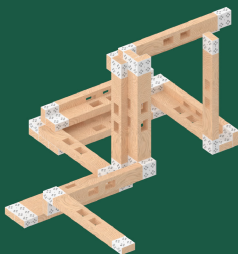
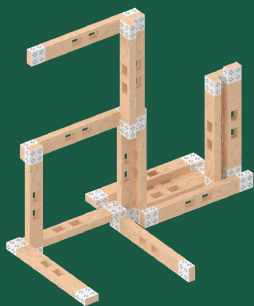
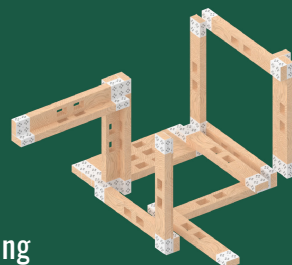
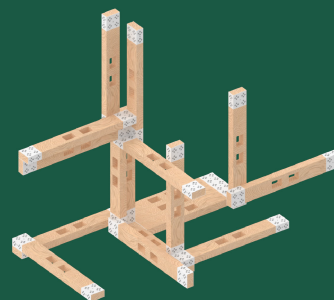
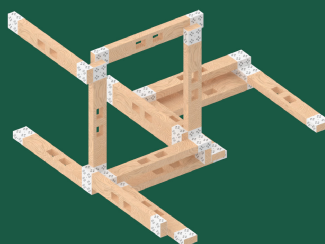
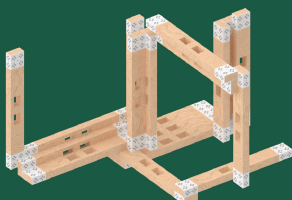
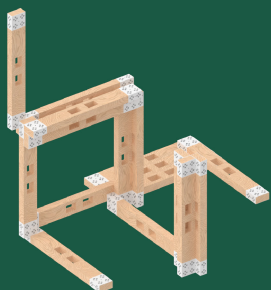
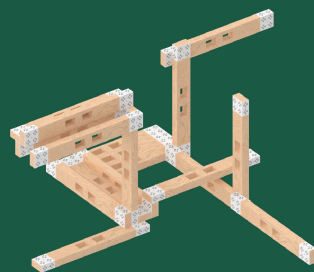


DISCRETE AUTOMATION

robotic construction workflow
for reconfigurable timber housing



Adalberto de Paula
Master Thesis Research
TU Delft
2023



Discrete Automation

robotic construction workflow
for reconfigurable timber housing

Adalberto de Paula

Master Thesis Research
2023

MSc in Architecture, Urbanism and Building Sciences
Building Technology track

Mentors

Dr. Serdar Aşut

Dr. Stijn Brancart



Delft University of Technology
Faculty of Architecture and the Built Environment



Acknowledgment

I would like to express my deepest gratitude and appreciation to all those who have supported and guided me throughout the course of my research and writing of this thesis.

First and foremost, I am immensely grateful to my supervisors, Serdar Aşut and Stijn Brancart, for their unwavering support, guidance, and invaluable insights. Their expertise and encouragement have been instrumental in shaping the direction of this research and helping me navigate the challenges that arose along the way.

I am also indebted to the staff of the Laboratory for Additive Manufacturing in Architecture, coordinated by Paul de Ruiter, for their support and for providing me with a stimulating research environment. The state-of-the-art facilities and collaborative atmosphere have been instrumental in carrying out the experiments and analysis for this thesis.

I would like to extend my gratitude to the Building Technology staff and the TU Delft Faculty of Architecture & the Built Environment. Their dedication to providing a conducive academic environment and access to resources has been vital to the successful completion of this thesis.

I am grateful to my family and friends for their unwavering support, understanding, and encouragement throughout this academic journey. Their belief in my abilities has been a constant source of motivation.

Finally, I would like to acknowledge the financial support provided by the Foundation Justus & Louise van Effen. Their support allowed me to focus solely on my research and concentrate on producing the best possible outcome.

To everyone who has played a role, big or small, in the completion of this thesis, I offer my heartfelt thanks. Your contributions have been invaluable, and I am sincerely grateful for your assistance and encouragement.

It is about time.

The systems of change.

The adaptable architecture.

The circular.

Abstract

The construction industry's impact on carbon emissions, pollution, and resource depletion necessitates innovative approaches to reduce environmental harm. This research explores the use of computational design, digital fabrication, and timber as a renewable material to mitigate the construction industry's environmental impact. Timber is recognized as a low-carbon solution for affordable housing, offering a means to decrease emissions in building construction.

This study presents an innovative automated construction workflow that involves human-robot collaboration (HRC) for a discretized timber construction system. To demonstrate the capabilities of the system, a housing design is developed for a specific location in Rotterdam. The research considers the site context as a guideline to establish boundary conditions for implementing the developed construction system. It addresses the issue of affordable housing, transcending the chosen site context, as it is a global concern. The design incorporates circularity principles, including modularity, design-for-disassembly, design-for-reuse, reconfigurability, and extension of material lifespan. A combinatorial design workflow is proposed, focusing on the assembly of generic discrete elements into function-based aggregated structures that can be rearranged over time.

In order to prove the concept, an HRC assembly prototype is established to mount the discretized aggregation structure, utilizing demountable connections to join the elements while asking the human participation. This approach enables the reassembly of the structure multiple times, promoting material reuse and extending the structure's potential.

The research contributes to the advancement of the circular agenda in the building industry by implementing essential digital design and manufacturing concepts into an automated construction process. By extending the material life cycle and carbon store, the proposed workflow demonstrates the potential for sustainable and efficient construction practices in the timber housing sector.

Key Words: Robotic construction workflow, Robotic assembly, Human-robot collaboration, Timber construction, Discrete architecture, Discrete aggregation, Combinatorial design, Reconfigurability, Circularity, Design-for-disassembly, Design-for-Reuse, Circular Housing, Mass customization housing.

Contents

Acknowledgment	5
Abstract	7
1 Introduction	10
1.1 Means of production	11
1.2 Problem statement	13
1.3 Design goal	17
1.4 Research question	19
1.5 Vision & Scope	20
1.6 Methodology	22
2 Robotic fabrication	24
2.1 Robotic Fabrication in Architecture	25
2.2 Human-Robot Collaboration	27
2.3 HRC state of the art	29
2.4 HRC on construction site	31
2.5 HRC usage evaluation criteria	33
3 Discrete architecture	34
3.1 Discrete Architecture Thinking	35
3.1.1 Discrete aggregation for mass customization	37
3.1.2 Combinatorial Design for reconfigurable housing	39
3.2 Circularity in discrete components	40
3.3 Timber construction	42
3.3.1 Timber characteristics	43
3.4 Discrete design projects	44
3.5 Discrete element typology analysis	46

4 Architectural design	48
4.1 Discrete element design	49
4.1.1 Connection design	53
4.2 Discrete elements aggregations	57
4.3 Generative design approach	62
4.4 The site	64
4.5 Housing platform’s program	68
4.5.1 User’s interface app	70
4.6 Discrete timber housing design	72
5 Robotic construction workflow	88
5.1 Design-to-build	89
5.2 Off-site prefabrication	92
5.3 On-site assembly	94
5.3.1 Human as automated Assembly’s tolerance	94
5.3.2 Workflow evaluation for HRC implementation	95
5.3.3 Robot context-awareness	96
5.3.4 On-site robotic construction workflow	98
5.3.5 On-site assembly sequence	102
6 Prototypes	104
6.1 1:1 connection prototype	105
6.2 1:5 robotic workflow prototype	112
6.2.1 Discrete elements fabrication	113
6.2.2 Prototyping phases	115
6.2.3 Robot experiment setup	115
6.2.4 Robot Simulation and Control	118
6.2.5 HRC assembly prototype	120
6.3 1:10 prototype	128
7 Conclusion	134
7.1 Process conclusion	135
7.2 Reflection	138
7.3 Discussion	140
7.4 Further research	141
References	142
List of Figures	145
Appendix	148

1 Introduction

- 1.1 Means of production
- 1.2 Problem statement
- 1.3 Design goal
- 1.4 Research question
- 1.5 Vision & Scope
- 1.6 Methodology

This master's thesis titled "Discrete Automation - Robotic construction workflow for reconfigurable timber housing" focuses on the development of a novel robotic construction workflow involving human-robot collaboration for a discrete timber housing design in Rotterdam. The design of the discrete element fuses circularity principles of modularity, disassembly, reuse, reconfigurability, and expansion of the material lifespan. The research is part of the requirements for obtaining a master's degree in the Building Technology track in the MSc Architecture, Urbanism, and Building Sciences program at TU Delft. Serdar Asut and Stijn Brancart are the mentors of the thesis, which is related to two chairs of the program named Design Informatics and Structural Design & Mechanics.

This introduction chapter expresses the thesis synopsis with the comprehension of the problem statement that guides the establishment of the design goal and main research questions. The research methodology specifying the thesis phases and methods is outlined in sequence.

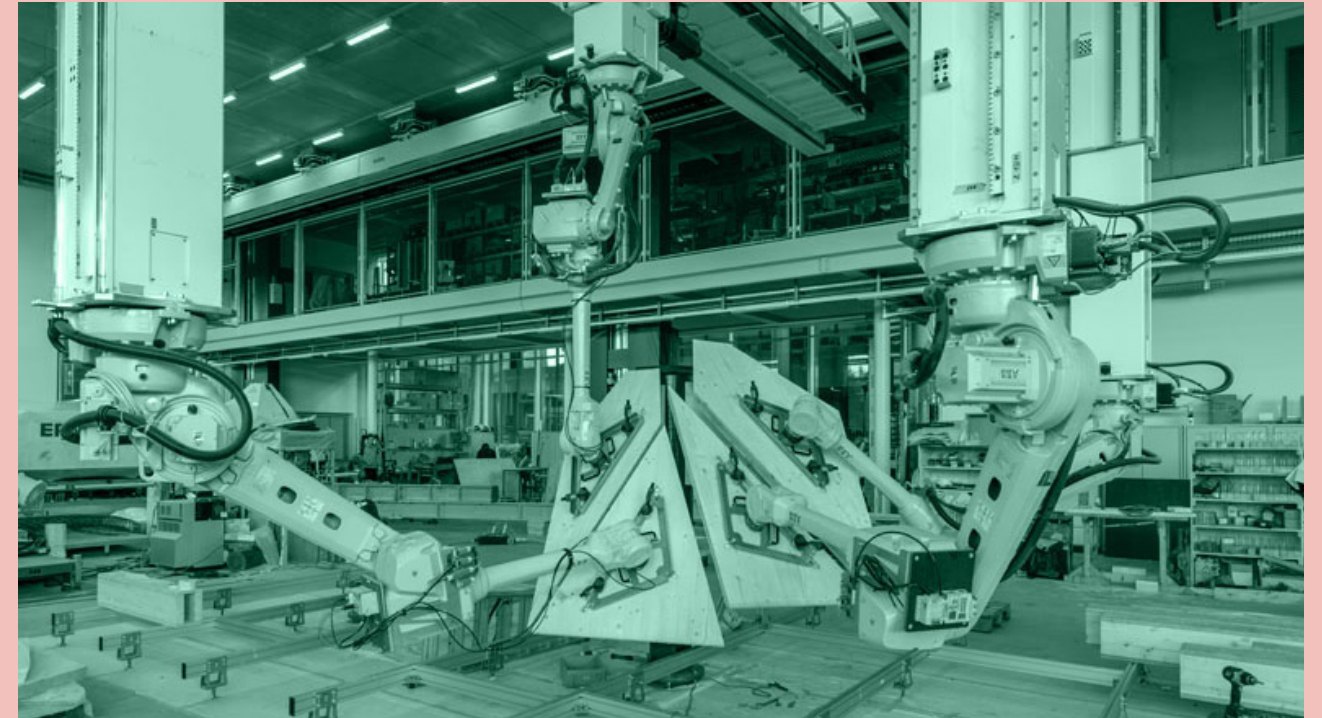


Figure 1. Cross-laminated timber plates robotically assembled with a novel assembly procedure. (Gramazio Kohler Research, ETH Zurich, 2022).

1.1 Means of production

Our ways of production have changed over time since the First Industrial Revolution. From the mechanization of the process and steam power, the beginning of the 20th century brought us the first step in basic automation with the utilization of electrical energy that allowed a higher level of control in the operational process of serial production. This step led to an important increase in productivity through rigorous standardization inside the production line. Afterward in the 70s, productivity was even boosted through more powerful and programmable electronics that also brought flexibility and adaptability to production. The notion of computer-integrated manufacturing entered the industry when the computer was integrated into the production processes. This led to still predominant design methods such as computer-aided design, computer-aided engineering, and computer-aided manufacturing. Originally, this means of production aimed at digital control and automation of traditional manufacturing processes in which the approach has typically been

task-specific (Menges, 2015). However, nowadays hardware has become generic enabling multifunctionality, allowing robots to perform diverse production tasks (Fig. 1).

We are now in the age of the Fourth Industrial Revolution, which embraces processes beyond automation, like cyber-physical systems, the internet of things, and networks. Inside this context, machines and robots are able to communicate, monitor, sense, react and act, creating a higher level of integration and link between the physical and digital domains (Menges, 2015). In this way, the production chain is becoming more and more self-aware being able to predict, configure and organize itself. Thus, this new way of making allows the emergence of new cyber-physical production systems that connect the physical making methods with the digital design realm, for instance, human-robot collaboration (Fig. 2). Gramazio et al. (2014) proposes that within this framework, behavior-driven processes supersede instruction-based construction by integrating design and fabrication.



Figure 2. Collaborative assembly by a team of humans and robots. (Mitterberger, 2022).

We currently reside in an era dominated by digital platforms, wherein individuals who were once consumers have now transformed into producers (Sanchez, 2017). Inside this new culture, the democratization of emergent technologies influenced all scales of production, including architecture. One of the transformations taking place in the building industry is how computational design thinking is changing both the design methods and way of making buildings (Man, 2021). The understanding of production is transitioning from the traditional mass-standardization factory to leaner, smarter, and more flexible methods. The main difference between the Fourth Industrial Revolution from the other three is that its initial goal is not to increase productivity, but instead to higher levels of flexibility, adaptability, and integration (Menges, 2015). Thus, the capability to adapt and evolve is an imminent factor in this new form of production. The desire for mass customization also alter the industry from a centralized model toward a distributed production. This decentralization in the production chain that fuses design and making enables a smarter process, as and provides its optimization. The decentralized production model and real-time fabrication sense and quickly respond to the eventualities of making (Man, 2021). The potential merging point of design and construction derives from when the materi-

alization becomes more computational and generative, and the digital design becomes more physical and procedural (Menges, 2015), for instance, when material properties, connection rules, and assembly order are taken into account in the design phase enabling the generation of its materialization. This combination of processes creates challenges for traditional well-established methods and techniques in both domains. Moreover, this industrial transformation affects the relationship among components in the building environment. As recently the hardware has become more and more generic and multifunctional, the architectural elements should also follow this trend for better integration.

As illustrated above, the development of new making technologies has always been a motivation for design innovation, and the insertion of cyber-physical production systems will have the same role in the next model change (Menges, 2015). Static plans no longer completely represent the spreading complexity and interdependencies in the computational design of buildings (Loo, 2022). These systems aligned with human-robot collaboration can facilitate the move away from instruction-based making towards behavior-based construction, having an impact on architectural design thinking.

1.2 Problem statement

Automation issue

The building industry is a major contributor to world-wide economies, accounting for 13% of the world's GDP. Regardless of this, it is often dominated by inefficiencies in the construction site, which is compounded by high-risk and low productivity (Burden et al., 2022). The building industry is still guided by manual processes, which results in high costs, questionable work quality, and a serious waste of human and material resources (Willmann et al., 2016). Although technology is considered the solution to treat these problems in many industries, in the building industry, construction is still conservative in adopting new technologies (Burden et al., 2022). Technology is present in the design process, but the construction methods are still inefficient together with an unsafe, chaotic, and dirty construction site. Frequently the building industry is compared with the manufacturing industry due to both being based on the production of physical objects by design means. The manufacturing sector increased exponentially its productivity over the last decade based on automated robotic technologies. On the other hand, the construction industry is lagging with stagnating levels of productivity apparently ignoring this direction (Wagner et al., 2020). The productivity of the construction industry did not change over time if compared to the manufacturing industry. According to the comparison in the ONS chart (Fig. 3), there is a productivity opportunity for the building industry that is

missing progress. In manufacturing, technology has been the leader of applied research that influences the industry to support the workforce while increasing the quality of products (Burden et al., 2022).

Robotics is one such technology that has the potential to develop new methods of construction for the building sector, originating with automation and developing into construable systems to replace repetitive tasks (Burden et al., 2022). However, the application of robots in the building industry is timidly concentrated on a controlled environment of factories to develop building components and missing extensively on construction sites. The fact that building construction is typically unique is one of the reasons for the low automation of the building industry. The customization required by different clients and sites cannot be summarized in one simple production. The changing nature of the construction sites, eminent safety hazards, the dependence on the context, and constant last-minute changes in projects are also the reasons for the lack of robot adoption. In addition, buildings are composed of thousands of distinct processes and parts that need to form a functional whole. This makes construction slow, expensive, and difficult to automate (Retsin et al., 2020). Although the complexity of a building is not greater than a car, the distinction is that a building

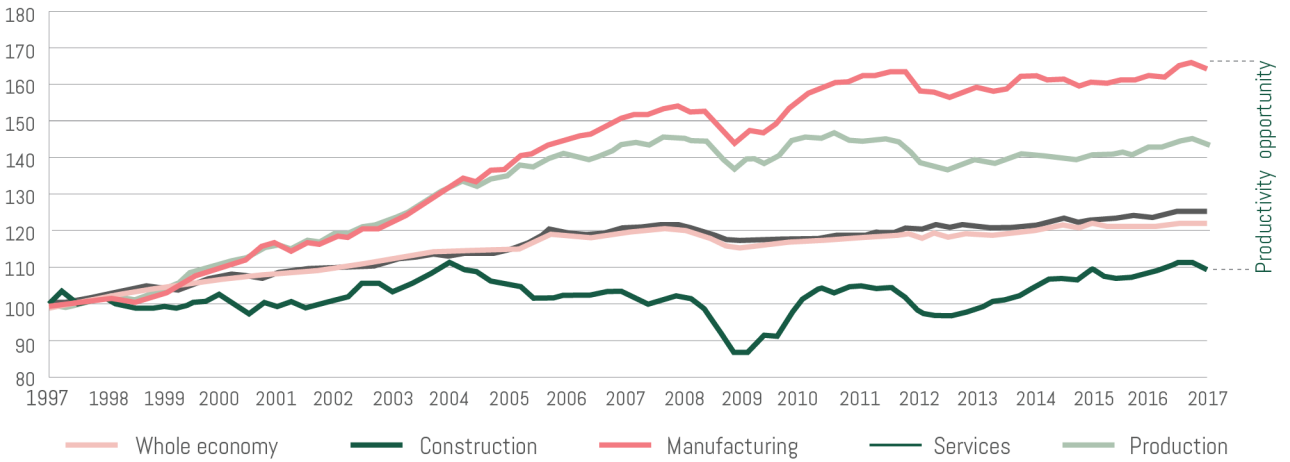


Figure 3. Chart of productivity growth - Output per worker. (UK Office for National Statistics, 2017).



Industrial Production

Structured
Static
Dedicated workspace
Highly automated
Controlled environment



Construction Site

Unstructured
Dynamic
Shared workspace
Low degree of automation
Digital gap

Figure 4. Problems in converting a solution for industrial production to the construction site. Adapted from Stumm et al. (2018).

cannot be materialized as a serial product of a production line, and so its construction cannot be executed according to the same standards (Gramazio et al., 2014, p.106). In the conventional model of prefabrication, robots, and buildings typically exist in separate spaces. Therefore, the relationship between the machine and the structure appears. (Gramazio et al., 2014, p.108).

In recent times, significant progress has been made in research regarding the integration of computational design and robotic fabrication, both in prefabrication and on-site construction. Nevertheless, there has been limited exploration of human-robot collaboration processes within the context of large-scale robotic construction scenarios (Mitterberger, 2022). Nowadays, industrial robotics are utilized mainly within a well-defined production setting. However, it is noticeable the shift from the repetition of static tasks toward dynamic human-robot collaboration (Stumm et al., 2018). Through these new developments, adaptable robotics can be utilized for new concepts of on-site robotic assistance, changing the static prefabrication level needed to make use of automation.

In this scenario, nowadays most of the HRC research in the building sector does not take place in construction sites, but still within laboratories or through digital simulations (Burden et al., 2022). In addition, machines with specific functions have usually low flexibility, while construction tasks can be very complex, especially when the tasks are executed manually and some few tasks are automated. Saying all this, the need for traditional construction workflows to adapt toward automated processes arises.

Architectural design issue

Since the beginning of architecture history, the design process is based on a specific function assigned to a geometry. All the elements of the building, such as columns, beams, walls, and floors, have a shape that characterizes this function. This paradigm got even more evident and clean with Modernism, where function-based design and material efficiency are the main design drivers. After Modernism, the other main movements were driven by aesthetics and the expression of the development of new technologies. The Domino House of Le

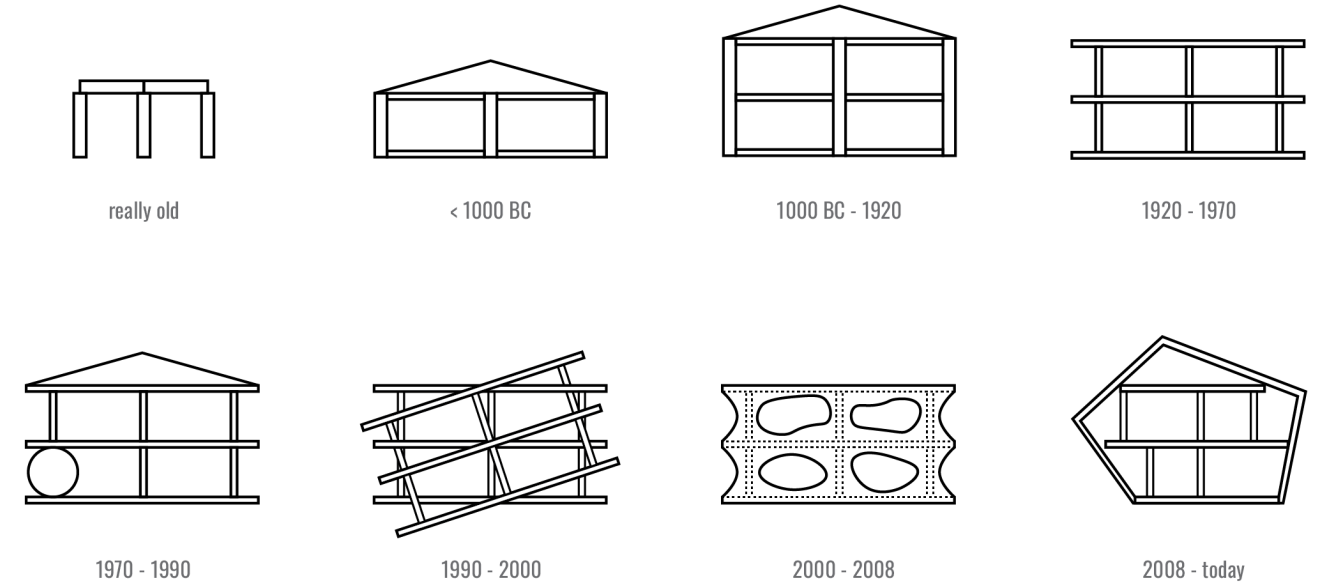


Figure 5. Architectural Evolution. Adapted from Gilles Retsin Lecture (2020).



Figure 6. Zaha Hadid's Heydar Aliyev Cultural Centre construction. (Zaha Hadid Architects, 2011).

Corbusier is still present in the methods of architectural design, however, now covered itself by complex geometries. Zaha Hadid and Frank Gehry's designs are examples of this idea of the production of extra architecture (Fig. 5 - 6), where a lot of material is used to cover the Modernist skeleton to achieve shape-plasticity. In the current age, architecture needs to relate to the new possibilities of construction. So enhancing productivity is not just a matter of adding technology to a construction site, but also embracing a breakdown point of how to think architecture.

Regardless of the evidence that architectural design is constantly getting more complex (Willmann et al., 2016), it is true that technology has been used to facilitate the construction processes of complex shapes, aiming for more efficiency; however, the design principles and construction methods did not change to have an efficient dialog with the new machinery in the field. Today there is a huge gap between computational design and digital fabrication that has been filled at a slow pace. Once the machines are imported from other areas like the automobile and naval industries, it is necessary to understand what are the inputs to develop a robotic workflow that responds to the architectural needs and how the construction site can be automated to create an efficient workflow from digital design to materialization. The

insertion of robots in construction processes is not about the renewal of Modernist efforts to transform the building industry into completely automated and rational production. Instead, it is the union of design and production that, aligned with digital processes, creates opportunities for the architectural materialization practice (Gramazio et al., 2014).

Digital fabrication emerged as the primary means of transforming intricate structures created in a digital environment into tangible forms, thereby enabling the efficient customization of large-scale productions. According to García (2019), it has been observed that this process, when applied to large volumes, is slower in comparison to traditional manufacturing methods. Additionally, a crucial point to consider is that the excessive variability often encountered in this process can result in a significant exponential decrease in assembly efficiency. In this way, discrete design processes promise to swift the continuous model thinking by replacing unique complex geometries with the assembly logic of the parts (García, 2019).

Circularity issue

The planet is experiencing a climate emergency at present. The building industry is recognized as one of the main ones responsible for the current ecological crisis (Rogea

ua et al., 2020). Just the building industry generates 30% of global greenhouse gas emissions, 40% of global energy use, and 50% of global waste (Wagner et al., 2020). Each new construction material is responsible for a significant share of carbon emissions and has a significant role in the generation of landfill waste. Thus the increasing emissions of carbon dioxide require a reformulation of the whole construction process taking material life cycles into account (Rogea

ua et al., 2020), and a shift towards more use of natural materials (Kunic et al., 2021b). In the international endeavor to decrease carbon emissions while producing new buildings, wood is becoming a significant topic of study in research and practice (Hansen et al., 2021). For centuries, timber has been the most significant construction resource being present in more than 80% of all buildings until the end of the eighteenth century (Menges et al., 2016). It represents a solution to the climate problem, being a renewable natural carbon store, and lightweight material. In addition, timber proposes the chance of solving current issues like the huge material usage in construction, and the global immense demand for affordable housing (Kaiser et al., 2021). However, thinking about modularity for reconfigurability and reuse aiming circularity levels, the use of timber in construction is currently based on design-specific elements making timber poorly reusable (Kunic et al., 2021b). In this way, as one of the main challenges for the building industry is material circularity, a deep reconsideration of construction practices needs to happen to achieve circular processes (Kunic et al., 2021a).

Even with the dissemination of robotic technologies, the automated assembly of timber structures on a building scale is still a challenging procedure when requires custom assembly strategies. Also, as wood is a heterogeneous material, it brings many questions to the timber and architectural field to use building-scale robotic assembly processes, for instance, piece material characteristics, weather-related shape variations, geometrical imprecisions, and tolerances (Kramberger et al., 2022). Therefore, beyond these implications to achieve an efficient construction process, it is also important to understand how design can contribute to expanding the life cycle of not only discrete timber elements but also the building as a whole through reconfigurability.

1.3 Design goal

Circularity is a priority subject within the current agenda of architecture (Kunic et al., 2021a). The Netherlands intends to have a circular economy by 2050 (Government of the Netherlands, 2016). It refers to a waste-free economy where products and raw materials are reused and which runs as much as possible on sustainable and renewable raw materials. The United Nations has an environmental program that aims to switch from the current linear economy to a circular mode of production as illustrated in Figure 8. It approaches the principle of reducing by design, as well as reducing-impact production processes such as refusing, reducing, reusing, repairing, refurbishing, remanufacturing, repurposing, and recycling (United Nations Environment Programme, 2019).

A profound rethinking of processes in design and making is required to accomplish such ambitions. Robotics plays a significant role in this goal by giving instruments and workflows for the automation of the assembly of circular products (Kunic et al., 2021a). In addition, discrete timber assembly presents various possible solutions to the lowering construction productivity, the housing crisis, and the global climate crisis (Retsin et al., 2020). The potential for integrating HRC into construction processes arises from recognizing the strengths of robotic systems. These strengths include their ability to handle extensive data processing, exhibit a high level of precision, and undertake tasks that are dirty, dangerous, and repetitive in nature (Loo, 2022). Thus, to face the mentioned issues, this research proposes the combination of robotic construction with discrete architectural thinking by creating an efficient human-robot collaboration assembly workflow through a mass customization structure of timber discrete parts.

Further, there is a renewed enthusiasm for modularity and prefabrication as an alternative to the housing crisis (Retsin et al., 2020). The combination of automated construction and discrete architecture aiming for circularity can lead the construction process to save capital that could be used for other societal issues and decrease housing prices, potentially having an economic impact. The words "industrialized building", "prefab" and "modular construction" summarize the attempt to minimize the complexity of the construction process through standardized systems and the repetition of modules. Currently, these organizational concepts give the

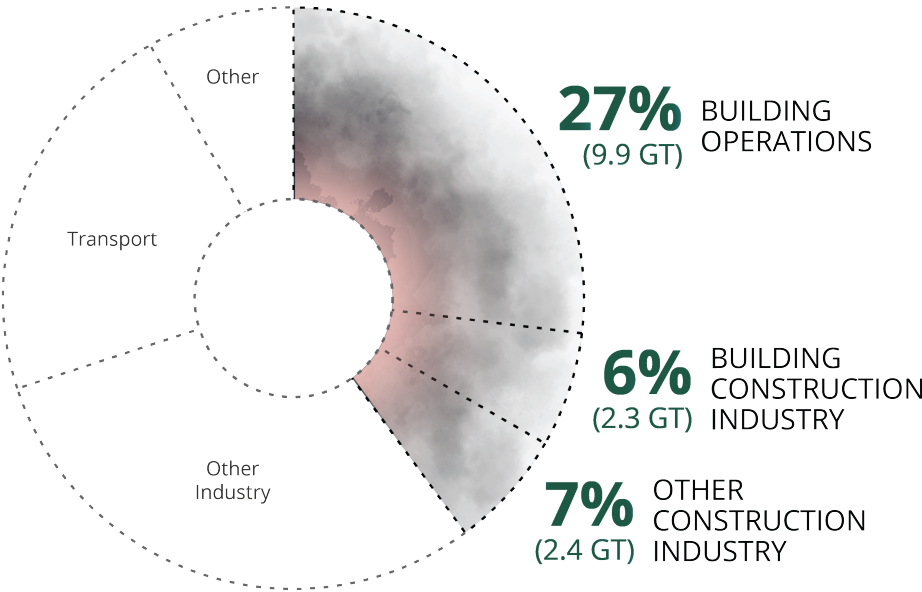


Figure 7. Annual Global CO2 Emissions. (Adapted from Architecture 2030). (Data Source: IEA (2022).

best direction for future greatly automated construction (Wagner et al., 2020). The merge of robotic fabrication with the renewable material wood shows the way to a possible future where local materials get aesthetic and structural expression through bespoke automated methods. (Willmann et al., 2016)

Robotic assembly methods are emerging. The operation of robots facilitates precision and efficiency through the integration of devices able to give feedback, adding possibilities for joining various generic systems made of wood profiles (Kunic et al., 2021b). Robotic construction talks about precision, labor efficiency, and a well-planned construction site which results in a clean and organized assembly site. Additionally, it is important to mention again that the manufacturing industry has fixed robots due the produced object can be easily moved. On the other hand, as in the construction industry, the final product needs to be placed on a specific site, it requires robots able to move around the construction,

and components designed to be built inside this work-frame. Recently, an approach linking discrete design with robotic making has risen based on combinatorial logic to develop an interlocked design from relatively simple modular elements. Thus, the importance of discrete timber structural systems is expanding guided by the idea of automated assembly, and the circular economy of resources (Kunic et al., 2021b). The discrete architecture brings to this workflow the flexibility and standardization of movements that robotics requires to achieve efficiency. In this design thinking, the part has no prescribed function. It always remains autonomous in the design. The function-based design thinking gives place to a design the part can change function over time. The same geometry can generate a stair, column, floor, or wall. Through the Discrete approach, a large number of discrete particles approximates function-based parts. Inside this scenario, the connection and aggregation rules pattern play a big role. Inside this paradigm, the final design is generated by these connections and combinatorial rules.

Thus, the research approaches a robotic assembly workflow and uses a timber discrete structural system to enable not only efficiency in the automated process, but also disassembly, future reuse, and reconfigurability, while aiming for levels of circularity as the central goal by linking these areas. The study focuses on one repetitive element and specific robotic kinematics to connect the parts in a reversible way. This combination merges concepts of prefabrication, discrete modularity, reconfigurability, and aggregation logic underfoot to calibrate the architectural design proposal to digital tools (Koerner-Al-Rawi, 2020). In addition, the proposed workflow is taking humans as the main character while working in collaboration with robots, having the housing requirements inputs, and understanding its societal impact. The research does not aim at the technical development of robots, but instead at the robot-driven materialization processes from an architectural perspective. The work explains the application of a multiphase procedure of an automated system of production from manufacturing, logistic, and robotic assembly sequences that contain spatial manipulation, positioning of timber elements, grabbing, and implementation of disassembling connections. The workflow pursues an additive assembly that uses computational design to set the efficient procedure order that generates the discrete timber structure. The discrete fabrication required the advancement of a new end-to-end workflow that not just digitized designs, but also materiality (Gershenfeld et al., 2015). The generative algorithm will also formulate constraints and guidelines for the assembly sequence and design process, as illustrated in the reference works (Kunic et al., 2021b) (Man, 2021). The intention is to expand the material life cycle, and in return, minimize the carbon footprint of the building by creating a unique component for assembly, disassembly, and reuse for other emergent configurations inside of the pre-determined aggregation rules. The research aims to contribute to integrative thinking regarding geometry, structure, joints, programming, assembly, fabrication, and quality requirements.

After referring to all these issues and concepts, the main research question emerges.

1.4 Research question

How can an automated assembly process for a discrete and reconfigurable timber construction system contribute to circular housing?

Sub-research questions:

- Q1 How can discrete architectural design thinking improve the feasibility of an automated assembly process?
- Q2 How to implement a human-robot collaboration into a site-specific construction workflow?
- Q3 How can a discrete timber element be designed to attend circularity levels and a human-robot assembly process?

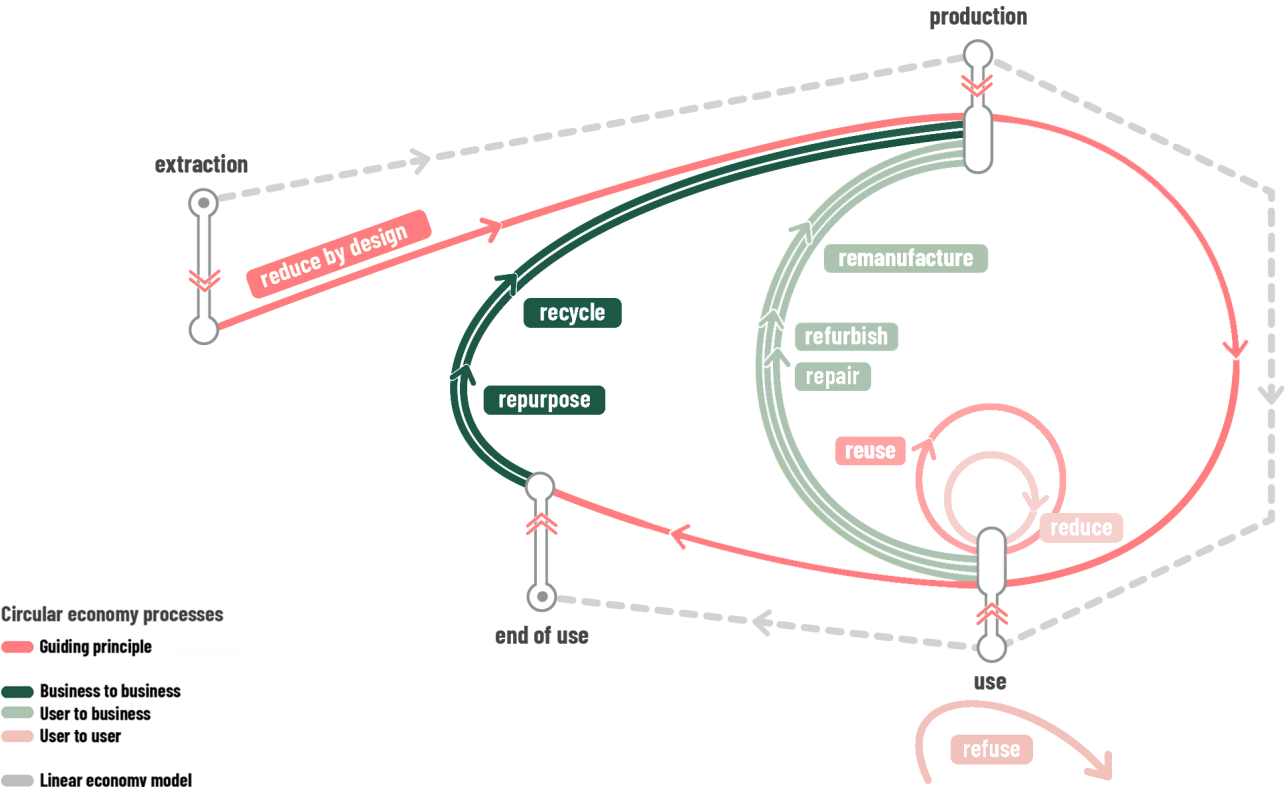


Figure 8. United Nations circularity approach. Adapted from United Nations Environment Programme (2019).

1.5 Vision & Scope

Work less,
work for everyone,
automate everything,
produce what is necessary,
redistribute everything.

The vision of this research has its roots in this statement. It is a call to action for a more equitable society. It calls for a reevaluation of work in society, advocating for automation, reduced working hours, necessary production, and resource redistribution. The original phrase is “trabajar menos, trabajar todos, producir lo necesario, redistribuir todo” with origin in the Zapatista movement in Mexico. It questions the prevailing notion that work inherently dignifies individuals and proposes implementing universal basic services and income. Sandoval (2023) emphasizes the need for planned automation and rejects arbitrary implementations that lead to unemployment. The background motivation of this research is to contribute to the shift away from a work-centric society and highlights the potential for increased productivity with reduced working hours. It suggests that the current discussion on pension reform is an opportunity to reformulate not only the construction methods but the role of work in society (Sandoval, 2023).

In order to contribute to the implementation of this vision in the world, the scope of this research focuses on creating an automated workflow for designing and constructing discrete timber structures using robotics, with the ability to disassemble them if needed. **The focus is on the connectivity development of a construction system that can work in the syntax of robotic construction and circularity.**

The discreteness aspect acts as the interface of the combination of these two areas of knowledge (Fig. 9). The scope concentrates on the feasibility assessment of the development of an open construction system that embraces time and changes by taking design-for-disassembly and design-for-future-reuse as guidelines. It goes in the opposite direction of function-based optimization which leaves the construction static. Genericness is the core key design aspect of the elements that precede a prescribed function used in this work to propose a construction system that has an organic way of growing and being adaptable. The research is about a methodological system that can generate architectural objects.

It is also the scope of this research to understand if an HRC automated assembly process has the potential as a circular construction method for housing. The research does not intend to make structural simulations and calculations. It looks only into the connectivity and how each stage of design influences the whole robotic construction workflow. The site context is used as a guideline and background to set some boundary conditions for the representation of a possible implementation of the developed construction system. The societal issue of affordable housing addressed in this research is present in the whole contemporary world and not just in the chosen site context.

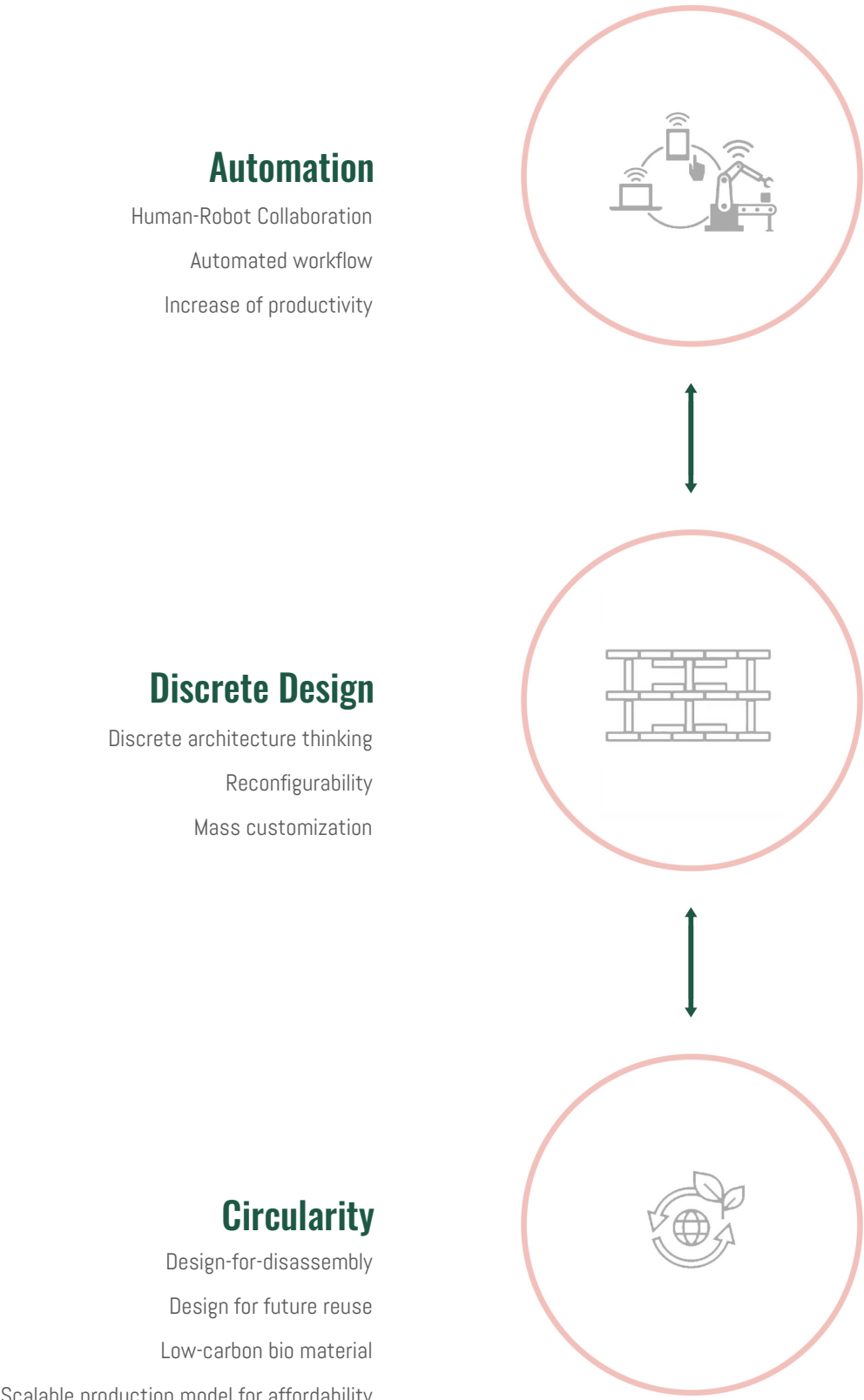


Figure 9. Common syntax diagram of the three areas of the research. (by author, 2023).

1.6 Methodology

Research through Design (RtD) is an approach to research that involves creating and evaluating new designs in order to advance our understanding of design itself. Unlike simply creating prototypes or improving existing designs, RtD utilizes design as a research method to explore and generate new knowledge (Stappers & Giaccardi, 2018). Consequently, this study investigates the workflow of assembling timber structures using robots by employing the RtD approach. This approach involves the use of digital design algorithms to generate sequences for assembly, and the timber elements are designed with principles that allow for easy disassembly and reconfiguration.

To address the main research question, the study combines two interconnected approaches: theoretical and technical design propositions. The theoretical aspect encompasses a broad exploration of topics such as robotic technology, discrete architecture, timber circularity in architecture, and

the importance of computational design tools for spatial housing configuration. On the other hand, the technical design aspect focuses on developing a construction system of discrete elements, incorporating generative design concepts, simulation, and 6-axis arm robot control.

The research progresses through several phases, as depicted in Figure 11. The initial phase involves developing the design of discrete elements to shape the architectural housing. Subsequently, the study investigates the optimal robotic construction workflow based on a construction site scenario. Finally, prototypes are designed to conduct testing experiments. An assembly prototype using human-robot collaboration (HRC) is established to assemble the discrete aggregated structure, employing demountable connections to join the elements. The research adopts a comprehensive approach to explore the research questions, combining theoretical exploration with practical design and experimentation.

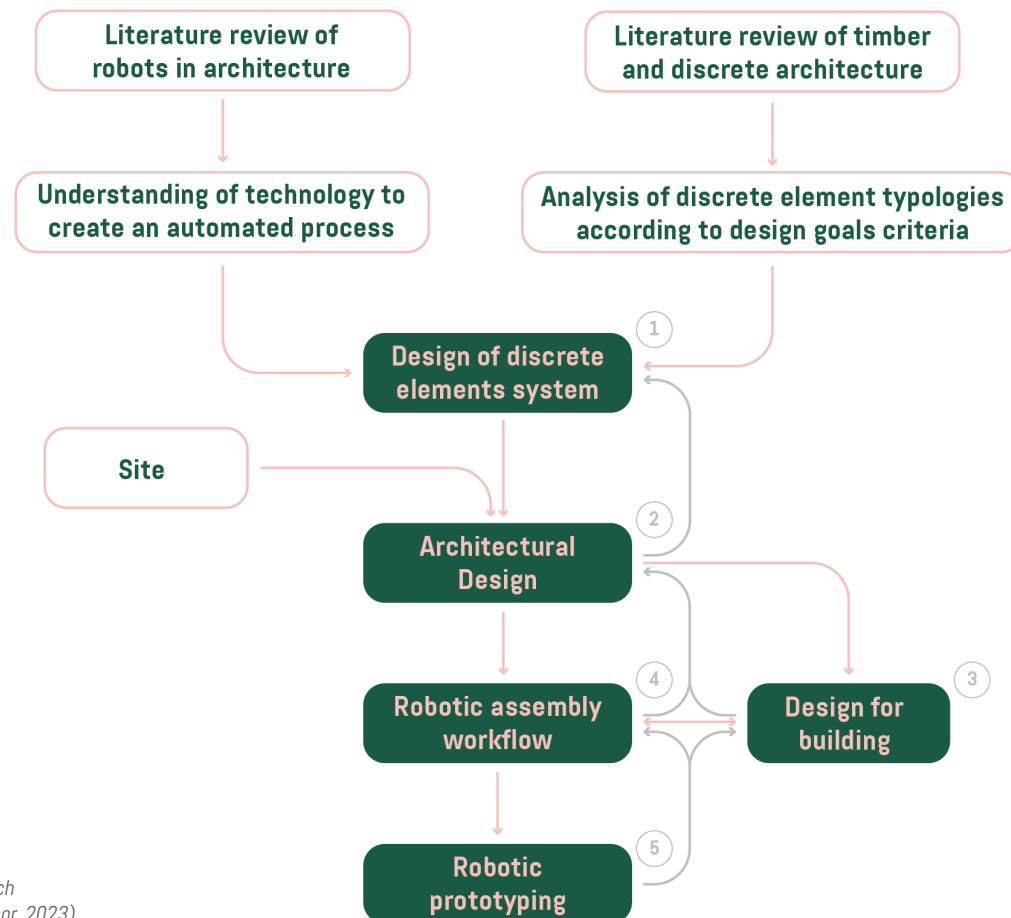


Figure 10. Research workflow. (by author, 2023).

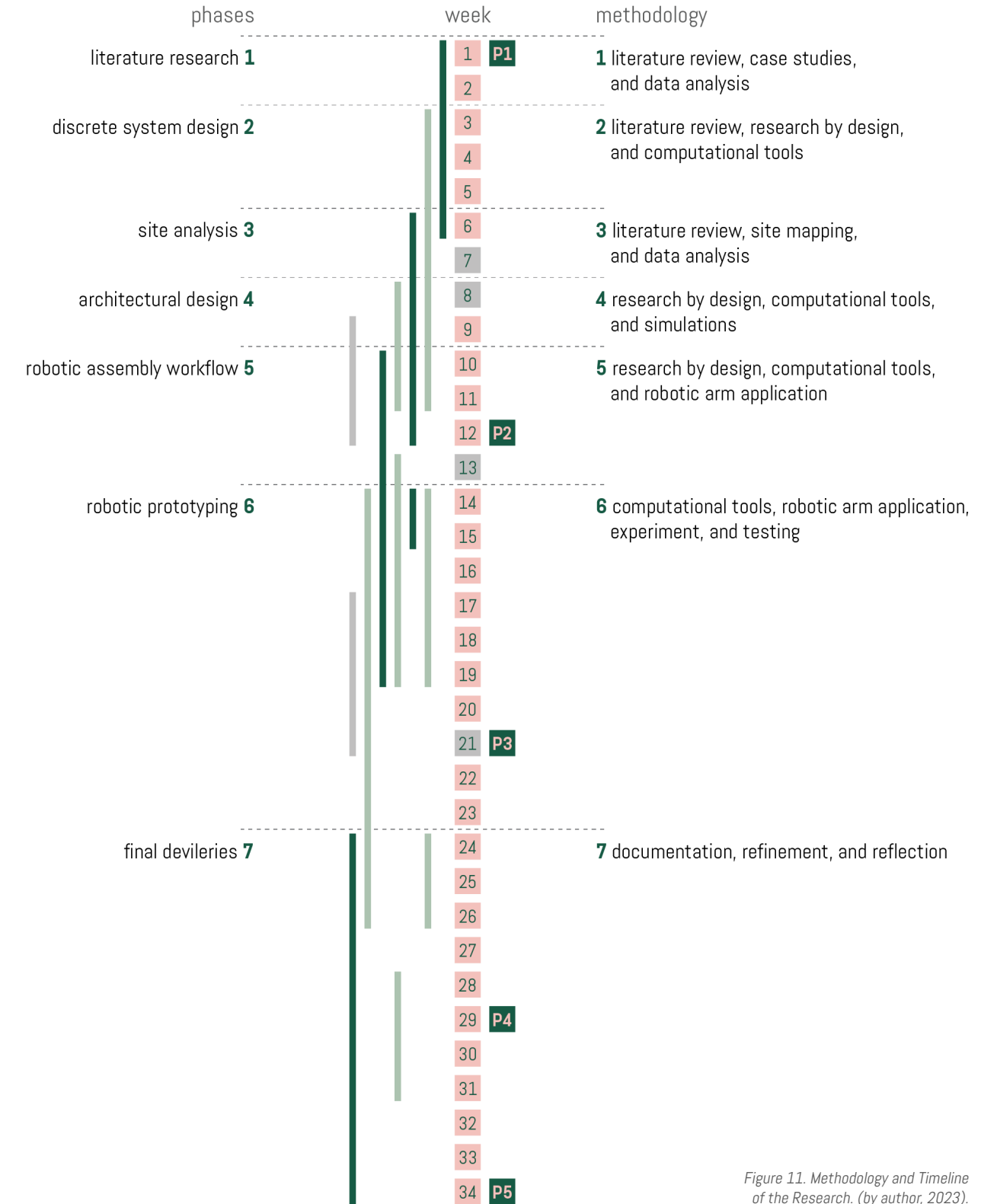


Figure 11. Methodology and Timeline of the Research. (by author, 2023).

2 Robotic Fabrication

- 2.1 Robotic Fabrication in Architecture
- 2.2 Human-Robot Collaboration
- 2.3 HRC state of the art
- 2.4 HRC on construction site
- 2.5 HRC usage evaluation criteria

This chapter elaborates on the robotic state-of-art in architecture and its requirements for the design to production. The building industry has a long research history in the insertion of robotics into the field, however, it has not yet been adapted yet into commercial practice (Gershenfeld et al., 2015). The current automation in construction has focused itself on augmenting rather than replacing traditional construction methods. The purpose of this work is to create a construction system that can enhance a novel automation process.

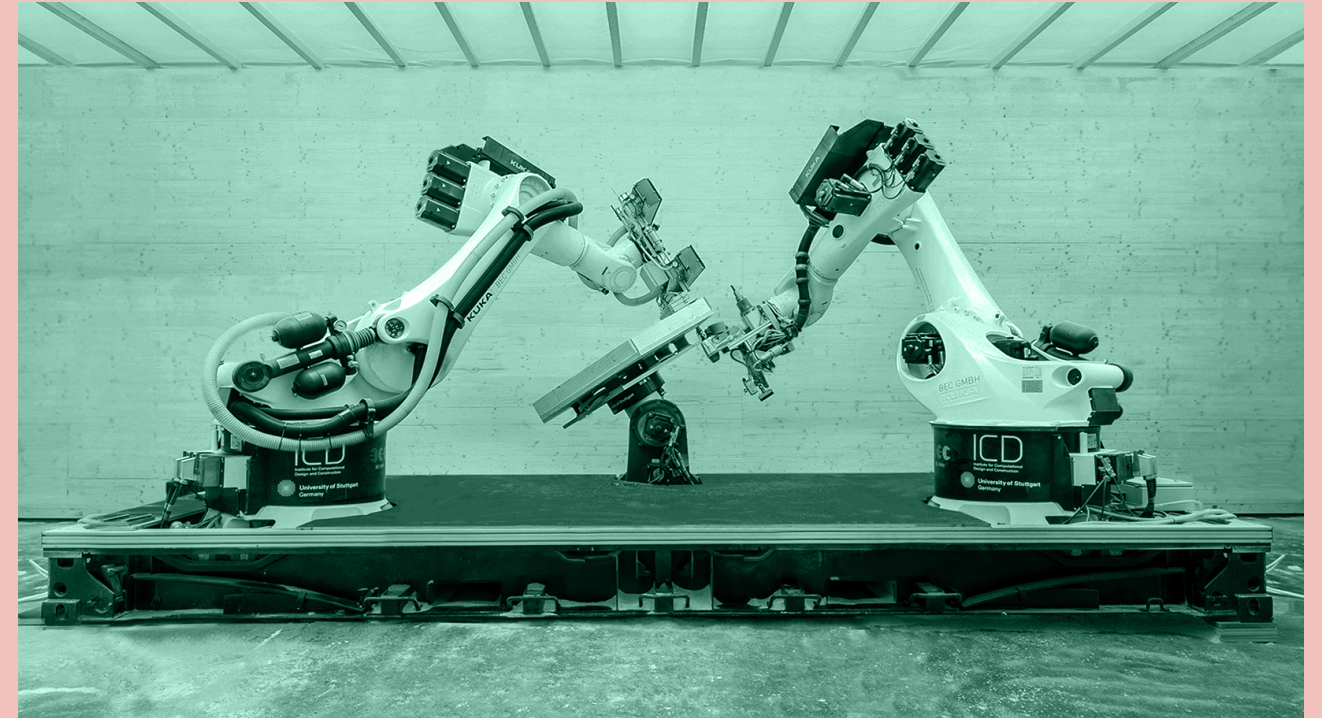


Figure 12. Flexible and transportable robotic timber construction platform, TIM. (Wagner et al., 2020).

2.1 Robotic Fabrication in Architecture

Robotic Fabrication in Architecture is related to neither a new philosophical model, a new style, or an aesthetic, nor the mere affirmation of machinery capabilities or the pure optimization of automation possibilities (Gramazio et al., 2014). On the other hand, it is about the materiality obtaining the main architectural importance. The focus is on physical behavior, material sense, and constructive details in a way that each aspect influences the coherence of the whole design (Gramazio et al., 2014, p. 20). In robotic assembly design processes, each element is part of a logical system including the specific data about its relation to the surrounding members that originate the final information and the spatial sequence for the robotic assembly (Willmann et al., 2016). Inside this paradigm, the architectural design is not created as a final geometrical shape, but as a digital generative process driven by the materialization rules. The design of an assembly system is strictly associated with the design of its fabrication process and tool. Thus, the methodology involved

guides toward a future of construction processes that do not rely on a static form, but instead on a programmed set of flexible rules of fabrication, assembly, and joint design (Helm et al., 2016) (Schwinn, 2016). Building with robots takes architecture to the design of formative material processes based on fabrication strategies, in which information and material, computational design, and construction, are thereby interwoven connecting the material reality of architecture with computational thinking (Gramazio et al., 2014, p. 15) as the design already have the knowledge of its machinery buildability at its initial concept phase.

A great number of elements, their accurate organization instructions, high degree of definition, and a clear distinction between the single elements and the whole design generally characterizes automated construction processes (Gramazio et al., 2014, p. 183). These robotic aspects share principles with the discrete architecture thinking that is covered in the next chapter. It is important to mention that as in this

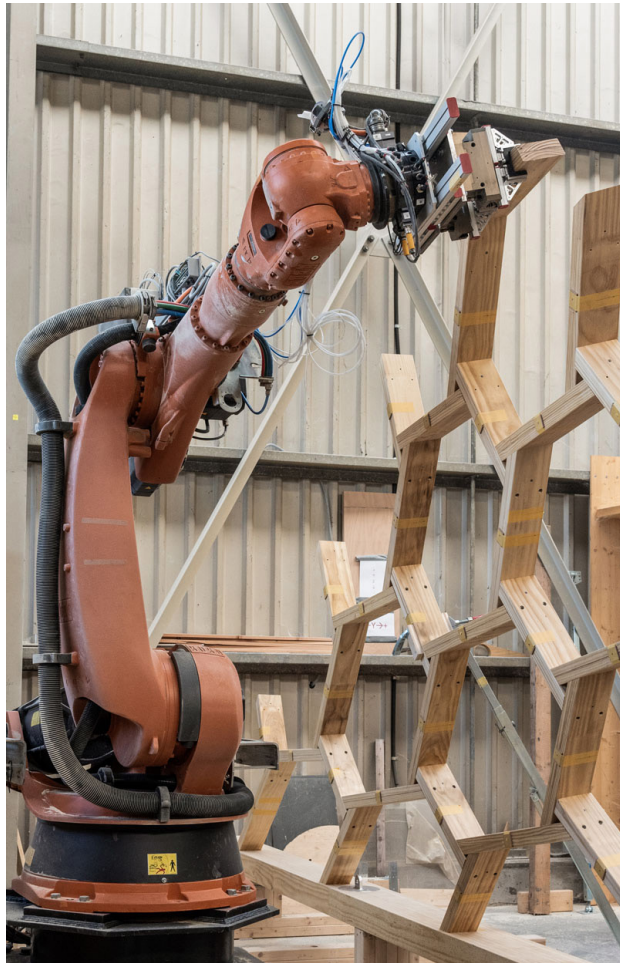


Figure 13. Future Tree roof fabrication robotic assembly. (ERNE AG Holzbau). (Gramazio Kohler Research, ETH Zurich, 2019).

methodology the architecture is generated by the elements and their joints, the robot needs to reach all assembled parts placement at the construction site. While this complexity can be reduced with various robots working together to decrease self-collisions, it is still difficult for the robot to operate through the spatial intricacy once more joints and modules are assembled into the overall structure (Koerner-Al-Rawi, 2020). In this way, the joint sequence and its design have a crucial impact on the uncertainty in the assembly system. The designer is required to learn the behavior of the

aggregate material system to create combinatorial strategies that permit progressive flexibility during the assembly, contrasting to the traditional building method in which the designer defines the place of the elements (Gramazio et al., 2014, p. 18).

Although it seems industrial robots are not made for the dynamic construction industry, they can act as mediators between design and construction by being multi-purpose machines that allow free workspace layouts, and by the capability to process digital data toward physical fabrications (Stumm et al., 2018). Industrial robots facilitated the increase in research attempts toward building systems innovation through the robotic construction approach due to their various degrees of freedom, comparably low price, robustness, generic design, custom end-effector, high precision, speed, and adaptability (Wagner et al., 2020). However many adaptations need to be done according to fit inside the nonstatic environment of construction. Due to the great level of customization in construction, material imprecisions, assembly tolerances, and applied forces during the assembly sequence, simple tasks can be quite challenging for industrial robots even though they are good at repetitive routines (Kunic et al., 2021a). The reachability of industrial robots and their pose restrictions is also the reason for complications in the assembly process (Koerner-Al-Rawi, 2020). Significant divergence can happen between digital models and physical prototypes as a result of joint tolerances, self-weight deformations, and humidity variations (Rogeauea et al., 2020). At the building scale, it is a fact that precision is not enough in either anisotropic materials or the positioning process. As a consequence, the accumulation of tolerances creates issues with assembly procedures (Willmann et al., 2016). Thus, a real-time closed feedback system assessing the tolerances constitutes a significant step toward the application of a complete adaptive fabrication process.

The possibility of adopting robots to connect basic elements into an intricate whole design is the base of the concept of generic building elements. It relies on the capability to create multiple assembly configurations resulting in complex and function-based building parts from a small element, such as a brick (Gramazio et al., 2014, p.183). Brick keeps until nowadays the most generic building element of construction. However, when a generic basic part transforms into a specific

object by having predetermined connections, their constructive freedom turns limited. Sometimes in these scenarios, the robot production is minimized to a mere manual work process that maybe would be assembled more smoothly and quickly by hand than with a robotic procedure (Gramazio et al., 2014 p. 185). Thus, the more generic the better for the robot because it has a higher degree of freedom, however, for efficient production, the design of the elements needs to be aligned with the machinery available and has to balance the genericness of the element with its connection specificity. The equilibrium of these factors also helps achieve design disassembly aspects and circularity levels.

Once some tasks in construction are challenging to automate and certain tasks require years of expertise from skilled workers, the fundamental aspect of a successful automated assembly process is the integration of humans into robotic production. Lately, it is possible to note a shifting trend of collaborative robots taking the place of conventional 6-axis robots for collaborative tasks that can be shared with humans in the construction workplace (Kramberger et al., 2022). The construction sector needs automated processes with a high degree of flexibility, and the cooperation of industrial robots with humans can be the solution (Kaiser et al., 2021).

2.2 Human-Robot Collaboration

The collaboration of robots with humans will perform a significant role in the future of construction sites, which will be a complex environment involving machines and human beings using their best superior abilities to supplement the abilities of machines (Gramazio et al., 2014). Inside this cooperation, the operability of the robot becomes valuable when some complexity in design is acquired, where multiple design and constructive associations outpace the human capacity to supervise and act over them. The relationship between humans and machines discussed here is more about complementarity than a dichotomy, whereby human ingenuity, intuition, and creativity connect with machinery speed and inexhaustible productivity. It refers to the activities in which both work together to finish a project-related task in a specific workplace (Burden et al., 2022). In theory, human strengths, like decision-making, problem-solving, and complex sensory-motor skills, and robot power, such as precision, efficient repetition, controlled use of force, and high productivity, should be exploited in HRC. By selecting the role each agent plays considering that humans and robots have different strengths, this combination talks about permitting humans to ally with robots whenever it can boost architectural design quality, not towards the mere compensation for the humans' deficits (Gramazio et al., 2014).

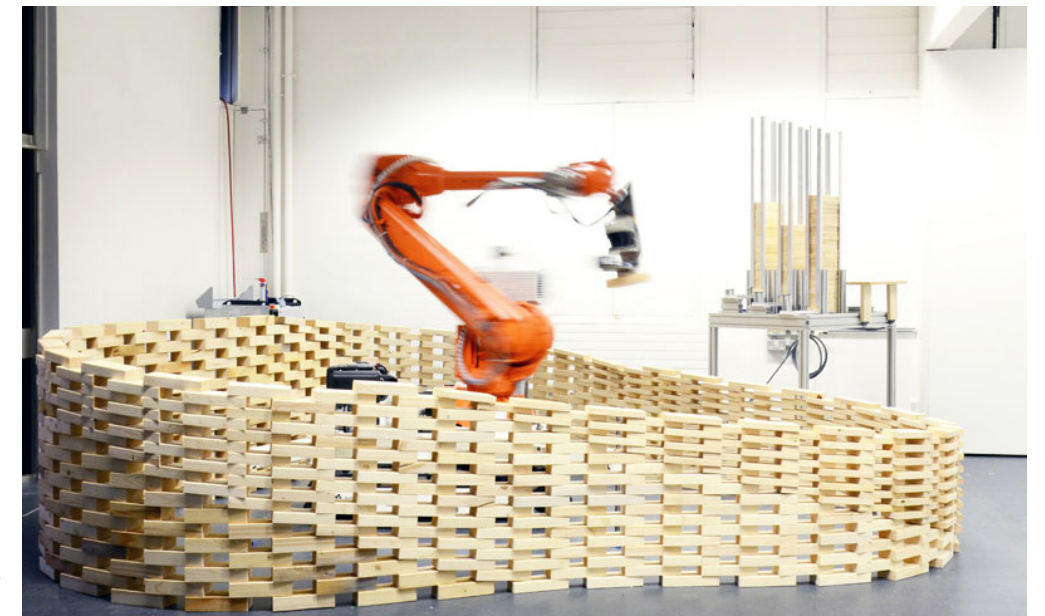


Figure 14. Stratifications. Adaptive robotic fabrication in which the design is generated from the robot's operation using a feedback loop to inform its assembly. (Gramazio Kohler Research, ETH Zurich, 2011).

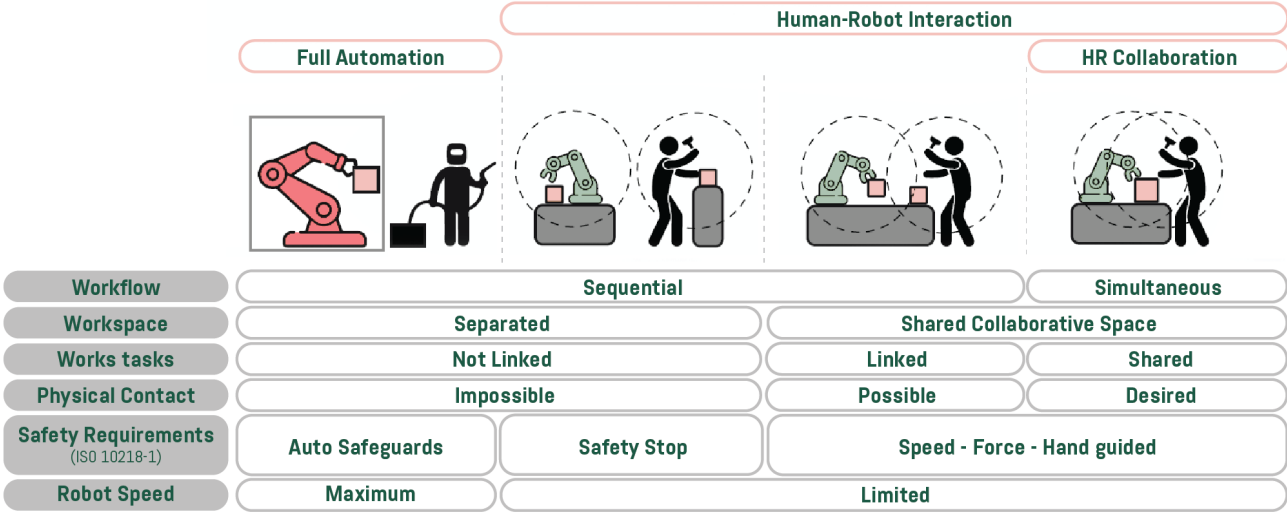


Figure 15. Types of Human-Robot Relationship. Adapted from Burden et al. (2022).

According to Burden et al. (2022) there are different types of human-robot relationships that are expressed in Figure 15. Collaborative robots, or cobots, are frequently confused with extremely automated industrial arms. Therefore, there is still the need to create confidence through successful case studies that establish fruitful results and productivity evaluations to bury the boundaries of cobot adoption in the building sector (Burden et al., 2022). Cobots, equipped with force-torque sensors integrated into each joint to detect contact and pressure, serve as the primary tools used in HRC. These cobots feature smooth contours and sleek designs, aiming to minimize any potential harm to humans during close collaborative interactions (Loo, 2022). The leading manufacturers of these collaborative robots are ABB, Kuka, Rethink Robots, and Universal Robots. They offer various models, including the YuMi-IRB 14000 from ABB, the LBR iiwa from Kuka, the Sawyer from Rethink Robots, and the UR Cobots from Universal Robots.

In collaborative processes exists a great variety of strategies for such task distribution between robots and humans. One of them is machine-assisted human fabrication, where a machine assists a human while fabricating (Mitterberger, 2022). In this case, automated processes are partially used in the assembly and the physical result still relies on the human overview. On the contrary, **the focus of this research is human-assisted robotic fabrication**, in which the robot is the main responsible for the assembly but still counts on human help for emergent situations during the construction

process. In such a hybrid strategy, the chosen tasks for the human side do not ask for refined context perception or great dexterity (Mitterberger, 2022). The robot can ask for help with a task if it fails on doing so, demanding a shared knowledge base between humans and robots (Stumm et al., 2018). For a successful collaborative workflow to happen the interaction between workers and the fabrication system must be clearly defined by sharing meaningful specific tasks necessary to manufacture, and a common information base (Kaiser et al., 2021). All individuals involved in the fabrication system share these steps of production as a flexible task shop. In such a workflow the tasks sequence depends on one another and absorbs spatial dependencies. In a case study explored by Mitterberger (2022), robots performed precise spatial operations while humans performed physical tasks that were difficult for the robot, as illustrated in Figure 16. On the human side, the execution of the tasks consisted in positioning elements that touch onto existing structures, tying knots, digitization of manually placed elements, spontaneous design decisions, and adjustments. The role of robots in this context involves performing spatially complex routines and providing structural stabilizations to assist in the assembly process, which can be challenging for humans. In order to facilitate a collaborative workflow between humans and robots, it becomes necessary to have a regularly updated digital model that allows for the mutual distribution of tasks between the two entities (Mitterberger, 2022).

2.3 HRC state of the art

This section discusses the state-of-the-art of human-robotic collaboration to understand its current technological advancements and create criteria to implement HRC into the construction workflow proposed. Thereby, recent researches have explored methods to understand how collaborative robots can safely interact with humans through complementary systems, for instance, vision and touch sensors, augmented interfaces, digital twin models, machine learning, and artificial intelligence. In addition, strategic methods, such as swarm behavior, have been used to enhance efficiency and prevent collisions when many robots are working collaboratively to build a structure (Man, 2021). According to Loo (2022), two main areas of improvement in the HRC affect its implementation in construction sites, the advancement in **collaborative operations** and **calibration methods**. As collaborative operations with humans require intuitive communication, and robots require calibrating and adapting to perform tasks with optimized precision, adaptability and calibration become key aspects for the implementation of HRC in construction processes. Thus, cyber-physical systems and human-computer interaction are key areas in the development of HRC while improving the workflow of design-to-built in the construction industry (Loo, 2022).

The application of cobots in construction workplaces offers many opportunities to add flexibility to the human ecosystem, such as performing tedious or unhealthy tasks allowing humans to focus on decision-based work, minimizing the cognitive load of workers, reducing errors, and sharing the labor of picking, placing, and assembling materials while helping to control production quality (Burden et al., 2022). However, it also comes with challenges like the design of an efficient workflow considering ergonomics and a safe environment for humans. In order to establish a collaborative production environment between robots and humans, several factors must be considered. These include ensuring safety measures, maintaining precision in operations, addressing variability in tasks, fostering adaptability to different scenarios, and managing the complexity of the tasks at hand (Loo, 2022).

A collaborative approach in the timber industry involving skilled workers can enhance productivity while maintaining flexibility. To achieve this, a modular, reconfigurable timber fabrication system needs to be designed with proper interfaces and collaboration workflow for effective human-machine communication (Kaiser et al., 2021).



Figure 16. A cooperative proof-of-concept prototype has been developed to evaluate the design principles and workflow of an assembly cycle comprising five key components: interactive design, robotic assembly, manual assembly, rope jointing, and element tracking (Mitterberger, 2022).

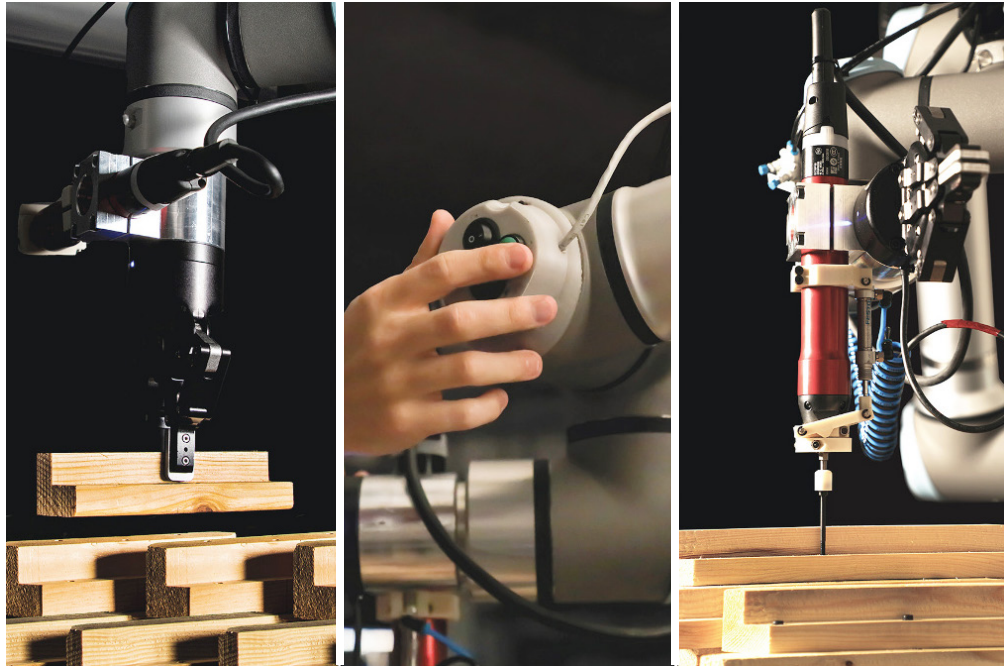


Figure 17. HRC sequence of tasks consisting of pick-and-place of the timber construction elements, human switching the robot's screwing configuration to collaborative mode, and robotic screwing procedure. (Kunic et al. 2021b).

The area of improvements corresponding to collaborative operations between humans and robots is about allowing the robots to achieve sensibility and communication methods approaching what is a collaboration among humans. Therefore, various advancements based on learning and recording human motions through machine learning have been done aiming to convert human behavior into digital language for robots (Loo, 2022). Some of them as outlined by Loo, 2022, consist of [1] distributed mobile robotics (DMR) as a semantic language, [2] digital twin models, [3] sensor information or 3D camera, [4] real-time-adaptable toolpath in response to human force, [5] haptic learning with neural networks, [6] machine learning, and [7] tactile sensing.

Therefore, Loo (2022) argues there are several methods for enhancing collaboration, including [1] intuitive programming, [2] optimizing the workplace for improved ergonomics, [3] utilizing hand gestures for interaction, [4] implementing auditory dialogue systems for interaction, [5] optimizing trajectory, [6] employing admittance control, [7] applying image processing for safety, and [8] incorporating camera systems. When we talk about the improving area of calibration methods is about the robotic system response to interference in the external environment and building parts. The input data asking for change can be either from the human involved

or attached sensors. Some calibration methods depend on vision, incorporating indicators, an accurate reference size, and a stereo manual-ocular system with adaptable camera coordinates (Loo, 2022).

According to illustrate some of these advancements applied in current research, two projects were selected as examples, [1] the HRC workflow for the assembly of wooden structures by utilizing rope joints by Mitterberger (2022), and [2] the automated assembly of reversible timber beams by Kunic et al. (2021b). In the first example (Fig. 16), the 5-day production of a prototype structure tested and validated an assembly method and a computational setup. Two humans and two robots assembled collaboratively in interdependent actions a timber structure with a floor area of 6x4m. In the experiment, three pre-assembled elements were anchored to the ground in order to start the assembly procedure, in which of 38 timber assembled timber pieces, 29 were manually placed (Mitterberger, 2022).

In the second case about reversible beams (Fig. 17), Kunic et al. (2021b) well implemented a novel robotic approach to automate the assembly of demountable timber structures, including its design and manufacturing. With this experiment, a prototype was materialized to present the process and the concept of reversible structure and to prove that

customized carbon-efficient structures can be done without processes that use voluminous material through the aggregation of pre-fabricated discrete elements. Also, according to avoid time for exchanging tools, a multi-phase and effector was designed to link the robotic assembly phases aligned with the successful workflow combining design phases and optimization of the structure. The robotic procedure made use of data exchange between the digital design and the reality of construction by a cyber-physical system, in which the screwing task demanded human aid because of tolerances during the procedure. The collaborative construction approach implemented the screwing operation as the main aspect of communication, as well as the unscrewing task (Kunic et al., 2021b). The unscrewing procedure was demonstrated to be mechanically feasible but challenging in a large assembly environment due to accumulated deviations in comparison with the digital model. Thus, the experiment was disassembled by hand and robotically remounted, extending substantially the life cycle and carbon-storing time of the material. Thus, the prototype proved its concept of how a discrete construction system based on robotic assembly can evolve over time by the reversibility of the joint employed. (Kunic et al., 2021b)

2.4 HRC on construction site

Currently, the building industry uses on-site prefabrication to manufacture large-scale assembly elements before a crane elevates them in their place. Such a method is only possible by an effective workflow that embeds site-specific planned setups and logistics (Wagner et al., 2020). This approach can shape large-scale building components on site avoiding the difficulties of transportation, or at least decreasing the transportation routes by placing the robotic production in a temporary near-site controlled environment factory. While explaining the development of the TIM platform (Fig. 18) for timber, Wagner et al. (2020) also presented some others of these robotic construction flexible platforms, such as R-O-B. Inside this production framework, the reuse of digital design and physical robotic technologies is possible while having the needed adaptability required by the unique site and client requirements and counting with the integration of the building generative design with the fabrication system by using a multi-scalar automation process (Wagner et al., 2020). However, there are general challenges in construction sites for cobots.

Human-robot collaboration can find many challenges when applied to a specific construction site in comparison with a controlled environment. These challenges guide the

Figure 18. Flexible and transportable robotic platform in transport position with robots and control cabinets being stored on the platform with removed cover. (Wagner et al., 2020).

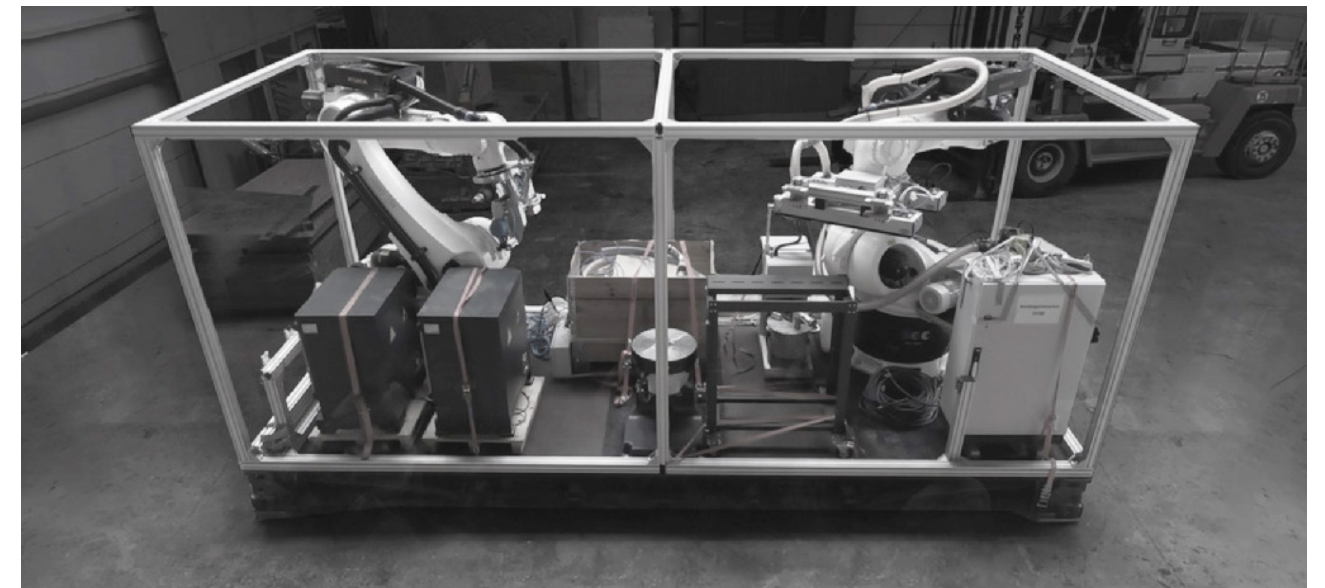




Figure 19. In Situ Robotic Fabrication in which the robot recognizes its own position, the surroundings, and its components with regard to the material tolerances. (Gramazio Kohler Research, ETH Zurich, 2012).

research's intention of creating a workflow using cobots in the construction industry. The main issue is the dynamicity of the construction site, which can affect the position of machines and building components even as weather conditions can disturb the functionality of the cobot's sensors (Burden et al., 2022). This also relates to the challenge of cobot self-contextual awareness and localization of its position and the precise end effector position. Adding to this, the eventual instability of mobile platforms attached to cobots together with irregular and unstable grounds can complicate even more the adoption of HCR in construction. Dirt, dust, fluids, and other heavy machines' vibrations could affect sensors, stability, and precision. In addition, the diversity of stakeholders increases the chance of untrained workers interfering in the robot operation, which could lead to injuries and inefficiency by the constant need to stop or recalculate the path movement (Burden et al., 2022). Thus, the new culture of how to build emerges with a structured and clean construction site based on the organization of distinct construction phases and logistics.

Therefore, a great development in the spatial relationship between robots and buildings is the way to achieve mobility and adaptability to global and local assembly placements in construction sites. Mobile robots are usually pointed as the solution for this spatial assembly configuration, however, it is important to note that although the movement of a robotic arm is extremely precise, the movement of a mobile platform

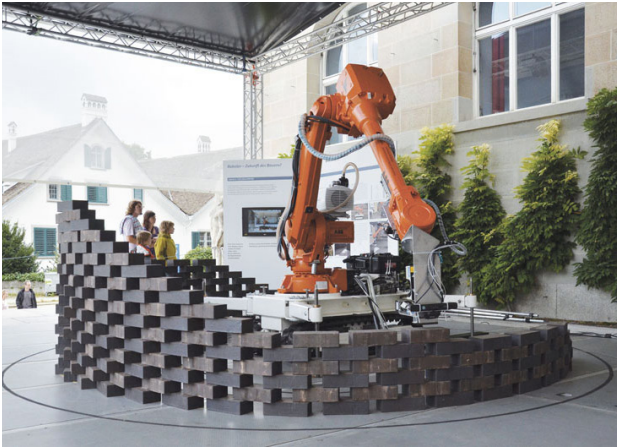


Figure 20. The Endless Wall with cognitive characteristics in which the robotic system has to respond autonomously to tolerances and adapt to changing conditions. (Gramazio Kohler Research, ETH Zurich, 2011).

is not. The accuracy rarely exceeds 5-10 mm even when laser sensors are inbuilt into the robot platform to track its position and location in space (Stumm et al., 2018). The quality of the floor, the situation of the mobile platform's tires, and the rotation and speed of the robot are some examples of aspects that cause imprecisions. One of the first examples of the implementation of this assembly process is the research project In Situ Robotic Fabrication (Fig. 19). In this project, an industrial robot was attached to a mobile caterpillar chassis, expanding its working reachability by being autonomously free to move on the construction site (Gramazio et al., 2014, p.109). The robot was able to identify autonomously its own position, the context, the construction elements, uncertainties, and material tolerances being able to produce comprehensive building structures, adaptively, within a complex environment. In the ideal scenario where the digital model is aligned with the real environment, the digital coordinates and real-world coordinates do not demand huge effort to balance them out; rather they update one another, until the point that the human becomes an indispensable factor in the robotic production. During both the design and fabrication phases, this process example is still flexible and open to human interaction in such a way that even during construction, the design can be adapted to changes in the site (Gramazio et al., 2014, p. 110).

Allowing direct collaboration with humans, the adaptability factor in construction greatly expands. The Endless Wall

installation (Fig. 20) is an example where this has been developed and successfully tested on a prototype scale (Gramazio et al., 2014, p.263). In this project, the robot is able to assemble a brick wall shaped by the recognition of a corresponding drawing motion curve designed on the ground by the human. The kinematic robot movements and sensor readings can be understood as a unified system in this framework, and can easily be adapted to a new situation with the help of modularity and new movements that can be made from the data of pre-recorded movements (Kramberger et al., 2022)

Pick and place operations are easy tasks for the robot once is known the initial and final points of placement, however, meaningful discrepancies can occur between the digital models and reality when it comes to large and heavy construction elements (Rogeaue et al., 2020). As wood has dimensional changes over time due to the sensitivity to water, even standardized component has tolerances around 1mm. Thus, these tolerances and gaps in the joints can lead the structure to large deviations stopping the robot to assemble the pieces. As a result, Rogeaue et al. (2020) point out three aspects that can be to improve the assembly process: [1] self-centering connections, [2] force-sensitive end effectors, and [3] visual feedback. Self-centering connections talk about the adaptation of the design connections to boost tolerance and progressively guide the elements to the final position. Force-sensitive end effectors express the strategy of using torque sensors to adapt the robot's position according to the measured forces. And visual feedback embraces the use of cameras and image recognition to track the robot's position in space as well as check the elements' connection alignment (Rogeaue et al., 2020). This visual function can also be attributed to humans in the process when requires intuition.

All the approaches presented so far to automating existing construction tasks seek generally to decrease the time needed in the assembly process or to increase the complexity by programming motions that cannot be made by hand. Nonetheless, these objectives have so far been in contrariety (Gershenfeld et al., 2015). In this way, the robotic discrete assembly can simultaneously address both aspects, so discrete architectural thinking is approached in the next chapter of this research.

2.5 HRC usage evaluation criteria

Seeking to understand the feasibility of implementing HRC in an automated design for building workflow processes, Loo (2022) consolidates a list of criteria for the evaluation of workflows where HRC could be implemented. As illustrated in Figure 21, the type of collaboration and quantity of tasks could emerge from an analysis of the multiple tasks involved within the proposed workflow and safety levels. (Loo, 2022). The evaluation criteria list includes five items: [1] task complexity, [2] safety level, [3] the scale of construction, [4] variability of working conditions, and [5] the complexity of assembly. These items consider aspects according to the indicated number. [1] Simple to complex. [2] Safe to dangerous. [3] Within a static workstation or larger than a robot, which requires robot mobility. [4] Leveled or uneven; constant (controlled, certain) or dynamic (uncontrolled, uncertain); uncluttered or cluttered. [5] Assembly scale, orientation, precision, the weight of individual members, and tools required (Loo, 2022).

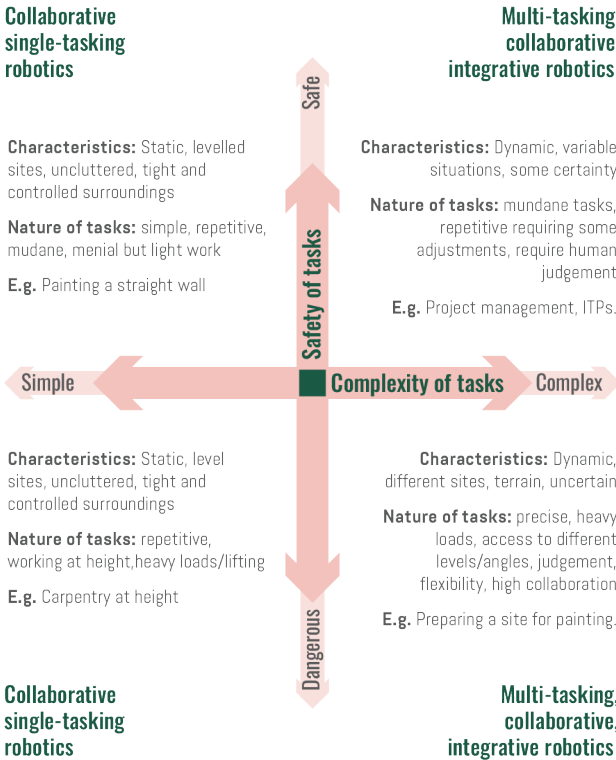


Figure 21. Tasks groups in relation to complexity and safety. Adapted from Loo (2022, p. 98)

3 Discrete Architecture

3.1 Discrete Architecture thinking

3.1.1 Discrete aggregation for mass customization

3.1.2 Combinatorial design for reconfigurable housing

3.2 Circularity in discrete components

3.3 Timber construction

3.3.1 Timber characteristics

3.4 Discrete design projects

3.5 Discrete element typology analysis

This chapter presents discrete architecture thinking in order to ground an efficient dialogue between robotic production and materialization principles. It is extremely important to realize that the issue of automation in architecture is a matter of design and not merely robots. There is no sense in automating the existing basis of construction. Once a building contains many different parts, any attempt to automate these many different processes without changing the language of communication is fruitless (Retsin, 2019c). However, the automation process becomes closer to feasible if the base organization of the building is minimized to just a few elements. Thus, the attempts to automate architecture should start with the syntax of the building and its basic building elements. In addition, this thesis understands that the designer while using digital tools is not only designing buildings but also systems (Man, 2021).

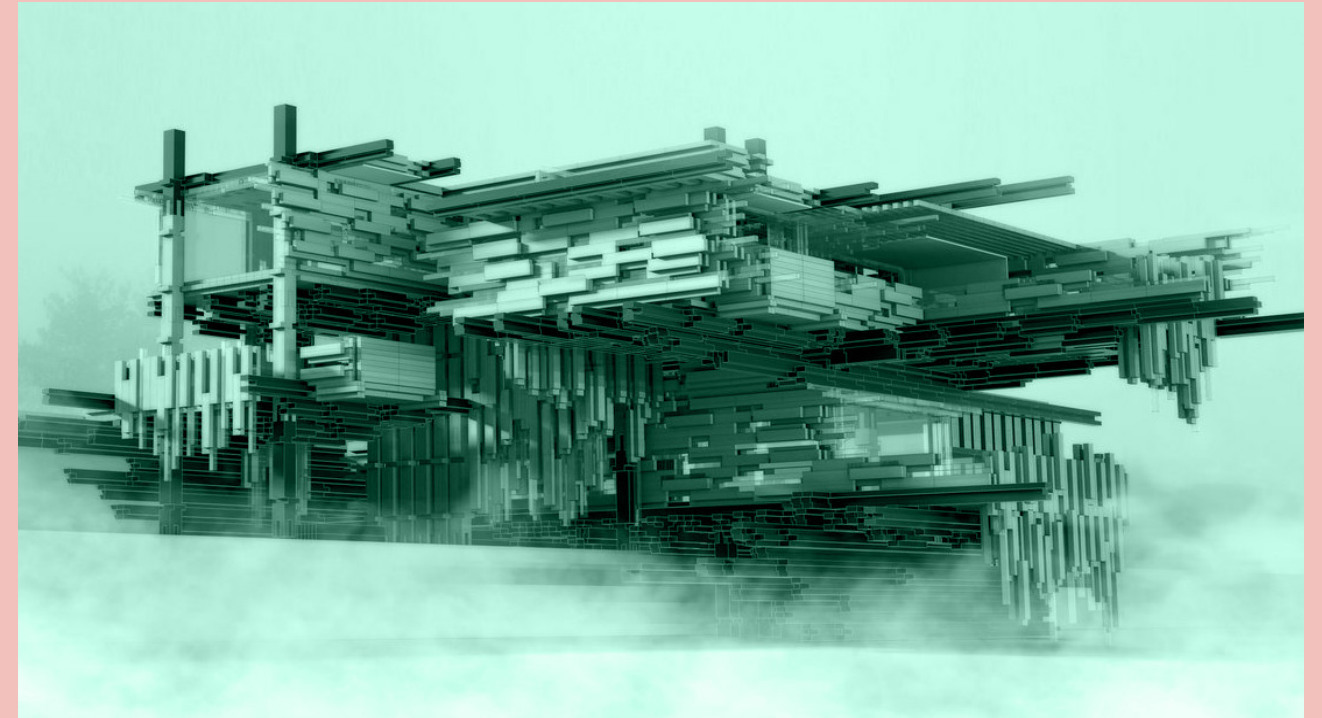


Figure 22. Discrete design of a modular housing. (Minfeng Xia, *Discrete Economies*, 2017)

3.1 Discrete Architecture Thinking

The concept of discreteness pertains to what is distinct and separate, contrasting with continuity, which refers to something uninterrupted and seamless (Retsin, 2019a). Within discrete architecture, the focus is not on designing the overall design, but rather on designing individual elements or functional units. In other words, the Discrete concept begins with the individual element, acknowledges the interrelationships among each element, and progressively extends to achieve the overall design (Chen et al., 2021). From this perspective, each individual element holds significance and contributes to the entirety of the structure, thereby altering the conventional expectations of their behavior. During the design process, the whole final geometry coexists with its parts with equal importance at the same time that each one influences the other. The design inside this paradigm is neither a simple linear aggregation nor mere subdivisions of a larger whole (Retsin, 2019b). In discrete design, there exists a parallel existence of the digital and physical realms, leading

to an algorithmic and structural procedure that exhibits a dual nature of being profoundly conceptual and tangible at the same time (Retsin, 2019b). The resolution of the used components is the driver of the approximation of a curvilinear space (García, 2019). Thus, discrete thinking is a framework seeking to reformulate the entire production chain of architecture through the boost of the notion of discreteness in digital and physical assembly (Retsin, 2019a), in which the design no longer relies on an overall figure (Retsin, 2019b).

Discrete elements do not have any fixed functional connotation (Tessmann & Rossi, 2019). In the context of discrete architecture, when elements are freely arranged during assembly, they lose the fixed connotation of traditional architectural components like columns, floors, and walls. Instead, they become generic physical units in a cyber-physical assembly. Discrete architecture can be seen as organic, as a single element and its properties have the potential to define the entire building, challenging the conventional modernist

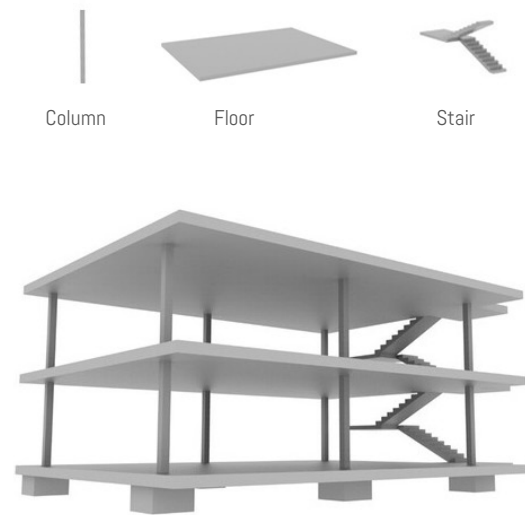


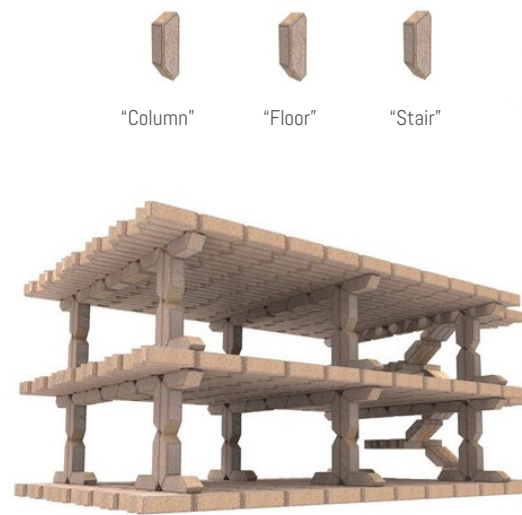
Figure 23. Discrete elements as a reassembly of Domino House. (Ivo Tedbury, Semblr, 2017)

form-function logic (Retsin, 2019d). The function of an individual component is determined by its contextual associations with adjacent elements, rather than being constrained to a static interpretation as a structural beam (Tessmann & Rossi, 2019). Figure 23 illustrates the contrast between an approach that emphasizes specific function units and an approach that emphasizes generic distributed function units. Therefore, the discrete thinking of architecture is not based on strict hierarchies between predefined parts, but rather on open and adaptable parts (Retsin, 2019c).

Sanchez (2017) highlights four key elements in the discrete design ecosystem to better demonstrate the connections between them: [1] parts, [2] links, [3] patterns, and [4] commons. The parts refer to the individual components that can be combined together. These components follow specific communication rules, referred to as links or grammar. As a result, patterns are formed through the arrangement of these basic components, creating various structures. This process of combining elements leads to the emergence of patterns. Finally, the commons represent the recognition that a substantial supply of freely available elements is necessary for such processes to occur and for experimentation to take place. These ingredients of discrete thinking also relate with what Picon (2020) says about the aspiration for efficiency whose final hope is the production of an architecture that can be of greater relevance to the current social

and environmental world situations. Thus, this research project approaches the first three items to understand how to create aggregation grammar considering the patterns for robotic production.

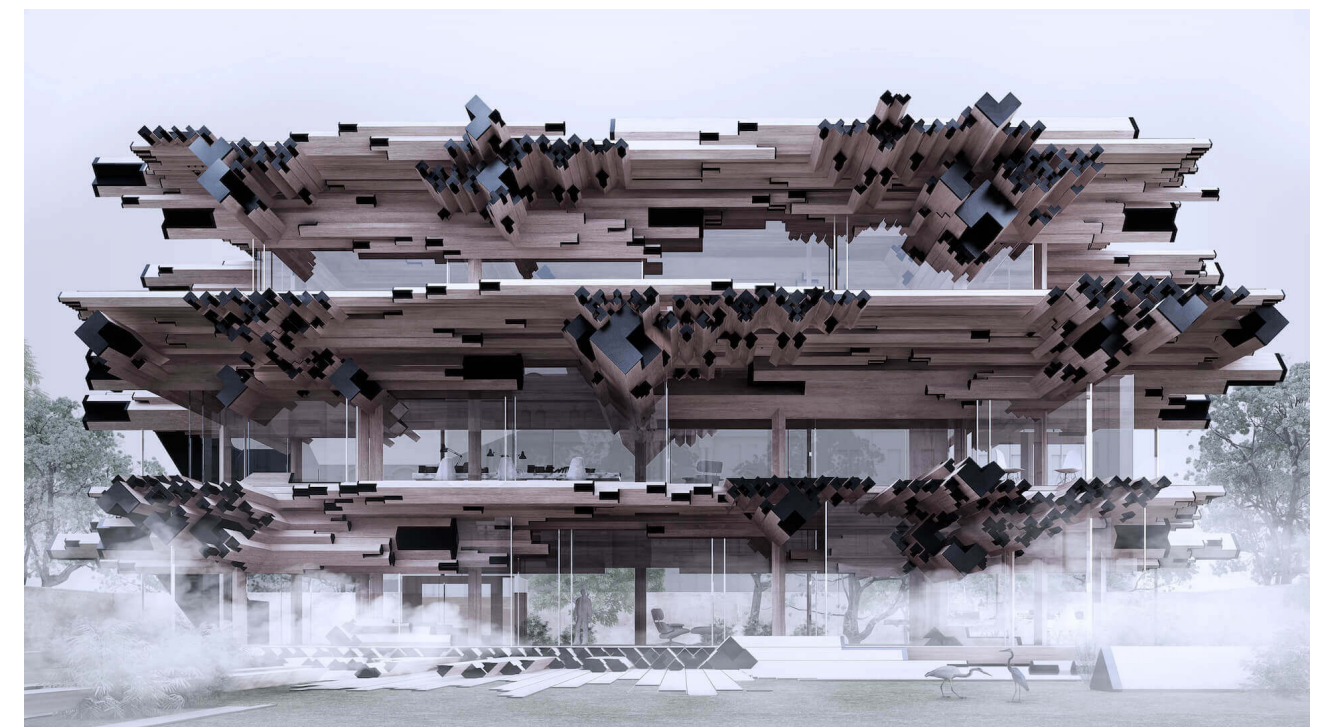
The concept of continuity is contradicted by the discrete design method, causing a shift in the focus of shape complexity from the overall design to the assembly process of individual parts. In the discrete model, independent parts are given priority and are not considered subordinate elements of a larger whole (Sanchez, 2019). In discrete architecture, the aesthetic is generated from aggregation and assembly while the individual part remains untouched, different from parametric design, in which the assembly relies on many complex unequal parts. The discrete design honors the modularity of individual equal parts in such a way that the differentiation is through patterns (Koerner-Al-Rawi, 2020). Therefore, it becomes crucial to comprehend the various architectural assembly configurations that discrete aggregations can generate for achieving mass customization through combinatorial design. As architectural systems evolve to establish compatible connections between individual elements, the combinatorial design approach enhances our understanding of how parts can logically and physically connect with each other in both digital and physical environments (Wysocki, 2021). Fabrication restrictions can be inserted in the early stages of the design process leading to an efficient design for



building workflow (García, 2019). In this sense, the representational gaps between physical and digital realities no longer exist due to what is assembled is what is computed and vice-versa, enabling increased automation of construction while requiring minimal handling on-site (Retsin et al., 2020). Thus, the designer does not design unique function-based objects but rather becomes a designer of generating systems that are able of generating many other components (Sanchez, 2020).

In order to discuss the concept of a generative system, its notion can be generalized in a set of parts together with rules of combining them to shape feasible objects (Sanchez, 2020). Inside this open-ended logic, it is important to clearly identify three aspects: [1] the holistic behavior to be focused on; [2] the parts within the object, and their interaction with themselves, which cause the holistic behavior, and [3] the way in which this interaction causes the holistic behavior (Sanchez, 2020).

Figure 24. Discrete architectural design for a multi-family house. (Gilles Retsin Architecture, Diamonds, 2016).



3.1.1 Discrete aggregation for mass customization

According to Celani et al. (2006), the process of developing an element of grammar begins by defining the vocabulary of shapes, spatial relationships, rules, and an initial form. In this approach, the first step involves identifying a finite set of basic shapes that will be used in the grammar, followed by establishing the desired spatial combinations between these shapes. Transformation rules of a specific type are then defined based on spatial relationships. To initiate the application of rules, an initial form is selected from the set of shapes. Once all the elements are defined, the rules are successively applied to interact with the initial form until the desired aggregation is achieved (Celani et al., 2006). The discrete element design serves as the foundation for the aggregation process.

The outcome of the aggregation process is a structure that consists of discrete elements. The form of the aggregation relies on the methodology of utilizing basic design elements (Xiao et al., 2020). However, it is the definition of the interacting connections between physical components that plays a crucial role in deciding and ensuring the suitability of the discrete elements (Wysocki, 2021). Considering that as the number of different geometrical interface types increases,

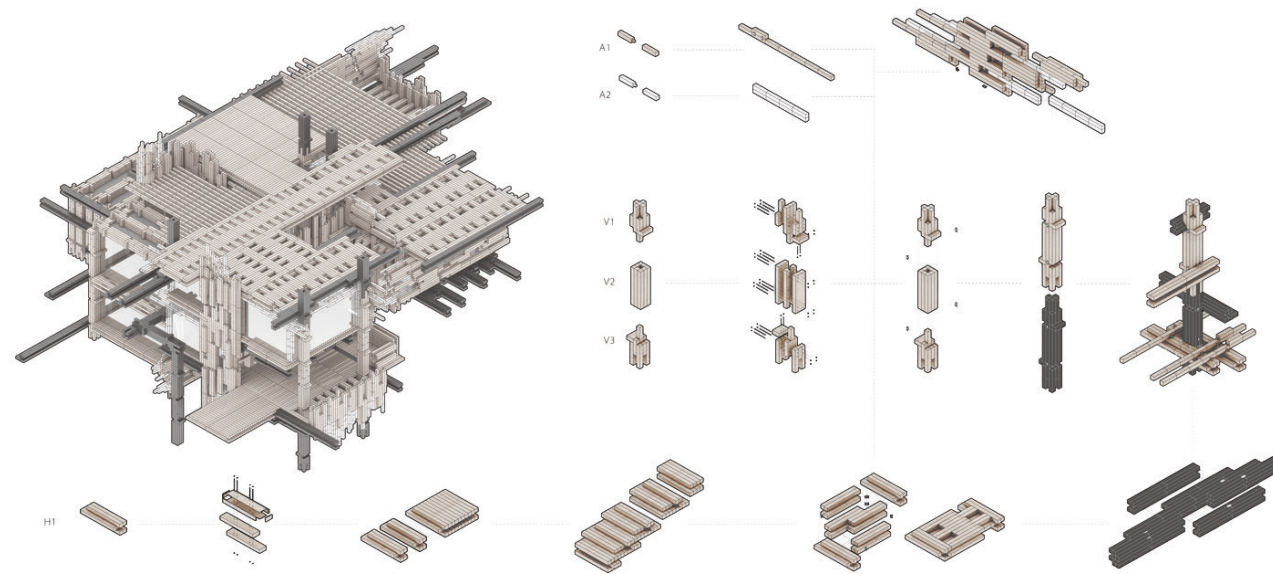


Figure 25. Discrete aggregation study involving serial repetition, pattern languages and combinatorial tectonics. (Minfeng Xia, *Discrete Economies*, 2017)

the flexibility to create various configurations decreases. Wysocki (2021) defends that the most favorable form of connection is sectional, wherein each element possesses an identical type of interface, eliminating the necessity for a single part to connect all others. Consequently, simpler connections allow for greater flexibility in configuring components. Once a set of dimensions is established, it enables the creation of a system that facilitates effortless spatial arrangement.

Xiao et al. (2020) describe two approaches to understand the underlying principles of aggregation. [1] The first approach involves converting a shape into a voxelized geometry by dividing it, while [2] the second approach entails establishing a set of rules that dictate the generation of the aggregation pattern. The process of voxelization encompasses converting a solid shape into a voxel-based representation, followed by the creation of an aggregated structure. This technique grants designers the authority to control the borders of the resulting aggregated structure. On the other hand, the aggregation-by-rules process involves generating a group of geometries within a predetermined domain scale, thereby increasing the opportunity to control the generation process by setting specific size limitations (Xiao et al., 2020). Thus, this research intends to use an equilibrium of both methods in the generation of the proposed structure.

Considering these two aggregation methods, the Wasp plug-in for Grasshopper is designed to generate discrete assembly designs by directly utilizing the geometrical characteristics of the components to be joined together, rather than relying on a pre-defined grid (Rossi & Tessmann, 2019). It offers resources for simulating and evaluating performance and production constraints during the process of combining components, effectively merging design and manufacturing considerations. Additionally, it presents methods for establishing hierarchical structures, enabling the incorporation of parts of varying sizes and levels of detail within the same combinations and transitions between scales (Rossi & Tessmann, 2019).

According to Xiao et al. (2020), when applying discrete thinking to the physical environment in the context of mass customization construction, the digital designing of aggregated components necessitates an extensive understanding of techniques for connecting them and the procedures involved in their fabrication. These aspects should be integrated into the overall design concept. Numerous factors impose limitations on the fabrication of physical elements, such as material properties, assembly complexity, structural integrity, time constraints, financial constraints, and practical considerations in the fabrication process.

3.1.2 Combinatorial Design for reconfigurable housing

"[...] combinatorial design is a design strategy that starts from the definition and individuation of parts, describing an open-ended series of relations with one another. These parts are coupled and aggregated to generate larger assemblies, describing meaning, performance, and function at different scales of configuration. The system always remains open-ended and malleable, allowing for the replacement of parts within it. The open-endedness of the system implies that there is no possible optimization, as the solution space of permutations grows with each unit added at an exponential rate, becoming computationally impossible to search for an optimum" (Sanchez, 2020, p. 79).

As discussed earlier, in a discrete model, elements gain independence and form a self-contained system that can exist separately from the overall design. These components have the ability to expand or contract based on their surroundings, allowing them to adapt to their specific context (Sanchez, 2019). This concept, when applied to housing spaces, allows for the flexible arrangement of components and the potential for them to be easily reconfigured or reversed using modular assemblies. The individual elements must be designed to fulfill not only a singular function but also multiple possible interactions with other parts. By utilizing serialized elements, the focus shifts toward design based on patterns and combinations. In this approach, each assembly represents a temporary state rather than a final product. (Sanchez, 2017). In this scenario, patterns function as a structural representation of architecture and emerge from the geometric concept of elements. As a result, they can be shared in a digital and social manner.

In a situation where architecture requires buildings that can be easily transformed and reconfigured, the assembly system must have the capability to share components throughout the lifespan of the building. This means that building components should be able to be taken apart and reused for another structure. This objective is challenging to accomplish using modernist and parametric design prin-

ciples, where each element has a specific position in the assembly. However, it becomes feasible when discrete parts are identical. Therefore, a generic element can be removed from one configuration and placed in a new assembly setup. (Tessmann & Rossi, 2019).

Furthermore, certain discrete features, such as the ability to reproduce the components and their restricted interconnection possibilities, facilitate a rapid construction procedure (Retsin, 2019d). These characteristics also aid in minimizing the need for extensive manual labor at the construction site, unlike the labor-intensive assembly of bespoke forms. The focus lies on the arrangement of space using standardized elements, where the arrangement of parts and their geometry shape and organize the construction and operational processes (Nourian, 2020). This approach systematically generates the final design entity based on the element's configuration and geometry.

The discrete approach for housing spaces offers future occupants the opportunity to personalize their final design by selecting and customizing sets of elements through a collaborative grammatical process (Azadi & Nourian, 2021). Despite the personalization aspect, this approach also leverages the advantages of large-scale housing economies, as it utilizes a limited number of element configurations.

3.2 Circularity in discrete components

Combinatorial and reconfigurable assembly processes have been enabled by many emerging innovative solutions in the design of discrete elements, in which robots and building elements are part of a harmonious construction process for buildings that permits disassembly and reassembly (Kunic et al., 2021a). This automated reconfigurability aligned with timber constructions aims to shift the linear and deterministic logic of construction to a circular and open-ended paradigm of construction. Reversibility is the main factor in permitting the circular use of resources (Kunic et al., 2021b). Thus, the research-by-design stage of this thesis implements it not only in the design but also in the workflow proposed taking into consideration demountable joints. The architectural thinking of circularity involves that materials, components, and entire systems can be reused to extend their life cycle while at the same time minimizing their environmental impact (Kunic et al., 2021a). In this way, the design for circularity demands understanding how these materials, components, and systems involved can be designed considering reconfigurability over time. The robotics applied in architectural processes has a significant role in this circular material agenda by enabling automated constructions from discrete elements reconfigurable in time.

The modular movement has its roots in how standardization can lead to uniform and efficient production, while discrete design thinking is still in the generative phase of the design aiming for a distributed and open-ended model of production (Sanchez, 2020). The utilization of combinatorial design techniques and discrete thinking, as mentioned earlier, has the potential to promote the development of durable and adaptable building components that have a long lifespan and minimize waste. As mentioned previously, within the context of discrete architecture, the act of assembly itself constitutes the building, and all the necessary instructions and information for both assembly and disassembly are contained within the structure. Consequently, the physical structure of the housing building becomes transient in nature, serving as a reservoir of materials for future construction. Its portable elements can be dismantled and reconfigured into various forms, allowing for adaptation to evolving needs and requirements over time. This permissibility has important ecological implications because the building blocks can be continuously reused in other buildings (Retsin et al., 2020).

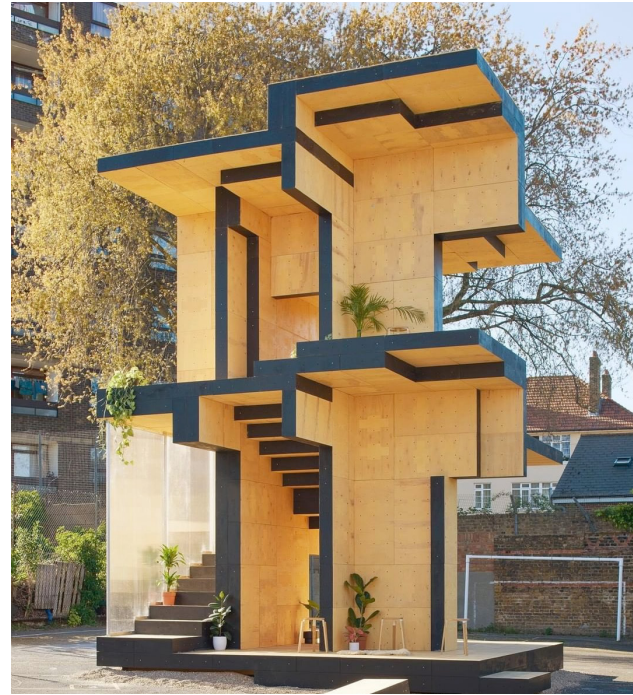


Figure 26. Temporary housing prototype consisting of a discrete framework for housing production. House Block. (Automated Architecture (AUAR) Labs, UCL, photo by James Harris, 2021)

When considering the fabrication aspect in relation to housing, it is crucial to consider an appropriate grid system. The organization of element dimensions should be based on a rectilinear modular coordination system, which ensures that spaces are aligned in a rectilinear manner. This facilitates collaboration with other suppliers and enables the sharing of environmental responsibilities. (Wysocki, 2021). Ergonomics is also a significant aspect when designing an architectural project, especially for housing. Wysocki (2021) argues that several basic dimensions such as the 90-120cm width of corridors inside dwellings should be taken into account in the design of the discrete element. The optimal grid size for this purpose would be 30x30cm, which not only accommodates the human scale with ergonomic dimensions but also includes measurements such as 0.6m, 0.9m, 1.2m, 1.5m, 1.8m, and 3m. In addition, the grid should also take into account the spatial regulations for stairs that require finding a multiple of the step module size that is an integer number, ensuring that all riser heights can be created accordingly (Nourian & Azadi, 2021). Thus, according to embracing modularity in timber discrete elements, the distance between the connections also needs to follow the standardized logic of



Figure 27. Robotic reversible timber beam. (CREATE group, SDU, 2021)

the chosen grid dimensions in such a way that regardless of where the connections fall in space, it is going to be able to line up with the connection of the other elements (Koerner-Al-Rawi, 2020). Once the element contains the logic of the overall grid, the structure can grow and expand in any direction without the need to customize any element. These approaches minimize waste and cost in flexible spaces while promoting spatial fluidity in designs by allowing users to modify their surroundings (Koerner-Al-Rawi, 2020).

The creation of structural reliable connections is what still makes automated assembly challenging (Hansen et al., 2021). The potential for material circularity in discrete elements decreases when a not demountable connection solution is used, such as glue and nail. Being able to reassemble the structure many times by creating a workflow about a dry connection expands the potential of the structure and material reuse (Koerner-Al-Rawi, 2020). At the first moment, a bolted connection seems to be ideal for reusability. Mechanically speaking, it transfers shear forces by dowel action and then by rope action. However, this type of connection had the disadvantage of concentrating the axial

stress around the bolt which can result in large displacements among connections (Hansen et al., 2021). Considering this, as presented before, Kunic et al. (2021b) successfully developed a robotic assembly and disassembly workflow of timber elements based on reversible connections (Fig. 27). The connection method was based on some important premises: **[1]** layered assembly performed by only one robotic arm, **[2]** self-alignment connection strategy, **[3]** circular reuse of the material by disassembly, and **[4]** moment and shear force-proof connections to support many applications.

The significance of rethinking the current system to a distributed mass production of units that operate under combinatorial rules is not only formal but also socio-economical (Sanchez, 2020). Circularity does not only include environmental impact aspects, but also economic and social. It is important to understand how scalable the discrete design is for a long-term life cycle while adopting modularity and reuse.

According to Picon (2020), the current architectural materiality has two dangerous obstacles, from which the discipline needs to distance itself in the future, the **[1]** naive gesticulation, and **[2]** the desire for uncompromising objectivity. Thus

architecture needs to maintain in light of the growing social challenges (Picon, 2020), such as the housing crisis and climate change. According to Sanchez (2019), the current approach to production and costs follows a linear trajectory, favoring only those with significant financial resources. On the other hand, Discrete thinking acknowledges the significance of scaling economies and promotes standardization as a means to accomplish mass customization. This approach does not seek to return to the production of identical units but rather emphasizes the combination and permutation of purposefully designed components to achieve customization, adaptation, and flexibility while adhering to economic scalability principles.

The discrete viewpoint further recognizes the societal and economic impact of repetitive elements, presenting an alternative thought process, especially in light of the increasing worldwide need for housing. Additionally, discreteness takes into account the collaborative endeavors of individuals and collectives engaged in contemporary architecture, presenting the potential for a participatory framework for collective creation. This approach emphasizes the design of adaptable systems, advocating for a collaborative economy and introducing a fresh role for social engagement in the field of architecture (Sanchez, 2019). Inside this circular logic, the housing market could open itself to a larger group of builders due to the speed and financial accessibility achieved from a small-scale infrastructure to construct building elements (Retsin et al., 2020). As the structures of the house could be assembled, disassembled, and adapted faster, modes of ownership could be also questioned. This encourages the hidden ambition of developing an infrastructure that permits the growth of a new geometrical vocabulary for decentralized forms of production (Sanchez, 2017).

Having said all that, this research takes modularity aligned with an ergonomic grid as a starting point for achieving circularity levels in reuse and reconfigurability, which requires principles of design for disassembly. By doing so, the material life cycle is expanded storing carbon for longer. In addition, the more generic is the discrete element, the better for achieving circular levels because it can be used for several other applications.

3.3 Timber construction



Figure 28. BetaPort. Scalable on-demand building system materialized by a fully automated planning process and ecologic construction methodology. (Urban Beta, photo by Naara, 2020)

The availability and natural renewability of wood, together with its workability and long tradition of craftsmanship, made it one of the most commonly used materials in history until the beginning of industrialization (Menges, 2016). However, this dominance of wood was replaced by other modern materials such as concrete and steel with a promise that they would better fit in the new Modernist methods of production because these were developed and produced to attend to the specific demands of the building sector. Wood's medieval techniques had struggled to adapt to the new reality with the down of mechanization especially because of the intrinsic heterogeneity and biological properties of wood to attend regulations of standardization.

Today, however, the wood goes through another meaningful change regarding the renewed interest as a construction material (Menges, 2016). The main reason for that is the need to reduce significantly the amount of environmental impact created by the building industry. Timber construction provides a great opportunity to decrease its carbon footprint, reduce waste and minimize the usage of non-renewable materials (Wagner et al., 2020). Even heavily industrial wood processing is taken into account because of the properties of wood in a low level of embodied energy and a positive carbon footprint. It is also a natural, recyclable, and renewable resource with a greatly differentiated internal structure. Wood can be seen as a cellulose-reinforced composite material due to its cellular microstructure with a large percentage of cavities, having the best load-bearing capacity of heat-insulating materials (Menges et al., 2016). It is lighter than steel for the same tensile loading capacity and it has almost the same compressive strength as concrete while offering a significantly smaller heat transfer coefficient. It is easy therefore to affirm that wood is recognized as one of the best environmentally friendly and energy-efficient materials currently available (Menges et al., 2016).

Timber construction is a significant sector of the building industry that presents great potential to solve the huge material consumption of construction and the large demand for affordable housing (Kaiser et al., 2021). As nowadays wood is the main material in construction capable of growing, its harvest increases the carbon confiscation in forests and its use as construction material represents carbon reservoirs (Wagner et al., 2020). Mass timber construction is part of the present for already couple of years in architectural projects such as wood columns, beams, and panels. Due to their lightweight and quick fabrication, cross-laminated timber and mass plywood panels are frequently used in construction enabling through standardization fast assembly, minimum labor, and small demand for storage space (Koerner-Al-Rawi, 2020). However, it is important to mention that even for such standardized elements, fabrication tolerances are usually around 1 mm because of slight dimensional changes over time due to hygrometric variations (Rogeaau et al., 2020).

3.3.1 Timber characteristics

One of the main keys to designing using timber is the understanding of the anisotropic characteristics of the material (Loo, 2022). It has different mechanical strengths in different directions. Figure 29 demonstrates that within the simplified representation of tubes running along the length of a tree, wood exhibits considerable compressive and tensile strength when aligned parallel to the fibers. However, it exhibits significantly lower stiffness when oriented perpendicular to the fibers (Loo, 2022). The effective transfer of loads in timber design is greatly influenced by aligning the force flow along the fiber elements, which has a significant impact on design considerations. Thus, the geometry of the timber discrete elements needs to follow the adaptability to the loading conditions of the design.

According to extend the lifespan and durability of timber elements is important for the design to address more than [1] reversible connections and [2] design for disassembly, but also conditions for [3] fire safety, [4] moisture management, [5] drainage, [6] ventilation, and [7] biological agent treatments by thermal or chemical processes. Timber is a natural material that requires being dry before its construction application avoiding every possible presence of moisture to consequently avoid deterioration by fungi and other organisms (Loo, 2022).

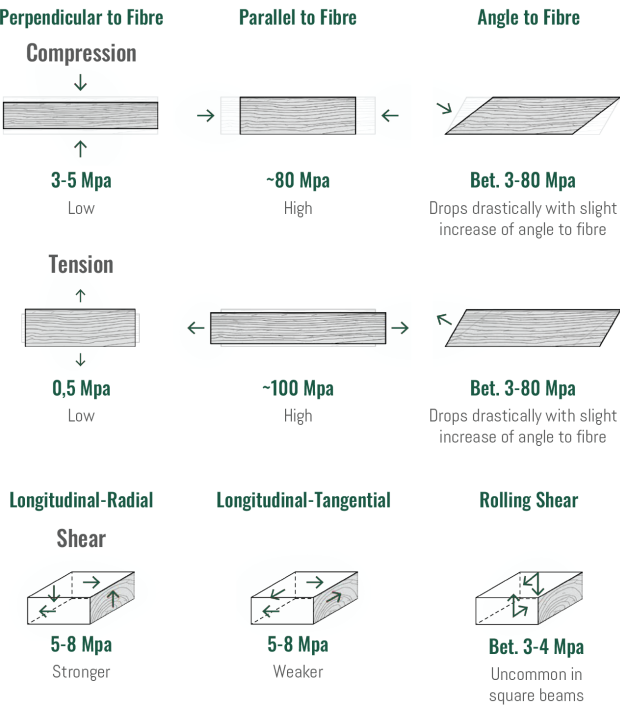


Figure 29. Overview of timber's strengths. Adapted from Loo (2022, p. 14)

3.4 Discrete design projects

To incorporate the latest advancements in digital design technology, this research reviewed several recent projects in discrete architecture and robot manufacturing. The purpose of this review was to gain insights and learn from these projects in order to design housing that aligns with the research objectives. Thus, this section approaches a brief overview of some of these relevant discrete projects, which are [1] the Tallinn Architecture Biennale pavilion, [2] the Coeda House, and [3] the Kodama pavilion, and [4] the Assembler Assemble project.

[1] The Tallinn Architecture Biennale pavilion (Fig. 30) by Gilles Retsin Architecture proposes a structure from a series of discrete building blocks that could be assembled into a variety of structures. These discrete building blocks were based on cheap locally available sheets of 18mm exterior plywood, cut by a CNC machine (Gilles Retsin Architecture, 2017). Each cut sheet was assembled into stiff building blocks with a shape variety of straight, 45 degrees, 90 degrees, and 135 degrees elements capable of bearing structural loads. The design of these building blocks aimed the performance in any structural condition, such as under compression, tension, as a cantilever, or as a column. Thus, 80 building blocks were

produced with black bitumen tar or varnish in order to protect from the exterior weather with around 380 m² of plywood sheets (Gilles Retsin Architecture, 2017). Local manufacturers fabricated the structure locally.

[2] The Coeda House (Fig. 32) by Kengo Kuma is a project that bases on randomly stacking 8cm square cedar beams. The structure has a tree-like shape with one single trunk with huge branches enabled by a reinforced carbon fiber rod (Kengo Kuma & Associates, 2017). Due to the tree form, perimeter columns were eliminated for an unobstructed landscape view.

[3] The Kodama pavilion (Fig. 33) from Kengo Kuma & Associates office. In this project, the assembly of solid larch pieces created a 4-meter-high spherical shape pavilion. The structure was assembled without using any metal fitting or glue, in which each discrete element had a thickness of 58mm and a tolerance of 1mm (Kengo Kuma & Associates, 2018). It was about an experiment with a new methodology of construction that proved how a large space could be built from the work of humans with small-discretized components.



Figure 30. The Tallinn Architecture Biennale pavilion. (Gilles Retsin Architecture, 2017).



Figure 31. Assembler Assemble project. (Man, 2021)



Figure 32. Coeda House (Kengo Kuma & Associates, photo by Kobayashi Kenji Photograph Office, 2017)

And finally, [4] the Assembler Assemble project (Fig. 31) by Nguyen Xuan Man is part of the Research Cluster 4 of The Bartlett School of Architecture. The project approaches automation in the architectural context of housing and proposes not only a design but also a system of production (Man, 2018). It introduces automating the construction by modularizing the assembly process and the robot responsible for it. The design of the proposed robot is identical to the discrete construction element, reducing the entire housing system to just two types of units, the active robots, and the passive construction elements (Man, 2018). The project also developed a computational framework that uses the advancements in machine learning and robotics to propose housing construction in a completely autonomous method.



Figure 33. Kodama pavilion. (Kengo Kuma & Associates, 2018).

3.5 Discrete element typology analysis

In order to understand which is the most suitable typology of discrete elements to design a discretized construction system, this research looked to categorize some already-developed discrete elements in practice and research based on their materialization aspects. Once these physical-geometrical typologies were identified, an analysis was made evaluating the pros and cons of each type depending on the requirements criteria of each area of knowledge covered in this research. The analysis criteria are also the design goals that this research aims for in discrete element design. Thus the analysis evaluated how suitable are each discrete typology according to the following aspects:

- [1] human-robot collaboration,
- [2] component’s mass for humans and robots,
- [3] precision,
- [4] fabricability,
- [5] ease of assembly and disassembly,
- [6] ease of reconfiguration,
- [7] connectivity,
- [8] material consumption,
- [9] insulation, and
- [10] geometrical orthogonality

The typologies and their pros and cons are expressed in Figure 34. Overall, the open-ended discrete design approach has a tendency to consume large quantities of material in its architectural aggregations because of its genericness nature. However, the possibility of easy reconfigurability and reusability compensates for this issue by expanding the life usage of the components. Therefore, the final geometry of the element should also take into consideration a balance between genericness and function-based design by an efficient symmetric composition.

Hollow discrete geometry is lightweight and can make use of already-standardized materials, but it also demands more energy and time of production to make the smaller pieces of the discrete component than a solid geometry that requires just basic cut procedures. The production could also be more expansive once they have more procedures embedded in the design. In addition, the solid geometry provides better insulation in aggregations with fewer cavities. Hollow geometries start to be better than solid geometries in huge discrete components that are out of the human scale, which is not the case in this research.

Therefore, after the comparison, [1] solid blocks, [2] solid plates, and [3] orthogonal beams are the most suitable typologies that attend the criteria. None of them is perfect, but a good balance between the requirements and design is needed to achieve the ideal scenario of performance. The orthogonal beam type was selected to continue the research.

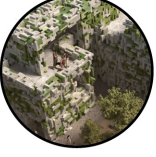
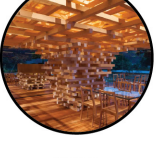
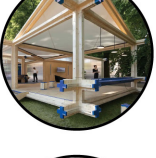
types		pros	cons
solid blocks		- worldwide known - easy production and assembly - human scale lightweight - new materials studies - good insulation	- dense use of material - usually fixed by mortar - size deviation in some materials
hollow blocks		- material efficiency - human scale lightweight - usually reversible connections - good insulation	- some sliding connections - many smaller parts - require high precision
solid-bar blocks		- covers larger areas placement - can have dry connection by weight - new materials studies - good insulation	- dense use of material, heavy - usually fixed by mortar - size deviation in some materials
hollow-bar blocks		- material efficiency - lightweight - usually reversible connections - good insulation	- some sliding connections - many smaller parts - require high precision
solid plates		- simplification of parts - material efficiency - easy production and assembly - usually reversible connections - human scale lightweight	- some sliding connections - require high precision - bad insulation when many cavities
hollow plates		- material efficiency - lightweight - usually reversible connections - good insulation	- some sliding connections - many smaller parts - require high precision
orthogonal beams		- simplification of parts - material efficiency - easy production and assembly - usually reversible connections - human scale lightweight	- some sliding connections - require high precision - bad insulation when many cavities
shape-specific beams		- material efficiency - easy production and assembly - usually reversible connections - human scale lightweight	- function based design - some sliding connections - many smaller parts - require high precision - bad insulation when many cavities
complex blocks		- geometry diversity - engaging aesthetic - organic appealing	- specific design - complex assembly logic - some sliding connections - require high precision - bad insulation when many cavities

Figure 34. Discrete element typology analysis. (by Author, 2023).

4 Architectural Design

- 4.1 Discrete element design
 - 4.1.1 Connection design
- 4.2 Discrete elements aggregations
- 4.3 Generative design approach
- 4.4 The site
- 4.5 Housing platform's program
 - 4.5.1 User's interface app
- 4.6 Discrete timber housing design

From the theory discussed in the previous chapters, this section presents the design proposition of the structural construction system. It involves the discrete element design, its aggregations' possibilities, generative design approaches, and the timber housing design with its program and site analysis. Programming can optimize the entire structure by linking formal, constructive, and fabrication parameters. This level of complexity cannot be managed by conventional manual design techniques. The use of digital design and fabrication processes becomes necessary when there is a "critical mass" of construction components with mutual dependencies. (Gramazio et al., 2014, p. 187) In this case, the design is described by a set of programmed rules that allow for seamless adaptations, even at the late stages of the design process.

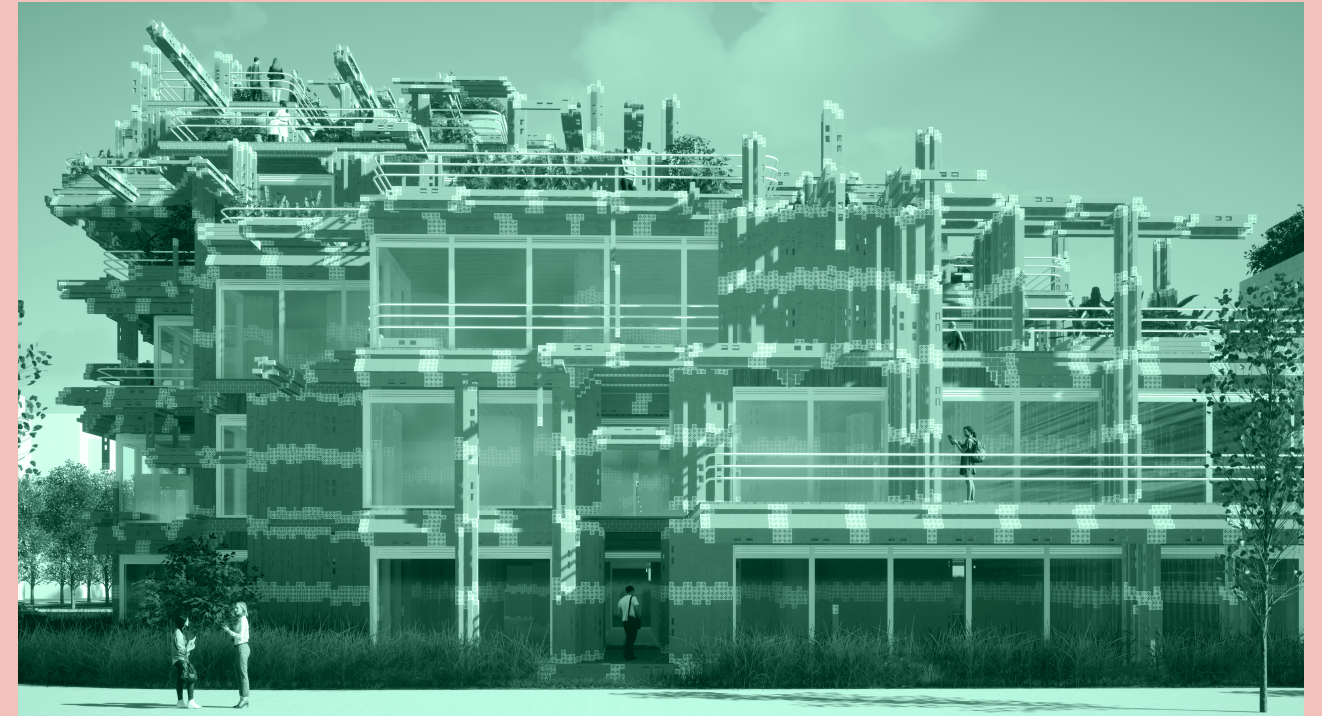


Figure 35. Discrete element aggregation shaping the housing structure (by author, 2023).

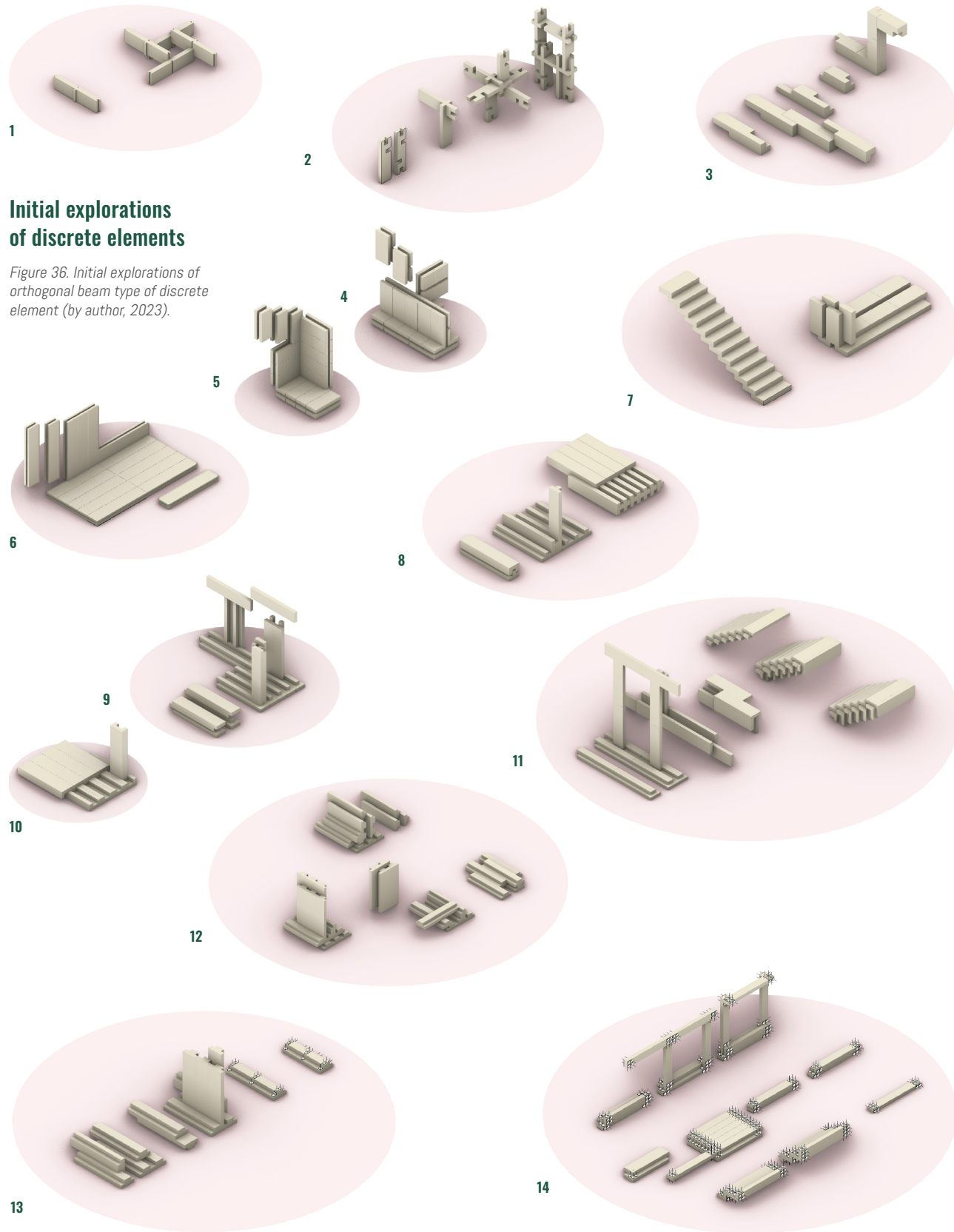
4.1 Discrete element design

Everything written so far are key characteristics that significantly guide the concept of the structural system, the housing design, and its construction process. The first consideration that must be taken into account is the geometry of the elements to be assembled (Gershenfeld et al., 2015). The discrete element is the starting point of the design proposal. It is the base unit of the design as a whole. From the analysis of the discretized parts typology, the solid orthogonal beam type was selected to continue the research by design. Inside the building categories, the discrete element was designed to resist structural loads, so it is mainly part of the structure category, but it can also be used as a skin and space plan.

Design-for-disassembly and **design for future reuse** are the main design driving forces concepts. The research understands Design-for-disassembly as an approach that aims to optimize the ability of a product to be easily taken apart and separated into its component parts at the end of its useful life. Its main principles are **[1]** modular design, **[2]** standardized fasteners,

[3] minimizing adhesives, **[4]** designing for easy understandable access, and **[5]** clear labeling and documentation. These principles help to ensure that disassembly is done correctly and efficiently. This can also aid in the recycling or disposal of individual components, as they can be easily identified and sorted.

Initial explorations of discrete element possibilities were made to understand their connectivity. The connection between elements is fundamental to guarantee discretization and circular goals as well as enabling the growth and expansion of the structure. Inside the orthogonal beams category, the design experimentation led to beams that connect themselves from four faces at each end (Fig. 36). This type of connectivity allows the elements to aggregate horizontally and vertically on the same axis as well as perpendicularly to the previous element. Thus growth, reconfigurability, and interlocking are guaranteed from one piece.



Initial explorations of discrete elements

Figure 36. Initial explorations of orthogonal beam type of discrete element (by author, 2023).

Figure 38 expresses the discrete element design workflow. It took the wood material as an input of design regarding the relation among density, volume, and the maximum weight that a human can take comfortably. According to the Dutch Working Conditions Decree (Arbobesluit), workers should not lift or carry loads that weigh more than 25kg without the use of lifting aids, such as cranes, hoists, or trolleys. However, the research uses **23kg** for calculation to attend other countries that this limit may be lower. The chosen wood density for calculation was 700kg/m^3 looking for embracing the world widely most types of wood while enabling future local-based adaptation. The grid chosen was $10\times 10\text{cm}$ due to allow easy integration with other standardized systems and to allow easy volumetric symmetry for the aggregations in space. Taking 10cm^3 as a voxel unit, the discrete bar was dimensioned by adding this volume to shape the initial geometry. From this sequence of thinking, the **$10\times 20\text{cm}$ section** was chosen because offers the opportunity to structurally perform in different scenarios when positioned horizontally or vertically. It also enables the section to interlock within different patterns throughout its symmetry. Although the discrete element is detached from a performative function, the beam type with a rectangular section gives the chance to be used to achieve span with fewer connections once its design is slightly functional-oriented. The railway sleeper (Fig. 37) is an already existing standard wood product with similar geometry with dimensions of $10\times 20\times 240\text{cm}$.



Figure 37. The railway sleeper (Mitchell Turf, 2023).

Discrete element design workflow

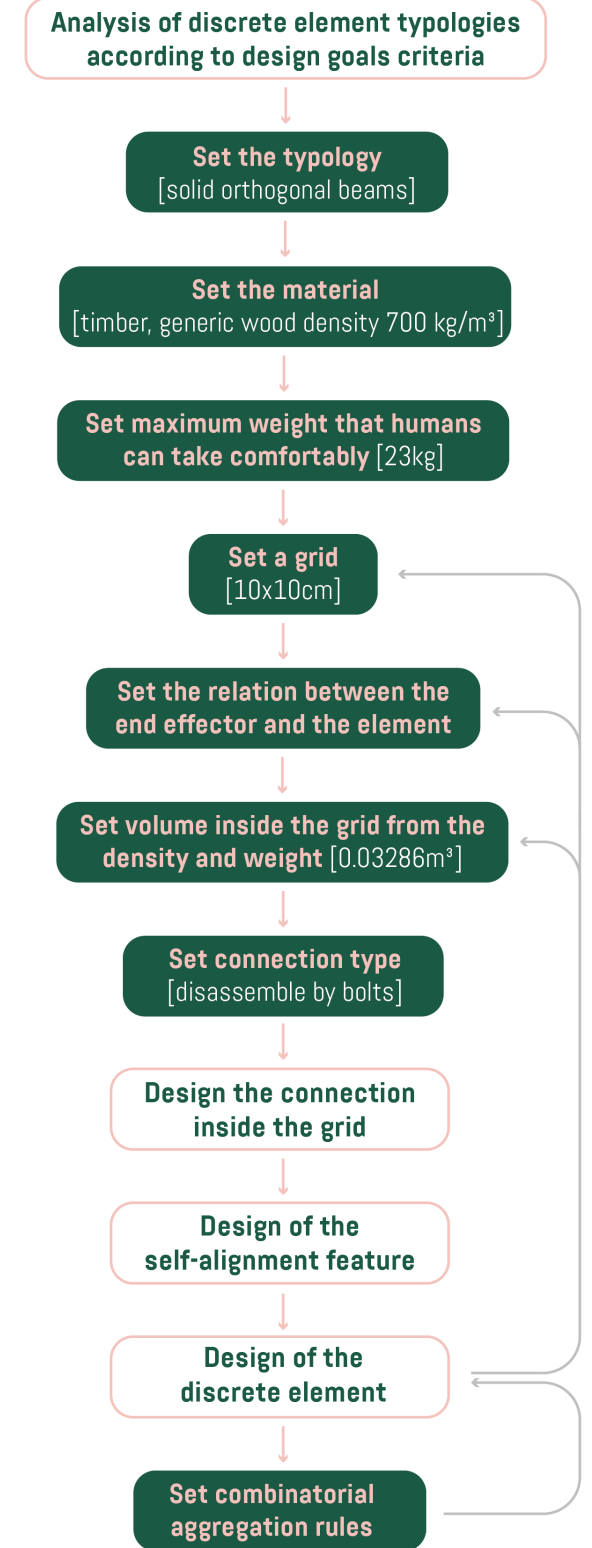
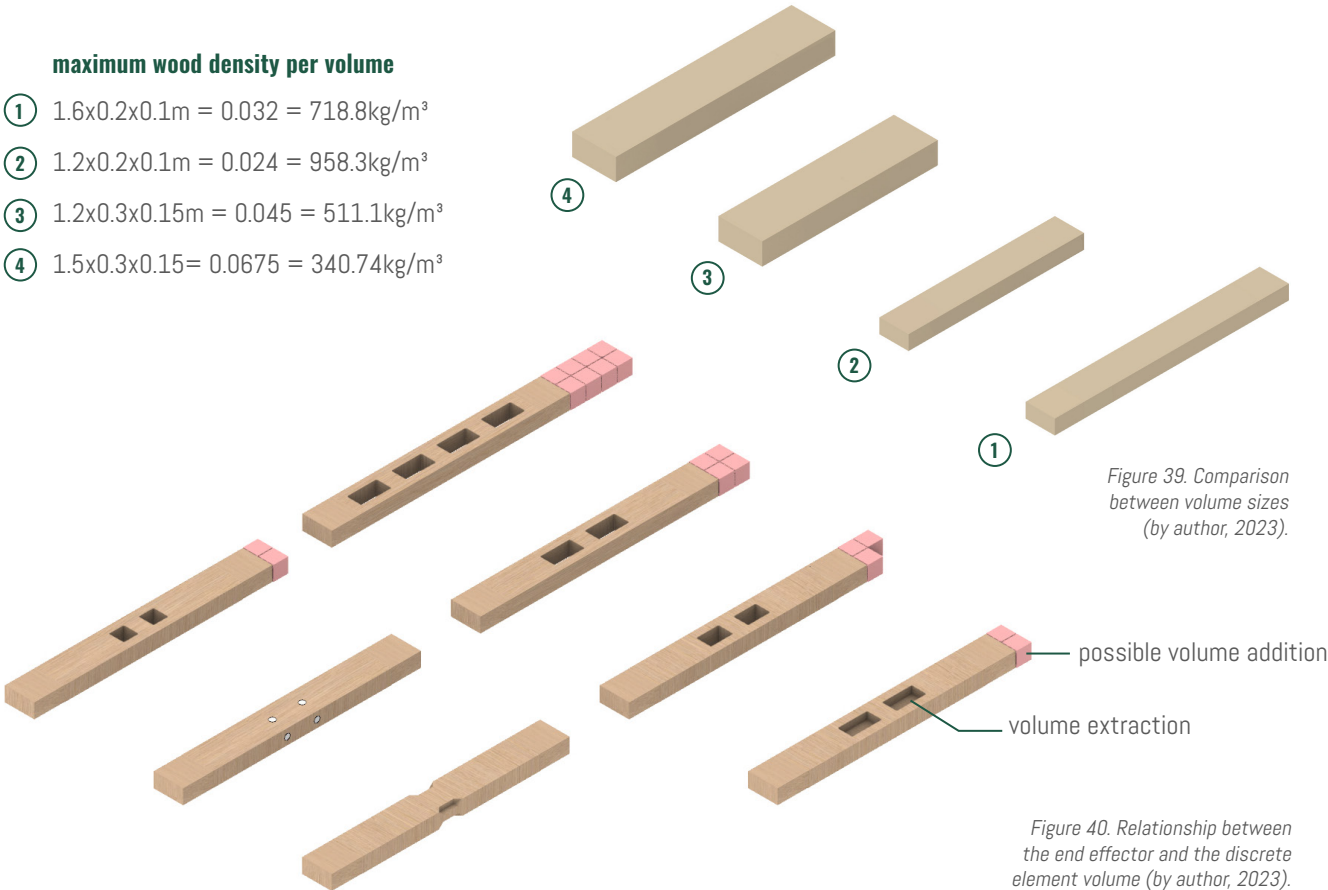


Figure 38. Discrete element design workflow (by author, 2023).

Once set these items, the volume calculation was made by adding 10cm³ inside the 10x20cm until gets the 23kg that a human can take. The length of 160cm showed as the most suitable for the design proposition because it is inside of the 700kg/m³ of type of wood while under 23kg. Another similar section of 15x30cm was tested to compare as well as with different lengths of 120cm and 150cm as illustrated in Figure 39. The observation is that although the shorter length of the wood beam is more efficient for transportation, it also requires more connections in the assembly of the whole aggregation design, consequently being more expensive for construction. So to keep the balance among density, volume, weight, and the longest length, 160cm was chosen.

The relation between the robotic end effector and the discrete element is also important to involve in the initial phases of design. As the connection points are at the two ends, the central position of the beam is presented to be the

best position for placement of the interface of interaction with the robot. Coincidentally it is also the center of gravity of the piece. Inside this location, some possible scenarios were studied expressed in Figure 40. Perforating internally the wood showed a good simple solution for enabling the robot to grab it and place it without colliding itself with another piece, and also for enabling making the element even longer once the subtracted volume could be added in the ends. A scenario with multiple internal holes could lead to a light and long piece capable of decreasing the number of used connections making cheaper the construction as a whole. As a result, just two simple holes (10x15cm) were placed in the wood piece keeping the 160cm length for attending an ideal housing floor height of 3.20m which is just two discrete elements placed vertically. Instead of a bigger piece, 160cm also attends a good relationship between human ergonomics, stair formation, and robotic movements in space.



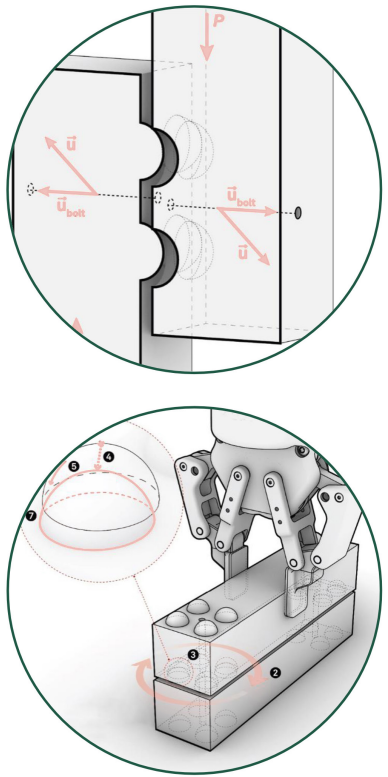
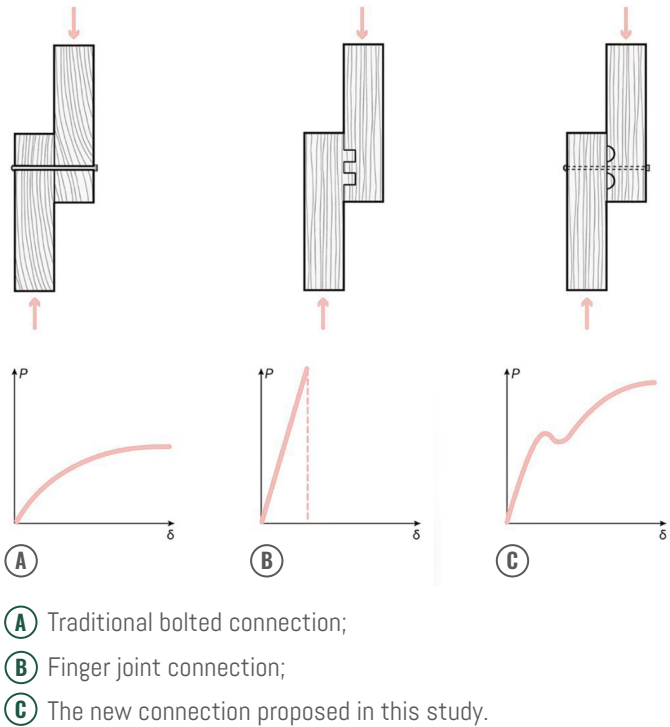
4.1.1 Connection design

As a design-for-disassembly guideline, the proposed connection design is made by reversible bolts. Glue and nails were disregarded due to hindering the disassembly process. The decision for bolts instead of screws is due to obtain longevity of the wood pieces interfering less with the material. Also, initial studies on the use of robots in assembling autonomous prefabricated systems identified challenges in the insertion of timber joints. Two main obstacles have been identified: growing friction forces with more connections and the need for precise insertion to prevent gap formation, reducing connection rigidity (Rogeaou et al., 2020).

Placing the connecting points at intervals of 60cm conforms to the regulations and norms for interior partitions and rooms (Koerner-Al-Rawi, 2020). So the bolting points' position is within the base 10x10cm grid and has 4 bolts per point to prevent that one bolting point is occupied. For each point of connection, just 2 predrilled roles will be used, remaining the 2 other holes for the coming piece. The designed connection has specific bolts that include the self-alignment concept in its geometry design. The research from Hansen et al. (2021) proved this self-alignment concept is validated and the way

this type of connection can bend without breaking is well-suited for discrete structures that depend on the strength of multiple connections (Fig. 41). Ductility refers to a material's strength being determined by all its connections rather than being limited by the weakest connection, unlike brittle connections (Hansen et al., 2021).

The main structural idea behind the connection proposed by Hansen et al. (2021) is that shear should be transferred by the interlocking of CNC-milled shear keys instead of relying on dowel and rope action provided by steel bolts. With precise fabrication, interlocking connections can be activated without significant displacement, unlike bolted connections. Modern CNC machinery enables high-precision milling with small tolerances and few geometric limitations, making it easier to fabricate complex shapes. (Hansen et al., 2021) The proposed connection method utilizes semi-spherical shear keys arranged in a square pattern to support both parallel and perpendicular intersections, along with conventional steel bolts. The arrangement and shape of the shear keys ensure accurate robotic assembly without requiring complicated procedures, as they act as an alignment feature.



Taking this design feature into the research, the proposed bolt design adds the self-alignment characteristic in its head. The head of the bolt that connects the two elements is responsible for the alignment of the next coming discrete element. In such a way the fixation of the pieces works as a network. The head has a symmetrical 45 degrees cone geometry that fits perfectly within the specific nut design (Fig. 43). The head works as convex geometry while the nut has a recipient concave geometry (Fig. 42). The nut has a square with rounded corners geometry to prevent itself from rotating while bolting. This squarish geometry just works because the nut is inserted in pre-drilled sockets with the same geometry. This type of connection offers several notable advantages, including the ability to self-align during construction, greater shear capacity than traditional connections, and ductile behavior. The morphological features of the bolts facilitate the process of robotic assembly and disassembly regarding precision, locking rotation, and enhancing shear resistance and moment capacity as proved the concept in the research of Kunic et al. (2021a). The connection is made by bolts of 20 and 30cm that cross all the sections of the discrete element until finding the other ending nuts.

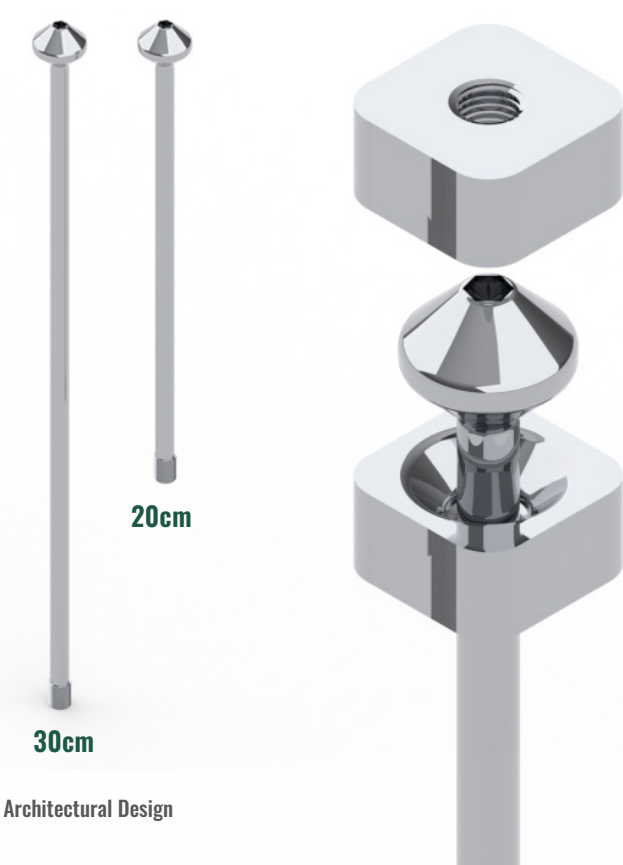


Figure 43. Connection detail of the bolt and nut (by author, 2023).

On top of it, a rectangular tubular metallic profile with perforated holes is added to the connection in order to lock the nuts inside the wood. The main reason for the addition of the metallic profile covering the whole connection point is for achieving durability aiming for a longer lifespan once the discrete element is designed to be disassembled and assembled by humans and robots many times. It also prevents the wood of open itself with humidity, keeping accuracy at the ends of the beams.

Over the metallic profile, each discrete element receives a unique identification through Fiducial codes (Fig. 45). They are often used in computer vision applications, such as augmented reality and robotics, to help machines identify and locate objects in the physical world. The most efficient way to keep track of the position of the various timber elements when using autonomous prefabricated systems is by utilizing fiducial markers, due to the numerous possible configurations available. Using fiducial markers that can be readily placed on top of timber elements for visual detection was identified as a low-cost and efficient method for updating the elements' position through a feedback loop (Rogeaou et al., 2020).

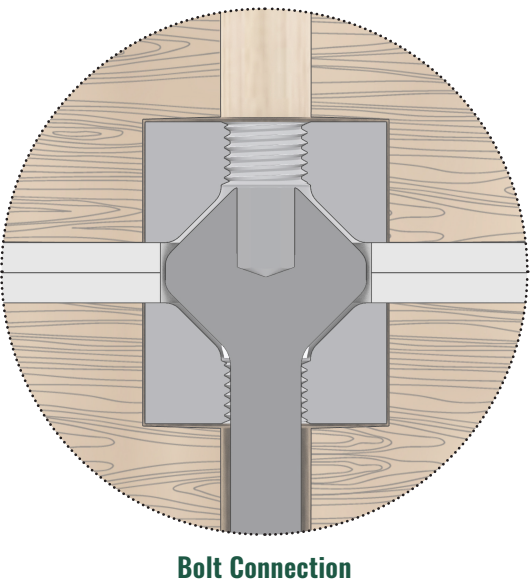


Figure 42. Design of the bolt and nut with self-alignment feature (by author, 2023).

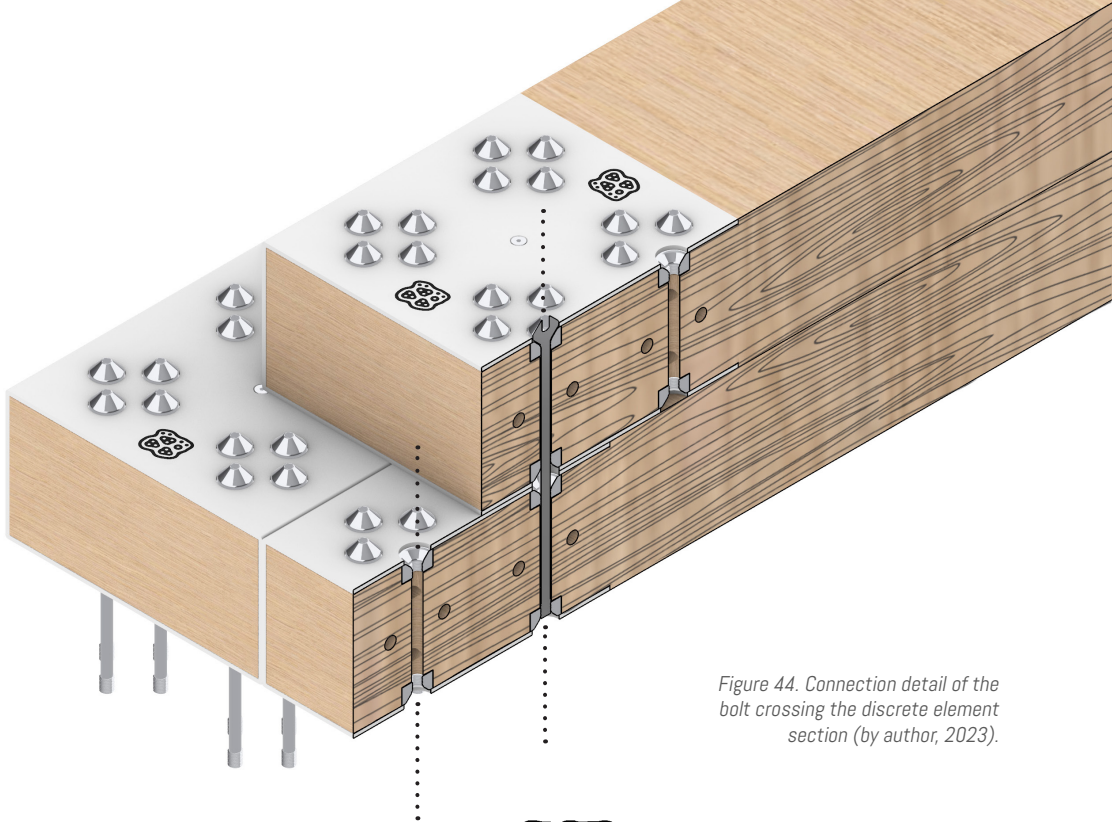


Figure 44. Connection detail of the bolt crossing the discrete element section (by author, 2023).

Thus, the codes in the connection aim for 3 main objectives. **[1]** The code would work as a circular material passport of the discrete element storing all the historical data related to its materials and movements around the globe together with its trajectory lifespan. **[2]** As each connection surface would receive two fiducial codes, the distance between them would help the robot to recognize by computer vision which surface of the discrete element it is needed to interact with. **[3]** Once the connection surface is identified, the codes would work also as calibration for the bolt placement due to the natural displacement of the elements in the assembly site.

Each connection of the beam offers 12 points of possibilities for connection inside of the 10x10cm grid. Each point of connection can host 4 bolts in the pre-drilled holes. Considering that each connection point uses a maximum of 2 bolts per joint, there will be always room for the next discrete element to connect with the other remaining 2 bolts' space. Therefore each connection is open for adaptation with 48 bolts placement possibilities. Their arrangements are symmetrical inside the grid to guarantee openness to all the possible configurations.

Figure 46. Section detail of the connection (by author, 2023).

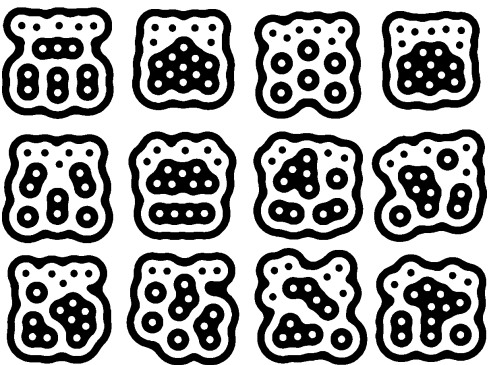
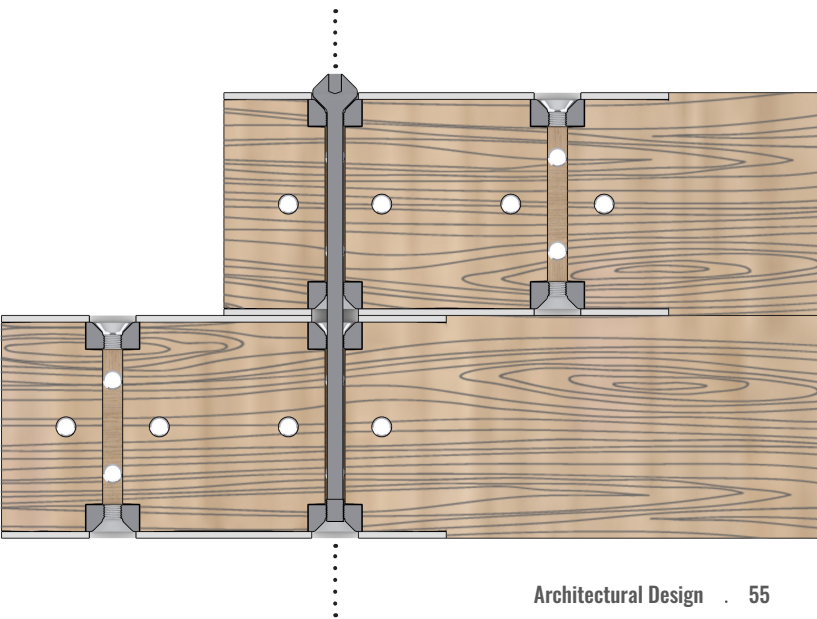


Figure 45. Fiducial codes (Bencina & Kaltenbrunner, 2005).



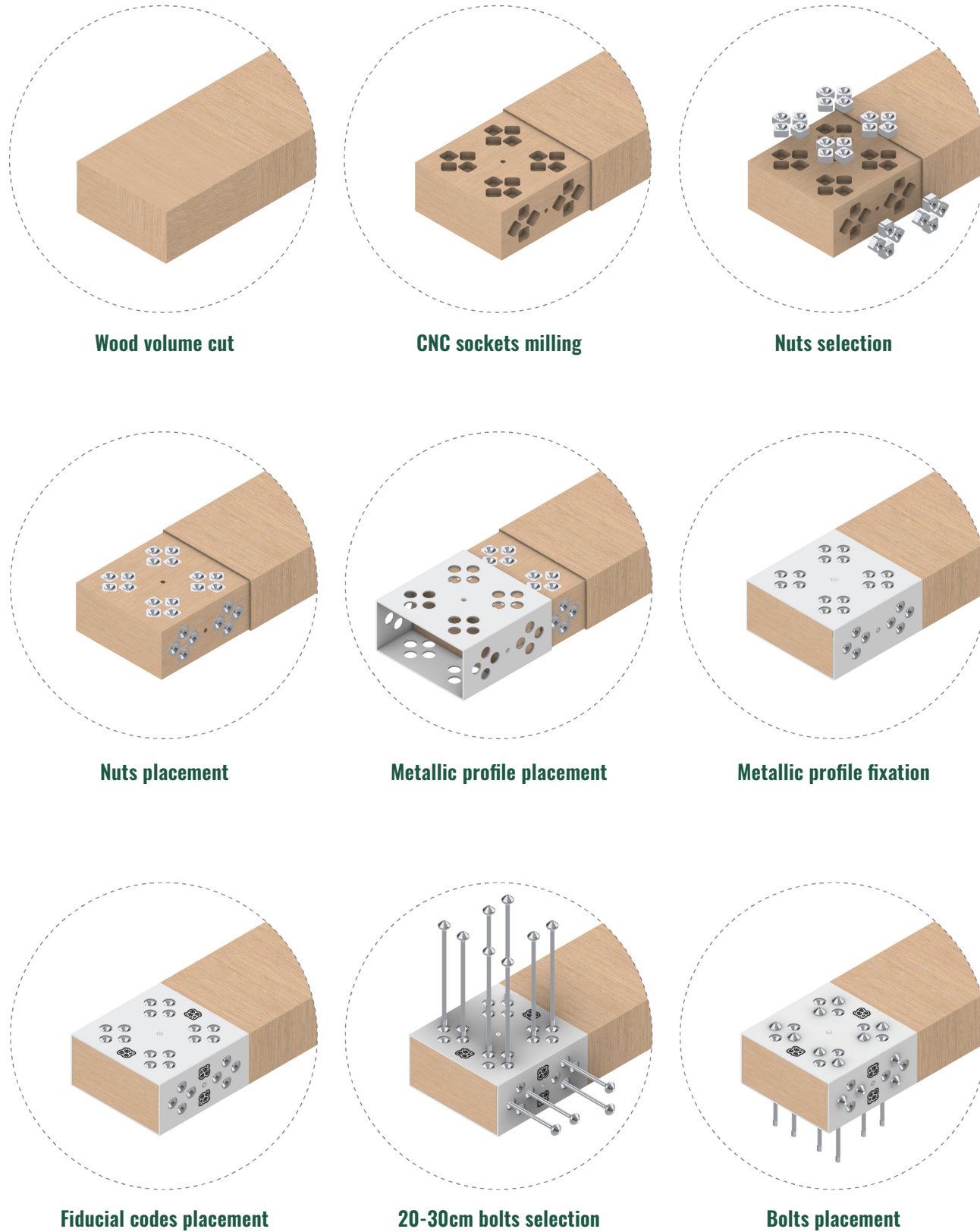


Figure 47. Step by step of the connection assembly (by author, 2023).



Figure 48. Discrete element design (by author, 2023).

4.2 Discrete elements aggregations

The discrete element presents hole patterns that allow different combinatorial aggregations and give higher design freedom compared to typical modular elements, enabling easy reconfiguration. The discrete element by itself is considered a generic entity without any specific function that can be used in many scenarios. The geometrical function is just assigned to the pieces by their aggregation aiming at a specific functionality. Therefore it is needed to understand the relationship between the possible connectivity and the overall functional design to set the approach for its generation. The combinatorial rules of all the connection possibilities are responsible for the generative aggregation design.

The discrete element leaves its generic characteristic when the generative design tool assigned a specific place for it in the whole design. With a final placement position, it is possible to predefine where the bolts will be installed. Thus the discrete element becomes a customized piece within the whole generated customization. The manner of grouping them by having preplaced bolts for the final position is what gives its uniqueness and meaning within the whole. However, it conserves its generic nature after the disassembly, being able for new use purposes in another different configuration and functionality. This framework allows the automation of the process once robots can easily move objects for customized placement with spatial precision. The automation of the process will be discussed in the next chapter, but before it is needed to set the aggregation rules for its generation.

The research used the Wasp plugin in Grasshopper to simulate the aggregations. The combinatorial rules were created by naming each of the 24 connection points and giving vertical and horizontal directions to these points (Fig. 50). Thus each discrete element offers 48 possible ways of connection. This sums up a total of 2,304 configuration possibilities rules when related to another discrete element. However many configuration rules looked the same because of the piece symmetry and not all the possibilities offered stable connections for a building structure. So in order to efficiently simplify the rules, 84 combinatorial rules were extracted from the total possibility to conduct the research (Fig. 51). These 84 rules include horizontal growth by stacking elements and perpendicular connections to change the growth axis, while excluding all the symmetrical repetitiveness and the connections that would require 40cm bolts.

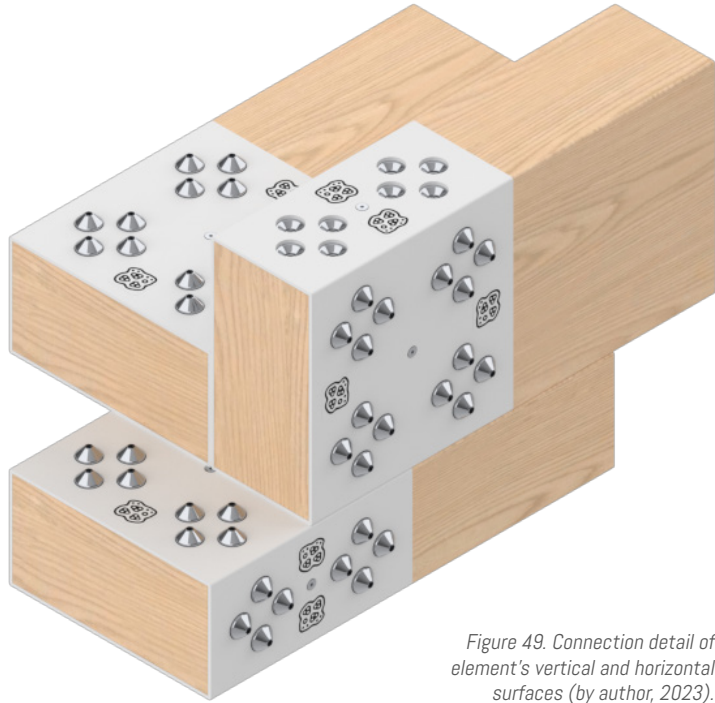
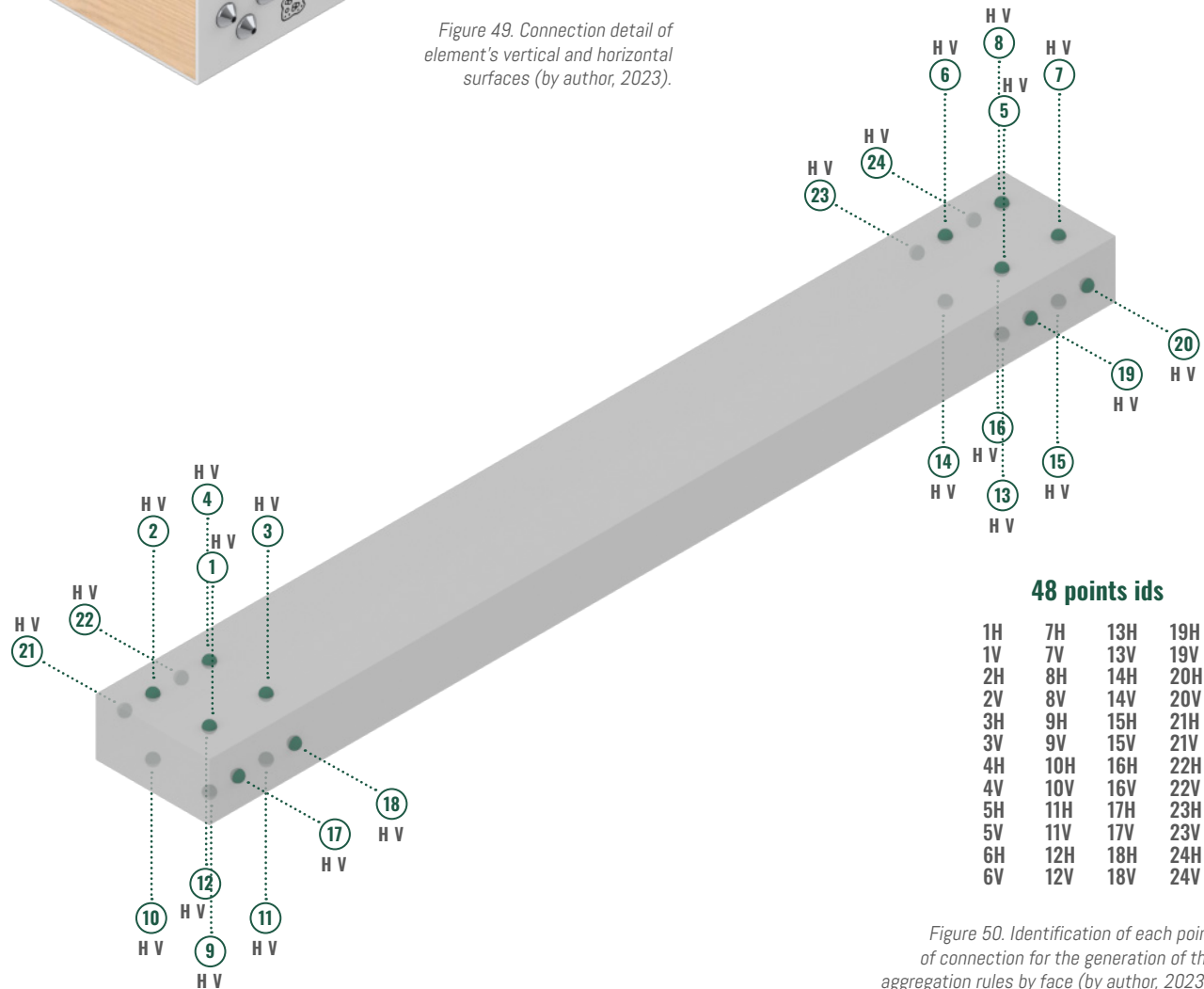


Figure 49. Connection detail of element's vertical and horizontal surfaces (by author, 2023).



48 points ids

1H	7H	13H	19H
1V	7V	13V	19V
2H	8H	14H	20H
2V	8V	14V	20V
3H	9H	15H	21H
3V	9V	15V	21V
4H	10H	16H	22H
4V	10V	16V	22V
5H	11H	17H	23H
5V	11V	17V	23V
6H	12H	18H	24H
6V	12V	18V	24V

Figure 50. Identification of each point of connection for the generation of the aggregation rules by face (by author, 2023).

The 84 aggregation rules are the base for the combinatorial generative design process used in this research. Their arrangement is capable of populating the space by connecting the discrete element following the grid logic (Fig. 52). This research focused on understanding the connectivity aspect between discrete elements and how it influences the whole workflow of robotic construction, however, a further development of this method of construction is certainly to test their behavior within a structural analysis calculation in which aggregation follows a certain structural logic. As illustrated in Figure 53-58, the aggregation formation can be shaped to perform efficiently certain structural-functional behavior.

Aggregation rules 84 possibilities

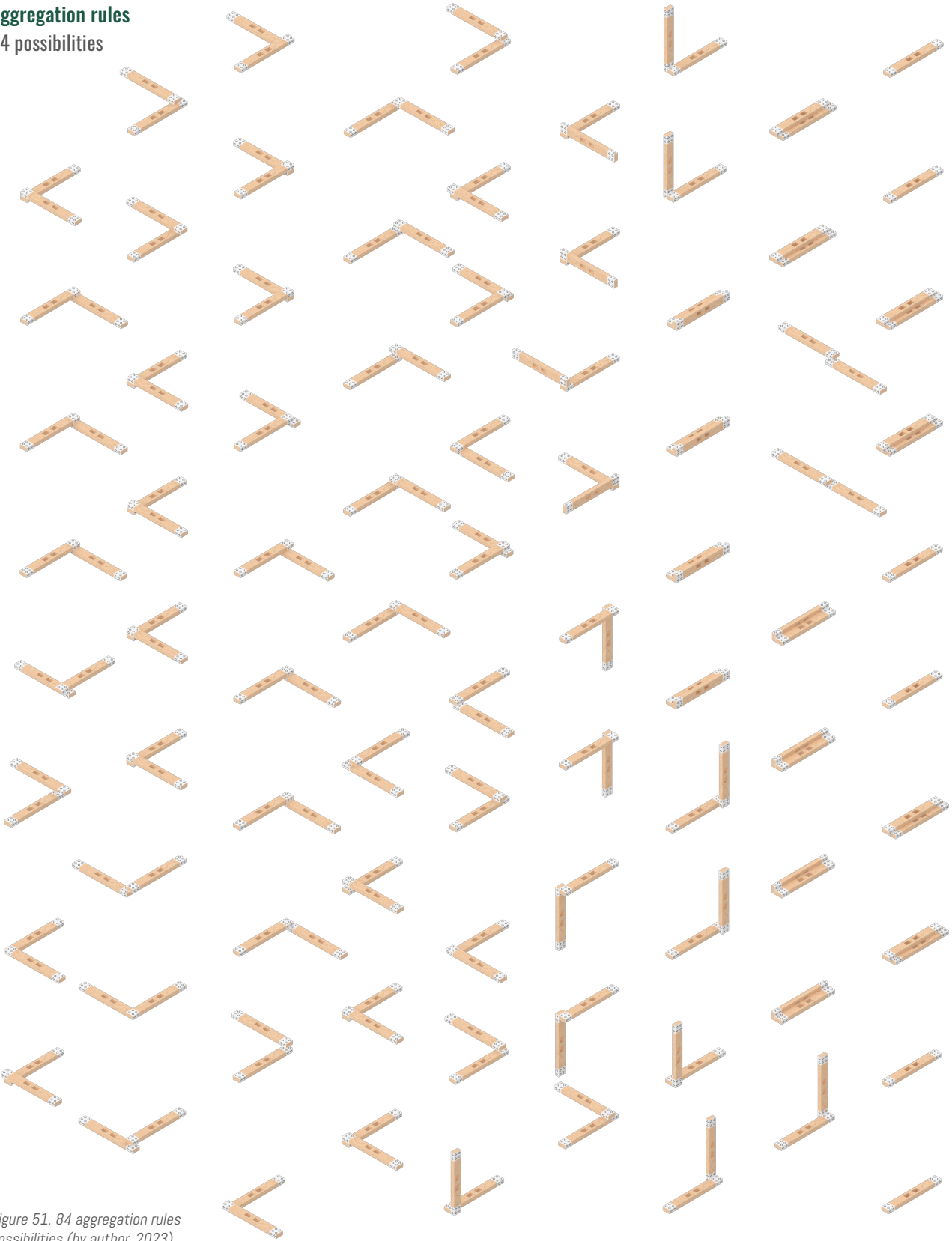


Figure 51. 84 aggregation rules possibilities (by author, 2023).

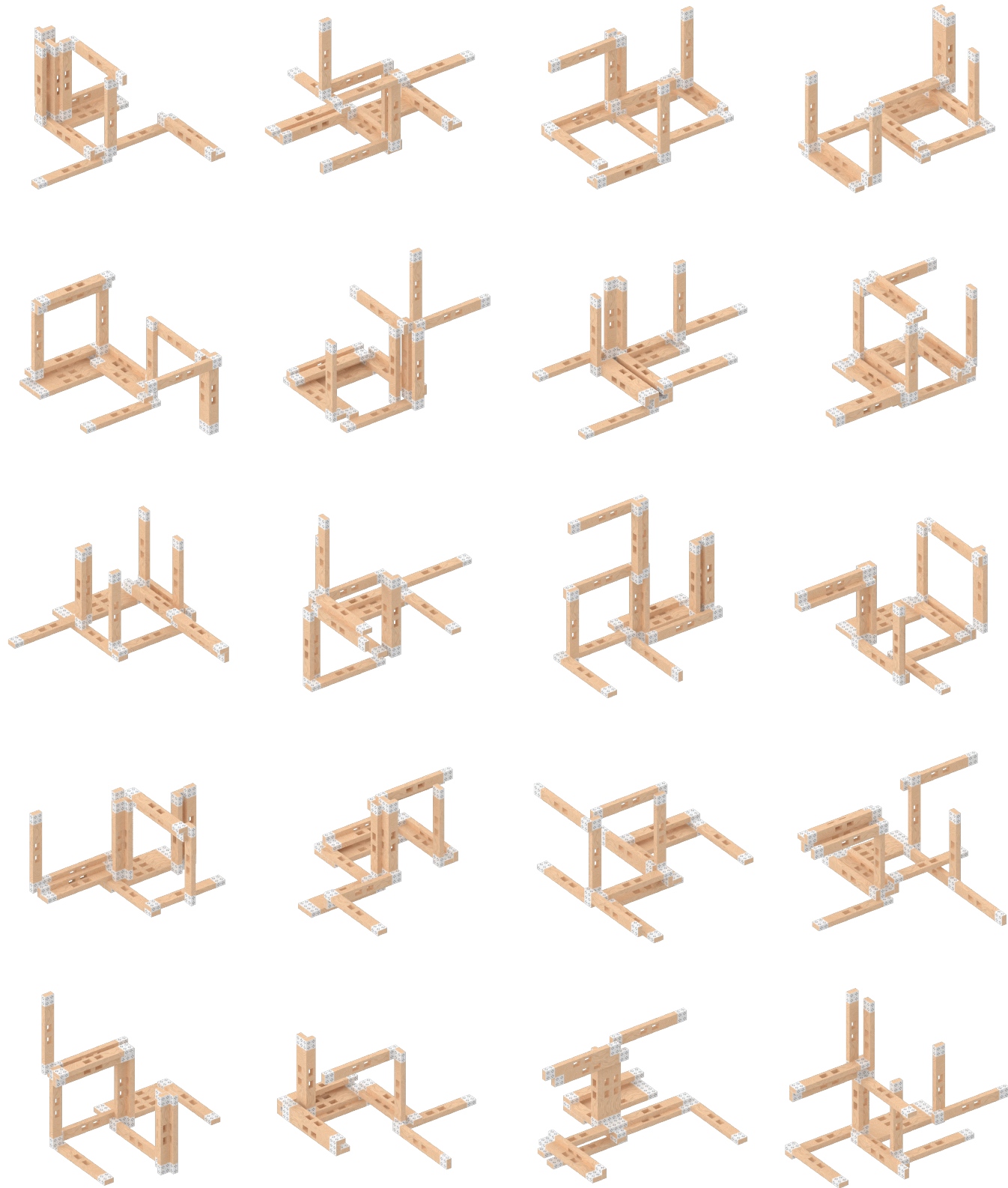


Figure 52. Aggregation variability opportunity from generative rules (by author, 2023).

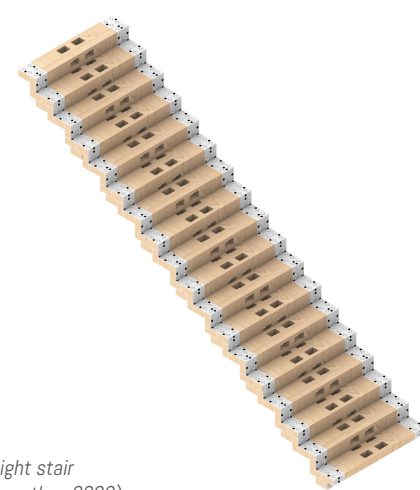


Figure 53. Straight stair aggregation (by author, 2023).

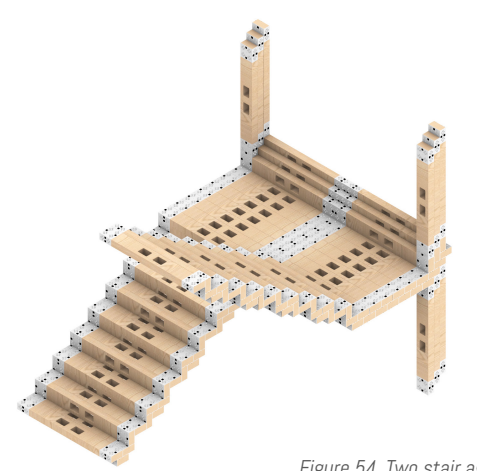


Figure 54. Two stair aggregation (by author, 2023).

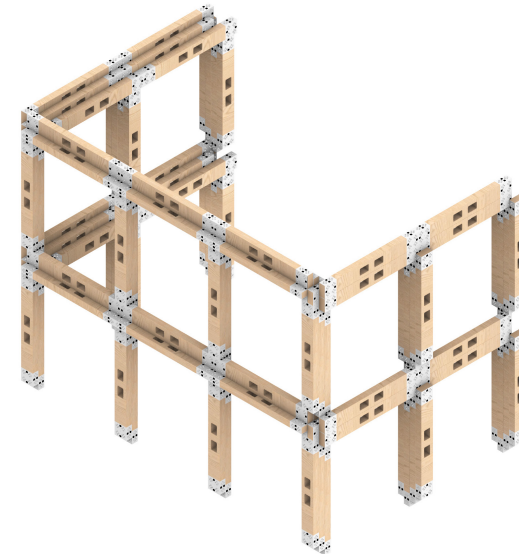


Figure 55. Traditional columns and beams aggregation (by author, 2023).

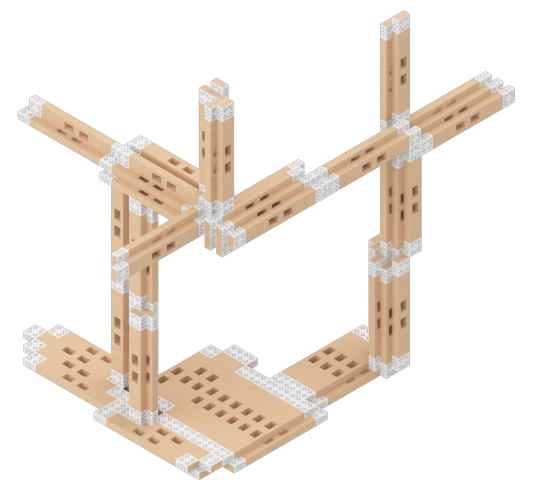


Figure 56. Customized column and beam aggregation (by author, 2023).

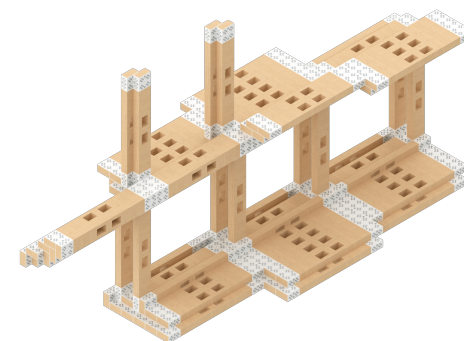


Figure 57. Transition aggregation of horizontal and vertical planes (by author, 2023).

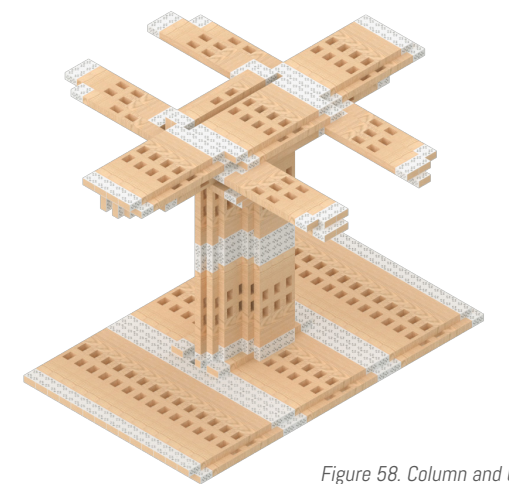


Figure 58. Column and beams aggregation (by author, 2023).

4.3 Generative design approach

Generative design is a design process that uses algorithms and computational techniques to generate multiple design options based on a set of input parameters or constraints, leading to more innovative or optimized designs (Autodesk, n.d.). Thus computational design is the key tool for the generative architectural design approach. Therefore, taking the combinatorial rules expressed in the previous section, the research explored the potential of 4 generative methods to develop the base architectural structure as illustrated in Figure 59. Each method has its pro and cons relating to the design scale and its functionality.

Method 1: This method consists in populating the space in one base boundary geometry with the discrete element based directly on its combinatorial rules created from its connection possibilities. Inside this approach, the aggregation rules require a lot of computer processing power to aggregate a huge quantity of parts at once. Also, the boundary boxes need to be generated beforehand in a manner to adopt the structural logic. The control limited the connectivity available for growth, so the resulting aesthetic can look abstract, which is ideal for sculptural purposes but not for structural efficiency.

Method 2: This approach was based on the orthogonal voxelization of the construction site and afterward the replacement of the voxels by a modular aggregation. The voxel dimensions were the boundary box for the aggregation which contributed to reducing the required computer processing power. The aggregation inserts already the structural function of the floor, column, and beam in its formation process. It was successful in its implementation, however, this approach generated a very repetitive and rational structure following Modernist aesthetics that do not express the whole potential of discrete architectural thinking.

Method 3: The third method combines the first two approaches. The concept here is about creating bigger aggregations from the initial element and afterward aggregating these bigger components between them inside of a base boundary geometry to shape the whole design. It is about two levels of aggregation. The first aggregates elements while shaping pre-oriented structural components for vertical and horizontal placement inside the desired housing dimensions. The second aggregates these larger components between them populating and creating the spaces. Their dimensions and aggregation rules determine the housing formation. Inside of this logic, the floor plan is structurally generated and populated afterward in a custom manner being open to future adaptations. It offers room for a huge range of variability and contemporary aesthetics of mass customization. This is the chosen method to perform the next phases of the research on-site.

Method 4: This approach was based on the previous method but with three levels of aggregation. The idea was to generate apartment units from bigger aggregated elements and aggregate them aiming to shape a cloud of apartment units in a third-level aggregation. It offers huge potential for spatial variability and aesthetic quality, however, the structural logic is placed in the background needing to follow the resulting overall geometry. As the balance between structural efficiency, mass customization, and spatial quality is a goal of the research, this method was not explored any further.

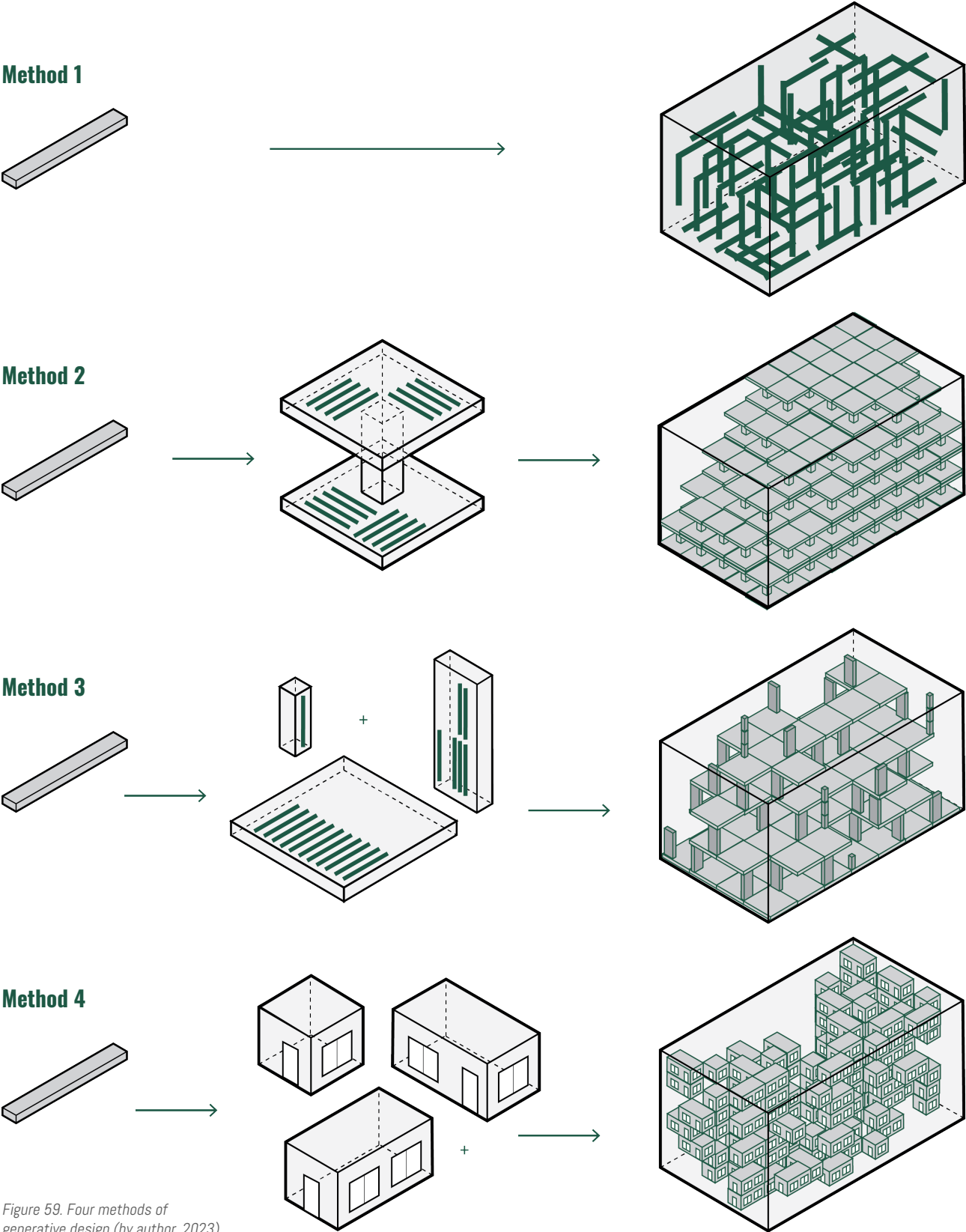


Figure 59. Four methods of generative design (by author, 2023).

4.4 The site

Netherlands
Rotterdam
Feijenoord District
Kop van Zuid-Entrepot
Laan op Zuid Avenue

Rotterdam, one of the largest and most important cities in the Netherlands with a population estimated at 655468, is known for its innovative and modern architecture (AllCharts, n.d.). The city has a rich history of experimental and innovative architectural projects, from the iconic Cube Houses to the striking Markthal. Therefore, due to its dynamic and forward-thinking nature, strong tradition of innovation, diverse population, world-class architectural firms, and commitment to sustainability, Rotterdam is ideal to be used as a context for the implementation of the concepts discussed in this research.

The area of the site is the neighborhood **Kop van Zuid-Entrepot** right on the border with the neighborhoods Kop van Zuid and Afrikaanderbuurt in the **Feijenoord district**. The Kop van Zuid-Entrepot neighborhood is an urban renewal project that transformed an abandoned industrial area (Fig. 61) into a vibrant mixed-use district. The neighborhood is located on the southern bank of the Nieuwe Maas River, opposite Rotterdam's historic city center (Fig. 60). Back in the day, it

was an important commercial and industrial area, with many warehouses and shipping facilities located along the waterfront. The redevelopment of the area began in the 1990s, and it has since become a symbol of Rotterdam's transformation and is still in constant renovation. In the near future, the area will receive the Rijnhaven urban plan and Maashaven urban plan which will densify even more the neighborhood bringing more development and dynamicity. The area is in constant construction and needs adaptable and circular construction systems to face this fast-paced growth. Therefore, the area also needs to have housing, and it must be affordable housing for future generations.

The Kop van Zuid-Entrepot neighborhood is home to a diverse population. The neighborhood attracts a mix of residents, including young professionals, families, students, and individuals from different cultural backgrounds. The Feijenoord district is also known for its multicultural population, with a mix of Dutch, Surinamese, Moroccan, and Turkish residents (Council of Europe, n.d.).



Figure 61. Industrial port area history of the neighborhood (Beeldarchief Rijkswaterstaat, between 1900 and 1926).

Figure 62. Kop van Zuid-Entrepot neighborhood (Google Earth, 2022).



Figure 60. Rotterdam southern bank of the Nieuwe Maas River (Google Earth, 2022).

The chosen specific site is located at **Laan op Zuid Avenue** with simple rectangular dimensions of 50x80m. The main reasons for the site selection were the simple shape, the extremely flat topography that is ideal for the implementation of automated construction processes, and the easy logistical access with three open fronts to streets. The site also has a good position to catch sunlight and easy access to transportation such as metro, tram, buses, and cycle lanes, showing to be ready to host more inhabitants. The site aesthetics and impressions are characterized by a sense of innovation, creativity, and modernity, with many buildings and public spaces incorporating unique and eye-catching designs.

As expressed in the graphics (Fig. 64), the most common family composition in the neighborhood is a single household. The age composition is majorly between 27-54 years old

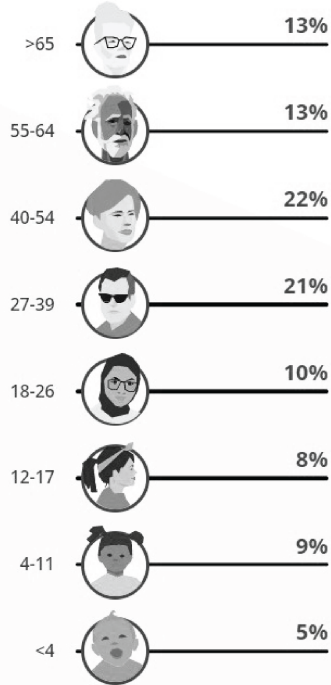
which is inside the productive adult life. The most common house typology is apartments with elevators, and 47% of the occupation is based on social rent instead of owning the apartment with just 28% (Wonen in Rotterdam, n.d.).

It is important to mention that the design project is not context-based. The site context is used as a guideline and background to set some boundary conditions for the representation of a possible implementation of the developed construction system. The societal issue of affordable housing addressed in this research is present in the whole contemporary world, in the whole Netherlands. The user's requirements adopted by the research are not specifically from inhabitants of the area, but instead from all generations from the Millennials onwards who is lacking access to housing.

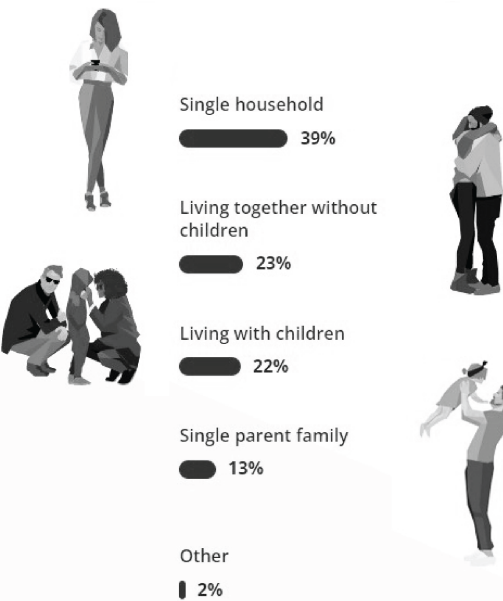
Figure 63. Topography and accessibility of the site (Google Earth, 2022).



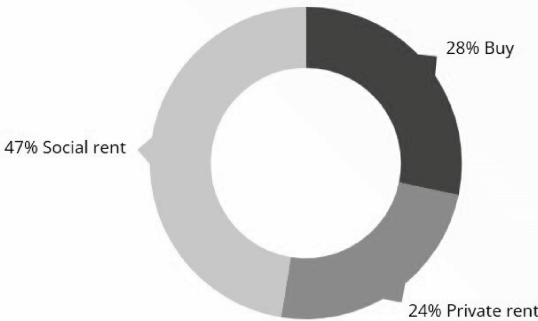
Age composition



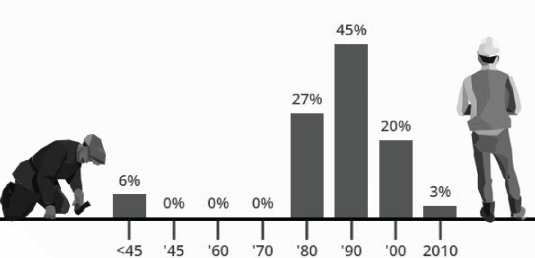
Family composition



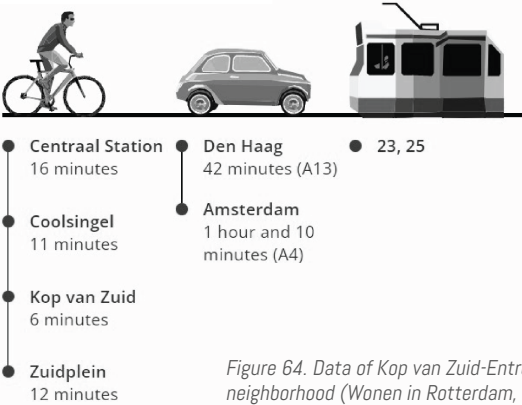
Buy versus rent



Construction period



Accessibility



Housing type

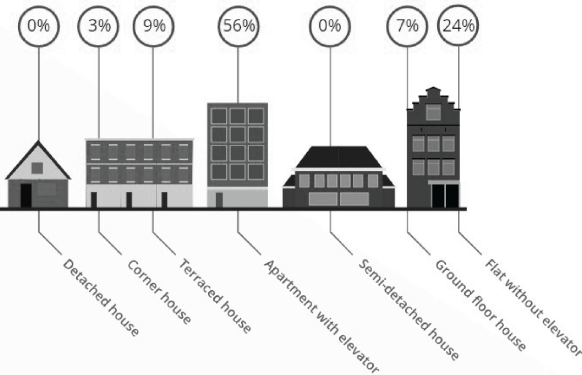


Figure 64. Data of Kop van Zuid-Entrepot neighborhood (Wonen in Rotterdam, n.d.).

4.5 Housing platform’s program

The user’s requirements adopted by the research take into consideration the generation’s demands from the Millennials onwards who are world-widely lacking access to affordable housing. Therefore the housing program is planned for the early stages of adult life for scenarios where there is not much-accumulated capital to invest in the housing private market. The real-estate market nowadays is based on the land’s price and its potential-produced square meters. However, as urban land is not a commodity to be produced and housing is a human right, different housing typologies and business models are needed to fulfill the increasing demand.

The housing program proposed in this research looks to foster the changes in the ownership of the physical housing unit to a needed number of square meters required per person. Inside of this logic, the person owns a certain amount of square meters that is not attached to the land but is part of a digital platform that regulates the distribution of square meters per inhabitant. As potential, the proposed housing platform considers adaptation according to life changes, in which people can use their square meters for another location depending on availability. A bigger housing unit can be

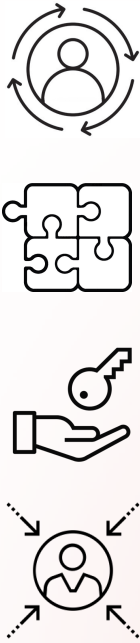
shaped by the union of people. The revised measurement system would be determined by the number of users or the structure of families. Instead of owning a standardized apartment, residents possess personalized and customized spaces promoting fair resource allocation. Thus, the building geometry of distributed and occupied by the square meters per person, and not by how much money they have.

Additionally, advancements in construction methods allow housing blocks to be adaptable to various spatial layouts, ensuring that their form remains dynamic rather than fixed (Cheng et al., 2020). This dynamic approach to living is made possible by the proposed timber construction system that allows these spatial changes relating to people’s needs due to the design for disassembly guidelines and human-scale discretized elements. The people living in the building can shape their floor plan according to their current life, and not the opposite. The construction system allows the easy expansion of the building, its retraction, and its internal changes over time. Structural logic is also part of this concept by being capable of changing its configuration over time from the addition, subtraction, reuse, or reallocation

of its discrete elements aggregations. The housing platform is connected to the digital twin model of the construction to enable its recalculation.

The proposed housing platform’s program works as a service provider in the design’s narrative. The use-oriented housing service can be read as the users owning part of the cooperative company by square meters rights from a subscription but not physically a specific space. The system separates ownership from a specific physical location, and its programming is continually evolving throughout its existence. The use is exclusive, but without owning a closed product. The user buys square meters and sells square meters. Initially, the housing function is predefined as a purpose but it can be reprogrammed as needed as time goes.

Based on the work of Cheng et al. (2020) that proposes a housing platform while addressing tools, strategies, and prototypes for post-boom life, the housing program of this research considers the bellow design aspects as main guidelines (Cheng et al., 2020, p. 137).



- [1] Recognize Humans as a variable element.** It is crucial to regard humans as a dynamic factor that undergoes constant changes.
- [2] Adaptability through Reconfiguration.** To prevent the building from becoming a passive design, it is necessary to prioritize the reconfiguration of space following the departure of a tenant.
- [3] Changing ownership notion.** By eliminating the notion of ownership, the physical space is defined solely by variable components rather than a constant owner.
- [4] User-centric approach.** Taking into consideration specific functional requirements of young generations and what rooms they desire or are willing to share with others.

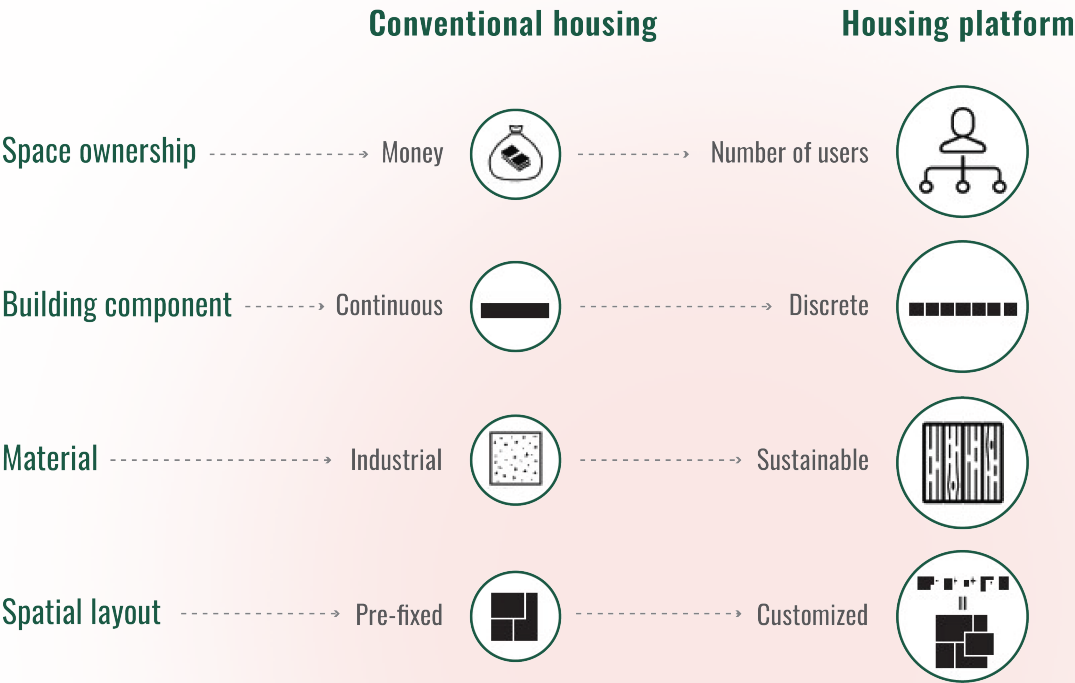
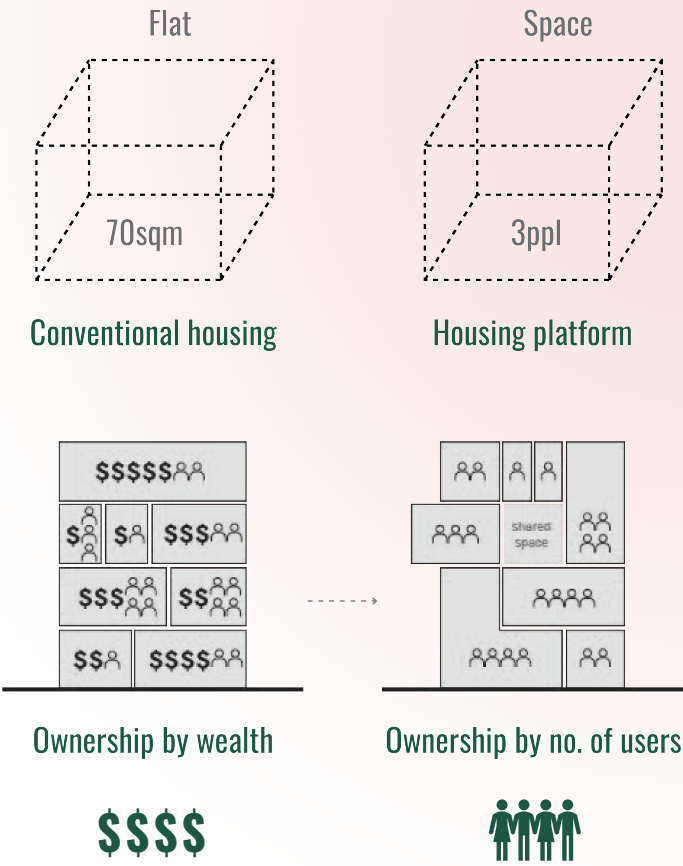


Figure 65. Relationship between conventional housing and housing platform. Adapted from Cheng et al. (2020).

Figure 66. Relationship between ownership by wealth and ownership by number of users. Adapted from Cheng et al. (2020).



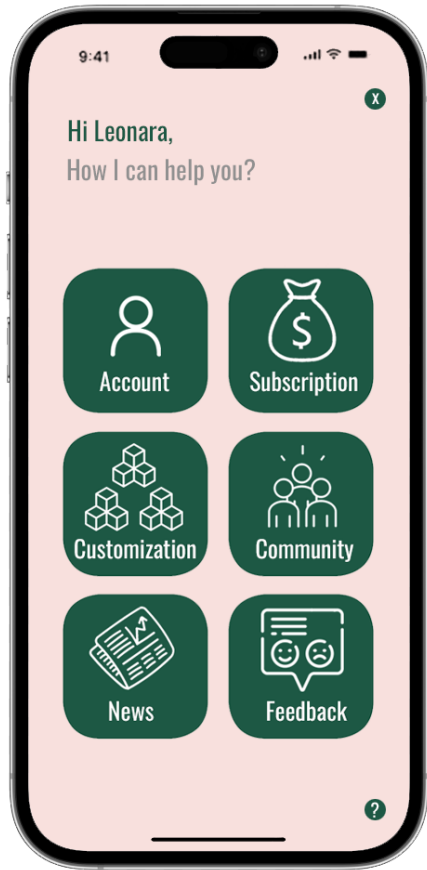


Figure 67. User's platform interface app.
Adapted from Cheng et al. (2020)

4.5.1 User's interface app

The proposed user's interface app is the input for the platform for the generative design methods and afterward for the assembly digital twin model as illustrated in Figure 68. It is the integration of what is digitally designed and what is physically assembled and reconfigured. Each change in the building would update every part of the system's platform. Based on the work of Cheng et al. (2020), the app would have four main functionalities:

[1] Enrollment. During this phase, essential information such as family structure and desired location is collected. Users are then grouped based on their family size, which serves as the primary determinant for allocating spaces.

[2] Membership. During this period, users gain ownership of their allocated spaces and access to a certain living standard. The membership operates on a subscription-based pricing model for maintenance, offering users plans that can be purchased for specific durations at predetermined prices.

[3] Space customization. Upon acquiring ownership of a space, users have the freedom to customize and arrange its components according to their individual living preferences and behaviors. Users can also apply for future adaptations and modifications.

[4] Community. It is about a space to encourage social participation and resource-sharing. It fosters the establishment of an open and thriving network within the community.

Automated workflow overview

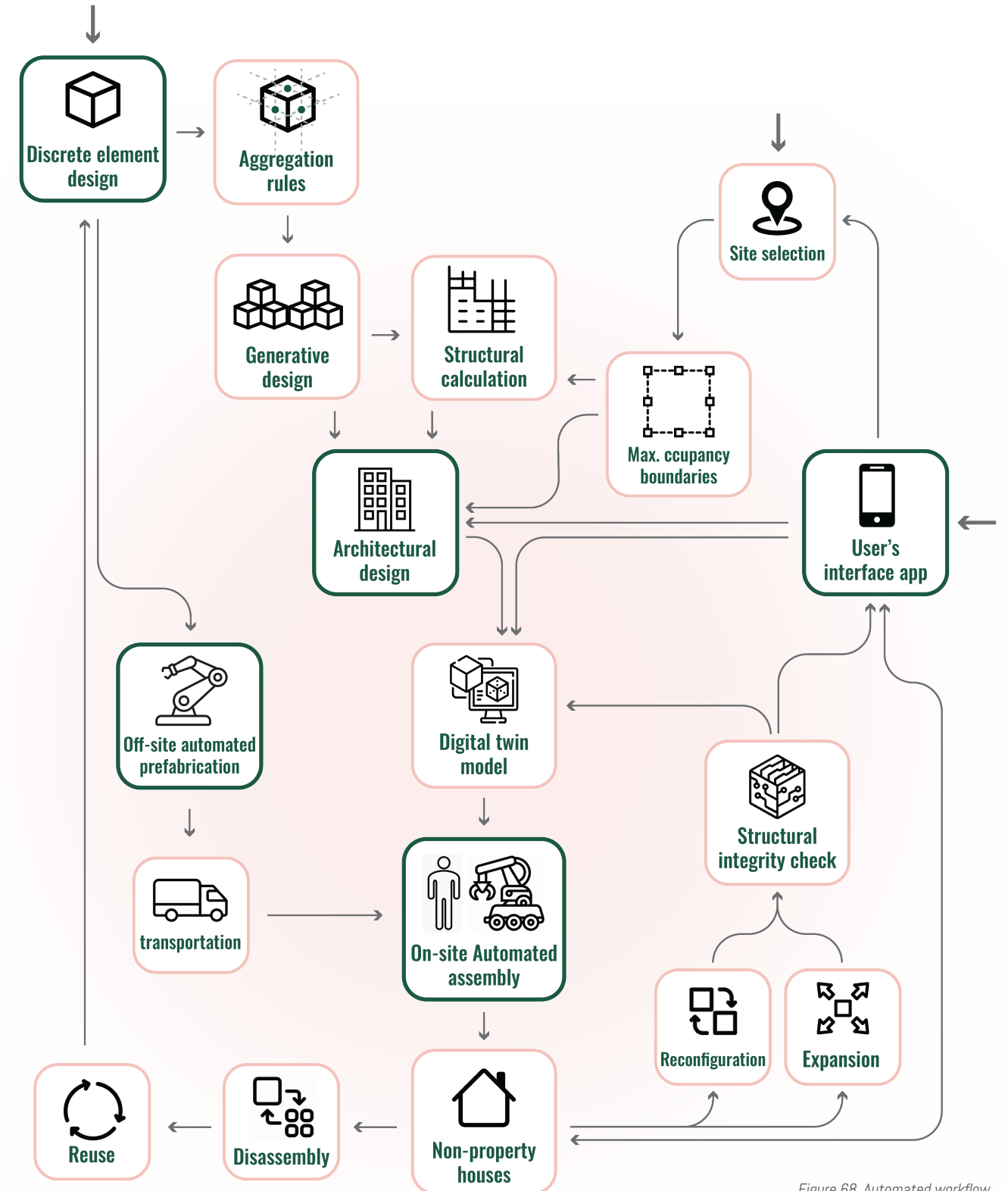


Figure 68. Automated workflow overview (by author, 2023).

4.6 Discrete timber housing design

By promoting systemic thinking, universal and flexible frameworks, economies of scale, platforms, open-source, decentralization, the prototypical, mobility, prosumerism, the digital, scalability, and continuity in design production, we can propose an 'all digital' or 'wholly digital' Discrete approach to the automation of housing production. (Claypool, 2019)

Creating large structures using discretized building blocks has immense possibilities in architecture (Koerner-Al-Rawi, 2020). The incorporation of the addressed concepts so far is the base for the architectural design implementation on the site. The integration in one central system of the user interface app, the generative discrete design, and the assembly digital twin model is the confluent point for the production chain with a common syntax between the robots and the building elements. The remaining components of the multi-layered building, such as insulation, waterproofing, and cladding layers, can be digitally manufactured using the same digital information model and pre-assembled in the factory setting (Schwinn, 2016). Therefore the architectural design workflow used to develop the building is in Figure 68.

The architectural result is seen as part of the research on the potential of the construction system regarding spatial qualities and aesthetics. It is one face of the potential of what the construction system can contribute to the building industry. As explained in section 4.3, the third combinatorial design method was used to generate the architectural structure base. This method combines the initial discrete elements to create larger components, which are then arranged within a defined boundary to shape the overall design.

The site occupation strategy is the boundary box geometry used to aggregate the larger components. It was created using CAD tools to express a result, but in further development, it could be generated by a generative definition. Its creation followed the relationship with the buildings in the

context and the southern sunlight position as a guideline as illustrated in Figure 69. It took also into consideration the space for greenery on the ground which before is also the space for the on-site automated construction logistics.

Afterward, the two-level aggregation approach involved shaping first the structural components for placement to fit the housing dimensions and then aggregating them to shape the spaces (Fig. 70). These components are 3 basic geometries: **[1]** horizontal aggregation of 480x480x30cm, **[2]** vertical aggregation of 160x50x320cm, and **[3]** multi-axis aggregation of 50x50x160cm. The aggregation rules of these 3 components were written as a structural strategy in order to have always a horizontal component supported by at least one vertical component. The number of components and the proportion between them are variables in the design process that can be adjusted accordingly to satisfy for instance the need for more vertical components. The third component is treated as an extension for future expansion, so their aggregation rules place them just after when the base structure is set. On top of that, any additional individual discrete element can be added to the whole building aggregation to fill gaps or to create customized geometries. After all, the result is a broad range of variability and contemporary aesthetics through mass customization based on minimal structural considerations that generate floor plans open to be customized for future adaptations. In this paradigm, flexibility, variability, and spatial diversity are on top of being structurally efficient. What is behind the scene is that even after being built the building is still a material storage that can be reused.

Site occupation strategy

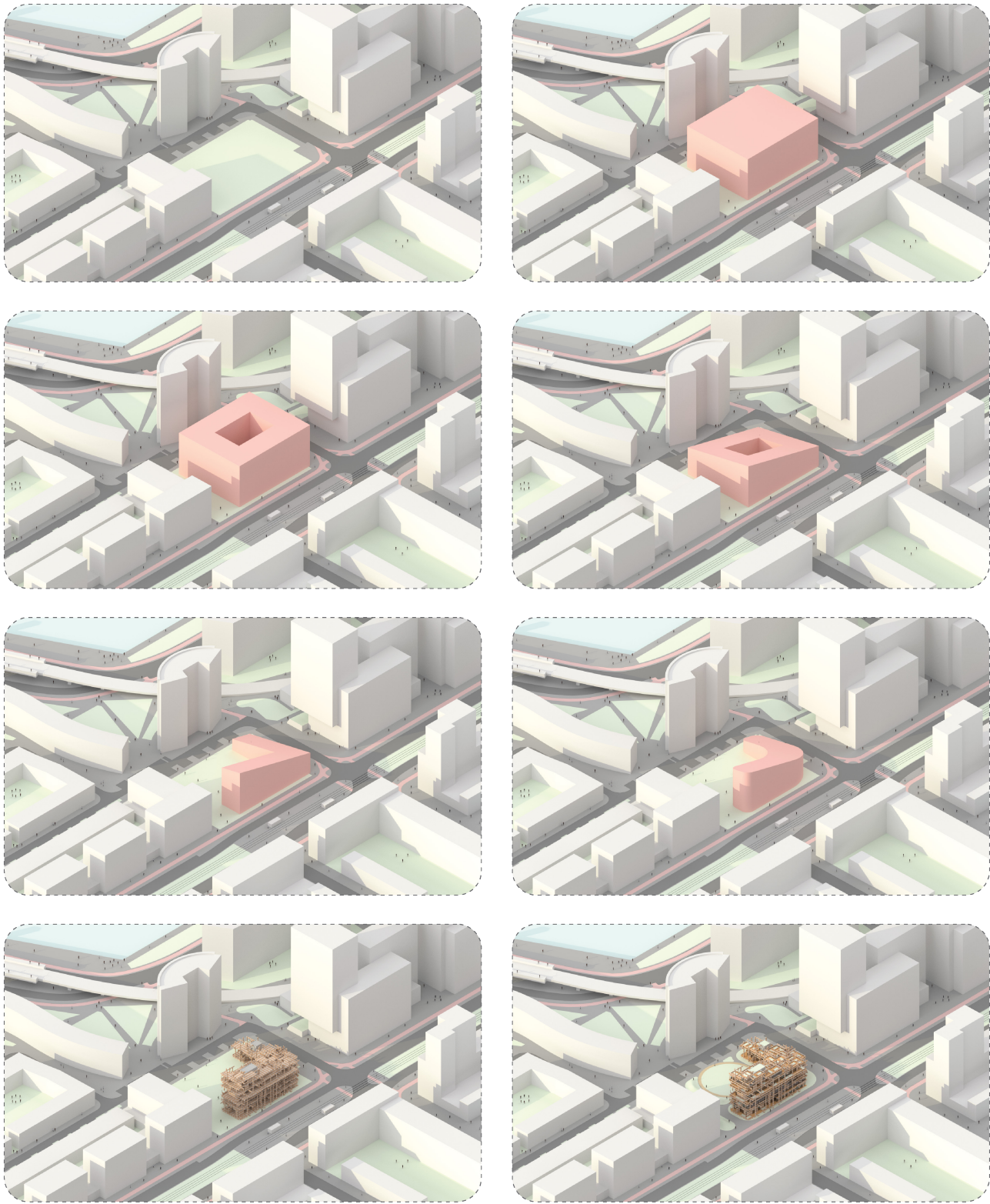


Figure 69. Boundary box geometry. The site occupation strategy (by author, 2023).

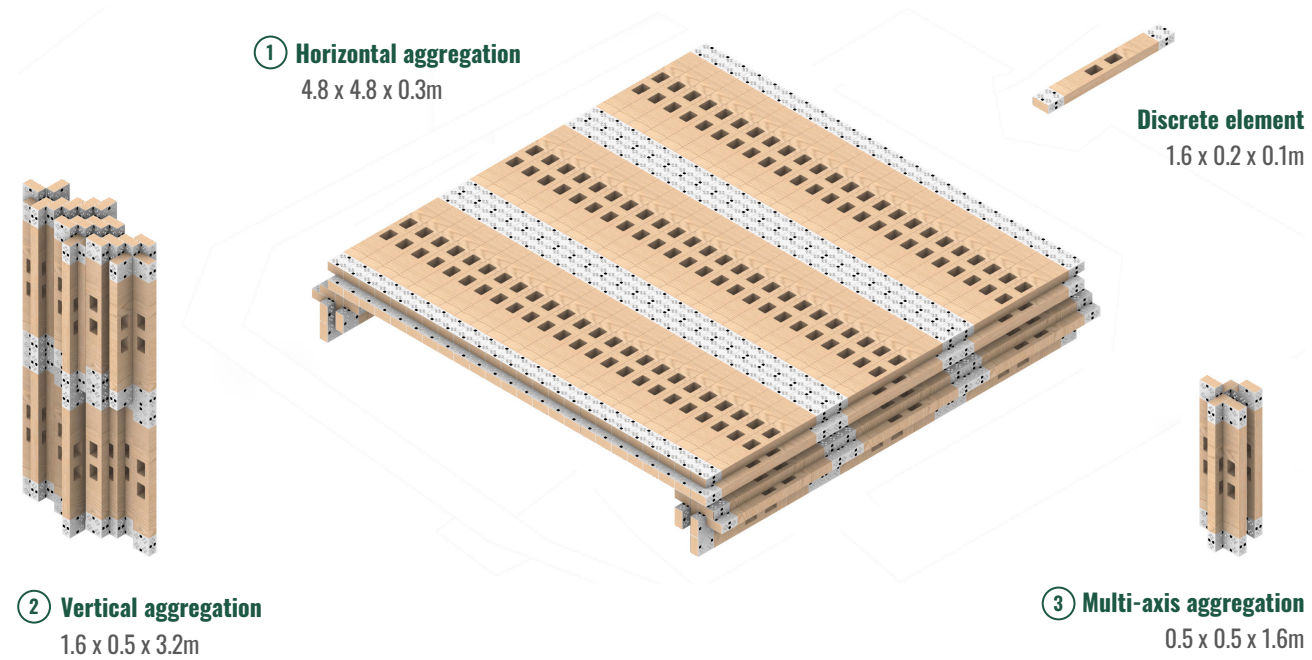


Figure 70. Aggregated components to shape the building in two aggregation levels (by author, 2023).

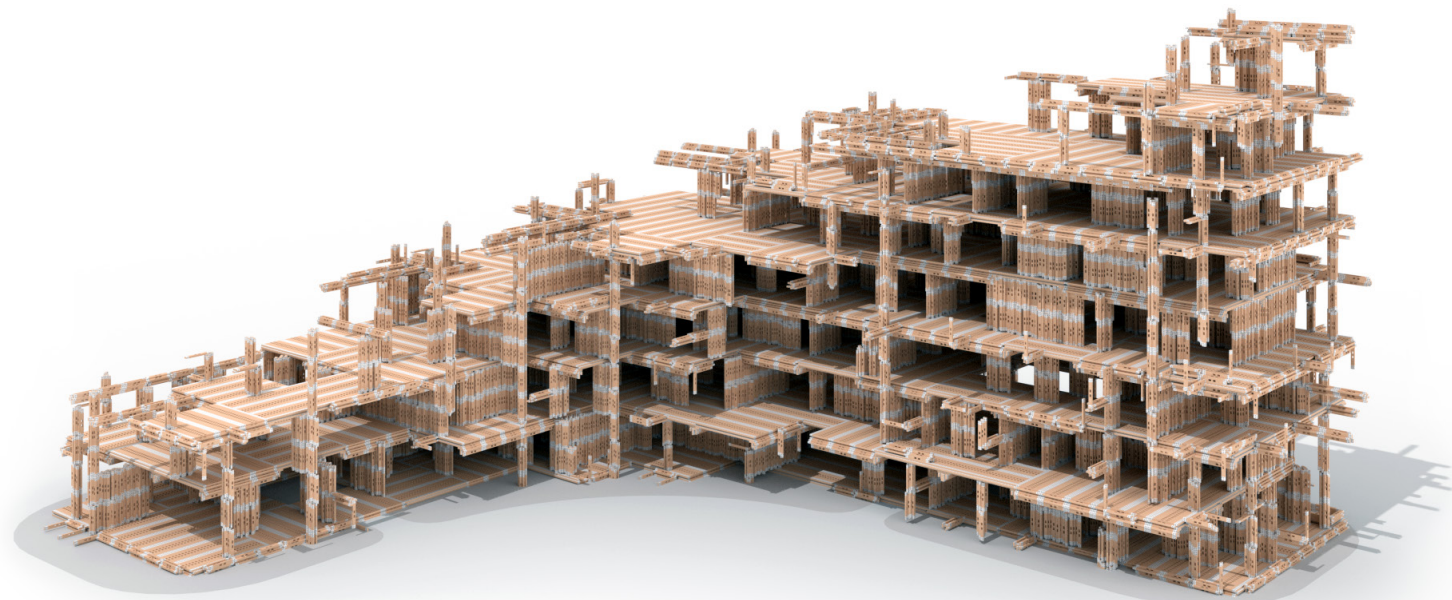


Figure 71. One of the endless possibilities of the generated structure (by author, 2023).



Figure 72. Ground floor (by author, 2023).



Figure 73. Architectural Design overview (by author 2023).



Figure 74. Housing design overview (by author 2023).



Figure 75. Housing assembled structure (by author 2023).



Figure 76. Columns formations on the facade (by author 2023).



Figure 77. External facade structure aggregation (by author, 2023).



Figure 78. Balcony and structure extension (by author 2023).



Figure 80. Housing balcony (by author, 2023).



Figure 79. Facade viewing the Laan op Zuid Avenue (by author, 2023).



Figure 81. Community space and possible future housing expansion (by author 2023).



Figure 82. Facade viewing the community space backyard (by author, 2023).



Figure 83. Community terraces on each floor with urban farming in cachepots (by author 2023).

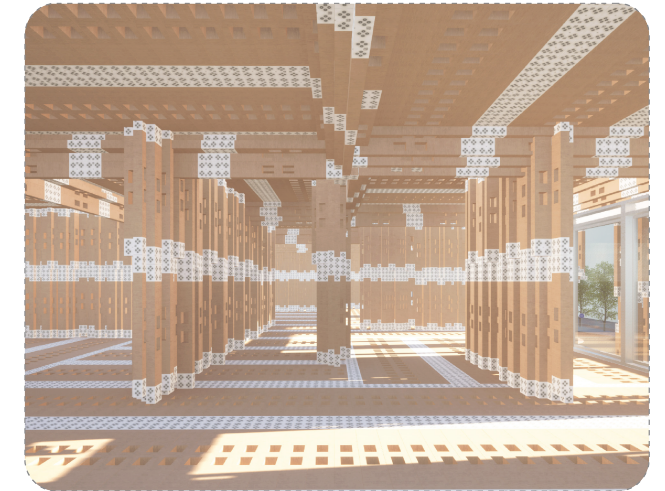
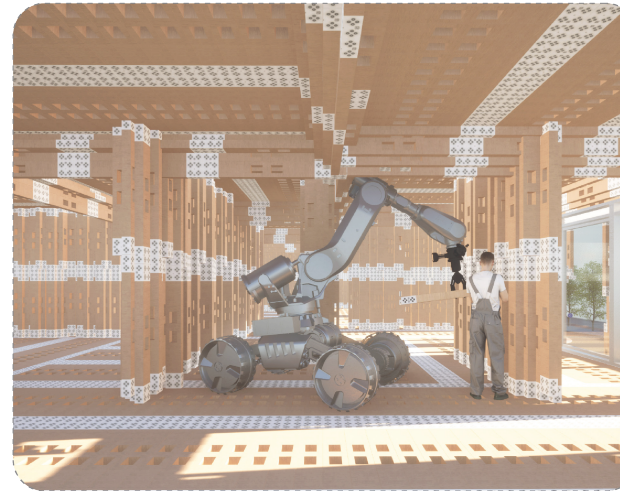


Figure 84. Interior views from the assembly to the inhabit. It is up to the inhabitant to customize her own space (by author, 2023).



Figure 85. Housing entrance from the community backyard (by author 2023).

5 Robotic Construction Workflow

5.1 Design-to-build

5.2 Off-site prefabrication

5.3 On-site assembly

5.3.1 Human as automated Assembly's tolerance

5.3.2 Workflow evaluation for HRC implementatio

5.3.3 Robot context-awareness

5.3.4 On-site robotic construction workflow

5.3.5 On-site assembly sequence

This section of the work expresses how the automated processes are inserted in the materialization, primarily to the discrete elements and secondly to the architectural design. It presents the robotic construction workflow designed for the specific site that could also be implemented in other scenarios. This research focuses on creating a design-to-build involving automated processes and HRC for timber housing structures, with the ability to disassemble them.

Robots have the capability to autonomously assemble, disassemble and reassemble entire houses by efficiently handling and distributing the necessary parts. These advanced techniques significantly reduce the need for human labor compared to conventional methods of construction and assembly, allowing for a more equitable distribution of resources throughout society (Claypool, 2019).

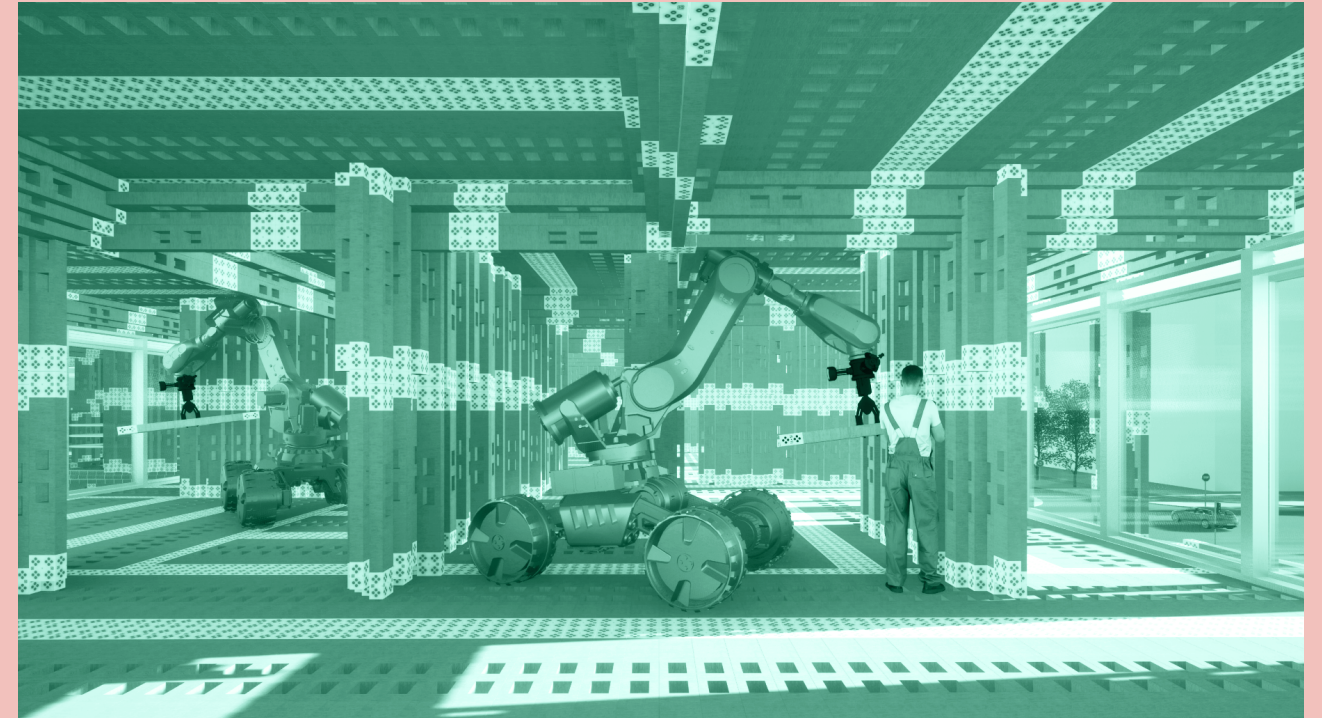


Figure 86. Human-robot collaboration on the construction site (by author, 2023).

5.1 Design-to-build

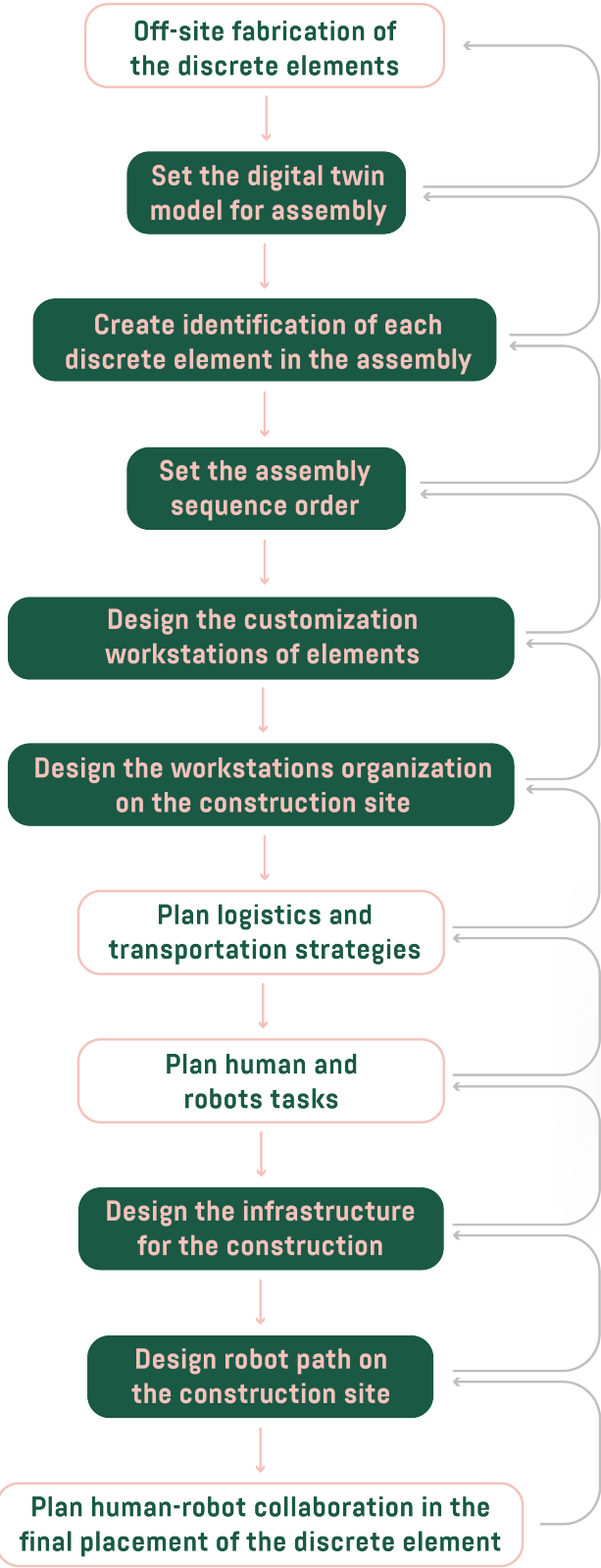
Based on the work of Kunic et al. (2021b), during this phase, the design is finalized and transformed into a comprehensive digital twin model that contains detailed information about all building discrete elements, including their quantity and precise position. It also specifies the joinery pattern for connecting these elements, involving all bolts out of the potential connections. In order to achieve that, an algorithm would assess and determine the optimal connections between each individual element and its adjacent counterparts, resulting in a continuous solid structure with the minimum number of bolts required. With a final placement position, the discrete element leaves its generic characteristic and becomes a customized piece within the whole generated customization. The manner of grouping them by having preplaced bolts for the final position is what gives its uniqueness and meaning within the whole.

With this information, the building's discrete elements are prefabricated for assembly by embedding steel nuts in them.

Additionally, referenced in Kunic et al. (2021b) work, an algorithm would be implemented to perform local stability checks on each element during the assembly process. Temporary supporting elements are generated as necessary to ensure stability. The digital design data provides a direct link to the robotic assembly process, facilitating a seamless and efficient connection between the design phase and the manufacturing process.

The design-to-build process addressed in this research is majorly divided into two main stages: [1] the off-site prefabrication and [2] the on-site assembly involving HRC. This hybrid approach combines the efficiency of both on-site and off-site construction methods (Koerner-Al-Rawi, 2020). As illustrated in Figure 87 and detailed in the following sub-sections, the off-site discrete elements' prefabrication is fully automated while the on-site assembly is done in a hybrid collaboration between humans and robots. Transportation is also taken into consideration in the off-site logistics preparation.

Design-to-build workflow



Digitizing the process brings significant benefits, as standardization of components ensures that modifications in the assembly also affect fabrication. The project's success relies on the interconnectedness of each design component (Koerner-Al-Rawi, 2020). This discretized approach not only minimizes material waste on the construction site but also leads to additional savings by eliminating the need for additional scaffolding or external building references (Willmann et al., 2016).

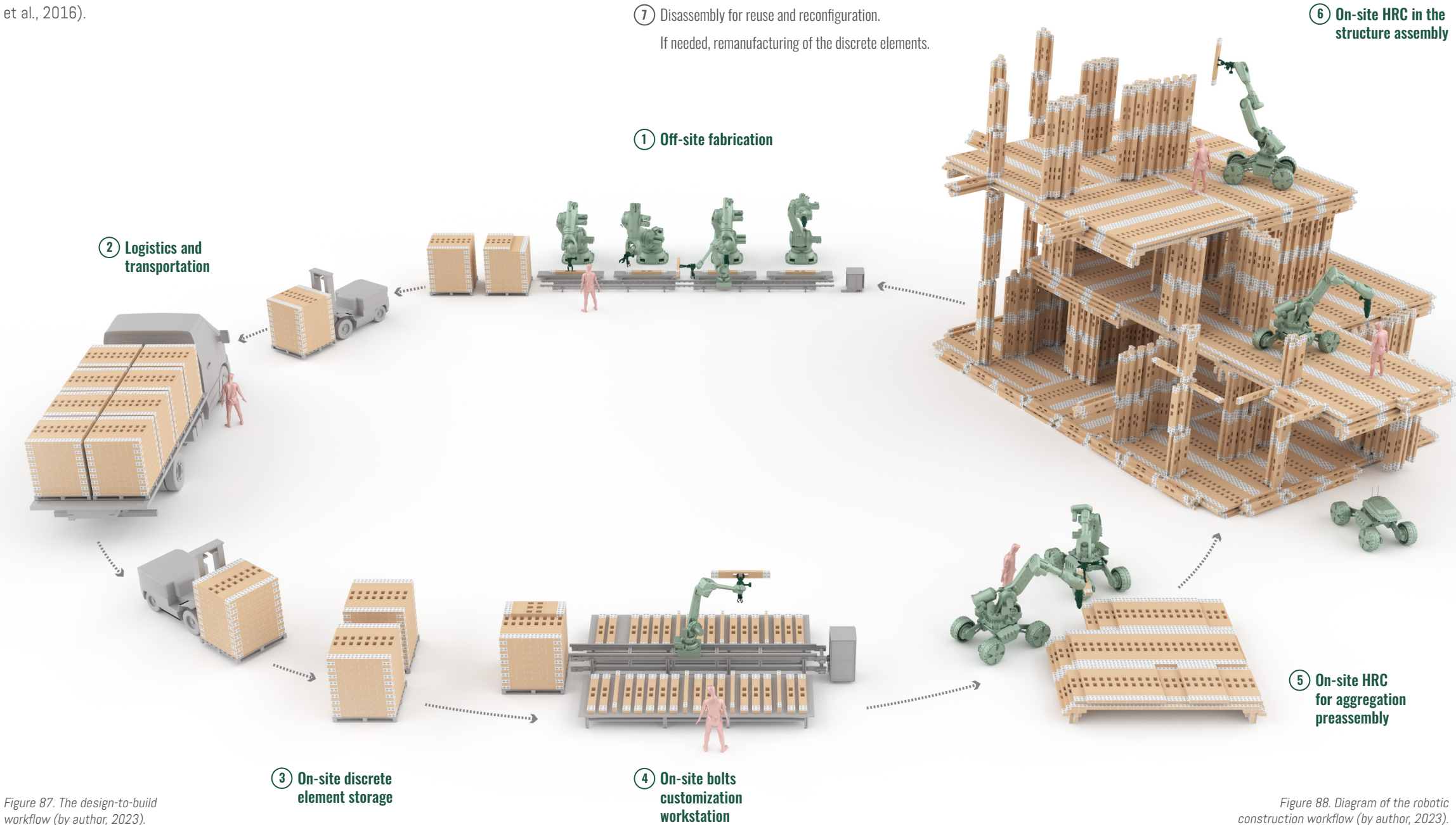


Figure 87. The design-to-build workflow (by author, 2023).

Figure 88. Diagram of the robotic construction workflow (by author, 2023).

5.2 Off-site prefabrication

The processes are fully automated at this stage of materialization according to the following diagram (Fig. 91). The output of the prefabrication phase is pallets of stacked discrete elements with around 3000kg depending on the density of the chosen wood and the maximum stack height of 24m. The stack is interlocked by different directions of layers (Fig. 89). The logistical transportation of these pallets takes into account their optimum layout inside the truck to minimize the transport (Fig. 90).

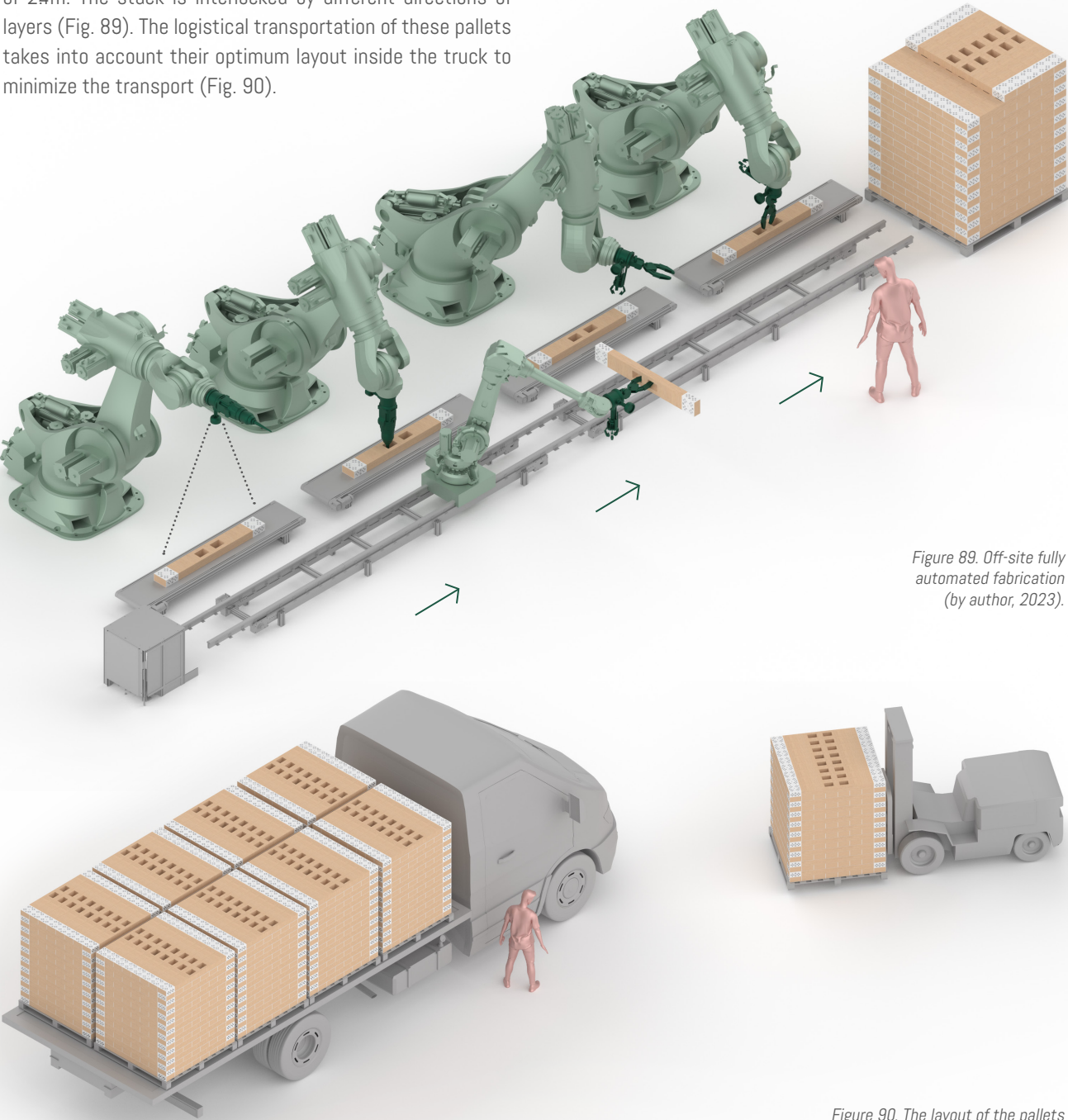


Figure 89. Off-site fully automated fabrication (by author, 2023).

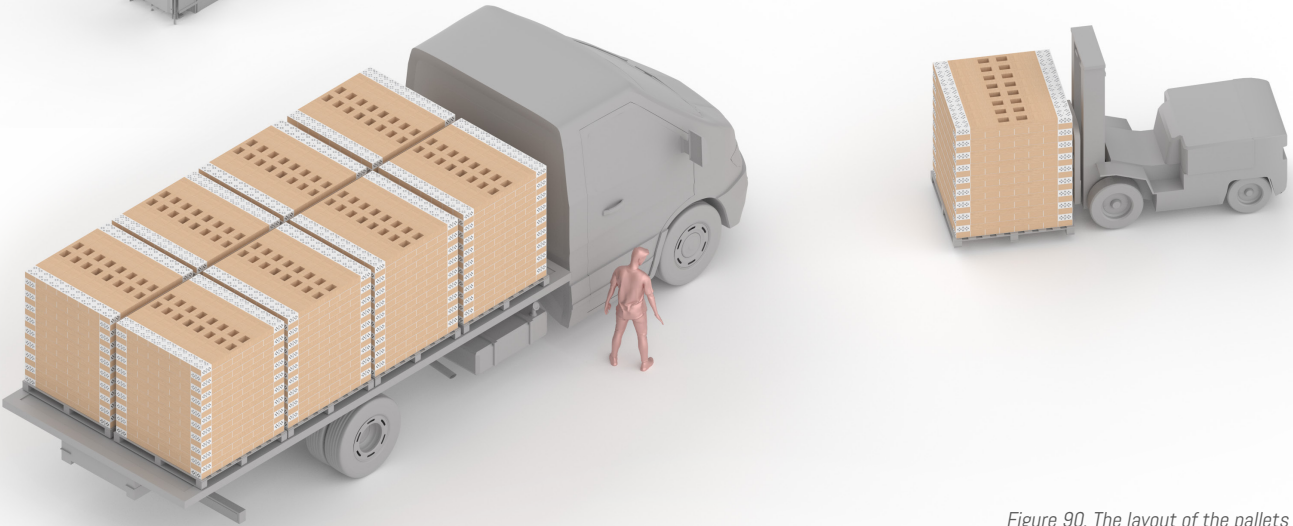


Figure 90. The layout of the pallets and it on a truck (by author, 2023).

Off-site fabrication workflow
Fully automated

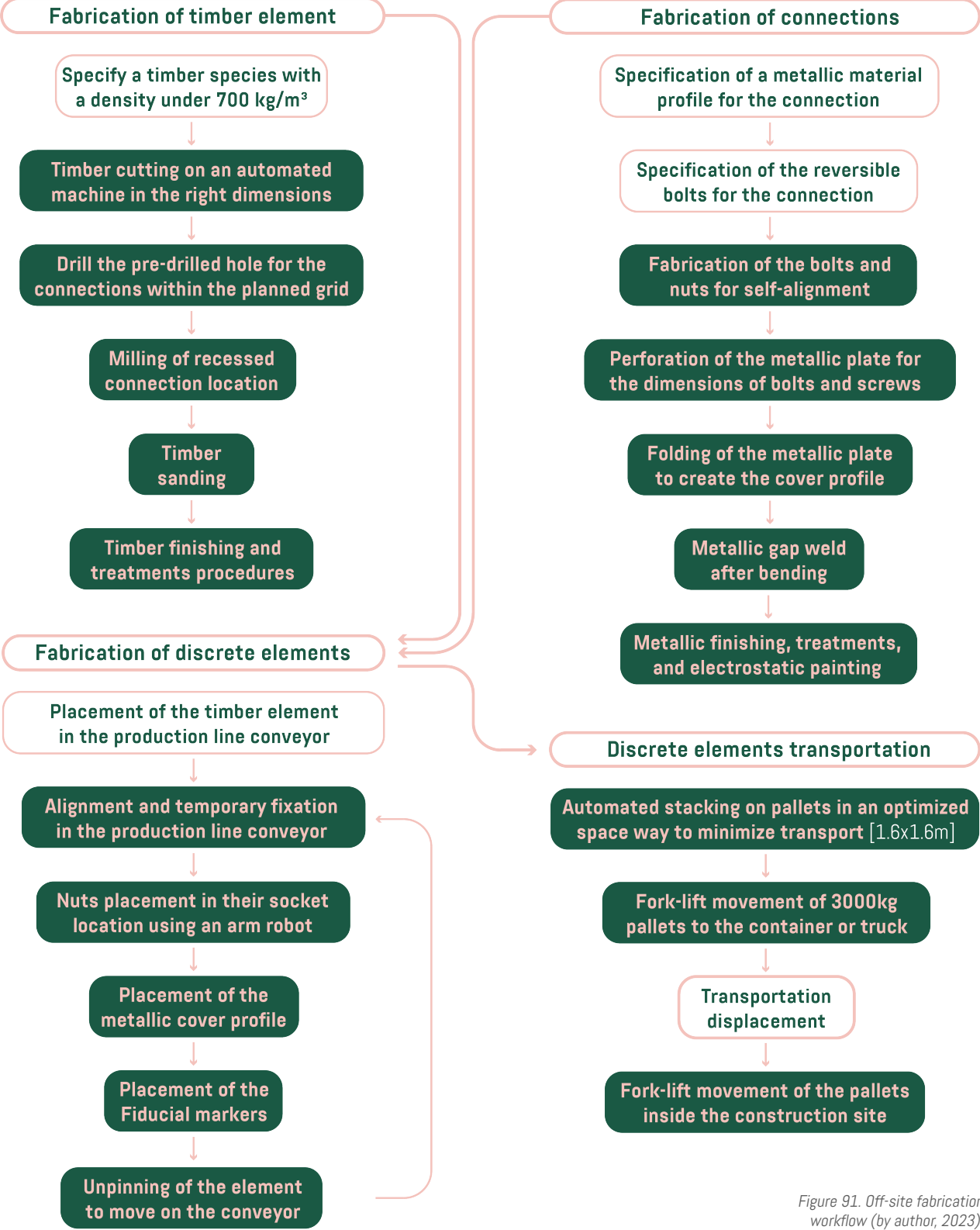


Figure 91. Off-site fabrication workflow (by author, 2023).

5.3 On-site assembly

At this stage of materialization, the proposed processes are done in a hybrid collaboration between humans and robots. The construction system syntax allows it to be completely assembled by robots, however, as human intuition for errors is something that is still far from being achieved by robots, the human-robot collaboration is proved to be efficient. The discrete element syntax also allows it to be fully assembled by humans due to its ergonomics. So both can work together taking their best expertise in cooperation as an ideal scenario, but also switch positions if the conditions changes. It is also important while considering the dynamicity of a construction site to embrace flexibility in the methods of construction to deal with possible changes. Therefore, the robot could take over the assembly when in hugely predictable situations, and the HRC approach in critical scenarios. This research considers the human as the main tolerance in the design assembly.

5.3.1 Human as automated assembly’s tolerance

The real world is not as precise as the digital environment. The difficulty of achieving precision on-site assemblies due to the constant dynamicity and vibrations that can occur is a factor that interferes with the implementation of fully-automated construction processes using mobile robots. Thus tolerances are important to be part of the design-to-build procedure. There are several strategies that can be considered when dealing with imprecision issues that aim to ensure accurate and reliable assembly despite variations in component dimensions and tolerances. These strategies are: [1] sensing and feedback systems, [2] adaptive control algorithms, [3] machine learning and ai, [4] active compliance, [5] error detection and correction, [6] precise fixturing and jigs, [7] process optimization and iterative refinement, and [8] human collaboration and intervention. Each construction system, assembly site, and task have unique requirements and challenges. It’s important then to assess the specific context and tailor the strategies to ensure successful assembly. This research incorporates conceptually four of these strategies.

- [1] Sensing and feedback systems relying on computer vision calibration;
- [2] Adaptive control algorithms taking the computer vision calibration output as a reference;
- [3] Precise fixturing in the self-alignment bolts design,
- [4] Human collaboration and intervention providing manual adjustments, fine-tuning, and quality inspections when necessary to achieve the desired precision.

Humans here are considered the main tolerance facilitator due to subjective judgment. As some of the automated assembly's variations can be attributed to human error factors, human intuition is the key to embracing the on-site assembly imperfections. Inside such construction sites, the labor conditions and skillsets are different from traditional construction. As it requires less manual work, training people to interact with this new typology of construction is needed. The organization of the construction site needs to have separate paths for robots and humans and an understanding of where and when they should be. Human error can add a lot of problems to the workflow if people are not trained. Thus, while working alongside robots, it is important to evaluate beforehand the division of the tasks inside the HRC workflow.

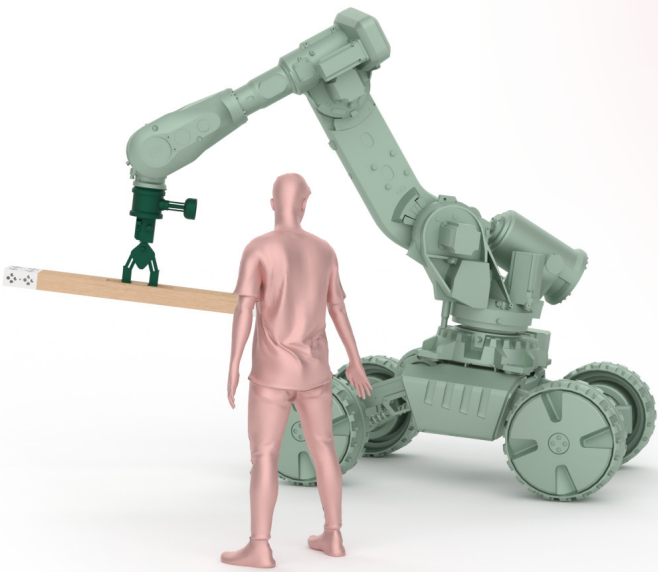


Figure 92. Human as automated assembly’s tolerance (by author, 2023).

5.3.2 Workflow evaluation for HRC implementation

STANDARD TASKS WORKFLOW DIVISION					EVALUATION CRITERIA			RESULTS
Main phases	Tasks		Actions	Resource	Complexity	Safety Level	Scale	HRC Applicability
					Simple - Complex	Safe 0 - 5	Local - Global	
Phase 1 Discrete elements and bolts preparation in the workplace	1.1	Discrete elements stack station preparation	Grip, move, and place	1 operator	Simple	0	Global	Human or robot
	1.2	20cm and 30cm bolts station preparation	Grip, move, and place	1 operator	Simple	0	Global	
	1.3	Scan identification of the discrete element surface if matches the correct placement's order	Visual recognition and assembly data interpretation	1 visual sensor 1 data processor	Complex	0	Local	Robot with computer vision camera connected to the digital twin model
	1.4	Pick the discrete element considering placing the correct surface facing up	Visual recognition and grip	1 visual sensor 1 operator	Simple	1	Local	
	1.5	Place the discrete element with the correct surface facing up for its customization	Approach move and place	1 operator	Simple	1	Local	
Phase 2 Customization of the discrete elements by placing the bolts	2.1	Pick the correct bolt length considering the final placement connection order	Assembly data interpretation, visual recognition, and grip	1 data processor, 1 visual sensor, 1 operator	Complex	0	Local	Robot with computer vision camera connected to the digital twin model
	2.2	Scan identification of the correct pre-drilled hole considering the final placement connection order	Visual recognition, alignment, and assembly data interpretation	1 vision sensor 1 data processor	Complex	0	Local	
	2.3	Place the bolt in the correct hole	Approach move, alignment, place, and bolt	1 visual sensor 1 operator	Complex	0	Local	
	2.4	Repeat this phase until all the needed bolts are placed in their connection holes	Repeat loop	1 data processor	Simple	0	Local	
	2.5	Check if the bolts are stably pre-fixed in the holes	Visual recognition and assembly data interpretation	1 visual sensor 1 data processor	Complex	0	Local	
Phase 3 Aggregating the discrete element in the final placement	3.1	Scan identification of the customized element in the order of placement to plan the trajectory	Visual recognition, assembly data interpretation, plan trajectory move	1 visual sensor 1 data processor	Complex	0	Local	Human intuition check with augmented reality
	3.2	Pick the customized discrete element	Grip	1 operator	Simple	1	Local	
	3.3	Movement on space calculating the approach point for final placement	Trajectory move	1 operator	Simple	2	Global	
	3.4	Position the discrete element on final placement	Approach move	1 operator	Complex	2	Global	
	3.5	Check if the placement is satisfactory to bolt	Visual recognition and assembly data interpretation	1 visual sensor 1 data processor	Complex	0	Local	
Phase 4 Next loop position	3.6	Correct the position if it exists displacement	Move	1 visual sensor 1 operator	Complex	1	Local	Human calibration using Nordbo Mimic Kit
	3.7	Bolt all the bolts	Bolt	1 operator	Simple	0	Local	Human or mobile robot with computer vision
	4.1	Repeat all the previous phases until all the discrete elements are in their final placement	Repeat loop	1 data processor	Simple	0	-	-

Figure 93. Table 1. Workflow evaluation for HRC implementation (by author, 2023). Table based on Loo (2022, p.100).

5.3.3 Robot context-awareness

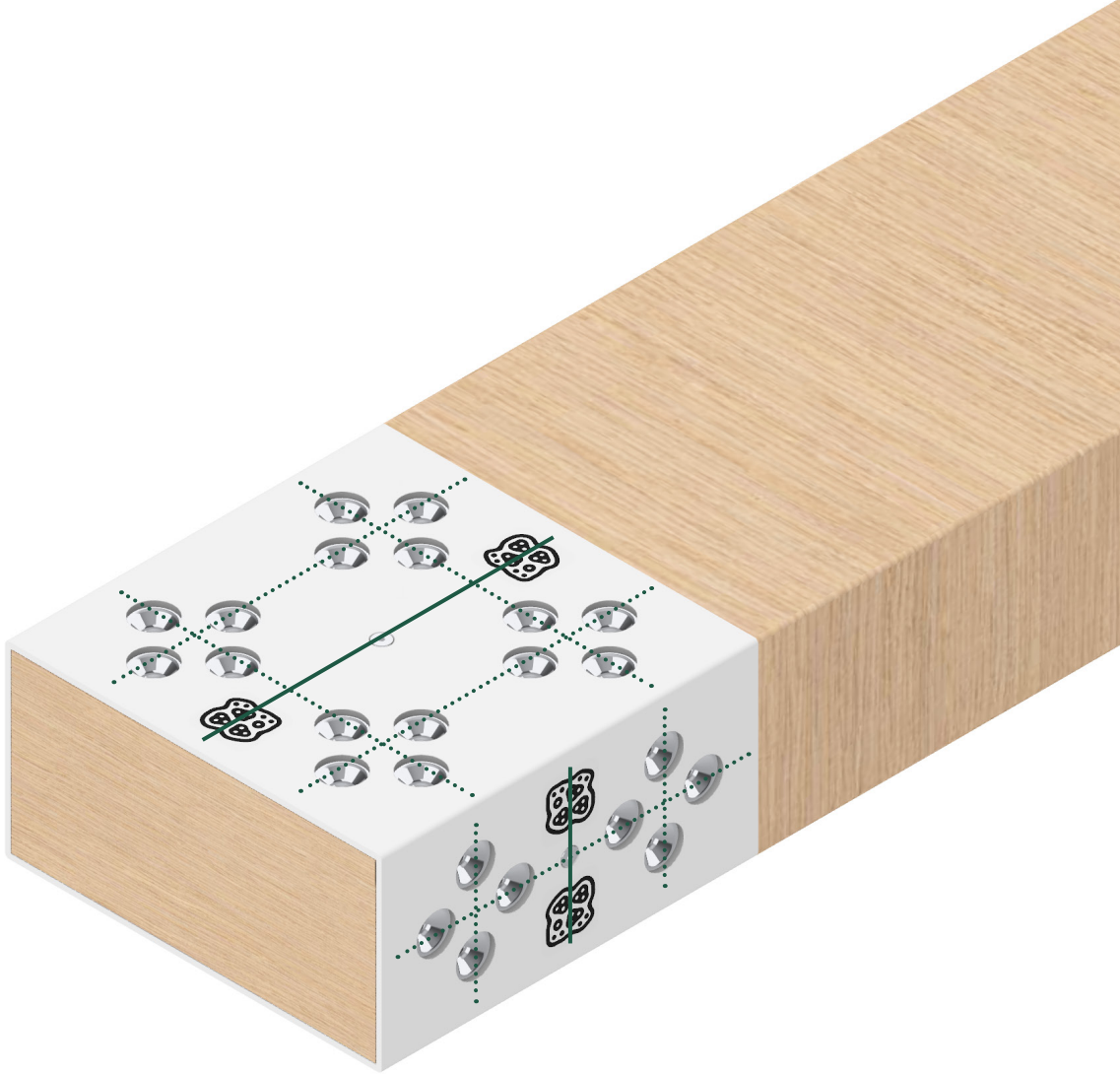
The process of discretizing operations allows for natural division, enabling multiple assemblers to work concurrently in parallel (Gershenfeld et al., 2015). Inside the proposed cooperative assembly procedure, task distribution and its evaluation are key factors for successful implementation. Based on the work of Loo (2022), Table 1 (Fig. 93) expresses the standard tasks workflow division in detailed tasks and its evaluation according to the Evaluation Criteria addressed in **section 2.5** of this research. The workflow evaluated is the standard procedure that will repeat each time to place a discrete element. The off-site prefabrication of the discrete elements was excluded from the evaluation because its standardized production will be fully automated. The pieces arrive at the construction site ready to be customized for assembly. Therefore, the standard procedure comprises 4 main phases: **[1]** Discrete elements and bolts preparation in the workplace, **[2]** Customization of the discrete elements by placing the bolts, **[3]** Aggregating the discrete element in the final placement, and **[4]** repetition of next loop positioning. Every procedure consists of several stages where the determination of actions, resources, and equipment takes place. Through this evaluation, the suitability of the group of tasks for the HRC application is determined. Based on Loo (2022), the assessment of each task or group of tasks was based on three criteria: complexity, safety level, and scale.

The HRC application showed to be advantageous in scenarios where complex tasks require error checks, correction, and approval to keep running the procedure. By interpreting the assembly data by scanning the fiducial codes of each discrete element, augmented reality is used to check the position of the bolts are in the correct places and that the pieces are in their correct final position. As assembly tolerance, the human also performs error correction of pieces that have deviations by moving them through the robot without touching them using the **Nordbo Mimic Kit** or a similar system. In the event of a failure, the robot transitions to free drive mode, enabling the operator to manually guide the tool to the correct position and assist the robot in completing the task. By utilizing the robot's ability strength of calculate precise spatial positioning, heavy and placement tasks are performed by the robot, such as grip, move, and place. This HRC procedure will be elaborated in greater detail in the subsequent **section 5.3.4** about the whole on-site robotic construction workflow.



Figure 94. Robot computer vision and sensing checking the elements for calibration (by author, 2023).

In order to implement a successful automated process on the construction site, mobile robots need to be capable of recognizing their surrounding environment. Due to the presence of substantial deviations that arise between virtual models and physical prototypes (Rogeauea et al., 2020), context-aware robots move around while interpreting relevant information in order to make informed decisions and adapt their behavior. This recognition includes factors such as the physical context (location, obstacles), social context (human interactions), temporal context (time and events), and task context (specific goals and constraints). To achieve that, robots are supplemented by various sensors and technologies, along with machine learning techniques, to perceive and analyze contextual data. These include cameras, depth sensors, microphones, GPS, inertial measurement units, computer vision, natural language processing, sensor fusion, and other environmental sensors. Being context-aware allows robots to navigate, recognize objects, interact with humans, and perform tasks more effectively and intelligently in real-world environments anticipating changes, and handling uncertainties.



- ① Surface recognition ———
- ② Bolt holes calibration ······

Figure 95. Recognition of vertical and horizontal surface and bolting role calibration of discrete elements (by author, 2023).

As a concept, the research's proposed workflow adopts context-awareness gadgets to supplement the automated construction process. Computer vision relying on cameras and sensors is the main source of recognition while moving and performing tasks of pick and place. The fiducial codes embedded in the discrete design connection facilitate the awareness of the discrete element surfaces and calibration of the bolts' position in the holes (Fig. 95). Based on the work of Rogeauea et al. (2020), an algorithm would operate by capturing an image of the designated fiducial codes on the discrete elements' surface to determine its orientation and position's coordinates and then storing these calculated results in a text file. Subsequently, the robot can access

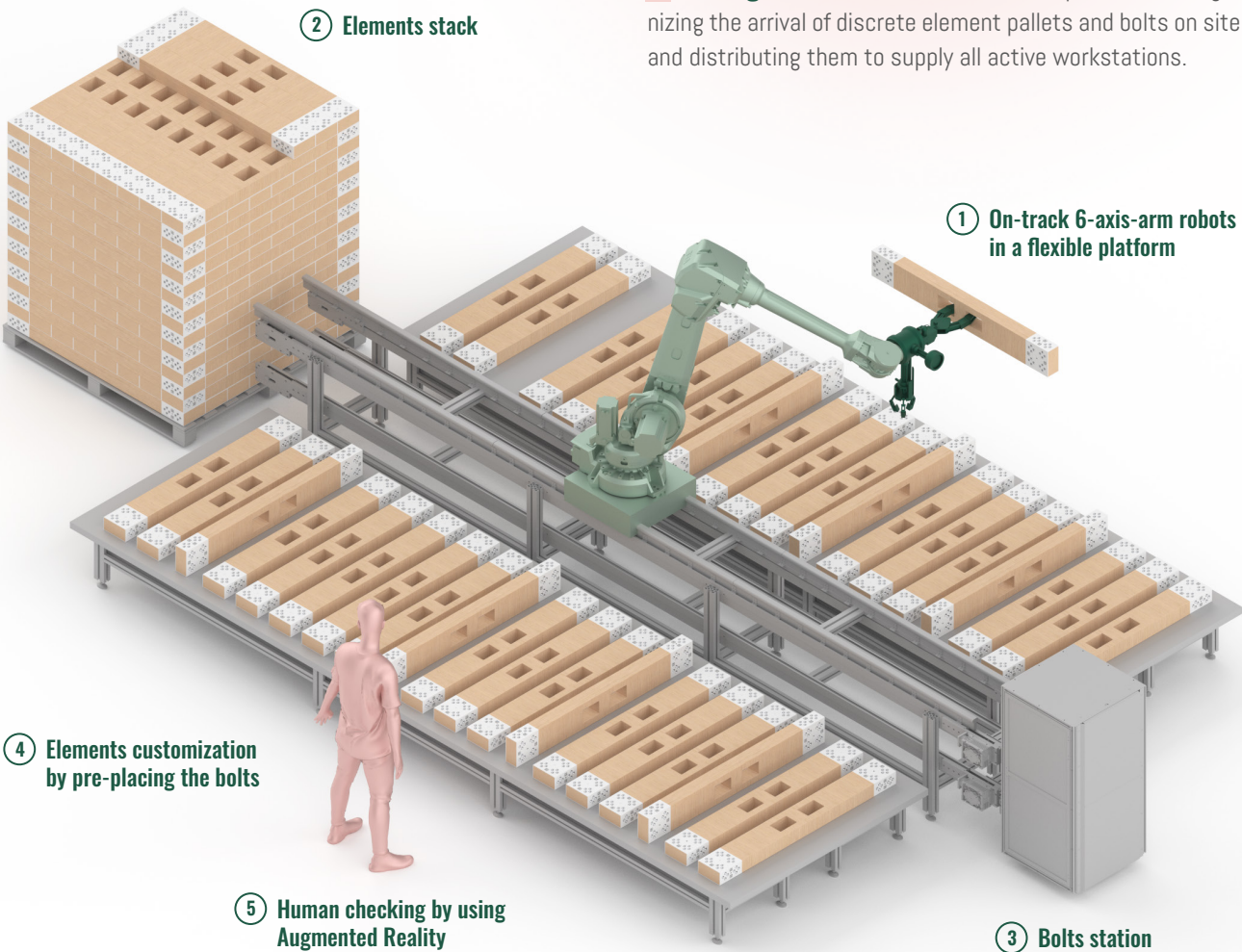
this information to update its trajectory accordingly. All the on-site robots would be in real-time connected to the central digital twin model of the assembly to take into consideration the correct assembly order, the bolts' positions according to the final placement, and which elements are already placed to be counted as support for the next elements and understood as a possible collision obstacle.

The whole on-site automated construction workflow is presented in the next **section 5.3.4**. Due to time constraints, a simplified calibration approach will be conceptually adopted, omitting complex algorithms and equipment mentioned in the extensive existing literature describing different calibration methods.

5.3.4 On-site robotic construction workflow

Each construction site demands different customized solutions for its automated construction workflow. Taking the architectural design presented in the previous chapter as starting point, the proposed assembly process is divided according to the spatial reachability of each involved robot's workplace. These different workplaces perform the standard tasks presented in section 5.3.2 according to their physical context. Therefore, there are 5 workplaces frameworks: [1] on-track 6-axis-arm robots in flexible platforms fixed on the ground, [2] 6-axis-arm mobile robots on the ground, [3] automated crane, [4] 6-axis-arm mobile robots throughout the building structure, and [5] logistics mobile robots. The first three workplaces are interconnected due to each of them is responsible for specific tasks inside of the same group of standard tasks.

- [1] **The on-track 6-axis-arm robots in flexible platforms fixed on the ground** are responsible for the discrete elements and bolts preparation in the workplace and their customization by placing the bolts in the correct position.
- [2] **The 6-axis-arm mobile robots on the ground** are responsible for aggregating the discrete elements in their final assembly position because they have broader reachability.
- [3] **The automated crane** is responsible for the global movement of the horizontal aggregated components in their final placement location within the structure.
- [4] **The 6-axis-arm mobile robots** throughout the building structure are responsible for the assembly of the vertical components including all the required phases. This work frame requires to have multifunctional end effectors.
- [5] **The logistics mobile robots** are responsible for organizing the arrival of discrete element pallets and bolts on site and distributing them to supply all active workstations.

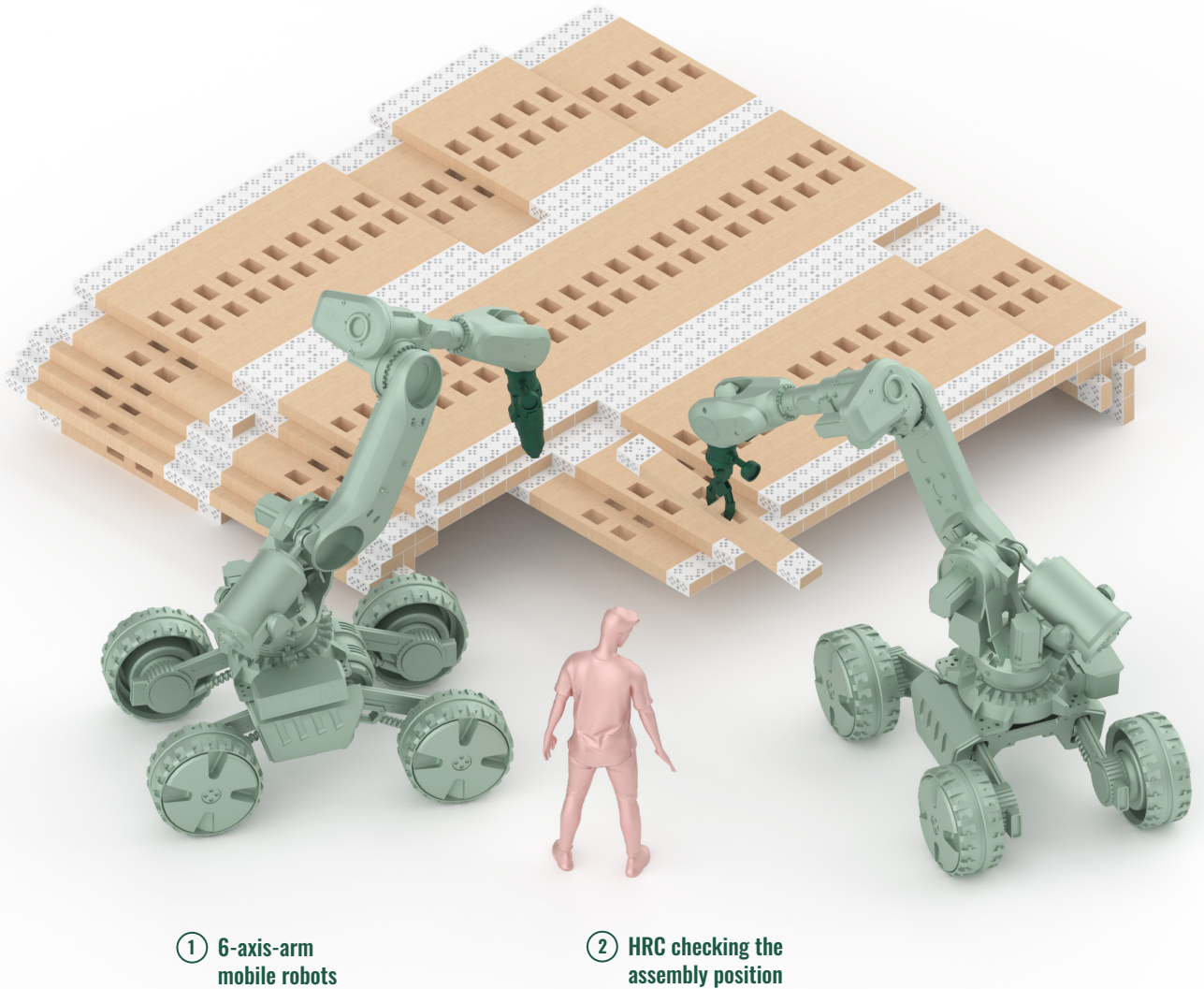


These robots move around the whole site requiring specific path roads and temporary vertical circulations. The horizontal and vertical aggregations are assembled simultaneously in their different workstations. While the horizontal aggregation is produced on the ground floor and placed afterward by the crane, the vertical aggregation is assembled at its local final destination. These workplaces are illustrated as following Figure 96-101.

Figure 96. The on-track 6-axis-arm robots in flexible platforms fixed on the ground (by author, 2023).

Figure 97. The 6-axis-arm mobile robots pre-assembling on the ground (by author, 2023).

The HRC is present in the four first workstations. Figure 98 represents the detailed on-site HRC assembly workflow that uses human intuition as assembly tolerance. This representative diagram combines the standard tasks workflow division (Fig. 93, Table 1) with the automated workstations and their procedure of collaboration. The human carries out the checking process by collaborating with the robot, ensuring that the pieces are in the right place so they can be bolted.



On-site assembly workflow

Human-robot collaboration

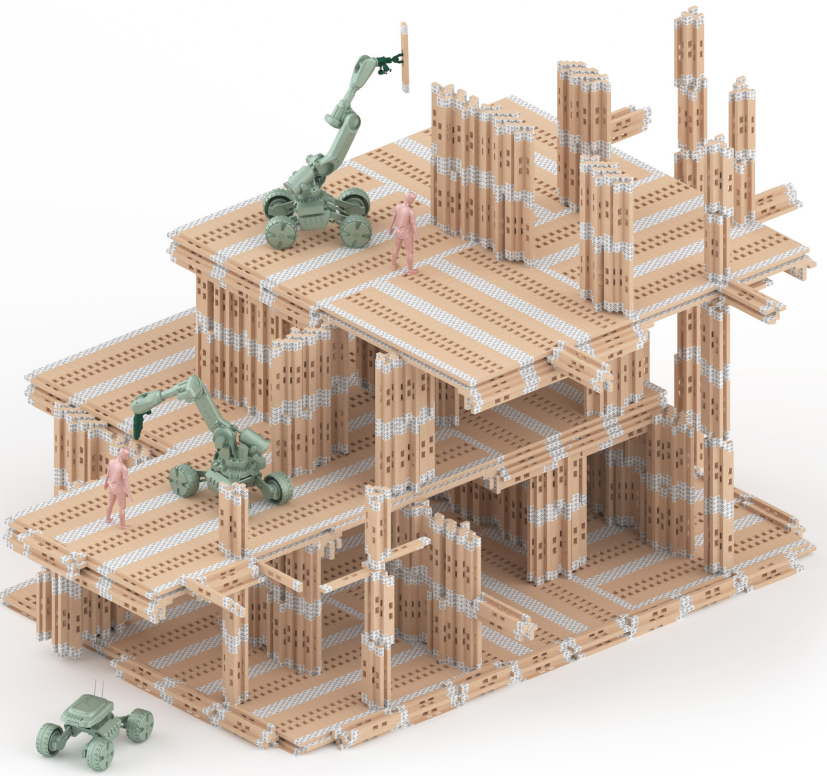
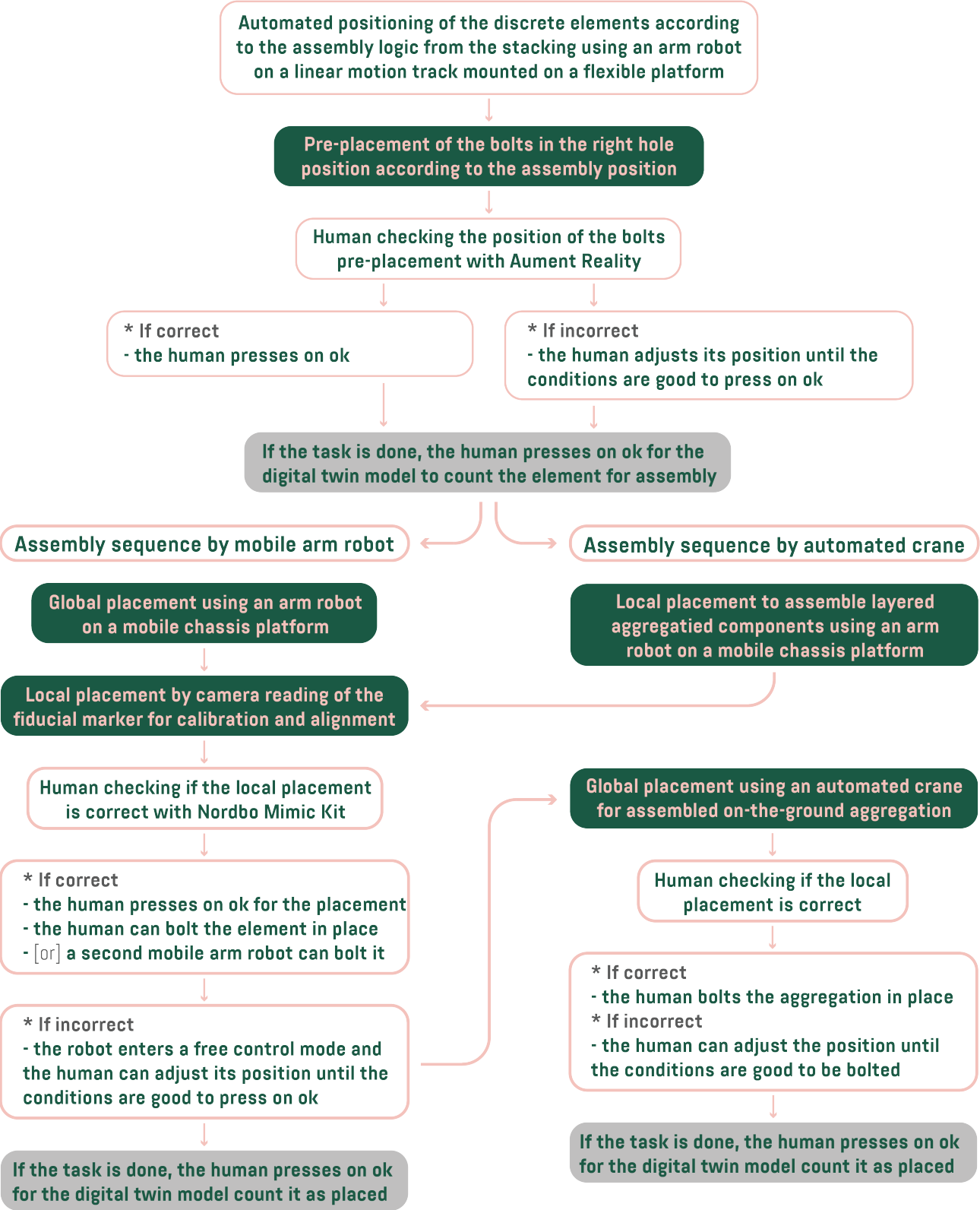
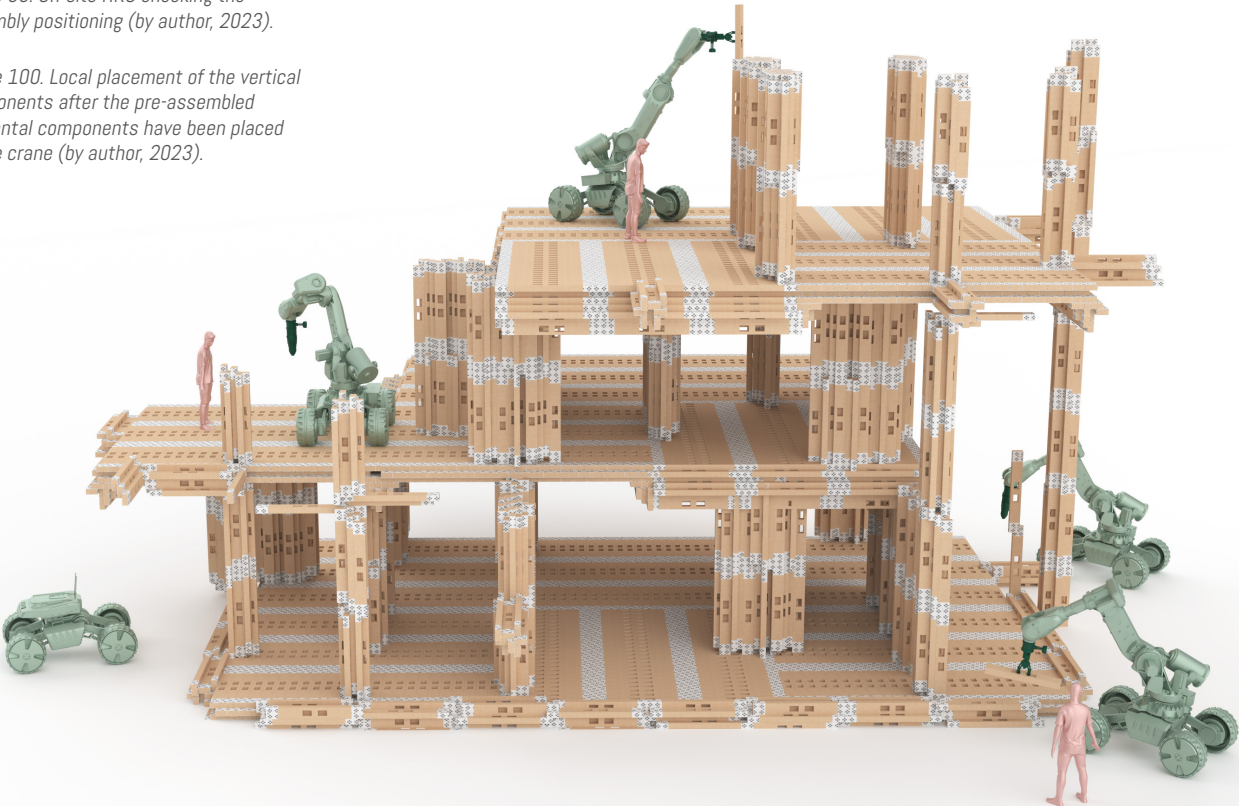


Figure 98. On-site assembly workflow (by author, 2023).

Figure 99. On-site HRC checking the assembly positioning (by author, 2023).

Figure 100. Local placement of the vertical components after the pre-assembled horizontal components have been placed by the crane (by author, 2023).

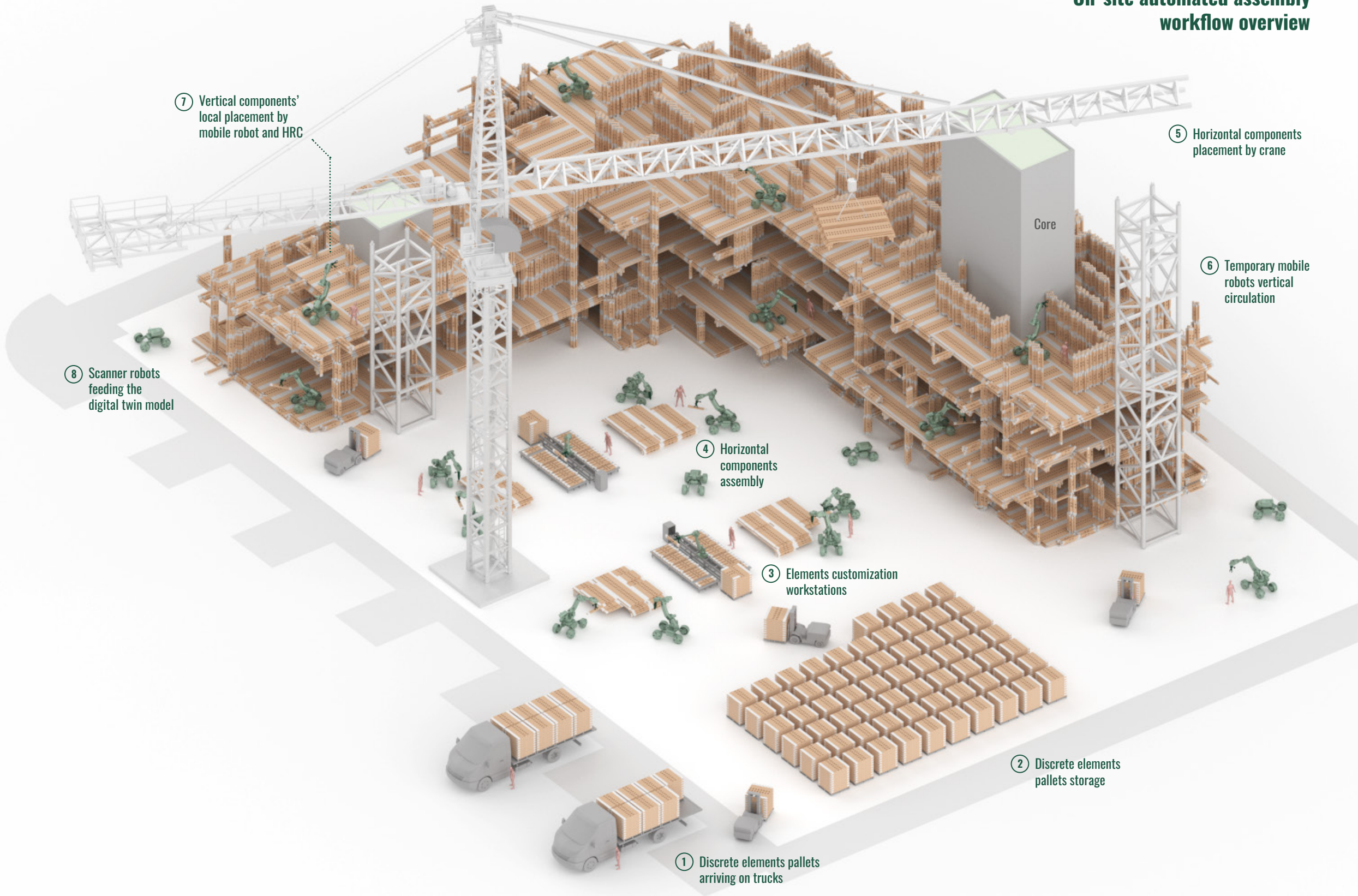


5.3.5 On-site assembly sequence

After understanding how is the on-site robotic construction workflow, this section presents the assembly sequence of the architectural design. First of all the foundation needs to be dimensioned in the same grid framework of the discrete element and set connector points in the same syntax of the automated construction system. The subsequent phases would follow as illustrated in Figure 101. As explained in the last section, the horizontal aggregated component is produced on the ground floor and placed afterward by crane, while simultaneously the vertical aggregated component is locally assembled at its final destination. The vertical aggregated components can just start to be assembled once the horizontal components are placed enabling the movement of mobile robots on top of it.

Figure 101. Overview of the on-site automated assembly workflow (by author, 2023).

On-site automated assembly workflow overview



6 Prototypes

- 6.1 1:1 Connection prototype
- 6.2 1:5 Robotic workflow prototype
 - 6.2.1 Discrete elements fabrication
 - 6.2.2 Prototyping phases
 - 6.2.3 Robot experiment setup
 - 6.2.4 Robot Simulation and Control
 - 6.2.5 HRC assembly prototype
- 6.3 1:10 Prototype

In this section of the research, three prototypes in different scales were developed in order to test and prove the design concept. These experimental prototypes were done as part of the research by design to understand the intrinsic aspects of the discrete element design and its connection to then to be able to draw conclusions about the construction system’s materialization. Each prototype scale looks for a different testing purpose. The **1:1 prototype** scale looked for testing the connection and the bolts in the real dimensions taking into consideration all their details. The **1:5 prototype** tested the standard tasks of the robotic construction workflow, the human-robot collaboration, and the self-alignment feature embedded in the bolts and nuts design. The **1:10 prototype** looked for testing the assignment of a function to the generic elements and the aesthetic quality of a larger-quantity aggregation.

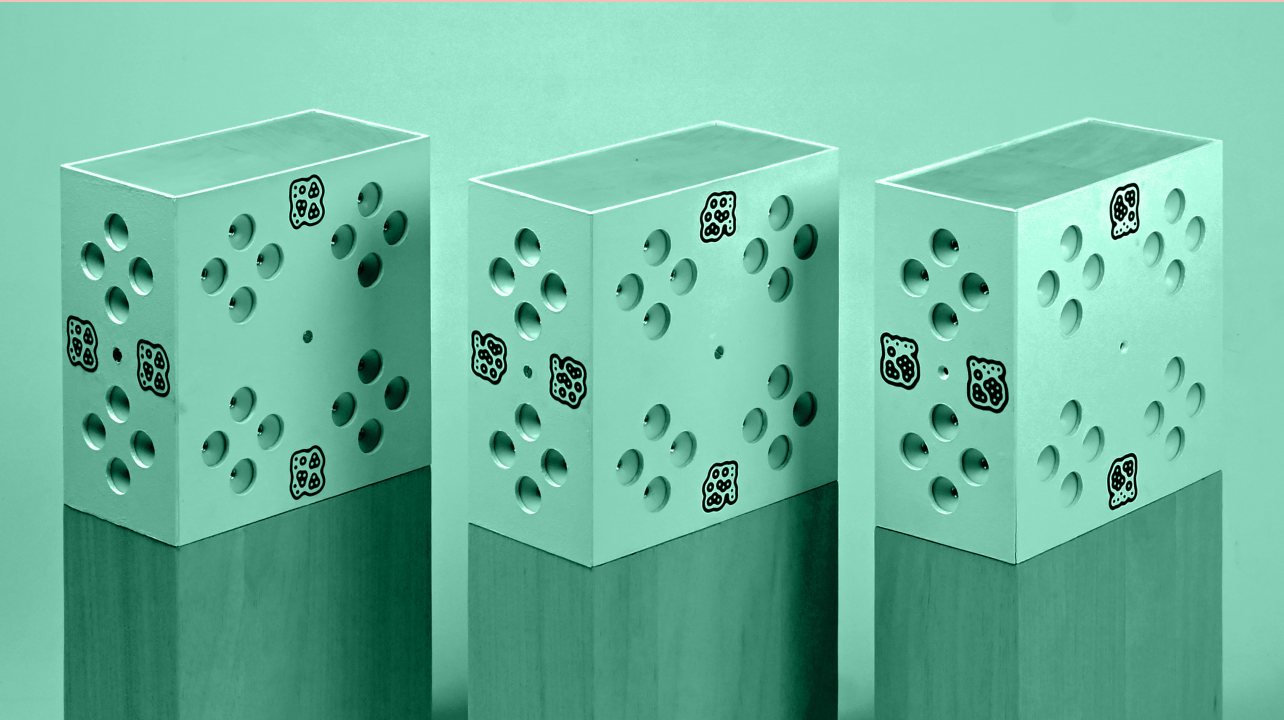


Figure 102. 1:1 prototype discrete elements (by author, 2023)

6.1 1:1 Connection prototype

The 1:1 prototype scale looked for testing the connection and the bolts in the real dimensions taking into consideration all their details. As illustrated in Figure 102, three wood elements of **10x20x40cm** representing three entire discrete elements were produced to check the connectivity among them. All the fabrication procedure phases were performed to achieve the closest precision possible to what is their industrial production. The assembly procedure was manual due to the available robot for the research having a weight limit of 5kg.

The first challenge of the prototype was to understand the bolts and nuts materialization as they have specific shapes that cannot be found in the current market. Many 3D printing tests were done in order to achieve the precision that the pieces required for the assembly (Fig. 103). Resin 3D printing showed to be the best 3D printing procedure to attain resolution. Therefore, a hybrid solution of resin 3D-printed with metallic M6 bolt threads and nuts was done to conduct the

research experiment. The bolts' heads and the overall nut geometry were 3D printed considering later connecting the metallic M6 bolt threads and metallic M6 nuts respectively.

After this stage, Abachi was selected as the wood type to shape the volume of each element. CNC milling procedure was used to obtain precision in the materialization of the sockets to, later on, receive the resin nuts. Different drills were used in the process, one for the CNC milling and a longer one to drill the holes throughout the wood profile section. A 2mm tolerance was adopted to drill the holes crossing the section. The tubular metallic profile is represented by glued-laser-cut 3mm MDF with the circles’ position precision matching the nuts inserted in the sockets.

The prototype was successful and proved the design concept. Inside the limitations of the prototype, the connection showed to be stable and stiff enough in many positions configurations. The bolts’ occupancy provided self-alignment

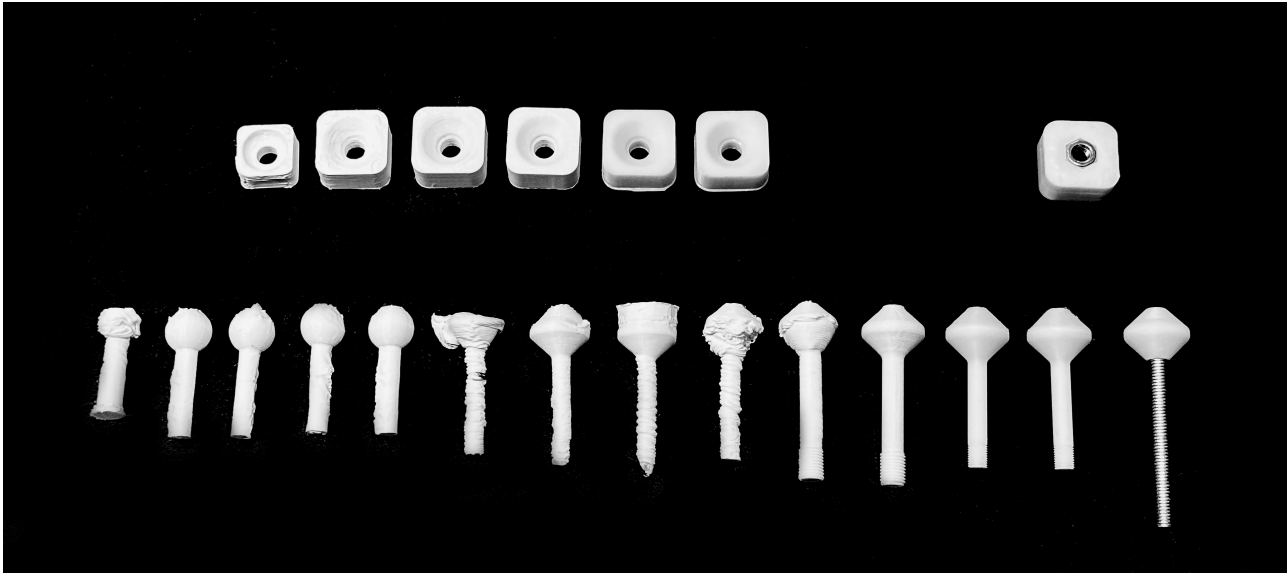


Figure 103. 3D printing tests for precision achievement (by author 2023).

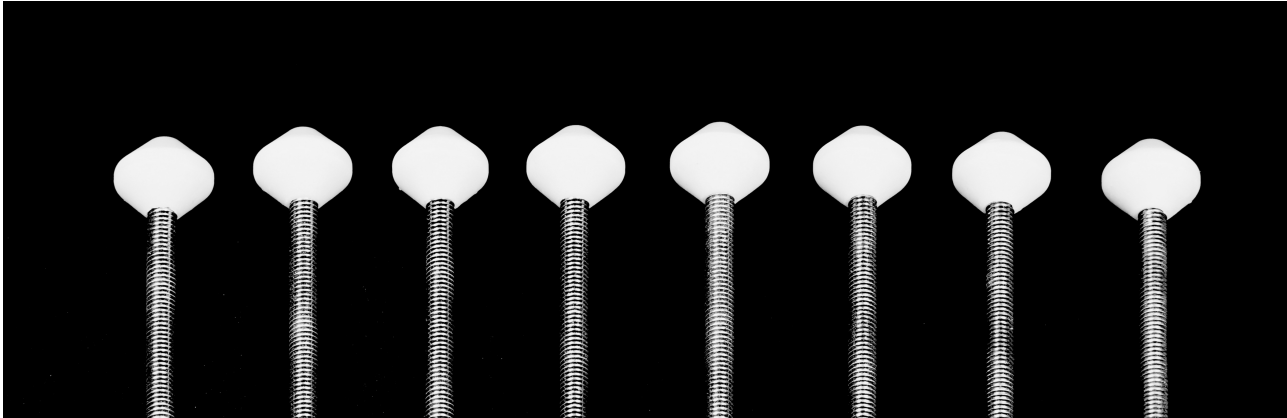


Figure 104. Hybrid solution of resin 3D printed bolt heads and M6 standardized metallic threads (by author 2023).



Figure 105. Resin 3D printed nuts with M6 nuts (by author 2023).



Figure 106. Laser-cut wood representing the metallic profile (by author 2023).



Figure 107. CNC milled sockets for nuts placement (by author 2023).

for the next piece placement and fit perfectly between the discrete elements. The implemented tolerance of 0.5-1mm in the overall dimensions proved to be enough to create precision in the assembly. However, sagging deformations should be taken into consideration in larger aggregations. While assembling the prototype, the bolting procedure showed to be time-consuming due to the metallic threads used being all the length of the bolts. As the bolts need to go through 4 nuts, the procedure would be improved if the design of the bolt have treads on specific places where it will connect with the nuts. It was observed that the more nuts the bolts connect, the connection get stiffer.

The connection design would benefit from other design iterations by taking into consideration larger bolt heads to embrace more deformation, less number of preinstalled nuts, and a different metallic cover design.



Figure 108. All parts of the 1:1 connection prototype (by author 2023).

Prototype 1:1

10 x 20 x 40cm



Figure 109. Discrete elements used in the prototype (by author 2023).



Figure 110. 1:1 connection prototype assembled (by author 2023).

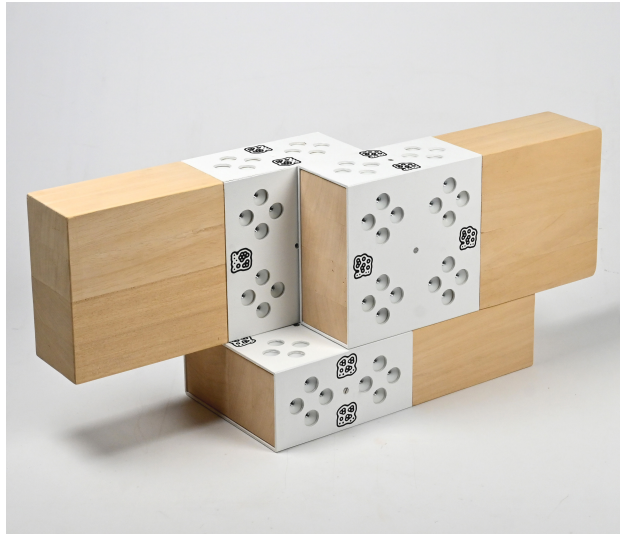


Figure 111. Possible connection configuration 1 (by author 2023).



Figure 112. Possible connection configuration 2 (by author 2023).



Figure 113. Possible connection configuration 3 (by author 2023).



Figure 114. Possible connection configuration 4 (by author 2023).



Figure 115. Connection prototype vertically assembled (by author 2023).

6.2 1:5 Robotic workflow prototype

Of the prototypes, the 1:5 scale prototype is the only one that uses an automated process by using a Universal Robots UR5 6-axis robot arm. It was envisioned to test 3 aspects of the design: [1] the standard tasks of the robotic construction workflow, [2] the human-robot collaboration, and [3] the self-alignment feature embedded in the bolts and nuts design. The UR5 robot has an outreach diameter measuring 850mm and the ability to handle loads weighing up to 5kg. It is classified as a medium-sized collaborative robot. So the balance between these limitations and the discrete element sizes was the main reason for the 1:5 scale selection to perform the test.

However, an adaptation of the discrete element design was needed in order to achieve the testing goals. The real proposed connection has 4 bolting holes per point of connection, summing 24 predrilled holes crossing the wood section. This number of holes would be difficult to materialize on the 1:5 scale considering the standard bolts and nuts sizes crossing the section. The wood would have so many holes in a small volume and the bolts and nuts would need to be minuscule making it difficult for the robot's gripping proce-

dure. Therefore, a balance between the number of holes and their sizes to fit the bolts and nuts was the guideline to create a specific design connection for this prototype. Here, **the 4 bolting holes are represented by 1 bolting hole per point of connection** as illustrated in Figure 116.

This simplification decreases the number of possible connections because one point can be occupied to place the next one, but enables the experiment to run with bolts big enough to be grabbed close to what would be an industrial production. The adapted connection follows the same grid principle of connection, so the same aggregation rules are applied to the experiment. Due to the volume scarcity, the bolt and the nut occupy the same space in the connection design for this experiment, so each discrete element already has a specific place in the assembly before the bolt placement.

The design of the connection incorporated tolerances of 1mm. While robots initially appear to offer precise control, even minor inaccuracies can accumulate and result in misaligned joints (Bouza & Asut 2020).

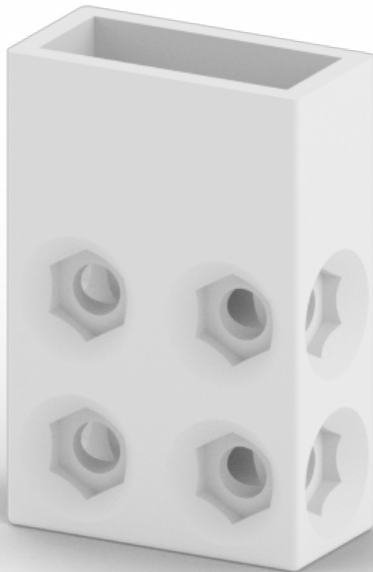


Figure 116. 1 bolting hole per point of connection representing the 4 bolting holes (by author 2023).

6.2.1 Discrete elements fabrication

As described in the last section, the fabrication of the discrete elements in the 1:5 scale is an adaptation of the proposed design. Therefore the materialization processes were used specifically for this experiment. Instead of CNC milling the sockets and holes in the wood, all the connection points were combined into one connector geometry and later 3D printed using resin (Fig. 119). This condensed geometry was designed to accept either the self-alignment bolts or the standard M6 nuts in its holes. The decision of using resin 3D printing was due to simplifying the materialization process of each beam end while guaranteeing high precision and quality in the connection.

Linking the two connectors, the center of the beam was made using the wood Abachi considering the internal fixture with the 3D printed geometry. Once the wood pieces were in the correct dimension, the gripping central holes were done by CNC machine aiming precision. This mechanized process enables rapid production, exceptional precision, and consistent repeatability, all of which are essential for establishing a reliable assembly process involving a vast array of elements (Kunic et al., 2021b). The 1:5 bolts' materialization was made by using the hybrid solution of resin 3D printed heads and standard M6 treads, similar to the 1:1 scale. The nuts here were represented by standard M6 nuts.

16 Discrete elements
32 Connections
32 Bolts

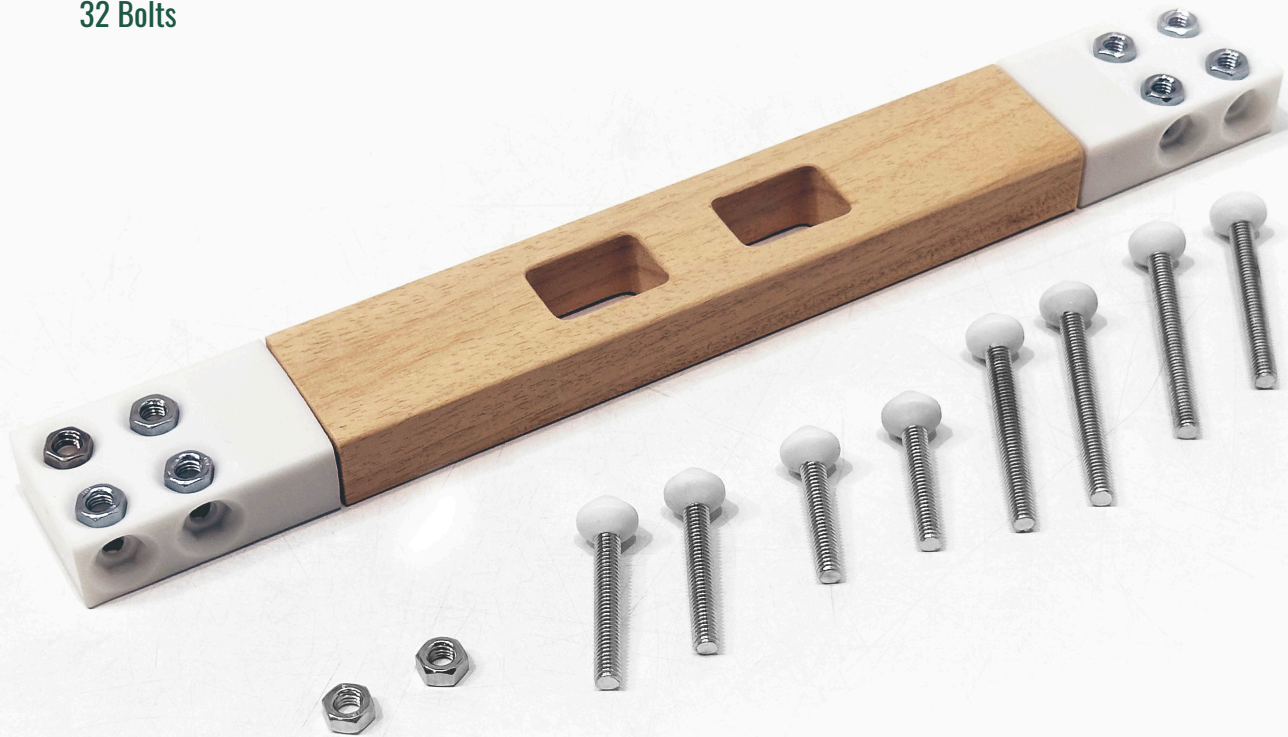


Figure 117. Hybrid solution of resin 3D printed connections and bolt heads, wood CNC milling, and standardized metallic M6 nuts and threads (by author 2023).



Figure 118. All parts of the 1:5 prototype before being assembled (by author 2023).

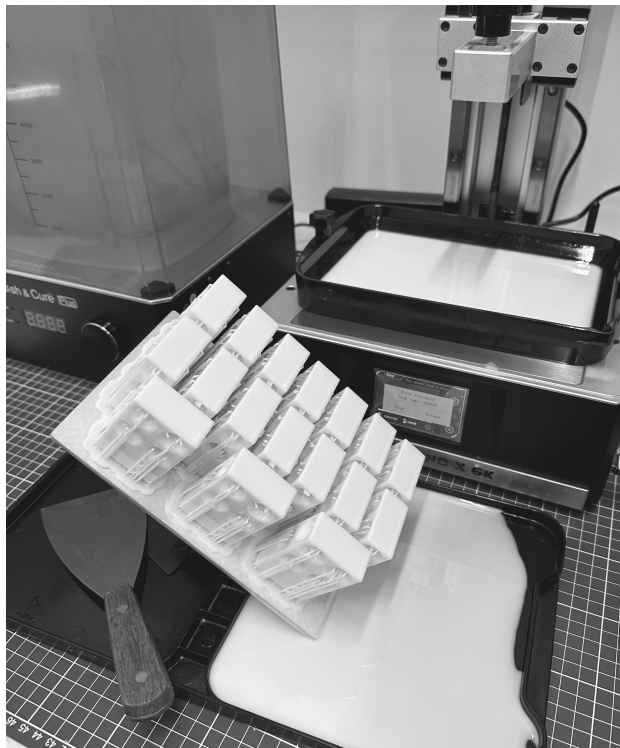


Figure 119. 3D printing on the resin (by author 2023).

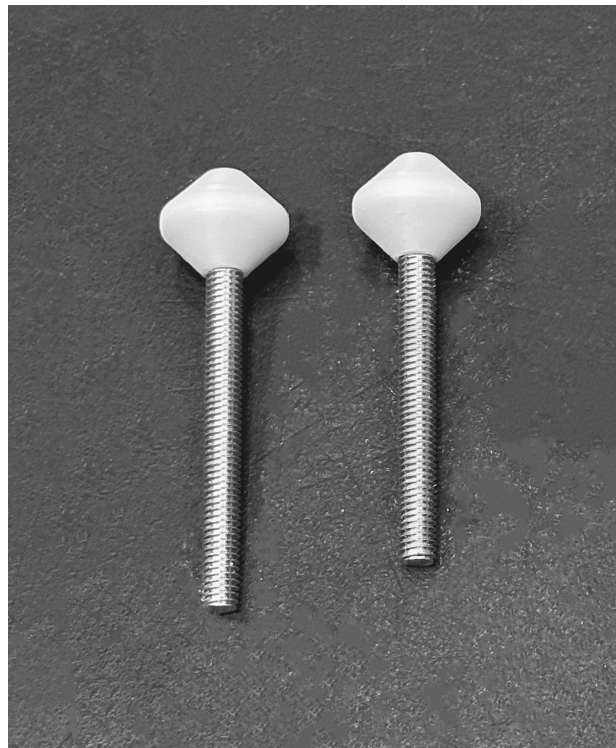


Figure 120. 1:5 hybrid solution assembled bolts (by author 2023).

6.2.2 Prototyping phases

In order to test HRC standard tasks division (Fig. 93, Table 1) explained in **section 5.3.4** about the on-site robotic construction workflow, the 1:5 prototype phases use a similar procedure approach inside of what was possible to test inside this research's timeframe. After the materialization of the pieces, there are 7 phases. The last phase is the result of all the previous phases, in which each one has an important contribution to this sum. The phases are detailed as follows.

- [1] the connection points identification;
- [2] the generative aggregation design of the robotic assembly inside the UR5 robot's limitations;
- [3] the bolts and nuts placement according to the generated discrete aggregation;
- [4] the setting of the assembly order of the discrete elements and subsequently the bolts' placement order;
- [5] the preparation of the workstations set up on the experiment table according to the assembly order;
- [6] the robot simulation involving the HRC procedure and collisions check, and finally;
- [7] the execution of the HRC assembly prototype.

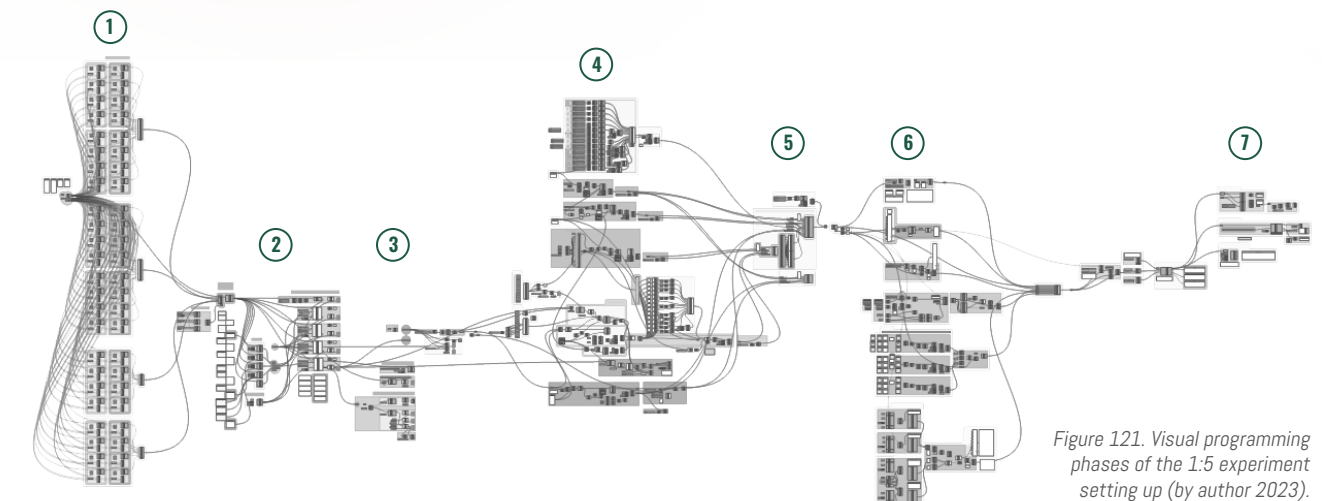
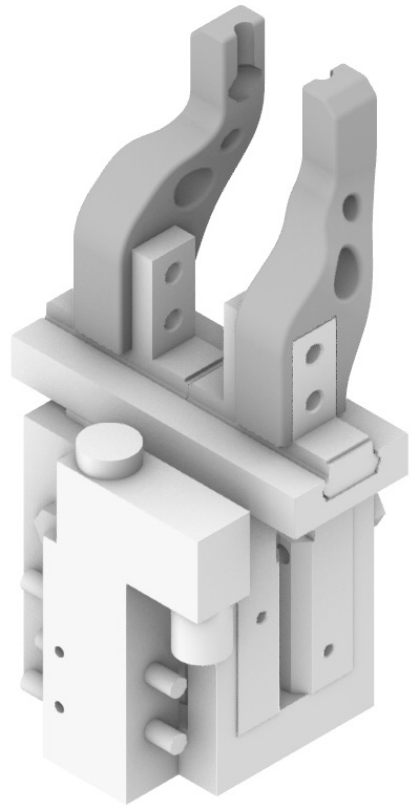


Figure 121. Visual programming phases of the 1:5 experiment setting up (by author 2023).

6.2.3 Robot experiment setup

Inside the experiment workflow logic, as on the construction site, the experimental setup can just be shaped after the discrete elements fabrication and the generated aggregation structure is done. Therefore, the same aggregation rules for the generation of the architectural design were applied to the generation 1:5 prototype structure. One boundary box considering the robot's reachability and the position on the table was inserted in the algorithm for the aggregation of the 16 discrete elements. Many generated results were discarded due to structural stability or repetition of connections. The selection of the structure to be assembled took into consideration the variability of connection and position of the pieces. The selected design attended all the requirements for testing.

Afterward, the bolts were inserted into the assembly structure in a manner each element received 2 bolts in strategic places. To make it happen, the bolt placement procedure filtered the holes on the structure's external surface of each element that would be the robot's approach surface. However, after this point, the bolts were placed manually due to some overlaps that the simplification of the connection design brought to the 1:5 assembly workflow. On the 1:1 scale, this would not happen. Different length of bolts was strategically placed in order to solve these connection overlaps, they are 25mm, 40mm, and 60mm. After that, the assembly order of the bolts was set accordingly to the discrete elements' order. The order of the bolt sizes also had a big influence on the workflow.



The robot's end effector design was customized for this procedure with 2 functionalities: pick the discrete elements and pick the bolt (Fig. 122). Its materialization was made by using PLA 3D printing. Half of the bolt's negative shape was built-in on each side of the end effector. It facilitates the grab of the bolts and contributes to straightening up them as tolerance if they are slightly angled on the station. The end effector design also moved away the tool center point (TCP) from the center axis in order to have at least one side free from colliding with other already assembled pieces.

For this experiment, 3 workstations were set: **1** the discrete element stack, **2** the bolts station, and **3** the assembly structure place (Fig. 124). The stack of generic elements represents the storage pallets on the construction site. The bolts station depicts the customization of the elements by placing bolts in specific holes, 2 per piece in this case. And the 1:5 prototype structure represents the architectural design that is going to be assembled by the robot positioning the pieces in their final position and orientation.

Figure 122. End effector design (by author 2023).



Figure 123. Negative concave bolt's geometry in the robot's end effector (by author 2023).

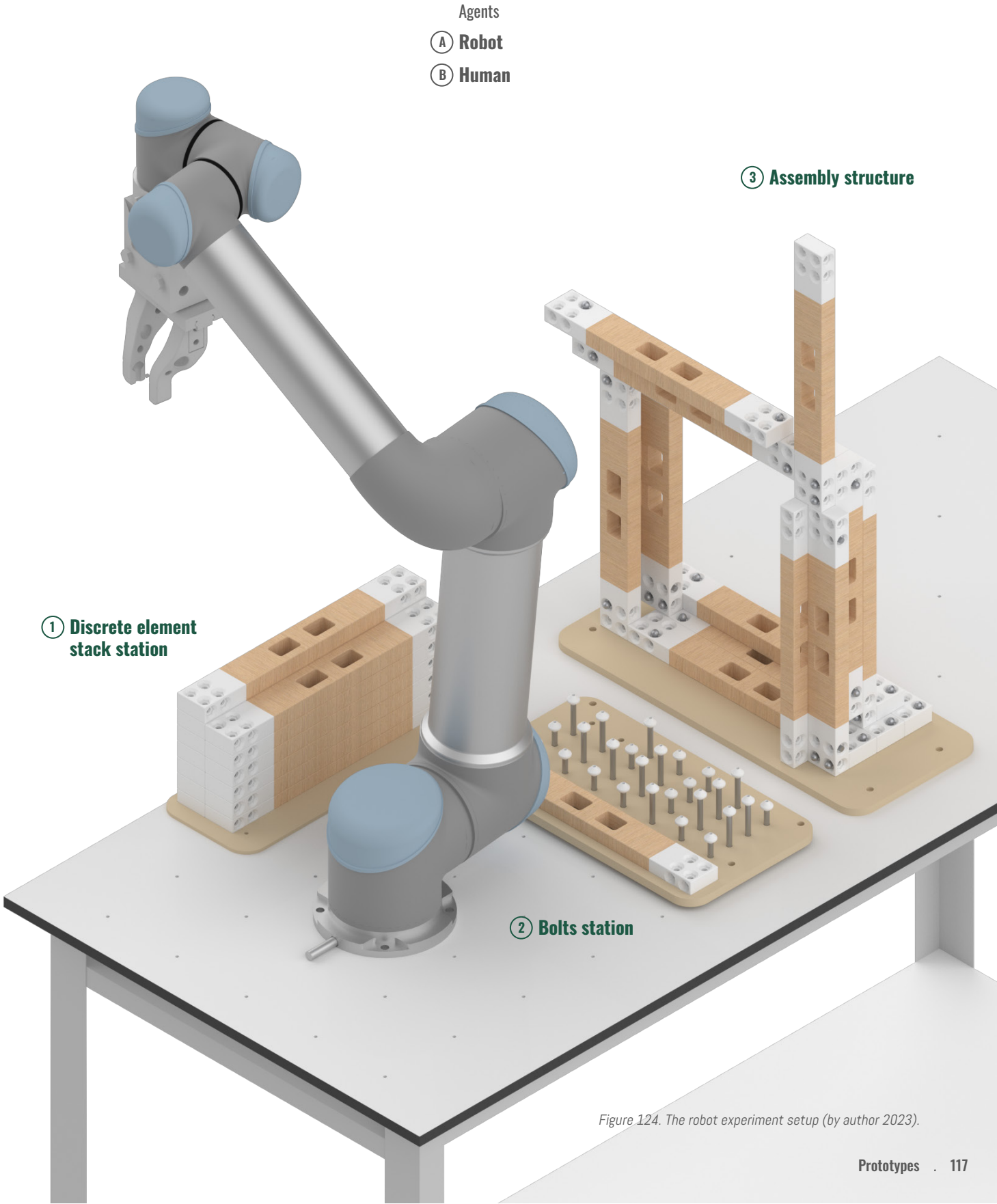


Figure 124. The robot experiment setup (by author 2023).

6.2.4 Robot Simulation and Control

While the mentioned 3 workstations represent different procedures of the assembly, the UR5 robot represents the on-site automated process. The simulation and control of the UR5 robot were developed within **Grasshopper**, utilizing the **Robots plugin**. The decision to choose this plugin was influenced by its capability to create a direct link between the Rhino and the UR5 robot. Unlike other software options, such as RoboDK, the Robots plugin enables seamless data transfer without the need for exporting or transferring data to a different environment (Loo, 2022). This significantly reduces the delay between design and execution, optimizing the computational process. The Robots plugin facilitates the translation of geometric and target information within Rhino into URScript, which is then directly transferred to the UR5

robot through a LAN cable connection. The integration of the design environment with the UR5's control using the Robots plugin allows the geometric features of the discrete elements to be converted into target frames and subsequently translated into robot commands.

The generation of robotic movements is based on two sets of reference target points. Based on Kunic et al., (2021b), for pick-and-place operations, the center points of the discrete element's top face are used as reference points for both picking and placing positions. Additionally, intermediate points are defined to ensure safe operation. These intermediate points include an average of 80 mm moving in the normal direction from the current picking and place target, and transition points created specifically for each element place-

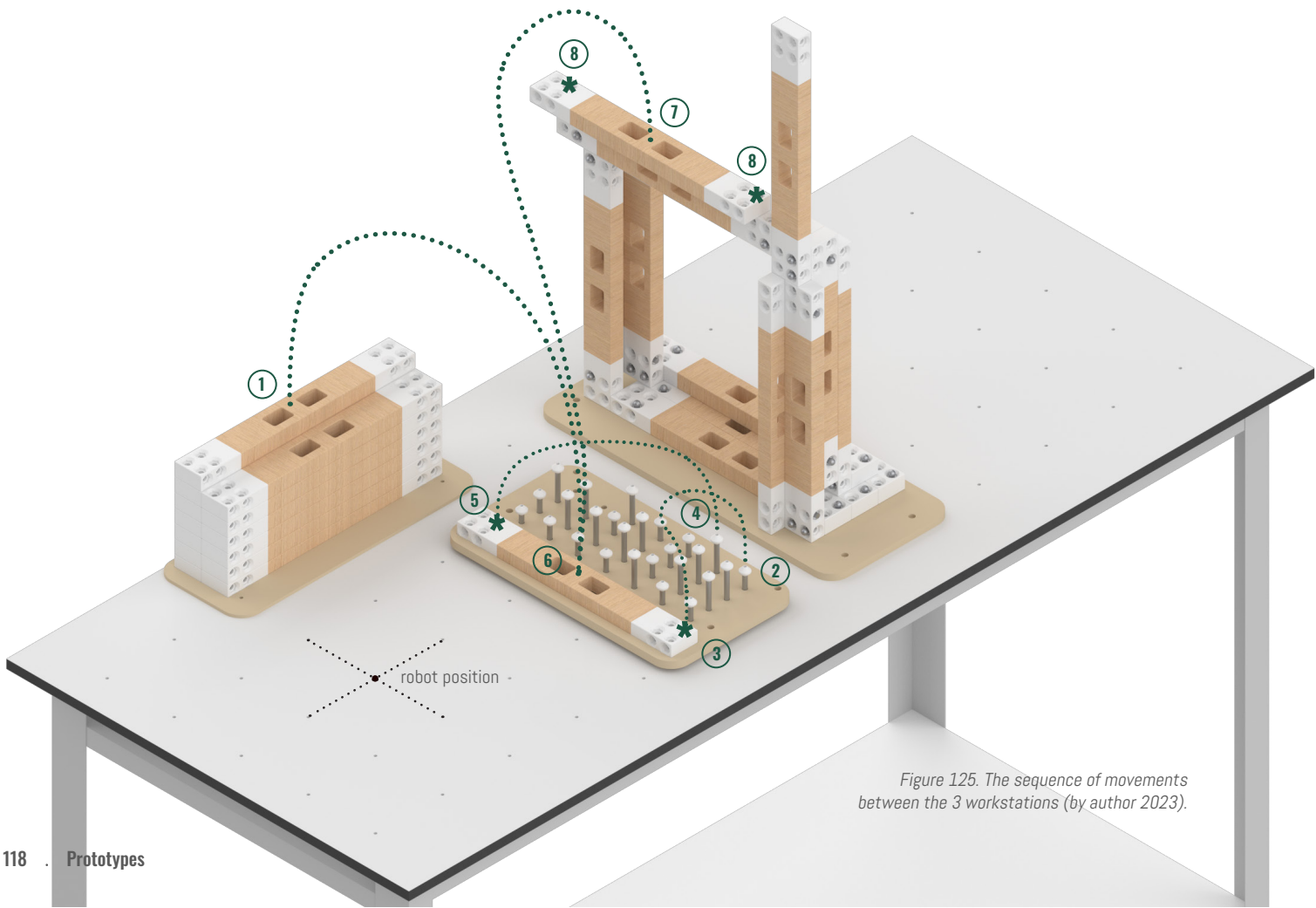


Figure 125. The sequence of movements between the 3 workstations (by author 2023).

ment approach. The main reason for the creation of these transitional points was to avoid internal collisions that the simulation could not identify, which was challenging to solve as each approach needed to be custom. On the construction site, this could be solved with mobile robots being able to move in space to accommodate each approach point each part needs. However, as the robot used in the experiment is fixed, some adaptations were necessary. Regarding the bolting operation, the frames are defined by the center points of the top surface of each hole that requires a bolt.

The HRC is present in the experiment in two moments:

- [1] The first HRC moment is after the placement of the bolts on the element by checking if the bolts are in the correct position, correcting them if not, pre-bolting them, and pressing “ok” for the procedure to continue beyond the bolts station.
- [2] The second HRC moment is after the positioning of the element in its final place while the robot still holding it by checking if the element's position is satisfactory, slightly correcting them if not, bolting them to the structure, and pressing “ok” to continue to the next element.

Both pressing “ok” procedures are fundamental for the robot to keep the assemblies loop. In the real construction site, this procedure would be counted as placed in the twin model.

All the frame points mentioned are referenced in the customized robot's end-effector frame, which means that the positions and orientations of these points are specified relative to the TCP. With the experiment, the whole automated process of each discrete element has 26 frame points. The workflow and each functionality of the 26 frames are illustrated in Figure 126. The sequence of movements between the 3 workstations is illustrated in Figure 125.

The developed UR5 control program code is in **Appendix A**.

- Elements stack station
- Bolts station
- Assembly structure station

Figure 126. The Grasshopper workflow and each functionality of the 26 target points (by author 2023).

- 0 Stack approach arrival point
- 1 Discrete element pick-up point
- 2 Stack approach departure point
- 3 Element preparation approach arrival point
- 4 Element preparation place point
- 5 Element preparation approach departure point
- 6 1st bolt approach arrival point
- 7 1st bolt pick-up point
- 8 1st bolt approach departure point
- 9 1st bolt placement approach arrival point
- 10 1st bolt placement point
- 11 1st bolt placement approach departure point
- 12 2nd bolt approach arrival point
- 13 2nd bolt pick-up point
- 14 2nd bolt approach departure point
- 15 2nd bolt placement approach arrival point
- 16 2nd bolt placement point
- 17 2nd bolt placement approach departure point
- 18 Element preparation approach arrival point
- 19 Element preparation pick-up point
- 20 Element preparation approach departure point
- 21 Customized transition point
- 22 Final placement approach arrival point
- 23 Final placement point
- 24 Final placement approach departure point
- 25 Customized transition point

6.2.5 HRC assembly prototype

The prototype was successful and proved the design concept. Despite the limitations of the UR5 robot and the 1:5 scale that restricted the connections' potential, overall the prototype fulfilled its role of testing the HRC workflow procedure. The division of tasks between humans and robots showed to rely on the collaboration procedure by assembling the pieces as planned. The overall experiment took 45min.

To make it possible, calibration was employed to establish alignment between the simulation model and the real-table workplace. This calibration process was based on the existing reference frames of the table to ensure accurate correspondence between the virtual and real environments (Fig. 130). Each of the 3 workstations had specific frames that matched the measurements in the physical environment. The average deviation between real and digital frames was about 2-5mm.

As the discrete elements were materialized by digital fabrication techniques, the precision among pieces was accurate. However, the deviations between the digital environment

and the real assembly table brought small imprecisions to the experiment while positioning the pieces and the bolts in their holes. Some bolts fell apart due to these deviations and were corrected in the assembly with the HRC procedure. Also, it was observed the presence of deformation from gravity in creating small friction and vibration from the robot.

The assembly process begins with the pick action from the discrete elements stack. Each element is placed in the designated position within the M6 bolts station for its customization preparation, followed by the pick-and-place bolts procedure in their specific holes. After the discrete element is customized, the robot waits for human checking and confirmation to afterward place the element in its final position within the structure. Again the robot waits for the human to check and confirm the position while still holding the piece. After the bolting procedure is performed by the human, the robot starts again the procedure for the next piece. The result of the 1:5 prototype experiment is pictured in Figures 127-136.

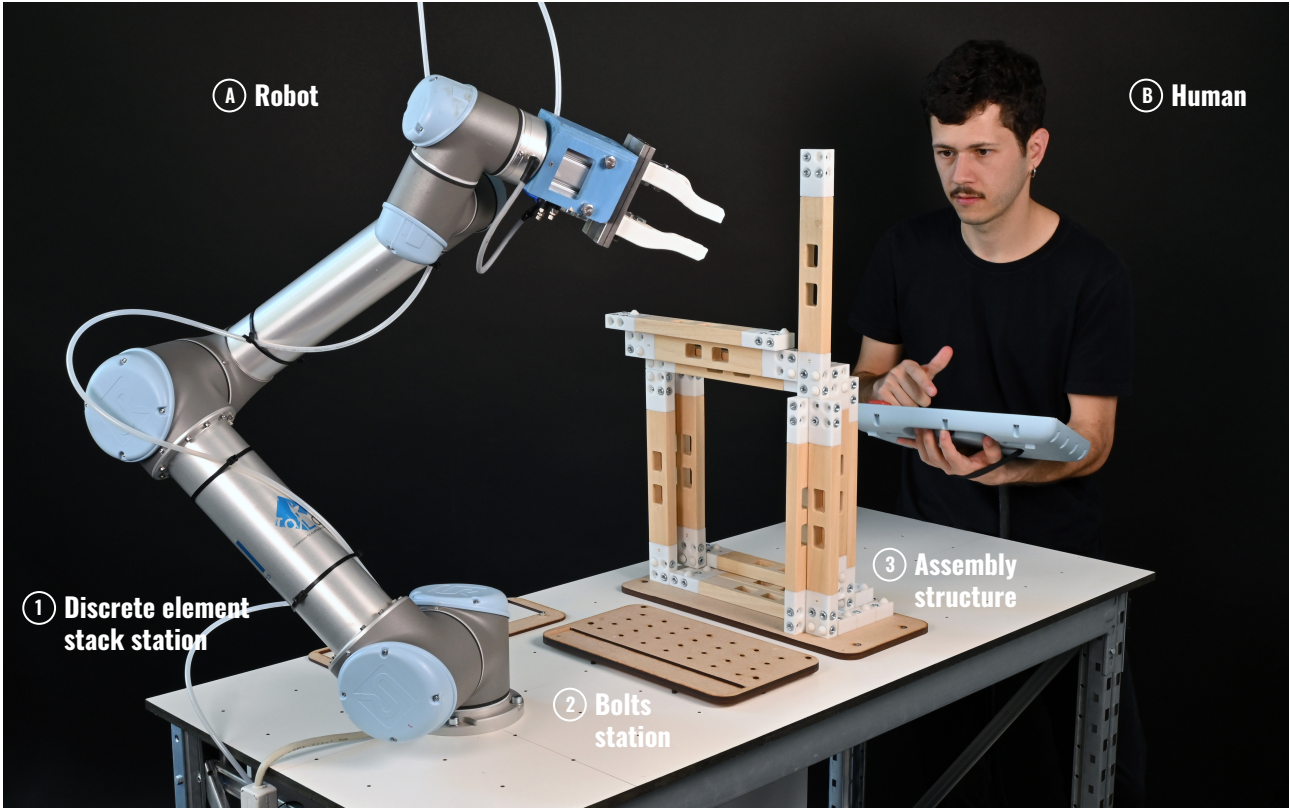


Figure 127. The human and robot agents involved in the experiment and the three workstations (by author 2023).



Figure 128. Prototype assembled by HRC (by author 2023).



Figure 129. Assembly process (by author 2023).

Figure 130. Calibration process (by author 2023).

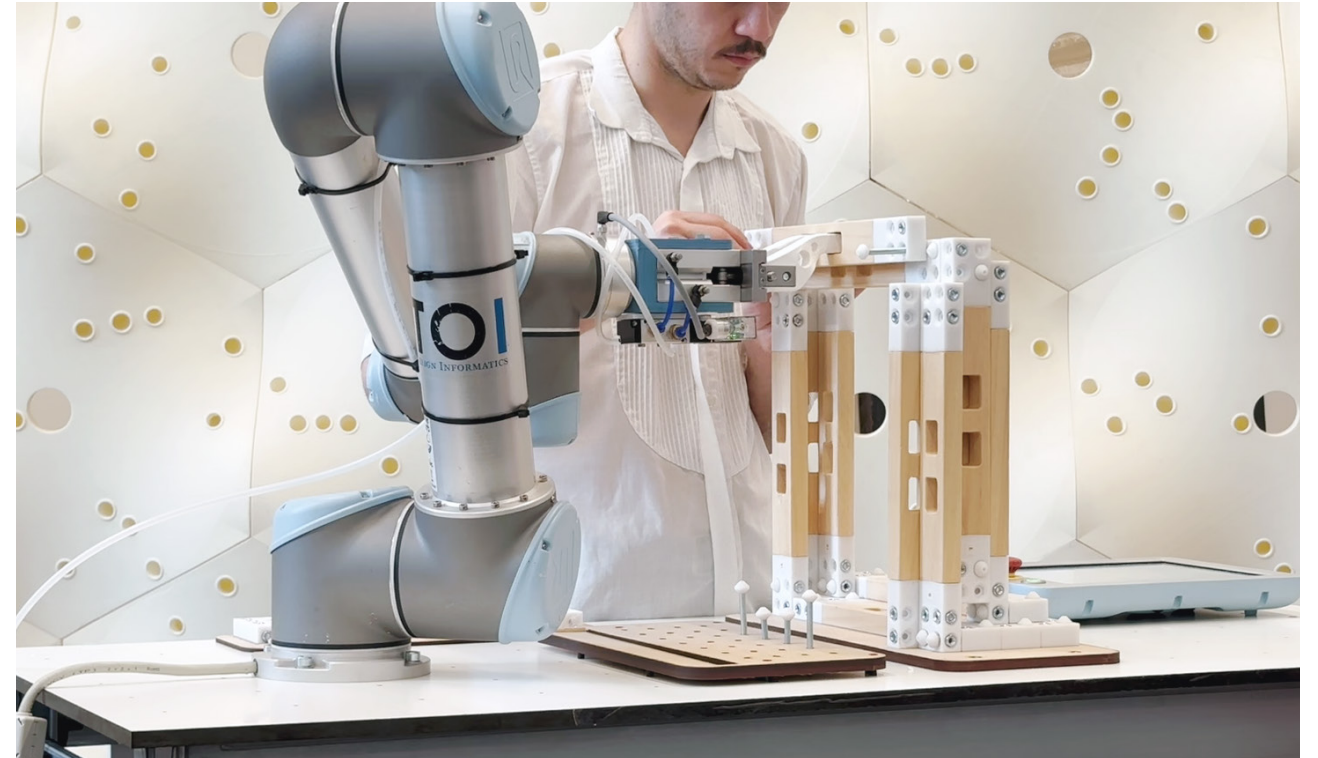
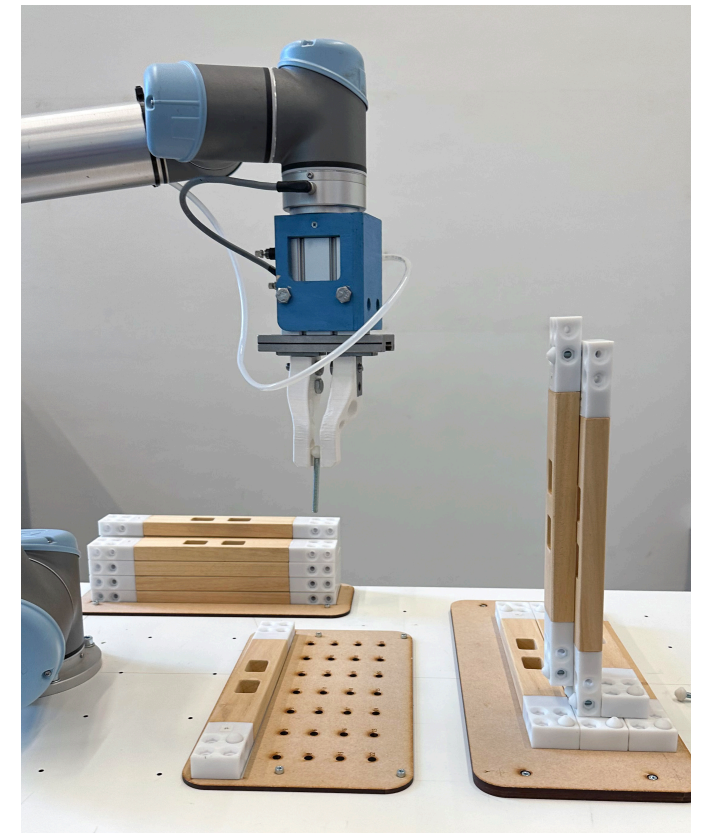


Figure 131. Bolting process while the robot waits for the conclusion of the task (by author 2023).

Figure 132. Robot testing of the bolt pick-up process (by author 2023).



Prototype 1:5
2 x 4 x 32cm



Figure 133. 1:5 prototype overview (by author 2023).



Figure 134. 1:5 prototype connections (by author 2023).



Figure 135. 1:5 prototype connection details (by author 2023).



Figure 136. 1:5 prototype connection details (by author 2023).

6.3 1:10 Prototype

The 1:10 prototype has as its objective testing the assignment of a function to the generic elements and the aesthetic quality of a larger quantity aggregation. The resulting formation is composed of **162** elements of **1x2x16cm** each.

Like in the 1:5 prototype, one connector point represents 4 points of connection in real scale due to the size. A larger tolerance was used to guarantee the alignment of the holes as the pre-drilled holes were made manually. The initial intention was to connect the elements with wood sticks but along the assembly, it showed to be unstable due to the tolerances and lack of fixation. Therefore, as the intention of this prototype was not to check the connectivity, glue was added to the connection to create stability.

During the assembly, the prototype also validated the need for a digital twin model of the whole design with a specific order of assembly. The assembly was time-consuming by the constant need of checking the spatial position of the elements and their connection, which robots can efficiently perform this tasks. Also, it was observed by assembling the pieces per block was beneficial for the whole procedure. The formation of bigger components simplifies the assembly and consequently reduces the spatial movements on the construction site.

Regarding aesthetics, the construction system evokes the organic aspect. The organic is present in the design not as a curvilinear and continuous entity, but in its essence of mathematical rules that enable growth, adaptation, and mutation.



Figure 137. Detail of the connection in prototype 1:10 (by author 2023).



Figure 138. Prototype 1:10 overview (by author, 2023).

Prototype 1:10
1 x 2 x 16cm



Figure 139. The aggregated aesthetic of the 1:10 prototype (by author 2023).



Figure 140. Discretized wood formation of the 1:10 prototype (by author 2023).



Figure 141. Close view of the aggregated formations patterns 1 (by author 2023).



Figure 142. Close view of the aggregated formations patterns 2 (by author 2023).



Figure 143. Close view of the aggregated formations patterns 3 (by author 2023).



Figure 144. Discrete structure formations aesthetics of the prototype 1:10 (by author 2023).

7 Conclusion

- 7.1 Process conclusion
- 7.2 Reflection
- 7.3 Discussion
- 7.4 Further research

This section of the work presents a summary of the findings during the research that led to the conclusion. It also exhibits a reflection on the graduation process and the social impact of the automated construction system. Finally, this section also indicates unsolved areas of the research that are open to further research developments.

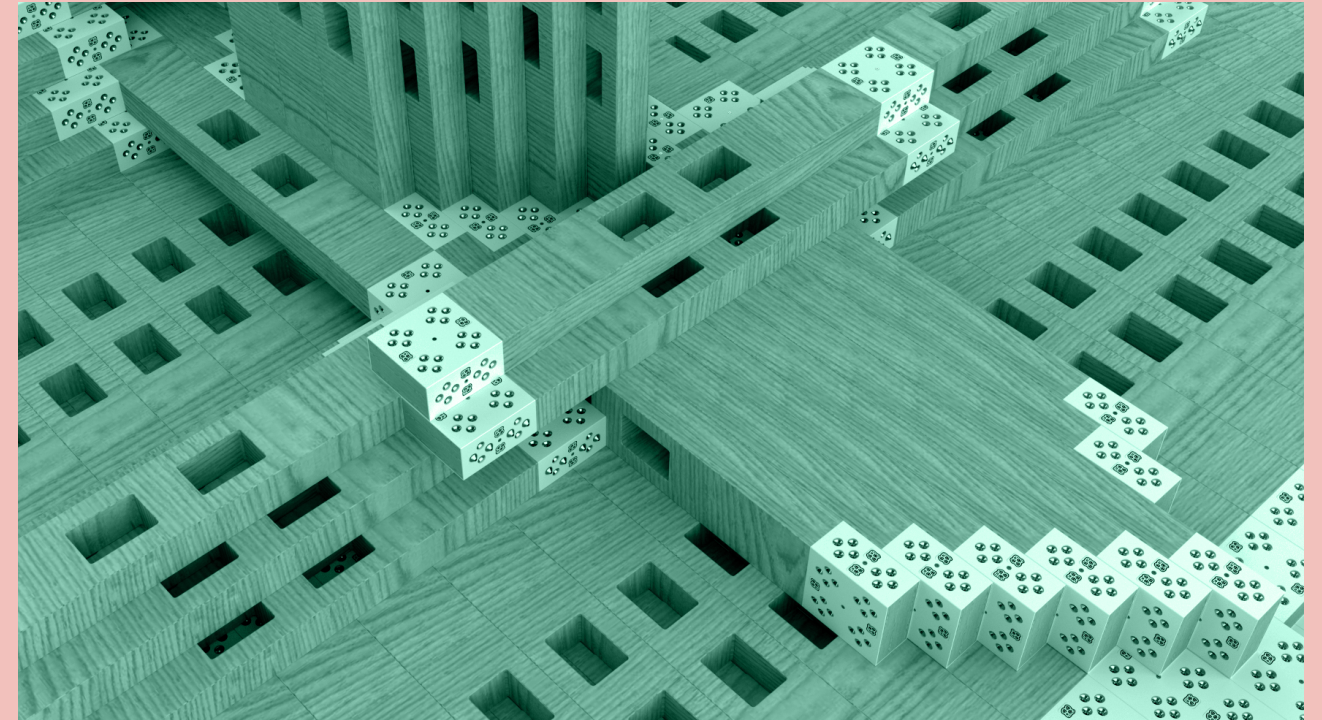


Figure 145. Discrete element aggregation (by author, 2023).

7.1 Process conclusion

The cultural importance of new technologies in architecture is shaped by their usage, flexibility, and societal impact. Technology develops together with its cultural environment, influenced by social, political, and economic factors (Gramazio et al., 2014). This cultural mechanism also applies to robots. Therefore, the robotic construction workflow discussed in the previous chapters for housing design represents the potential for societal impact that digital design and HRC in the assembly of discretized timber structures has while reflecting the principles of the 4th industrial revolution. The housing design embraced circularity through the use of demountable connections, renewable low-carbon materials, and design for reuse and reconfiguration.

This research process implemented a novel approach for designing and manufacturing timber structures using discrete elements, introducing the HRC assembly process and developing physical prototypes at different scales to showcase the concept and validate the results. The 1:1

prototype demonstrated the fabricability of components and their connection. The 1:5 prototype enabled the testing of the HRC assembly, while the 1:10 prototype showcased aesthetic qualities and potential design variations. The assembly approach combined the advantages of digitally fabricating timber elements in a serial production manner and using automation with a UR5 robot working directly with a digital design model. The prototypes served as evidence that customized circular structures can be created by aggregating prefabricated discrete elements. The 1:5 prototype was successfully disassembled and reassembled multiple times by hand. This ability to be repeatedly taken apart and reused extends the lifespan and carbon storage capacity. As a result, it can be concluded that discrete timber elements can be used in assembling circular housing structures and the HRC process can effectively automate the assembly of complex discretized structures.

This thesis proposed the use of just one discrete element for the formation of the architectural design as a simplification of the assembly and manufacturing process. However, other dimensions and shapes of discrete elements can be added to these formations. With different elements, a higher geometrical resolution of spaces can be achieved. By having just one type of element the reconfigurability is based on an easier understanding of the system.

In relation to the combinatorial design, the proposed design approach relies on utilizing a generic identity for each element initially, which is later assigned a specific function when they are aggregated together. In this process, both the individual elements and the overall design hold equal importance, and they continually impact and shape each other during the design process. Therefore, the architectural housing design presented in the work is just one of the potential outcomes that the construction system can create. It has the potential to also create a bench, a small house, a bridge, a sculpture, a pavilion, or be used as an infill for a tower.

The initial intention was to experiment with an HRC workflow prototype that utilizes sensors, computer vision, and the Nordbo Mimic Kit. However, during the process, the design of the discrete elements, their materialization, and the digital aggregation required more problem-solving attention than expected in order to successfully operate the 1:5 assembly prototype. Based on this fact, it can be concluded that in order to achieve a successful automated assembly, the design of the construction system must be fully integrated into the syntax of the robotic workplace. By utilizing

a consistent digital syntax throughout, there is no need for translation into the language of construction. This allows for a smooth transition of materials, modules, and aggregations between the digital and physical domains, ensuring a coherent and efficient process. Thus it is possible to have a workflow that integrates design, fabrication, and assembly.

Nevertheless, discrete thinking was considered as an alternative construction approach that complements automated processes, rather than a universal solution applicable to all building environment scenarios. There are challenges to address in the research, such as the need for further optimization and testing of the design and the robotic construction workflow. The feasibility of the construction process and the disassembly of components also require additional testing and development. The implementation of the method on a construction site will also arise questions that the controlled environment used for testing cannot exhibit. In addition, as the research involved HRC, it is also important to investigate the impact of human error on the whole assembly.

In summary, the design assembly workflow expressed in this thesis aims to inspire and push the boundaries of circular discrete timber structures and affordable housing construction. The discrete model facilitates the fair sharing of resources, promoting inclusiveness and leading to a more equal and democratic process of housing production. The research not only illustrates how discrete components and the HRC approach reveal new tectonic possibilities but also fosters the emergence of a new contemporary digital construction culture.

Answers to the research questions

[main question]

How can an automated assembly process for a discrete and reconfigurable timber construction system contribute to circular housing?

The reconfigurable nature of the system allows for easy disassembly and reassembly of components, promoting adaptive and customizable housing designs that can adapt to changing needs. The automated assembly process accelerates construction timelines, lowers labor costs, and improves affordability, making sustainable housing more accessible. Additionally, it reduces the environmental impact by minimizing emissions and waste generation. The scalability of the process enables efficient mass production, facilitating the widespread adoption of circular housing principles and promoting sustainable practices on a larger scale. The automated assembly process contributes to circular housing by enhancing design flexibility, reducing costs, and minimizing environmental impact.

Q1 *How can discrete architectural design thinking improve the feasibility of an automated assembly process?*

Discrete thinking enhances the feasibility of an automated assembly process by incorporating standardization, modularity, easy connections solutions, and integrated digital design-to-built. Standardized element design ensures compatibility with the automated system, simplifying production and assembly. A modular design approach allows for efficient assembly and reconfiguration, supporting circular housing objectives. Designing easy connections solutions is about improving the connectivity to align with the capabilities of the assembly system, facilitating smooth and accurate operations. Integrated digital design-to-build streamlines communication and minimizes errors between design and assembly.

Q2 *How to implement a human-robot collaboration into a site-specific construction workflow?*

Implementing HRC on a construction site workflow entails identifying suitable tasks for humans and robots, selecting appropriate robots, ensuring safety measures, providing human collaboration training, and prototypes, and integrating communication systems. By carefully planning and coordinating HRC, construction processes can benefit from increased productivity, efficiency, and safety.

Q3 *How can a discrete timber element be designed to attend circularity levels and a human-robot assembly process?*

In order to design a discrete timber element for circularity and HRC assembly, it is needed to choose the typology of wood, employ a modular grid, ensure compatibility with the assembly process, incorporate features for accurate positioning, and enable easy disassembly. It is also important to consider the element's lifecycle impacts, including reusability, recyclability, and compatibility with other materials. By integrating design features that facilitate the robot's manipulation and integration, the timber element can promote reusability, and sustainable construction practices, while accommodating efficient HRC

7.2 Reflection

The following personal reflection is according to the questions in the Graduation Manual of the Master of Science Architecture, Urbanism & Building Sciences, 2022-2023.

Graduation process

[1] *How is your graduation topic positioned in the studio?*

The topic of this research is the combination of robotic construction and discrete timber architecture and relates to two chairs within the Building Technology track, respectively Design Informatics and Structural Design & Mechanics. The relation of the research with the chair Design Informatics is the use of combinatorial design principles to generate the architectural geometry of a discrete timber housing, the formulation of the robotic construction workflow involving human-robot collaboration as well as the simulation and control of the robotic assembly in a prototype. The relation with the chair Structural Design & Mechanics is in the area of sustainable structures with the creation of a discrete timber construction system integrated with robotic construction. The design of the discrete element fuses circularity principles of modularity, disassembly, reuse, reconfigurability, and material life cycle expansion aiming at levels of circularity that can largely contribute to the climate change demands in architecture.

Both robotic construction and discrete architecture relate to the MSc AUBS program. They touch on three current issues that the building environment faces, the low productivity by the missing of automation, the extra architectural design that does not open itself for adaptability, and the need for circularity levels in the building industry to impact positively the environment, society, and economics. Both areas aim to extend knowledge in the field of robotic construction of discrete timber housing. The use of advanced technology in the design process and fabrication aligned with timber as a renewable material aim to make the built environment more circular.

[2] *How did the research approach work out? And did it lead to the results you aimed for?*

During the thesis project development, extensive investigations have been carried out in three key domains to have a seamless workflow: combinatorial design, digital fabrication, and HRC assembly. These inquiries have served as a foundation for developing a holistic design that aligns with the site's boundary conditions while also making noteworthy contributions to sustainable timber structures within the building industry.

The research for the design process of this study took a nonlinear path, with all aspects of the research unfolding simultaneously. This approach resulted in constant loops of adjustments in the architectural design concept, materiality, computational design strategies, and HRC workflow. The initial intention was to experiment with an HRC workflow prototype that utilizes sensors, computer vision, and the Nordbo Mimic Kit. However, during the process, the design of the discrete elements, their materialization, and the digital aggregation required more problem-solving attention than expected in order to successfully operate the 1:5 assembly prototype. Based on this fact, it was concluded that in order to achieve a successful automated assembly, the design of the construction system must be fully integrated into the syntax of the robotic workplace. However, the research led to the intended results, although with a less complex HRC workflow prototype. This outcome confirmed the success of the design concept.

[3] *How are research and design related?*

The research focused on two main areas: the design of a housing platform using discrete timber elements to shape its structure, and the implementation of HRC construction workflow. The literature review on these topics, along with the specific site, guided the exploration of the discretized timber housing design. The HRC robotic assembly influenced design factors such as the size of the discrete elements and their connections. The interconnection between the two main fields provided valuable insights into current trends in timber structures and HRC technology which established the foundation of knowledge for this work.

Societal impact

[1] *To what extent are the results applicable in practice?*

Further development needs to be carried out for the whole proposed workflow to be applicable in practice. However, some aspects of the whole project are already feasible. The timber manufacturing industry is well-equipped to provide the necessary machinery support for the production of the discrete elements design. The design-to-build workflow proposed can be implemented using existing machinery and robotic systems, but there are still challenges to solve in calibration, human-robot collaboration implementation, and real-time feedback for robotic systems awareness. The integration of HRC technology into the construction industry is still in the developmental stage, requiring further research and refinement before widespread adoption. Challenges may arise from the structure's dynamic exerted by the connection of a large number of discrete elements' aggregations. However, optimization algorithms have the potential to streamline and simplify the intricate nature of this structure. The next section presents some additional research areas that emerge from the necessary further investigation, leading to new research paths.

[2] *To what extent has the projected innovation been achieved?*

The designed innovation was achieved by the implementation of the 1:5 HRC workflow prototype. The experiment necessitated the creation of a robot program that relies on human input responses to maintain the procedure. The prototype worked as a proof-of-concept of its HRC capability to be assembled, disassembled, and accommodate future changes effectively.

[3] *Does the project contribute to sustainable development? What is the impact of your project on sustainability? How does the project affect architecture and the built environment?*

The combination of an automated assembly process and a discretized construction system has a significant impact on housing. The system's reconfigurable nature allows for easy disassembly and reassembly of components, enabling adaptive and customizable housing designs that can meet changing needs. By employing an automated assembly process, construction timelines are accelerated, labor costs are reduced, and overall affordability is improved, making sustainable housing more accessible to a wider population. Moreover, this process plays a crucial role in reducing the environmental footprint of housing construction by minimizing emissions and waste generation.

The scalability of the automated assembly process further enhances its benefits. It enables efficient mass production, facilitating the widespread adoption of circular housing principles and promoting sustainable practices on a larger scale. This process not only contributes to circular housing by enhancing design flexibility but also by reducing costs and minimizing environmental impact. The integration of robotics and automation in the construction industry paves the way for a more sustainable future, where housing can be efficiently produced, adapted, and reused, in line with the principles of a circular economy.

By combining the principles of discrete thinking, which emphasizes standardized parts and combinatory permutations, with the advantages of automated assembly, the construction industry can address the increasing global housing demand while minimizing resource consumption. This approach promotes participatory frameworks for collective production, fosters a sharing economy, and challenges traditional modes of ownership. Furthermore, the proposed automation of the assembly process not only brings economic benefits but also opens doors to innovative housing solutions, benefiting both individuals and communities.

7.3 Discussion

The construction system proposed in this thesis reflects a new role that architects and engineers have in designing not deciding how the building will be but instead designing the rules and boundaries to shape it. It is about designing open-ended systems and not final objects. While designing these systems, they are open to adaptation to any scenario and condition needed. This system brings to the construction industry a new method of thinking and design.

However, it is important to mention that the proposed discrete design needs more interaction of design and further research development for its implementation in the construction industry. The system has many pros when related to adaptability, disassembly, mass customization, and design for future reuse, but its cons also should be mentioned. Each discrete element has 96 nuts pre-installed in order to achieve genericness and large reconfigurability possibilities. In which not all these nuts will be used in the assembly. So a balance between material optimization and genericness needs to be further researched. Also, the discrete elements in the way that is designed have many manufacturing processes embedded in their materialization resulting in a significant amount of energy consumption. A simplification of the connections may be beneficial to the overall method in order to foster the adoption of the market by having faster connections with fewer bolts and nuts.

The construction system opens room for discussion of a new way of creating spaces and reconfiguring them while life changes happen at the same time that offers potential for new models of ownership. The complexity of the construction system is embedded in the aggregation of the simplicity of the discrete elements. However, the system uses more materials than a traditional construction method that had a specific design for a specific function. The costs of the prefabrication of future reuse may also be more expensive than a consolidated construction method. However, a balance between these aspects needs to be investigated, and the potential of scalability is one of them to make the process cheaper.

Regarding aesthetics, the construction system evokes the organic aspect. The organic is present in the design not as a curvilinear and continuous entity, but in its essence of mathematical rules that enable growth, adaptation, and mutation. Although it is a largely industrialized design, the overall spatial quality resulted connects to the aesthetical organic approach of an open fractal structure found in crystals and basalt formations (Fig. 146). It is a novel aesthetic proposition and interpretation of what can be a biophilic approach to architecture. As a metaphor, the building could be understood as atoms that shape a bigger whole that is more than just their agglomeration. It is about a way of inhabiting and building to embrace the constant mutation.



Figure 146. Basalt Columns in Iceland (CarSiceland, n.d.).

7.4 Further research

The research opens numerous potential paths for further development. There are significant yet unexplored areas that remain crucial for implementing the project workflow at a construction site. The subsequent challenges listed herein are yet to be addressed or overcome.

- [1] Since the disassembly procedure was completed by hand, the next step would be to understand what are the arising challenges that the automated disassembly procedure will have. The inversion of the assembly movements sequence could be the starting point, but structural sagging and friction could difficult the procedure.
- [2] Topology optimization of the boundary box used to create the building aggregation. A definition that follows the climate analysis output of the surrounding could be applied to generate an efficient and site-related boundary box geometry.
- [3] Research on creating a set of aggregation rules that thoroughly incorporate the structural logic during the initial combinatorial stage, thereby enhancing efficiency.
- [4] Structural calculation and optimization in order to generate efficient aggregations while following the structural loads' demand.
- [5] Connection testing to understand its mechanical properties, as well as the bolts' shear force resistance.
- [6] Testing the self-alignment feature of the bolts within an automated 1:1 prototype with the element fully materialized.
- [7] Additional tolerance studies may be required to enhance the feasibility of on-site robotic assembly at a 1:1 scale, which could not be adequately demonstrated by the 1:5 prototype.
- [8] Development and testing of the HRC on a real scale on-site with sensors, computer vision, and the Nordbo Mimic Kit.

- [9] Depth research on feedback systems that utilize robot vision to enhance the reliability of the proposed workflow. Systems that are responsible for object detection, compensating for tolerances, and providing real-time validation of the assembly process.
- [10] Research and analysis to explore the possibilities and practicality of establishing a centralized digital twin model that integrates with the envisioned housing platform.
- [11] Research and implementation of real-time structural analysis. This analysis would take into consideration the static aspects of the system to ensure that decisions made on the spot are structurally sound.
- [12] The research solely focused on 6-axis cobot arms for the proposed HRC assembly and testing. Nonetheless, it would be beneficial to investigate alternative assembly methods involving different types of robots.
- [13] Research that focuses on exploring the potential development of a customized robot that can effectively address the specific requirements of the process. This investigation aims to identify how a specialized robot can be tailored to fulfill the unique needs of the process, potentially leading to improved efficiency and effectiveness.
- [14] Research on how insulation could be addressed in the discrete elements and their aggregations.
- [15] Investigation of the financial impact. Conducting a cost analysis is crucial in order to assess the feasibility of the proposed system.
- [16] Investigation of the implementation of connections at 45 degrees. It is already possible but just at one of the ends of the discrete element. Different length sizes would be beneficial for the connectivity at 45 degrees to have both ends connected.

References

AllCharts. (n.d.). *Municipality Rotterdam*. Retrieved from <https://allcharts.info/the-netherlands/municipality-rotterdam/>

Autodesk. (n.d.). *Generative design*. Retrieved May 14, 2023, from <https://www.autodesk.com/solutions/generative-design>

Azadi, S., & Nourian, P. (2021). GoDesign: A modular generative design framework for mass-customization and optimization in architectural design. *39th eCAADe, volume 1*, p.285-294. DOI:10.52842/conf.ecaade.2021.1.285

Bencina, R., & Kaltenbrunner, M. (2005). *The Design and Evolution of Fiducials for the reacTIVision System*. Semantic Scholar. Retrieved from <https://www.semanticscholar.org/paper/The-Design-and-Evolution-of-Fiducials-for-the-Bencina-Kaltenbrunner/a2ddcdf15dda1f384e67d73942f6d859c68a5ed0>

Bouza, H. (2019). *Reed Robotics: Discrete Digital Assembly of Biodegradable Reed Structures* [Master's thesis, Delft University of Technology]. <http://resolver.tudelft.nl/uuid:6af99d6e-7f5b-4cad-a773-5df9163b7f9e>

Bouza, H., & Asut, S. (2020). *Advancing Reed-Based Architecture through Circular Digital Fabrication*. In Proceedings of the 38th eCAADe: Anthropologic - Architecture and Fabrication in the cognitive age (Vol. 1, pp. 117-126) http://papers.cumincad.org/cgi-bin/works/paper/ecaade2020_240

Burden, A. G., Caldwell, G. A., & Guertler, M. R. (2022). Towards human-robot collaboration in construction: current cobot trends and forecasts. *Construction Robotics*. <https://doi.org/10.1007/s41693-022-00085-0>

Celani, G., Cypriano, D., Godoi, G., & Vaz, C. (2006). A gramática da forma como metodologia de análise e síntese em arquitetura. *Conexão: Comunicação e Cultura, UCS, Caxias do Sul, v. 5, n. 10*. p. 180-197.

Chen, D., Wang, G., & Chen, G. (2021). Lego architecture: Research on a temporary building design method for post-disaster emergency. *Frontiers of Architectural Research, 10*. p. 758-770. ISSN 2095-2635. 10.1016/j.foar.2021.08.001.

Cheng, C. Y. S., Fung, G. K. C., Lee, G. Y. H., Man, J. K. T., Siu, W. J. J., Sohn, H., Tam, Y. H. H., Tang, C. Y., Tsang, H. L. J. M., Tse, W. C. K., Wong, T. C. C., Yeung, H. O., & Zhu, C. F. (2020). *PARTI. Millennial architecture: Tools, strategies and prototypes for post-boom life. Research + platform proposal*. The University of Hong Kong, Department of Architecture.

Claypool, M. (2019). Our Automated Future: A Discrete Framework for the Production of Housing. *Architectural Design, 89, Issue 2*, p. 46-53. <https://doi.org/10.1002/ad.2411>

Council of Europe. (n.d.). *Rotterdam, Netherlands - Intercultural City*. Retrieved from <https://www.coe.int/en/web/interculturalcities/rotterdam>

García, M.J. (2019). Discrete Flexibility: Computing Lightness in Architecture. *Architectural Design, 89, Issue 2*, p. 70-77. <https://doi.org/10.1002/ad.2414>

Gershenfeld, N., Carney, M., Jenett, B., Calisch, S., & Wilson, S. (2015). Macrofabrication with Digital Materials: Robotic Assembly. *Architectural Design, 85*, p. 122-127. DOI:10.1002/ad.1964.

Gilles Retsin Architecture. (2017). *Tallinn Architecture Biennale Pavilion*. <https://www.retsin.org/Tallinn-Architecture-Biennale-Pavilion>

Government of the Netherlands. (2016). *Circular economy*. <https://www.government.nl/topics/circular-economy>

Gramazio, F, Kohler, M., & Willmann, J. (2014). *The Robotic Touch: How Robots Change Architecture*. Park Books.

Hansen, S. G., Kunic, A., & Naboni, R. (2021). A reversible connection for robotic assembly of timber structures. *Engineering Structures, 245*, 112795. <https://doi.org/10.1016/j.engstruct.2021.112795>

Helm, V., Knauss, M., Kohlhammer, T., Gramazio, F., & Kohler, M. (2016). Additive robotic fabrication of complex timber structures. In Menges, A., Schwinn, T., & Krieg, O. D. (Eds.), *Advancing Wood Architecture: a Computational Approach*. (pp.29-43). Taylor & Francis Group.

Kaiser, B., Strobel, T., & Verl, A. (2021). Human-Robot Collaborative Workflows for Reconfigurable Fabrication Systems in Timber Prefabrication using Augmented Reality. *27th International Conference on Mechatronics and Machine Vision in Practice*. DOI:10.1109/M2VIP49856.2021.9665011

Kengo Kuma & Associates. (2017). *Coeda House*. <https://kkaa.co.jp/en/project/coeda-house/>

Kengo Kuma & Associates. (2018). *Kodama*. <https://kkaa.co.jp/en/project/kodama/>

Koerner-Al-Rawi, J., Park, K.E., Phillips, T.K. et al. (2020). Robotic timber assembly. *Construction Robotics, 4*, p.175-185 (2020). <https://doi.org/10.1007/s41693-020-00045-6>

Kramberger, A., Kunic, A., Iturrate, I., Sloth, C., Naboni, R., & Schlette, C. (2022). Robotic Assembly of Timber Structures in a Human-Robot Collaboration Setup. *Frontiers in Robotics and AI, volume 8*, 768038. <https://doi.org/10.3389/frobt.2021.768038>

Kunic, A., Kramberger, A., & Naboni, R. (2021a). Cyber-Physical Robotic Process for Re-Configurable Wood Architecture. *39th eCAADe, Towards a New, Configurable Architecture, Volume 2*. 181-188

Kunic, A., Naboni, R., Kramberger, A., & Schlette, C. (2021b). Design and assembly automation of the Robotic Reversible Timber Beam. *Automation in Construction, 123*, 103531. <https://doi.org/10.1016/j.autcon.2020.103531>

Loo, J. W. (2022). *Sustainable Timber Bridge Design with Design-to-Build Workflow Involving Human-Robot Collaboration (HRC)* [Master's thesis, Delft University of Technology]. <http://resolver.tudelft.nl/uuid:4da6966d-fae9-4391-b761-afafb7e4d546>

Man, N. X. (2018). *Assembler Assemble*. https://issuu.com/x.man/docs/final_m4g_potfolio

Man, N. X. (2021). Automating the design and assembly process of timber block construction system. *AIP Conf. Proc. 2428*, 050004-1–050004-15. <https://doi.org/10.1063/5.0071200>

Menges, A. (2016). Integrative design computation for advancing wood architecture. In Menges, A., Schwinn, T., & Krieg, O. D. (Eds.), *Advancing Wood Architecture: a Computational Approach*. (pp.97-109). Taylor & Francis Group.

Menges, A. (2015). The New Cyber-Physical Making in Architecture: Computational Construction. *Architectural Design, 85*, p. 28-33. DOI:10.1002/ad.1950.

Menges, A., Schwinn, T., & Krieg, O. D. (2016). *Advancing Wood Architecture: a Computational Approach*. Taylor & Francis Group.

Mitterberger, D., Atanasova, L., Dörfler, K., Gramazio, F., & Kohler, M. (2022). Tie a knot: human-robot cooperative workflow for assembling wooden structures using rope joints. *Construction Robotics*. <https://doi.org/10.1007/s41693-022-00083-2>

Nourian, P. (2020). *Rudiments of Geometry and Topology for Computational Design*. 10.13140/RG.2.2.10497.68966.

Nourian, P., & Azadi, S. (2021). *Customizable Quality Housing for the Masses Pixel Planet*. DOI:10.13140/RG.2.2.11226.26565

Picon, A. (2020). Architecture and Materiality in the Digital Era. In A. Picon (Ed.), *The Materiality of Architecture*. (pp. 105-136). University of Minnesota Press.

Retsin, G. (2019a). Discrete Architecture in the Age of Automation. *Architectural Design, 89, Issue 2*, p. 6-13. <https://doi.org/10.1002/ad.2406>

Retsin, G. (2019b). In Part Whole: The Aesthetics of the Discrete. *Architectural Design, 89, Issue 5*, p. 120-127. <https://doi.org/10.1002/ad.2488>

Retsin, G. (2019c). Bits and Pieces - Digital Assemblies: From Craft to Automation. *Architectural Design, 89, Issue 2*, p. 38-45. <https://doi.org/10.1002/ad.2410>

Retsin, G. (2019d). Toward Discrete Architecture: Automation takes Command. In: Bieg, K., Briscoe, D., & Odom, C. (Eds.). *39th Annual Conference of the Association for Computer-Aided Design in Architecture*. p. 532-541.

Retsin, G., Burry, J., Sabin, J., Sheil, B., & Skavara, M. (2020). Discrete Timber Assembly. In *Fabricate 2020: Making Resilient Architecture* (pp. 264–271). UCL Press. <https://doi.org/10.2307/j.ctv13xpsvw.38>

Rogeaua, N., Tiberghiena, V., Latteurb, P., & Weinanda, Y. (2020). Robotic Insertion of Timber Joints using Visual Detection of Fiducial Markers. *37th International Symposium on Automation and Robotics in Construction*. pp. 491-498. DOI:10.22260/ISARC2020/0068

Rossi, A., & Tessmann, O. (2019). From voxels to parts: Hierarchical discrete modeling for design and assembly. *18th International Conference on Geometry and Graphics, Advances in Intelligent Systems and Computing 809*, pp. 1001-1012. https://doi.org/10.1007/978-3-319-95588-9_86

Sanchez, J. (2017). Combinatorial Commons: Social Remixing in a Sharing Economy. *Architectural Design, 87, Issue 4*, p.16-21. <https://doi.org/10.1002/ad.2190>

Sanchez, J. (2019). Architecture for the Commons: Participatory Systems in the Age of Platforms. *Architectural Design, 89, Issue 2*, p. 22-29. <https://doi.org/10.1002/ad.2408>

Sanchez, J. (2020). In Defense of Parts. In J. Sanchez (Ed.), *Architecture for the Commons: Participatory Systems in the Age of Platforms*. pp. 57-85. Routledge. 104324/9780429432118

Sandoval, L. (2023). *Una Reflexión Poslaborista Para el Futuro del Trabajo: Trabajar menos, automatizar todo, producir lo necesario, redistribuir todo*. Brecha. Retrieved from <https://brecha.com.uy/trabajar-menos-automatizar-to-do-producir-lo-necesario-redistribuir-todo/>

Schwinn, T. (2016). Landesgartenschau Exhibition Hall. In Menges, A., Schwinn, T., & Krieg, O. D. (Eds.), *Advancing Wood Architecture: a Computational Approach*. (pp. 111-124). Taylor & Francis Group.

Stappers, P. J., & Giaccardi, E. (2018). *Research through Design*. In M. Soegaard & R. F. Dam (Eds.), The Encyclopedia of Human-Computer Interaction, 2nd Ed. Aarhus, Denmark: The Interaction Design Foundation. Retrieved from <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/research-through-design>

Stumm, S., Devadass, P., & Brell-Cokcan, S. (2018). Haptic programming in construction: Intuitive on-site robotics. *Construction Robotics, 2*, 3-13. <https://doi.org/10.1007/s41693-018-0015-9>

Tessmann, O., & Rossi, A. (2019). Geometry as Interface: Parametric and Combinatorial Topological Interlocking Assemblies. ASME. *Journal of Applied Mechanics, 86*, 111-002. <https://doi.org/10.1115/14044606>

United Nations Environment Programme. (2019). *UNEP circularity platform*. <https://www.unep.org/circularity>

Wagner, H. J., Alvarez, M., Kyjanek, O., Bhiri, Z., Buck M., & Menges, A. (2020). Flexible and transportable robotic timber construction platform - TIM. *Automation in Construction, Volume 120*, 103400. <https://doi.org/10.1016/j.autcon.2020.103400>.

Willmann, J., Gramazio, F., & Kohler, M. (2016). New paradigms of the automatic: Robotic timber construction in architecture. In Menges, A., Schwinn, T., & Krieg, O. D. (Eds.), *Advancing Wood Architecture: a Computational Approach*. (pp.13-27). Taylor & Francis Group.

Wonen in Rotterdam. (n.d.). *Kop van Zuid-Entrepot*. Retrieved from <https://www.woneninrotterdam.nl/en/feijenoord/kop-van-zuid-entrepot/>

Wysocki, J. (2021). *Customized Collective Housing: Towards Participatory Design of Dwellings* [Master's thesis, Delft University of Technology]. <http://resolver.tudelft.nl/uuid:e5d3c10c-3199-4b4e-95a2-5accf10e904e>

Xiao, K., Chen, C., Guo, Z., Wang, X., & Yan, C. (2020). Research on voxel-based aggregation design and its fabrication. *25th International Conference on Computer-Aided Architectural Design Research in Asia*, volume 1, p. 13-22.

List of Figures

Figure 1. Cross-laminated timber plates robotically assembled with a novel assembly procedure. (Gramazio Kohler Research, ETH Zurich, 2022).

Figure 2. Collaborative assembly by a team of humans and robots. (Mitterberger, 2022).

Figure 3. Chart of productivity growth - Output per worker. (UK Office for National Statistics, 2017).

Figure 4. Problems in converting a solution for industrial production to the construction site. Adapted from Stumm et al. (2018).

Figure 5. Architectural Evolution. Adapted from Gilles Retsin Lecture (2020).

Figure 6. Zaha Hadid's Heydar Aliyev Cultural Centre construction. (Zaha Hadid Architects, 2011).

Figure 7. Annual Global CO2 Emissions. (Adapted from Architecture 2030). (Data Source: IEA (2022).

Figure 8. United Nations circularity approach. Adapted from United Nations Environment Programme (2019).

Figure 9. Common syntax diagram of the three areas of the research. (by author, 2023).

Figure 10. Research workflow. (by author, 2023).

Figure 11. Methodology and Timeline of the Research. (by author, 2023).

Figure 12. Flexible and transportable robotic timber construction platform, TIM. (Wagner et al., 2020).

Figure 13. Future Tree roof fabrication robotic assembly. (ERNE AG Holzbau). (Gramazio Kohler Research, ETH Zurich, 2019).

Figure 14. Stratifications. Adaptive robotic fabrication in which the design is generated from the robot's operation using a feedback loop to inform its assembly. (Gramazio Kohler Research, ETH Zurich, 2011).

Figure 15. Types of Human-Robot Relationship. Adapted from Burden et al. (2022).

Figure 16. A cooperative proof-of-concept prototype has been developed to evaluate the design principles and workflow of an assembly cycle comprising five key components: interactive design, robotic assembly, manual assembly, rope jointing, and element tracking (Mitterberger, 2022).

Figure 17. HRC sequence of tasks consisting of pick-and-place of the timber construction elements, human switching the robot's screwing configuration to collaborative mode, and robotic screwing procedure. (Kunic et al. 2021b).

Figure 18. Flexible and transportable robotic platform in transport position with robots and control cabinets being stored on the platform with removed cover. (Wagner et al., 2020).

Figure 19. In Situ Robotic Fabrication in which the robot recognizes its own position, the surroundings, and its components with regard to the material tolerances. (Gramazio Kohler Research, ETH Zurich, 2012).

Figure 20. The Endless Wall with cognitive characteristics in which the robotic system has to respond autonomously to tolerances and adapt to changing conditions. (Gramazio Kohler Research, ETH Zurich, 2011).

Figure 21. Tasks groups in relation to complexity and safety. Adapted from Loo (2022, p. 98)

Figure 22. Discrete design of a modular housing. (Minfeng Xia, Discrete Economies, 2017)

Figure 23. Discrete elements as a reassembly of Domino House. (Ivo Tedbury, Semblr, 2017)

Figure 24. Discrete architectural design for a multi-family house. (Gilles Retsin Architecture, Diamonds, 2016).

Figure 25. Discrete aggregation study involving serial repetition, pattern languages and combinatorial tectonics. (Minfeng Xia, Discrete Economies, 2017)

Figure 26. Temporary housing prototype consisting of a discrete framework for housing production. House Block. (Automated Architecture (AUAR) Labs, UCL, photo by James Harris, 2021)

Figure 27. Robotic reversible timber beam. (CREATE group, SDU, 2021)

Figure 28. BetaPort. Scalable on-demand building system materialized by a fully automated planning process and ecologic construction methodology. (Urban Beta, photo by Naaro, 2020)

Figure 29. Overview of timber's strengths. Adapted from Loo (2022, p. 14)

Figure 30. The Tallinn Architecture Biennale pavilion. (Gilles Retsin Architecture, 2017).

Figure 31. Assembler Assemble project. (Man, 2021)

Figure 32. Coeda House (Kengo Kuma & Associates, photo by Kobayashi Kenji Photograph Office, 2017)

Figure 33. Kodama pavilion. (Kengo Kuma & Associates, 2018).

Figure 34. Discrete element typology analysis. (by Author, 2023).

Figure 35. Discrete element aggregation shaping the housing structure (by author, 2023).

Figure 36. Initial explorations of orthogonal beam type of discrete element (by author, 2023).

Figure 37. The railway sleeper (Mitchell Turf, 2023).

Figure 38. Discrete element design workflow (by author, 2023).

Figure 39. Comparison between volume sizes (by author, 2023).

Figure 40. Relationship between the end effector and the discrete element volume (by author, 2023).

Figure 41. The conceptual schematization of three types of connections and their load response diagram. Self-alignment features in the connection (Hansen et al., 2021).

Figure 42. Design of the bolt and nut with self-alignment feature (by author, 2023).

Figure 43. Connection detail of the bolt and nut (by author, 2023).

Figure 44. Connection detail of the bolt crossing the discrete element section (by author, 2023).

Figure 45. Fiducial codes (Bencina & Kaltenbrunner, 2005).

Figure 46. Section detail of the connection (by author, 2023).

Figure 47. Step by step of the connection assembly (by author, 2023).

Figure 48. Discrete element design (by author, 2023).

Figure 49. Connection detail of element's vertical and horizontal surfaces (by author, 2023).

Figure 50. Identification of each point of connection for the generation of the aggregation rules by face (by author, 2023).

Figure 51. 84 aggregation rules possibilities (by author, 2023).

Figure 52. Aggregation variability opportunity from generative rules (by author, 2023).

Figure 53. Straight stair aggregation (by author, 2023).

Figure 54. Two stair aggregation (by author, 2023).

Figure 55. Traditional columns and beams aggregation (by author, 2023).

Figure 56. Customized column and beam aggregation (by author, 2023).

Figure 57. Transition aggregation of horizontal and vertical planes (by author, 2023).

Figure 58. Column and beams aggregation (by author, 2023).

Figure 59. Four methods of generative design (by author, 2023).

Figure 60. Rotterdam southern bank of the Nieuwe Maas River (Google Earth, 2022).

Figure 61. Industrial port area history of the neighborhood (Beeldarchief Rijkswaterstaat, between 1900 and 1926).

Figure 62. Kop van Zuid-Entrepot neighborhood (Google Earth, 2022).

Figure 63. Topography and accessibility of the site (Google Earth, 2022).

Figure 64. Data of Kop van Zuid-Entrepot neighborhood (Wonen in Rotterdam, n.d.).

Figure 65. Relationship between conventional housing and housing platform. Adapted from Cheng et al. (2020).

Figure 66. Relationship between ownership by wealth and ownership by number of users. Adapted from Cheng et al. (2020).

Figure 67. User's platform interface app. Adapted from Cheng et al. (2020)

Figure 68. Automated workflow overview (by author, 2023).

Figure 69. Boundary box geometry. The site occupation strategy (by author, 2023).

Figure 70. Aggregated components to shape the building in two aggregation levels (by author, 2023).

Figure 71. One of the endless possibilities of the generated structure (by author, 2023).

Figure 72. Ground floor (by author, 2023).

Figure 73. Architectural Design overview (by author 2023).

Figure 74. Housing design overview (by author 2023).

Figure 75. Housing assembled structure (by author 2023).

Figure 76. Columns formations on the facade (by author 2023).

Figure 77. External facade structure aggregation (by author, 2023).

Figure 78. Balcony and structure extension (by author 2023).

Figure 79. Facade viewing the Laan op Zuid Avenue (by author, 2023).

Figure 80. Housing balcony (by author, 2023).

Figure 81. Community space and possible future housing expansion (by author 2023).

Figure 82. Facade viewing the community space backyard (by author, 2023).

Figure 83. Community terraces on each floor with urban farming in cachepots (by author 2023).

Figure 84. Interior views from the assembly to the inhabit. It is up to the inhabitant to customize her own space (by author, 2023).

Figure 85. Housing entrance from the community backyard (by author 2023).

Figure 86. Human-robot collaboration on the construction site (by author, 2023).

Figure 87. The design-to-build workflow (by author, 2023).

Figure 88. Diagram of the robotic construction workflow (by author, 2023).

Figure 89. Off-site fully automated fabrication (by author, 2023).

Figure 90. The layout of the pallets and it on a truck (by author, 2023).

Figure 91. Off-site fabrication workflow (by author, 2023).

Figure 92. Human as automated assembly's tolerance (by author, 2023).

Figure 93. Table 1. Workflow evaluation for HRC implementation (by author, 2023). Table based on Loo (2022, p.100).

Figure 94. Robot computer vision and sensing checking the elements for calibration (by author, 2023).

Figure 95. Recognition of vertical and horizontal surface and bolting role calibration of discrete elements (by author, 2023).

Figure 96. The on-track 6-axis-arm robots in flexible platforms fixed on the ground (by author, 2023).

Figure 97. The 6-axis-arm mobile robots pre-assembling on the ground (by author, 2023).

Figure 98. On-site assembly workflow (by author, 2023).

Figure 99. On-site HRC checking the assembly positioning (by author, 2023).

Figure 100. Local placement of the vertical components after the pre-assembled horizontal components have been placed by the crane (by author, 2023).

Figure 101. Overview of the on-site automated assembly workflow (by author, 2023).

Figure 102. 1:1 prototype discrete elements (by author, 2023)

Figure 103. 3D printing tests for precision achievement (by author 2023).

Figure 104. Hybrid solution of resin 3D printed bolt heads and M6 standardized metallic threads (by author 2023).

Figure 105. Resin 3D printed nuts with M6 nuts (by author 2023).

Figure 106. Laser-cut wood representing the metallic profile (by author 2023).

Figure 107. CNC milled sockets for nuts placement (by author 2023).

Figure 108. All parts of the 1:1 connection prototype (by author 2023).

Figure 109. Discrete elements used in the prototype (by author 2023).

Figure 110. 1:1 connection prototype assembled (by author 2023).

Figure 111. Possible connection configuration 1 (by author 2023).

Figure 112. Possible connection configuration 2 (by author 2023).

Figure 113. Possible connection configuration 3 (by author 2023).

Figure 114. Possible connection configuration 4 (by author 2023).

Figure 115. Connection prototype vertically assembled (by author 2023).

Figure 116. 1 bolting hole per point of connection representing the 4 bolting holes (by author 2023).

Figure 117. Hybrid solution of resin 3D printed connections and bolt heads, wood CNC milling, and standardized metallic M6 nuts and threads (by author 2023).

Figure 118. All parts of the 1:5 prototype before being assembled (by author 2023).

Figure 119. 3D printing on the resin (by author 2023).

Figure 120. 1:5 hybrid solution assembled bolts (by author 2023).

Figure 121. Visual programming phases of the 1:5 experiment setting up (by author 2023).

Figure 122. End effector design (by author 2023).

Figure 123. Negative concave bolt's geometry in the robot's end effector (by author 2023).

Figure 124. The robot experiment setup (by author 2023).

Figure 125. The sequence of movements between the 3 workstations (by author 2023).

Figure 126. The Grasshopper workflow and each functionality of the 26 target points (by author 2023).

Figure 127. The human and robot agents involved in the experiment and the three workstations (by author 2023).

Figure 128. Prototype assembled by HRC (by author 2023).

Figure 129. Assembly process (by author 2023).

Figure 130. Calibration process (by author 2023).

Figure 131. Bolting process while the robot waits for the conclusion of the task (by author 2023).

Figure 132. Robot testing of the bolt pick-up process (by author 2023).

Figure 133. 1:5 prototype overview (by author 2023).

Figure 134. 1:5 prototype connections (by author 2023).

Figure 135. 1:5 prototype connection details (by author 2023).

Figure 136. 1:5 prototype connection details (by author 2023).

Figure 137. Detail of the connection in prototype 1:10 (by author 2023).

Figure 138. Prototype 1:10 overview (by author, 2023).

Figure 139. The aggregated aesthetic of the 1:10 prototype (by author 2023).

Figure 140. Discretized wood formation of the 1:10 prototype (by author 2023).

Figure 141. Close view of the aggregated formations patterns 1 (by author 2023).

Figure 142. Close view of the aggregated formations patterns 2 (by author 2023).

Figure 143. Close view of the aggregated formations patterns 3 (by author 2023).

Figure 144. Discrete structure formations aesthetics of the prototype 1:10 (by author 2023).

Figure 145. Discrete element aggregation (by author, 2023).

Figure 146. Basalt Columns in Iceland (CarSiceland, n.d.).

Figure 147. Discrete element aggregation. (by author, 2023).

Appendix

Appendix A

UR5 Control Program Code

```
0. def Program():
1.  GripperOpenTcp = p[0, -0.031, 0.203, 0, 0, 1.5708]
2.  GripperOpenWeight = 1.78
3.  GripperOpenCog = [0, 0, 0.077]
4.  GripperClosedTcp = p[0, -0.031, 0.203, 0, 0, 1.5708]
5.  GripperClosedWeight = 1.78
6.  GripperClosedCog = [0, 0, 0.077]
7.  Speed000 = 0.1
8.  Speed001 = 0.1
9.  Speed002 = 0.07
10. Speed003 = 0.07
11. Speed004 = 0.1
12. Speed005 = 0.07
13. Speed006 = 0.07
14. Speed007 = 0.1
15. Speed008 = 0.07
16. Speed009 = 0.07
17. Speed010 = 0.1
18. Speed011 = 0.07
19. Speed012 = 0.07
20. Speed013 = 0.1
21. Speed014 = 0.07
22. Speed015 = 0.07
23. Speed016 = 0.1
24. Speed017 = 0.07
25. Speed018 = 0.07
26. Speed019 = 0.1
27. Speed020 = 0.07
28. Speed021 = 0.07
29. Speed022 = 0.1
30. Speed023 = 0.1
31. Speed024 = 0.07
32. Speed025 = 0.07
33. Speed026 = 0.1
34. Speed027 = 0.1
35. Speed028 = 0.07
36. Speed029 = 0.07
37. Speed030 = 0.1
38. Speed031 = 0.07
39. Speed032 = 0.07
40. Speed033 = 0.1
41. Speed034 = 0.07
42. Speed035 = 0.07
43. Speed036 = 0.1
44. Speed037 = 0.07
45. Speed038 = 0.07
46. Speed039 = 0.1
47. Speed040 = 0.07
48. Speed041 = 0.07
49. Speed042 = 0.1
50. Speed043 = 0.07
51. Speed044 = 0.07
52. Speed045 = 0.1
53. Speed046 = 0.07
54. Speed047 = 0.07
55. Speed048 = 0.1
56. Speed049 = 0.1
57. Speed050 = 0.07

58. Speed051 = 0.07
59. Speed052 = 0.1
60. Speed053 = 0.1
61. Speed054 = 0.07
62. Speed055 = 0.07
63. Speed056 = 0.1
64. Speed057 = 0.07
65. Speed058 = 0.07
66. Speed059 = 0.1
67. Speed060 = 0.07
68. Speed061 = 0.07
69. Speed062 = 0.1
70. Speed063 = 0.07
71. Speed064 = 0.07
72. Speed065 = 0.1
73. Speed066 = 0.07
74. Speed067 = 0.07
75. Speed068 = 0.1
76. Speed069 = 0.07
77. Speed070 = 0.07
78. Speed071 = 0.1
79. Speed072 = 0.07
80. Speed073 = 0.07
81. Speed074 = 0.1
82. Speed075 = 0.1
83. Speed076 = 0.07
84. Speed077 = 0.07
85. Speed078 = 0.1
86. Speed079 = 0.1
87. Speed080 = 0.07
88. Speed081 = 0.07
89. Speed082 = 0.1
90. Speed083 = 0.07
91. Speed084 = 0.07
92. Speed085 = 0.1
93. Speed086 = 0.07
94. Speed087 = 0.07
95. Speed088 = 0.1
96. Speed089 = 0.07
97. Speed090 = 0.07
98. Speed091 = 0.1
99. Speed092 = 0.07
100. Speed093 = 0.07
101. Speed094 = 0.1
102. Speed095 = 0.07
103. Speed096 = 0.07
104. Speed097 = 0.1
105. Speed098 = 0.07
106. Speed099 = 0.07
107. Speed100 = 0.1
108. Speed101 = 0.1
109. Speed102 = 0.07
110. Speed103 = 0.07
111. Speed104 = 0.1
112. Speed105 = 0.1
113. Speed106 = 0.07
114. Speed107 = 0.07
115. Speed108 = 0.1
116. Speed109 = 0.07
117. Speed110 = 0.07
118. Speed111 = 0.1
119. Speed112 = 0.07
120. Speed113 = 0.07
121. Speed114 = 0.1
122. Speed115 = 0.07

123. Speed116 = 0.07
124. Speed117 = 0.1
125. Speed118 = 0.07
126. Speed119 = 0.07
127. Speed120 = 0.1
128. Speed121 = 0.07
129. Speed122 = 0.07
130. Speed123 = 0.1
131. Speed124 = 0.07
132. Speed125 = 0.07
133. Speed126 = 0.1
134. Speed127 = 0.1
135. Speed128 = 0.07
136. Speed129 = 0.07
137. Speed130 = 0.1
138. Speed131 = 0.1
139. Speed132 = 0.07
140. Speed133 = 0.07
141. Speed134 = 0.1
142. Speed135 = 0.07
143. Speed136 = 0.07
144. Speed137 = 0.1
145. Speed138 = 0.07
146. Speed139 = 0.07
147. Speed140 = 0.1
148. Speed141 = 0.07
149. Speed142 = 0.07
150. Speed143 = 0.1
151. Speed144 = 0.07
152. Speed145 = 0.07
153. Speed146 = 0.1
154. Speed147 = 0.07
155. Speed148 = 0.07
156. Speed149 = 0.1
157. Speed150 = 0.07
158. Speed151 = 0.07
159. Speed152 = 0.1
160. Speed153 = 0.1
161. Speed154 = 0.07
162. Speed155 = 0.07
163. Speed156 = 0.1
164. Speed157 = 0.1
165. Speed158 = 0.07
166. Speed159 = 0.07
167. Speed160 = 0.1
168. Speed161 = 0.07
169. Speed162 = 0.07
170. Speed163 = 0.1
171. Speed164 = 0.07
172. Speed165 = 0.07
173. Speed166 = 0.1
174. Speed167 = 0.07
175. Speed168 = 0.07
176. Speed169 = 0.1
177. Speed170 = 0.07
178. Speed171 = 0.07
179. Speed172 = 0.1
180. Speed173 = 0.07
181. Speed174 = 0.07
182. Speed175 = 0.1
183. Speed176 = 0.07
184. Speed177 = 0.07
185. Speed178 = 0.1
186. Speed179 = 0.1
187. Speed180 = 0.07

188. Speed181 = 0.07
189. Speed182 = 0.1
190. Speed183 = 0.1
191. Speed184 = 0.07
192. Speed185 = 0.07
193. Speed186 = 0.1
194. Speed187 = 0.07
195. Speed188 = 0.07
196. Speed189 = 0.1
197. Speed190 = 0.07
198. Speed191 = 0.07
199. Speed192 = 0.1
200. Speed193 = 0.07
201. Speed194 = 0.07
202. Speed195 = 0.1
203. Speed196 = 0.07
204. Speed197 = 0.07
205. Speed198 = 0.1
206. Speed199 = 0.07
207. Speed200 = 0.07
208. Speed201 = 0.1
209. Speed202 = 0.07
210. Speed203 = 0.07
211. Speed204 = 0.1
212. Speed205 = 0.1
213. Speed206 = 0.07
214. Speed207 = 0.07
215. Speed208 = 0.1
216. Speed209 = 0.1
217. Speed210 = 0.07
218. Speed211 = 0.07
219. Speed212 = 0.1
220. Speed213 = 0.07
221. Speed214 = 0.07
222. Speed215 = 0.1
223. Speed216 = 0.07
224. Speed217 = 0.07
225. Speed218 = 0.1
226. Speed219 = 0.07
227. Speed220 = 0.07
228. Speed221 = 0.1
229. Speed222 = 0.07
230. Speed223 = 0.07
231. Speed224 = 0.1
232. Speed225 = 0.07
233. Speed226 = 0.07
234. Speed227 = 0.1
235. Speed228 = 0.07
236. Speed229 = 0.07
237. Speed230 = 0.1
238. Speed231 = 0.1
239. Speed232 = 0.07
240. Speed233 = 0.07
241. Speed234 = 0.1
242. Speed235 = 0.1
243. Speed236 = 0.07
244. Speed237 = 0.07
245. Speed238 = 0.1
246. Speed239 = 0.07
247. Speed240 = 0.07
248. Speed241 = 0.1
249. Speed242 = 0.07
250. Speed243 = 0.07
251. Speed244 = 0.1
252. Speed245 = 0.07

253. Speed246 = 0.07
254. Speed247 = 0.1
255. Speed248 = 0.07
256. Speed249 = 0.07
257. Speed250 = 0.1
258. Speed251 = 0.07
259. Speed252 = 0.07
260. Speed253 = 0.1
261. Speed254 = 0.07
262. Speed255 = 0.07
263. Speed256 = 0.1
264. Speed257 = 0.1
265. Speed258 = 0.07
266. Speed259 = 0.07
267. Speed260 = 0.1
268. Speed261 = 0.1
269. Speed262 = 0.07
270. Speed263 = 0.07
271. Speed264 = 0.1
272. Speed265 = 0.07
273. Speed266 = 0.07
274. Speed267 = 0.1
275. Speed268 = 0.07
276. Speed269 = 0.07
277. Speed270 = 0.1
278. Speed271 = 0.07
279. Speed272 = 0.07
280. Speed273 = 0.1
281. Speed274 = 0.07
282. Speed275 = 0.07
283. Speed276 = 0.1
284. Speed277 = 0.07
285. Speed278 = 0.07
286. Speed279 = 0.1
287. Speed280 = 0.07
288. Speed281 = 0.07
289. Speed282 = 0.1
290. Speed283 = 0.1
291. Speed284 = 0.07
292. Speed285 = 0.07
293. Speed286 = 0.1
294. Speed287 = 0.1
295. Speed288 = 0.07
296. Speed289 = 0.07
297. Speed290 = 0.1
298. Speed291 = 0.07
299. Speed292 = 0.07
300. Speed293 = 0.1
301. Speed294 = 0.07
302. Speed295 = 0.07
303. Speed296 = 0.1
304. Speed297 = 0.07
305. Speed298 = 0.07
306. Speed299 = 0.1
307. Speed300 = 0.07
308. Speed301 = 0.07
309. Speed302 = 0.1
310. Speed303 = 0.07
311. Speed304 = 0.07
312. Speed305 = 0.1
313. Speed306 = 0.07
314. Speed307 = 0.07
315. Speed308 = 0.1
316. Speed309 = 0.1
317. Speed310 = 0.07

318. Speed311 = 0.07
319. Speed312 = 0.1
320. Speed313 = 0.1
321. Speed314 = 0.07
322. Speed315 = 0.07
323. Speed316 = 0.1
324. Speed317 = 0.07
325. Speed318 = 0.07
326. Speed319 = 0.1
327. Speed320 = 0.07
328. Speed321 = 0.07
329. Speed322 = 0.1
330. Speed323 = 0.07
331. Speed324 = 0.07
332. Speed325 = 0.1
333. Speed326 = 0.07
334. Speed327 = 0.07
335. Speed328 = 0.1
336. Speed329 = 0.07
337. Speed330 = 0.07
338. Speed331 = 0.1
339. Speed332 = 0.07
340. Speed333 = 0.07
341. Speed334 = 0.1
342. Speed335 = 0.1
343. Speed336 = 0.07
344. Speed337 = 0.07
345. Speed338 = 0.1
346. Speed339 = 0.1
347. DefaultZone = 0
348. Wait000 = 1
349. Wait001 = 1
350. Wait002 = 1
351. Wait003 = 1
352. Wait004 = 1
353. Wait005 = 1
354. Wait006 = 1
355. Wait007 = 1
356. set_tool_voltage(24)

357. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
358. movej([-0.0581, -0.9857, -1.6392, -0.5166, 0.0581, 3.1416], a=3.1416,
v=0.3142, r=DefaultZone)
359. movej([-0.1607, -1.208, -1.2776, -2.2174, 1.5643, 6.121], a=3.1416,
v=0.7531, r=DefaultZone)
360. move(p[0.26889, -0.1528, 0.14206, -0.00228, 3.13338, -0.01244], a=1,
v=Speed002, r=DefaultZone)
361. sleep(Wait000)
362. set_digital_out(8,True)
363. sleep(Wait001)
364. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
365. move(p[0.26643, -0.15042, 0.44204, -0.00228, 3.13338, -0.01244], a=1,
v=Speed003, r=DefaultZone)
366. movej([-1.3652, -1.0208, -2.2571, -1.4251, 1.5692, 3.357], a=3.1416,
v=0.5899, r=DefaultZone)
367. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed005, r=DefaultZone)
368. sleep(Wait002)
369. set_digital_out(8,False)
370. sleep(Wait003)
371. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
372. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed006, r=DefaultZone)
373. movej([-1.7813, -1.2666, -2.1015, -1.3351, 1.5731, 2.9409], a=3.1416,
v=0.3135, r=DefaultZone)
374. move(p[-0.17903, -0.29076, 0.04216, -2.21056, 2.23224, -0.01101], a=1,
v=Speed008, r=DefaultZone)
375. sleep(Wait000)
376. set_digital_out(8,True)
377. sleep(Wait001)
378. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
379. move(p[-0.17906, -0.28999, 0.12216, -2.21056, 2.23224, -0.01101], a=1,
v=Speed009, r=DefaultZone)
380. movej([-0.6273, -0.903, -1.8874, -1.9161, 1.5633, 0.9532], a=3.1416,
v=0.6231, r=DefaultZone)
381. move(p[0.09695, -0.24064, 0.07378, 2.23225, 2.21056, -0.01015], a=1,
v=Speed011, r=DefaultZone)
382. sleep(Wait002)
383. set_digital_out(8,False)
384. sleep(Wait003)
385. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
386. move(p[0.09689, -0.23873, 0.27377, 2.23225, 2.21056, -0.01015], a=1,
v=Speed012, r=DefaultZone)
387. movej([-1.6586, -1.2148, -2.1346, -1.3535, 1.572, 3.0636], a=3.1416,
v=0.7443, r=DefaultZone)
388. move(p[-0.13704, -0.29116, 0.04218, -2.21056, 2.23224, -0.01101], a=1,
v=Speed014, r=DefaultZone)
389. sleep(Wait000)
390. set_digital_out(8,True)
391. sleep(Wait001)
392. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
393. move(p[-0.13706, -0.2904, 0.12218, -2.21056, 2.23224, -0.01101], a=1,
v=Speed015, r=DefaultZone)
394. movej([-1.8947, -1.0223, -1.8467, -1.8345, 1.5742, 5.969], a=3.1416,
v=1.7356, r=DefaultZone)
395. move(p[-0.18303, -0.23791, 0.07366, 2.23225, 2.21056, -0.01015], a=1,
v=Speed017, r=DefaultZone)
396. sleep(Wait002)
397. set_digital_out(8,False)
```



```
398. sleep(Wait003)
399. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
400. move(p[-0.1831, -0.23601, 0.27365, 2.23225, 2.21056, -0.01015], a=1,
v=Speed018, r=DefaultZone)
401. movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=1.2655, r=DefaultZone)
402. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed020, r=DefaultZone)
403. sleep(Wait004)
404. set_digital_out(8,True)
405. sleep(Wait005)
406. while not get_digital_in(6):
    sleep(0.008)
    end
407. sleep(Wait006)
408. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
409. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed021, r=DefaultZone)
410. movej([-0.721, -1.1498, -2.3413, 0.3735, 0.7119, 3.1237], a=3.1416,
v=04452, r=DefaultZone)
411. movej([-1.3465, -1.7292, -1.6453, -1.3225, 1.5675, 0.2335], a=3.1416,
v=0.9778, r=DefaultZone)
412. move(p[0.01418, -0.58989, 0.05526, 2.22363, 2.20318, -0.01712], a=1,
v=Speed024, r=DefaultZone)
413. sleep(Wait005)
414. while not get_digital_in(6):
    sleep(0.008)
    end
415. sleep(Wait006)
416. set_digital_out(8,False)
417. sleep(Wait007)
418. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
419. move(p[0.01417, -0.58863, 0.13525, 2.22363, 2.20318, -0.01712], a=1,
v=Speed025, r=DefaultZone)
420. movej([-0.721, -1.1498, -2.3413, 0.3735, 0.7119, 3.1237], a=3.1416,
v=0.9778, r=DefaultZone)
421. movej([-0.1912, -1.109, -1.3878, -2.206, 1.5646, 6.0905], a=3.1416,
v=0.7849, r=DefaultZone)
422. move(p[0.22905, -0.15302, 0.12174, -0.00228, 3.13338, -0.01244], a=1,
v=Speed028, r=DefaultZone)
423. sleep(Wait000)
424. set_digital_out(8,True)
425. sleep(Wait001)
426. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
427. move(p[0.2266, -0.15064, 042172, -0.00228, 3.13338, -0.01244], a=1,
v=Speed029, r=DefaultZone)
428. movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=0.6395, r=DefaultZone)
429. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed031, r=DefaultZone)
430. sleep(Wait002)
431. set_digital_out(8,False)
432. sleep(Wait003)
433. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
434. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed032, r=DefaultZone)
435. movej([-1.5281, -1.1646, -2.1126, -14257, 1.5707, 3.194], a=3.1416,
v=0.2481, r=DefaultZone)
436. move(p[-0.09504, -0.29138, 0.0622, -2.21056, 2.23224, -0.01101], a=1,
v=Speed034, r=DefaultZone)
437. sleep(Wait000)
438. set_digital_out(8,True)
439. sleep(Wait001)
440. set_tcp(GripperClosedTcp)
```

```
    set_payload(GripperClosedWeight, GripperClosedCog)
441. move(p[-0.09507, -0.29062, 0.14219, -2.21056, 2.23224, -0.01101], a=1,
v=Speed035, r=DefaultZone)
442. movej([-1.968, -1.0701, -1.7728, -1.8609, 1.5748, 5.8957], a=3.1416,
v=1.3927, r=DefaultZone)
443. move(p[-0.20304, -0.23753, 0.09365, 2.23225, 2.21056, -0.01015], a=1,
v=Speed037, r=DefaultZone)
444. sleep(Wait002)
445. set_digital_out(8,False)
446. sleep(Wait003)
447. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
448. move(p[-0.20311, -0.23562, 0.29364, 2.23225, 2.21056, -0.01015], a=1,
v=Speed038, r=DefaultZone)
449. movej([-1.3934, -1.1471, -2.1738, -1.382, 1.5695, 3.3287], a=3.1416,
v=1.0947, r=DefaultZone)
450. move(p[-0.05304, -0.29198, 0.04222, -2.21056, 2.23224, -0.01101], a=1,
v=Speed040, r=DefaultZone)
451. sleep(Wait000)
452. set_digital_out(8,True)
453. sleep(Wait001)
454. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
455. move(p[-0.05307, -0.29122, 0.12221, -2.21056, 2.23224, -0.01101], a=1,
v=Speed041, r=DefaultZone)
456. movej([-1.9409, -1.0944, -1.8136, -1.7956, 1.5746, 5.9228], a=3.1416,
v=1.199, r=DefaultZone)
457. move(p[-0.20323, -0.25772, 0.07384, 2.23225, 2.21056, -0.01015], a=1,
v=Speed043, r=DefaultZone)
458. sleep(Wait002)
459. set_digital_out(8,False)
460. sleep(Wait003)
461. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
462. move(p[-0.2033, -0.25581, 0.27383, 2.23225, 2.21056, -0.01015], a=1,
v=Speed044, r=DefaultZone)
463. movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=1.1687, r=DefaultZone)
464. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed046, r=DefaultZone)
465. sleep(Wait004)
466. set_digital_out(8,True)
467. sleep(Wait005)
468. while not get_digital_in(6):
    sleep(0.008)
    end
469. sleep(Wait006)
470. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
471. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed047, r=DefaultZone)
472. movej([-3.1015, -14828, -1.8389, 04888, 3.0899, 34502], a=3.1416,
v=0.3977, r=DefaultZone)
473. movej([-1.7949, -1.9511, -2.6222, 14478, 1.7856, 4.7161], a=3.1416,
v=0.2819, r=DefaultZone)
474. move(p[-0.12547, -0.5464, 0.19457, -0.01053, -2.20388, 2.23884], a=1,
v=Speed050, r=DefaultZone)
475. sleep(Wait005)
476. while not get_digital_in(6):
    sleep(0.008)
    end
477. sleep(Wait006)
478. set_digital_out(8,False)
479. sleep(Wait007)
480. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
481. move(p[-0.12473, -046642, 0.19332, -0.01053, -2.20388, 2.23884], a=1,
v=Speed051, r=DefaultZone)
482. movej([-3.1015, -14828, -1.8389, 04888, 3.0899, 34502], a=3.1416,
```

```
v=0.2819, r=DefaultZone)
483. movej([-0.1612, -1.1798, -1.355, -2.1682, 1.5643, 6.1205], a=3.1416,
v=04895, r=DefaultZone)
484. move(p[0.26905, -0.15296, 0.12206, -0.00228, 3.13338, -0.01244], a=1,
v=Speed054, r=DefaultZone)
485. sleep(Wait000)
486. set_digital_out(8,True)
487. sleep(Wait001)
488. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
489. move(p[0.2666, -0.15058, 042204, -0.00228, 3.13338, -0.01244], a=1,
v=Speed055, r=DefaultZone)
490. movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=0.6076, r=DefaultZone)
491. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed057, r=DefaultZone)
492. sleep(Wait002)
493. set_digital_out(8,False)
494. sleep(Wait003)
495. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
496. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed058, r=DefaultZone)
497. movej([-1.2592, -1.1356, -2.18, -1.3876, 1.5682, 34629], a=3.1416,
v=0.1893, r=DefaultZone)
498. move(p[-0.01104, -0.29239, 0.04224, -2.21056, 2.23224, -0.01101], a=1,
v=Speed060, r=DefaultZone)
499. sleep(Wait000)
500. set_digital_out(8,True)
501. sleep(Wait001)
502. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
503. move(p[-0.01107, -0.29163, 0.12223, -2.21056, 2.23224, -0.01101], a=1,
v=Speed061, r=DefaultZone)
504. movej([-0.7647, -0.9315, -1.8786, -1.8955, 1.5642, 0.8158], a=3.1416,
v=1484, r=DefaultZone)
505. move(p[0.07676, -0.26044, 0.07396, 2.23225, 2.21056, -0.01015], a=1,
v=Speed063, r=DefaultZone)
506. sleep(Wait002)
507. set_digital_out(8,False)
508. sleep(Wait003)
509. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
510. move(p[0.07669, -0.25854, 0.27395, 2.23225, 2.21056, -0.01015], a=1,
v=Speed064, r=DefaultZone)
511. movej([-1.1302, -1.1406, -2.1771, -1.3859, 1.567, 3.5919], a=3.1416,
v=1.7141, r=DefaultZone)
512. move(p[0.03096, -0.2928, 0.04225, -2.21056, 2.23224, -0.01101], a=1,
v=Speed066, r=DefaultZone)
513. sleep(Wait000)
514. set_digital_out(8,True)
515. sleep(Wait001)
516. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
517. move(p[0.03093, -0.29204, 0.12225, -2.21056, 2.23224, -0.01101], a=1,
v=Speed067, r=DefaultZone)
518. movej([-1.9409, -1.0944, -1.8136, -1.7956, 1.5746, 5.9228], a=3.1416,
v=0.8285, r=DefaultZone)
519. move(p[-0.20323, -0.25772, 0.07384, 2.23225, 2.21056, -0.01015], a=1,
v=Speed069, r=DefaultZone)
520. sleep(Wait002)
521. set_digital_out(8,False)
522. sleep(Wait003)
523. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
524. move(p[-0.2033, -0.25581, 0.27383, 2.23225, 2.21056, -0.01015], a=1,
v=Speed070, r=DefaultZone)
525. movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=1.1687, r=DefaultZone)
```

```
526. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed072, r=DefaultZone)
527. sleep(Wait004)
528. set_digital_out(8,True)
529. sleep(Wait005)
530. while not get_digital_in(6):
    sleep(0.008)
    end
531. sleep(Wait006)
532. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
533. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed073, r=DefaultZone)
534. movej([-2.6223, -1.1171, -1.8497, -0.1437, 2.6129, 3.1688], a=3.1416,
v=0.2444, r=DefaultZone)
535. movej([-1.7116, -1.8252, -2.701, 14004, 1.7023, 4.7148], a=3.1416,
v=0.3501, r=DefaultZone)
536. move(p[-0.10529, -0.52628, 0.21426, -0.01053, -2.20388, 2.23884], a=1,
v=Speed076, r=DefaultZone)
537. sleep(Wait005)
538. while not get_digital_in(6):
    sleep(0.008)
    end
539. sleep(Wait006)
540. set_digital_out(8,False)
541. sleep(Wait007)
542. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
543. move(p[-0.10455, -044629, 0.21301, -0.01053, -2.20388, 2.23884], a=1,
v=Speed077, r=DefaultZone)
544. movej([-2.6223, -1.1171, -1.8497, -0.1437, 2.6129, 3.1688], a=3.1416,
v=0.3501, r=DefaultZone)
545. movej([-0.1917, -1.0818, -14611, -2.1599, 1.5646, 6.09], a=3.1416,
v=0.5645, r=DefaultZone)
546. move(p[0.22922, -0.15318, 0.10174, -0.00228, 3.13338, -0.01244], a=1,
v=Speed080, r=DefaultZone)
547. sleep(Wait000)
548. set_digital_out(8,True)
549. sleep(Wait001)
550. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
551. move(p[0.22676, -0.1508, 040172, -0.00228, 3.13338, -0.01244], a=1,
v=Speed081, r=DefaultZone)
552. movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=0.6612, r=DefaultZone)
553. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed083, r=DefaultZone)
554. sleep(Wait002)
555. set_digital_out(8,False)
556. sleep(Wait003)
557. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
558. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed084, r=DefaultZone)
559. movej([-1.0106, -1.1656, -2.1813, -1.3572, 1.566, 3.7115], a=3.1416,
v=0.2656, r=DefaultZone)
560. move(p[0.07296, -0.29328, 0.03527, -2.21056, 2.23224, -0.01101], a=1,
v=Speed086, r=DefaultZone)
561. sleep(Wait000)
562. set_digital_out(8,True)
563. sleep(Wait001)
564. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
565. move(p[0.07293, -0.29251, 0.11527, -2.21056, 2.23224, -0.01101], a=1,
v=Speed087, r=DefaultZone)
566. movej([-0.7649, -0.9261, -1.8991, -1.8803, 1.5642, 0.8156], a=3.1416,
v=1.8626, r=DefaultZone)
567. move(p[0.07676, -0.26051, 0.06696, 2.23225, 2.21056, -0.01015], a=1,
v=Speed089, r=DefaultZone)
```



```
568. sleep(Wait002)
569. set_digital_out(8,False)
570. sleep(Wait003)
571. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
572. move(p[0.07669, -0.2586, 0.26695, 2.23225, 2.21056, -0.01015], a=1,
v=Speed090, r=DefaultZone)
573. movej([-1.767, -1.3199, -2.0116, -1.3717, 1.573, 2.9551], a=3.1416,
v=0.7362, r=DefaultZone)
574. move(p[-0.1793, -0.31756, 0.06242, -2.21056, 2.23224, -0.01101], a=1,
v=Speed092, r=DefaultZone)
575. sleep(Wait000)
576. set_digital_out(8,True)
577. sleep(Wait001)
578. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
579. move(p[-0.17933, -0.3168, 0.14242, -2.21056, 2.23224, -0.01101], a=1,
v=Speed093, r=DefaultZone)
580. movej([-1.8719, -1.0742, -1.7705, -1.8588, 1.5739, 5.9919], a=3.1416,
v=1.8597, r=DefaultZone)
581. move(p[-0.18323, -0.25772, 0.09385, 2.23225, 2.21056, -0.01015], a=1,
v=Speed095, r=DefaultZone)
582. sleep(Wait002)
583. set_digital_out(8,False)
584. sleep(Wait003)
585. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
586. move(p[-0.1833, -0.25581, 0.29384, 2.23225, 2.21056, -0.01015], a=1,
v=Speed096, r=DefaultZone)
587. movej([-1.3652, -1.0208, -2.2571, -1.4251, 1.5692, 3.357], a=3.1416,
v=1.1855, r=DefaultZone)
588. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed098, r=DefaultZone)
589. sleep(Wait004)
590. set_digital_out(8,True)
591. sleep(Wait005)
592. while not get_digital_in(6):
    sleep(0.008)
    end
593. sleep(Wait006)
594. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
595. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed099, r=DefaultZone)
596. movej([-2.6223, -1.1171, -1.8497, -0.1437, 2.6129, 3.1688], a=3.1416,
v=0.2444, r=DefaultZone)
597. movej([-2.1706, -2.5514, -0.6346, -3.0977, 2.5511, 1.5546], a=3.1416,
v=0.6448, r=DefaultZone)
598. move(p[-0.14528, -0.52591, 0.21425, -1.21621, 1.20836, -1.2277], a=1,
v=Speed102, r=DefaultZone)
599. sleep(Wait005)
600. while not get_digital_in(6):
    sleep(0.008)
    end
601. sleep(Wait006)
602. set_digital_out(8,False)
603. sleep(Wait007)
604. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
605. move(p[-0.22528, -0.52517, 0.21422, -1.21621, 1.20836, -1.2277], a=1,
v=Speed103, r=DefaultZone)
606. movej([-2.6223, -1.1171, -1.8497, -0.1437, 2.6129, 3.1688], a=3.1416,
v=0.6448, r=DefaultZone)
607. movej([-0.1617, -1.154, -1.4287, -2.1203, 1.5643, 6.12], a=3.1416,
v=0.5338, r=DefaultZone)
608. move(p[0.26922, -0.15312, 0.10206, -0.00228, 3.13338, -0.01244], a=1,
v=Speed106, r=DefaultZone)
609. sleep(Wait000)
610. set_digital_out(8,True)
```

```
611. sleep(Wait001)
612. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
613. move(p[0.26676, -0.15074, 0.40204, -0.00228, 3.13338, -0.01244], a=1,
v=Speed107, r=DefaultZone)
614. movej([-1.3652, -1.0208, -2.2571, -1.4251, 1.5692, 3.357], a=3.1416,
v=0.6256, r=DefaultZone)
615. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed109, r=DefaultZone)
616. sleep(Wait002)
617. set_digital_out(8,False)
618. sleep(Wait003)
619. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
620. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed110, r=DefaultZone)
621. movej([-1.6527, -1.2737, -2.044, -1.3852, 1.5719, 3.0694], a=3.1416,
v=0.2563, r=DefaultZone)
622. move(p[p[-0.13731, -0.31797, 0.06244, -2.21056, 2.23224, -0.01101], a=1,
v=Speed112, r=DefaultZone)
623. sleep(Wait000)
624. set_digital_out(8,True)
625. sleep(Wait001)
626. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
627. move(p[-0.13733, -0.31721, 0.14243, -2.21056, 2.23224, -0.01101], a=1,
v=Speed113, r=DefaultZone)
628. movej([-1.9662, -1.0692, -1.7732, -1.8613, 1.5748, 5.8975], a=3.1416,
v=1.5387, r=DefaultZone)
629. move(p[-0.20254, -0.23753, 0.09365, 2.23225, 2.21056, -0.01015], a=1,
v=Speed115, r=DefaultZone)
630. sleep(Wait002)
631. set_digital_out(8,False)
632. sleep(Wait003)
633. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
634. move(p[-0.20261, -0.23563, 0.29364, 2.23225, 2.21056, -0.01015], a=1,
v=Speed116, r=DefaultZone)
635. movej([-1.5322, -1.2488, -2.1125, -1.3416, 1.5708, 3.1899], a=3.1416,
v=1.2414, r=DefaultZone)
636. move(p[-0.0953, -0.31857, 0.04246, -2.21056, 2.23224, -0.01101], a=1,
v=Speed118, r=DefaultZone)
637. sleep(Wait000)
638. set_digital_out(8,True)
639. sleep(Wait001)
640. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
641. move(p[-0.09533, -0.31781, 0.12245, -2.21056, 2.23224, -0.01101], a=1,
v=Speed119, r=DefaultZone)
642. movej([-1.9392, -1.0935, -1.814, -1.796, 1.5745, 5.9245], a=3.1416,
v=1.3972, r=DefaultZone)
643. move(p[-0.20273, -0.25772, 0.07384, 2.23225, 2.21056, -0.01015], a=1,
v=Speed121, r=DefaultZone)
644. sleep(Wait002)
645. set_digital_out(8,False)
646. sleep(Wait003)
647. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
648. move(p[-0.2028, -0.25581, 0.27383, 2.23225, 2.21056, -0.01015], a=1,
v=Speed122, r=DefaultZone)
649. movej([-1.3652, -1.0208, -2.2571, -1.4251, 1.5692, 3.357], a=3.1416,
v=1.1713, r=DefaultZone)
650. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed124, r=DefaultZone)
651. sleep(Wait004)
652. set_digital_out(8,True)
653. sleep(Wait005)
654. while not get_digital_in(6):
    sleep(0.008)
```

```
    end
655. sleep(Wait006)
656. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
657. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed125, r=DefaultZone)
658. movej([-2.043, -0.7116, -2.2436, -0.1689, 2.0337, 3.1498], a=3.1416,
v=0.3124, r=DefaultZone)
659. movej([-1.9857, -1.2323, -2.5263, -2.5074, 4.3068, 1.5779], a=3.1416,
v=0.6634, r=DefaultZone)
660. move(p[-0.14491, -0.48652, 0.17513, -0.01053, -2.20388, 2.23884], a=1,
v=Speed128, r=DefaultZone)
661. sleep(Wait005)
662. while not get_digital_in(6):
    sleep(0.008)
    end
663. sleep(Wait006)
664. set_digital_out(8,False)
665. sleep(Wait007)
666. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
667. move(p[-0.14417, -0.40653, 0.17387, -0.01053, -2.20388, 2.23884], a=1,
v=Speed129, r=DefaultZone)
668. movej([-2.043, -0.7116, -2.2436, -0.1689, 2.0337, 3.1498], a=3.1416,
v=0.6634, r=DefaultZone)
669. movej([-0.1923, -1.0566, -1.5314, -2.1148, 1.5646, 6.0894], a=3.1416,
v=0.7322, r=DefaultZone)
670. move(p[0.22938, -0.15334, 0.08174, -0.00228, 3.13338, -0.01244], a=1,
v=Speed132, r=DefaultZone)
671. sleep(Wait000)
672. set_digital_out(8,True)
673. sleep(Wait001)
674. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
675. move(p[0.22692, -0.15096, 0.38172, -0.00228, 3.13338, -0.01244], a=1,
v=Speed133, r=DefaultZone)
676. movej([-1.3652, -1.0208, -2.2571, -1.4251, 1.5692, 3.357], a=3.1416,
v=0.6836, r=DefaultZone)
677. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed135, r=DefaultZone)
678. sleep(Wait002)
679. set_digital_out(8,False)
680. sleep(Wait003)
681. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
682. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed136, r=DefaultZone)
683. movej([-1.4085, -1.2144, -2.0823, -1.4062, 1.5696, 3.3137], a=3.1416,
v=0.2583, r=DefaultZone)
684. move(p[-0.05331, -0.31879, 0.06247, -2.21056, 2.23224, -0.01101], a=1,
v=Speed138, r=DefaultZone)
685. sleep(Wait000)
686. set_digital_out(8,True)
687. sleep(Wait001)
688. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
689. move(p[-0.05334, -0.31803, 0.14247, -2.21056, 2.23224, -0.01101], a=1,
v=Speed139, r=DefaultZone)
690. movej([-0.6871, -0.9003, -1.8334, -1.9724, 1.5636, 0.8934], a=3.1416,
v=1.1245, r=DefaultZone)
691. move(p[0.07745, -0.24026, 0.09377, 2.23225, 2.21056, -0.01015], a=1,
v=Speed141, r=DefaultZone)
692. sleep(Wait002)
693. set_digital_out(8,False)
694. sleep(Wait003)
695. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
696. move(p[0.07738, -0.23835, 0.29376, 2.23225, 2.21056, -0.01015], a=1,
v=Speed142, r=DefaultZone)
```

```
697. movej([-1.285, -1.2044, -2.0883, -1.4104, 1.5684, 3.4371], a=3.1416,
v=1.3194, r=DefaultZone)
698. move(p[-0.01131, -0.3192, 0.06249, -2.21056, 2.23224, -0.01101], a=1,
v=Speed144, r=DefaultZone)
699. sleep(Wait000)
700. set_digital_out(8,True)
701. sleep(Wait001)
702. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
703. move(p[-0.01134, -0.31844, 0.14249, -2.21056, 2.23224, -0.01101], a=1,
v=Speed145, r=DefaultZone)
704. movej([-1.8931, -1.0352, -1.7881, -1.8802, 1.5741, 5.9707], a=3.1416,
v=1.043, r=DefaultZone)
705. move(p[-0.18254, -0.23773, 0.09366, 2.23225, 2.21056, -0.01015], a=1,
v=Speed147, r=DefaultZone)
706. sleep(Wait002)
707. set_digital_out(8,False)
708. sleep(Wait003)
709. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
710. move(p[-0.18261, -0.23582, 0.29365, 2.23225, 2.21056, -0.01015], a=1,
v=Speed148, r=DefaultZone)
711. movej([-1.3652, -1.0208, -2.2571, -1.4251, 1.5692, 3.357], a=3.1416,
v=1.1775, r=DefaultZone)
712. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed150, r=DefaultZone)
713. sleep(Wait004)
714. set_digital_out(8,True)
715. sleep(Wait005)
716. while not get_digital_in(6):
    sleep(0.008)
    end
717. sleep(Wait006)
718. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
719. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed151, r=DefaultZone)
720. movej([0.2717, -1.4807, -1.5913, -0.1262, -0.2813, 3.1963], a=3.1416,
v=0.2836, r=DefaultZone)
721. movej([-0.4933, -2.6873, -0.4725, -3.123, 1.0867, 1.5863], a=3.1416,
v=0.6309, r=DefaultZone)
722. move(p[-0.10491, -0.48689, 0.17514, -1.22529, -1.19509, 1.21382], a=1,
v=Speed154, r=DefaultZone)
723. sleep(Wait005)
724. while not get_digital_in(6):
    sleep(0.008)
    end
725. sleep(Wait006)
726. set_digital_out(8,False)
727. sleep(Wait007)
728. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
729. move(p[0.34507, -0.49105, 0.17528, -1.22529, -1.19509, 1.21382], a=1,
v=Speed155, r=DefaultZone)
730. movej([0.2717, -1.4807, -1.5913, -0.1262, -0.2813, 3.1963], a=3.1416,
v=0.6309, r=DefaultZone)
731. movej([-0.1622, -1.1303, -1.4993, -2.0734, 1.5643, 6.1195], a=3.1416,
v=1.0203, r=DefaultZone)
732. move(p[0.26938, -0.15328, 0.08206, -0.00228, 3.13338, -0.01244], a=1,
v=Speed158, r=DefaultZone)
733. sleep(Wait000)
734. set_digital_out(8,True)
735. sleep(Wait001)
736. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
737. move(p[0.26692, -0.1509, 0.38205, -0.00228, 3.13338, -0.01244], a=1,
v=Speed159, r=DefaultZone)
738. movej([-1.3652, -1.0208, -2.2571, -1.4251, 1.5692, 3.357], a=3.1416,
v=0.644, r=DefaultZone)
```



```
739.  move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed161, r=DefaultZone)
740.  sleep(Wait002)
741.  set_digital_out(8,False)
742.  sleep(Wait003)
743.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
744.  move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed162, r=DefaultZone)
745.  movej([-1.1657, -1.2087, -2.0856, -14092, 1.5674, 3.5564], a=3.1416,
v=0.1766, r=DefaultZone)
746.  move(p[0.03069, -0.31961, 0.06251, -2.21056, 2.23224, -0.01101], a=1,
v=Speed164, r=DefaultZone)
747.  sleep(Wait000)
748.  set_digital_out(8,True)
749.  sleep(Wait001)
750.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
751.  move(p[0.03066, -0.31885, 0.14251, -2.21056, 2.23224, -0.01101], a=1,
v=Speed165, r=DefaultZone)
752.  movej([-1.9627, -1.0675, -1.774, -1.8623, 1.5748, 5.901], a=3.1416,
v=0.8103, r=DefaultZone)
753.  move(p[-0.20154, -0.23754, 0.09365, 2.23225, 2.21056, -0.01015], a=1,
v=Speed167, r=DefaultZone)
754.  sleep(Wait002)
755.  set_digital_out(8,False)
756.  sleep(Wait003)
757.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
758.  move(p[-0.20161, -0.23564, 0.29364, 2.23225, 2.21056, -0.01015], a=1,
v=Speed168, r=DefaultZone)
759.  movej([-1.054, -1.2376, -2.1192, -1.3471, 1.5664, 3.6682], a=3.1416,
v=0.6686, r=DefaultZone)
760.  move(p[0.07269, -0.32021, 0.04253, -2.21056, 2.23224, -0.01101], a=1,
v=Speed170, r=DefaultZone)
761.  sleep(Wait000)
762.  set_digital_out(8,True)
763.  sleep(Wait001)
764.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
765.  move(p[0.07266, -0.31944, 0.12253, -2.21056, 2.23224, -0.01101], a=1,
v=Speed171, r=DefaultZone)
766.  movej([-1.9359, -1.0919, -1.8149, -1.7969, 1.5745, 5.9278], a=3.1416,
v=0.7066, r=DefaultZone)
767.  move(p[-0.20173, -0.25773, 0.07384, 2.23225, 2.21056, -0.01015], a=1,
v=Speed173, r=DefaultZone)
768.  sleep(Wait002)
769.  set_digital_out(8,False)
770.  sleep(Wait003)
771.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
772.  move(p[-0.2018, -0.25582, 0.27383, 2.23225, 2.21056, -0.01015], a=1,
v=Speed174, r=DefaultZone)
773.  movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=1.1765, r=DefaultZone)
774.  move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed176, r=DefaultZone)
775.  sleep(Wait004)
776.  set_digital_out(8,True)
777.  sleep(Wait005)
778.  while not get_digital_in(6):
    sleep(0.008)
    end
779.  sleep(Wait006)
780.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
781.  move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed177, r=DefaultZone)
782.  movej([0.5996, -1.0018, -148, -2.2393, 1.5577, 3.7505], a=3.1416,
```

```
v=04403, r=DefaultZone)
783.  movej([-0.5818, -1.316, -24084, -2.5298, -0.5727, 1.5467], a=3.1416,
v=04047, r=DefaultZone)
784.  move(p[0.15452, -0.54897, 0.19616, -0.01053, -2.20388, 2.23884], a=1,
v=Speed180, r=DefaultZone)
785.  sleep(Wait005)
786.  while not get_digital_in(6):
    sleep(0.008)
    end
787.  sleep(Wait006)
788.  set_digital_out(8,False)
789.  sleep(Wait007)
790.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
791.  move(p[0.15498, -049897, 0.19538, -0.01053, -2.20388, 2.23884], a=1,
v=Speed181, r=DefaultZone)
792.  movej([0.5996, -1.0018, -148, -2.2393, 1.5577, 3.7505], a=3.1416,
v=04047, r=DefaultZone)
793.  movej([-0.1929, -1.0333, -1.5992, -2.0704, 1.5646, 6.0888], a=3.1416,
v=1.384, r=DefaultZone)
794.  move(p[0.22955, -0.1535, 0.06174, -0.00228, 3.13338, -0.01244], a=1,
v=Speed184, r=DefaultZone)
795.  sleep(Wait000)
796.  set_digital_out(8,True)
797.  sleep(Wait001)
798.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
799.  move(p[0.22709, -0.15112, 0.36172, -0.00228, 3.13338, -0.01244], a=1,
v=Speed185, r=DefaultZone)
800.  movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=0.7065, r=DefaultZone)
801.  move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed187, r=DefaultZone)
802.  sleep(Wait002)
803.  set_digital_out(8,False)
804.  sleep(Wait003)
805.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
806.  move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed188, r=DefaultZone)
807.  movej([-1.7545, -1.3833, -1.9627, -1.3571, 1.5729, 2.9676], a=3.1416,
v=0.2422, r=DefaultZone)
808.  move(p[-0.17957, -0.34456, 0.06268, -2.21056, 2.23224, -0.01101], a=1,
v=Speed190, r=DefaultZone)
809.  sleep(Wait000)
810.  set_digital_out(8,True)
811.  sleep(Wait001)
812.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
813.  move(p[-0.17959, -0.3438, 0.14267, -2.21056, 2.23224, -0.01101], a=1,
v=Speed191, r=DefaultZone)
814.  movej([-0.6887, -0.8998, -1.8335, -1.9727, 1.5637, 0.8918], a=3.1416,
v=0.6573, r=DefaultZone)
815.  move(p[0.07695, -0.24025, 0.09377, 2.23225, 2.21056, -0.01015], a=1,
v=Speed193, r=DefaultZone)
816.  sleep(Wait002)
817.  set_digital_out(8,False)
818.  sleep(Wait003)
819.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
820.  move(p[0.07688, -0.23835, 0.29376, 2.23225, 2.21056, -0.01015], a=1,
v=Speed194, r=DefaultZone)
821.  movej([-1.6476, -1.3413, -1.9951, -1.3665, 1.5719, 3.0745], a=3.1416,
v=0.7716, r=DefaultZone)
822.  move(p[-0.13757, -0.34497, 0.06269, -2.21056, 2.23224, -0.01101], a=1,
v=Speed196, r=DefaultZone)
823.  sleep(Wait000)
824.  set_digital_out(8,True)
825.  sleep(Wait001)
```

```
826.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
827.  move(p[-0.1376, -0.34421, 0.14269, -2.21056, 2.23224, -0.01101], a=1,
v=Speed197, r=DefaultZone)
828.  movej([-1.895, -1.0361, -1.7877, -1.8797, 1.5742, 5.9688], a=3.1416,
v=1.5127, r=DefaultZone)
829.  move(p[-0.18304, -0.23772, 0.09366, 2.23225, 2.21056, -0.01015], a=1,
v=Speed199, r=DefaultZone)
830.  sleep(Wait002)
831.  set_digital_out(8,False)
832.  sleep(Wait003)
833.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
834.  move(p[-0.18311, -0.23582, 0.29365, 2.23225, 2.21056, -0.01015], a=1,
v=Speed200, r=DefaultZone)
835.  movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=1.1751, r=DefaultZone)
836.  move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed202, r=DefaultZone)
837.  sleep(Wait004)
838.  set_digital_out(8,True)
839.  sleep(Wait005)
840.  while not get_digital_in(6):
    sleep(0.008)
    end
841.  sleep(Wait006)
842.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
843.  move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed203, r=DefaultZone)
844.  movej([-0.2342, -1.7377, -0.9216, -2.0493, 1.5555, 44874], a=3.1416,
v=0.2213, r=DefaultZone)
845.  movej([-0.6073, -2.6055, -0.5827, -3.0945, 0.9727, 1.5863], a=3.1416,
v=0.6369, r=DefaultZone)
846.  move(p[0.19452, -0.54936, 0.19467, -1.22529, -1.19509, 1.21382], a=1,
v=Speed206, r=DefaultZone)
847.  sleep(Wait005)
848.  while not get_digital_in(6):
    sleep(0.008)
    end
849.  sleep(Wait006)
850.  set_digital_out(8,False)
851.  sleep(Wait007)
852.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
853.  move(p[0.27452, -0.5501, 0.1947, -1.22529, -1.19509, 1.21382], a=1,
v=Speed207, r=DefaultZone)
854.  movej([-0.2342, -1.7377, -0.9216, -2.0493, 1.5555, 44874], a=3.1416,
v=0.6369, r=DefaultZone)
855.  movej([-0.1628, -1.1087, -1.567, -2.0273, 1.5643, 6.119], a=3.1416,
v=0.7568, r=DefaultZone)
856.  move(p[0.26954, -0.15344, 0.06207, -0.00228, 3.13338, -0.01244], a=1,
v=Speed210, r=DefaultZone)
857.  sleep(Wait000)
858.  set_digital_out(8,True)
859.  sleep(Wait001)
860.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
861.  move(p[0.26709, -0.15106, 0.36205, -0.00228, 3.13338, -0.01244], a=1,
v=Speed211, r=DefaultZone)
862.  movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=0.6627, r=DefaultZone)
863.  move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed213, r=DefaultZone)
864.  sleep(Wait002)
865.  set_digital_out(8,False)
866.  sleep(Wait003)
867.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
```

```
868.  move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed214, r=DefaultZone)
869.  movej([-1.5358, -1.3207, -2.0625, -1.3197, 1.5708, 3.1864], a=3.1416,
v=0.2847, r=DefaultZone)
870.  move(p[-0.09556, -0.34557, 0.04271, -2.21056, 2.23224, -0.01101], a=1,
v=Speed216, r=DefaultZone)
871.  sleep(Wait000)
872.  set_digital_out(8,True)
873.  sleep(Wait001)
874.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
875.  move(p[-0.09559, -0.34481, 0.12271, -2.21056, 2.23224, -0.01101], a=1,
v=Speed217, r=DefaultZone)
876.  movej([-0.6258, -0.9036, -1.8872, -1.9157, 1.5633, 0.9547], a=3.1416,
v=0.8357, r=DefaultZone)
877.  move(p[0.09745, -0.24064, 0.07378, 2.23225, 2.21056, -0.01015], a=1,
v=Speed219, r=DefaultZone)
878.  sleep(Wait002)
879.  set_digital_out(8,False)
880.  sleep(Wait003)
881.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
882.  move(p[0.09739, -0.23874, 0.27377, 2.23225, 2.21056, -0.01015], a=1,
v=Speed220, r=DefaultZone)
883.  movej([-14213, -1.2879, -2.0335, -1.3815, 1.5697, 3.3008], a=3.1416,
v=1.0362, r=DefaultZone)
884.  move(p[-0.05357, -0.34579, 0.06273, -2.21056, 2.23224, -0.01101], a=1,
v=Speed222, r=DefaultZone)
885.  sleep(Wait000)
886.  set_digital_out(8,True)
887.  sleep(Wait001)
888.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
889.  move(p[-0.0536, -0.34502, 0.14273, -2.21056, 2.23224, -0.01101], a=1,
v=Speed223, r=DefaultZone)
890.  movej([-1.8931, -1.0352, -1.7881, -1.8802, 1.5741, 5.9707], a=3.1416,
v=1.1782, r=DefaultZone)
891.  move(p[-0.18254, -0.23773, 0.09366, 2.23225, 2.21056, -0.01015], a=1,
v=Speed225, r=DefaultZone)
892.  sleep(Wait002)
893.  set_digital_out(8,False)
894.  sleep(Wait003)
895.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
896.  move(p[-0.18261, -0.23582, 0.29365, 2.23225, 2.21056, -0.01015], a=1,
v=Speed226, r=DefaultZone)
897.  movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=1.1775, r=DefaultZone)
898.  move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed228, r=DefaultZone)
899.  sleep(Wait004)
900.  set_digital_out(8,True)
901.  sleep(Wait005)
902.  while not get_digital_in(6):
    sleep(0.008)
    end
903.  sleep(Wait006)
904.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
905.  move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed229, r=DefaultZone)
906.  movej([-0.6562, -1.9741, -04338, -2.2947, 1.5584, 4.0654], a=3.1416,
v=0.322, r=DefaultZone)
907.  movej([2.1685, -2.2055, 2.6532, -04669, 2.1776, 1.5602], a=3.1416,
v=0.8658, r=DefaultZone)
908.  move(p[0.17489, -0.50947, 0.17554, -0.01053, -2.20388, 2.23884], a=1,
v=Speed232, r=DefaultZone)
909.  sleep(Wait005)
910.  while not get_digital_in(6):
```



```

sleep(0.008)
end
911. sleep(Wait006)
912. set_digital_out(8,False)
913. sleep(Wait007)
914. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
915. move(p[0.17674, -0.3095, 0.1724, -0.01053, -2.20388, 2.23884], a=1,
v=Speed233, r=DefaultZone)
916. movej([-0.6562, -1.9741, -0.4338, -2.2947, 1.5584, 4.0654], a=3.1416,
v=0.8658, r=DefaultZone)
917. movej([-0.1935, -0.1118, -1.6646, -2.0265, 1.5646, 6.0882], a=3.1416,
v=0.6582, r=DefaultZone)
918. move(p[0.22971, -0.15366, 0.04174, -0.00228, 3.13338, -0.01244], a=1,
v=Speed236, r=DefaultZone)
919. sleep(Wait000)
920. set_digital_out(8,True)
921. sleep(Wait001)
922. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
923. move(p[0.22725, -0.15128, 0.34172, -0.00228, 3.13338, -0.01244], a=1,
v=Speed237, r=DefaultZone)
924. movej([-1.3652, -1.0208, -2.2571, -1.4251, 1.5692, 3.357], a=3.1416,
v=0.7298, r=DefaultZone)
925. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed239, r=DefaultZone)
926. sleep(Wait002)
927. set_digital_out(8,False)
928. sleep(Wait003)
929. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
930. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed240, r=DefaultZone)
931. movej([-1.3072, -1.2903, -2.0838, -1.329, 1.5686, 3.4149], a=3.1416,
v=0.2548, r=DefaultZone)
932. move(p[-0.01157, -0.34639, 0.04275, -2.21056, 2.23224, -0.01101], a=1,
v=Speed242, r=DefaultZone)
933. sleep(Wait000)
934. set_digital_out(8,True)
935. sleep(Wait001)
936. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
937. move(p[-0.01159, -0.34562, 0.12275, -2.21056, 2.23224, -0.01101], a=1,
v=Speed243, r=DefaultZone)
938. movej([-0.6273, -0.903, -1.8874, -1.9161, 1.5633, 0.9532], a=3.1416,
v=1.1477, r=DefaultZone)
939. move(p[0.09695, -0.24064, 0.07378, 2.23225, 2.21056, -0.01015], a=1,
v=Speed245, r=DefaultZone)
940. sleep(Wait002)
941. set_digital_out(8,False)
942. sleep(Wait003)
943. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
944. move(p[0.09689, -0.23873, 0.27377, 2.23225, 2.21056, -0.01015], a=1,
v=Speed246, r=DefaultZone)
945. movej([-1.1961, -1.2828, -2.0368, -1.3838, 1.5676, 3.526], a=3.1416,
v=1.4151, r=DefaultZone)
946. move(p[0.03042, -0.3466, 0.06277, -2.21056, 2.23224, -0.01101], a=1,
v=Speed248, r=DefaultZone)
947. sleep(Wait000)
948. set_digital_out(8,True)
949. sleep(Wait001)
950. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
951. move(p[0.0304, -0.34584, 0.14276, -2.21056, 2.23224, -0.01101], a=1,
v=Speed249, r=DefaultZone)
952. movej([-1.9412, -1.1063, -1.7551, -1.8423, 1.5746, 5.9226], a=3.1416,
v=0.8192, r=DefaultZone)
953. move(p[-0.20323, -0.25753, 0.09384, 2.23225, 2.21056, -0.01015], a=1,

```

```

=Speed251, r=DefaultZone)
954. sleep(Wait002)
955. set_digital_out(8,False)
956. sleep(Wait003)
957. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
958. move(p[-0.2033, -0.25562, 0.29383, 2.23225, 2.21056, -0.01015], a=1,
v=Speed252, r=DefaultZone)
959. movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=1.094, r=DefaultZone)
960. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed254, r=DefaultZone)
961. sleep(Wait004)
962. set_digital_out(8,True)
963. sleep(Wait005)
964. while not get_digital_in(6):
    sleep(0.008)
    end
965. sleep(Wait006)
966. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
967. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed255, r=DefaultZone)
968. movej([04209, -1.9749, -0.2721, -24717, 1.5564, 3.5718], a=3.1416,
v=0.2669, r=DefaultZone)
969. movej([-0.463, -1.0362, -2.7356, 0.666, 0454, 6.2513], a=3.1416, v=0446,
r=DefaultZone)
970. move(p[0.03487, -0.50537, 0.35398, 1.20451, 1.19305, -1.22321], a=1,
v=Speed258, r=DefaultZone)
971. sleep(Wait005)
972. while not get_digital_in(6):
    sleep(0.008)
    end
973. sleep(Wait006)
974. set_digital_out(8,False)
975. sleep(Wait007)
976. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
977. move(p[0.03561, -0.42539, 0.35272, 1.20451, 1.19305, -1.22321], a=1,
v=Speed259, r=DefaultZone)
978. movej([04209, -1.9749, -0.2721, -24717, 1.5564, 3.5718], a=3.1416,
v=0.446, r=DefaultZone)
979. movej([-0.1633, -1.089, -1.6324, -1.9816, 1.5643, 6.1184], a=3.1416,
v=0.6975, r=DefaultZone)
980. move(p[0.26971, -0.1536, 0.04207, -0.00228, 3.13338, -0.01244], a=1,
v=Speed262, r=DefaultZone)
981. sleep(Wait000)
982. set_digital_out(8,True)
983. sleep(Wait001)
984. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
985. move(p[0.26725, -0.15122, 0.34205, -0.00228, 3.13338, -0.01244], a=1,
v=Speed263, r=DefaultZone)
986. movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=0.6813, r=DefaultZone)
987. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed265, r=DefaultZone)
988. sleep(Wait002)
989. set_digital_out(8,False)
990. sleep(Wait003)
991. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
992. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed266, r=DefaultZone)
993. movej([-1.091, -1.2991, -2.0253, -1.3793, 1.5667, 3.6311], a=3.1416,
v=0.1723, r=DefaultZone)
994. move(p[0.07242, -0.34701, 0.06279, -2.21056, 2.23224, -0.01101], a=1,
v=Speed268, r=DefaultZone)
995. sleep(Wait000)

```

```

996. set_digital_out(8,True)
997. sleep(Wait001)
998. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
999. move(p[0.07239, -0.34625, 0.14278, -2.21056, 2.23224, -0.01101], a=1,
v=Speed269, r=DefaultZone)
1000. movej([-0.7025, -0.9667, -1.8126, -1.9266, 1.5637, 0.878], a=3.1416,
v=1.5601, r=DefaultZone)
1001. move(p[0.09675, -0.26045, 0.09397, 2.23225, 2.21056, -0.01015], a=1,
v=Speed271, r=DefaultZone)
1002. sleep(Wait002)
1003. set_digital_out(8,False)
1004. sleep(Wait003)
1005. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
1006. move(p[0.09668, -0.25854, 0.29396, 2.23225, 2.21056, -0.01015], a=1,
v=Speed272, r=DefaultZone)
1007. movej([-1.7434, -1.4583, -1.952, -1.2928, 1.5728, 2.9787], a=3.1416,
v=0.6106, r=DefaultZone)
1008. move(p[-0.17982, -0.37175, 0.04293, -2.21056, 2.23224, -0.01101],
a=1, v=Speed274, r=DefaultZone)
1009. sleep(Wait000)
1010. set_digital_out(8,True)
1011. sleep(Wait001)
1012. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
1013. move(p[-0.17985, -0.37099, 0.12293, -2.21056, 2.23224, -0.01101],
a=1, v=Speed275, r=DefaultZone)
1014. movej([-1.8716, -1.0615, -1.8292, -1.8127, 1.5739, 5.9921], a=3.1416,
v=1.588, r=DefaultZone)
1015. move(p[-0.18323, -0.25791, 0.07385, 2.23225, 2.21056, -0.01015], a=1,
v=Speed277, r=DefaultZone)
1016. sleep(Wait002)
1017. set_digital_out(8,False)
1018. sleep(Wait003)
1019. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
1020. move(p[-0.1833, -0.256, 0.27384, 2.23225, 2.21056, -0.01015], a=1,
v=Speed278, r=DefaultZone)
1021. movej([-1.3652, -1.0208, -2.2571, -1.4251, 1.5692, 3.357], a=3.1416,
v=1.2767, r=DefaultZone)
1022. move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101],
a=1, v=Speed280, r=DefaultZone)
1023. sleep(Wait004)
1024. set_digital_out(8,True)
1025. sleep(Wait005)
1026. while not get_digital_in(6):
    sleep(0.008)
end
1027. sleep(Wait006)
1028. set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
1029. move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed281, r=DefaultZone)
1030. movej([-0.0355, -1.5799, -0.7859, -2.3459, 1.5551, 4.6861], a=3.1416,
v=0.246, r=DefaultZone)
1031. movej([-0.1721, -0.9097, -2.7111, 0.5758, 0.1637, 6.1882], a=3.1416,
v=0.6314, r=DefaultZone)
1032. move(p[0.05505, -0.48525, 0.37366, 1.20451, 1.19305, -1.22321], a=1,
v=Speed284, r=DefaultZone)
1033. sleep(Wait005)
1034. while not get_digital_in(6):
    sleep(0.008)
end
1035. sleep(Wait006)
1036. set_digital_out(8,False)
1037. sleep(Wait007)
1038. set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)

```

```

1039.  move(p[0.05579, -0.04526, 0.37241, 1.20451, 1.19305, -1.22321], a=1,
v=Speed285, r=DefaultZone)
1040.  move(p[[-0.0355, -1.5799, -0.7859, -2.3459, 1.5551, 4.6861], a=3.1416,
v=0.6314, r=DefaultZone)
1041.  move(p[[-0.194, -0.9919, -1.728, -1.9828, 1.5646, 6.0877], a=3.1416,
v=0.5839, r=DefaultZone)
1042.  move(p[0.22987, -0.15381, 0.02174, -0.00228, 3.13338, -0.01244], a=1,
v=Speed288, r=DefaultZone)
1043.  sleep(Wait000)
1044.  set_digital_out(8,True)
1045.  sleep(Wait001)
1046.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
1047.  move(p[0.22742, -0.15143, 0.32172, -0.00228, 3.13338, -0.01244], a=1,
v=Speed289, r=DefaultZone)
1048.  move(p[[-1.3652, -1.0208, -2.2571, -1.4251, 1.5692, 3.357], a=3.1416,
v=0.7532, r=DefaultZone)
1049.  move(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101],
a=1, v=Speed291, r=DefaultZone)
1050.  sleep(Wait002)
1051.  set_digital_out(8,False)
1052.  sleep(Wait003)
1053.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
1054.  move(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed292, r=DefaultZone)
1055.  move(p[[-1.6432, -1.4074, -1.9426, -1.3529, 1.5718, 3.0789], a=3.1416,
v=0.2548, r=DefaultZone)
1056.  move(p[-0.13783, -0.37197, 0.06295, -2.21056, 2.23224, -0.01101],
a=1, v=Speed294, r=DefaultZone)
1057.  sleep(Wait000)
1058.  set_digital_out(8,True)
1059.  sleep(Wait001)
1060.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
1061.  move(p[-0.13786, -0.3712, 0.14295, -2.21056, 2.23224, -0.01101], a=1,
v=Speed295, r=DefaultZone)
1062.  move(p[[-1.9391, -1.1817, -1.7139, -1.808, 1.5745, 4.3538], a=3.1416,
v=0.6527, r=DefaultZone)
1063.  move(p[-0.18323, -0.25772, 0.09385, 0.0153, 3.14152, -0.01496], a=1,
v=Speed297, r=DefaultZone)
1064.  sleep(Wait002)
1065.  set_digital_out(8,False)
1066.  sleep(Wait003)
1067.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
1068.  move(p[-0.1833, -0.25581, 0.29384, 0.0153, 3.14152, -0.01496], a=1,
v=Speed298, r=DefaultZone)
1069.  move(p[[-1.5388, -1.3902, -2.0089, -1.3038, 1.5708, 3.1833], a=3.1416,
v=0.5219, r=DefaultZone)
1070.  move(p[-0.09583, -0.37257, 0.04297, -2.21056, 2.23224, -0.01101],
a=1, v=Speed300, r=DefaultZone)
1071.  sleep(Wait000)
1072.  set_digital_out(8,True)
1073.  sleep(Wait001)
1074.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
1075.  move(p[-0.09585, -0.3718, 0.12297, -2.21056, 2.23224, -0.01101], a=1,
v=Speed301, r=DefaultZone)
1076.  move(p[[-1.9967, -1.2042, -1.7528, -1.7469, 1.575, 4.2962], a=3.1416,
v=0.5092, r=DefaultZone)
1077.  move(p[-0.20323, -0.25772, 0.07384, 0.0153, 3.14152, -0.01496], a=1,
v=Speed303, r=DefaultZone)
1078.  sleep(Wait002)
1079.  set_digital_out(8,False)
1080.  sleep(Wait003)
1081.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
1082.  move(p[-0.2033, -0.25581, 0.27383, 0.0153, 3.14152, -0.01496], a=1,

```



```
v=Speed304, r=DefaultZone)
1083.  movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=04278, r=DefaultZone)
1084.  movej(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101],
a=1, v=Speed306, r=DefaultZone)
1085.  sleep(Wait004)
1086.  set_digital_out(8,True)
1087.  sleep(Wait005)
1088.  while not get_digital_in(6):
    sleep(0.008)
    end
1089.  sleep(Wait006)
1090.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
1091.  movej(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed307, r=DefaultZone)
1092.  movej([-2.5517, -1.56, -1.1209, -2.0228, 1.5839, 5.3116], a=3.1416,
v=04094, r=DefaultZone)
1093.  movej([-1.8858, -1.0009, -2.2757, 0.1515, 1.8765, 4.7177], a=3.1416,
v=0.5727, r=DefaultZone)
1094.  movej(p[-0.12496, -04817, 049359, -0.01053, -2.20388, 2.23884], a=1,
v=Speed310, r=DefaultZone)
1095.  sleep(Wait005)
1096.  while not get_digital_in(6):
    sleep(0.008)
    end
1097.  sleep(Wait006)
1098.  set_digital_out(8,False)
1099.  sleep(Wait007)
1100.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
1101.  movej(p[-0.12422, -040171, 049234, -0.01053, -2.20388, 2.23884], a=1,
v=Speed311, r=DefaultZone)
1102.  movej([-2.5517, -1.56, -1.1209, -2.0228, 1.5839, 5.3116], a=3.1416,
v=0.5727, r=DefaultZone)
1103.  movej([-0.1638, -1.0713, -1.6956, -1.9361, 1.5643, 6.1179], a=3.1416,
v=0.3409, r=DefaultZone)
1104.  movej(p[0.26987, -0.15376, 0.02207, -0.00228, 3.13338, -0.01244], a=1,
v=Speed314, r=DefaultZone)
1105.  sleep(Wait000)
1106.  set_digital_out(8,True)
1107.  sleep(Wait001)
1108.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
1109.  movej(p[0.26741, -0.15138, 0.32205, -0.00228, 3.13338, -0.01244], a=1,
v=Speed315, r=DefaultZone)
1110.  movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=0.6999, r=DefaultZone)
1111.  movej(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101],
a=1, v=Speed317, r=DefaultZone)
1112.  sleep(Wait002)
1113.  set_digital_out(8,False)
1114.  sleep(Wait003)
1115.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
1116.  movej(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed318, r=DefaultZone)
1117.  movej([-14324, -1.3709, -2.024, -1.308, 1.5698, 3.2898], a=3.1416,
v=0.2829, r=DefaultZone)
1118.  movej(p[-0.05383, -0.37298, 0.04299, -2.21056, 2.23224, -0.01101],
a=1, v=Speed320, r=DefaultZone)
1119.  sleep(Wait000)
1120.  set_digital_out(8,True)
1121.  sleep(Wait001)
1122.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
1123.  movej(p[-0.05386, -0.37221, 0.12299, -2.21056, 2.23224, -0.01101],
a=1, v=Speed321, r=DefaultZone)
1124.  movej([-0.6273, -0.903, -1.8874, -1.9161, 1.5633, 0.9532], a=3.1416,
```

```
v=0.9289, r=DefaultZone)
1125.  movej(p[0.09695, -0.24064, 0.07378, 2.23225, 2.21056, -0.01015], a=1,
v=Speed323, r=DefaultZone)
1126.  sleep(Wait002)
1127.  set_digital_out(8,False)
1128.  sleep(Wait003)
1129.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
1130.  movej(p[0.09689, -0.23873, 0.27377, 2.23225, 2.21056, -0.01015], a=1,
v=Speed324, r=DefaultZone)
1131.  movej([-1.326, -1.3508, -1.9871, -1.3652, 1.5688, 3.3961], a=3.1416,
v=1.1292, r=DefaultZone)
1132.  movej(p[-0.01184, -0.37319, 0.06301, -2.21056, 2.23224, -0.01101],
a=1, v=Speed326, r=DefaultZone)
1133.  sleep(Wait000)
1134.  set_digital_out(8,True)
1135.  sleep(Wait001)
1136.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
1137.  movej(p[-0.01186, -0.37243, 0.143, -2.21056, 2.23224, -0.01101], a=1,
v=Speed327, r=DefaultZone)
1138.  movej([-1.895, -1.0361, -1.7877, -1.8797, 1.5742, 5.9688], a=3.1416,
v=0.9677, r=DefaultZone)
1139.  movej(p[-0.18304, -0.23772, 0.09366, 2.23225, 2.21056, -0.01015],
a=1, v=Speed329, r=DefaultZone)
1140.  sleep(Wait002)
1141.  set_digital_out(8,False)
1142.  sleep(Wait003)
1143.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
1144.  movej(p[-0.18311, -0.23582, 0.29365, 2.23225, 2.21056, -0.01015],
a=1, v=Speed330, r=DefaultZone)
1145.  movej([-1.3652, -1.0208, -2.2571, -14251, 1.5692, 3.357], a=3.1416,
v=1.1751, r=DefaultZone)
1146.  movej(p[-0.05312, -0.24956, 0.03381, -2.21056, 2.23224, -0.01101],
a=1, v=Speed332, r=DefaultZone)
1147.  sleep(Wait004)
1148.  set_digital_out(8,True)
1149.  sleep(Wait005)
1150.  while not get_digital_in(6):
    sleep(0.008)
    end
1151.  sleep(Wait006)
1152.  set_tcp(GripperClosedTcp)
    set_payload(GripperClosedWeight, GripperClosedCog)
1153.  movej(p[-0.05315, -0.2488, 0.11381, -2.21056, 2.23224, -0.01101], a=1,
v=Speed333, r=DefaultZone)
1154.  movej([0.325, -1.2569, -1.1186, -3.9556, 0.3346, 3.1871], a=3.1416,
v=0.3945, r=DefaultZone)
1155.  movej([-1.1621, -1.7116, -0.6825, -2.3038, 1.5647, 04179], a=3.1416,
v=1.0085, r=DefaultZone)
1156.  movej(p[0.07504, -04848, 041367, 2.22363, 2.20318, -0.01712], a=1,
v=Speed336, r=DefaultZone)
1157.  sleep(Wait005)
1158.  while not get_digital_in(6):
    sleep(0.008)
    end
1159.  sleep(Wait006)
1160.  set_digital_out(8,False)
1161.  sleep(Wait007)
1162.  set_tcp(GripperOpenTcp)
    set_payload(GripperOpenWeight, GripperOpenCog)
1163.  movej(p[0.07503, -048355, 049366, 2.22363, 2.20318, -0.01712], a=1,
v=Speed337, r=DefaultZone)
1164.  movej([0.325, -1.2569, -1.1186, -3.9556, 0.3346, 3.1871], a=3.1416,
v=1.0085, r=DefaultZone)
1165.  movej([-0.0581, -2.5397, 1.6392, -2.2411, 0.0581, 3.1416], a=3.1416,
v=2.7717, r=DefaultZone)
1166.  end
```

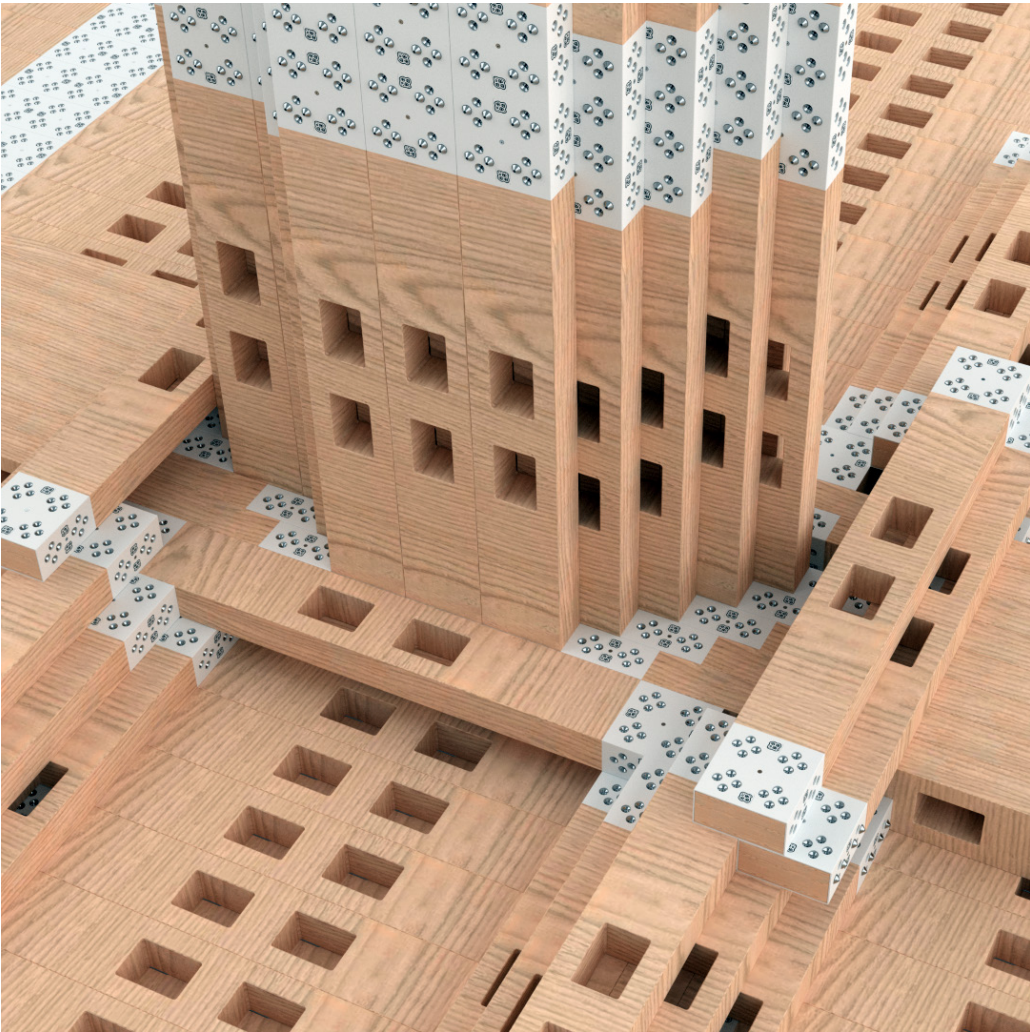


Figure 147. Systems of change (by author, 2023).

