the 3D representations of these reinforcements. Only a crude geometric estimate is presently formed to determine the lengths of the bars for the optimisation process. Grasshopper possesses a recently added component to generate reinforcements to Tekla as well as the concrete geometry. To utilise the link, an algorithm would have to be developed in Grasshopper to generate the individual bars in the various situations that can arise.

The system does not include possible wall accessories either. The most common accessory

is possibly the railing that is fastened to the top of the wall. Others, such as lighting posts are also possible. Adding features to the Grasshopper system that generate at least the railing is necessary.

The Tekla-link in Grasshopper does export geometries and attribute data presently only to a Tekla model that is open simultaneously with Grasshopper. The model itself must be created manually, with information about coordinate systems and a local origin point, for example. Such information can be entered into Grasshopper, but it can not currently be transferred to Tekla like the model data. An area of interest may be to investigate possibilities to direct even the model production from Grasshopper. Tekla can also create reports, such as bills of quantities for the modelled structures, coordinate lists and others. Initiating report generation from Grasshopper would be convenient.

One major development area is the drawing production. Creating the model and importing data into Tekla from Grasshopper is only a part of the overall production information. The drawings have to be generated, which requires arranging the views, dimension-marks, additional text and labels onto the drawing sheet. Tekla has some capabilities to automatically generate drawings based on templates, but practical implementation must be researched further.

The production of written reports with Latex-code was briefly examined. Full implementation would require writing a report template which included a wide variety of options for variations. Producing a template would also serve as a study determining which vales need to be exported from Grasshopper. Many companies also have a specific appearance-template that the document must adhere to. A Latex document class to set the overall look of report would have to be created before the system could be utilised. AI and machine learning can be utilised to aid the programming work, but their practical use would require additional study.

5. CONCLUSIONS

Construction planning is an area which has experienced a near total computerisation. The level of automation is still rather low, necessitating much human work. This thesis examined methods to automate the human work involved in performing the individual tasks of the construction planning of a reinforced concrete cantilever retaining wall. The aim was to explore also approaches to create a automation structure to automate the whole planning process within one system. Automation is used to increase the effectiveness of construction planning, by reducing the time an engineer uses to perform routine tasks.

Two of the methods explored, the master model and KBE both focus on building a unified automated system. The systems consist of a central database that is interconnected with components that perform the design tasks. The master model as a procedure does not inherently result in the automation separate design tasks. The system is more of a coordinator of information and actions, leaving the automation of the actual work as a separate consideration. Yet even the ability to manage data in an orderly fashion is an improvement over the common present design practise in which the stages and the software of design are disjointed. The KBE approach differs in that it does not integrate only structural data, but also structural knowledge, both formal and instinctive. The master model can be produced by the KBE procedure. Some software has been developed to produce master models and KBE systems. They still depend on programming prowess and a comprehensive understanding of the system structure. The development of such systems for practical use will require to be thoroughly planned.

Automation is affected by the rapid development in technology. Computers continue to become more powerful and research into the automation methods advances. New software is written and existing software is improved. From an end users perspective a off-the-shelf programme that A BIM already functions somewhat alike to a master model, holding much structural data. The development of software may result in more programmes having capacities for structural design as well as modelling.

A change in the professional culture of the construction industry is also possible. In the age of building information modelling, all data concerning a structure can be integrated into the model. A model can also extend beyond the planning stage. A building information model can be used in the planning of construction work also, containing structural member specific scheduling data for instance. At the present, 2D drawings are still produced out of 3D building information models, as they remain a preferred medium for construction information. This may however change at some point in the future, precipitating the move to models only.

The case study focused on devising an automated system that created the 3D model of the retaining wall. The system performed also the structural design of the structure according to the rules of the eurocodes. The system is based heavily on the AAD software Grasshopper. The model created in Grasshopper generates the wall-geometry from at minimum the ground surfaces in front and behind the wall as well as a alignment line. Previous retaining wall projects were used as reference, to establish a reasonable rate of necessary variance. The automated system took an estimated 250 hours to construct.

The system performed well within the extent to which it was completed. Using reference data created specifically for the thesis, the model generation and structural design were successfully performed. The system functions appropriately both under completely manual control as well as with the optimisation tool, the use of which is optional. Based on the results, the utilisation of optimisation is however quite recommendable. With the added benefit of optimisation, potential exists to produce more cost effective designs. The system can also be further developed to encompass the structural design and modelling of other types of retaining walls, such as anchored or pile foundation walls.

REFERENCES

- [1] F. T. Agency, *Tien rakennussuunnitelma, sisältö ja esitystapa - Construction plans for roads, content and presentation*. Helsinki: Finnish Traffic Agency, 2013.
- [2] E. Alpaydin, *Introduction to Machine Learning*, 3rd ed. Cambridge: The MIT Press, 2016.
- [3] J. S. Arora, *Introduction to optimum design, second edition*. Elsevier Academic Press, 2004.
- [4] A. Asatiani and E. Penttinen, “Turning robotic process automation into commercial success – case opuscapita,” *Journal of Information Technology Teaching Cases*, vol. 6, no. 2, pp. 67–74, 2016. [Online]. Available: <https://search-proquest-com.libproxy.tut.fi/docview/1872245273?accountid=27303>
- [5] A. D. Belegundu and T. R. Chandrupatla, *Optimization Concepts and Applications in Engineering, second edition*. Cambridge University Press, 2011.
- [6] EN 1990, *Eurocode 0: Basis of structural design*. CEN, 2006.
- [7] EN 1991-1-1, *Eurocode 1: Actions on structures. Part 1-1: General actions. Densities, self-weight, imposed loads for buildings*. CEN, 2002.
- [8] EN 1992-1-1, *Eurocode 2: Design of concrete structures. Part 1-1: General rules and rules for buildings*. CEN, 2005.
- [9] EN 1997-1, *Eurocode 7. Geotechnical design. Part 1: General rules*. CEN, 2014.
- [10] S. Erkkilä, “Algoritmiavusteisen suunnittelun hyödyntäminen betonielementtirakenteiden suunnittelussa - utilization of algorithm-aided design in the design process of precast concrete elements,” Master’s thesis, 2017.
- [11] Finnish Association of Civil Engineers, *Geotekninen suunnittelu: eurokoodin EN 1997-1 suunnitteluohje - Geotechnical design: design instructions for eurocode EN 1997-1*. Helsinki: Finnish Association of Civil Engineers, 2009, vol. 207-2009; 207-2009.
- [12] A. H. Gandomi, A. R. Kashani, D. A. Roke, and M. Mousavi, “Optimization of retaining wall design using evolutionary algorithms,” *Structural and Multidisciplinary Optimization*, vol. 55, no. 3, pp. 809–825, Mar 2017. [Online]. Available: <https://doi.org/10.1007/s00158-016-1521-3>

- [13] A. A. Hopgood, *Intelligent Systems for Engineers and Scientists, Second Edition*, 2nd ed. Baton Rouge: CRC Press, 2001.
- [14] H. Humppi and T. Österlund, “Algorithm-aided bim,” in *Complexity & Simplicity - Proceedings of the 34th eCAADe Conference - Volume 2, University of Oulu, Oulu, Finland, 22-26 August 2016*, 2016. [Online]. Available: http://papers.cumincad.org/cgi-bin/works/paper/ecaade2016_158
- [15] M. C. Lacity and L. P. Willcocks, “Robotic process automation at telefónica o2,” *MIS Quarterly Executive*, vol. 15, no. 1, pp. 21–35, 2016. [Online]. Available: <http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=114020351&site=ehost-live&scope=site>
- [16] Liikennevirasto-Finnish traffic agency, *Eurokoodin soveltamisohje - Betonirakenteiden suunnittelu – NCCI 2, Eurocode application instructions - Concrete design - NCCI 2*. Liikennevirasto-Finnish traffic agency, 2017.
- [17] ———, *Eurokoodin soveltamisohje - Geotekninen suunnittelu - NCCI 7, Eurocode application instructions - Geotechnical design - NCCI 7*. Liikennevirasto-Finnish traffic agency, 2017.
- [18] ———, *Eurokoodin soveltamisohje - Siltojen kuormat ja suunnitteluperusteet - NCCI 1, Eurocode application instructions - Bridge loads and design principles - NCCI 1*. Liikennevirasto-Finnish traffic agency, 2017.
- [19] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine learning: an artificial intelligence approach*. Los Altos, CA: Kaufmann, 1983.
- [20] R. N. Palmer and B. W. Mar, “Automation of civil engineers:some observations,” *Journal of Management in Engineering*, vol. 5, no. 3, pp. 286–300, 1989.
- [21] Y. Pei and Y. Xia, “Design of reinforced cantilever retaining walls using heuristic optimization algorithms,” *Procedia Earth and Planetary Science*, vol. 5, pp. 32 – 36, 2012, 2012 International Conference on Structural Computation and Geotechnical Mechanics (SCGM 2012). [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1878522012000070>
- [22] C. Preidel, S. Daum, and A. Borrmann, “Data retrieval from building information models based on visual programming,” *Visualization in Engineering*, vol. 5, no. 1, p. 18, Oct 2017. [Online]. Available: <https://doi.org/10.1186/s40327-017-0055-0>
- [23] S. Ramanna, L. C. Jain, and R. J. Howlett, *Emerging Paradigms in Machine Learning: An Introduction*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–8. [Online]. Available: https://doi.org/10.1007/978-3-642-28699-5_1

- [24] G. L. Rocca, “Knowledge based engineering: Between ai and cad. review of a language based technology to support engineering design,” *Advanced Engineering Informatics*, vol. 26, no. 2, pp. 159 – 179, 2012, knowledge based engineering to support complex product design. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1474034612000092>
- [25] M. Sandberg, R. Gerth, W. Lu, G. Jansson, J. Mikkavaara, and T. Olofsson, “Design automation in construction : An overview,” in *Proceedings of the 33rd CIB W78 Conference 2016, Oct. 31st – Nov. 2nd 2016, Brisbane, Australia* :, 2016.
- [26] M. Sandberg, H. Johnsson, and T. Larsson, “Knowledge-based engineering in construction - the prefabricated timber housing case,” *Journal of Information Technology in Construction (ITcon)*, vol. 13, p. 408, 2008.
- [27] E. Särkkä, “Paalulaatan suunnittelu parametrisen tietomallin avulla - design process of pile supported embankment slab with parametric bim,” 2015.
- [28] T. Tanska, T. Österlund, *Algoritmit puurakenteissa - Algorithms in timber structures*. Oulu: DigiWoodLab, 2014.
- [29] Tiehallinto, *Siltojen suunnitelmat - Construction plans for bridges*. Helsinki: Edita Oy, 2000.
- [30] Tom M. Mitchell, *Machine learning*. McGraw-Hill, 1997.
- [31] Tony Gaddis, *Starting out with Python*. Pearson, 2017.
- [32] T. R. Tyson, “Effective automation for structural design,” *Journal of Computing in Civil Engineering*, vol. 5, no. 2, pp. 132–140, 1991. [Online]. Available: [https://doi.org/10.1061/\(ASCE\)0887-3801\(1991\)5:2\(132\)](https://doi.org/10.1061/(ASCE)0887-3801(1991)5:2(132))
- [33] W. M. P. van der Aalst, M. Bichler, and A. Heinzl, “Robotic process automation,” *Business & Information Systems Engineering*, vol. 60, no. 4, pp. 269–272, Aug 2018. [Online]. Available: <https://doi.org/10.1007/s12599-018-0542-4>
- [34] W. J. Verhagen, P. Bermell-Garcia, R. E. van Dijk, and R. Curran, “A critical review of knowledge-based engineering: An identification of research challenges,” *Advanced Engineering Informatics*, vol. 26, no. 1, pp. 5 – 15, 2012, network and Supply Chain System Integration for Mass Customization and Sustainable Behavior. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S147403461100036X>
- [35] Z. Xie, “Neural text generation: A practical guide,” *CoRR*, vol. abs/1711.09534, 2017. [Online]. Available: <http://arxiv.org/abs/1711.09534>