Performance Driven Design and Design Information Exchange

Establishing a computational design methodology for parametric and performance-driven design of structures via topology optimization for rough structurally informed design models

Sina Mostafavi1, Mauricio Morales Beltran2, Nimish Biloria3
1,3TU Delft, Hyperbody, AE+T, Faculty of Architecture, The Netherlands, 2TU Delft, Structural Design, AE+T, Faculty of Architecture, The Netherlands
1s.mostafavi@tudelft.nl, 2m.g.moralesbeltran@tudelft.nl, 3n.m.biloria@tudelft.nl

Abstract. This paper presents a performance driven computational design methodology through introducing a case on parametric structural design. The paper describes the process of design technology development and frames a design methodology through which engineering, -in this case structural- aspects of architectural design could become more understandable, traceable and implementable by designers for dynamic and valid performance measurements and estimations. The research further embeds and customizes the process of topology optimization for specific design problems, in this case applied to the design of truss structures, for testing how the discretized results of Finite Elements Analysis in topology optimization can become the inputs for designing optimal trussed beams or cantilevers alternatives. The procedures of design information exchange between generative, simulative and evaluative modules for approaching the abovementioned engineering and design deliverables are developed and discussed in this paper.

Keywords. Performance driven design; design information; design technology; topology optimization; parametric design.

INTRODUCTION

One of the challenges in performance driven design methodologies is the way that designers can effectively integrate simulation and optimization techniques with parametric design and generative procedures (Oxman, 2008). This challenge can also be attributed to as the lack of tools to support effective knowledge integration in Computer Aided Design (CAD) techniques and methods (Cavieres et al., 2011). In design practice, theoretically this gap is bridged via simultaneous consultations with engineers and specialists. However, for many design problems this concurrency might not be achievable
and applicable. In this paper, as one of the directions towards achieving this concurrency we specifically focus on the implementation of optimization techniques in structural design, to see how they can be integrated with parametric design techniques. To be more explicit from a computational design point of view, and to the design methodology itself, the focus of the article is on design information modeling, exchange and interoperability. The paper structure here onwards addresses questions and objectives, the process, the tool and the methodology. Subsequently, the results from the examples and a case study are briefly reported and eventually the discussion focuses on performative design methodology, its supporting design technology, rough Building Information Modeling (BIM) systems (Eastman et al., 2011) and future directions.

**Question and Objectives**

Two major questions are the subjects of exploration in this research. The first one, which is more from a computational design perspective, questions the possibility to appropriately integrate optimization algorithms and procedures, in our case, topology optimization, within a parametric design system. pertaining to this question, the objective is to design a system with connected sub-procedures and feedbacks with appropriate methods for design information exchange and translation from different CAD and programming platforms (operating as design decision support) for performance driven design (in this case is a truss structural system). The concurrency and consistency in extracting, generating and structuring of geometric design information such as size, resolution, etc., and non-geometric design information such as load conditions, Degree of Freedom (DOF), etc., will be discussed.

The second question is how to make the process of topology optimization more suitable for the scale of architectural design and what are the benefits of doing so? This method and in general Finite Element Analysis (FEA) have been widely used at the scale of industrial design for uni-body or monocoques structures like bike frames (Figures 1a and 1b). However, for the scale of a building, while we usually have building elements like bars, beams, columns and joints, directly using the discretized result of topology optimization might not be much applicable and might impose choosing in-site concrete casting (Figures 1c and 1d). While in this research, we build on the assumption that it would be more relevant for designing a truss or frame structures, if we translate the finite geometry to a proper geometric system with nodes and bars (Figure 1e).

While to a certain extent the process is defined and developed as a generic design technology, the type of structural system is intentionally and precisely defined as a trussed beam or cantilever. Besides the developed algorithms, from a technical point of view, testing and developing of various methods for information exchange between software and platforms like Matlab, Rhinoceros, Grasshopper plus some of its add-on plug-ins and the needed structural analysis software shall also be elaborated.

**DESIGN PROCESS AND METHODOLOGY**

The process, as illustrated in the flowchart (Figure 2), is a set of sub-procedures A to D, such that the output, input and procedures are systematically correlated. In each sub-procedure there are four kinds of modules, which are decisions or inputs (), processes (), outputs () and visualizations (). To make this technology-based design process an interactive, cyclic and performative one, the following aspects have been taken into consideration:
The decision(s) or input(s) of each sub-procedure are used as common inputs for more than one of the sub-procedures, whenever and wherever needed.

The final translated output in each of the sub-procedures would automatically or semi-automatically be processed as the input of the next sub-procedure(s).

After each single measurement or evaluation module there is either a visualization for alerting or a feedback loop to the previous stages.
The detailed descriptions of each of the sub-procedures are as follows:

**Definition, design domain, discretization and load condition [A]**

In this phase, the designer defines the geometrical properties on which the supports and loads can be parametrically added and modified. These properties are, so far, the span and the height of a cantilever or a beam with either upper or lower distributed or point loads on sides. However, the process in this stage and other stages is designed in a way that more irregular initial shapes are also possible to implement, by just removing some portions of the initial planar design domain. The main inputs in this sub-procedure are the dimensions, the magnitude and coordination of loads, supports and the mesh resolution (Figure 3a). Since this mesh resolution is indeed the discretization of the design domain for the following FEA, the acceptable resolution is a variable depending on the available computation time, power and the desired refinement. The output is a two-dimensional matrix or data list in .txt format that contains the relative dimensions of the geometry based on the discretization resolution, magnitude, the relative coordinates and calculated DOFs of each load positions based on the defined resolution. This step is done through visual programming using Grasshopper in Rhino3D. The generated geometry attributes and alert messages (if either the geometry or resolution is not within some predefined range) are simultaneously visualized (Figure 3b and 3c).

**Material distribution (MD); topology optimization [B]**

In this stage, the goal is to find the optimal material distribution of the discretized generated design. This step is in Matlab and is based on the implementation and development of a topology optimization code, originally written by Sigmund (2001) with the purpose of solving linear compliance minimization using an optimizer and finite element subroutine. Modifications in the code are set up, with the objective of making it compatible with the input data files and supports interoperability of the output for the next sub-procedures.

The geometrical properties, DOFs and loads will be automatically called in the code and what has to be defined by the designer is the percentage of total remaining material. Consequently, two parallel results are the outputs of this phase, one a set of images that in real-time illustrates the results of material distribution simulation and the other, a set of
excel spreadsheets, in which numerical values ranging from zero to one are stored. In the tested cases, four spreadsheets, respectively, with 30, 40, 50 and 60 percentage of remaining material have been the final outputs. In order to make this process more semi-automatic, further modifications can also be done in the code to pre-define the range for remaining material in previous sub-procedures (Figure 4).

**Typology, defining the type of structure [C]**

The goal in this sub-procedure is to translate discrete or pixelated geometry, which is the result information from the topology optimization to a vector-based geometric system with nodes and lines (Figures 5a and 5b). Although in the initial visualized topology the lines are detectable with the eyes of the designer, they are not automatically distinguishable for the CAD platform. So one of the main crucial challenges here was to extract the nodes and define the bars by using and developing appropriate algorithms in a way that the topologies do not change. This implies that if in a resultant image we see nine white polygons in the resultant vector geometry we should have also the same condition. Finally, the output is a matrix as .txt file with the required information of nodes, bars and load conditions in the desired format (Figure 5c).

Figure 6 illustrates the applied and developed
methods for extracting the nodes from the resultant discrete geometry. After reading the float values on the spreadsheets and re-visualizing the results through using visual programming in Grasshopper, and tagging each cell with its corresponding zero to one value, a filter separates the cells into two lists of data. The reason for having this buffer is to let the designer find the appropriate continuous topology similar to the image result but this time composed of surfaces with the size of defined resolution. For instance, in the Figure 6 this filter value is 0.3, which means that all values less than this would be within a list to create the negative shape and those cells with values equal or more than this threshold will create the positive shape (Figures 6a-c).

In the next step, after joining the negative shapes and retrieving the outer boundary curves, the goal is to transform the jagged edges of these shapes into straight lines extract polygons. This is done through minimizing the difference between the areas of shapes with straight lines from the original one with jagged edges. (Figure 6d and 6e). This part is mainly done through visual and script based programming in Grasshopper, and Galapagos (evolutionary solver) for finding the shapes with optimum areas. By having the straight lines of the positive shapes (Figure 6f), it would also be possible to develop and apply a skeletonization technique based on Voronoi algorithms (Aurenhammer and Klein, 2000) to get axial curves with similar original topology (Figure 6g). Then by means of a Boolean gate the generated points through skeletonization algorithm can be achievable in a separate point cloud list (Figure 6h). After connecting the points to their neighboring, the nodes are those which have three or more connections. Therefore, another algorithm is developed to automatically detect nodes based on the numbers of connected neighbors (Figure 6i and 6j). Subsequent to this step another optional procedure is also developed in which the detected nodes would be anchor points of physical spring systems and other points will be stretched while having the fixed nodes as their supports. Therefore, with this method the poly-lines, which are not geometrically straight lines, will be stretched to form the bars.

Using this sub-procedure for all cases would allow us to have a persistent method to retrieve four
set of nodes and bars for each of the volume fractions for any parametrically defined design domain with distinct load and support conditions in the first sub-procedure. After having the nodes and bars the structural determinacy of the each vector-based topologies will also be measured in advance through putting the numbers of the bars nodes and supports conditions in static equilibrium.

**Analysis, structural behavior and search for optimal solution [D]**

This stage starts with reading the input file in Matlab with the information on nodes, bars and load conditions from the previous step coming from the Rhino/Grasshopper. By having this information set for each of the four topologies, a static structural analysis will be run for obtaining local stresses and global displacement of the truss with the initial load and support conditions. Other variables such as material properties and available profiles can also be parametrically defined or extracted form a data set in this stage. Figure 7 presents an overview of this sub-procedure for a beam case. Here, the generated data list store the results that will be used for further visualization and profile assignation in 3D design environment. Further information for evaluation and comparisons for different input parameters and topologies like total volume, maximum and minimum length of the elements can be extracted from the optimum result depending on the design requirements.

The fitness criteria in the search process are allowable stress of the bars and global displacement. The search process finds the minimum required cross sectional area from the defined input sets for each of the bar elements and simultaneously checking the allowable global displacement. This part of the process is mainly done implementing a code in Matlab for cross sectional optimization. Moreover, in order to check the reliability of the process, some results have been compared with the results in the GSA suite. Figure 8 represents an overview for a cantilever that has started from the discrete geometry to vector geometry with nodes and lines in which the cross sectional optimization results are directly used as input data for tubular profile assignment, results in differentiation in the size of the each profile.
TESTS AND CASES

In addition to separate examinations inside each of the sub-procedures to improve and test the functionality and generalizability of the applied methods are conducted, two A-to-Z cases have been tested which will be briefly reported and shortly discussed here. First one is a cantilever case with one point load at its end (Figure 9). As it is illustrated here the results of optimization based on the initial design domain and load conditions are translated to a set of optimized truss structure. In this case and for any of its variation, besides the topological difference between the final topologies, the corresponding information sets pertaining to the structural performance and geometric properties of elements are also available for further evaluation and comparison.

The second case is a beam but in this case within a real world background design scenario for further validation of developed methods. This exercise builds upon a featured connecting bridges based project by Steven Hall Architects (Figure 10).

One of the benefits in this case is that there are similar design problems but with different sizes and proportions. This means that parametrically defining the initial design domain while having concurrent performance measurements would add to the efficiency of the design process itself. Additionally, as it is illustrated in Figure 10, for each design domain with different load conditions we have four optimized topologies in vector format with nodes and bars that can be translated to steel, wood or any other profiles. Moreover, based on cross sectional optimization we will have a differentiation in profile properties which might be a source of new performance driven design idea for designers. In other words, in addition to automatic evaluations and comparisons based on the generated and stored quantitative information, the developed design system might also suggest some implicit hints based on the visualized information and rough performance estimation. For instance in this case the architects might decide just to have one support for the roof of the bridge at a specific coordinate and have lateral beams to support the walking deck at
every six meters. With these presumptions, based on what the designer perceived from the way the algorithms lead to optimum solutions, he or she could alter the input parameters, go back to the very beginning stages, and find the optimum result with required conditions and acceptable proportions simultaneously (Figure 11).

**DISCUSSION**

In terms of design methodology, this research address the integration of performance measurement and evaluation modules in a parametric design system, opening the black box of topology optimization, making it more traceable, specific and applicable by designers, particularly at the scale of architectural design. Proposed algorithmic-design-information-exchange scenario between steps of the design procedure parallel to CAD and programming platforms have been considered and tested as an appropriate approach for this goal. Behind the benefits that can be implicitly and explicitly enumerated for this specific case, a conclusion is that knowledge integration in parametric design needs customized scenarios for integration and structur-
ing of geometric and non-geometric information. The extent till which extraction and visualization of this information is needed, is dependent on one hand on the knowledge of the user as designer, and on the other hand on the design requirements and goals for a specific design problem.

From a design technology point of view, in addition to developed algorithms for solving the issues on interoperability or data exchange, what was peculiarly challenging in this research was developing and customizing a method for translating finite or discrete geometry to vector based and continuous topology. It is possible to deduce that the translation procedure can be considered as a more generic issue in parametric and performative driven design strategies. In addition to the process of FEA-based topology optimization methods in the realm of structural design, FEA methods are omnipresent in the basis of many simulation techniques. Considering this fact, it might be beneficial to facilitate performance driven design methodologies with methods, tools and strategies for such translational procedures. The implementation of skeletonization and node finding methods can be considered as some of these cases for such translational algorithms. Future cases need to be defined with similar methodological schemes to test their generalizability and functionality.

ACKNOWLEDGEMENTS
The authors highly appreciate the permission given by professor Ole Sigmund on the use of the 99 lines topology optimization code for the purposes described in this paper.

REFERENCES