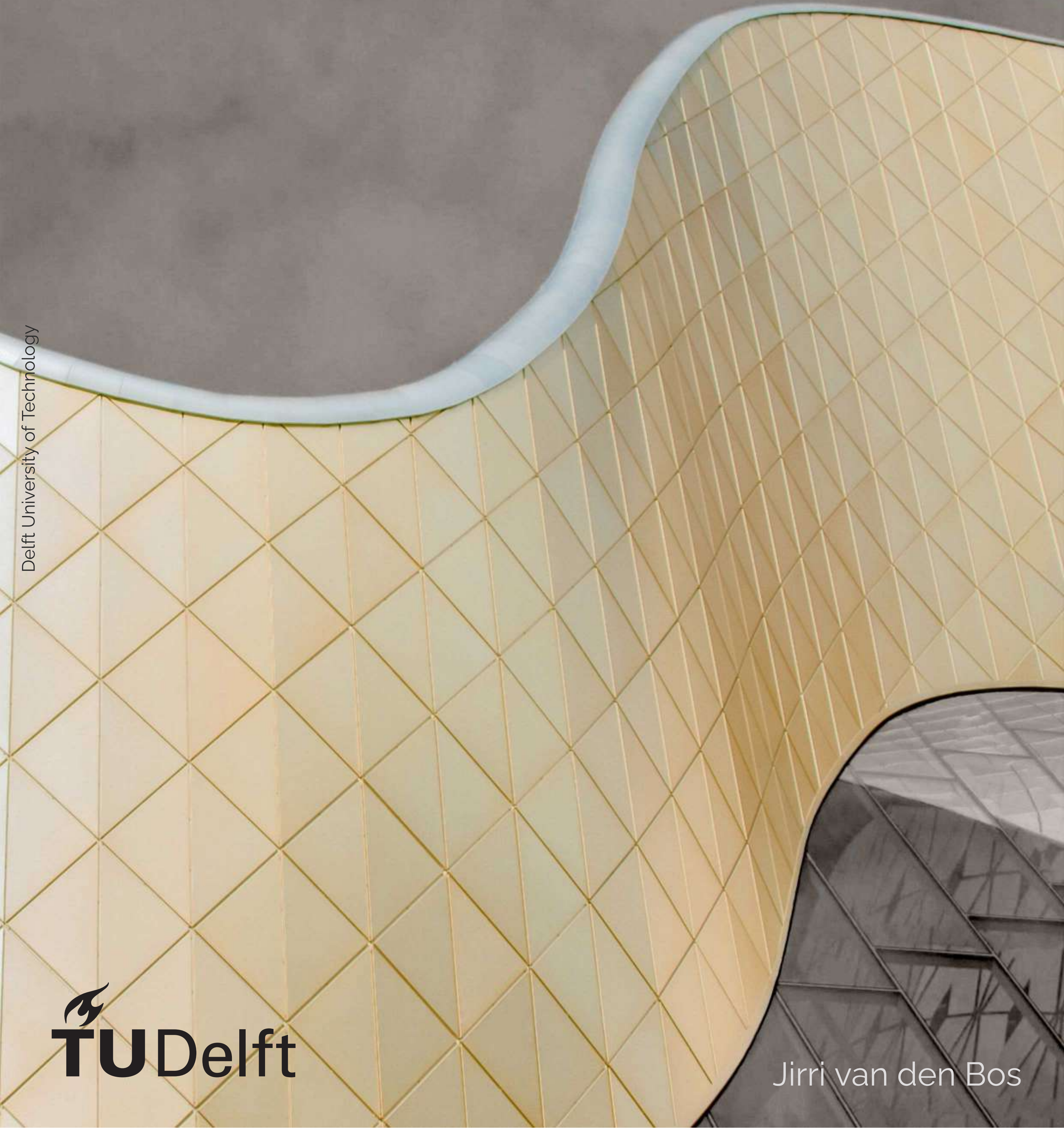
(re)assembly

towards a future of automatic reuse and reconfiguration



Delft University of Technology

TU Delft

Jirri van den Bos

(re)assembly

towards a future of automatic reuse and reconfiguration

by

Jirri van den Bos

to obtain the degree of Master of Science
at the Delft University of Technology,

Student number:	4801792	
Project duration:	September 1, 2022 – January 17, 2024	
Thesis committee:	Dr. ir. S. Asut,	TU Delft, 1st supervisor
	Dr. -Ing. M. Billow,	TU Delft, 2nd supervisor
	Ir. M.H. Meijs,	TU Delft, Delegate examiner

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

Abstract

Improved productivity and a higher level of sustainability are modern day challenges faced by the building industry that are currently being and could further be solved by a higher degree of automation. Freeform architecture is still in its infancy but the demand for it is growing. Reusable building elements are a rarity in freeforms. This thesis aims for the implementation of a new node and beam system for increased reusability and for more automation in the construction of building facades.

This thesis introduces a novel reusable node and beam system for use in the automatic assembly of freeform architecture. By optimising input shapes, applying computational placement of the elements and generating instructions for robotic systems, the building sector can not only improve its productivity and reduce its emissions, it can furthermore revolutionise the stylistic nature of architecture and facilitate the fluid adaptation of new forms and functions.

Contents

1 Introduction

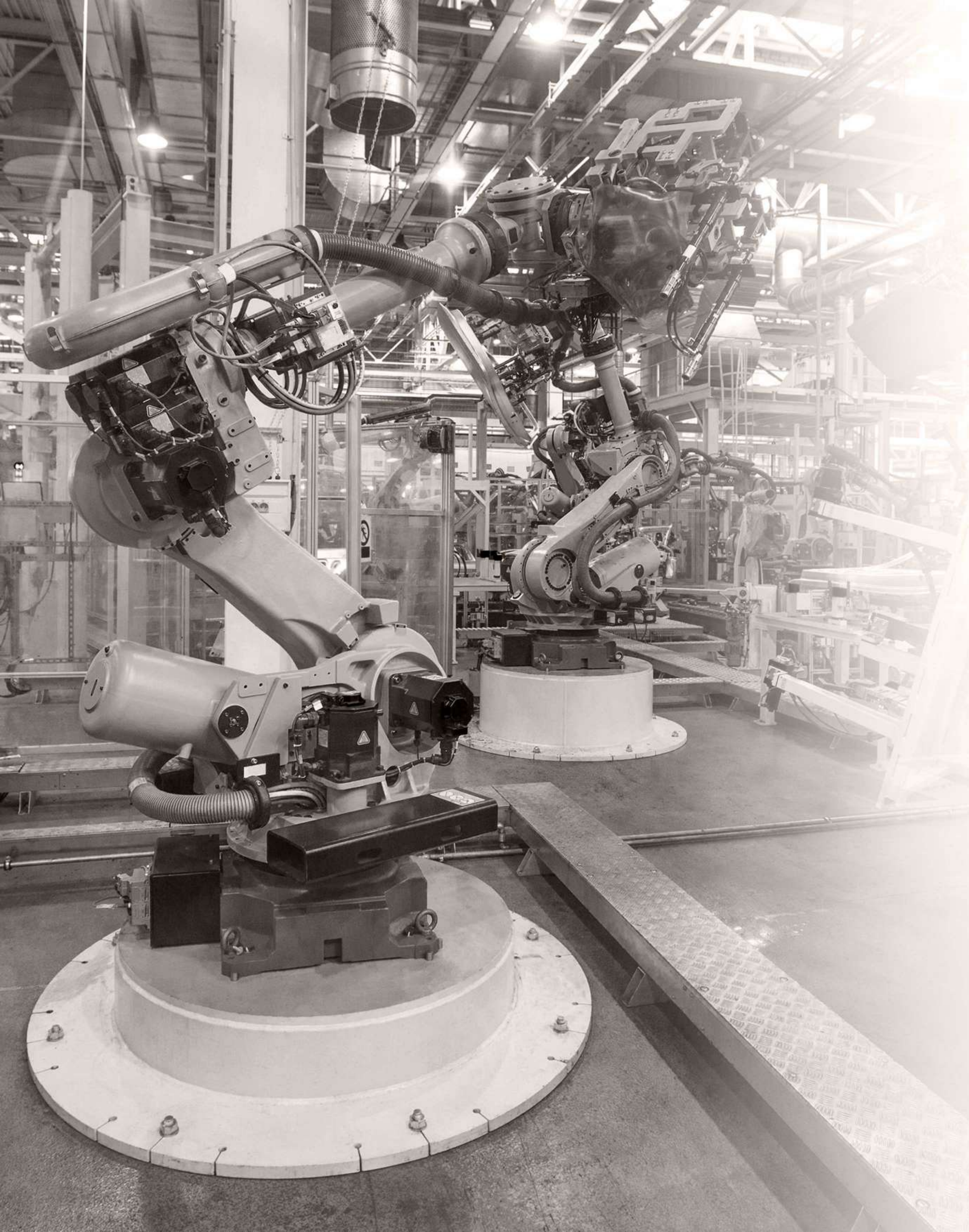
The building industry is a slow paced sector that faces enormous challenges in an incredibly fast paced modern world. Much has been said on the worldwide responsibilities of the construction sector to lower emissions and solve major housing crises by improving productivity and sustainability [1]. Although still a slow process, improvements offered by the digital revolution are starting to show increasing potential to tackle these apparent problems. Examples include the use of BIM technology for integral planning, increased use of prefab elements and broad automation of factory processes using CNC manufacturing and robotics [2].

The digital revolution has not only brought practical implementation, but has also had a clear influence on stylistic architectural expression. Designing freeform architecture, like Frank Gehry or Zaha Hadid have done, has become significantly easier with 3D CAD software [3]. So easy in fact that it has created a detachment between the designer's expectations and the physical complexities of these shapes, often leading to prohibitively expensive, environmentally costly and low productivity custom solutions. This in turn creates a disparity between freeform high-end architecture and the previously mentioned future ambitions of the building sector.

Freeform shapes in architecture are currently reserved to a small subsection of all construction projects. Although the implication seems to be that improvements will only have a relatively little effect on the industry as a whole, certain trends paint a different picture. The exemplary influence that high-end architecture has on both the aspirations of clients and designers, should not be underestimated. In addition, recent developments have shown that optimised freeform expressions are increasingly available to standard building projects. Also, the growth of freeform architecture has been intrinsically linked to digital software, which is still being rapidly developed. All these factors combined show a trend of freeform architecture becoming more desired, more feasible and therefore more prevalent.

With growth in freeform architecture expected to take off, the unique opportunity to preemptively tackle issues of productivity and sustainability for new developments presents itself. The exemplary function of high end architecture should not only be used to promote new architectural expression, but should also claim its responsibility to be an example of increasing productivity and sustainability. One way sustainability can be increased is by making use of reusable building elements [4]. Automation can significantly increase productivity [1] and should not only be considered during the fabrication process but should be implemented in on site construction [5].

As one of the most intricate parts of contemporary construction, any improvement to freeform architecture will inherently be complex. Many improvements can, and should, be explored and developed but not all will fit into one master thesis. To narrow the scope, this thesis will focus on analysing, motivating and optimising circularity and automation for nodes and beams in freeform architecture, organised under the following research question: "How can a design to production workflow be developed towards automatic assembly and circularity of nodes and beams in different freeform building facades?". This research aims to educate on and implement a novel node and beam system to enable automatic reassembly of freeform building facades.



2 Methodology

This chapter will provide a research method by first formulating the problem statement that outlines the foundations of this research in a single paragraph. Second, the practical objectives to solve the proposed problem will be introduced. Different objectives will then be used to inform the research question and sub-questions and a methodical approach to answer these questions is provided. The entire system of research will then be summarised in a framework and a planning will be provided to reach a conclusion for all objectives within the available time-frame.

2.1. Problem statement

Realising freeform building geometry requires complex and time-consuming processes in computational shape rationalisation, fabrication of custom nodes and beams and in-situ construction. Custom building elements are not suitable for reuse and are likely to be recycled in a relatively high energy-consuming melting process.

2.2. Objective

A symbiotic and supporting relationship between literature research, technological development and iterative design is desired. Throughout the entire process the relative weight of these aspects is liable to change and yet not one of them can ever be discarded as each can positively reinforce the process of the others. Development is only possible with a broad understanding of the appropriate literature and conversely focused literature review is only possible when developmental limitations are understood. Similarly, any design process should be approached with due regard to boundaries understood through technological development and literature. This framework aims to support an integral research and design approach directed at developing a design to production framework towards automatic assembly and circularity of nodes and beams in different freeform building facades.

Before any design process can be considered, preliminary research and technological integration has to be conducted to establish boundary conditions. One such technological integration is the rationalization of freeform architectural shapes. Any interpreted adaptation of the initial shape of an architectural object to refine constructability can be labelled as a rationalization, one example is the process of panelizing an input shape into developable elements [6]. Increasingly complex steps can be taken to further increase the simplicity and thus the feasibility of the realisation. For this thesis, rationalization decisions in the development of freeform architecture must be researched first as these choices are of paramount importance to the feasibility of all further steps. To facilitate research into different considerations and designs within this thesis a computational tool was integrated that will form the backbone to all further development.

By generating and testing many diverse design configurations in the computational model, several design parameters like rotation axes, angle ranges and node similarities can be established. The results can then be used to make an informed decision on the direction of the design process to create a system that will enable the reemployment of beams and nodes. The found parameters can furthermore be used for a list of requirements. Hypothetically, nodes or beams with high similarity could be reused or nodes and beams with adjustable configurations could be developed. Quantitative analysis will determine the preferred approach.

Since optimisation of geometric starting conditions has been a main consideration in the computational workflow it should also be a main consideration in the physical fabrication and assembly processes. Designing parts inside the limits of automatic fabrication methods is referred to as design for manufacture. How well this is implemented is often a deciding factor in the feasibility of a design and as such should be central to the design process. Additionally, where robotic automation has undeniably skyrocketed productivity in factories, implementation of this technology outside a controlled environment has only recently seen development. Although still in its infancy, robotic automation of the construction sector promises to increase productivity through on-site automation. The state of the art in robotic construction will be reviewed and findings will be used to both inform the design process of node reusability and to recommend technical development towards a well-integrated robotic construction solution. A practical implementation of robotic construction at a small scale is used as a proof of concept and a theoretical implementation at full scale is designed to sketch a vision of the future of robotic construction and reusable building elements.

2.3. Research questions

Main question

The aim of this research is to develop a novel design to production workflow for freeform building facades with a focus on computational optimisation, automatic processes and circularity of elements. This research will be conducted under the following research question.

> "How can a design to production workflow be developed towards automatic assembly and circularity of nodes & beams in different freeform building facades?"

Sub-questions

The structure of this research is defined by a collection of sub-questions that further specify each step along the process. In this section the sub-questions will be introduced and, where applicable, the literature search methodology will be described and a limited selection of the associated literature will be presented.

> "How can optimal rationalisations of freeform building facades be determined?"

To answer this question a literature review into rationalization theorems is required. A research survey titled "Architectural Geometry" by Pottmann et al. [6] is comprehensive and useful source. Since rationalization theory has a significant overlap with Discrete Differential Geometry a general understanding and explanation of this topic is also required. "A Glimpse into Discrete Differential Geometry" by Crane and Wardetzky [7] clearly explains these mathematical intricacies. A more specific literature search has also been started via Scopus and through analysis of specific journals and this yielded 13 references.

> "How can theory on mesh rationalization be applied to define the design requirements and boundaries of a reusable nodes and beams system for freeform building facades?"

This sub-question adopts much of the same literature used in the previous section but here it will be utilised to apply the computational method. This subsection will contain technical development that will enable the creation of a shape generator which can be used to collect quantitative data to inform design boundaries. Some preliminary research will be done into facade technology and statistical principles will be applied for data analysis.

> "What is the state of the art in robotic construction and how can it be used to inform the design of reusable nodes and beams in freeform building facades?"

This subsection will again be characterised by an extensive literature review into the state of the art of robotic construction. A research survey titled "On Site Autonomous Construction Robots: Unsupervised Buildings" by Melenbrink et al. [5] is a useful general reference. An exhaustive literature review using Scopus yielded 226 sources of which 27 were deemed relevant to this research. A selection of secondary sources led to a total of 68 sources.

> "How can a reusable node and beam system for freeform building facades be designed and automatically assembled?"

Previously gathered literature and data is used to define the design boundaries of the reusable node and beam system. This subsection will outline the iterative design process. The computational method needs to be extended and tested in a case study. The final design will

be detailed and produced to scale. Automatic assembly will be be achieved using the UR5 robot arm and DHAG95 gripper at the TUDelft Lama Lab. The software choice for robotic programming will be explored and a scale model will be automatically assembled.

> "How can the designed nodes and beams be used in a computationally informed robotic construction process to automatically assemble full scale architecture?"

This chapter will be structured into five parts. First, an architectural design case that is highly related to development and expressions of freeform architecture will be introduced. Second, a design context which supports the introduced architectural expression is chosen. Third, the design concept will be developed and its significance discussed. Fourth, the design process will be described and the final design introduced. Finally, the process of part manufacture and automatic construction of full scale architecture will be elaborated on.

Throughout the entire research the quality of sources are scrutinised to ensure a high academic quality of this thesis. Abundant contemporary sources will be consulted, compared and discussed to ensure the academic relevance of the provided research.

Scope

The scope of this research is purposely limited to nodes & beams. Although facade panels will be discussed as an important part of rationalisation theories, the added complexity of also considering the automation and circularity of facade panels could not be afforded within this master thesis. For this research to be compatible with the many different generation methods for freeform geometry in existence, it should be provided with an amount of flexibility. Additionally, various rationalization methods exist and to limit the scope, a quad panel approach based on orthogonal lines was chosen. A distinction is made between a practical and theoretical implementation. The practical implementation is limited to the available technology at the Faculty of Architecture and the Built Environment at the TU Delft. The practical design and assembly process may conceivably have to be adapted to be compatible with available hardware. A theoretical design and assembly process will be explored that is not bounded by hardware limitations and specifically considers potential of future technologies.

2.4. Framework

In order to further clarify the system of research used in this graduation thesis a visual representation is found in figure 2.1. This framework is both vertically and horizontally divided into three parts. The activities required to finish this thesis are split between research, development and design. The linear process contains three phases: exploration, experimentation and integration. The visual framework also shows that the process is not purely linear. Many steps, mainly in the design process, create loops to iteratively improve versions.

2.5. Planning

In figure 2.2 a planning for the graduation process is visualised using a Gannt chart. Due to ancillary activities as a student assistant for a faculty project spanning two years from September 2022-2024, this graduation process spans 1.5 years which is two times longer than a standard process. This delay does not indicate more than standard hours spent on the graduation process and has as such been approved by the board of examiners of the Faculty of Architecture and the Built environment.

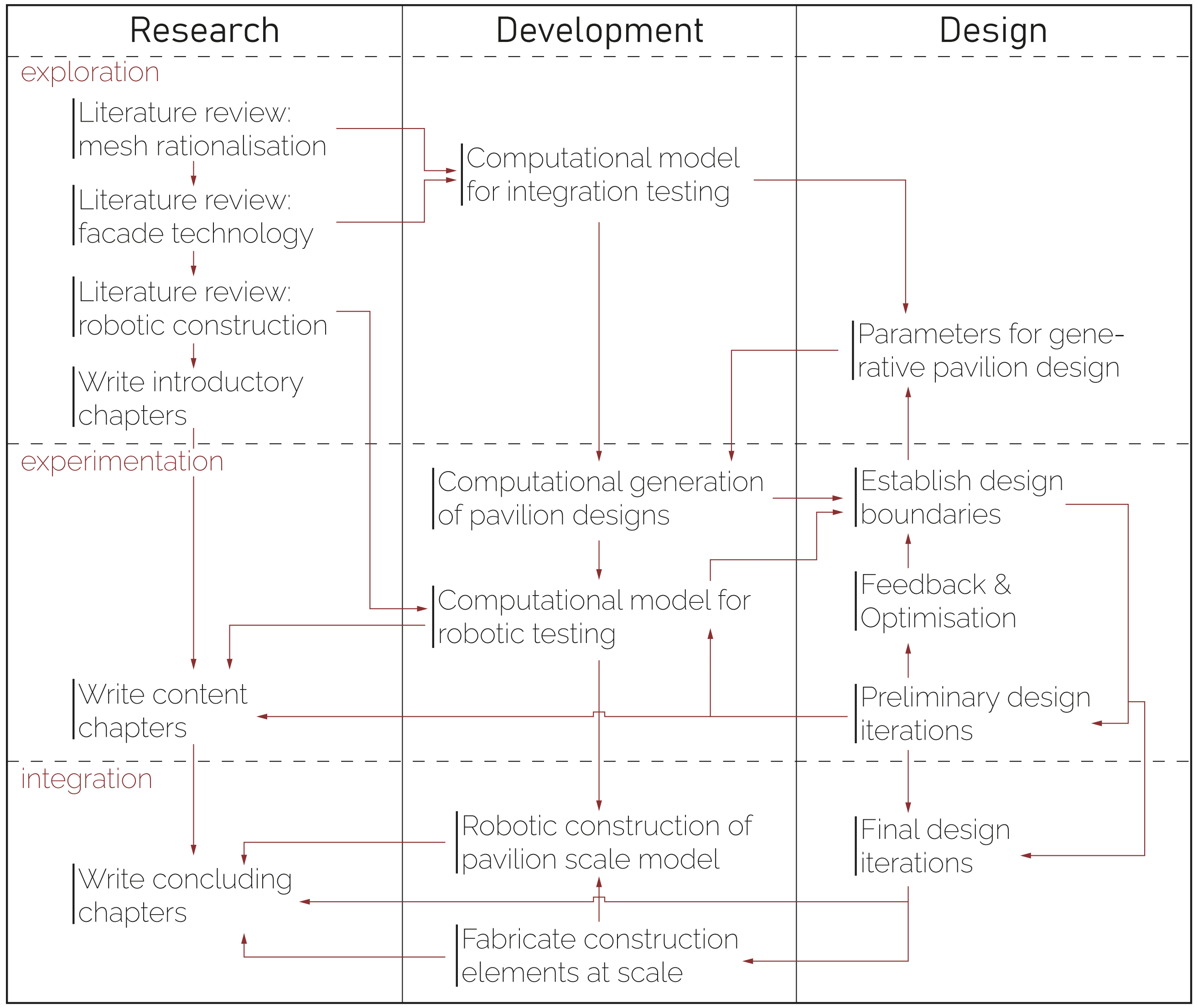


Figure 2.1: Research Framework

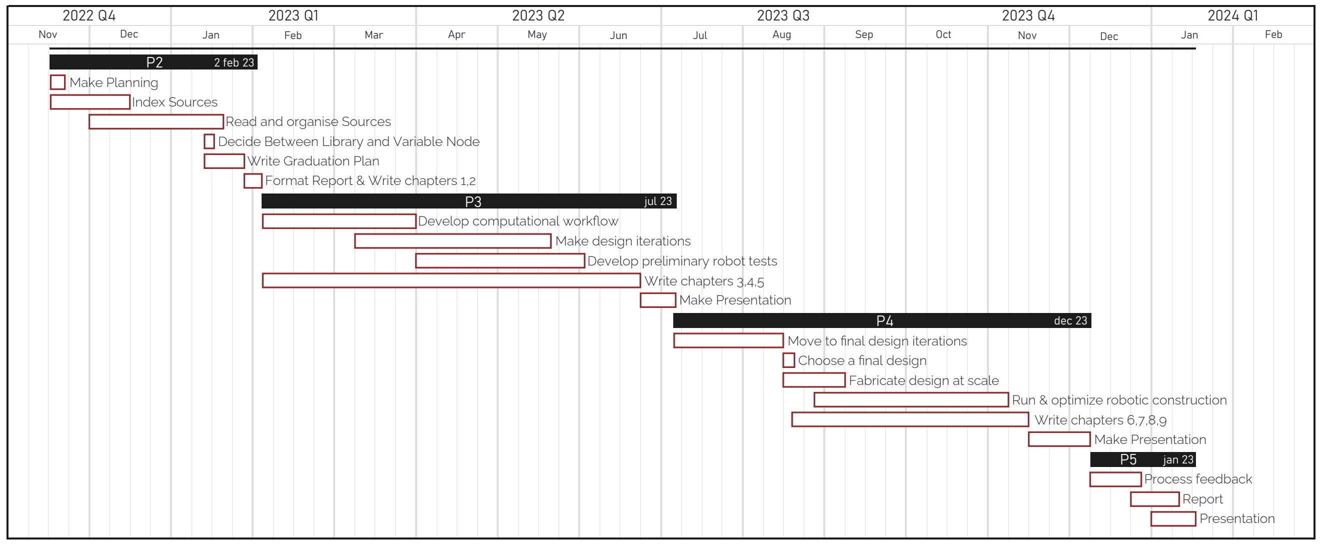


Figure 2.2: Gantt chart for planning of the graduation process

3 Mesh rationalization

Transformation of theoretical smooth surfaces to rational building systems is of the utmost importance in both the practical realisation and the design language of the architecture. This chapter will answer the sub-question:

> "How can optimal rationalizations of freeform building facades be determined?"

To approach this multi faceted problem, this chapter will first provide an overview of contemporary literature on both freeform modelling and mesh rationalization. After which the developed approach will be presented and substantiated. Further optimisation methods will be introduced, tested and implemented.

3.1. Freeform Modelling

Before the topic of generating freeform shapes can be approached the definition of *freeform* must be clarified. Though freeform can be interpreted as any shape imaginable, in literature the term is often coupled to geometry that needs to be generated with a strict rule set. Research shows that rationalization can become significantly easier when the generation of shapes adheres to certain rules [8] [9]. For the scope of this thesis, freeform is deliberately defined as any geometry containing a single surface. How this surface is generated should not be relevant and there are no other expected characteristics inherent to this geometry. This is done to create the broadest design space within the early modelling phase.

The next logical step is to actually generate a freeform surface. To achieve this, surface modelling software Rhinoceros 7 in combination with its plugin Grasshopper is utilised. The Freeform Surface modelling category in Rhino already contains more than 20 different methods of generating geometry, too many to discuss individually. Within this thesis a method using a loft between two interpolated curves is used which enables the creation of a true freeform surface that is also doubly curved, as seen in figure 3.1, adding a challenging complexity to the next steps.

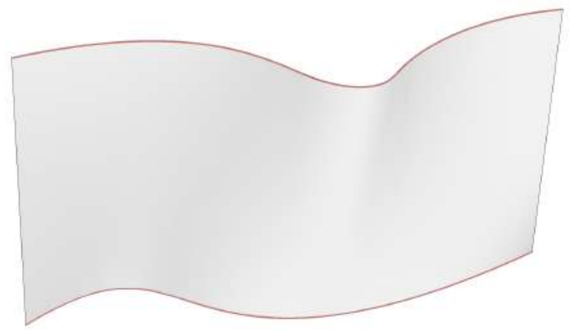


Figure 3.1: Doubly Curved input Geometry, own work

3.2. Rationalization

Since an infinitely smooth surface is impossible to construct, it has to be rationalized. *Rationalization* can be defined as the approximation of an ideal design surface by a surface which is suitable for fabrication [6]. Another term for rationalization often used in architecture is *panelization*: generating manageable panels to construct a larger overall shape. Since panels are often not part of the load bearing structure, architectural rationalization also entails generating a structural system of beams and nodes. Optimised rationalization is often a significantly more complex process than generating the initial shape [6].

Triangles & Quads

The origin of rationalization can be found in the need for computer graphics to work with *polygon meshes*: a collection of vertices, edges and triangular faces. The geometry generated in this way is called *discrete geometry*, as it is composed out of discrete parts. An infinitely smooth surface is defined as *differential geometry*. Research into combining these two geometric principals is done under the umbrella name of *discrete differential geometry*. The importance of such research for freeform architecture cannot be overstated. Someone with an architectural background might experience difficulties comprehending the full mathematical principles, but once the mathematical theories are implemented within software packages their real impact becomes quite clear. Crane and Wardetzky [7] provide an approachable introduction to this topic.

Once geometry needs to be converted to an actual facade system, reality demands a different approach to rationalization. The research area that deals with the raised complexity of geometry as a result of real world requirements is called *architectural geometry*, as described by Pottman et al. [10] in their comprehensive book by the same name, see figure 3.2 for a visual example. Although perfectly fine for a computer visualisation the top mesh, due to its long and varied edges, will never be practically viable to construct. Rationalizing this model with the objective of even edge lengths, as shown on the bottom, will already make it more structurally feasible.

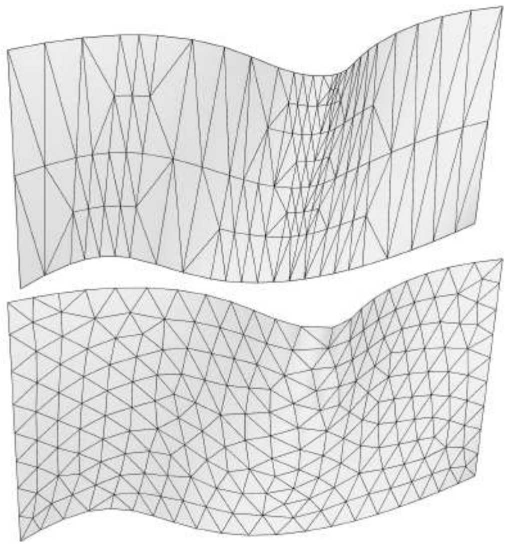


Figure 3.2: Top: A discrete mesh usable for computational graphics, Bottom: Edge-Length optimised rationalization for better constructability, own work

Rationalizing complex geometry for architecture is a multilayered balancing act which can be compared to a performing juggler balancing on a moving seesaw. An important balance to achieve is the one between constructability and designer freedom, but also a balance between different systems and their pros and cons. Research in this field in ongoing but has recently shown a tendency that prefers quad meshes over triangular meshes [11].

Triangle meshes come with two major advantages, the faces are guaranteed to be planar and the construction is intrinsically stable [10]. Triangles accommodate easier adaptation to the designed shape all the while keeping their intrinsic advantages. A classic example is figure 3.3, the Great Court Roof of the British Museum, for which aesthetic optimisation was performed to create an ordered distribution of triangular panels [6]. The most significant disadvantages to triangular meshes are the complex 6 valence nodes, see figure 3.4, the visual presence and the weight of the construction and the inefficiency of cutting triangular panels from rectangular blanks [10].

Figure 3.3: The triangulation of the Great Court Roof of the British Museum, via The Guardian

Figure 3.4: A 6-valence node for a triangular construction, by Waagner Biro

Quad meshes are meshes with rectangular faces and can be seen as the opposite of

triangular meshes when it comes to their pros and cons. Quad panels are efficient to cut from blanks, need less complex 4 valence nodes and have a lighter and less present construction. The fact that they are not intrinsically stable or planar [10] raises the question if these issues can be solved. Planarity of quad meshes is an extensive focus area of architectural geometry research [12] [13].

Before the attention of the building industry fell on *planar quad* (PQ) meshes, they were already known to discrete geometry research as rationalizations of surfaces using their conjugate curve network, visualised in figure 3.5. However, the use of a conjugate network does not allow for the desired design freedom as not all conjugate curve networks are practical to use for construction purposes [6]. In an attempt to solve these inadequacies the idea of *PQ mesh perturbation* was first established by Liu et al. [14], by allowing the mesh to settle through small adjustments optimising a planarity cost function an input shape can be perturbed to become a PQ-Mesh.

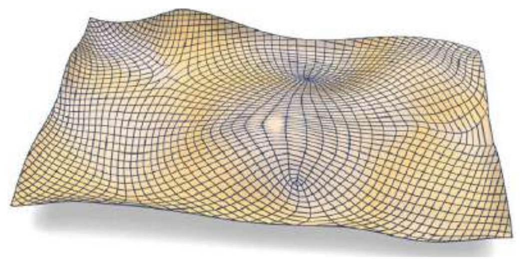


Figure 3.5: The Roof of the Islamic Art Museum in the Louvre, re-envisioned to use its conjugate curve network, by Wallner and Pottmann [11]

Torsion Free

Another more complex issue that arises when geometry is considered for architectural purposes is that of torsion between neighbouring nodes. This arises when two neighbouring vertex normals are not in the same rotation around the axis of the connecting edge. Compared to infinitely thin computational meshes where torsion can be ignored, in architectural implementations unresolved torsion will cause a higher node complexity or non-planar support beams, both costly and undesired [15]. The most useful characteristic of a torsion free mesh is the existence of an offset mesh at constant face-face distance [13]allowing nodes to be simpler and support beams to be planar, reducing cost and manufacturing complexity.

The most prevalent way in literature to test if a mesh node is torsion free, is by checking if that vertex is conical. As introduced by Liu et al. [14] and defined as: "A vertex v of a quad mesh is a conical vertex if all the four face planes meeting at v are tangent to a common sphere. This is equivalent to saying that these oriented face planes are tangent to a common oriented cone of revolution G". This conicality can be calculated using the following theorem, as proven by Wang, Wallner, and Liu [16]: "A vertex v of valence 4 is conical if and only if the sums of opposite angles are equal, i.e., $\omega 1+\omega 3 = \omega 2 + \omega 4.$". A visual aid for this can be found in figure 3.6. A freeform shape is never intrinsically torsion free, this holds true for both triangular and quad meshes. However, research shows that a triangular mesh can never be torsion free, while a quad mesh can be perturbed to become torsion free [17].

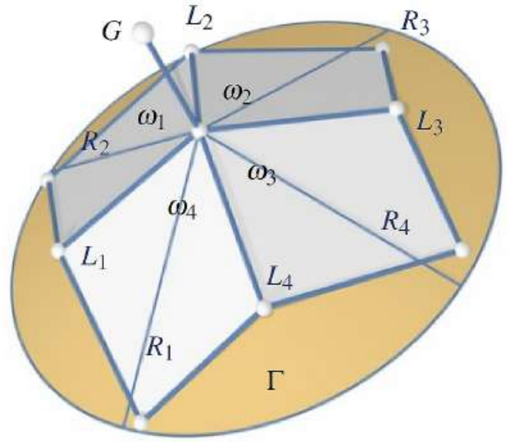


Figure 3.6: Visual aid in calculating if a node is torsion free, by Wang, Wallner, and Liu [16]

To conclude, although triangle meshes offer stability and planarity, due to their high node valence, inefficient panel manufacturing, and unresolvable torsion there has been a growing preference to using quad meshes. A combination of a lower node valence, more efficient panels and the ability to solve both planarity and torsion using perturbation establishes the quad mesh as the rationalization method with the best potential.

3.3. Chosen Approach

As the previous research presented a preference of quad over a triangle meshes, the choice is made to narrow the scope to only include quad meshes in this thesis. More specifically, a rationalization based on a division using horizontal and vertical lines is chosen, as shown in figure 3.7. Although using intrinsic qualities of the input surface, like the conjugate curve network, can result in more optimal rationalizations, these methods are not possible with every shape. Therefore, to guarantee compatibility with any shape, the chosen rationalization method does not use the intrinsic qualities.

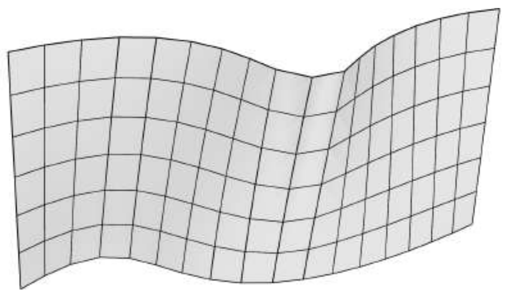


Figure 3.7: The chosen rationalization, based on orthogonal lines, own work

The quads rationalized on the surface in figure 3.7 are not intrinsically planar or conical as initially calculated by analysis scripts in figure 3.8. The method to perturb a mesh to allow for small movements to create a planar and conical mesh has seen development in literature [14] [17]. Although important and valuable, this type of development can only be taken furthered to a higher level when access to the technology is simple and open. Grasshopper is a great example and building on its strong foundation Piker [18] developed the plugin Kangaroo, which allows for computational form finding through perturbation of meshes. Already implemented with the ability to planarize and conicalize meshes, Kangaroo is perfect for the optimisation process.

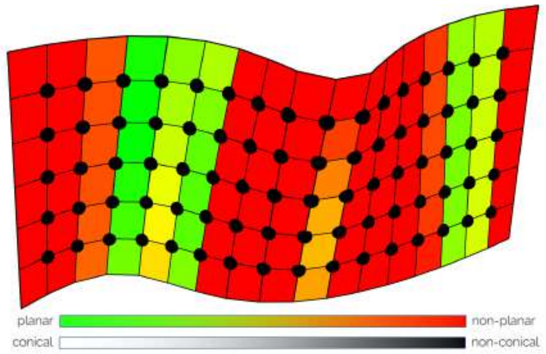


Figure 3.8: Initial analysis proves the mesh to be non-planar and non-conical, own work

As with most other steps of the rationalization process, Kangaroo also demands a balancing act between designer freedom, physical limitations and simulation success. A simple simulation would be to activate the planarization and conicalization optimisation goals without any provided boundary conditions. Due to the lack of constraints this does not output a desired result, as shown in figure 3.9. In order to prevent these undesired results a soft pullback force back to the initial location can be added for every node. For architectural purposes it is often desired to align structural elements with floors, established by locking movement in the vertical axis. This is visualised in figure 3.9.

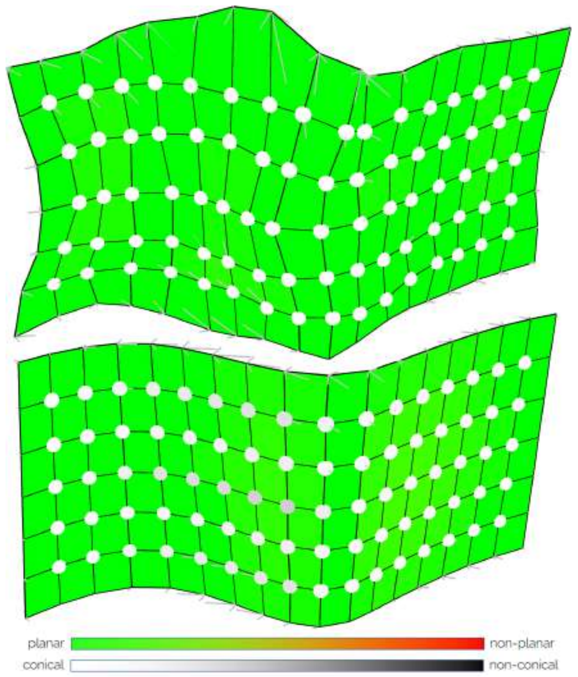


Figure 3.9: Top: Result of no constraints. Bottom: Result with the introduced boundary conditions, own work

Although the result with boundary conditions is better, it is still varies significantly from the input shape, especially high curvature areas have become flattened. When accuracy to the input shape is important, multiple spaces of complexity can be considered as shown by Passas [19]. Thereby less curving areas of the input shape are more rigorously optimised than high curvature areas. A balance is established between reducing costs in low curvature and increasing accuracy in high curvature areas. To add this functionality to the Kangaroo simulation the strength of the pullback force is modulated based on the curvature at the node location on the input shape. The data modulation formula is graph-based and user controlled to provide the designer the tools to control the balance between optimisation and design. The result of this is shown in figure 3.10.

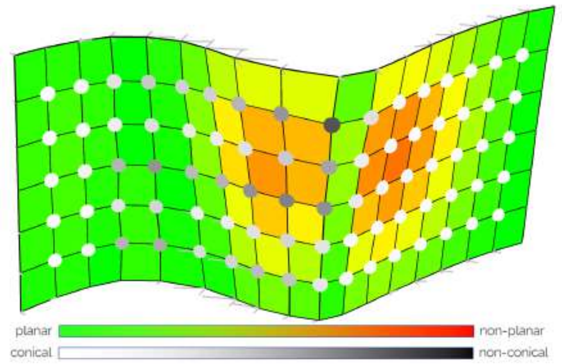


Figure 3.10: Result of a balanced optimisation process, own work

3.4. Approach Complexity

Although working with any doubly curved surface is challenging, the chosen surface is both horizontally and vertically finite. A valid argument can be made that this is not the most complex doubly curved surface to rationalize and optimise, especially compared to dome like surfaces with multiple convex and concave areas. Which will either not allow rationalization using orthogonal directions or they are so inefficient for later optimisation that they cannot be considered. It is these more complex surfaces where using the conjugate curve network or other intrinsic qualities is significantly more important to create feasible meshes for later optimisation [20].

Conjugate curve networks or other curvature based algorithms to rationalize surfaces are complex. In the panelization process certain areas will devolve into panels that become infinitely smaller, this is called a *singularity* [21]. Solving singularities is a significant part of discrete differential geometry research and for architectural implementation it will often result in a non-quad node with pentagonal panels around it. Not only is the computational solving of singularities complex, it also complicates all further steps as the data structure becomes non-uniform. While this thesis acknowledges the potential of using these more optimal rationalizations. Due to their complexity, the scope is limited to the method shown in figure 3.7. However it is this authors opinion that, with enough time to develop a robust computational system, it will be possible to implement the more optimal rationalizations throughout all consequent steps in this thesis.

3.5. Conclusion

Central to this chapter is the sub-question: "How can optimal rationalizations of freeform building facades be determined". If anything, this chapter shows that the answer to this question is not clear cut. Due to the complexity of the material, no answer can provide the objective truth the question desires, rather the question should be answered in the following way: objective optimal rationalizations of freeform building facades do not exist, a balance between optimal results and design freedom always needs to be fine-tuned. Using a mesh perturbation method to balance between goals to planarize and conicalize a mesh and a curvature based variable pullback force, the designers receive direct control over the balance and while they will not find an objective solution, they are now in full control to find their own subjective, optimal result.

4 Design Boundaries

Before a system of reusable nodes and beams can be designed, its design requirements and boundaries need to be determined. With the background of mesh rationalization established, this chapter will put theory into practice by answering the following question.

> "How can theory on mesh rationalization be applied to define the design requirements and boundaries of a reusable nodes and beams system for freeform building facades?"

The answer to the question is five-fold. First, a concrete definition of nodes and beams and their internal parameters will be established. Secondly, a distinction will be made between two possible systems of reuse. Thirdly, a shape generation algorithm will be explained and implemented. Next, a conclusion to the proposed hypothetical will be based on the similarity of nodes and beams between distinct shapes with varying rationalization parameters. Lastly, design requirements and boundaries will be established based on the previous conclusion and the mesh theory of chapter 3.

4.1. Nodes and Beams

Before design boundaries concerning nodes and beams can be determined, their definition within the design space need to be specified. The technology that currently enables freeform building façades has its origins in the historic desire to have increasingly more transparent façades, the end goal being a fully glass façade. Since glass panes are not structural elements this requires the addition of a slender, strong structure to support the glass, a curtain wall [22]. Beams within this system are referred to as transoms and mullions, horizontal and vertical respectively. In a flat and rectangular curtain wall all internal connections have angles of 90°. This foregoes the need for complex nodes as connections can be made directly by using simple brackets, as seen in figure 4.1.

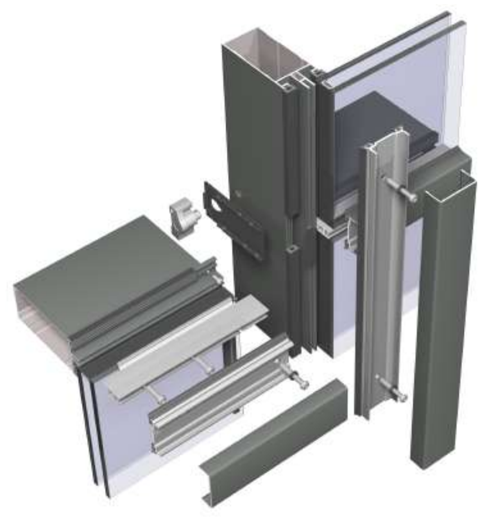


Figure 4.1: An example of a standard curtain wall system with a 90° connection, by Reynaers Aluminium

The rise of CAD applications in the early 21st century enabled any shape imaginable to be modelled, causing a significant increase in projects containing freeform shapes [3]. When compared to the refined and simple solution of flat curtain walls, creating curved geometry poses a complex new challenge for façade engineers. A construction system to realise freeform shapes needs to connect elements at many different angles, causing a need for discrete pieces for connecting the beams: a node. Different systems of discrete nodes have been developed, each with different capacities to adapt to freeform shapes [23], a few examples are shown in figure 4.2. The fact that torsion between individual nodes still has to be compensated for, is a common limiting factor in these systems. Any rotation axis that a node should be able to compensate for, needs to be machined into its connection to the beam. Compensating for an extra axis therefore requires an extra axis of machining freedom, requiring more complex machines or resulting in less automation of the manufacturing process due to necessary human intervention, both resulting in increased costs.

While costs of a freeform architectural sys-

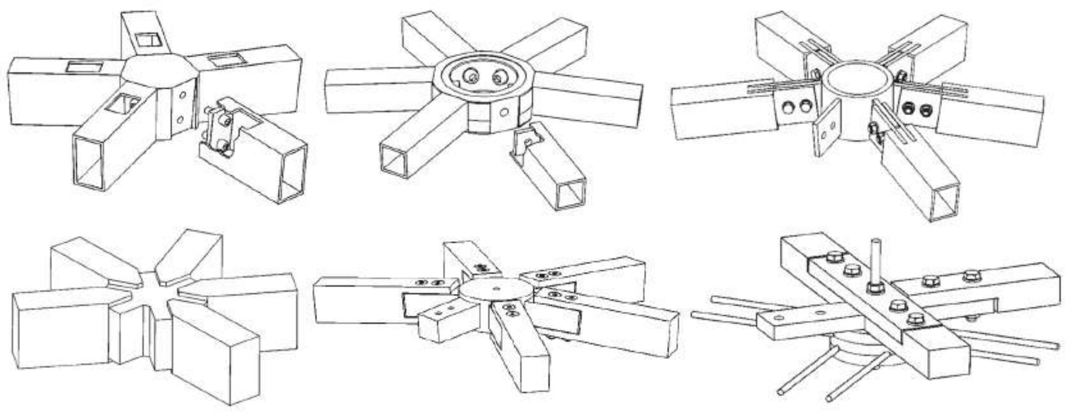


Figure 4.2: Examples of different node systems, by Stephan, Sánchez-Alvarez, and Knebel [23]

tem is an important consideration, it is not the only factor that should be considered. In a world that is increasingly aware of its environmental footprint, a system in which discrete parts lack reusability should be seen as a dissonant element towards a circular future. In order of importance circularity is colloquially defined as Reduce, Reuse, Recycle. Considerable steps have already been made in recycling end of life façade systems. The next step is designing a freeform facade system that can readily be reused.

4.2. Library or Variable

In the initial phase of this research two hypothetical ideas for systems in which nodes and beams can be reused were proposed. Since these two potential systems are very distinct, empirically substantiating a choice is an important step to specify the early research direction. The first system that was considered is based on a kit of parts approach. Nodes and beams would first be manufactured to the required angles and lengths and used in the initial façade. Upon reuse in a different facade of another freeform shape an analysis would have to be made to determine which nodes and beams are similar enough within tolerance to be eligible for reuse. The missing elements would then have to be manufactured and the leftover elements from the old façade could be stored in a warehouse as a sort of library of parts to become available for other, differently shaped facades.

The second hypothetical system that was considered is a variable node and beam system. In this system the nodes and beams are designed to be variable. These elements will be first be manufactured and later be configured to the correct parameters for the first freeform façade shape. Upon reuse they would have to be reconfigured for different angles and lengths and used in a new facade. The elements will have to be designed to both be configurable and lockable in different configurations. As long as the old and new façade have an equal number of beams and nodes this could theoretically achieve a 100% reuse rate.

Both systems have pros and cons. A library of parts would facilitate the use of the simple and proven monolithic elements as long as their system is standardised. However, the size this library needs to be before it is effective is directly linked to how similar the elements are between different freeform shapes, defined as the element overlap. The variable approach adds a significant complexity to the design of the node and beam. Designing for lockable degrees of freedom increases complexity and cost and furthermore will raise many technological uncertainties that would need to be proven before adoption could be considered. Additionally the variable system can promise a 100% reuse rate.

In conclusion, if the element overlap is large enough to support a library of parts, this system is preferred due to its use of simpler and proven elements. However if the overlap is not large enough to support a library of parts, a more complex variable system is preferred as it can achieve a 100% reuse rate. The general element overlap between different freeform shapes will have to be measured.

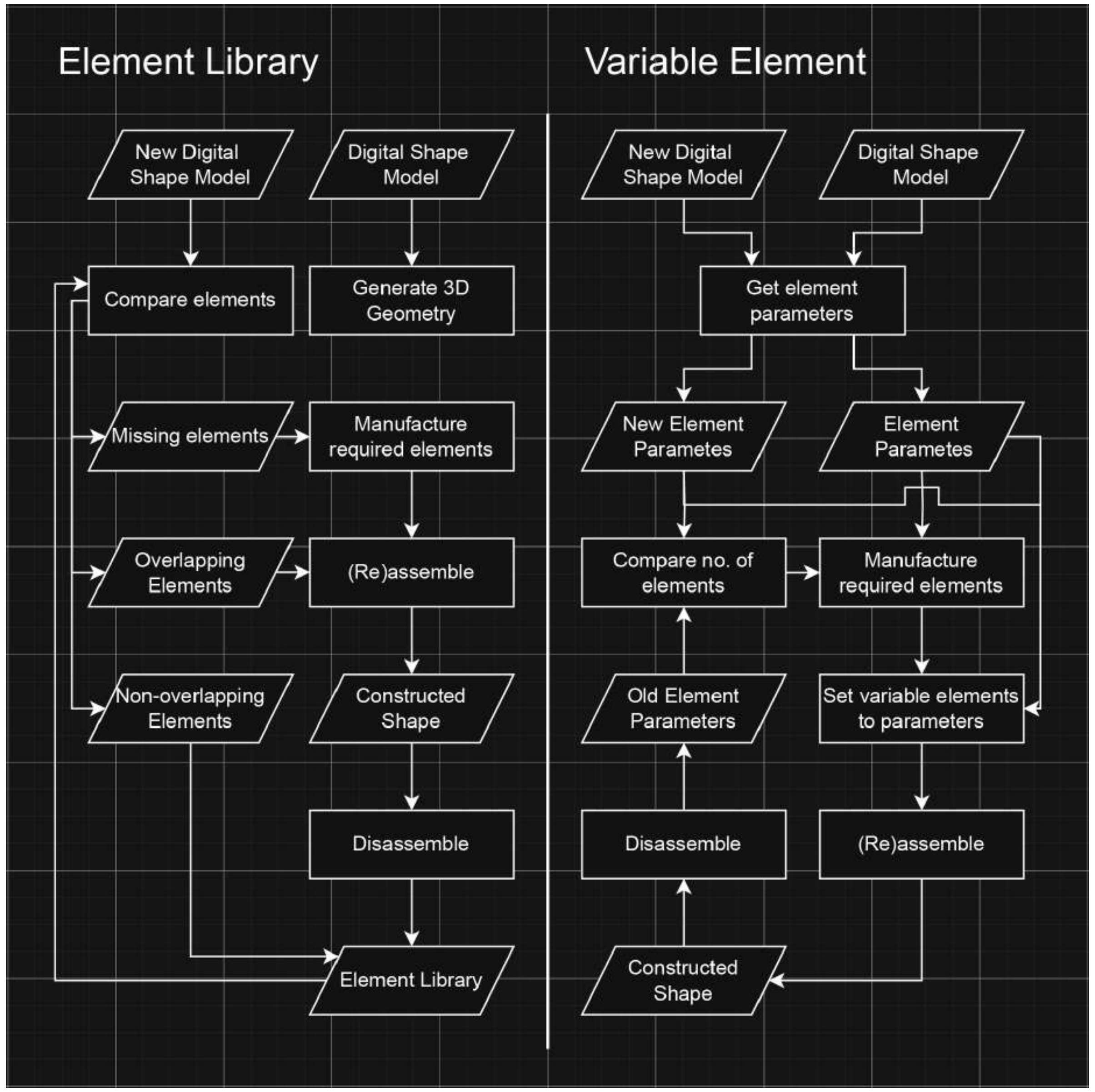


Figure 4.3: Flowcharts of the two hypothetical systems of reuse, own work

4.3. Shape Generation

The choice between a library of parts or a variable system depends on the overlap of elements between different freeform shapes. In order to measure element overlap a robust computational system was developed in Grasshopper for this research. Using many differently generated freeform shapes defined by a high degree of randomness to reduce any inherent similarity between the shapes the element overlap can be measured. The generated shapes are somewhat similar to the chosen input shape from chapter 3 as they are also defined by a loft between two interpolated curves.

The interpolated curves are generated based on six points that are randomly defined within a bounding box. The length and height of this bounding box remain stable while the width can be varied. A larger width will allow the generator points to vary more and consequently the generated shapes have a higher curvature. Since only the width is variable, all the shapes will be of the same length and height. As this would create an inherent similarity in beam length, the UV-division of the surface is randomised, which results in a randomised amount of panels and a broad spectrum of beam lengths.

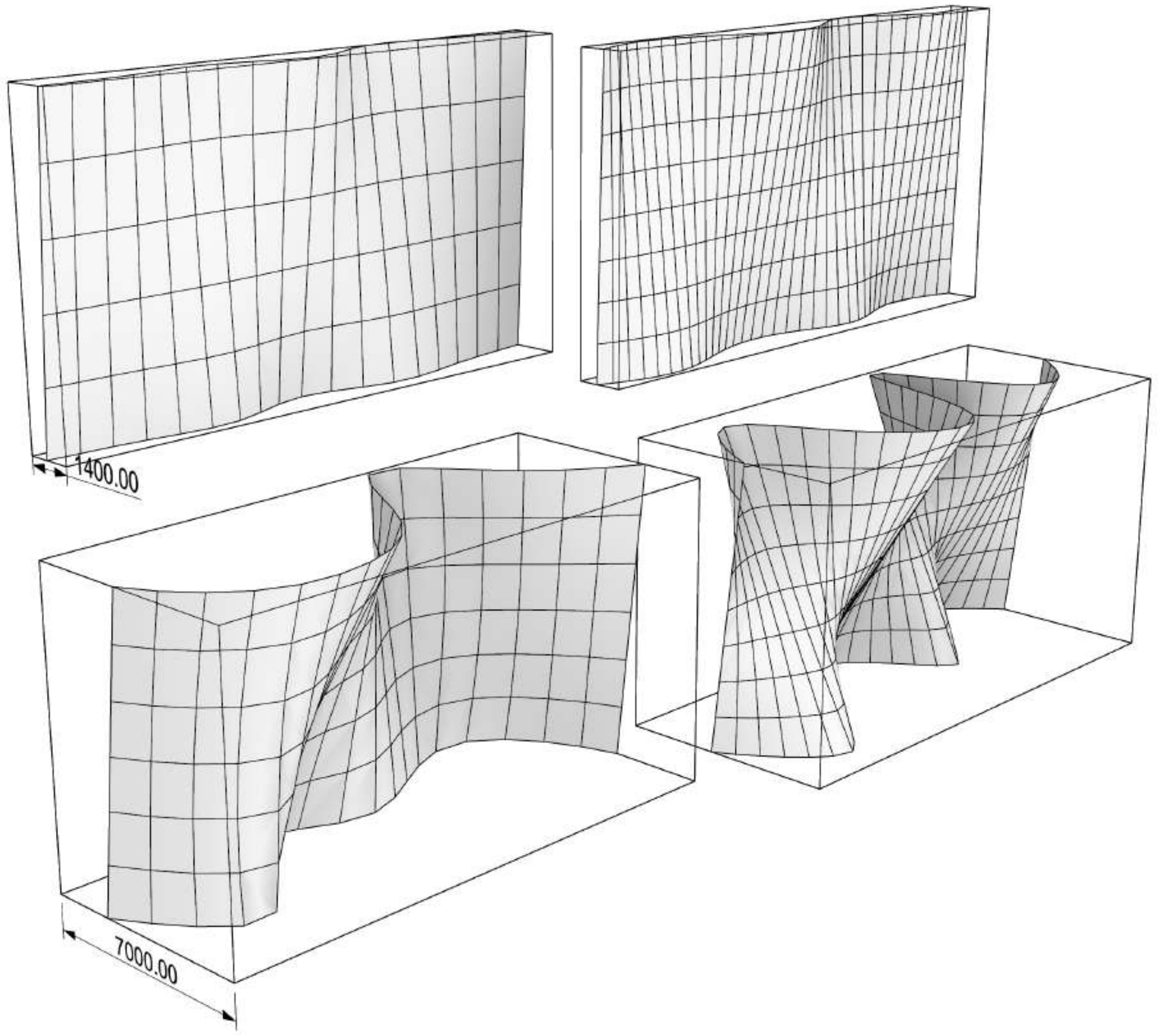


Figure 4.4: The difference between randomly generated shapes of narrow and wide curvature spaces, own work

From the generated shape the nodes and beams are extracted and then defined as their internal parameters. The nodes are defined as four X and Y rotation angles, the method to find these will be explained in chapter 6. Torsion, or rotation around the Z-axis, is not considered to reduce complexity. Beams are solely defined by their length. To allow for some tolerances the values are rounded: the beams are rounded to the nearest millimetre and the node angles are rounded to the nearest 1/8th of a degree. Since beams only have one internal parameter, their lengths are added to a list of which the duplicates are removed. While not fully accurate, as one beam should only be reused once, the reduction percentage of the list length can be considered as a simple and sufficient indication of element overlap.

Due to the many internal parameters for the angles of the nodes the system of list reduction needs to be more complex for it to provide a decent approximation. As the nodes can be rotated and thus change the order of the angles, there are four orientations that should be considered. All possible orientations are added to the list and the duplicates are removed. Again, this is not a fully accurate method as different orientations of nodes can match with each other multiple times, therefore the accuracy of this list reduction method will be worse than that of the beams. These inaccuracies in combination with not considering torsion will result in greater overlap percentages than expected. It is important to discuss the final results as approximations and keep any inaccuracies in mind when coming to a conclusion.

For the final results the width of the bounding box is gradually increased over 500 steps, gradually allowing for more curvature in the generated shapes. For each of these steps 100 shapes are generated and the list reduction percentage for the nodes and beams are calculated. The result of this computation can be seen in figure 4.5: the graph shows that the element overlap of nodes has a significantly more negative relation to increasing curvature than that of the beams. It should be noted that the earlier mentioned inaccuracies will make the found overlap percentages higher, more so for the nodes than for the beams.

An element overlap for nodes of around 50% at higher curvatures means that for every node that would be in use another would have to be stored in the parts library. A higher percentage of 90% in element overlap of beams shows more promise. Especially when considered that the simple geometry of straight beams opens another option in reuse that was not yet considered in this computation. Beams can simply be cut shorter to reduce their length, creating minimal waste and increasing the reuse percentage even more. It is safe to conclude that empirical analysis shows that for the reuse of nodes a variable system is preferred while for the reuse of beams a library system is recommended.

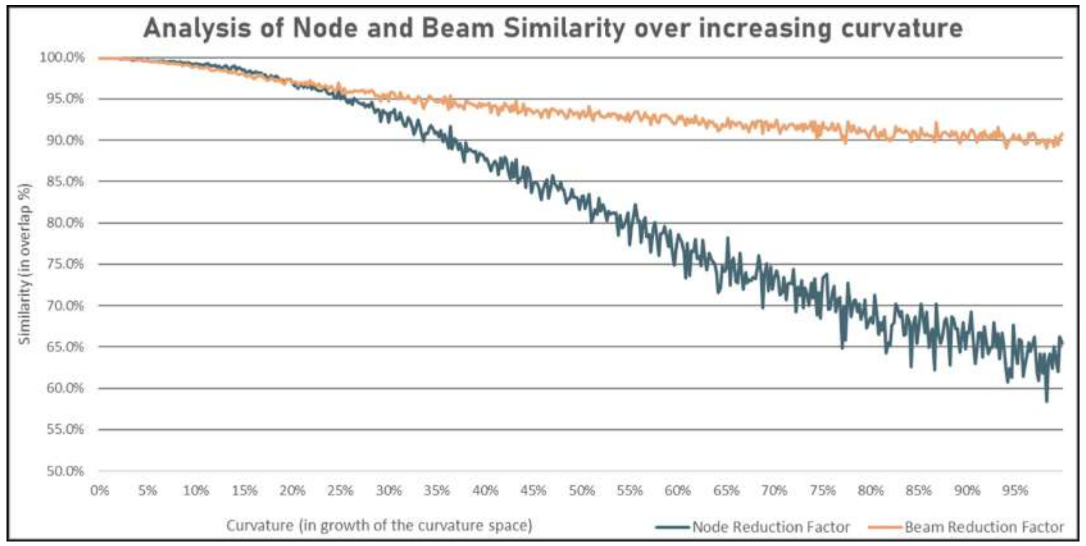


Figure 4.5: The result of element overlap computation, own work

4.4. Design Boundaries

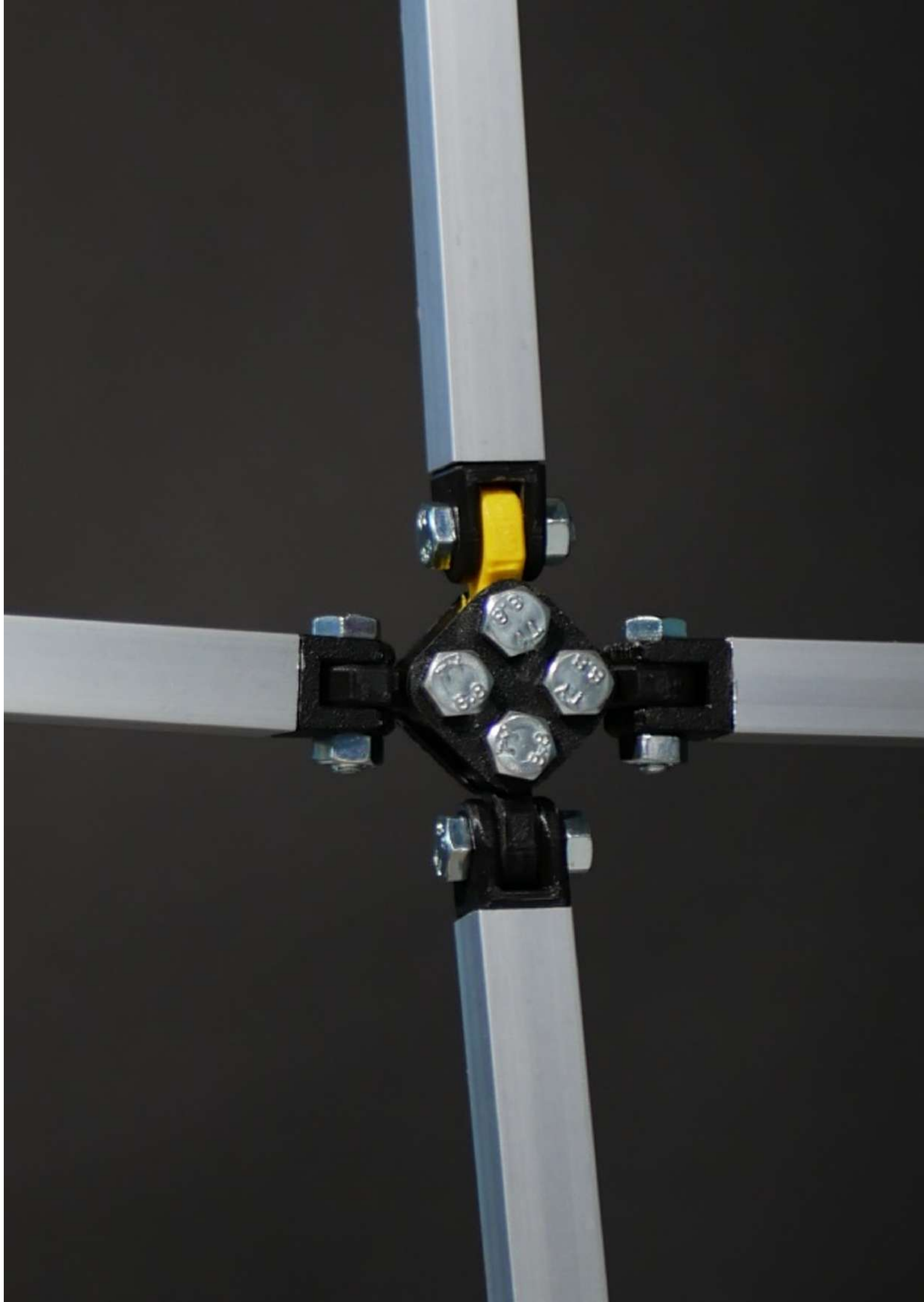
Having established that a combined system of variable nodes and library beams is preferred, the exact design requirements and boundaries have to be determined for this combined system. The most basic parameters on which all further parameters can be based are the width, height and thickness of the chosen beam profile. Every distinct version of a beam's profile can be seen as a separate standardised version of the entire system, as different profiles will not be compatible in one system. Designing nodes and beams to be interchangeable between sizes would add such an amount of complexity, if at all possible, that it will not be considered in this research. The only other relevant parameter for the beams is length and together these 4 values define the whole design space of the beam.

The design parameters for the nodes are significantly more complex. In the most basic sense the node needs to be an interface between 4 beams that point in different directions. In a torsion-free shape this entails compensating for four rotations around the X & Y axis, one for each connecting beam. In a mesh with torsion a compensation of rotation around the Z axis has to be added for every beam. All these rotational axes should be lockable once a configuration is made. Additionally, the node should be optimised to be as small as possible while achieving all these goals.

4.5. Conclusion

This chapter aims to answer the sub-question: "How can theory on mesh rationalization be applied to define the design requirements and boundaries of a reusable nodes and beams system for freeform building façades?" In order to answer this question clear definitions of nodes and beams have first been provided. Two theories on how to design a reusable system have been introduced: a variable system in which parameters can be changed upon reuse and a library system where elements that overlap are reused and those that do not overlap are stored in a parts library. Both have their pros and cons, the variable system adds significant complexity to the design but promises a 100% reuse rate, the library system can use much simpler monolithic parts, but will have a lower reuse rate. If the element overlap is high the library is preferred, if not the variable system is the most viable option.

In order to research the feasibility of the two hypothetical systems, thousands of shapes with increasing curvature were generated to find the relation between element overlap and curvature. It was found that the overlap of nodes at higher curvatures is far from sufficient and a variable system was chosen. Beams do show a stable and high element overlap at higher curvatures thus a library system has been chosen. The answer to the sub-question can be stated as follows: theory on mesh rationalization is used to generate many shapes from which node and beam parameters were extracted and these were used to make a substantiated choice between a variable and library system, which in turn defines their design requirements and boundaries.



5 Robotic Construction

With design boundaries established based on mesh rationalization and a choice made for a variable node and a library beam system, a central component of the main research question needs to be addressed: automation. This chapter aims to elaborate on this topic by answering the following sub-question:

> "What is the state of the art in robotic construction and how can it be used to inform the design of reusable nodes and beams in freeform building facades?"

In order to design a system for automatic (re)assembly an extensive review of automation in the construction sector is needed. This chapter will first introduce the general history of robotic automation in factory settings which will then be related to an extensive literature review on robotic automation in the construction sector. Mature technologies, recent technological improvements and exciting future potentials will be introduced through four levels of technological maturity and the knowledge obtained will benefit a further expansion of the design boundaries of a reusable node and beam system. Two different implementations of these boundaries will be considered. Because the first is a theoretical implementation it does not have to be limited to this thesis' practical scope of manufacture and cost. Contrary, the second is a practical implementation that is limited to this scope and will be manufactured and constructed at a model scale.

5.1. A Short History of Robotics

As with many modern day inventions, robotics has its origin in human fantasy and storytelling. Imaginations of what we would presently call robotics go back to mythologies of ancient peoples, with principles of human mimicry or task automation already explored in those times. The first use of the actual word *robot* comes from a 1921 play by Karel Čapek called Rossum's Universal Robots. The word is derived from the Czech word *robotnik* referring to peasant or serf. The play itself is a very early example of a recurring theme in science-fiction literature, a robot as a subservient human-like clone that in the end rises up to defeat its master. Another science-fiction writer with a profound impact on the history of robotics is Isaac Asimov. Writing his book *I, Robot* in 1950 containing the famous Three Laws of Robotics, which is also the first time the word *robotics* was used [24].

After the second world war a combination of an economically booming America and a generation that grew up with science-fiction created a hotbed for advancements in technology. For robotics this came in the form of a chance meeting between Joseph Engelberger and George Devol in 1956. After being inspired by Asimov's ideas on robotics, Engelberger (1925 - 2015) pursued both a degree in physics and a career in aerospace engineering. Devol (1912 - 2011) applied for a patent in 1954 for his invention of universal automation (unimation) which he defined as a general purpose machine that has universal application to a vast diversity of applications where cyclic control is desired [25]. Their meeting inspired visionary Engelberger to license the patent and start a company, creating the first Unimate robot arm in 1961 for General Motors. There it unloaded very hot parts from a die casting machine, an unpopular job among human workers. In 1983 Unimate held 25% of the world market share, blossoming into a successful company. Engelberger is widely considered to be the founder of modern robotics [24].

The implementation of the first robots provides a very good indication of the economic driving force behind robotic automation. The most successful implementations of early robots replaced jobs that fell within three categories: dirty, dangerous and demanding. The three words were combined as the '3Ds' in the 1980's. Especially in industrial sectors and in the military automating 3D jobs increased safety and comfort and has proven to be economically

Figure 5.1: Unimate, the first mass produced industrial robot arm starts work at General Motors, via Yaskawa Motoman

profitable in the long term. Another aspect accompanied by a fourth capitalised D was added later: Dull. Jobs where a worker faces a high degree of repetition can also be automated and moreover this process can be made profitable in the short term [26]. The definition of the 3Ds broadened over time, considering more jobs within more sectors for automation. Although the construction sector contains many jobs that fit the 3Ds perfectly and so could be eligible for automation, it is currently the least productive sector in the world due to its low level of digitisation and automation [1]. With the robotic revolution in factories proving its enormous value, the construction industry should also adapt to a safer and more efficient future.

5.2. Literature Review

In order to comprehend the state of the art in robotic construction a literature review structured into four levels of growing technological maturity is provided in this section. Because of very expensive hardware and the complexity of its implementation and because interest in robotic construction has only recently started growing, recent research in this area is still spread over all four levels of maturity. These levels are split into four categories. The first field of interest is movement: on the mechanics used for movement of the robotic system and what freedom they provide. The second category is sensing: on which sensors are used and how sensory information can be interpreted. Third comes solving: how are instructions for robotic movement and operations calculated or generated? The last category is operation and what the practical capabilities of the robotic system are.

5.2.1. Level 1: Simple Placing

Movement: stationary & linear

In order for a robot arm to be able to reach all possible positions and orientations within its range it needs to have six degrees of freedom, often abbreviated to *6 dof*. Except for a few disparate usages, nearly all industrial robot arms are equipped with six joints to move in all six degrees of freedom. The maximum range of a robot arm is defined as an orb around the center pivot with some variations in reachability caused by the robot's geometry and its joint freedom. In factories a robot arm is often one part of a large assembly line where parts are moved

	LVL 1: Simple Placing	LVL 2: Complex Systems	LVL 3: Smart Instructions	LVL 4: Diversified Agents
Movement	stationary & linear	XYZ Gantry	free	system integrated
Sensing	sensorless	force limiting	computer vision	swarm communication
Solving	interpolation	collision free	task planning	multi robot
Operation	pick & place	system building	symbolic instructions	heterogeneous

Figure 5.2: This review of the state of the art in robotics will follow four levels and topics, own work

by a conveyor belt. The range of a stationary robot can be sufficient in this setting. When handling large assemblies, for example in the automotive or aerospace industry, a stationary robot's range can be insufficient. A linear track can be installed to add motion on a specific linear axis to the robot system. An example of a linear track in literature can be found in Bonwetsch et al. [27]. Due to the high cost of an expanded motion system a lot of research is still done using stationary robots, as evident in Belousov et al. [28].

Sensing: sensorless

All robot arms contain basic sensors, called *absolute encoders*, that measure the exact rotation of a joint in order to analyse the accuracy of the repetitive motion system. The data is used in a continuous feedback loop between defined and measured angles to ensure that the robot is always in its intended location [29]. In a closely controlled factory setting no other specific sensors are required: a robot can repeat its program indefinitely as long as all outside conditions remain unaltered. Within a walled safety cell a robot has no need for knowledge of its surroundings but once a robot is located outside its cell or the factory it was programmed for, conditions inherently change and a significant technological challenge in robotic sensing is proposed.

Solving: interpolation

A robot arm with six degrees of freedom can reach any orientation and position within its range. A combination of position and orientation is called a *plane*, defined by an origin point (x,y,z) and 3 rotations (rX, rY, rZ) within Cartesian coordinates. The six joints of the robot can vary their angles, combined this positioning is called a *joint position*, as shown in figure 5.4. The mathematical conversion from joint positions to a Cartesian plane is called *forwards kinematics* while in opposite order such a conversion is called *inverse kinematics* (IK) [30]. For each unique plane a 6 dof robot arm will compute a finite amount of IK solutions, from which the best option has to be chosen by the motion planning software. A full robot program consists of a list of planes and for each multiple IK solutions are calculated. The simplest planning software will consecutively optimise the solution to generate a robot program with the least amount of movement. Movement between consecutive joint positions is done by interpolating between the required joint angles between position A and B. This system is not aware of its surroundings and barely of itself. Potential collisions between the robot and its environment or between the robot's joints need to be mapped by simulating the entire program. Discovered issues often have to be fixed by the programmer before launch of the robot program.

Operation: pick & place

The first task robot arms were specialised for was that of *pick & place* [25], initially for dirty and dangerous environments and later for production lines as automation for repetitive and monotonous tasks. Pick & place of rigid objects has seen so much integration in industry that it can be considered as the first fully mature operation in robotics. The lack of additional motion systems, sensors or complex solving algorithms are not of concern in this operation. The robot would be programmed step by step and all arising issues would be solved by the programmer, the time this takes is not relevant as the task would be repeated thousands of times.

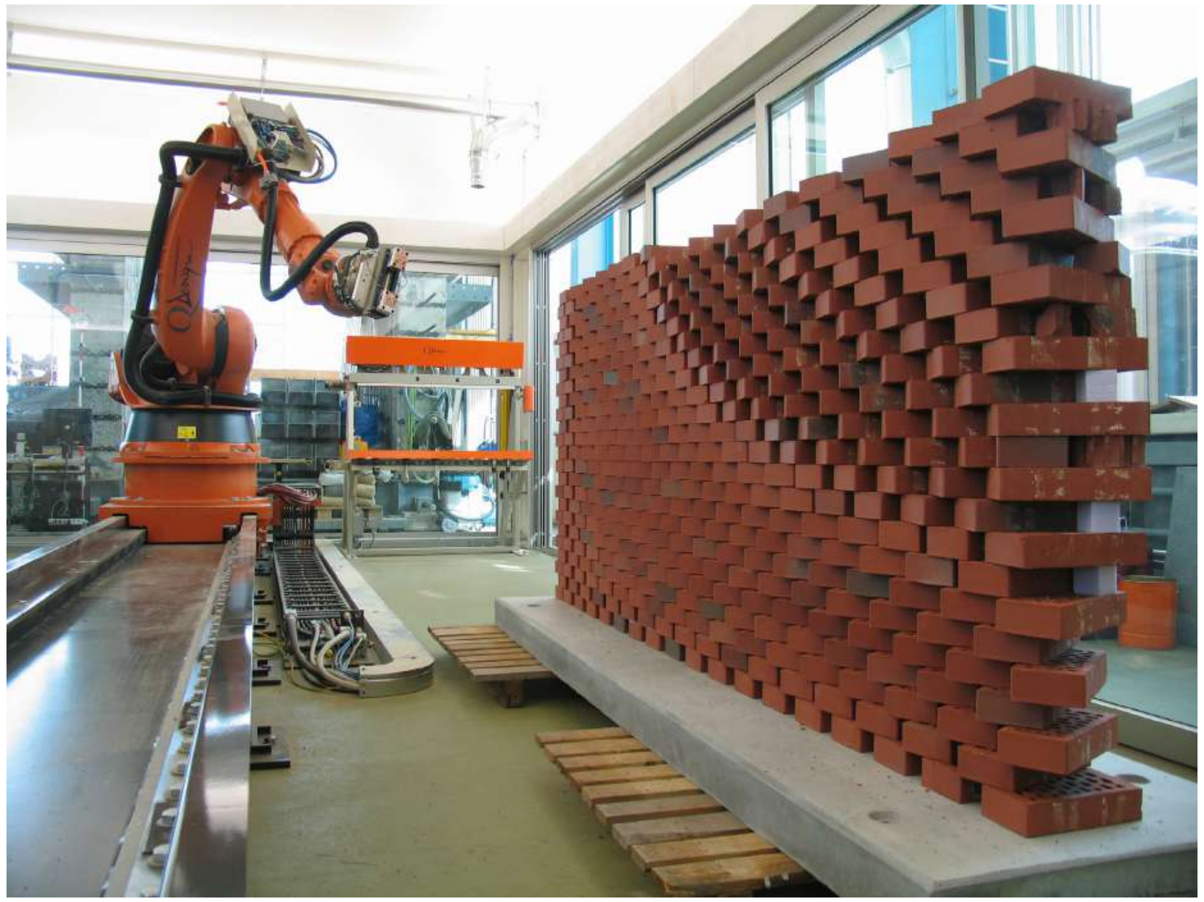


Figure 5.3: A robot arm on a linear track to extend its reach, by Bonwetsch et al. [27]

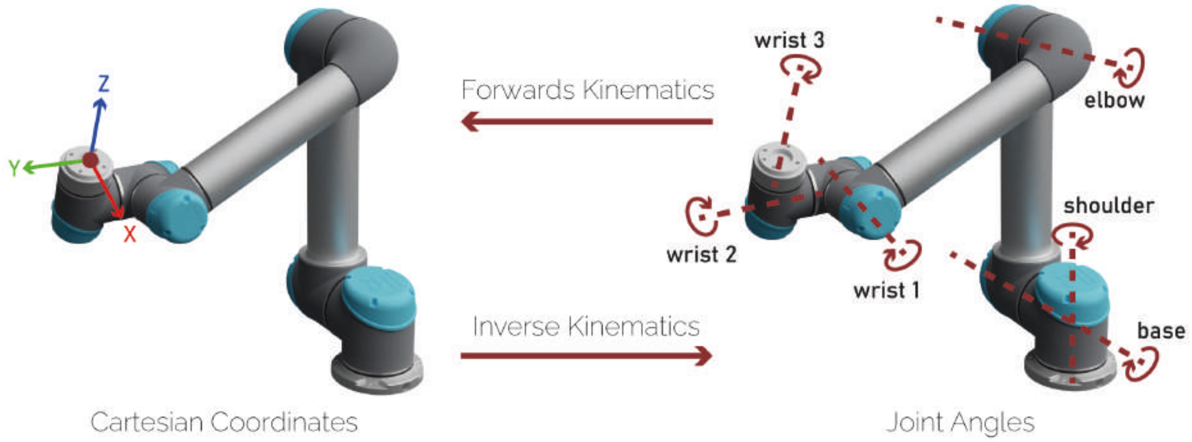


Figure 5.4: Visualisation of robot arm anatomy and kinematics

With pick & place operations quickly reaching maturity in factory settings, they were a logical first implementation in robotic automation of the construction sector in which there is one pick & place operation more prevalent than any other: brick stacking. Therefore it is no surprise that the first broadly researched topic in robotic construction was that of brick stacking, motivated by both automation and artistic expression. Combining robotic construction with parametric design allows for a computationally informed artistic exterior, introducing a real symbiotic relationship between robotic capabilities and human creativity [27]. The operation of pick & place is quite matured in both artistic expression and in practical implementation. Artistically with many different proven implementations of the same principle as shown by Bonwetsch et al. [27], Dörfler et al. [31] and El-Mahdy and Alaa [32] and practically with industrial grade implementations that are currently seeing the first limited on site use in the form of the Hadrian X [33] and SAM100 Robotic systems [34].

5.2.2. Level 2: Complex Systems

Movement: XYZ Gantry

Enabling a robotic system to reach anywhere when working within a large envelope is a challenge that has seen an increasing amount of attention both in industry and in construction. One way to guarantee reachability within a large envelope is to enable movement of the robot system in all axes using a large gantry system which is a motion system designed for multi-axis operation with an overhead bridge. In industrial applications large gantry systems have seen many different uses, not necessarily with a robot arm attached. Smaller scale gantry systems have also recently seen extensive use in 3D printers, allowing the printing nozzle to reach anywhere with precision. The printing of concrete has steadily been gaining ground in automatic construction and large gantry systems have already seen use at several construction sites in order to facilitate large scale 3D printing [35, 36]. Gantry systems have also been used for automatic tying of rebar [37]. These 3 degree of freedom gantries, moving in X,Y and Z, can only reach planes that are aligned with its working plane. Adding a robot arm to the end of a gantry will allow for reachability of every point in the envelope at any orientation. In industry this can be used in large scale welding operations of complex geometries, for example in the aerospace industry when working on large hulls [38]. If used on site, a system like this could build an entire house within its envelope. However, it should be considered that a gantry needs to be larger than its provided envelope which limits building in confined spaces and requires a time investment in setup and removal of the motion system.

Figure 5.5: An example of a gantry system for large scale concrete printing, via TU Eindhoven

Sensing: force limiting

Large industrial robots that have to lift heavy objects or perform other repetitive and dangerous tasks are often not equipped with any sensors beyond those needed for positioning. The robot arms are carefully programmed, tested and verified before starting their repetitive task in their controlled environment. When working with large industrial robots it is paramount to create a safe working environment because a robot without dedicated sensors cannot detect anything going amiss. For safety, the robots are placed within locked cells and are wired to automatically and fully disable if the cell ever becomes unlocked. In normal operation the robots work completely individually without human intervention. As the costs to set up robot cells are steep and not every task requires large industrial robots, smaller robots were developed to cooperate closer with human operators. Using internal force sensors, the cooperative robot arms can detect collisions with the environment, itself or the operator and stop its operation without causing any damage. With the launch of the Kuka LBR-3 in 2004 and the Universal Robots UR5 in 2008 the revolution of cooperative robotics or *cobots* began [39]. Allowing human operators to work closely together with the robot arm creates an effective collaboration between the robots precision and the human intuition. Force sensors can also be applied outside of the robot arm to inform construction processes by measuring local stresses in material system as shown by Melenbrink et al. [40] and Belousov et al. [28]

Figure 5.6: Examples of Human robot collaboration in manufacturing, via KUKA Robotics

Solving: collision-free

Motion planning can be defined as a computational problem of finding a sequence of valid configurations that moves an object from A to B. Motion planning is used in both computational animation and in robotics and has been a growing research area since the nineties. Collision-free motion planning is defined as programming a robot so that it does not collide with its environment. Because of its computational complexity, especially in higher configuration spaces with multiple degrees of freedom, such a calculation becomes unfeasible and thus calls for intuitive solutions. The performance of a motion planning algorithm is measured both in its computational complexity and in its completeness. A complete motion planning algorithm should always return a solution as long as one exists and it will correctly report no solution if one does not exist. An early example of a complete algorithm with good complexity is the *Probabilistic Roadmap* (PRM) by Kavraki et al. [41] which first analyses the planning scene and then randomly establishes a graph of possible connections so that solutions can quickly be derived with the aid of shortest path calculations from graph theory. Another example of a similar system is the *Rapidly-exploring Random Tree* (RRT) by Amato and Wu [42] which, instead of creating a graph based on randomly generated points, builds a graph from an initial starting point. A superior version is *RRT-Connect* by Kuffner and La Valle [43] where the graph is grown from both the start and goal. These algorithms, that first analyse or sample the scene, are called *sampling-based motion planners*.

While both PRM and RRT are efficient in finding feasible collision-free motion paths between two points in higher configuration spaces, they cannot provide a motion plan when constraint handling, energy minimization and smooth paths have to be taken into account. Although sampling based algorithms could be expanded with more stages of calculation to enable them to use shortcuts in order to refine the paths [44], a completly new approach was deemed preferable by Ratliff et al. [45] in their Covariant Hamiltonian Optimization for Motion Planning (CHOMP). CHOMP can quickly find a local optimum for an initial naive guess by using a gradient cost function in covariant gradient descent. The algorithm is fully focused on trajectory optimisation and can find smooth paths while taking joint limit constraints into account. A prevalent problem with any gradient descent based approach is the possibility of the algorithm getting stuck in one local optimum that could be significantly worse than another local optimum. Building on the foundation established by CHOMP, Stochastic Trajectory Optimization for Motion Planning (STOMP) by Kalakrishnan et al. [46] uses a stochastic or randomised approach to trajectory optimisation. By exploring randomly generated noisy trajectories optimisation of the cost function is no longer limited to only gradient functions, which is a significant improvement over CHOMP. Work has been done on further optimisation of these algorithms [47], but to the best of the authors knowledge no new large leap has been made since STOMP.

Collision-free motion planning is a very complex topic. Understanding, integrating and let alone innovating in this area requires extensive knowledge only gathered over many years.

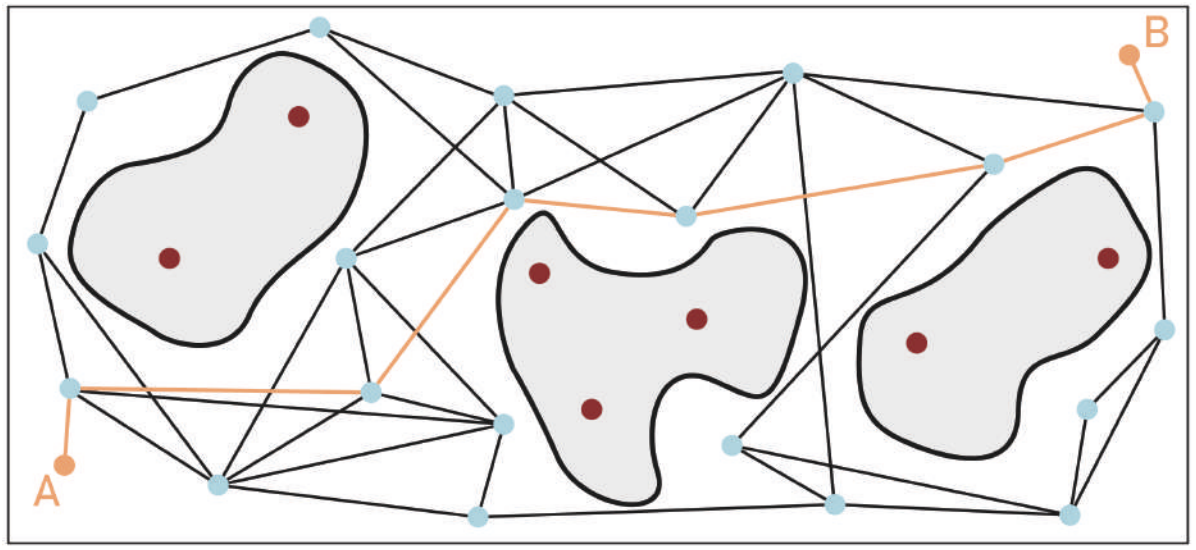


Figure 5.7: Visualisation of the PRM algorithm, a collision free path is found from A to B, own work

Since robotic integration in the construction sector is an area of research that is currently mainly motivated from an architectural standpoint, integration of these complex optimisation systems has not been widely applied in robotic construction research [48]. The potential these algorithms bring to robotic construction are however great. As a sub-category of robotics where controlled repeatability is often not possible, environments are complex, carried parts are large and movements are scaled up, automatic optimisation of trajectories can reduce wasted programming and execution time, increasing economic feasibility. Some examples of implementation in robotic construction do exist, Huang et al. [48] uses RRT-Connect, Shu, Li, and Gao [49] integrate an altered version of RRT* and Zhu et al. [50] developed a novel method based on a collision free workspace. When building a structure of many parts that are not necessarily repeatable a robotic construction system needs to prove that it can build something faster and more efficient than its human counterpart, consequently automatic generation of optimal trajectories will play a deciding role.

Collision-free motion planning is complex and in order to understand, integrate and innovate in this area extensive knowledge is required. Since robotic integration in the construction sector is an area of research that is currently mainly motivated by an architectural standpoint, the integration of complex optimisation systems has not been widely studied in robotic construction research [48], yet implementation of these algorithms will undoubtedly benefit robotic construction. In the sub-category of robotics where controlled repetitiveness is often not possible, where environments are complex, carried parts are large and movements are scaled up, automatic optimisation of trajectories can reduce programming and execution time and as such increase economic feasibility. Examples of implementation of collision free motion planning in robotic construction do exist, e.g. Huang et al. [48] uses RRT-Connect, Shu, Li, and Gao [49] integrated an altered version of RRT* and Zhu et al. [50] developed a novel method based on a collision free workspace. When building a structure out of many parts that have no direct need for repetitive motion, proof will have to be provided that a robotic construction system will be able to build something faster and more efficient than its human counterpart and consequently, automatic generation of optimal trajectories will become a deciding factor to this proof.

Operation: system building

Since robotics started with simple top-down pick and place operations the development in the field has been pushed by the ever expanding desire to automate and optimise more complex operations. One such ambition is to enable the automatic construction of structures consisting of different parts. As each part might have a unique shape, location or function a

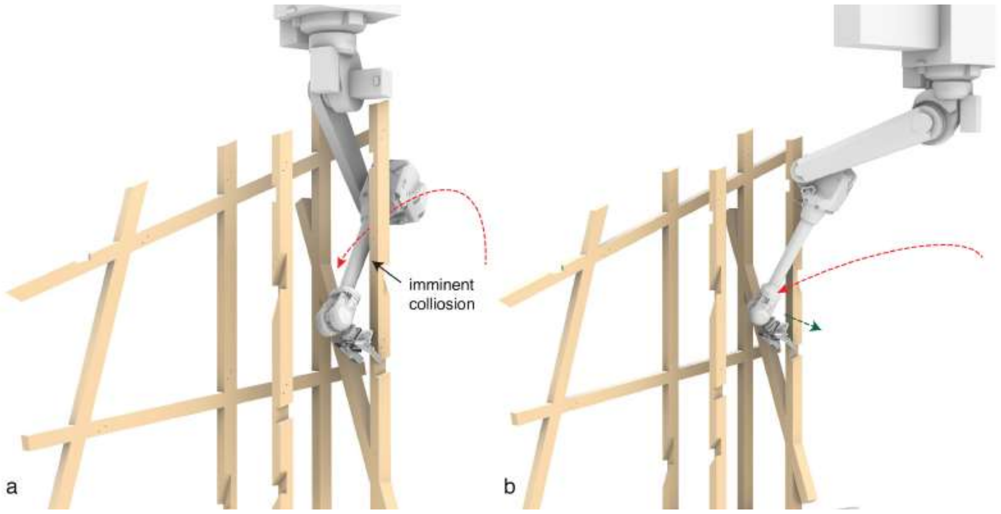


Figure 5.8: Example of collision avoidance implemented in Robotic Construction, by Huang et al. [48]

robotic system will need to be able to adapt to a range of different requirements and this encourages a form of independence between the robot system and the human programmer. Robotic autonomy can be obtained by the automatic generation of robotic programs [51] through the modelling of a complex structure in parametric design software. Depending on the complexity of a structure and its shape intricate movements can be required and within a standard interpolation based solver these could result in collisions in need of a manual fix. Collisions can be avoided in the programming phase by optimising robot poses and by imposing a building order based on the inherent logic of the system, for example by using theories of rigidity as shown by Bruun, Adriaenssens, and Parascho [52] for the automatic deconstruction of a truss system. Automatic generation of programs can be combined with motion planning algorithms for collision-free programs [48].

A current trend in the construction sector that shows potential for increased automation is the prefabrication of building elements in a factory setting. Prefab construction has the potential to reduce construction costs up to 20%, mainly by reducing the on-site time by 75% [53]. Prefab is closely related to the development of modular, transportable building elements that can quickly be coupled on-site and working with prefab can therefore be considered a good example of a complex system of parts for robotic construction. In relation to modular prefab construction robotic assembly has a dual potential: first, the trend of moving production off-site and into a controlled factory environment has short term potential for robotic automation as the technology has already been proven in a factory setting, and, second, the reduction of on-site steps to a relatively simple construction of modular elements will significantly reduce complexity of robotic operations done outside a controlled environment. Once modular systems will be developed with robotic construction in mind, it will not be long before robots become an integral part of construction sites.

5.2.3. Level 3: Smart Instructions

Movement: free

Since the first robot-like machines were considered for building sites, a desire for easy usability and thus free movement of the equipment became apparent. As mentioned in level 1, the innovation initially started with brick stacking. The earliest examples of brick stacking robotics are *ROCCO* [54] and *BRONCO* [55], both robot arms on mobile bases that needed an operator to move them into correct positions. ROCCO and BRONCO lacked sensing capabilities and were thus unable to determine their location within the site or to adapt to tolerances of building materials. With the development of *DimRob* in 2012 by Helm et al. [56] scanning sensors were added to enable it to adapt to local tolerances. The machine had to be moved manually and set up with outrigger jacks for it was not yet able to accurately locate itself within a working environment. The most

Figure 5.9: A complex system created using parametrically informed robotic construction, by Xian, Hoban, and Peters [51]

advanced mobile robotic platform currently in use is *Industrial Fabricator* (IF), developed in 2016 by Dörfler et al. [31]. This tracked robot arm has the ability to localise itself within the working environment and uses sensors to adapt to tolerances in materials and for positioning, it can move autonomously and does not require additional setup when moved.

Figure 5.10: The In situ Fabricator (IF), via Gramazio Kohler Research

Sensing: computer vision

Connecting a robot system to its surroundings through perception has been a research topic since the 1970s with camera guided robots that were able to traverse a course at a speed of 1 meter every 15 minutes [24]. The research is closely tied to machine learning and 3D computer vision which has made great leaps since the 70s. The recognition of unique fiducials is a robust computer vision system to recognise and register components as shown by Feng et al. [57] and Chai et al. [58]. Although fiducials could even be made invisible to the human eye [59], it is inconvenient to apply these to all building components, a more robust system is desired that approaches the problem through human-like recognition of the environment. This is especially important when a robotic system is taken outside of a controlled factory environment and allowed to freely move around the human environment.

Connecting robotics to their surroundings by making use of perception has been a research topic since the 1970s. Early research developed camera guided robots able to traverse a course at a speed of one meter every fifteen minutes [24]. Research on perception is closely tied to machine learning and 3D computer vision, both of which have made significant leaps since the 1970s. A robust computer vision system is based on the recognition of *fiducials*, examples are shown in figure 5.11. These unique markers can be used to recognise and register building components as shown by Feng et al. [57] and Chai et al. [58]. Fiducials can be made invisible to the human eye [59], but because it would still be inconvenient having to apply these to all building components, a more robust system is desired that approaches the problem through a human-like recognition of the environment. When a robotic system is taken outside of a controlled factory environment and allowed to freely move around the human environment, human-like perception and recognition become indispensable.

Figure 5.11: Example of fiducials used for orientation and object recognition, own work

Making robots function in an uncontrolled and changing environment is a challenge that has recently seen much attention in self-driving car research. These use cameras, lidar and radar to create a 3D model of the environment and safely navigate through it. For a computer system to understand what it perceives in a human way advanced machine learning techniques are used to segment and classify the 3D model [60]. These lessons can be translated to robotic locomotion in a construction site. 3D scanning technologies in combination with robotic movement is already being implemented for automatic building site inspection using quadruped, driving and flying robots [61, 62, 63]. However these systems have limited capabilities for automatic movement and need to be set up with trained paths. The models generated by these inspections are compared to the BIM model of the building to find discrepancies. The same tech could be used for a robot system to localise itself on the construction site. When motion planning in a BIM model is combined with localisation through model comparison automatic locomotion does not seem far off [64, 65]. By integrating compared 3D models directly in the robotic control loop these systems can adjust to imperfection of the real world [31, 56].

Enabling robots to function in an uncontrolled and evolving environment is a field of interest that bears similarities to recent research on self-driving cars. By creating a 3D model of its environment with the aid of cameras, lidar and radar, such a car can autonomously navigate. For a computer system to interpret what it perceives like a human would, advanced machine learning techniques are used to segment and classify the 3D model and camera images [60] and these lessons could be translated to robotic locomotion on a construction site. 3D scanning technologies in combination with robotic movement is already being implemented in automatic building site inspections for which quadruped, driving and flying robots are deployed [61, 62, 63]. These inspection systems have limited capabilities for automatic movement as they need to be set up with trained paths. The 3D models generated by inspections get compared to the BIM model of the building to find discrepancies and this same technology could be used for a robot system to localise itself within a construction site. Automatic movement does not seem far off when motion planning and localisation can are enabled by comparing a BIM model with actual sensor data [64, 65]. By integrating the ability to compare 3D models directly in the robotic control loop, it should be able to adjust to the imperfections of the real world [31, 56].

Solving: task planning

Approaching robotic motion problems by describing and combining specific groups of robotic actions is called *task planning* [66]. For a simple pick and place operation there is a certain intuitive logic to the order of tasks required: move to A, pick up object, move to B, place object. Tasks can easily become more complex for example when a robot is desired to place a number of objects provided at different locations with a gripper and subsequently attach several types of objects with an automatic screwdriver, whilst the robot can only change tools at a specific location. To add even more complexity both energy use and time consumption can be optimised, so deciding intuitively what order of actions should be taken in a task becomes infeasible. Task planning is relevant because it distinguishes between tasks that for example describe a movement to a specified location, a gripping action to pick or place an object, a screwing action and a tool change.

To find the best order of operations for any robotic task all of its separate actions need to be associated with their specific costs. A cost func-

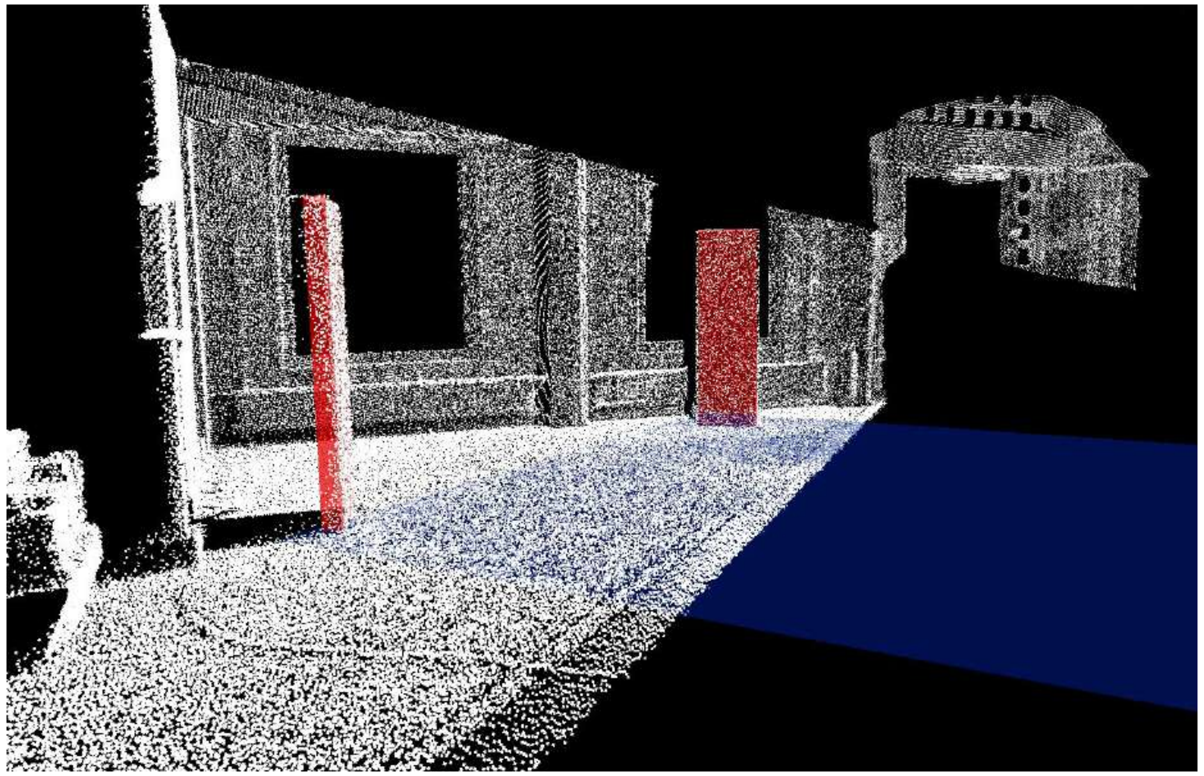


Figure 5.12: Recognition of CAD elements using lidar point cloud scanning, by Dörfler et al. [31]

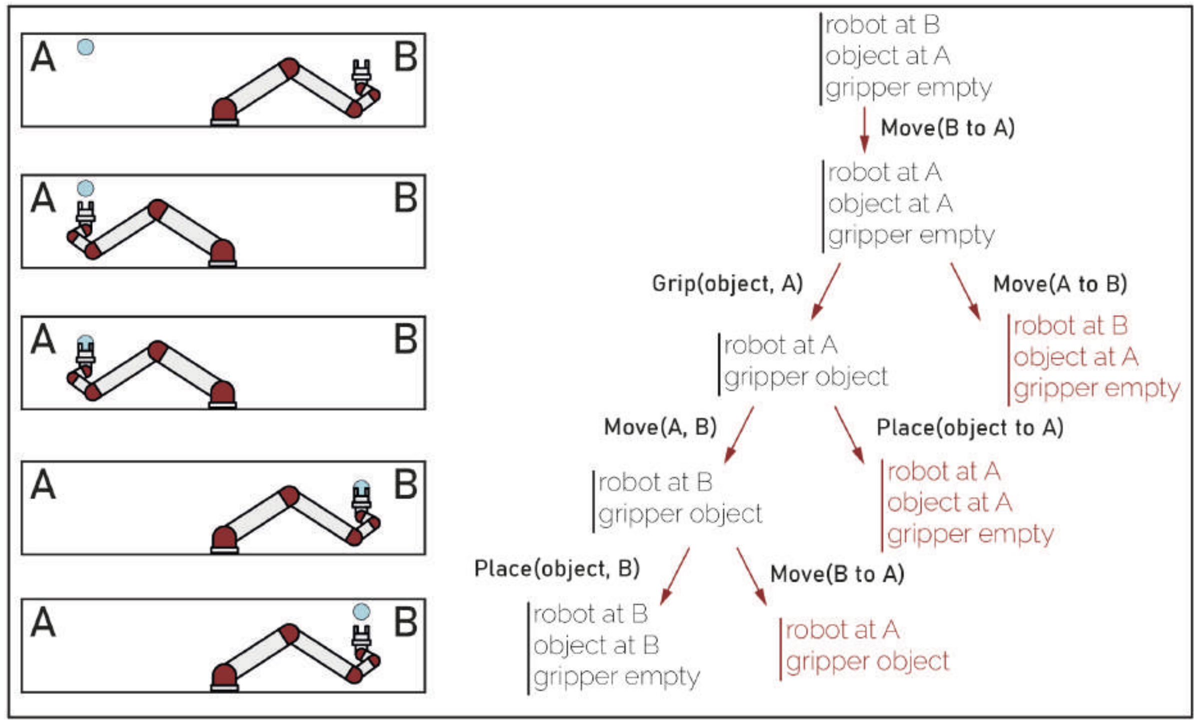


Figure 5.13: In task planning a tree is made with all potential next options through a high level environment. Paths that result in duplicates are ignored leading to task success in this simple pick & place scenario, own work.

tion can be established as a combination of variables such as time and energy use. For robotic systems these values have to be calculated through computationally complex collision-free motion planning algorithms. Tasks have a tree-like data structure in which all actions are defined with their associated cost. With a myriad of options these trees can become prohibitively large and finding the lowest cost solution can be of NP-complete computational complexity. Smart integration of optimisation heuristics is the only way to find near-optimal solutions while maintaining computational feasibility [67]. Task planning should be broad and modular to allow for a host of integrations that all require different cost calculations [68].

In task planning for robotic applications, motion planning is the the most complex algorithm required. Therefore, the combination of broad task planning frameworks with motion planning algorithms is a logical next step, this is aptly named *Task and Motion planning* [69]. By calculating the costs of many motion planning options throughout the task tree, this framework has proven to reliably find near-optimal solutions where non-task based direct motion planners fail [70]. These frameworks are being integrated in the most state of the art robotic software [69], but have not seen much integration in more mainstream software [66]. For robotic construction with many different tasks to complete and complex motion planning to optimise, a task and motion planning framework is the perfect tool.

Operation: symbolic instructions

In the previous section the principle behind task planning was introduced. These systems can be very strong in solving higher level tasks in a well defined domain. For example, a user can ask the robotic system to retrieve an object. Through task optimisation the robot system can interpret this and decide what its order of operations should be. For this to work well a robust domain has to be manually filled with well defined actions [68]. Even in a simple item retrieval the domain needs to contain many actions of motion planning, item recognition and tracking, picking and placing, quality control and feedback. While already many times more efficient than planning an entire program by hand, these systems still need a programmer to define every individually required task. Imagine the complexity of this operation when a user instead asks a robot to build a house. Just defining the possible actions may become humanly infeasible.

Task planning systems are very capable at solving higher level tasks as long as they have access to a well defined domain. For example, a user can ask a robotic system to retrieve an object. Through task optimisation a robot system can interpret its assignment and decide what the order of operations should be. For such an optimisation to work properly a robust domain has to be manually filled with accurately defined possible actions [68]. Even for a simple item retrieval action such a domain has to contain many possible motions, item recognition and tracking, picking and placing, quality control and feedback. Although already much more efficient than planning robotic actions by hand, task planning systems still do require a programmer to define all aspects of every potential, individual task. Imagine that instead of a 'simple' retrieving action a user asks a robot to build a house; determining and defining all possible associated actions for such an elaborate task may simply turn out to be unfeasible for a human.

Asking a robot system to retrieve an item or build a house are examples of *symbolic*, human-readable, instructions, strongly abstracted from the robotic programming running in the background. Combining task planning algorithms with artificial intelligence methods aims to create intelligent agents that can adapt to symbolic planning instructions using high-level learning [71]. *Reinforcement Learning* is used for agent task optimisation through iteratively rewarding or punishing an agent based on certain reward functions. While less manual work, a reward function does not guarantee a complete result and as such tweaking the reward functions can take much computation time, after successful learning the agent becomes a specialist and is no longer able to generalise for different scenarios [72]. Reinforcement learning has been successfully implemented to add tactile understanding to robotic construction, as shown by Belousov et al. [28].

Hierarchical Reinforcement Learning allows an agent to store and remember actions and

quickly reapply them to other tasks, this should allow for a greater generalisation of task descriptions although computation at this moment is still more inefficient than computation to plan an near-optimal task [71]. Contemporary research focuses on finding a symbiotic relationship between task planning and reinforcement learning so that reward functions and task environments can be generated more automatically [72]. Combining this symbiosis with artificial intelligence capable of remembering, of having knowledge and the ability to make assumptions [73, 74], enhances the possibility of a future where a robot can master how to build a house.

5.2.4. Level 4: Diversified Agents

Movement: system integrated

Where combining robotic systems with tracked bases allow them to freely move in any horizontal direction, it does not allow for vertical motion which, at the building scale, is a significant limitation. Gantry systems do allow for vertical movement, but have their own drawbacks in setup and envelope size, as argued above. Attempting to solve the problem using contemporary construction equipment one could imagine a temporary elevator or crane being used for vertical motion between floors. Although this could be a suitable solution for a standard building, it does assume that the robot system can reach anywhere within the volume of the floor. The capabilities of such a motion system will be limited for buildings with large floor heights or irregular shapes. For example, space frames are often constructed at significant height and in irregular shapes, requiring workers to climb around the system, a difficult and dangerous task that could be automated.

A substantial share of recent research on and development of construction robots employs a union of construction elements and robotic motion, where the robot's motion is supported by construction elements. Melenbrink et al. [75] offers a robotic system capable of navigating a 2D truss and that could also potentially bear an extra arm to transport components, with which it can expand the truss it uses for movement. Combining a motion system with the intrinsic qualities of the material of the construction system can create a symbiotic relation, as shown by Lochnicki et al. [76] with a robot that uses the flexibility of the material to swing forward. Delikanlı and Gül [77] shows a concept for a movement system that is compatible with a 3D truss because it allows for rotation around cylindrical beams. Leder et al. [78] currently show the most advanced system in which multiple robots work together and at the same time make use of the construction system's elements for motion and assembly. Another method of motion that could be feasible is using a two sided robot, similar to the space technology used for the ISS' European Robotic Arm, that does not slide across beams but instead is able to find and slide into a socket, then leave its starting point and with the freed end can find another, next socket [79]. Similar to all these system are the small sizes and simple functions of the climbing robots. To be able to cope with more complex tasks, all the above mentioned researchers propose systems of collaboration between multiple robots.

Sensing: swarm communication

Because a robot cannot predict what a human is about to do, robots that work alongside humans need to be restricted in their movement and force to guarantee a safe working environment for their human counterparts. Contrary, in robot-robot collaboration all the information on future actions can be communicated, allowing for pre-planned smooth collaboration between multiple robots. Multi-robot setups in which machines collaborate without collisions have been proven in industrial implementations using adapted motion planning algorithms. The collaborating robots are centrally controlled and synchronised on one server, containing the low-level code for the entire repeating program [81]. A centralised system has been proven to work well in factory settings with a limited amount of robotic systems, although it can be more optimal to allow for direct robot-robot communication when more robots are considered. Disconnecting individual robots from a central system and relying on inter-robot communication is called *swarm robotics* [82].

Inspired by ants, termites and birds, the research on swarm robotics focuses on in the inter-mechanics of many-agent systems creating results larger than any individual agent's actions. While nature is inherently chaotic, it is

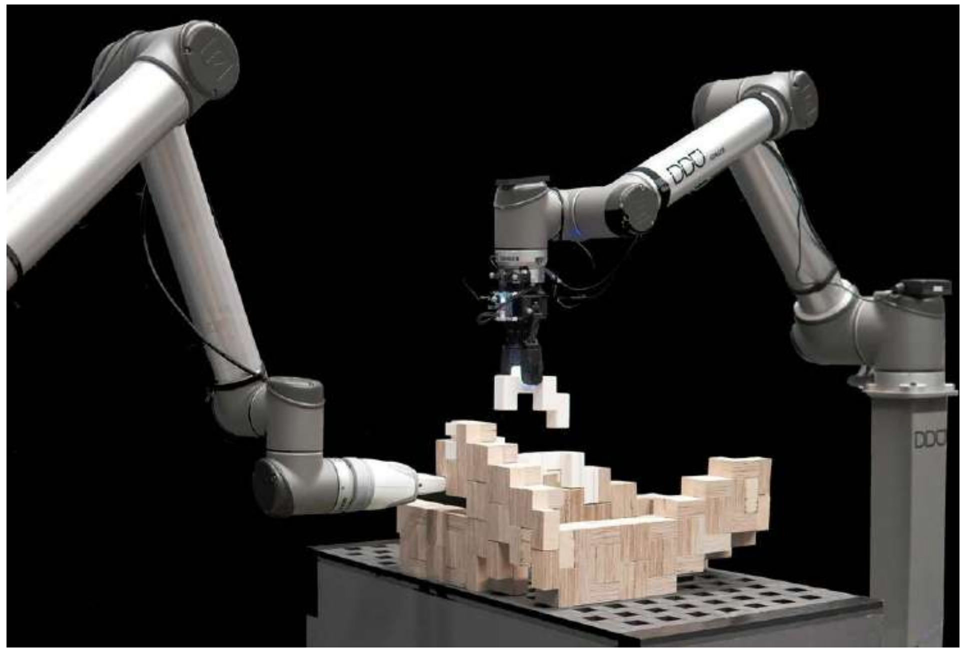


Figure 5.14: Robots assemble a block system using tactile feedback powered by Reinforcement Learning, via Wibranek Lab

often observed that systems with very simple rules create order, for example in the form of patterns, this is called *emergent behaviour* [83]. Similar to how termites build their nests without a centralised plan, swarm robotics could construct buildings based only on simple rules. Combining swarm robotics with reinforcement learning and symbolic instructions could allow the swarm to learn through their work and adapt their rules to improve the final result. In the future an emergent swarm might not only construct our houses, we might even learn from their architectural expression. An implementation of swarm robotics in construction has been researched in a 2D visualisation using simple blocks [84, 85]. Similar to how reinforcement learning needs a human input to define a cost function, balancing human intuition with low-level rules has shown to be challenging. Often some form of centralised model is still used to inform the process and motivate it into a certain outcome.

Although the idea of an emergent construction swarm is fascinating, it is not currently feasible or practical. However, specific theories behind swarm robotics could have some specific practical implementations in the not so distant future. The self-driving car sector is a main proponent of swarm robotics research because cars are not connected to a central server and they should be able to communicate all relevant information locally. All cars can then share their current and future plans to allow for smooth traffic control. For construction robots a certain balance between centralised and local communication will be optimal. It is safe to assume that a central planning server using a BIM model will decide the higher level collaboration between all robots, however robots may communicate and process locally relevant information with each other to reduce central computation. The locally shared information could be local error corrections, safety related observations or near future paths to optimise low-level and local collaboration.

Solving: multi robot

Solving a higher-level task for one robot system is already a complex problem, solving the same task for multiple collaborative robots might exponentially increase computational complexity and could quickly become prohibitive. Similar to the use of intuitive optimisation heuristics for single robot solutions, heuristics are used to enable multi-robot collision-free motion planning. This has been proven in a factory setting using a limited amount of robots [81]. For a larger number of robots more experimental optimisations have to considered, for example inspired by the movement of ants. In *Ant Colony Optimisation*

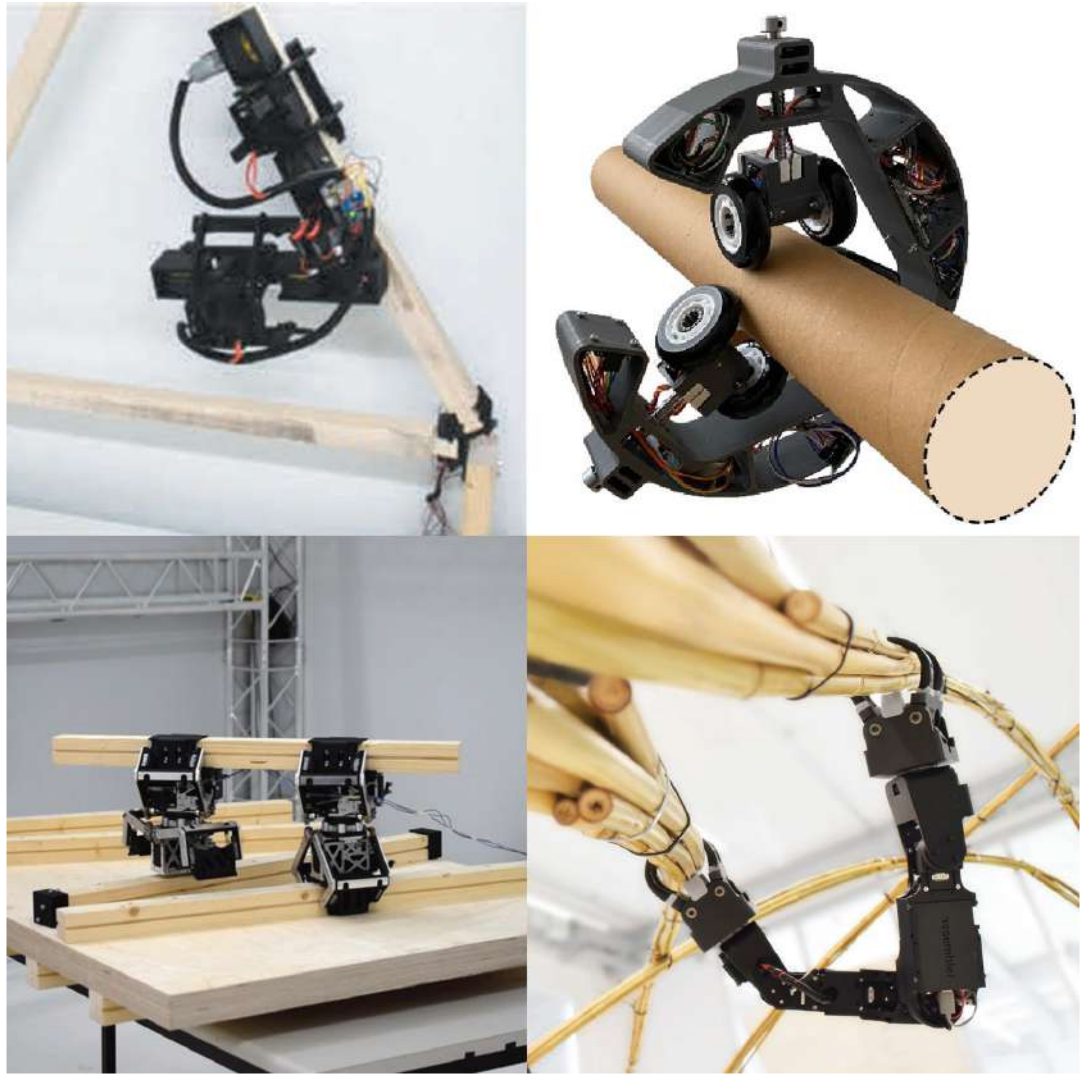


Figure 5.15: Physical prototypes of robots that use system integrated movement. top left by Melenbrink et al. [40], top right by Delikanlı and Gül [80], bottom left by Leder et al. [78], bottom right by Lochnicki et al. [76]

Figure 5.16: European Robotic Arm attached to the ISS, via NASA

viable paths are marked with a computational pheromone allowing other agents to learn from previous agents experiences [86]. In this way collaborating robot swarms can learn and adapt to their environment. Currently, such an algorithm does not yet ensure robot-robot collision free movement. Though much work is being done on solving planning and motion problems in multiple agent settings, 3D environments or the complex kinematics of a robot are often not taken into consideration [87, 88].

Although practical integration of collaborating robot swarms with complex movement systems that automatically find paths, collaborate and do not collide might still be distant, collaboration between multiple free moving robots on a tracked base has already been implemented. Standard buildings can easily be abstracted to 2D floor plans using BIM which allows for motion planning of multiple agents in a 2D space [88]. The movement of a platform on tank tracks is simple and thus requires no complex kinematics: the robots can be placed in the correct location by the motion planner, after which the collision free motion planner takes over to complete the specific task. Robots communicating locally for error correction can collaborate effectively and without collisions.

Operation: heterogeneous

An idea introduced in swarm robotics research is the concept of a heterogeneous swarm where different agents can follow different rules and so fulfil different functions. A similar concept in construction is one where tasks require many different tools, this is translated to robotic construction and in current research there are examples where a robot arm is equipped with multiple tools to switch between [58, 89]. Because a 6 dof robot arm is a generalist it does not have a specific function apart from positioning. It is the attached tool, or *end-effector*, that defines the function a robot arm can fulfil. In a factory setting a robot arm is the perfect tool for almost every implementation but in construction this might not be the case. It is unlikely that a mobile robot arm will ever replace an excavator or crane. On a fully autonomous building site existing construction equipment should be robotised [5]. Some integrations already exist, for instance the automated excavator HEAP [90]. Other technologies are being researched, Burkhardt and Sawodny [91] for example, shows that the accuracy of crane load positioning can be increased when smaller cranes are used for finer end-positioning of the load, this principle can make a real impact on collaboration between equipment with heterogeneous functions.

Task planning for heterogeneous construction robots is an extra complex issue but regrettably much of the current literature only deals with heterogeneous drone systems [92]. After further research a broad framework for heterogeneous task planning could be established that will allow all the specific technologies required for construction to be defined as simple tasks. A BIM model could automatically be deconstructed into all required tasks and by using their inter-dependencies the framework could be able to create a robust high-level plan for construction and would furthermore enable all robotic systems to collaborate through both local and centralised communication. Only after more research a fully autonomous building site throughout all phases of construction can become a reality.

Figure 5.17: An example of heterogeneous robot functions where both a gripper and a robotic screwdriver are applied, by Kunic et al. [89]

5.3. Informing Design

The previous section gave a comprehensive review of research development in robotic construction. Based on the researched literature this thesis' scope will be further delineated and the design outline will be further specified. To

further structure this thesis, a distinction will be made between practical and theoretical implementations. The theoretical implementation of the mentioned technologies will be unbounded by hardware and software limits and will only be limited by what is theoretically feasible, as concluded in level 4 of the previous section. To illustrate the potential for robotic reassembly, an ever evolving pavilion will be designed and the construction technologies required will be discussed and visualised.

Practical implementation will be limited to available hardware and software. Hardware is provided by the *LAMA Lab* at the *TU Delft's Faculty of Architecture*, here a *CB-Series UR5* and a *DH Robotics AG95 Gripper* are available. For software implementations this research will attempt to implement *ROS2 and MoveIt2* to run collision-free programs using the *Moveit Task Constructor*. A link to *Rhino and Grasshopper* is desired to synchronise geometric and positional data. Integration of ROS and Moveit is novel for the faculty and linking Grasshopper to the Moveit Task Constructor would be novel in the field of robotic construction. A part of a facade will be manufactured and robotically constructed at scale. Following the level and topic diagram this practical implementation will reach the topics as shown in figure 5.18. For all technologies that are unavailable human-robot collaboration can be used to compensate.

5.4. Conclusion

This chapter was structured around the research question: "What is the state of the art in robotic construction and how can it be used to inform the design of reusable nodes and beams in freeform building facades?". To find the state of the art of robotic construction a comprehensive literature review was done. This was structured into four levels of increasing complexity and thus decreasing technological maturity. All four levels were split into four topics, to present the state of the art in movement, sensing, solving and operation. It can be concluded that the state of the art in implemented research is a free moving robot system that builds complex construction systems enabled by task and motion planning and supported by sensors and advanced computer vision. Theoretical research showed that the future for robotic construction could be optimised collaboration between many robots with different tasks that might use the building system they construct for locomotion.

A theoretical implementation informed by the conclusions will be presented at architectural scale. The outline of robotics research will be used to clarify the hardware and software limits in the practical implementation of the research. Specific attention will be given to researching the potential implementation of collision-free task and motion planning synchronised with Rhino and Grasshopper. The answer to the sub question can be stated as follows: the state of the art in implemented robotic construction research is a free moving robot arm equipped with vision and other sensors to build complex systems enabled by task planning. An optimised collaboration between many robots with different tasks that possibly use the building they work on for their locomotion could well be the future.

	LVL 1: Simple Placing	LVL 2: Complex Systems	LVL 3: Smart Instructions	LVL 4: Diversified Agents
Movement	stationary & linear	XYZ Gantry	free	system integrated
Sensing	sensorless	force limiting	computer vision	swarm communication
Solving	interpolation	collision free	task planning	multi robot
Operation	pick & place	system building	symbolic instructions	heterogeneous

Figure 5.18: The review diagram, the topics reached by the practical implementation of this research are in blue, own work

6 Design & Assembly

The previous chapters established a clear definition of nodes and beams used in freeform building facades. Theory and testing on mesh rationalization, a statistical analysis of shape similarity and a review on the state of the art of robotic construction have further specified the research direction and established certain design requirements. With all this information introduced, this chapter will discuss the design of the node and beam system and use the developed system for a practical assembly process. This will be done under the following sub-question.

> "How can a reusable node and beam system for freeform building façades be designed and automatically assembled?"

This chapter aims to answer the sub-question through a process of product design, scripting, analysis and assembly. The process will start with a free exploration phase where ideas are loosely introduced, refined and the concepts pros and cons are defined. A design concept will be chosen, further developed into a fully parametric node design and fabricated. To extract the internal parameters for the nodes and beams from a quad mesh this thesis' computational system will be expanded. Next, robotic planning software will be explored, defined and discussed and the practical assembly sequence will be scripted to automatically construct a practical scale model. To conclude, results and insights will be discussed and future challenges proposed.

6.1. Node Design Exploration

In the early design phase ideas can be quickly devised, developed and implemented to find their limitations and potential. Being free to explore any idea, even those that don't initially seem reasonable, helps to further define the design space and can consequently inspire progressively more practical ideas. Before a final design direction for the nodes was chosen four options were explored, these are shown in figure 6.1.

The first design is based on a *compliant mechanism* where flex in the material allows limited rotation around all axes. Although a very interesting research direction, compliant mechanism were found to not be lockable, relatively large and complex in both design and manufacture and the design direction was abandoned. The second design is based on a *lockable ball-joint*, commonly found in studio or camera equipment. A ball-joint allows for movement in all axes and can be locked with a set screw. Despite being more reasonable than a compliant mechanism, the manufacture of ball-joints is still complex and expensive and although a financial analysis is outside of this thesis' scope, the chosen solution should at least be financially feasible.

The third design is much more reasonable. Based on rotation around a central axis, this design allows for rotation around the X and Y axes but will not allow for torsion around the Z axis. The design is lockable around the central shafts using shaft collars and the Y axis is locked using bolts. The limited number of unique parts, and low complexity allows for optimised manufacturing methods and thus financial feasibility. Only the shaft collars need accurate dimensions as the solid contact between shaft and collar will increase the locking strength. Using precision machining this can be achieved, but that does add another step to the production method. Another problem with this system is the significant offset of the connecting beams normal direction. The node will have a thickness of four times the beam height which is undesirable for both the construction and the visual appearance.

The fourth design is the culmination of the more defined design space established by the previous explorations. As established: a node should be lockable, easy to manufacture and connect the beams close to the plane of intersection. Similar to the previous design, this node will allow rotation around the X and Y axes and will not support torsion around the Z axis. By rotating around four axes instead of one, the beams are connected close to their plane of intersection. The low number of parts will again allow for optimised manufacturing and in this design there is no requirement for accurate bores as the hinges use use friction to lock the rotation, which does not need precision machining. Although this early design is not perfect, it does show a promising potential and was chosen for continuation in this design process.

6.2. Parametric Node Design

Problems of the chosen design should be further identified and optimised. First, the strength of the friction lock should be increased. This is defined by the area of the contacting surface, the applied force and the *friction coefficient*, an empirical material property that is related to the surface roughness of the materials. All of these parameters can increase the locking strength of the node's hinges. The contacting surface can be increased by changing the hinge geometry, the force can be increased by applying more torque to the locking bolts and the friction coefficient can be increased by choosing a high friction material like aluminium or by increasing the surface roughness. Design variation one in figure 6.2 illustrates these changes. More research is required to test if this locking mechanism will have enough strength for real world use.

Another issue is the low stability caused by the empty center of the node. This issue is partly alleviated by increasing the contacting area of the hinges and is further improved by filling the centers empty space. Next, the size of the node can be optimised by reducing the supported range of rotation which will allow the centres of rotation to be spaced closer together. In the initial design a rotational freedom of 90° is possible. Through the computational rationalization process it was found that the required angle range is often much smaller, even in high curvature shapes. The size of the node can therefore be reduced to the lowest limit that is geometrically possible, which still supports an angle range of 45°. Design variation two shows the center support and size reduction.

Lastly the question of feasible and accurate connections needs to be addressed. In the current design a very accurate press fit is required to connect the beams to the node. Additionally there is no clear end for this press fit. The required accuracy to make a connection can be reduced by adding significant chamfers for self-alignment to the end of the node. By adding an edge of the same thickness as the profile a clear end to the press fit is defined. For more security it was considered to first press fit and then bolt the connection. For the practical implementation of the system it was chosen to only rely on the press fit to reduce assembly complexity. The feasibility of the connection will be further expanded on in the next chapter. Design variation three shows the added chamfers and edges.

Design variation three of the node fixes all issues found in the exploratory design phase, and is thus chosen as the final node design for the practical assembly. The chosen manufacturing method to create the physical models was 3D printing as the technology is widely available and will allow for quick prototyping to optimise tolerances and fits. In order to quickly iterate, the node was parametrically defined in Grasshopper. This was specifically done for square beams, but could easily be adapted to rectangular beams. The node design is based on 8 parameters, as defined in table 6.1. The most important parameter is the beam profile height and for the practical model an aluminium profile of 15mm x 15mm was chosen. The chosen values of the parameters are also shown in table 6.1. Technical drawings of the node elements are provided in figure 6.3.

Profile height	Ph	15.0
Profile thickness	Pt	1.0
Tolerance	tol	0.15
Nut height	Nh	4.0
Required angle	ReqA	45°
Chamfer standard	Cst	1.0
Chamfer bolt	Cbo	1.2
Chamfer connect front	Ccf	4.8

Table 6.1: Required parameters for the variable node

Figure 6.1: Node design: early exploration, own work

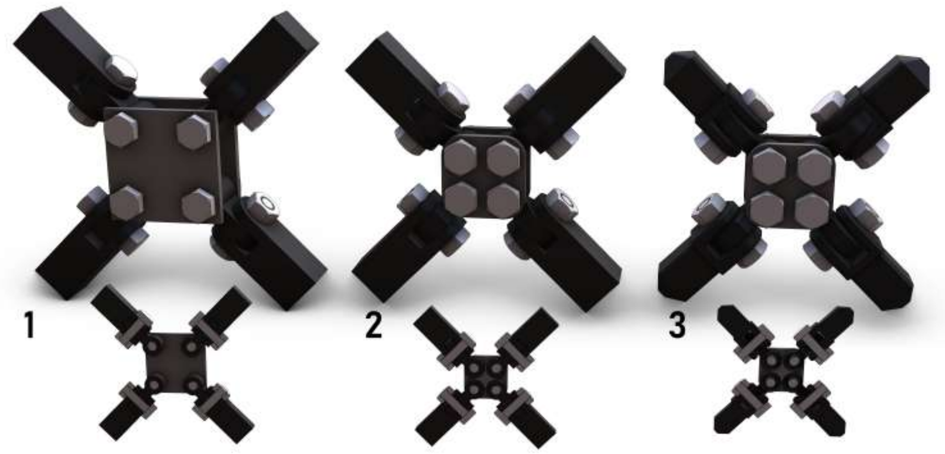


Figure 6.2: Iterative node design with sections, own work

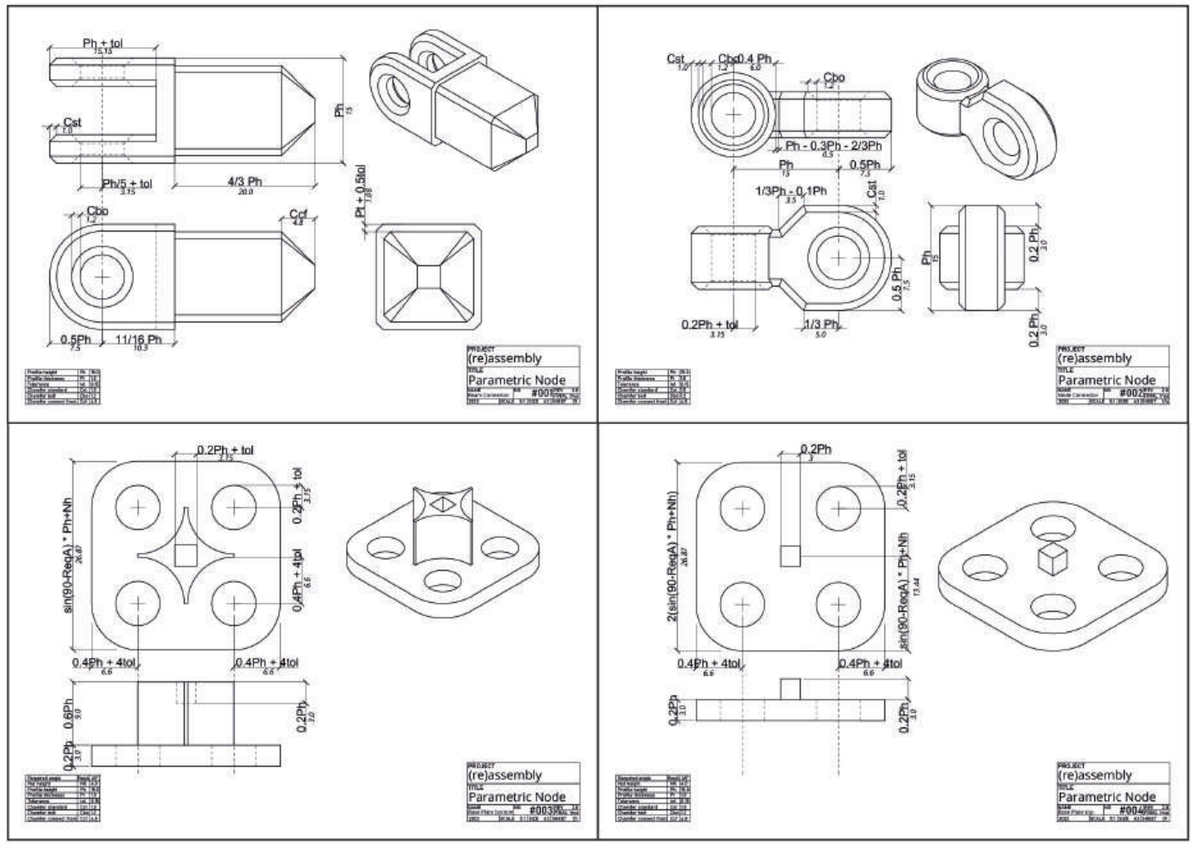


Figure 6.3: Technical drawings of Parametric Node Design, own work

6.3. Computational Placement

The internal parameters of the nodes and the beams in a freeform shape are influenced by the chosen node parameters. The beams are not connected at the exact intersection of the mesh' edges but rather connect to the node's hinges. The node parameters define the axes of the hinge's rotation and the hinges are rotated to closely follow the mesh' edge. The beams are then generated to connect two hinges on either side of an edge. In this way, the rotation angles of the node's hinges and the length of the beams can be extracted. This thesis' computational backbone in Grasshopper has been further developed to extract the internal parameters of the nodes and beams from the optimised quad geometry of chapter 3. The use of grasshopper allowed for the coding of complex geometric calculations with direct visual feedback, which immensely sped up the bug-fixing process. The computational complexity of the script has been optimised by not using heavy geometric calculations and employing C# for looping and advanced data management. The script is split into a number of individual parts, as shown in figure 6.4. The topics of Shape Generation, Shape Relaxation, Conicality Result and Parametric Node have been previously discussed in chapter 3 and the previous section. To explain how the computational placement algorithm works this section will go in depth on the topics of Frame Generation & Node Angle Calculation. The topics relating to robotic construction will be discussed in section 6.6.

Frame Generation

The input of the frame generation algorithm is the optimised quad mesh generated in chapter 3. The goal of this algorithm is to output 4 frames for every mesh vertex, or node, that point to the vertex' neighbours. A flowchart and visual aid for this explanation are shown in figure 6.5. First an approximated base frame is generated using the vertex normal (1). Four vectors to the vertex neighbours are used to generate four points around the node (2). The points are used to further refine the frame, by first moving them

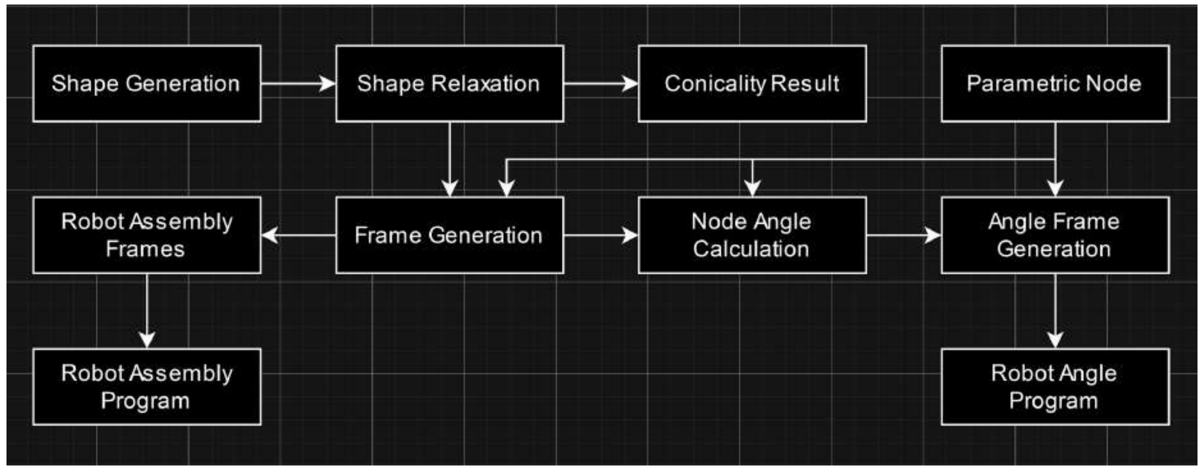


Figure 6.4: General framework of the computational placement algorithm, own work

to the original frame (3), sorting them based on rotation around the frame (4) and then using the average and a selection of two points to generate the refined frame(5). Using the node parameters this frame is then moved to the X-rotational axis and mirrored around the refined plane. This generates the required four planes. Using C# the frames are used to generate lines between the appropriate frames (6) and everything is ordered as a list of horizontal and vertical lines with their respective start and end frames to forego generation of duplicates.

Angle Calculation

The lines and planes that are output by the Frame Generation algorithm are used to calculate the internal parameters of the nodes and beams. First the standard vector of the node is found (1), shown in red in figure 6.6. The angle between this vector and the vector along the line to the next node is measured at the X rotation axis. Based on the Node Parameters the Y rotation axis plane is generated, moved and rotated around the X-axis (2). A new line is generated between the new Y planes and the angle between the standard vector and line vector is again used (3). Reflex angles in the data are transformed and the data is restructured using C# to output four X and Y angles per node. The length of the line between the Y axes can be used to find the beam length. The edge cases in this script are the naked edges of the mesh where a node does not have four neighbours. The bottom connectors of the bottom nodes have their angles calculated to align with the World XY plane. Other nodes on naked edges have the unused hinges' angles set to 0. The optimisation of the full algorithm is tested and generated internal parameters of 1250 nodes in 6.0 seconds, around 4.8ms per node.

6.4. Architectural Case Study

To further test the algorithm, an experiment on a real world case is desirable. As a torsion free result is important for the application of the nodes a real world example of a torsion free façade was preferred. According to Wallner and Pottmann [11] the curved roof of the Yas Hotel in Abu Dhabi exhibits this torsion free quality. As there are not many other sources that second this information, it is hard to verify. To test the algorithm on the Yas Hotel a digital version of the building geometry is required. Based on satellite photos, google earth, sections and available 3D mesh models the surface of the roof was made using a Sweep2, this is shown in figure 6.7. Since the Yas Hotel has 5800 panels across its entire geometry the choice was made to cut out a subsection of the surface to reduce computation time. Also the building façade uses a diamond grid, this is based on quads but has a diagonal U/V direction with triangles at the edges. For the frame generation algorithm this would require a different data structure and while it is possible to adapt the algorithm for a diamond grid, due to time considerations it was decided to use a square cutout in the diamond grid which does allow for use of the originally implemented data structure.

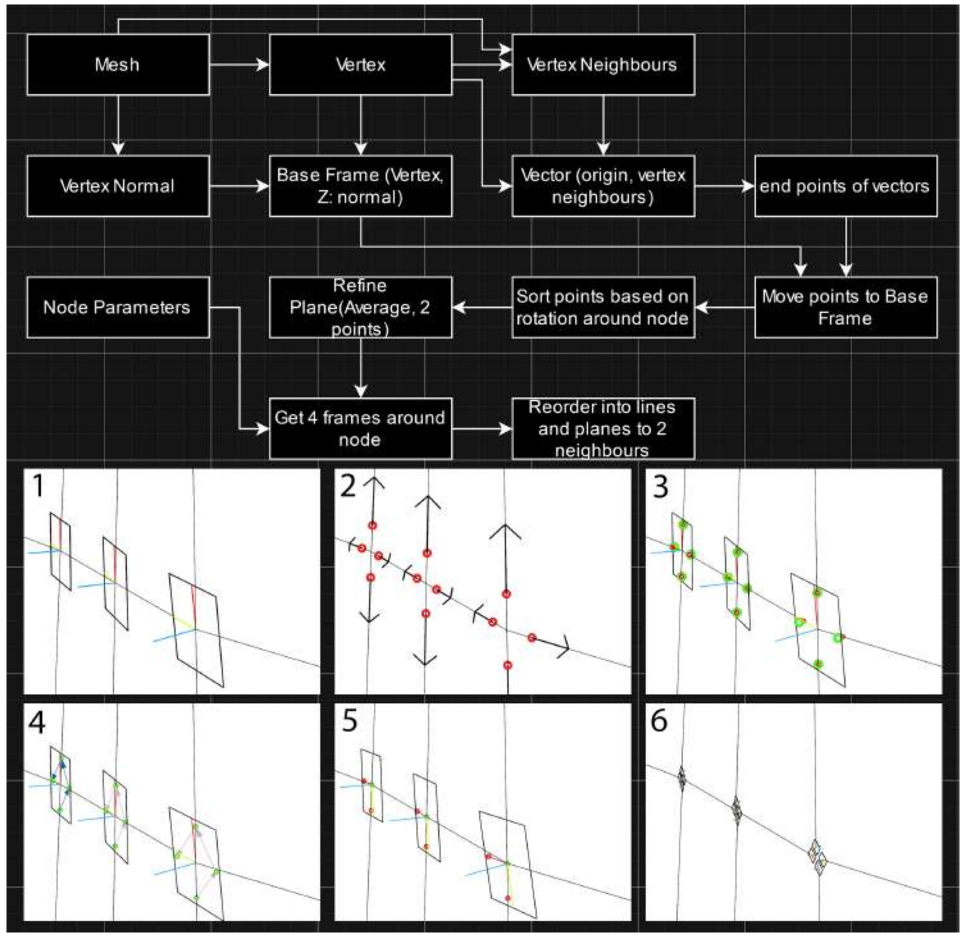


Figure 6.5: Flowchart and visual aids for frame generation, own work

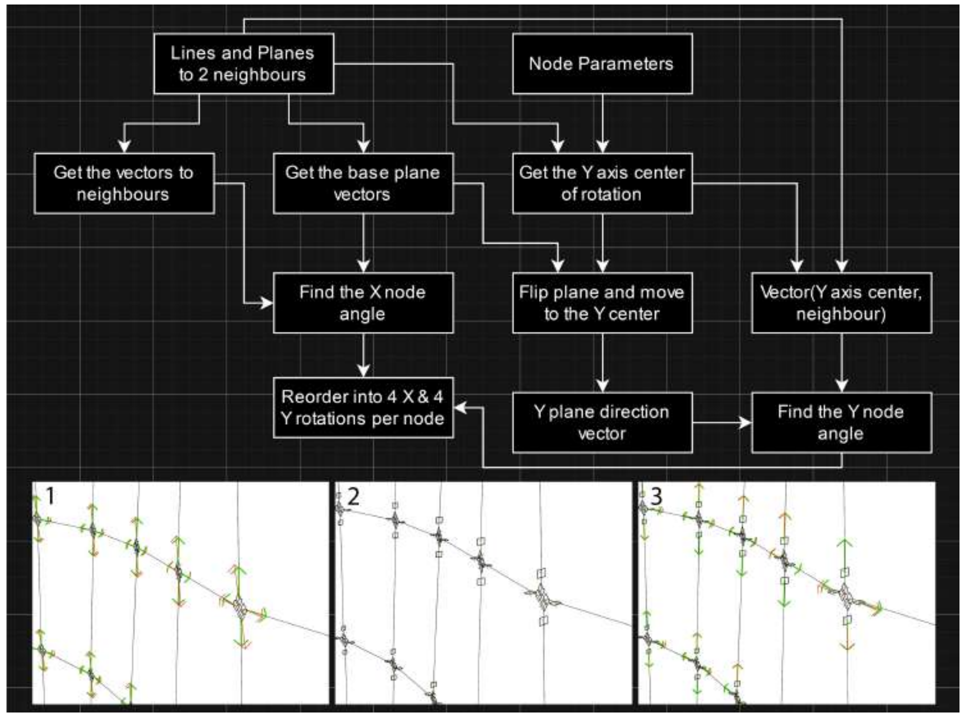


Figure 6.6: Flowchart and visual aids for angle calculation, own work

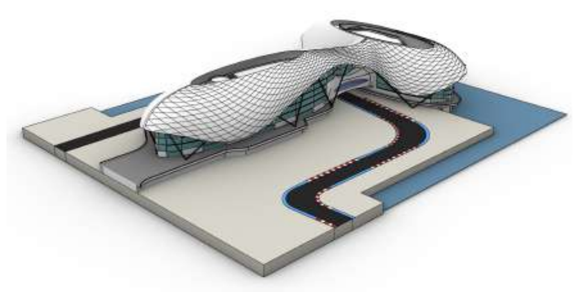


Figure 6.7: Yas Hotel as a 3d model, own work based on [93]

The first step of the algorithm is optimising the input shape to have torsion free nodes and planar quad panels. Inputting the generated geometry shows that it is not currently planar or torsion free. Based on the limited literature it would be expected that the shape would have these qualities [11]. Possible explanations could be inaccuracy of the literature or this thesis' less careful shape generation method and thus less optimised initial panelization, this cannot established with certainty. Although it is unexpected, this inaccuracy will require implementation of the Yas Hotel geometry in the shape relaxation algorithm. although the algorithm showed promising results on the arbitrary shapes chosen in chapter 3, the result on the Yas Hotel Geometry was disappointing, limited to the results shown in figure 6.8. When strength of planarization and conicalization is increased the simulation quickly becomes unstable. A possible explanation for this could be an input panelization that is not close to optimised. Next to that it is found that the conicalize goal of Kangaroo can cause instability in the simulation for specific input panelizations. It is expected that these factors relate to each other, but that cannot be stated with certainty and requires future research.

Although the mesh could not be fully optimised, it will still allow for implementation in further steps of the computational algorithm. The only inaccuracy is the unresolved torsion that will cause a visible misalignment between the beams and the nodes. The beams and the nodes' simplified geometry are generated and positioned. Within the chosen cutout a feasible result is generated using the variable nodes. A couple of visual observations are immediately clear, first due to the diamond grid the node X angles are more extreme and second the unresolved torsion can be seen at the beam-node connection, both shown in figure 6.9. To statistically support these observations the node angle data of X, Y and Torsion is shown in histograms in figure 6.10. The X angle data shows two distinct ranges of 0° to 18° and 44° to 50°. The Y angles and torsion in this shape are small, under 3.5° and 5° respectively.

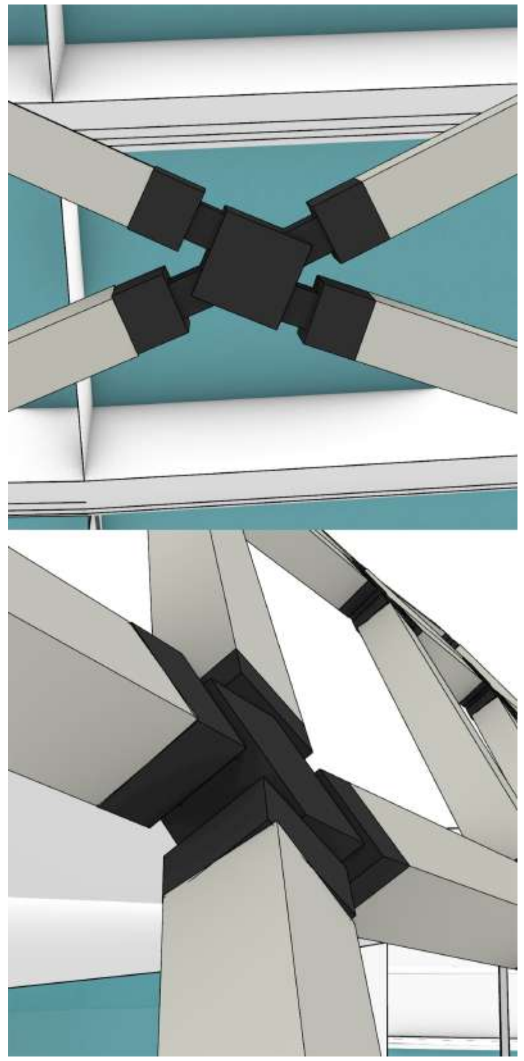


Figure 6.9: The generated node and beam geometry, top: large X angles, bottom: small torsion angles, own work

Through attempts at optimisation, visual observations and statistical proof certain aspects of the algorithm and node design become clear. Testing on the Yas Hotel shape with a high number of panels shows that relaxation of an input mesh can be computationally expensive, does not always provide a desirable result and is therefore not the ultimate solution. If the initial shape is more optimised a desirable result

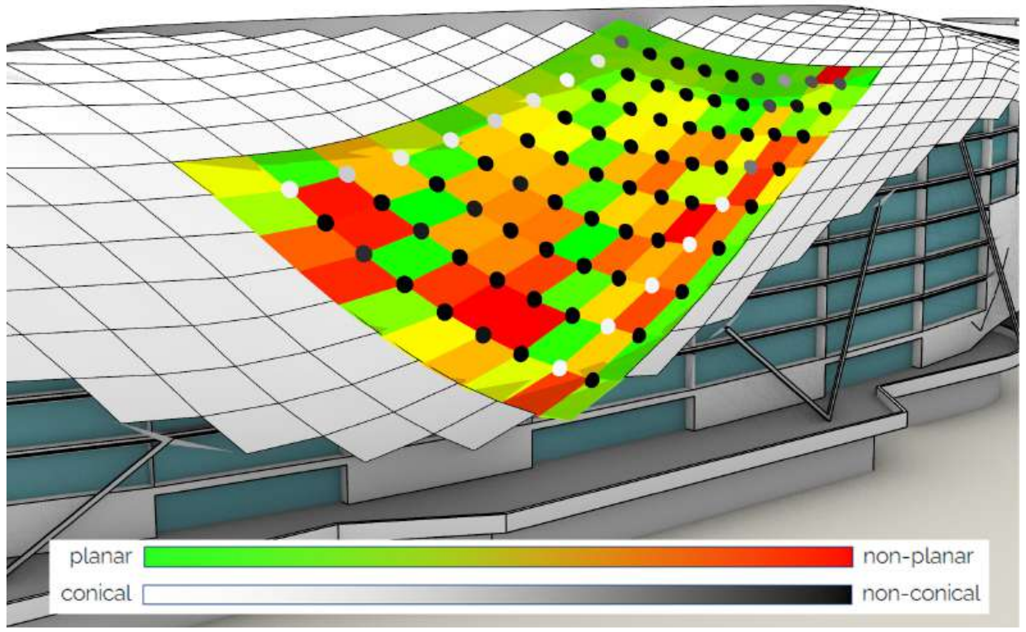


Figure 6.8: An attempt at optimisation of the Yas Hotel geometry, own work

is much more likely. More research into the relation between the designer and the many different generation and rationalization algorithms is needed. Next, the found angles in the Y-axis and torsion are very small and therefore it could be hypothesised that tolerances and flex in the system can compensate for small values. This requires further experimentation. Although, the computational placement algorithm is currently limited to a square grid, further development could allow it to work on any grid. All in all the algorithm shows promising results but should accept more diverse inputs and be developed to a more stable state.

6.5. ROS & MoveIt

As previously mentioned in section 5.3 the practical assembly process will be limited by the available hardware and aims to implement contemporary software with ROS and MoveIt. This section will motivate the choice for and describe the attempted implementation of this software. The experience, results, insights and limitations found through this process will be portrayed through the eyes of someone inexperienced not only with ROS and MoveIt, but also with Linux, C++, CMake and Terminal commands.

Through past experience with robotic construction using the Grasshopper plugin Robots [94] and the offline simulator RoboDK [95] it became clear that the most pressing issue in complex robot programs was the large time investment in manual collision solving. In order for robotic construction of non-repeating structures to be efficient the coding and development time needs to be reduced as much as possible. Automatic collision free path planning shows great potential and is a logical next step for the faculty's progress in robotic construction.

With the previous problem stated, an exploratory process to find the right software package started. Using search terms like "robot arm motion planning software" the MoveIt motion planning framework was the first search result. Not without reason, as the provided description states: "MoveIt is the most widely used software for manipulation and has been used on over 150 robots. It is released under the terms of the BSD license, and thus free for industrial, commercial, and research use." [96]. A robust platform with plenty of documentation, community support and an open source license is perfect for this application. Further searching also shows different commercial applications with these capabilities, however the cost of these would likely be prohibitive for this graduation project.

MoveIt is a software package that implements a broad range of motion planning algorithms specifically for use on robot arms, many

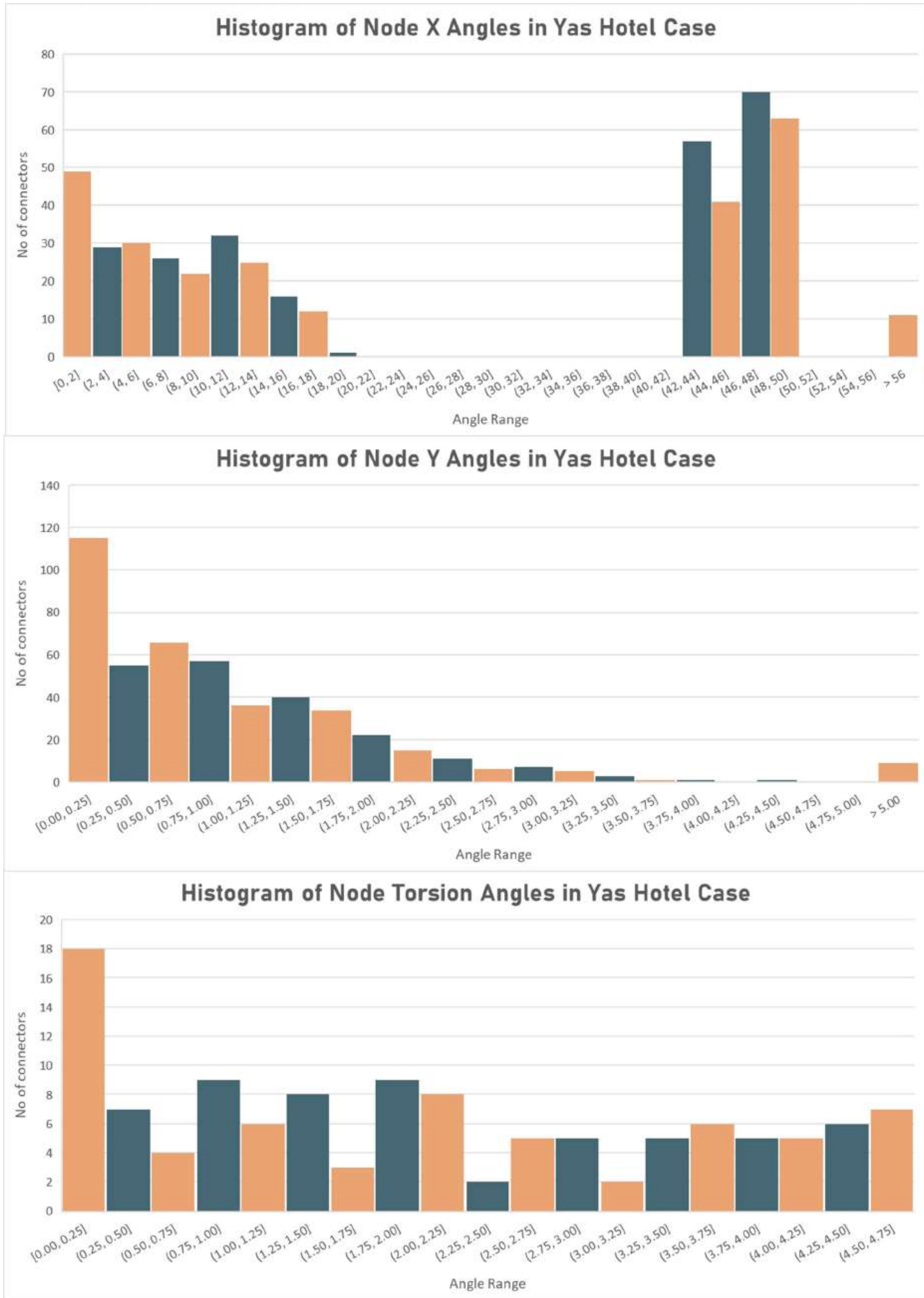


Figure 6.10: Histograms on the found X,Y and torsion angles in the Yas Hotel geometry, own work

from the *Open Motion Planning Library* (OMPL) [97]. It is a package for *Robot Operating System* (ROS) [98] which is a widely used, open source, middleware for robotic operations. Middleware is a standardised translation layer between different levels of operation. Robot drivers are implemented separately from solving algorithms with a common language, allowing for mix and matching of all software packages. Before work on implementation could start the software had to be installed, as advised in the documentation the most recently stable ROS 2 Humble and MoveIt 2 Humble versions.

Trying to follow the seemingly simplest installation method first led to an attempt to install on Windows 11. ROS is originally developed for Linux but has been ported to Windows in 2018. With little initial success and not much documentation on fixing the windows related issues it was decided to abandon this idea and move to Linux. There is still great potential in porting the entire system to Windows as Rhino and Grasshopper do not have Linux integration and synchronously working on two operating systems is difficult to streamline, this would require future research.

After a couple unsuccessful attempts to install in a Virtual Linux environment. A dual boot method was used to install Linux Ubuntu 22.04 LTS. Ignorant to the exact definition, the entire MoveIt library was first built from source, running into insufficient RAM issues. Once it was understood that packages should only be built from source if one intends to change the code, the binary MoveIt package was installed and the first standard tutorial was launched successfully. The next step would be the implementation of the available hardware.

Integration of the UR5 was relatively simple as an extensive ROS and MoveIt package for Universal Robots was developed by the company itself. Installation and launch of the UR5 in the MoveIt framework was successful and initial collision free movements around a simple box, shown in figure 6.11, looked promising. Integration of the DHAG95 gripper was significantly more arduous. The gripper itself is not one from a very established company and lacks documentation. It does have an outdated driver package for ROS 1, but that sadly fails to build on the newer ROS 2 versions. After many hours of bug-fixing in C++ an improvised solution was found to solve the issues and the plugin was built successfully to use the gripper. An offline simulator of the gripper was also developed to test without having the hardware connected.

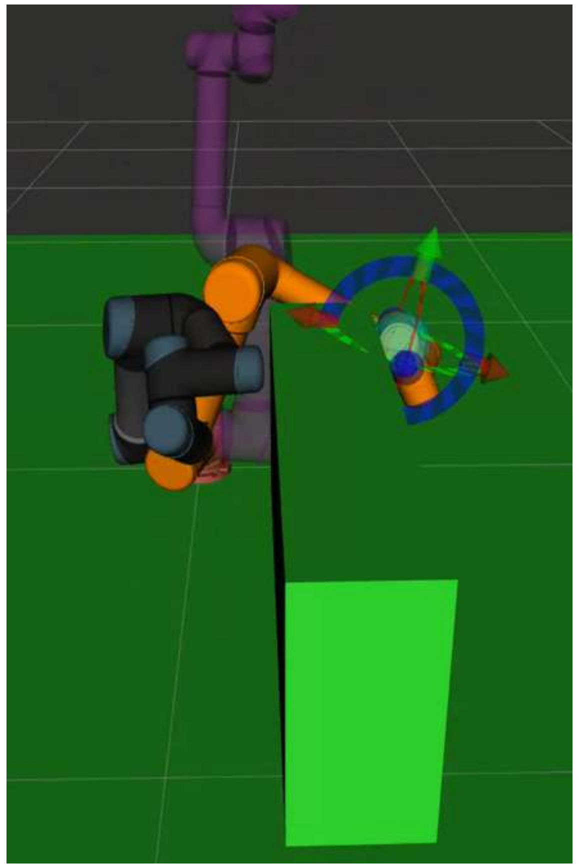


Figure 6.11: UR5 moving closely around a box without collisions, the generated path is previewed in purple, own work

Both pieces of hardware are now separately integrated into MoveIt 2, however as the gripper is not attached to the UR5, the two hardware simulators needed to be combined into one. Combining the geometry of both systems was simple enough and only required merging the *Universal Robot Description Files* (URDF) and the *Semantic Robot Description Files* (SRDF). In order for MoveIt to control the robot arm and gripper it needs to be able to read states and send commands. States are published by the UR5 and DHAG95 drivers in a joint states message, a custom script was written to combine these two messages into one. Commands are sent back to the drivers that contain controllers to interpret them. The DHAG95 Controller was updated and combined with the UR5 controller

provided for MoveIt integration. With the hardware integrations combined the real UR5 and DHAG95 Gripper could be moved through ROS & MoveIt.

Although working well for collision free movement from A to B this framework is not yet sufficient for a full system building workflow with multiple parts. A promising recent addition to the MoveIt framework is the MoveIt Task Constructor (MTC) [69]. Within a pick & place pipeline there are usually six frames, approach-grab-retreat-approach-place-retreat. In the now outdated MoveIt pick & place pipeline the direction for the approaches had to be user defined and only one solution attempt would be calculated. The MoveIt Task Constructor generates many grasp poses and approaches to be combined in the most effective program, as shown in figure 6.12. Where the pick & place pipeline could get stuck in its linear solving, the task constructor generates solutions non-linearly, ensuring a near-optimal solution for a list of feasible operations.

The task constructor code provided by the tutorial is only able to move one primitive object from A to B. Naturally, to build a complex system, the assembly of more objects will be required. Initially, the chosen approach was combining multiple tasks of one object after another. However, this would require every consequent task's end and start to be the same position and did not show initial success in code implementation. Combining all objects into one task by looping through the task generation script was successful in moving multiple objects.

In order to support a parametric system building workflow, a link between the MTC and Grasshopper is necessary. While it has not yet been implemented in code, how the system would work has been envisioned. As shown in figure 6.13, the MTC needs to receive objects and instructions from grasshopper. Objects can be transferred as an STL file, if geometry is complex a simpler collision geometry should be added. Instructions can be packaged in any generic data transfer format like JSON, XML or CSV. A flexible instruction framework should be used in which omitted instructions are automatically generated by the MTC.

Packaging all object and instruction data into files that are manually transferred between operating systems is not a fully synchronised solution and will not support synchronous feedback to Grasshopper. A server-client setup would be much preferred, but is also significantly more complex. In such a system the MTC client would be hosted on an accessible server, the generated program and other important information will then be synced back to Grasshopper. This type of system would greatly simplify the use of the MTC for any other project within the faculty. As mentioned in section 3.3 the easier it is to access software, the more it will be used and further developed. Connecting the MTC to Grasshopper using file transfer or a server-client connection will require future research.

An important function in human-robot collaboration is the ability to pause, resume and partition a program. Many robotic applications do not have a robust system for this, with MoveIt seemingly being no exception. The MTC is focused on full automation and does not currently implement any collaboration features. Without all needed hardware for full automation, collaboration becomes key to successful robotic construction projects. Developing collaborative features in more advanced automation tools will allow for more flexible application for many parties limited by hardware constraints and should be considered in future research.

To conclude, the use of ROS & MoveIt to implement task and motion planning into the robotic construction process shows great promise. The MoveIt Task Constructor has the potential to automate many steps that previously had to be coded manually such as collisions and approach directions. However implementation of ROS & MoveIt is quite complex, especially for someone with an architectural background not very experienced in robotics or lower-level coding languages. Due to time constraints caused by the complexity, further development of the ROS & MoveIt framework for this project was limited. For the continuation of the practical assembly the Robots plugin for Grasshopper will be used and the collisions and approaches will be manually coded.

6.6. Practical Assembly

With the parametric node design, the placement algorithm and the robot software choice

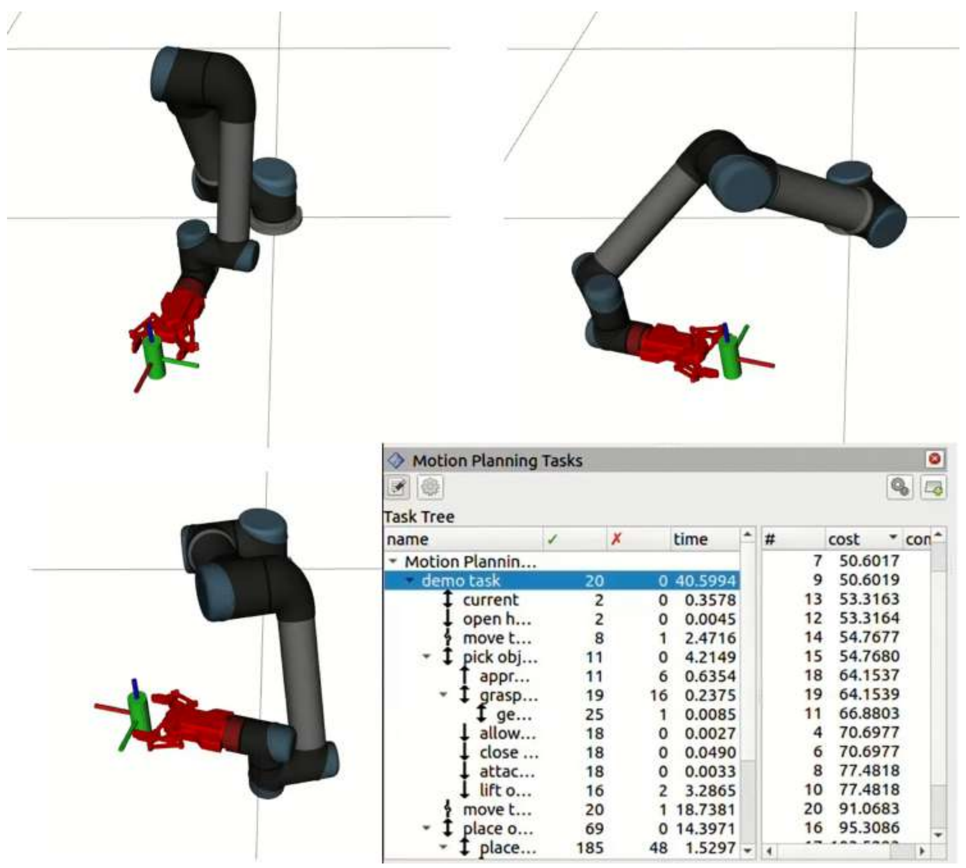


Figure 6.12: Different grasp poses generated by the MoveIt Task Constructor, for this simple task 20 valid solutions were found, own work

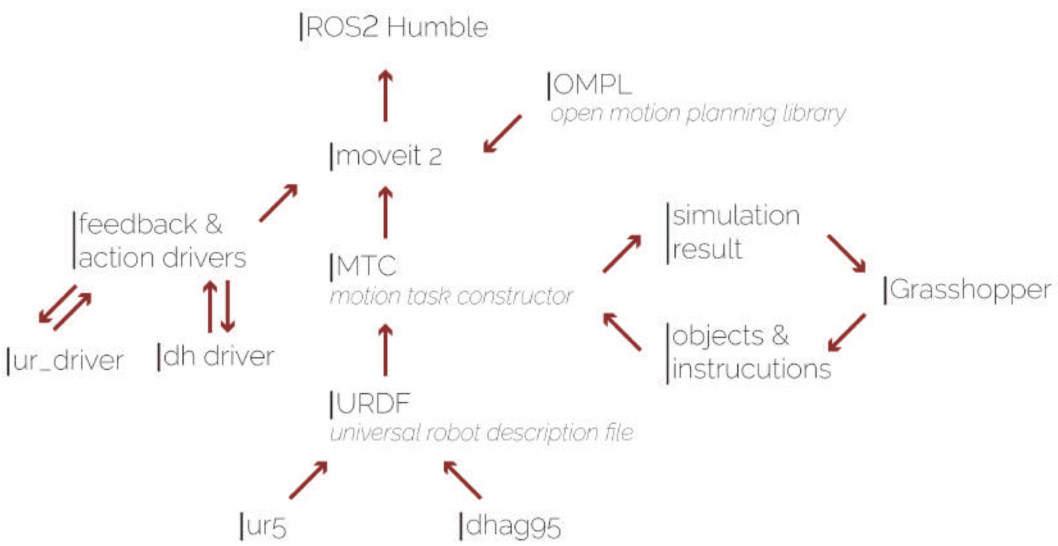


Figure 6.13: Software setup of the MTC, everything is implemented except for the Grasshopper connection, which is theoretical, own work

finalised, the preparations for the practical assembly can be initiated. This section will approach this challenge through the following steps. First, the size and scale of the model are specified and the shape and curvature are generated. Second, the placement algorithm is expanded to generate all the needed frames for the robotic construction and two scripts are generated. Last, the environment of the robot is designed, the internal parameters of the elements are configured and the model is robotically constructed with human collaboration. Based on insights into the coding experience, the building process and the human-robot collaboration this section will conclude with possible future research directions to expand on this project.

As with any design process, the design of the geometry for the practical model is limited by design boundaries. Physically, the size of the model is limited to the working area of the UR5 and the size of the table. The scale of the model is directly related to the height of the aluminium square profile used for the beams, which in turn defines many dimensions within the parametric node script. Choosing a small scale will allow for more elements to fit in the workspace and up to a certain point this is desired. The limiting factor on the lower bound of the scale is the local availability of small aluminium profiles, which in this case is a profile of 15X15mm. The amount of elements in the model is also defined by the horizontal and vertical division of the surface. Experience with previous robotic construction projects using the Robots plugin shows that a lower element count significantly increases the success rate of robotic construction. In order to properly showcase the system a minimal horizontal and vertical panel count of two is required, as four panels in total will provide one center node that is connected on all sides.

Using the established design boundaries a shape is generated based on two input curves. Since the panel count is very low, the real shape of the curve cannot be approximated very well, only the start, end and middle points are used. However, moving those points around does allow for the generation of a challenging curved shape, shown in figure 6.14. Applying the shape relaxation algorithm to this geometry will generate a proper planar and torsion-free mesh, but also removes much of the overall curvature that makes this shape challenging. As discovered in the Yas Hotel case, the torsion of non-optimised geometry can be very small. The hypothesis was established that the tolerance and flex in the system might be able to compensate for small torsion values. In the non-optimised geometry for the practical model the torsion is found to be quite small, as shown in figure 6.15. To better illustrates the capabilities of the system with a low element count and to test the torsion compensation hypothesis the choice was made to not optimise this shape with the relaxation algorithm, but build it with the internal imperfections. The final model, with the node placement algorithm applied, is shown in figure 6.16.

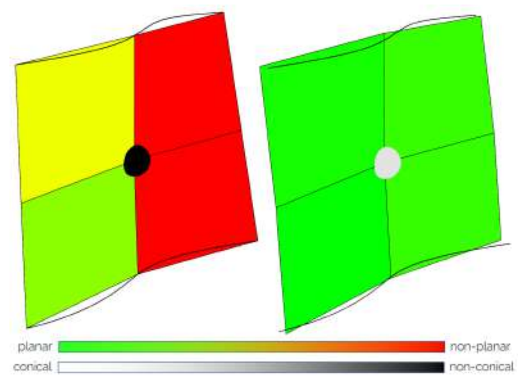


Figure 6.14: The shape difference between the non-optimised and optimised model versions, own work

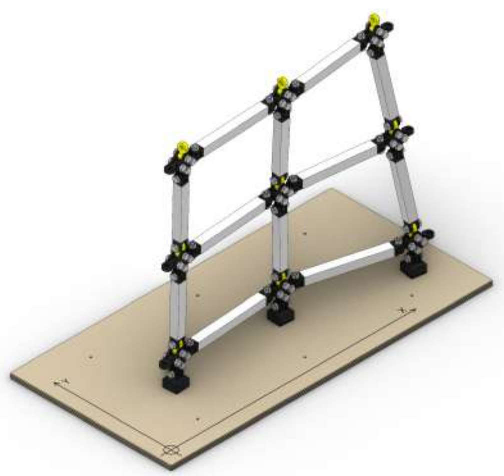


Figure 6.16: Computational result of the practical model design, own work

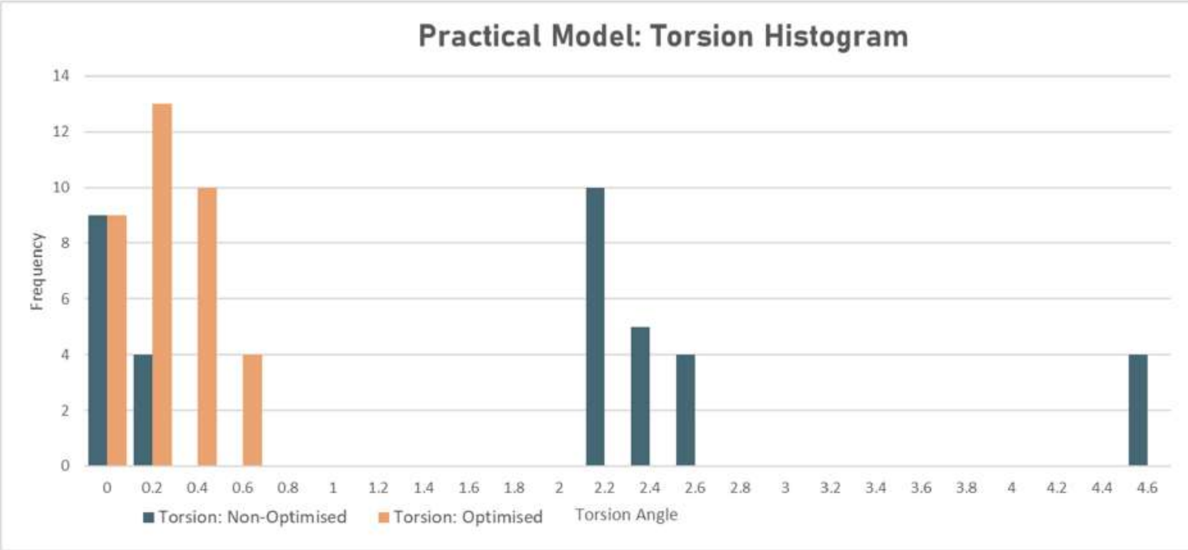


Figure 6.15: Histogram of torsion in both the optimised and non-optimised versions of the model, own work

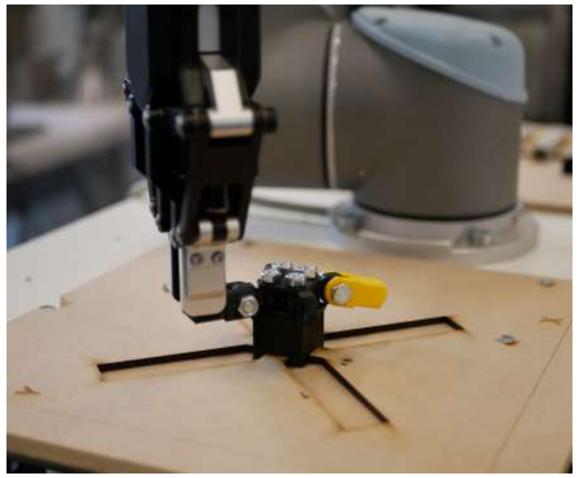


Figure 6.17: Execution of the angle setup script, own work

With the final model for the practical assembly generated, the node placement algorithm needs to be expanded to create the Robots plugin scripts. This is done in two parts: A script to set the nodes to a specific angle, and a script for the construction of all the elements. For the first script the angles are used to generate arcs, which are then split into frames with an interpolation zone. These are combined with approach and retreat frames, gripper commands, and joint resets. The latter being a position of standard joint rotations that prevent the Robots plugin from running into joint limits. All the frames and commands are ordered and combined into one robot program which configures the nodes to the required angles. The corresponding flowchart is shown in figure 6.18. The script is successfully executed, as shown in figure 6.17. After the beams are manually cut to the required length, all the elements are ready for construction.

The second script for the construction of all elements is generated in combination with the environment setup. The environment consists of base plates for picking and placing the elements which are automatically generated, visible in figure 6.20. The position of the base plates can be measured using a TCP measurement attachment connected to the UR5 end-effector. By measuring the center and X,Y directions, the difference between real world measurements and the digital perfect conditions can be offset in the script to increase accuracy. With both the pickup and place locations defined, all relevant planes are extracted from the placement algorithm and used to generate approach and retreat planes. All planes are organised and combined with the relevant wait and gripper commands to generate a full robot program. After initial testing collisions were found between robot-table and robot-model. By manually changing specific poses, approaches and retreats the program was generated collision-free. A flowchart for this script is shown in figure 6.19.

Before initial setup of the scripts it was already clear that the order of operations would be very important. Initially the chosen order was based on a press fit of the 3D printed

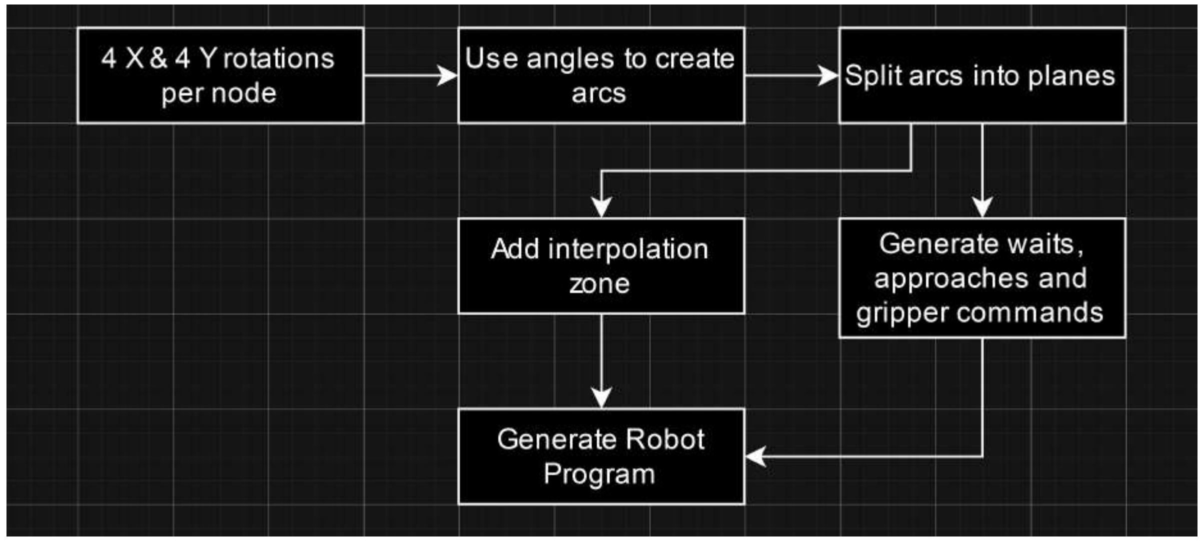


Figure 6.18: Flowchart of angle setup program generation, own work

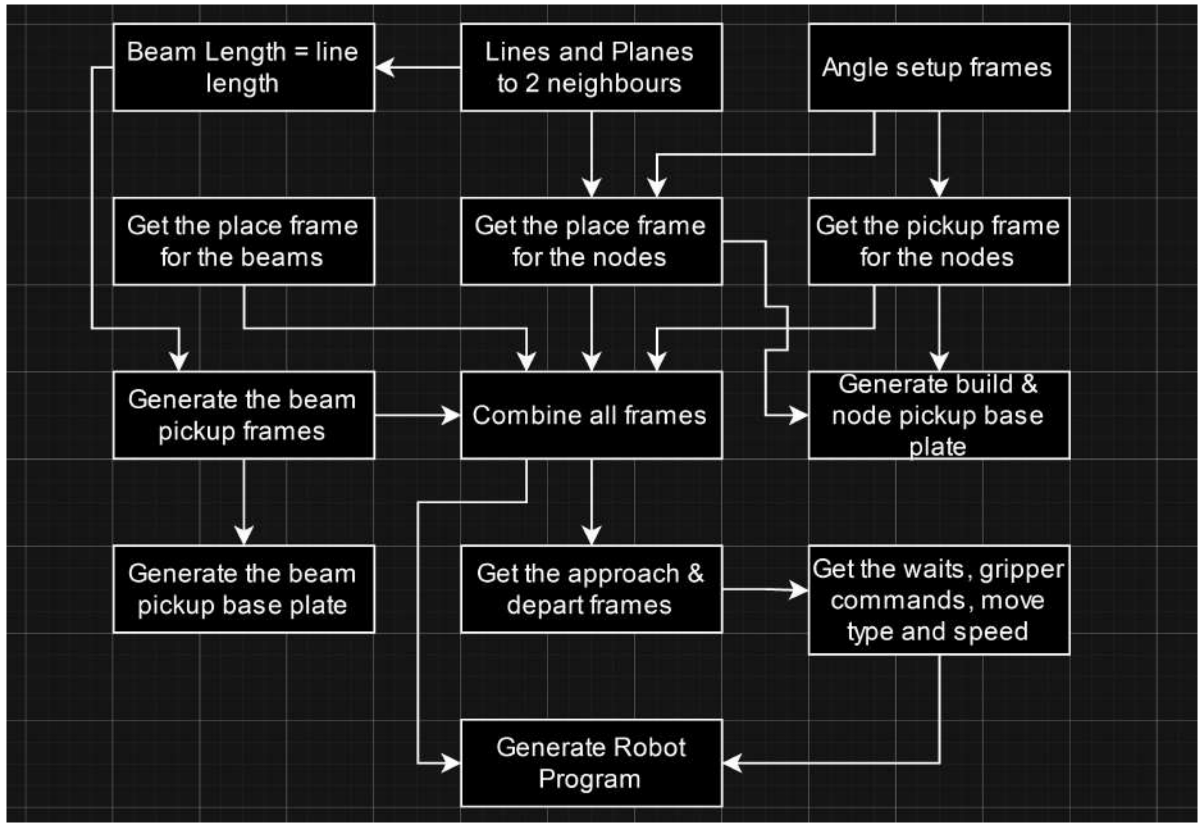


Figure 6.19: Flowchart of the construction program generation, own work

PLA connector and the aluminium beams. The exact dimensions for this fit were iteratively optimised to balance between the required accuracy and the strength of the connection. An initially promising result was found and the decision was made to continue with the build order as shown in figure 6.21. The scripts were developed for this method and ready for initial testing, but despite the initial promising results the press fit was significantly tighter in the Lab, likely due to differences in thermal expansion. Quickly it became clear that this system would not work and a better solution was needed.

The consideration was made to change the principle of the connection to alleviate the press fit problem, for example by using another nut and bolt. However, this would further complicate the node and require a re-manufacture of many parts which, within the available time frame, was not feasible. In the initially considered order of operations the press fit was only used for vertical beams, the horizontal beams had the connector pre-installed. As preinstallation of the connector foregoes the need for a press fit this principle was applied on all beams. This not only made the construction feasible, but also simpler and more realistic. Simpler because the amount of needed steps in both scripts is reduced and more realistic because a press fit connection in a full scale implementation would rather be made in preparation than in situ. The scripts were adapted where required and the new order of operations is shown in figure 6.22.

Using the new order of operations and the updated scripts, the practical model was assembled successfully using human-robot collaboration. The human interventions needed in this construction process were quite prevalent. This thesis considers the goal of robotic construction to be full automation where human invention would no longer be required. The cases of human intervention in the building process will be scrutinised and potential automation of these actions will be proposed. Firstly, human intuition was needed to fix inaccuracies of at max 3mm in the alignment of the elements. In a real world situation tolerances will always cause inaccuracies, especially when they are stacked over multiple parts. Comparing models generated with 3D computer vision to CAD will allow for adaptation to these small inaccuracies. Similar systems have already been applied in automatic part inspection [99]. Secondly, human dexterity was required to fasten the nuts and bolts at the connections. A robotic screwdriver can be used to automate fastening and a more dexterous purpose built robot can be used to reach difficult spaces. Robot collaboration is needed to hold and fasten a component simultaneously.

By not optimising the input shape, torsion of 0 to 4.4 degrees between nodes went uncorrected. It was hypothesised that the small tolerances and flex in the system could potentially compensate for these small discrepancies. Initially during the construction process the remaining torsion did not cause any directly apparent issues, the first parts connected well. However, in the later stages of the construction a buildup of stresses was observed. When the gripper would open parts would spring back into a slightly different position, this could be due to the unresolved torsion. On the final node, placed on the side of the structure with more torsion, the misalignment between the to build node and the already built structure was clearly visible. Although the construction was successful, it is likely that the unresolved torsion caused inaccuracies when compared to the CAD, and complicated the construction by adding stresses and misalignment. It would be interesting to repeat this experiment with a semi-optimised shape containing torsion of 0 to 1 degrees, to test whether smaller values can be compensated. Using 3D scanning the accuracy of both could be analysed, providing objective results. To expand on the relation between the unresolved torsion and the accuracy of the structure, further research will be required.

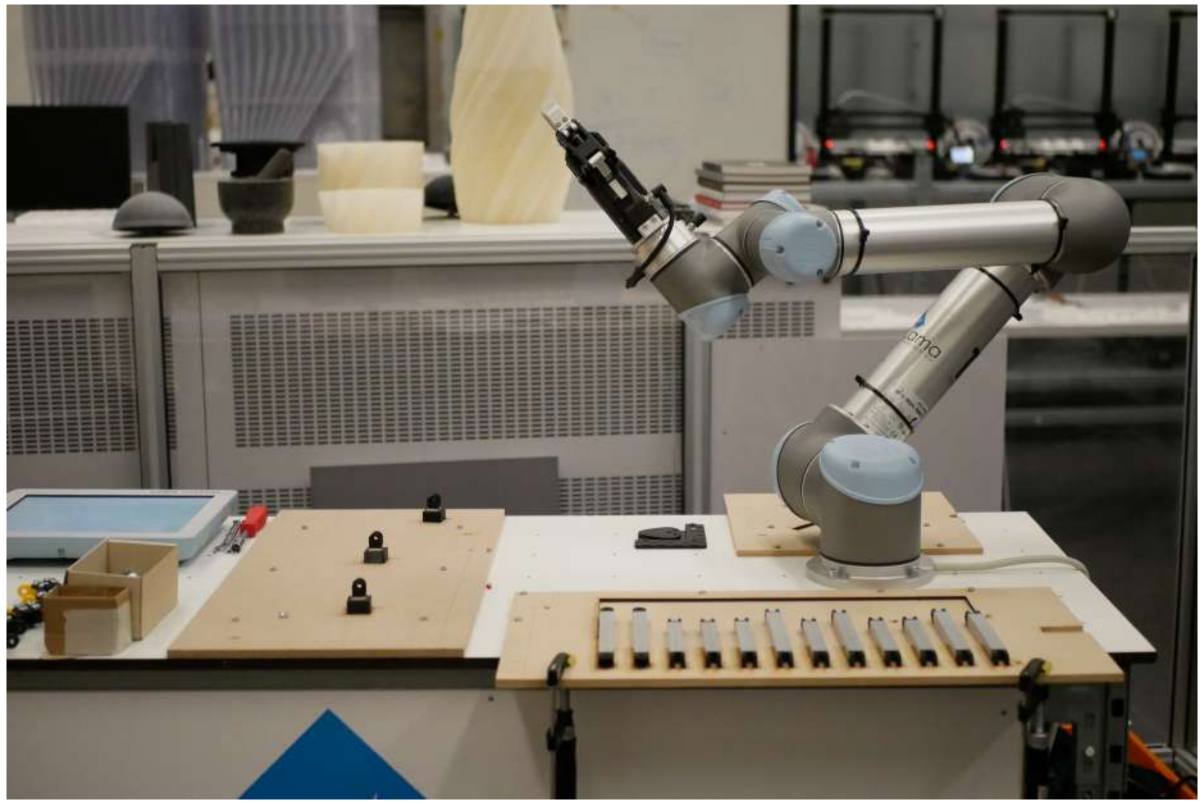


Figure 6.20: Setup of the robot workspace, own work

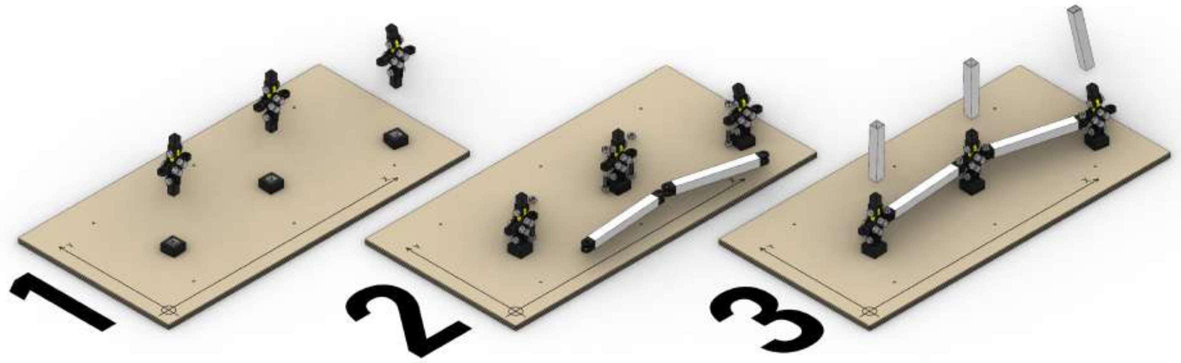


Figure 6.21: The first three steps of the initially chosen order of operation, own work

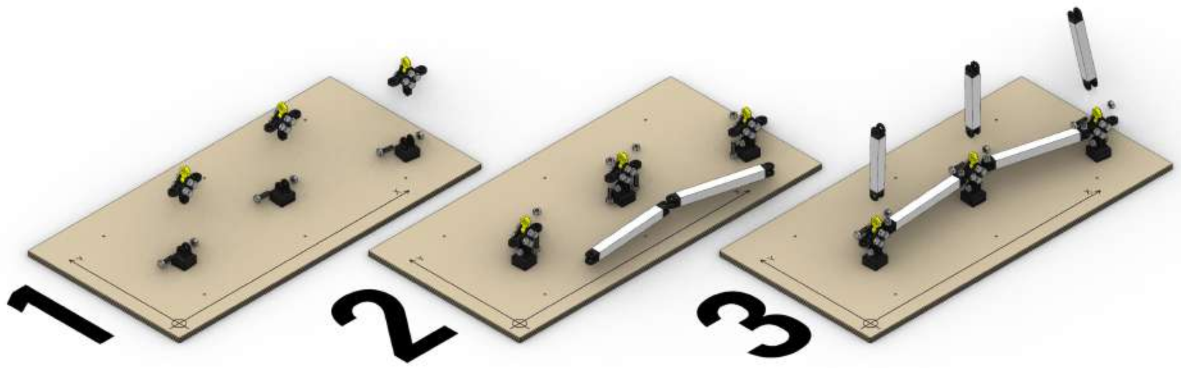
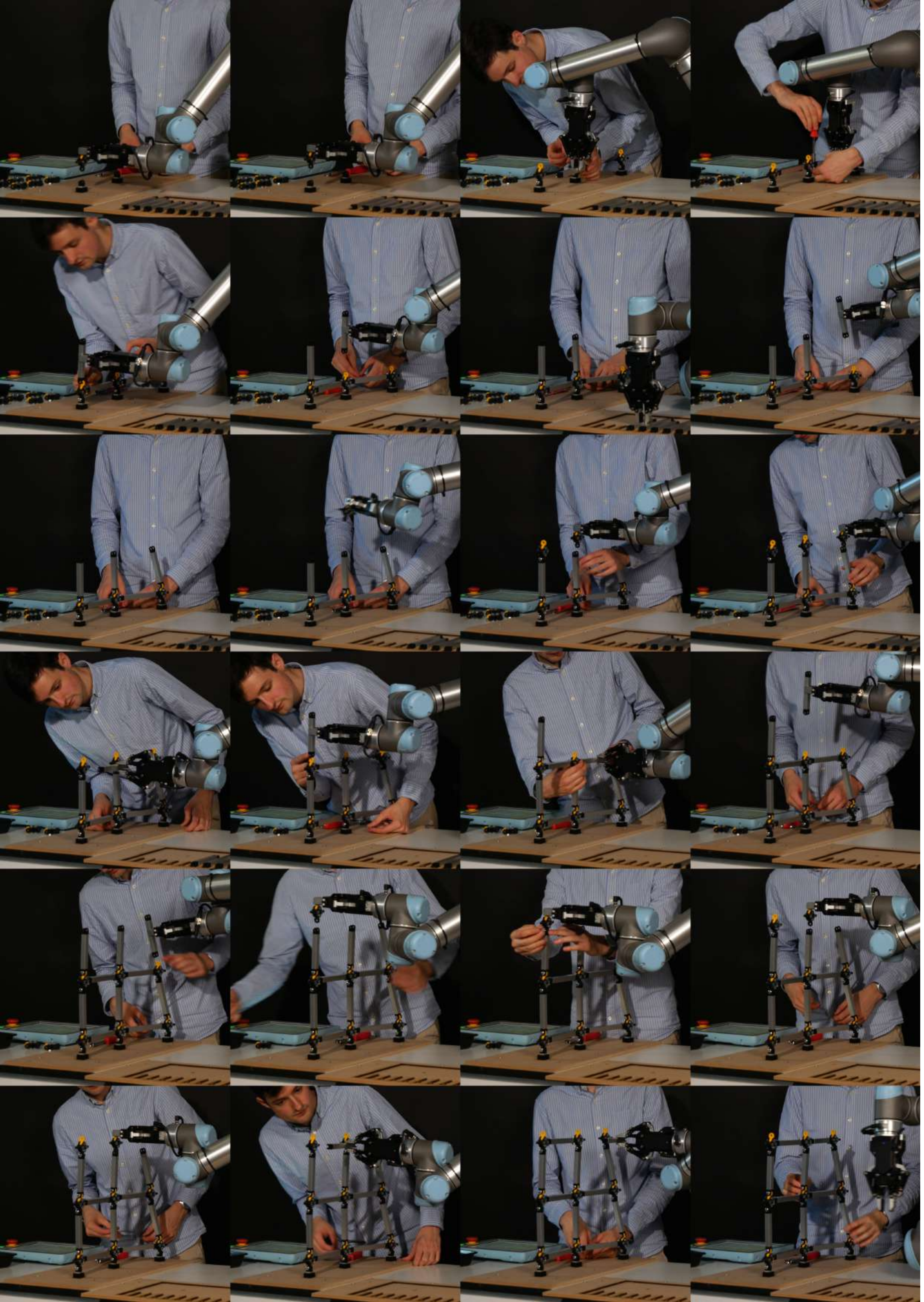


Figure 6.22: The first three steps of the final order of operation, own work

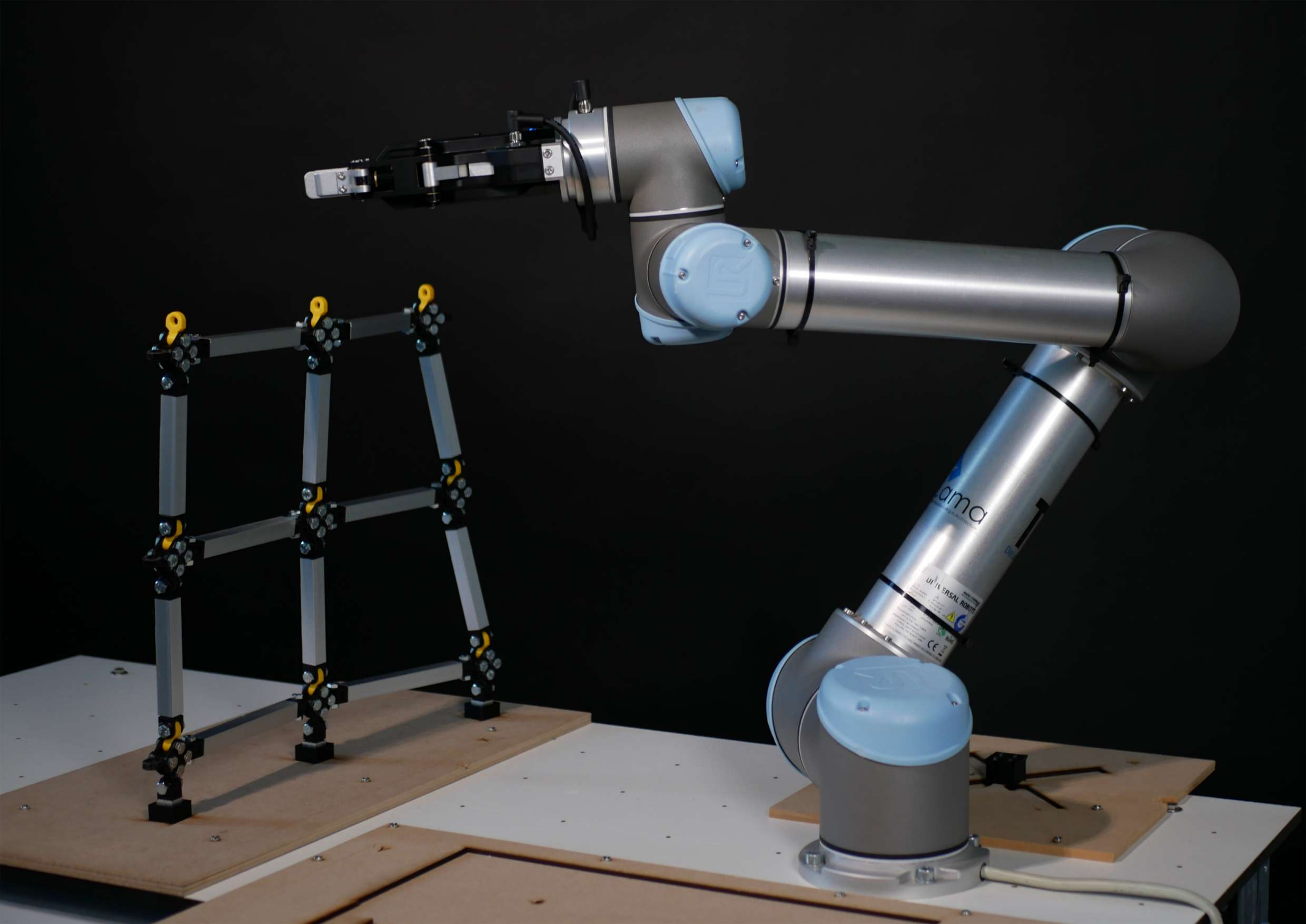


6.7. Conclusion

This chapter was structured around the question: "How can a reusable node and beam system for freeform building façades be designed and automatically assembled?". The design space of the reusable node and beam system was first freely explored to allow for quick ideas and iterations. A choice was made for a simple node design able to hinge in the X and Y axes which was further refined to increase the effectiveness and reduce the size. Next, a computational placement method that finds the required angles of the nodes and the lengths of the beams was developed. As a proof of concept the algorithm was tested on geometry of the Yas Marina Hotel in Abu Dhabi. Limitations of the relaxation algorithm were discovered which reduced the ability to optimise towards a planar and torsion free mesh. Continuing with a non-optimised mesh showed significantly small torsion amounts, which led to the hypothesis that tolerances and flex in the system might be able to compensate for remaining torsion.

The software to create a robot program for automatic assembly using task and motion planning was introduced in the form of ROS and MoveIt. Although this software shows great promise in automatic collision solving, approach generation and time optimisation, not all required features were developable within the time frame of this thesis. Missing the connection to Grasshopper and being unable to use this system with human collaboration made it unfeasible for use in the practical assembly. The choice was made to work with more familiar, but less automated software in the form of the Robots plugin in Grasshopper. Next, the model for assembly was generated within the available size and scale. To test the torsion compensation hypothesis the choice was made to use a non-optimised geometry. The scripts were expanded and adapted for the Robots plugin, the node angles were robotically set and the beams were cut to length. The practical model was successfully assembled, using human-robot collaboration. With additions of more technology and development the currently needed human interventions could be reduced and the system could be fully automated. During the assembly process it was found that not compensating for torsion likely resulted in internal stresses and a less accurate geometry.

Further research is proposed to expand and stabilise the computational placement script, to develop a synchronised connection between Grasshopper and MoveIt, to produce necessary features for human-robot collaboration within MoveIt, to implement computer vision, to reduce human collaboration for more automation and to further research the effect of unresolved torsion on the system. The sub-question can be answered as follows. A reusable node & beam system for freeform building façades can be designed through iteratively specifying and limiting the design space, the system can be automatically assembled by developing and testing a computational script which informs robotic software to generate a program for automatic construction, human-robot collaboration can be used if full automation is not possible due to technological constraints.



7 Pavilion Design

The previous chapter described the design process for a reusable node and beam system in freeform facades. The efficacy of the system on a larger scale was tested by applying a computational placement algorithm on an architectural case study and the constructability was tested with a practical assembly at a small scale. This chapter will extend on both these topics by answering the following research question.

> "How can the designed node and beam system be used in a computationally informed robotic construction process to automatically assemble full scale architecture?"

The answer to this question will be structured into five parts. First, an architectural design case that is highly related to development and expressions of freeform architecture will be introduced. Second, a design context which supports the introduced architectural expression is chosen. Third, the design concept is developed and its significance is discussed. Fourth, the design process is described and the final design is introduced. Finally, the process of part manufacture and automatic construction of full scale architecture is elaborated.

7.1. Design Case

Construction and architecture are generally seen as conservative sectors where innovation can be a sluggish process. Hardly a surprise when all requirements, laws and standards have to be applied before any new development can see real integration. While quality control and safety are of course of utmost importance, new technologies need a practical testing ground to iteratively improve their design. While many early steps can be performed in a controlled factory environment, at a certain point the technology and its architectural design needs to test the full scale feasibility and relation to the user in the real world. Frequently, this is done by designing and constructing a pavilion: a full scale testing ground in which the design experience is central.

A pavilion is not required to, and often will not, consider the direct functionality of the architecture. It may lack weatherproofing, insulation or windows and its practicality may not be directly obvious. However, it is exactly this freedom that inspires unbridled creativity at the crossroads of art, architecture and technology. The impact of pavilion design on architectural form and as a testing ground for new developments should not be understated.

A good example of pavilion construction powering technological research is the freeform tile-vault by the Block Research Group [100], shown in figure 7.1. For this pavilion a novel form finding method was developed, the shape was computationally generated and through construction verified in real world conditions. Bringing the research results one step closer to integration in the construction sector. Another example is the usage of folds to improve structural rigidity [101]. The research was proven by constructing pavilions and has been implemented in functional structures [102].

Figure 7.1: Tile vault pavilion, by Block Research Group

Figure 7.2: An example of a folded roof construction, by Hoenigschmid-DeVeaux [102]

Next to developments in form finding algorithm, pavilion construction is often used to inform material research. Such is the case in the Steampunk pavilion by Fologram searching for the limits in bent wood construction [103] or the Bamboo Stalactite Pavilion By Vo Trong Nghia using novel bamboo construction to prove structural efficacy [104]. Additionally pavilion construction is applied to proof new methods of digital manufacturing. As is the case with the Pillars of Dreams by Theverymany, using many precision laser cut aluminium pieces to create freeform shapes or the FUSE pavilion by Formlabs, illustrating the potential of 3D printing for light-weight discrete nodes [105]. These pavilions are shown in figure 7.3.

In conjunction with form finding and material research, pavilion construction has also pushed the envelope in robotic construction. The Research Pavilion by ICD and ITKE 2012 [106] uses robot arms to enable a carbon fibre weaving process. Similarly, the Robotically Fabricated Structure by Adel Design Research uses robotic collaboration to parametrically construct elements combining into a curved wooden pavilion [107]. Combining the use of pavilion construction as a proof of concept with a theoretical approach to the literature review of chapter 5 creates a design space of broad potential to develop a futuristic construction ideology. Both are shown ins figure 7.4

7.2. Design Context

Pavilions are often temporary installations seen as an art work and therefore auctioned off and moved at the end of an exhibition period of often one year. The past decades have seen a rise in pavilion exhibition spaces both as technological displays and as art installations. The creators can be competition winners or invited established architects and artist. The most famous pavilion exhibition is the yearly invitational at the Serpentine Exhibition in Kensington Garden, London. Since 2000 its success has been a key factor in the popularity of contemporary architectural exhibitions.

Having been the host of many of the worlds leading architects the Serpentine Exhibition has set a strong precedent in showing the artistic intent of pavilion architecture. It allows architecture to be independent from rules or expectations leading to the purest executions of the design intent. Consequently, it highlights the importance of a well defined intent and a properly matching design, as the quality of the pavilion is directly and only related to these two factors.

Although the pavilions of the serpentine exhibition are fully focused on architectural design, the technology required to construct some of the designs still challenges engineers and pushes the boundaries of technological development in the construction sector. For this thesis Kensington Garden will be chosen for the design of a pavilion using the developed node and beam system. This choice will provide a design space that is both influenced by its context as a public park and completely free to explore any design intent.

7.3. Fluid Space

Louis Kahn described architecture as the thoughtful making of spaces, it is the creating of spaces that evoke a feeling of appropriate use. Although said in the context of the modernist movement of the early 20th century, this description is still an accurate definition of contemporary architectural design. Spaces are designed with the balance between function and form at the behest of the designer. The way these two design foundations influence and inform each other should be thoughtfully applied as its rigid nature will not allow these spaces to change.

Shifts in desired artistic expressions of forms and in the objective requirements of functions come with time, which raises the principal

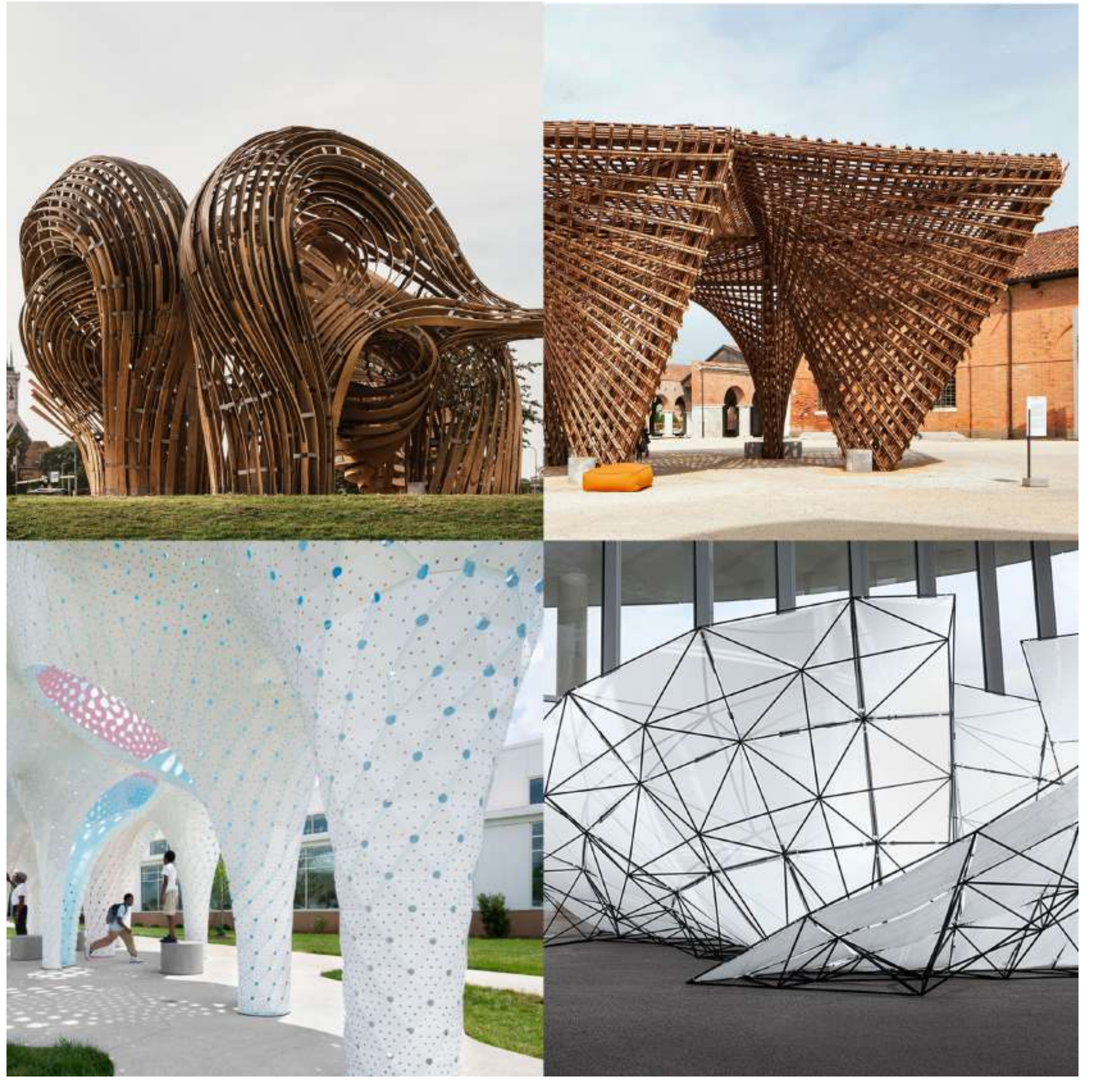


Figure 7.3: These pavilions showcase their use of material, authors in text

Figure 7.4: These pavilions are built using robotic construction processes, authors in text

Figure 7.5: Pavilions built as part of the Serpentine Exhibition, via ArchDaily

question if space should inherently be static. Limited by the contemporary technological scope, architecture cannot be anything else than static, considering future technologies might provide potential for a new architectural ideology.

Changing functions of spaces during a buildings lifetime is a factor that is increasingly taken into account in building design. For example modular partition walls are used in office buildings to prepare for a different functions. Although showing great potential in extending a buildings lifetime, it is limited to a change in function and cannot adapt to a new form.

The balance and interrelation between function and form is essential to the experience of spaces. Form might need to adapt to better suit a changing function. In this way space would no longer be rigid, but can be considered as fluid. Being able to fit into any changing requirement will open up a new method of design, where it is no longer the initial design, but rather the evolution of the design that takes center stage.

Combining reassembly of a flexible node and beam system with automatic reconstruction using robotics allows for an architectural implementation where all initially rigid design conditions can become fluid. In order to explore the potential of an ever-changing architectural expression a context is needed where fluidity can be emphasised. Considering the entire Kensington Gardens as the design context opens up many different positions with different local conditions. Moving the pavilion between these positions and adapting the form to the local conditions will concretely show the potential of reassembly to create fluid architecture.

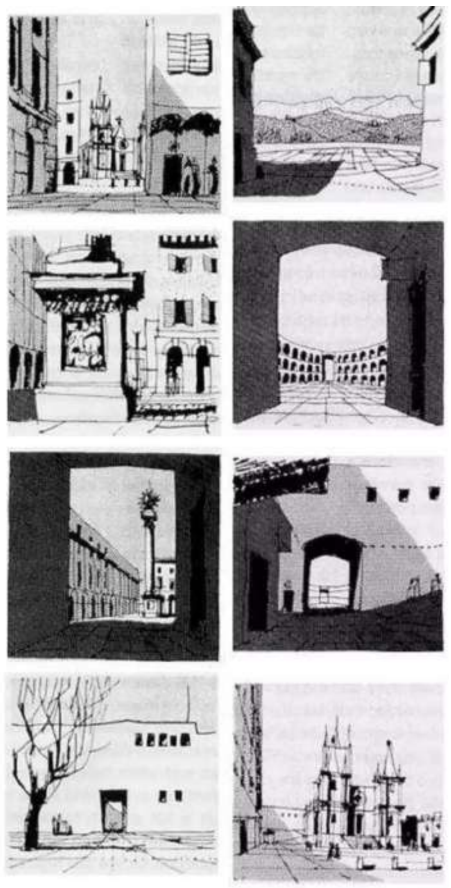


Figure 7.6: Serial vision as the experience of space, by Cullen [108]

7.4. Design Process

Throughout this chapter the design assignment is specified to a pavilion that fluidly moves and changes to its changing design contexts. First, it is explored how form can respond to one of the most important aspects of a park context, circulation. Next, the manifestations of the different shapes are freely designed, showing the full potential of the node and beam system.

Circulation through public spaces is often motivated by architectural elements that impose a sense of direction, which are applied at every scale of the built environment. For example horizontal banisters that limit and guide the flow of movement with a minimal visual presence. Hallways can lead straight to a destination or twist through the context, informing the user on the architectural intent. Alleys or tunnels impose a feel of mystery and curiosity and highlights can act as attractors in the distance. If a sense of direction should be applied in architecture, only considering the static appearance from one perspective is insufficient. In the book, The Concise Townscape, Gordon Cullen described the importance of movement in the experience of spaces as the principle of serial vision [108]. In his example of the townscape, shown in figure ??, every stage of movement is motivated by a new point of interest, keeping the passerby captivated by the twists and turns of a well designed urban fabric.

A combination of a dynamic form and motivation of circulation direction can inform a design for a fluid pavilion. The first design is parametrically generated to emphasise two directions. The height of the arcs is coupled to the amount of emphasis a path should have. A NURBS curve is generated with the end points of these lines as control points and an oval is generated around the Physical Energy statue. These are used as rails with a cross-section curve in a Sweep2 to generate the roof geometry. ??. A choice can also be made to equally highlight each circulation direction as shown in the second design. Of course, as a fluid implementation the system is not limited to crossroads and can also serve as a highlight along a single path, as shown in the third design.

When shifting between the different designs the internal parameters of the nodes and beams will change. For the nodes this is not an issue as their configuration can be unlocked, changed and locked again. However, in the current system the monolithic beams do not share this flexibility. During every change some building components will have to be exchanged at the part library. Additionally, although panels are shown in the visualisations, this is only for visual reference and not part of the scope of this research.

7.5. Manufacture & Construction

Before automation on the construction site can be considered every process in a controlled environment should be automated. Industrial manufacturing methods are required to fabricate the parts of the variable node and beam system at full scale. As many sub-structures, like curtain walls, are made from aluminium for the strength to weight ratio, this material is first considered for the full scale manufacture of the parts. The advantage of using non-discrete parts to construct the variable nodes is the possible repetition of their production process. Fabricating a high number of the same part opens up opportunities for processes with a significant startup cost. An example is aluminium die casting which allows for efficient automated creation of complex geometries, but does require the manufacture of a costly die, perfect for a system of non-discrete parts. Usually a machining step is performed on any cast surface where precision is important. Due to the design of the variable node this step is not required, the hinge can work with slightly inaccurate geometry. When the node arms are set to their angles and are clamped into position slight surface inaccuracies are not deal breaking. In fact, the added surface roughness and the inherent friction coefficient of the aluminium will likely increase the strength of the friction lock. The friction of the hinge can be further increased using multiple hinge flanges as it creates more surface area.

In the current node design, four different dies are required. With a slight redesign this could be reduced to three. The top plate of the node would then be cut on a water cutting table. The outer connector of the node currently has a press fit connection, this requires a high degree of precision manufacturing and would need surface milling after the die casting

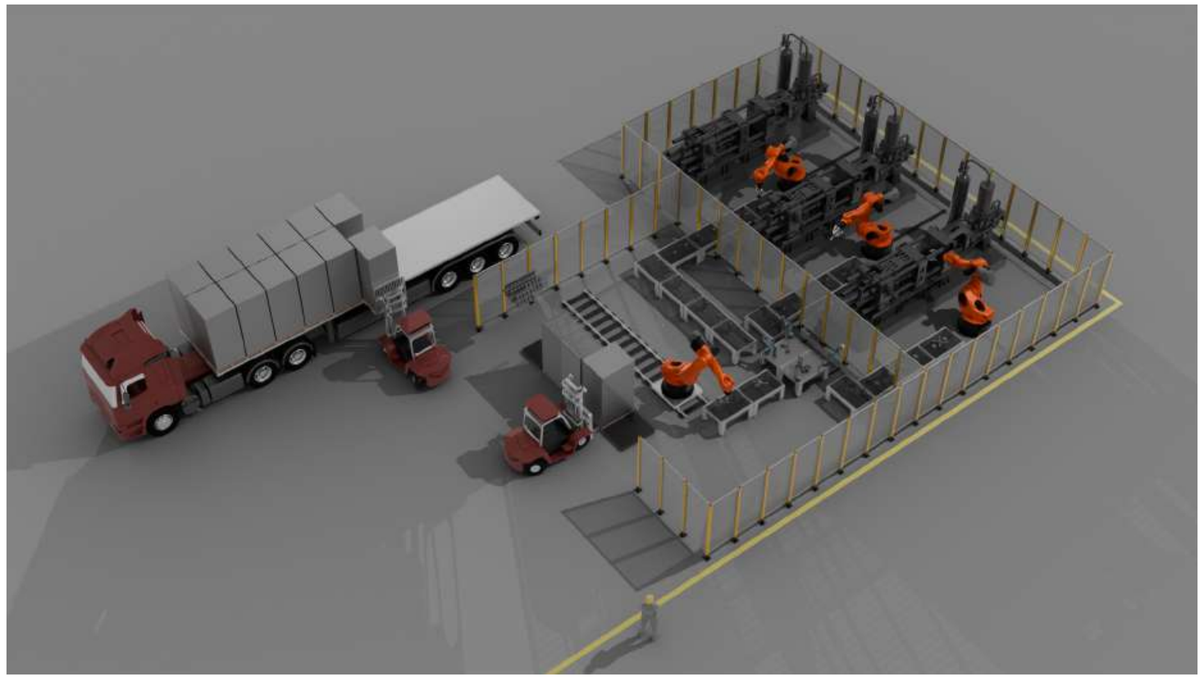


Figure 7.7: The factory process shows die casting, extrusion and automation, own work

process, which is inefficient. This connection can be changed to work with bolts instead of a press fit. It is also possible to automatically weld the outer connector to the beam, however that would make shortening of the beam in the reuse process significantly more complex. A face-face bolted connection is optimal for the transfer of forces and is used in many existing node and beam systems. An updated parametric definition of the node design is made to adapt to these requirement, a full scale node for 50x10mm beams is shown in figure 7.9.

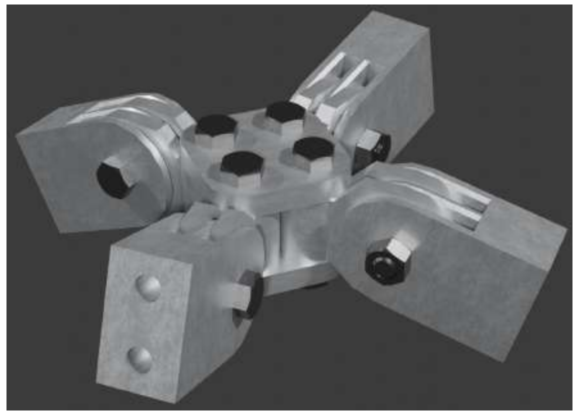


Figure 7.9: The design of a full scale node with multiple flanges, face-face connections and smaller bolts

The aluminium beams can be extruded. This is a process where aluminium is pushed through a die with high pressure to form long elements with a planar cross section, like a beam or window frame. The extruded beams can be cut to a standard length using an automatic pipe cutting machine. After transportation to site, they can be cut to the required length for their position. The hinged connectors and base plates are bolted together to form the nodes. As the final step in the controlled factory environment everything is organised for transport. In a fully automatic future the parts would be autonomously transported to the site.

Arriving on site, removable and reusable foundations should be laid. This can be automated by using adapted heavy construction equipment to enable robotic operation of these machines. As the foundation will remain stationary, heavy equipment is no longer required after this step. The nodes and beams will arrive on site together with the robotic construction crew. The system is then built in an optimised construction order using high-level task and motion planning. Using 3D sensing capabilities the robotic system can find its exact position based on the foundations. To place the first layer of nodes the mobile robotic system will collaborate with small material-system integrated robots. The mobile robot

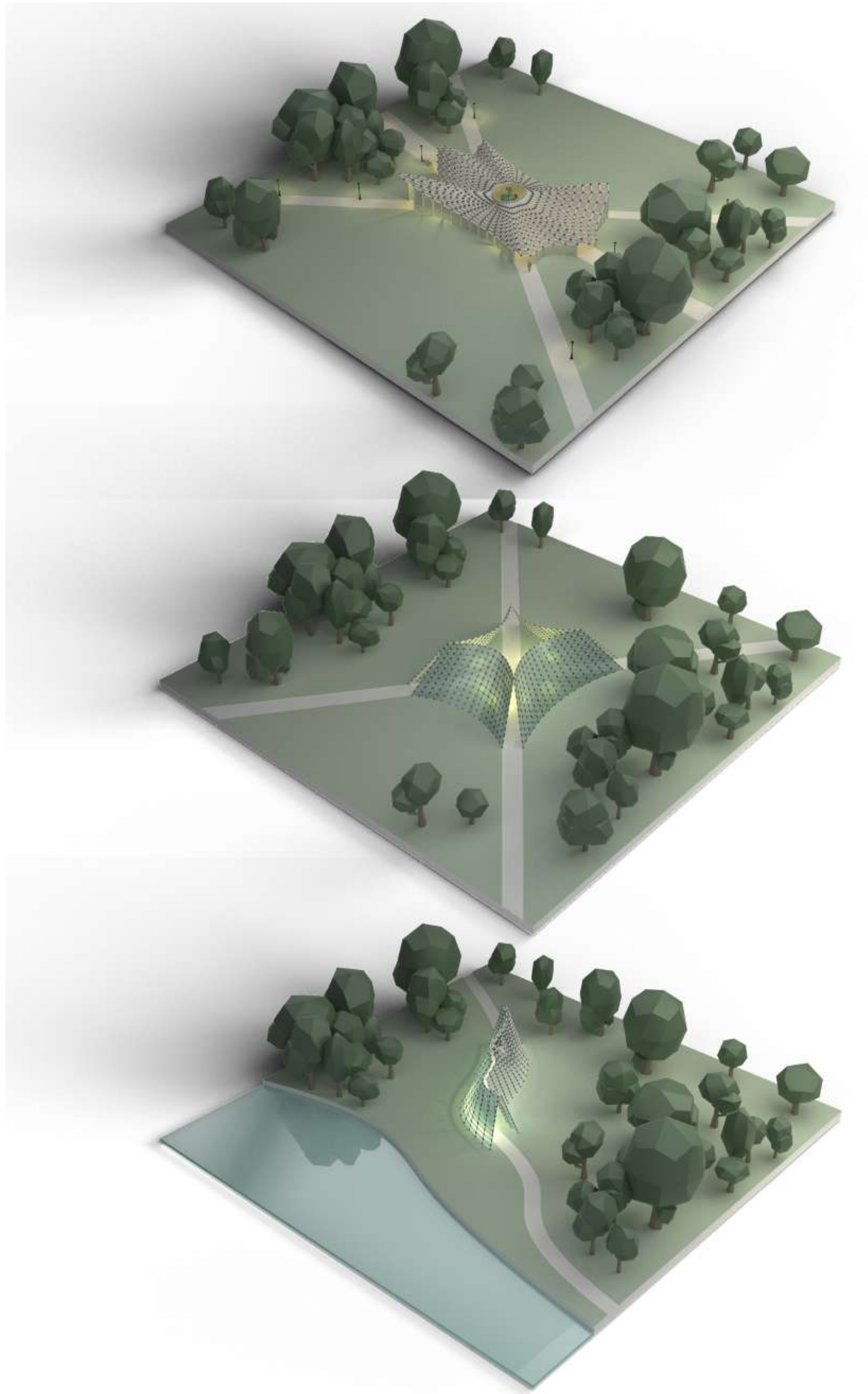


Figure 7.8: Different versions of pavilion design at different design contexts throughout Kensington Gardens

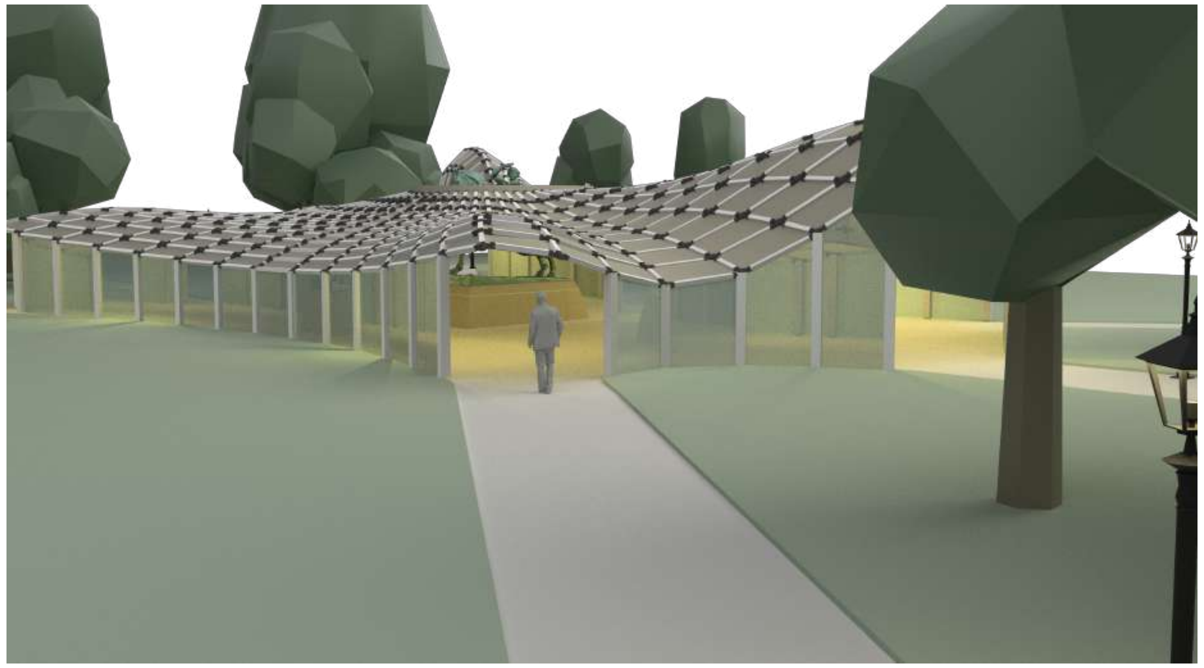
will transport the nodes close to the correct position after which they are precisely placed and fastened by the small robots. The internal parameters of the nodes and beams will be set by the small climbing robots. After the entire perimeter has been constructed the mobile robot arms will similarly collaborate with the small climbing robots to step by step construct the entire pavilion.

After construction is complete the robotic building crew is removed from the site and the pavilion will stay in the initial configuration. When the time is ripe, the robotic construction crew will return to disassemble the system step by step. Any parts that can be reused are locally transported to the new location and other parts are retrieved from the library. At the new location the system is reconfigured during the construction process and a new form can be realised.

7.6. Conclusion

This chapter was structured around the question: "How can the designed nodes and beams be used in a computationally informed robotic construction process to automatically assemble full scale architecture?". This question was answered through five steps. Firstly, pavilion construction was introduced as a motivator of architectural exploration and technical development through providing a creative design space with few limitations. Secondly, a context for the pavilion design was found in the serpentine exhibition in Kensington gardens, London. Being the driving force behind the renaissance of pavilion design since 2000 and having hosted many exploratory designs, this context was chosen for the further process. Thirdly, the design concept of fluid space was introduced. Variable building systems in combination with robotics could not only offer a reuse of components after deconstruction, the combination can also facilitate a changing fluid form that adapts to its local context. Lastly, a manufacturing process using industrial methods was introduced and the dynamics of a full scale robotic construction process were explored.

The answer to the sub-question can be formulated as follows. The designed nodes and beams can be used in a computationally informed robotic construction process for full scale architecture by first specifying the design case of freeform architecture as a pavilion construction, then computationally generating a pavilion design and lastly by detailing the manufacturing and construction process of a full scale node and beam system.



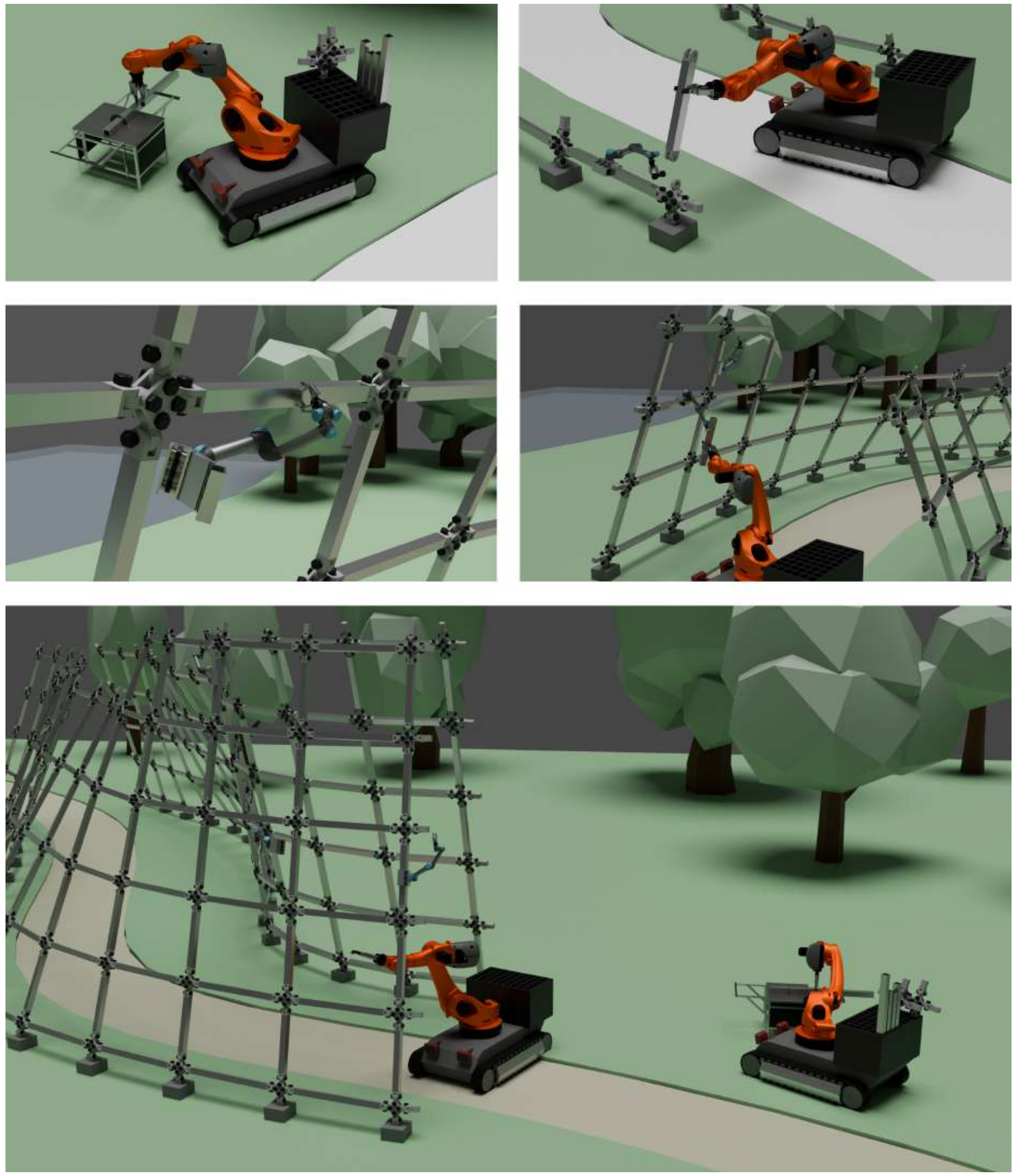
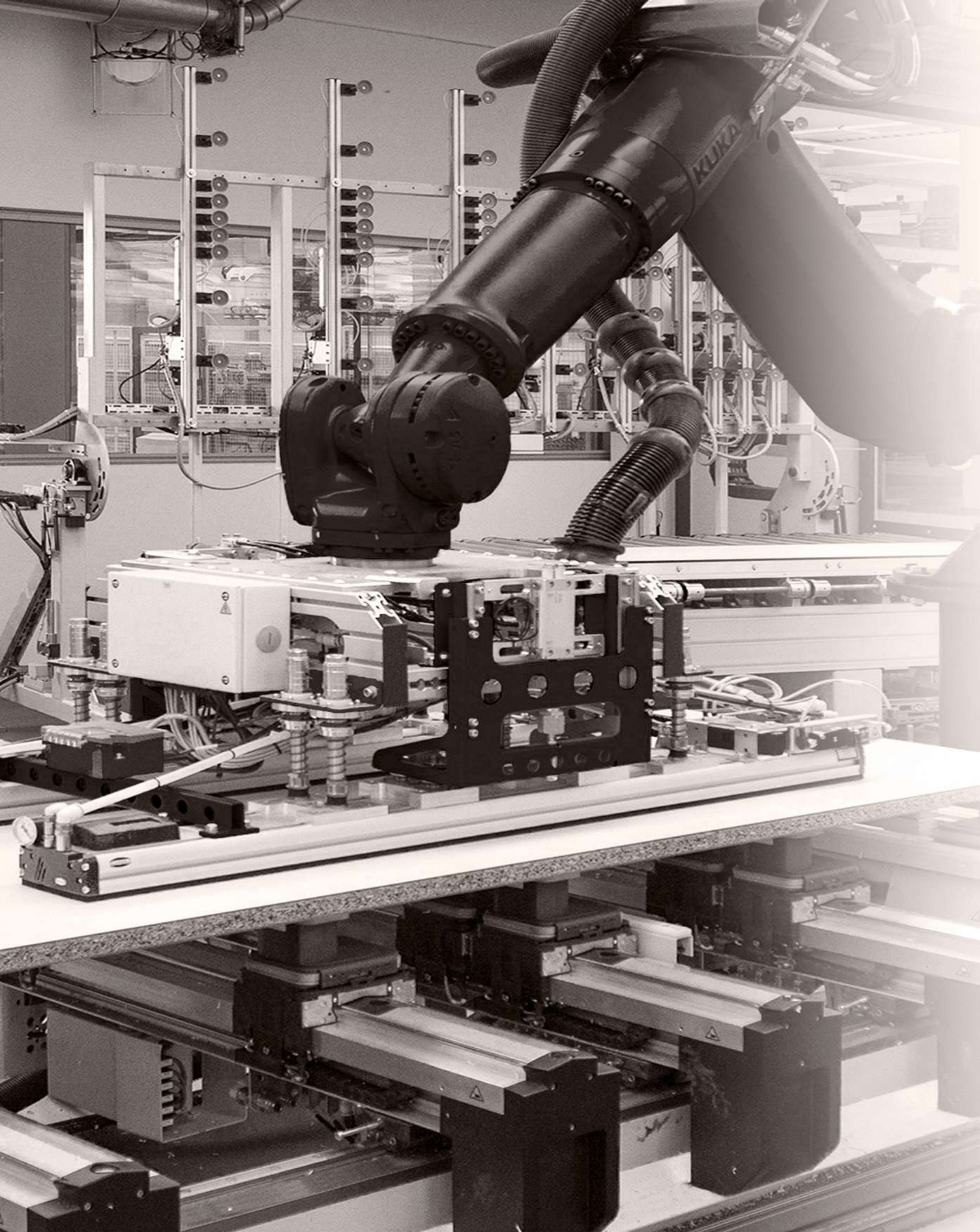


Figure 7.10: The steps in the construction of a full scale pavilion



8 Conclusion

8.1. Research Conclusion

The building industry as a sector of low productivity and high emissions should take responsibility for its shortcomings and actively seek improvement. The building industry is a conservative sector that has only recently started to slowly implement digital technologies even though in architecture digital implementations have already had a clear impact on the design of high-end structures. 3D CAD software has provided increasingly easier methods for modelling complex geometries, so easy in fact that it has created a detachment between the designer's expectations and the physical complexities of these shapes, leading to prohibitively expensive, environmentally costly and low productivity custom solutions. Due to the exemplary position of high end architecture and the rapid development of 3D CAD software a growing trend in freeform architecture can be expected, presenting the industry with a unique chance to tackle productivity and emission issues pre-emptively.

This research aimed to develop a design to production workflow towards automatic assembly and circularity of nodes and beams in freeform building facades, it has been structured into five parts and the conclusion will be structured similarly.

First, geometry rationalization and mesh optimisation were researched through a literature review. The choice was made to implement a computational optimisation for any freeform shape, not limited to a specific generation method. The thesis' scope was limited to rationalizations of quad meshes using only horizontal and vertical directions. A mesh perturbation algorithm was applied to the quad meshes to ensure results of torsion free nodes which is an important step toward reduced node complexity. It was found that rationalization and optimisation of doubly curved geometries will always be a balancing act without an objective optimal solution. Using a curvature based variable pullback force a user of the computational script will have direct control over the software and so will be able to find a subjective optimal solution.

Second, the acquired theory on mesh rationalisation helped to define the nodes and beams by their internal parameters of angles and length. Two hypothetical systems of reuse were argued: a relatively complex variable system where the elements have adjustable internal parameters and a library of simpler monolithic parts. By statistically analysing the element similarities between thousands of variedly curved geometries, the decision to design a variable node system and a library system for beams was substantiated.

Third, to facilitate the automatic assembly of building elements the state of the art in robotic construction was researched through a broad literature review which was structured into four levels of decreasing technological maturity. The levels were split into four topics: movement, sensing, solving and operation. The state of the art in practically implemented robotic construction research is a free moving robot system that uses advanced computer vision and task and motion planning to build complex systems. The future potential for robotic construction is likely an optimised collaboration between many robots with different tasks that might use the building itself for locomotion. Based on the hardware available for this thesis, the research on the assembly process is split into a practical and theoretical assembly.

Fourth, the node design was freely approached and the simplest and most robust design was chosen, iteratively improved upon and then parametrically defined. The thesis' computational backbone was developed further to extract node angles and beam lengths from the optimised quad meshes. As an architectural case study the computational system was tested on the geometry of the Yas Hotel in Abu Dhabi. Although the placement of the nodes and beams worked well, the optimisation of the quad mesh was unsuccessful in this test case. The small amounts of torsion observed in this non-optimised geometry led to the hypothesis that flex and tolerance could compensate for small amounts of unresolved torsion in the structure. *ROS and MoveIt* software applications were selected for robotic programming and the limitations of their implementation and the considerable work put in were discussed. Due to time constraints the robot program for the practical model was generated in the *Robots* plugin for *Grasshopper*. The model was generated within size and scale constraints and left non-optimised to test the torsion hypothesis. Through the assembly process it was found that not compensating for torsion would likely result in internal stresses and a less accurate geometry. The assembly process using human-robot collaboration was successful, the process can be further automated with more and elaborated hardware and further technological implementation.

Finally, research was done on how the node and beam system can be used in a computationally informed robotic construction process for assembling full-scale architecture. The importance of pavilions as testing grounds for architectural innovation was introduced. Pavilions serve as platforms for research on form finding algorithms, material properties, and robotic construction. For design context the Serpentine Exhibition in Kensington Garden, London, was chosen, as the exhibition's focus on artistic intent and freedom of design allows for exploration without rigid rules. Next, the proposition was made that architecture in its rigid form is limited. The concept of fluid form was introduced, emphasising the need for adaptable architecture that is able to evolve over time. Multiple pavilion designs were made throughout Kensington Garden. The manufacturing process for a full scale variable node and beam system was discussed. Automated industrial manufacturing in a controlled environment was applied, followed by on-site assembly using robotic systems.

In conclusion, this thesis introduces a novel reusable node and beam system for use in the automatic assembly of freeform architecture. By optimising input shapes, applying computational placement of the elements and generating instructions for robotic systems, the building sector can not only improve its productivity and reduce its emissions, it can furthermore revolutionise the stylistic nature of architecture and facilitate the fluid adaptation of new forms and functions.

8.2. Discussion

Based on the conceivable improvements uncovered through the research process, the limitations of this thesis' scope and the flaws in the system for nodes and beams, this section will introduce topics for future research on mesh rationalisation, facade design and robotic construction. Before implementation of this thesis' reusable node and beam system can be considered, the limitations and flaws have to be ironed out. To emphasise the impact that digital technologies can have on the building industry, the potential found throughout this thesis should be further explored.

First, this thesis' computational backbone is currently limited to one type of rationalization: a quad mesh based on orthogonal lines. To increase stability and allow for a wider application of the algorithm more rationalization methods should be supported. Although the researched literature supports that quad meshes are the preferred option, the choice for a specific rationalization can also be made on other factors such as visual appearance and such a choice should be facilitated by the algorithm. Additionally, literature on more optimised geometry rationalization techniques should be considered as it simplifies the optimisation that can currently be unstable. Additionally, supporting more rationalizations will require the node and beam system to work with different valence meshes, requiring more variety in node designs.

In chapter 4 it was illustrated that the facade industry has many options for constructing freeform shapes and all these systems have different characteristics, appearances and strength. One system might be preferred over another for any of these parameters. A broader market review is needed to survey the differences between the system and so inform design variants of the node and beam system. In addition, the most important limitation of the current node and beam system is the non-researched integration of facade panels.

As these panels will probably have widely varying internal parameters, future research into their reuse is needed. Furthermore, the development of a fully functional facade system compatible with the variable nodes should be researched. A crucial aspect of the node design is whether the strength of the friction lock is sufficient to withstand the forces of a facade. Future research should apply extensive strength testing to prove and optimise the design of the nodes. For the system to be able to adapt to any shape, a node type with a torsion axis should be designed.

Chapter 5 not only outlined the state of the art in currently implemented robotic construction solutions, such as tracked and sensing robot arms, it also discussed the future potential for robotic construction. While some research does focus on specific elements, a comprehensive approach of all future possibilities in robotic construction has not yet been pursued. A promising future of construction robotics could be an optimised collaboration between many robots with different tasks that might use the building itself for locomotion. Future research should be able to extend the capabilities of system integrated robotic movement, swarm-like communication and heterogeneous multi-robot collaboration through high-level task planning. Future research should also focus on improving the accessibility of software for robotics and on having a more robust framework for robot-human collaboration. The faculty of Architecture at the TU Delft currently lacks the specific hard- and software required to achieve a true state of the art level. Future robotics research in this faculty should first focus on sensing and collision-free movement. Multi-disciplinary research collaboration would be a great boost to the local research progress and should be pursued.

8.3. Reflection

How is your graduation topic positioned in the studio?

In my opinion the missions statement of the Building Technology studio can be described as: improving the impact of the built environment and building industry on the health of both the environment and the population through modernisation enabled by digital technologies.". Not only is this taught by paying attention to circularity, emissions and efficiency on a building scale, research is also focused on the relation between the modernising built environment and its users to ensure a future that is not only green but also healthy.

This research into automatic construction of reusable building components fits well within the environmental intent of the mission statement. Complexly shaped high end architecture, while often serving an exemplary role, has always been in a disconnect to the overall mission of sustainability. With complex geometries and discrete building components circularity has never been a consideration within many high end projects. By focusing specifically on the circularity of a novel node and beam system, this thesis shows that even the most complex architecture can take a step closer to the contemporary environmental reality.

Not only does this thesis discuss circularity of variable building components, more importantly it delves into the current state and the future potential of automatic building processes using robotics. Coined in the 80s in Japan, the 3D's: Dirty, Dangerous and Demanding describe jobs that are considered ripe for automation during the robotics revolution of the late 20th century, many jobs in the construction industry fall within these criteria. While the robotic revolution has proven itself undoubtedly in a controlled factory environment, it has just been recently that concrete steps have been taken to enable robotic automation in an uncontrolled environment such as a building site. Within this thesis I learned from, educated on and philosophised about the most advanced current technologies and the future potential within robotic construction in order to reduce the human cost and increase the efficiency of the building sector, moving closer to the future of the BT mission statement.

How did your research approach work out? And did it lead to the results you aimed for?

Since this research delves quite deeply into very recent developments in the world of robotics, one of the main challenges coming from a background in building technology has been to become proficient in understanding and implementing these recent technologies. Where understanding of broad concepts and developing an idea of future potential has been mainly lead by an extensive literature review, the actual implementation of robotic technologies can be seen as the exact opposite of scientific literature. Documentation often not being up to date and useful explanations being few and far between made the independent learning process quite arduous. Easier access to these cutting edge tools would definitely give research on the use of advanced robotics a significant boost.

Especially the ROS and MoveIt software was difficult. Because the implementation on the available hardware took significantly longer than expected the development of the actual link to Grasshopper was no longer possible in the available time. This is somewhat disappointing as this technological development would be novel, not only for the faculty but also for the robotic construction sector, which was definitely an important goal at the start of this thesis. Be that as it may, I did learn very valuable information and skills during the implementation and development process that significantly informed the further research direction. Therefore I do not see this as wasted time but rather as future potential.

Where the most ideal vision for the future of robotic construction was theoretically described, the practical implementation remained quite limited, especially without the software implementation. This is due to both the limited available hardware and limited time in implementing new tools. The current practical assembly is still far away from being in any way leading in the field. While it is understood that this is not currently within the scope of either this thesis or the faculty, it remains an ideal image that stays in the back of the mind during and after the process of this graduation.

How are research and design related?

Although this research is of a relatively more technical nature than most at this faculty, design processes still play a major role within the entire process. Firstly, chapters 3 to 5 are used to define practical design boundaries, either for the node and beam system or the assembly process. Secondly, the node design has been established through an exploratory design process and improved by iterative design using rapid prototyping. A design based research process is used to define design boundaries, context and intent for a fluid pavilion design. Throughout this entire thesis design and research are codependent, literature research informs design and design quality is researched within the iterative process.

To what extent are the results applicable in practice?

30 years ago it was pretty much unthinkable to see autonomous robots on the building site. Within the current technological scope we have seen robotic experiments function within unexpected environments, we have seen groups of multiple robots cooperate to achieve higher level tasks and we have seen imaging techniques and safety protocols be developed to cooperate closer with human workers. Suddenly the thought of seeing autonomous robots on building sites is not too far off. While the results of my specific implementation on the specific technology available to me is not directly applicable in practice, it is my firm believe that we will see the culmination of this research field come to practice within the next 30 years.

How does the project affect architecture / the built environment?

The usage of computational design for architectural implementation has significantly grown over the past years. This research offers a computational design workflow to optimise complex architectural shapes to increase their constructability. A variable node and beam system for façade construction of these complex shapes, moves high end architecture closer to the sustainable future of the built environment. Proving the potential of automatic (re)assembly using robots not only influences the efficiency of the building industry, it also inspires an architectural vision of continuously changeable fluid architecture. When changes are automated and fast, architecture has the potential to respond not only to factors present during the planning phase, but also to the changing factors of the built environment through its lifetime.

References

[1] McKinsey Global Institute. *Reinventing construction a route to higher productivity*. Tech. rep. 2017. URL: https://www.mckinsey.com/~/media/mckinsey/business%20functions/operations/our%20insights/reinventing%20construction%20through%20a%20productivity%20revolution/mgi-reinventing-construction-a-route-to-higher-productivity-full-report.pdf.

[2] Anil Sawhney, Mike Riley, and Javier Irizarry. *Construction 4.0*. Feb. 2020. ISBN: 9780429398100. DOI: 10.1201/9780429398100-1.

[3] Kazimierz Butelski. "The Architecture of Curved Shapes". In: *Nexus Network Journal 2000 2:1* 2.1 (June 2000), pp. 19–23. ISSN: 1522-4600. DOI: 10.1007/S00004-999-0004-X. URL: https://link.springer.com/article/10.1007/s00004-999-0004-x.

[4] Gabriel Luiz Fritz Benachio, Maria do Carmo Duarte Freitas, and Sergio Fernando Tavares. "Circular economy in the construction industry: A systematic literature review". In: *Journal of Cleaner Production* 260 (July 2020), p. 121046. ISSN: 0959-6526. DOI: 10.1016/J.JCLEPRO.2020.121046.

[5] Nathan Melenbrink, Katja Rinderspacher, Achim Menges, and Justin Werfel. "Autonomous anchoring for robotic construction". In: *Automation in Construction* 120 (Dec. 2020). ISSN: 09265805. DOI: 10.1016/J.AUTCON.2020.103391.

[6] Helmut Pottmann, Michael Eigensatz, Amir Vaxman, and Johannes Wallner. "Architectural geometry". In: *Computers & Graphics* 47 (Apr. 2015), pp. 145–164. ISSN: 0097-8493. DOI: 10.1016/J.CAG.2014.11.002.

[7] Keenan Crane and Max Wardetzky. "A Glimpse into Discrete Differential Geometry". In: *Notices of the American Mathematical Society* 64.10 (Nov. 2017), pp. 1153–1159. ISSN: 0002-9920. DOI: 10.1090/NOTI1578.

[8] Romain Mesnil, Cyril Douthe, Olivier Baverel, Bruno Léger, and Jean François Caron. "Isogonal moulding surfaces: A family of shapes for high node congruence in free-form structures". In: *Automation in Construction* 59 (Nov. 2015), pp. 38–47. ISSN: 09265805. DOI: 10.1016/j.autcon.2015.07.009.

[9] Xavier Tellier, Laurent Hauswirth, Cyril Douthe, and Olivier Baverel. "Discrete CMC surfaces for doubly-curved building envelopes". In: *Advances in Architectural Geometry* 2018 (2018). URL: https://hal.science/hal-01984201%20https://hal.science/hal-01984201/document.

[10] Helmut Pottman, Andreas Asperl, Michael Hofer, and Axel Kilian. *Architectural Geometry*. Ed. by Daril Bentley. First. Exton, PA: Bentley Institute Press, 2007.

[11] Johannes Wallner and Helmut Pottmann. "Geometric Computing for Freeform Architecture". In: *Journal of Mathematics in Industry* 1.1 (2011), pp. 1–19. ISSN: 21905983. DOI: 10.1186/2190-5983-1-4.

[12] Mirko Zadravec, Alexander Schiftner, and Johannes Wallner. "Designing quad-dominant meshes with planar faces". In: *Eurographics Symposium on Geometry Processing* 29.5 (2010), pp. 1671–1679. ISSN: 17278384.

[13] Caigui Jiang, Cheng Wang, Xavier Tellier, Johannes Wallner, and Helmut Pottmann. "Planar Panels and Planar Supporting Beams in Architectural Structures". In: *ACM Transactions on Graphics* 42.2 (Nov. 2022). ISSN: 15577368. DOI: 10.1145/3561050.

[14] Yang Liu, Helmut Pottmann, Johannes Wallner, Yong Liang Yang, and Wenping Wang. "Geometric modeling with conical meshes and developable surfaces". In: *ACM SIGGRAPH 2006 Papers, SIGGRAPH '06* (2006), pp. 681–689. DOI: 10.1145/1179352.1141941.

[15] Helmut Pottmann, Yang Liu, Johannes Wallner, Alexander Bobenko, and Wenping Wang. "Geometry of multi-layer freeform structures for architecture". In: *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics* (2007). DOI: 10.1145/1275808.1276458.

[16] Wenping Wang, Johannes Wallner, and Yang Liu. "An angle criterion for conical mesh vertices". In: *Journal for Geometry and Graphics* 11.2 (2007), pp. 199–208. ISSN: 14338157.

[17] Helmut Pottmann and Johannes Wallner. "The focal geometry of circular and conical meshes". In: *Advances in Computational Mathematics* 29.3 (Oct. 2008), pp. 249–268. ISSN: 10197168. DOI: 10.1007/S10444-007-9045-4.

[18] Daniel Piker. "Kangaroo: Form finding with computational physics". In: *Architectural Design* 83.2 (Mar. 2013), pp. 136–137. ISSN: 00038504. DOI: 10.1002/AD.1569.

[19] Christos Passas. "The Opus Building Dubai: Zaha Hadid Architects". In: *UAP Conference*. Dubai: UAP Dubai Chapter, Oct. 2020. URL: https://www.youtube.com/watch?v=jFJIfCDcPKc.

[20] Yang Liu, Weiwei Xu, Baining Guo, Jun Wang, Falai Chen, Lifeng Zhu, and Guoping Wang. "General Planar Quadrilateral Mesh Design Using Conjugate Direction Field". In: *ACM Transactions on Graphics* 30.6 (Dec. 2011), pp. 1–10. ISSN: 15577368. DOI: 10.1145/2070781.2024174.

[21] Stefan Sechelmann, Thilo Rörig, and Alexander I. Bobenko. "Quasiisothermic Mesh Layout". In: *Advances in Architectural Geometry 2012* (2013), pp. 243–258. DOI: 10.1007/978-3-7091-1251-9[_]20. URL: https://link.springer.com/chapter/10.1007/978-3-7091-1251-9_20.

[22] Tillmann Klein. *Integral Facade Construction: Towards a new product architecture for curtain walls*. TU Delft Open, June 2013. ISBN: 9789461861610.

[23] Sören Stephan, Jaime Sánchez-Alvarez, and Klaus Knebel. "Stabwerke auf Freiformflächen". In: *Stahlbau* 73.8 (Aug. 2004), pp. 562–572. ISSN: 00389145. DOI: 10.1002/STAB.200490149.

[24] Wesley L. Stone. "The History of Robotics". In: *Robotics and Automation Handbook*. Ed. by Thomas R. Kurfess. Florida: CRC Press, 2005. Chap. 1, pp. 9–20. ISBN: 0849318041.

[25] A. Gasparetto and L. Scalera. "From the unimate to the delta robot: The early decades of industrial robotics". In: *History of Mechanism and Machine Science* 37 (2019), pp. 284–295. ISSN: 18753426. DOI: 10.1007/978-3-030-03538-9[_]23.

[26] M. M. Haque and Ahmad F. Ismail. "Automation in foundry/casting industry". In: *Proceedings of the IEEE International Conference on Industrial Technology* 2 (2002), pp. 815–820. DOI: 10.1109/ICIT.2002.1189272.

[27] Tobias Bonwetsch, Daniel Kobel, Fabio Gramazio, and Matthias Kohler. "The informed wall: Applying additive digital fabrication techniques on architecture". In: *Synthetic Landscapes - ACADIA 2006 International Conference* (2006), pp. 489–495. DOI: 10.52842/conf.acadia.2006.489.

[28] Boris Belousov, Bastian Wibranek, Jan Schneider, Tim Schneider, Georgia Chalvatzaki, Jan Peters, and Oliver Tessmann. "Robotic architectural assembly with tactile skills: Simulation and optimization". In: *Automation in Construction* 133 (2022), pp. 926–5805. DOI: 10.1016/j.autcon.2021.104006. URL: https://doi.org/10.1016/j.autcon.2021.104006.

[29] Hajime Kuwahara, Yutaka Ono, Mitsuhiro Nikaido, and Takaharu Matsumoto. "PRECISION DIRECT-DRIVE ROBOT ARM." In: *Proceedings of the American Control Conference* (1985), pp. 722–727. ISSN: 07431619.

[30] Donald L. Peiper. *The Kinematics of Manipulators under Computer Control*. 1968.

[31] Kathrin Dörfler, Timothy Sandy, Markus Giftthaler, Fabio Gramazio, Matthias Kohler, and Jonas Buchli. "Mobile Robotic Brickwork". In: *Robotic Fabrication in Architecture, Art and Design 2016* (2016), pp. 204–217. DOI: 10.1007/978-3-319-26378-6[_]15.

[32] Deena El-Mahdy and Ahmed Alaa. "An experimental study of a curved brick wall using Robot assembly as a teaching tool in architectural curriculum". In: *International Conference on Electrical, Computer, and Energy Technologies, ICECET 2022* (2022). DOI: 10.1109/ICECET55527.2022.9872791.

[33] FBR. *Hadrian X: Precision Construction Robot*. 2023. URL: https://www.fbr.com.au/view/hadrian-x.

[34] Construction Robotics. *SAM: Semi Automated Mason*. 2023. URL: https://www.construction-robotics.com/sam-2/.

[35] Jianzhuang Xiao, Guangchao Ji, Yamei Zhang, Guowei Ma, Viktor Mechtcherine, Jinlong Pan, Li Wang, Tao Ding, Zhenhua Duan, and Shupeng Du. "Large-scale 3D printing concrete technology: Current status and future opportunities". In: *Cement and Concrete Composites* 122 (Sept. 2021), p. 104115. ISSN: 0958-9465. DOI: 10.1016/J.CEMCONCOMP.2021.104115.

[36] Stefan C. Figueiredo, Anne L.van Overmeir, Karsten Nefs, Erik Schlangen, Theo A.M. Salet, Branko Šavija, Akke S.J. Suiker, and Freek P. Bos. "Quality Assessment of Printable Strain Hardening Cementitious Composites Manufactured in Two Different Printing Facilities". In: *RILEM Bookseries* 28 (2020), pp. 824–838. ISSN: 22110852. DOI: 10.1007/978-3-030-49916-7[_]81.

[37] Advanced Construction Robotics. *TyBOT | The Rebar Tying Robot*. 2023. URL: https://www.constructionrobots.com/tybot.

[38] SAM XL. *STUNNING: Thermoplastic composite fuselage assembly*. 2023. URL: https://www.samxl.com/stunning.

[39] Swapnil Patil, V. Vasu, and K. V. S. Srinadh. "Advances and perspectives in collaborative robotics: a review of key technologies and emerging trends". In: *Discover Mechanical Engineering 2023 2:1* 2.1 (Aug. 2023), pp. 1–19. ISSN: 2731-6564. DOI: 10.1007/S44245-023-00021-8. URL: https://link-springer-com.tudelft.idm.oclc.org/article/10.1007/s44245-023-00021-8.

[40] Nathan Melenbrink, Paul Kassabian, Achim Menges, and Justn Werfel. "Towards force-aware robot collectives for on-site construction". In: *Disciplines and Disruption - Proceedings Catalog of the 37th Annual Conference of the Association for Computer Aided Design in Architecture, ACADIA 2017* (2017), pp. 382–391. DOI: 10.52842/CONF.ACADIA.2017.382. URL: https://www.researchgate.net/publication/329591348_Towards_Force-aware_Robot_Collectives_for_On-site_Construction.

[41] Lydia E. Kavraki, Petr Švestka, Jean Claude Latombe, and Mark H. Overmars. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces". In: *IEEE Transactions on Robotics and Automation* 12.4 (1996), pp. 566–580. ISSN: 1042296X. DOI: 10.1109/70.508439.

[42] Nancy M. Amato and Yan Wu. "Randomized roadmap method for path and manipulation planning". In: *Proceedings - IEEE International Conference on Robotics and Automation* 1 (1996), pp. 113–120. ISSN: 10504729. DOI: 10.1109/ROBOT.1996.503582.

[43] James J. Kuffner and Steven M. La Valle. "RRT-connect: an efficient approach to single-query path planning". In: *Proceedings - IEEE International Conference on Robotics and Automation* 2 (2000), pp. 995–1001. ISSN: 10504729. DOI: 10.1109/ROBOT.2000.844730.

[44] Marcelo Kallmann, Amaury Aubel, Tolga Abaci, and Daniel Thalmann. "Planning Collision-Free Reaching Motions for Interactive Object Manipulation and Grasping". In: *Computer Graphics Forum* 22.3 (2003), pp. 313–322. ISSN: 01677055. DOI: 10.1111/1467-8659.00678.

[45] Nathan Ratliff, Matt Zucker, J. Andrew Bagnell, and Siddhartha Srinivasa. "CHOMP: Gradient optimization techniques for efficient motion planning". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2009), pp. 489–494. ISSN: 10504729. DOI: 10.1109/ROBOT.2009.5152817.

[46] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. "STOMP: Stochastic trajectory optimization for motion planning". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2011), pp. 4569–4574. ISSN: 10504729. DOI: 10.1109/ICRA.2011.5980280.

[47] Yizhou Liu, Fusheng Zha, Mantian Li, Wei Guo, Yunxin Jia, Pengfei Wang, Yajing Zang, and Lining Sun. "Creating Better Collision-Free Trajectory for Robot Motion Planning by Linearly Constrained Quadratic Programming". In: *Frontiers in Neurorobotics* 15 (Aug. 2021). ISSN: 16625218. DOI: 10.3389/FNBOT.2021.724116.

[48] Yijiang Huang, Pok Yin Victor Leung, Caelan Garrett, Fabio Gramazio, Matthias Kohler, and Caitlin Mueller. "The new analog: A protocol for linking design and construction intent with algorithmic planning for robotic assembly of complex structures". In: *Proceedings - SCF 2021: ACM Symposium on Computational Fabrication* (Oct. 2021). DOI: 10.1145/3485114.3485122.

[49] Jiangpeng Shu, Wenhao Li, and Yifan Gao. "Collision-free trajectory planning for robotic assembly of lightweight structures". In: *Automation in Construction* 142 (Oct. 2022). ISSN: 09265805. DOI: 10.1016/j.autcon.2022.104520.

[50] Qi Zhu, Tianyu Zhou, Pengxiang Xia, and Jing Du. "Robot Planning for Active Collision Avoidance in Modular Construction: Pipe Skids Example". In: *Journal of Construction Engineering and Management* 148.10 (Oct. 2022). ISSN: 0733-9364. DOI: 10.1061/(asce)co.1943-7862.0002374.

[51] Ziju Xian, Nicholas Hoban, and Brady Peters. "Spatial Timber Assembly Robotically Fabricated Reciprocal Frame Wall". In: *Proceedings of the International Conference on Education and Research in Computer Aided Architectural Design in Europe* 2 (2020), pp. 403–412. ISSN: 26841843. DOI: 10.52842/conf.ecaade.2020.2.403.

[52] Edvard P.G. Bruun, Sigrid Adriaenssens, and Stefana Parascho. "Structural rigidity theory applied to the scaffold-free (dis)assembly of space frames using cooperative robotics". In: *Automation in Construction* 141 (Sept. 2022). ISSN: 09265805. DOI: 10.1016/j.autcon.2022.104405.

[53] Nick Bertram, Steffen Fuchs, Jan Mischke, Robert Palter, Gernot Strube, and Jonathan Woetzel. *Modular construction: From projects to products by McKinsey & Company*. Tech. rep. June 2019. URL: https://www.mckinsey.com/capabilities/operations/our-insights/modular-construction-from-projects-to-products.

[54] Jürgen Andres, Thomas Bock, Friedrich Gebhart, and Werner Steck. "First Results of the Development of the Masonry Robot System ROCCO: a Fault Tolerant Assembly Tool". In: *Automation and Robotics in Construction XI*. Amsterdam: Elsevier Science B.V., 1994. Chap. 1C, pp. 87–95.

[55] G. Pritschow, M. Dalacker, J. Kurz, and M. Gaenssle. "Technological aspects in the development of a mobile bricklaying robot". In: *Automation in Construction* 5.1 (Mar. 1996), pp. 3–13. ISSN: 0926-5805. DOI: 10.1016/0926-5805(95)00015-1.

[56] Volker Helm, Selen Ercan, Fabio Gramazio, and Matthias Kohler. "Mobile robotic fabrication on construction sites: DimRob". In: *IEEE International Conference on Intelligent Robots and Systems* (2012), pp. 4335–4341. ISSN: 21530858. DOI: 10.1109/IROS.2012.6385617.

[57] Chen Feng, Yong Xiao, Aaron Willette, Wes McGee, and Vineet R. Kamat. "Vision guided autonomous robotic assembly and as-built scanning on unstructured construction sites". In: *Automation in Construction* 59 (Nov. 2015), pp. 128–138. ISSN: 0926-5805. DOI: 10.1016/J.AUTCON.2015.06.002.

[58] Hua Chai, Hans Jakob Wagner, Zhixian Guo, Yue Qi, Achim Menges, and Philip F. Yuan. "Computational design and on-site mobile robotic construction of an adaptive reinforcement beam network for cross-laminated timber slab panels". In: *Automation in Construction* 142 (Oct. 2022), p. 104536. ISSN: 0926-5805. DOI: 10.1016/J.AUTCON.2022.104536.

[59] Mathew Schwartz, Yong Geng, Hakam Agha, Rijeesh Kizhakidathazhath, Danqing Liu, Gabriele Lenzini, and Jan P.F. Lagerwall. "Linking physical objects to their digital twins via fiducial markers designed for invisibility to humans". In: *Multifunctional Materials* 4.2 (June 2021). ISSN: 23997532. DOI: 10.1088/2399-7532/AC0060.

[60] Darsh Parekh, Nishi Poddar, Aakash Rajpurkar, Manisha Chahal, Neeraj Kumar, Gyanendra Prasad Joshi, and Woong Cho. "A Review on Autonomous Vehicles: Progress, Methods and Challenges". In: *Electronics 2022, Vol. 11, Page 2162* 11.14 (July 2022), p. 2162. ISSN: 2079-9292. DOI: 10.3390/ELECTRONICS11142162. URL: https://www.mdpi.com/2079-9292/11/14/2162/htm%20https://www.mdpi.com/2079-9292/11/14/2162.

[61] Srijeet Halder, Kereshmeh Afsari, Erin Chiou, Rafael Patrick, and Kaveh Akbari Hamed. "Construction inspection & monitoring with quadruped robots in future human-robot teaming: A preliminary study". In: *Journal of Building Engineering* 65 (Apr. 2023), p. 105814. ISSN: 2352-7102. DOI: 10.1016/J.JOBE.2022.105814.

[62] JeeWoong Park, Yong K. Cho, and Diego Martinez. "A BIM and UWB integrated Mobile Robot Navigation System for Indoor Position Tracking Applications". In: *Journal of Construction Engineering and Project Management* 6.2 (June 2016), pp. 30–39. ISSN: 2233-9582. DOI: 10.6106/JCEPM.2016.6.2.030. URL: www.jcepm.org.

[63] Youngjib Ham, Kevin K. Han, Jacob J. Lin, and Mani Golparvar-Fard. "Visual monitoring of civil infrastructure systems via camera-equipped Unmanned Aerial Vehicles (UAVs): a review of related works". In: *Visualization in Engineering* 4.1 (Dec. 2016), pp. 1–8. ISSN: 22137459. DOI: 10.1186/S40327-015-0029-Z/FIGURES/3. URL: https://link-springer-com.tudelft.idm.oclc.org/articles/10.1186/s40327-015-0029-z%20https://link-springer-com.tudelft.idm.oclc.org/article/10.1186/s40327-015-0029-z.

[64] Lu Wang, Tao Zhang, Hiroatsu Fukuda, and Yi Leng. "Research on the Application of Mobile Robot in Timber Structure Architecture". In: *Sustainability (Switzerland)* 14.8 (Apr. 2022). ISSN: 20711050. DOI: 10.3390/su14084681.

[65] A. McClymonds, R. Leicht, and S. Asadi. "System Architecture for Supporting BIM to Robotic Construction Integration". In: *Proceedings of the Canadian Society of Civil Engineering Annual Conference 2021, CSCE21 Construction Track*. Vol. 2. 2023, pp. 225–236. DOI: 10.1007/978-981-19-0968-9{_}18.

[66] Sebastian Castro. *Task Planning in Robotics - Robotic Sea Bass*. July 2022. URL: https://roboticseabass.com/2022/07/19/task-planning-in-robotics/.

[67] Stéphane Cambon, Rachid Alami, and Fabien Gravot. "A hybrid approach to intricate motion, manipulation and task planning". In: *International Journal of Robotics Research* 28.1 (Jan. 2009), pp. 104–126. ISSN: 02783649. DOI: 10.1177/0278364908097884.

[68] Yu qian Jiang, Shi qi Zhang, Piyush Khandelwal, and Peter Stone. "Task Planning in Robotics: an Empirical Comparison of PDDL-based and ASP-based Systems". In: *Frontiers of Information Technology and Electronic Engineering* 20.3 (Apr. 2018), pp. 363–373. ISSN: 20959230. DOI: 10.1631/FITEE.1800514. URL: https://arxiv.org/abs/1804.08229v3.

[69] Michael Gorner, Robert Haschke, Helge Ritter, and Jianwei Zhang. "Moveit! task constructor for task-level motion planning". In: *Proceedings - IEEE International Conference on Robotics and Automation* 2019-May (May 2019), pp. 190–196. ISSN: 10504729. DOI: 10.1109/ICRA.2019.8793898.

[70] Julien Bidot, Lars Karlsson, Fabien Lagriffoul, and Alessandro Saffiotti. "Geometric backtracking for combined task and motion planning in robotic systems". In: *Artificial Intelligence* 247 (June 2017), pp. 229–265. ISSN: 0004-3702. DOI: 10.1016/J.ARTINT.2015.03.005.

[71] George Konidaris, Leslie Pack Kaelbling, and Tomas Lozano-Perez. "Constructing symbolic representations for high-level planning". In: *Proceedings of the National Conference on Artificial Intelligence* 3 (2014), pp. 1932–1938. ISSN: 2159-5399. DOI: 10.1609/aaai.v28i1.9004.

[72] León Illanes, Xi Yan, Rodrigo Toro Icarte, and Sheila A. McIlraith. "Symbolic plans as high-level instructions for reinforcement learning". In: *Proceedings International Conference on Automated Planning and Scheduling, ICAPS* 30 (May 2020), pp. 540–550. ISSN: 23340843. DOI: 10.1609/icaps.v30i1.6750.

[73] Gregory P. Davis, Garrett E. Katz, Rodolphe J. Gentili, and James A. Reggia. "NeuroLISP: High-level symbolic programming with attractor neural networks". In: *Neural Networks* 146 (Feb. 2022), pp. 200–219. ISSN: 0893-6080. DOI: 10.1016/J.NEUNET.2021.11.009.

[74] Marc Hanheide, Moritz Göbelbecker, Graham S. Horn, Andrzej Pronobis, Kristoffer Sjöö, Alper Aydemir, Patric Jensfelt, Charles Gretton, Richard Dearden, Miroslav Janicek, Hendrik Zender, Geert Jan Kruijff, Nick Hawes, and Jeremy L. Wyatt. "Robot task planning and explanation in open and uncertain worlds". In: *Artificial Intelligence* 247 (June 2017), pp. 119–150. ISSN: 0004-3702. DOI: 10.1016/J.ARTINT.2015.08.008.

[75] Nathan Melenbrink, Panagiotis Michalatos, Paul Kassabian, and Justin Werfel. "Using local force measurements to guide construction by distributed climbing robots". In: *IEEE International Conference on Intelligent Robots and Systems* 2017-September (Dec. 2017), pp. 4333–4340. ISSN: 21530866. DOI: 10.1109/IROS.2017.8206298.

[76] Grzegorz Lochnicki, Nicolas Kubail Kalousdian, Samuel Leder, Mathias Maierhofer, Dylan Wood, and Achim Menges. "Co-Designing Material-Robot Construction Behaviors: Teaching distributed robotic systems to leverage active bending for light-touch assembly of bamboo bundle structures". In: *Association for Computer Aided Design in Architecture Annual Conference, ACADIA 2021* (2021).

[77] Burak Delikanlı and Leman Figen Gül. "Towards to the Hyperautomation: An integrated framework for Construction 4.0: A case of Hookbot as a distributed reconfigurable robotic assembly system". In: *Proceedings of the International Conference on Education and Research in Computer Aided Architectural Design in Europe* 2 (2022), pp. 389–398. ISSN: 26841843. DOI: 10.52842/CONF.ECAADE.2022.2.389. URL: https://www.researchgate.net/publication/364227101_Towards_to_the_Hyperautomation_-_An_Integrated_Framework_for_Construction_40_a_Case_of_Hookbot_as_a_Distributed_Reconfigurable_Robotic_Assembly_System.

[78] Samuel Leder, Hyun Gyu Kim, Ozgur Salih Oguz, Nicolas Kubail Kalousdian, Valentin Noah Hartmann, Achim Menges, Marc Toussaint, and Metin Sitti. "Leveraging Building Material as Part of the In-Plane Robotic Kinematic System for Collective Construction". In: *Advanced Science* 9.24 (Aug. 2022). ISSN: 21983844. DOI: 10.1002/advs.202201524.

[79] *ESA - European Robotic Arm*. URL: https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/International_Space_Station/European_Robotic_Arm.

[80] Burak Delikanlı and Leman Figen Gül. "A System for Truss Manipulation with Relative Robots: Designing and Prototyping HookBot". In: *Communications in Computer and Information Science* 1819 CCIS (2023), pp. 393–409. ISSN: 18650937. DOI: 10.1007/978-3-031-37189-9[_]26/FIGURES/12. URL: https://link-springer-com.tudelft.idm.oclc.org/chapter/10.1007/978-3-031-37189-9_26.

[81] R. Koeppe, D. Engelhardt, A. Hagenauer, P. Heiligensetzer, B. Kneifel, A. Knipfer, and K. Stoddard. "Robot-robot and human-robot cooperation in commercial robotics applications". In: *Springer Tracts in Advanced Robotics* 15 (2005), pp. 202–216. ISSN: 1610742X. DOI: 10.1007/11008941[_]22/COVER. URL: https://link-springer-com.tudelft.idm.oclc.org/chapter/10.1007/11008941_22.

[82] Heiko Hamann. "Swarm robotics: A formal approach". In: *Swarm Robotics: A Formal Approach* (Feb. 2018), pp. 1–210. DOI: 10.1007/9783319745282.

[83] David Andréen, Petra Jenning, Nils Napp, and Kirstn Petersen. "Emergent structures assembled by large swarms of simple robots". In: *ACADIA 2016: Posthuman Frontiers: Data, Designers, and Cognitive Machines - Proceedings of the 36th Annual Conference of the Association for Computer Aided Design in Architecture* (2016), pp. 54–61. DOI: 10.52842/conf.acadia.2016.054.

[84] Justin Werfel. "Collective Construction with Robot Swarms". In: *Morphogenetic Engineering* (2012), pp. 115–140. ISSN: 1860-0840. DOI: 10.1007/978-3-642-33902-8[_]5. URL: https://link-springer-com.tudelft.idm.oclc.org/chapter/10.1007/978-3-642-33902-8_5.

[85] Teshan Liyanage and Subha Fernando. "Optimizing Robotic Swarm Based Construction Tasks". In: *2021 7th International Conference on Control, Automation and Robotics, ICCAR 2021* (Apr. 2021), pp. 89–92. DOI: 10.1109/ICCAR52225.2021.9463439.

[86] S ; Lin, A ; Liu, J ; Wang, X Kong, Shiwei Lin, Ang Liu, Jianguo Wang, and Xiaoying Kong. "A Review of Path-Planning Approaches for Multiple Mobile Robots". In: *Machines 2022, Vol. 10, Page 773* 10.9 (Sept. 2022), p. 773. ISSN: 2075-1702. DOI: 10.3390/MACHINES10090773. URL: https://www.mdpi.com/2075-1702/10/9/773/htm%20https://www.mdpi.com/2075-1702/10/9/773.

[87] Atsushi Yamashita, Tamio Arai, Jun Ota, and Hajime Asama. "Motion planning of multiple mobile robots for cooperative manipulation and transportation". In: *IEEE Transactions on Robotics and Automation* 19.2 (Apr. 2003), pp. 223–237. ISSN: 1042296X. DOI: 10.1109/TRA.2003.809592.

[88] Aliakbar Akbari, Muhayyuddin, and Jan Rosell. "Knowledge-oriented task and motion planning for multiple mobile robots". In: *Journal of Experimental & Theoretical Artificial Intelligence* 31.1 (Jan. 2019), pp. 137–162. ISSN: 13623079. DOI: 10.1080/0952813X.2018.1544280. URL: https://www-tandfonline-com.tudelft.idm.oclc.org/doi/abs/10.1080/0952813X.2018.1544280.

[89] Anja Kunic, Roberto Naboni, Aljaz Kramberger, and Christian Schlette. "Design and assembly automation of the Robotic Reversible Timber Beam". In: *Automation in Construction* 123 (Mar. 2021). ISSN: 09265805. DOI: 10.1016/J.AUTCON.2020.103531.

[90] Dominic Jud, Simon Kerscher, Martin Wermelinger, Edo Jelavic, Pascal Egli, Philipp Leemann, Gabriel Hottiger, and Marco Hutter. "HEAP - The autonomous walking excavator". In: *Automation in Construction* 129 (Sept. 2021), p. 103783. ISSN: 0926-5805. DOI: 10.1016/J.AUTCON.2021.103783.

[91] Mark Burkhardt and Oliver Sawodny. "Towards Modeling and Control of a Crane-Collaboration for the Automated Assembly of Timber Structures". In: *IECON Proceedings (Industrial Electronics Conference)* 2021-October (Oct. 2021). DOI: 10.1109/IECON48115.2021.9589787.

[92] Rubin Luo, Hongxing Zheng, and Jifeng Guo. "Solving the Multi-Functional Heterogeneous UAV Cooperative Mission Planning Problem Using Multi-Swarm Fruit Fly Optimization Algorithm". In: *Sensors 2020, Vol. 20, Page 5026* 20.18 (Sept. 2020), p. 5026. ISSN: 1424-8220. DOI: 10.3390/S20185026. URL: https://www.mdpi.com/1424-8220/20/18/5026/htm%20https://www.mdpi.com/1424-8220/20/18/5026.

[93] A Lujain. *Yas Viceroy Hotel - 3D Warehouse*. Mar. 2014. URL: https://3dwarehouse.sketchup.com/model/e28b435e4e28e603eef212afc0c5bf28/Yas-Viceroy-Hotel.

[94] Visose. *GitHub - visose/Robots: Create and simulate ABB, KUKA, UR, and Staubli robot programs*. 2023. URL: https://github.com/visose/Robots.

[95] RoboDK. *Simulator for industrial robots and offline programming - RoboDK*. 2023. URL: https://robodk.com/.

[96] PickNik Robotics. *MoveIt Motion Planning Framework*. 2023. URL: https://moveit.ros.org/.

[97] Ioan A. Şucan, Mark Moll, and Lydia Kavraki. "The open motion planning library". In: *IEEE Robotics and Automation Magazine* 19.4 (2012), pp. 72–82. ISSN: 10709932. DOI: 10.1109/MRA.2012.2205651.

[98] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. "ROS: an open-source Robot Operating System". In: *ICRA Workshop on Open Source Software* 3 (Feb. 2009). URL: http://stair.stanford.edu.

[99] Krzysztof Kurc, Andrzej Burghardt, Piotr Gierlak, Magdalena Muszyńska, Dariusz Szybicki, Artur Ornat, and Marek Uliasz. "Application of a 3D Scanner in Robotic Measurement of Aviation Components". In: *Electronics 2022, Vol. 11, Page 3216* 11.19 (Oct. 2022), p. 3216. ISSN: 2079-9292. DOI: 10.3390/ELECTRONICS11193216. URL: https://www.mdpi.com/2079-9292/11/19/3216/htm%20https://www.mdpi.com/2079-9292/11/19/3216.

[100] Lara Davis, Matthias Rippmann, Tom Pawlofsky, and Philippe Block. "Innovative funicular tile vaulting: A prototype vault in Switzerland". In: *Structural Engineer* 90.11 (Nov. 2012), pp. 46–56. ISSN: 14665123.

[101] Pierluigi D'Acunto, Juan Castellón, Alessandro Tellini, and Shibo Ren. "foldKITE: An Ultra-Lightweight Folded Structure". In: Dec. 2015.

[102] Christian Hoenigschmid-DeVeaux. *Folded Plate Structures – Christian Hoenigschmid-DeVeaux – Journal – GRIMSHAW*. URL: https://grimshaw.global/journal/folded-plate-structures/.

[103] Peter Debney, Paul Jeffries, Alex Christodoulou, Rick Titulaer, Paul Shepherd, Sean Lineham, Hannah Lazenby, Stephen Melville, and James Solly. "Advanced Applications in Computational Design". In: *Structural Integrity* 20 (2022), pp. 77–102. ISSN: 25225618. DOI: 10.1007/978-3-030-82430-3[_]4/FIGURES/21. URL: https://link-springer-com.tudelft.idm.oclc.org/chapter/10.1007/978-3-030-82430-3_4.

[104] VTN Architects. *Bamboo Stalactite*. 2018. URL: https://vtnarchitects.net/bamboo-stalactite-pe212.html.

[105] Formlabs. *3D Printing at Scale: The FUSE Pavilion | Formlabs*. 2017. URL: https://formlabs.com/blog/3d-printing-at-scale-fuse-pavilion/.

[106] Jan Knippers, Riccardo La Magna, Achim Menges, Steffen Reichert, Tobias Schwinn, and Frédéric Waimer. "ICD/ITKE Research Pavilion 2012: Coreless Filament Winding Based on the Morphological Principles of an Arthropod Exoskeleton". In: *Architectural Design* 85.5 (Sept. 2015), pp. 48–53. ISSN: 1554-2769. DOI: 10.1002/AD.1953. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/ad.1953%20https://onlinelibrary.wiley.com/doi/abs/10.1002/ad.1953%20https://onlinelibrary.wiley.com/doi/10.1002/ad.1953.

[107] Amy Frearson. *University of Michigan completes Robotically Fabricated Structure*. Aug. 2022. URL: https://www.dezeen.com/2022/08/26/robotically-fabricated-structure-university-of-michigan/.

[108] Gordon Cullen. "THE CONCISE OF TOWNSCAPE". In: (1971). URL: https://books.google.com/books/about/The_Concise_Townscape.html?hl=nl&id=UoUJFju-10AC.