Robotic Fabrication Workflows for Environmentally Driven Facades

Pablo M. Cabrera

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
In
Architecture

Chair: Professor Robert Dunay
Committee: Professor James Jones
Committee: Professor Nathan King

May 13th, 2019
Blacksburg, VA

Keywords: Design computation, Design simulation, Design fabrication

Robotic Fabrication Workflows for Environmentally Driven Facades

Pablo M. Cabrera

ABSTRACT

Even though computer simulation of environmental factors and manufacturing technologies have experienced a fast development, architectural workflows that can take advantage of the possibilities created by these developments have been left behind and architectural design processes have not evolved at the same rate. This research presents design to fabrication workflows that explore data driven design to improve performance of facades, implementing for this purpose computational tools to handle environmental data complexity and proposes robotic fabrication technologies to facilitate façade components fabrication.

During this research three design experiments were conducted that tested variations on the design to fabrication workflow, approaching the flow of information in top-down and bottom-up processes. Independent variables such as material, environmental conditions and structural behavior, are the framework in which workflow instances are generated based on dependent variables such as geometry, orientation and assembly logic. This research demonstrates the feasibility of a robotic based fabrication method informed by a multi-variable computational framework plus a simulation evaluator integrated into a design to fabrication workflow and put forward the discussion of a fully automated scenario.

To my wife

This work was supported by the Center for Design Research
Design Technology Initiative at Virginia Tech

TABLE OF CONTENTS

1. Introduction

Advances in the computer simulation of environmental variables such as light, wind, and sound due to significant growth in the computing power of the day-to-day tools available to designers are expanding the scope of variables that affect design decisions, revealing like never before natural conditions that had remained hidden because of the means of representation traditionally employed by architects to represent their buildings and, consequently, are connecting architecture to the natural sciences (Peters, & Peters, 2018).

Architectural designs that respond objectively to these environmental constraints are increasingly complex (Schwitter, 2005), both in design processes and formal manifestations. The design process is oriented toward performance, but it is important to distinguish two types of performance in architecture: “the kind that can be exact and unfailing in its predictions of outcomes, and the kind that anticipates what is likely, given the circumstantial contingencies of built work. The first sort is technical and productive, the second contextual and projective. There is no need to rank these two in a theory of architectural performance; important instead is grasping their reciprocity and joint necessity" (Leatherbarrow, 2009, p. 18). In the context of this research, performance is understood as the second category.

On the other hand, formal complexity is pushing the boundaries of construction and demanding new means of fabrication. Computer-aided manufacturing processes are increasingly being employed in the fabrication of complex forms, from small prototypes to large architectural components, due to increases in the availability and versatility (Willmann, Block, Byrne, Hutter, & Schork, 2018) of technologies that until a few decades ago, were the exclusive domain of engineers, such as industrial robots. This versatility plays an important role in the adoption of fabrication technologies in the design and architecture fields.

Even though computer simulations of environmental factors and fabrication technologies have developed rapidly, architectural workflows that take advantage of them have been left behind, and architectural design processes have not evolved at the same rate. Despite some interesting proposals, this is an important area of research that is yet to be fully explored, and now is a good opportunity to create such an integral workflow, from design to fabrication (Hauck,

& Bergin, 2017), that can negotiate computational frameworks, environmental simulation, and fabrication.

According to the U.S. Green Building Council's "Buildings and Climate Change" (n.d.), the commercial and residential building sector produces 39% of carbon dioxide emissions in the United States, more than any other sector. Most of these emissions come from the combustion of fossil fuels to provide heating, cooling, and lighting. If embodied emissions, which are the first emissions generated from building materials, products, and construction processes, are taken into account, another problem surfaces. Currently, about 5.7 billion square feet of new buildings are erected in the U.S. every year, and their embodied emissions amount to around 300 million metric tons per year (Strain, 2016).

It is not only the performance of the buildings but their materials and construction processes that need to be more energy efficient and climate friendly. At present there is a disconnect between the performance optimization and the fabrication of buildings and their components. This research explores workflows designed to reconcile this disconnect by proposing new design processes, material systems, and fabrication methods with the aim of moving toward improved

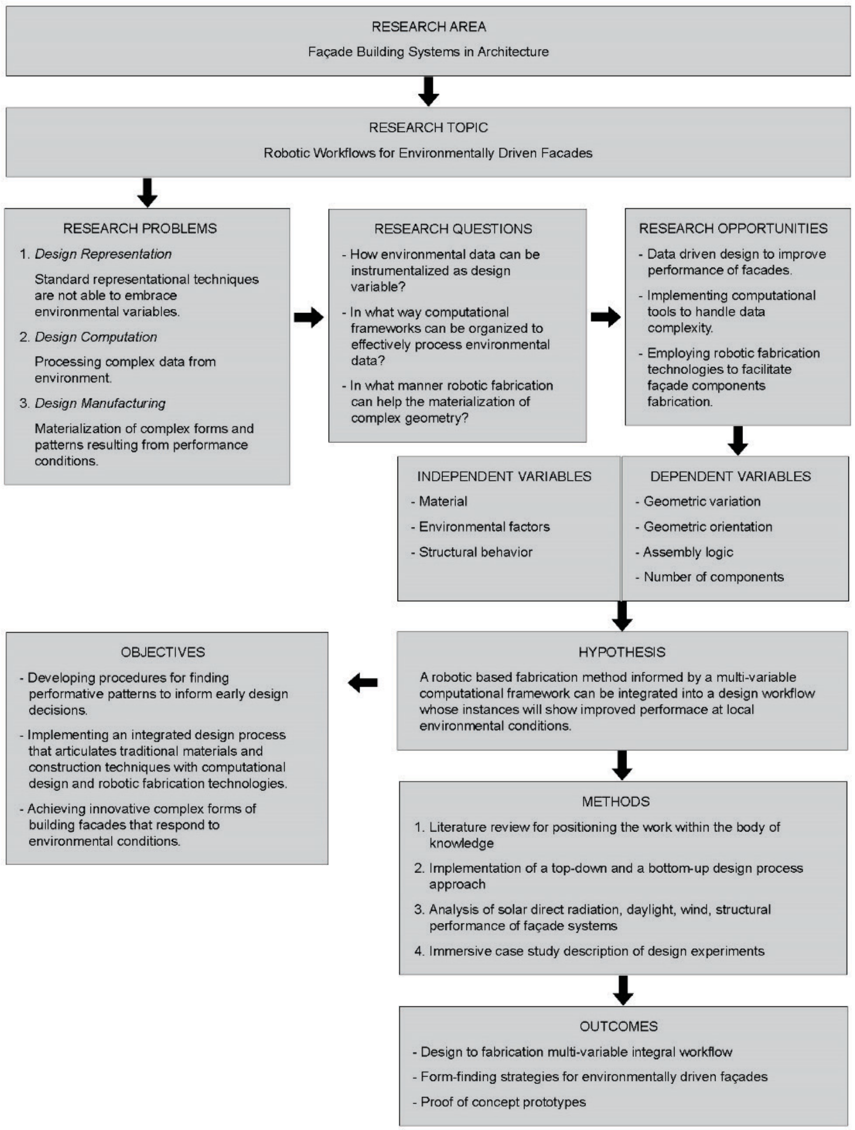

Figure 1

performance. The assumption is that by making the built environment more energy efficient and climate friendly, the building sector can play a major role in reducing the threat of climate change. Through immersive case studies focusing on fabrication, we propose workflows that explore data-driven design to improve the performance of facades, implementing for this purpose computational tools to handle complex environmental data and proposing robotic fabrication technologies to facilitate façade-component fabrication.

1.1 Research Problems

This research falls under the general umbrella of façade-building systems in architecture, and robotic fabrication workflows for environmentally driven facades as a more specific research topic. The problems it addresses fall into three discrete areas of architecture; their organization is shown in Figure 1.

1.1.1 Design Representation

Standard representational techniques in architecture cannot embrace environmental variables because they are designed to represent static geometry, whereas environmental variables such as wind and light are transient in nature and act over time. Designing ways to represent these variables creates an opportunity to also implement a data-driven design process for improving the performance of the façades. This in turn leads us to the question of how environmental data can be instrumentalized as design variables.

1.1.2 Design Computation

Due to their time-based nature, environmental data from simulations are complex both in quality and in quantity. A simulation of solar incident radiation over a facade can easily generate thousands of values of raw data depending on the geometric complexity of the facade. Computational tools are especially suitable for processing large amounts of data, which means they are useful processing the raw data produced by simulation processes and exploring how computational frameworks can be organized to effectively process environmental data.

1.1.3 Design Manufacturing

Complex forms and patterns often result from linking iterative processes of geometry generation and performance simulation. From a manufacturing point of view, however, a problem arises in the fabrication of such forms and patterns. The versatility of industrial robots, however, creates an opportunity to include these highly accurate and flexible tools in an automatic fabrication workflow, with a direct flow from geometric data to fabrication data. Given the open-ended nature of these technologies, the role of robotic fabrication and the construction of complex geometry drives the research process.

1.2 Hypothesis

The general hypothesis of this research is as follows:

A robotics-based fabrication method informed by a multivariable computational framework can be integrated into a design workflow whose instances show improved performance under local environmental conditions.

1.3 Objectives

The research objectives of this work are as follows:

- Developing procedures for finding performative patterns to inform early design decisions.

- Implementing an integrated design process that articulates traditional materials and construction techniques with computational design and robotic fabrication technologies.

- Achieving innovative complex forms for building facades that will respond to environmental conditions.

1.4 Contributions

The expected contributions of this work are weighted heavily toward the process rather than the products, and central to this is a design for multivariable, integral fabrication workflow that interrelates a computational framework, environmental performance simulation, and computer-aided manufacturing. A sub-process of this workflow is a set of form-finding computational strategies for environment-driven facades in the form of Python scripts. Instances of the workflow have been used to fabricate prototypes as a proof of concept of the process.

2. Research Design and Methods

The methods employed in this research operate within the boundaries of design research. First, a literature review positions the work within the existing body of knowledge and identifies advances in precedent cases.

Then two design experiments are employed to pursue the objectives of this research. The first is the implementation of a top-down design process in which design decisions are based on global results of environmental simulations and in turn change the geometric global configuration. The second uses a bottom-up approach in which geometric configuration and environmental simulation interact simultaneously at the component scale in an additive process.

For environment simulation, digital grid-based analysis of direct solar radiation and daylight, computer fluid dynamics, and finite-element structural analysis have been conducted. The chapters on each case study elaborate on this.

Finally, an immersive case-study description of each design experiment provides the results and findings.

2.1 Literature Review

This review covers a wide range of architectural workflows for early design decisions. It is not an exhaustive compilation; it is intended to provide punctual examples rather than paradigmatic ones in order to represent a wider group in which qualitative changes are present. For a comprehensive review of simulation workflows to support early design decisions, refer to the research areas proposed by Ostergard (Ostergard, Jensen, & Maargaard, 2016).

In a design context, workflows can be characterized in two ways (Bhooshan, 2017): drawing-based ones, which contain just geometric information, and model-based ones, which also include other data such as materials and constructability. Bhooshan proposed a hybrid scenario that would combine the flexibility of the drawing-based workflow in relation to the generation of forms with the more rigorous data structure of the model-based approach, effectively joining a computer-aided geometric design approach to a building-information-modeling one. An important step toward an integrated workflow is the synthesis of the two paradigms.

From a model-based workflow, it is possible to review simulation scenarios involving several geometric variations in relation to environmental variables and over specific time ranges (Naboni, Ofria, & Danzo, 2019). It is possible to run many digital environmental simulations at almost no cost and get immediate feedback on performance, which is especially suitable for early design decisions.

Another important aspect of model-based workflows is the opportunity to consider different scales at the same time. From an environmental point of view, this approach allows for the revision of microclimates (Hershcovich, Hout, Rinsky, Laufer, & Grobman, 2017) operating at lower scales in a global environmental regime.

Because fabrication machines in a computer-aided manufacturing scenario are numerically controlled by computers, it is possible to translate geometric information such as patterns directly into fabrication orders such as robotic toolpaths (Matiz, McMenomy, & Erdine, 2019), making possible the flow of information from geometry to simulation to manufacturing.

From this review, it is clear that current design workflows do deal with some aspects of the design problem, and we can see the viability of an integral workflow connecting a computational framework, an environmental performance evaluator, and a fabrication method.

3. Immersive Case Studies

Under the umbrella of immersive case studies as a research methodology, three design experiments were conducted to test variations in the design-to-fabrication workflow (Figure 2), approaching the flow of information through both top-down and bottom-up processes. Independent variables such as materials, environmental conditions, and structural behavior were taken as a framework in which workflow instances were generated based on dependent variables such as geometry, orientation, and assembly logic. Each design experiment explored a specific material system as a means of fabrication and as a manufacturing constraint.

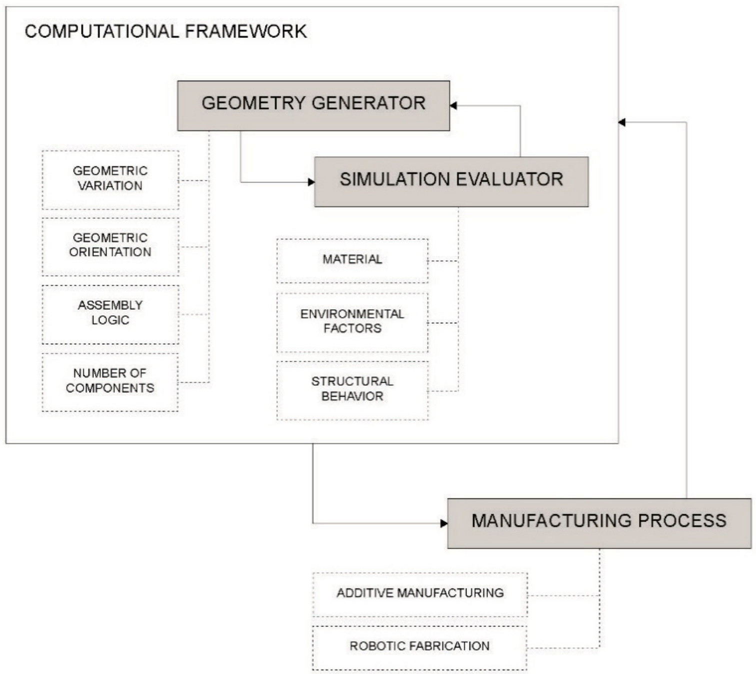

Figure 2

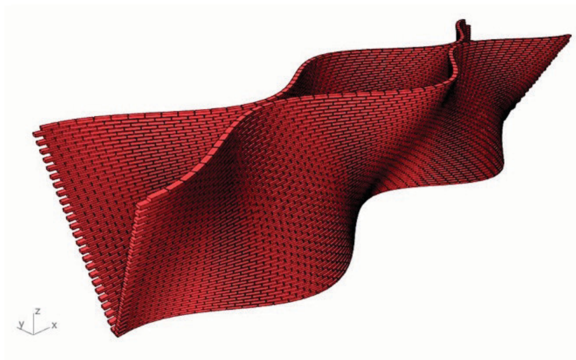

Figure 3

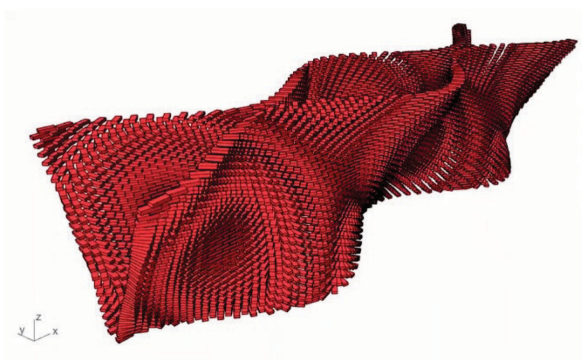

Figure 4

3.1 Design Experiment: Venturi Wall

The first case study was carried out as part of the design of the new Mzuni Library, a collaboration between Mzuzu University in Malawi and the Center for Design Research in the School of Architecture + Design at Virginia Tech. Located in Mzuzu, Malawi, the library embodies both tradition and technology and serves as an icon of the past and a beacon for the future.

3.1.1 Aim

Based on locally sourced bricks, a two-layer facade system was proposed as part of the concept design of the library. A geometric computational framework was developed that is capable of taking both local variables, such as the width, length, height of bricks and global variables such as the depth, height, sinuosity of the facade. Figures 3 and 4 show a range of rotational patterns.

The passive ventilation strategy for the library needed to be based on several cooperative principles. One potential strategy for the north facade could be based on a principle called the "Venturi effect," which causes a laminar airflow to accelerate when constricted to pass through an opening, as the same volume of air must now pass through a smaller area. If the constriction is abrupt enough to create turbulence, the

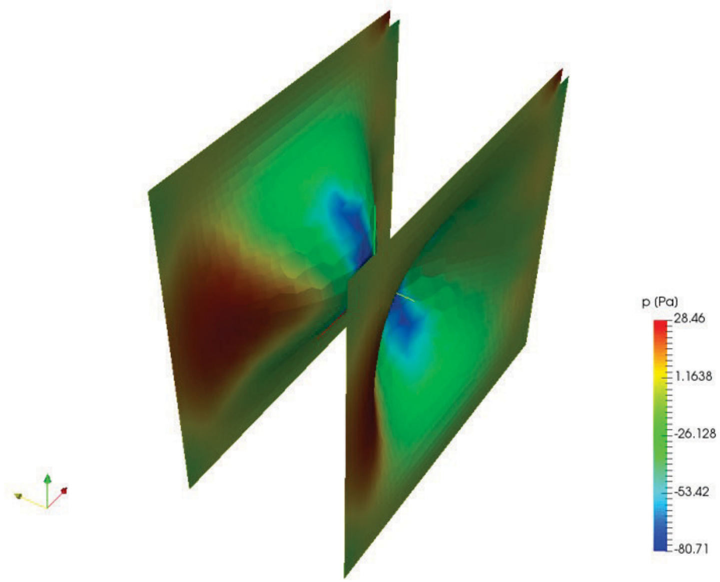

Figure 5

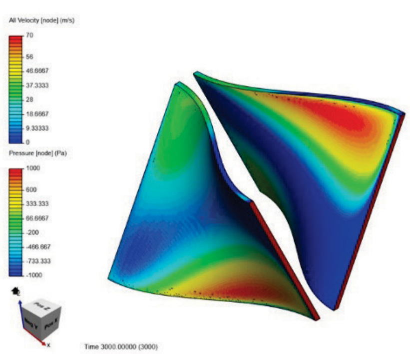

Figure 6

Venturi acceleration is minimized. Figure 5 shows the difference in velocity caused by the constriction by showing surfaces zones of positive pressure in red and zones of negative pressure, which can be used as extraction zones, in blue.

Orienting facade components toward the predominant wind can allow them to compress the incoming air, creating suction points due to differences in air velocity. The warm air is vented through these suction openings where the pressure is negative, resulting in cooler fresh air being pulled into the building from the outside.

3.1.2 Environmental Analysis

To link the computational framework with the environmental performance of a facade, several computational simulations were conducted to increment performance by sequentially adjusting the forms of the facade components.

3.1.2.1 Computer Fluid Dynamics

Starting with a single component, computer simulations of fluid dynamics scenarios were carried out to identify the surface zones of negative and positive pressure occurring in the model (Figure 6). The simulation was done on the cloud-based software SimScale, which is based on OpenFOAM. Being cloud-based, it allows large simulations to be carried out in a couple of hours because it is scalable in terms of power computing. The cloud-based nature is also helpful for preparing the geometry for simulation, and tasks like meshing and domain specification are simplified.

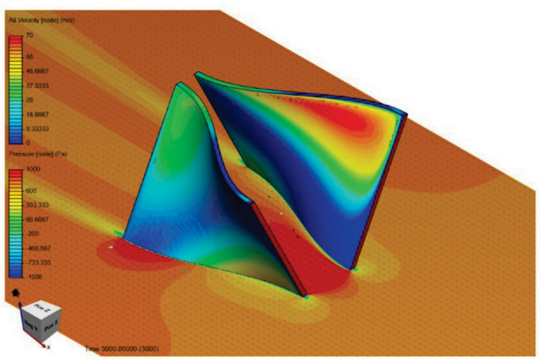

Figure 7

Because the purpose of the simulation process was to identify performance patterns, the initial conditions were purposely set higher than natural conditions. As expected, and as shown in Figures 7 and 8, negative pressure zones are localized where component layers are closer, effectively compressing air and increasing its velocity. Figure 9 shows that no turbulent flow is produced due to the smooth geometrical variations on the component flow of air remaining laminar.

Because the component strategy for facades consists of repetitions of the same component, it follows that environmental behavior at

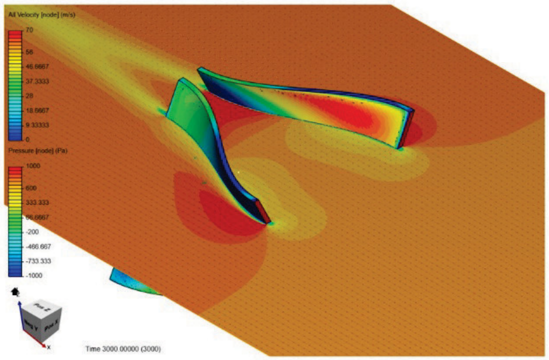

Figure 8

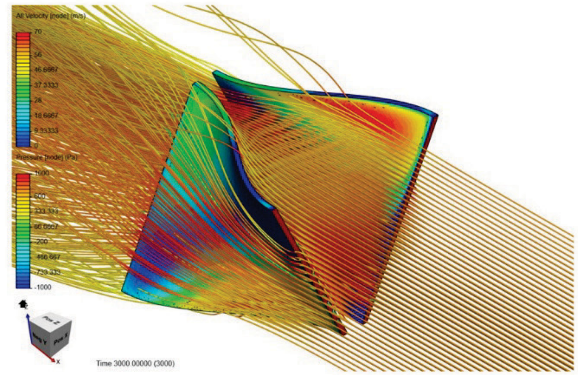

Figure 9

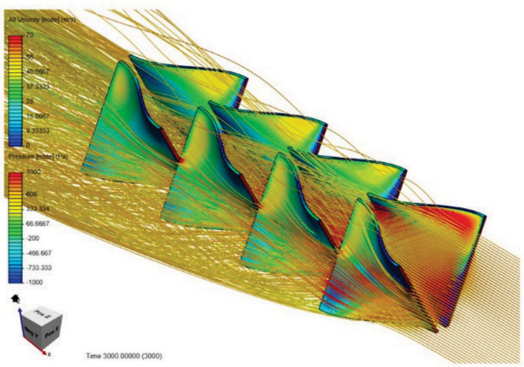

Figure 10

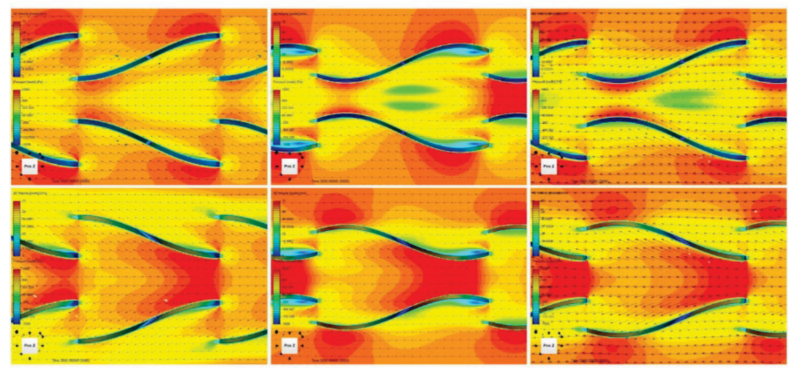

Figure 11

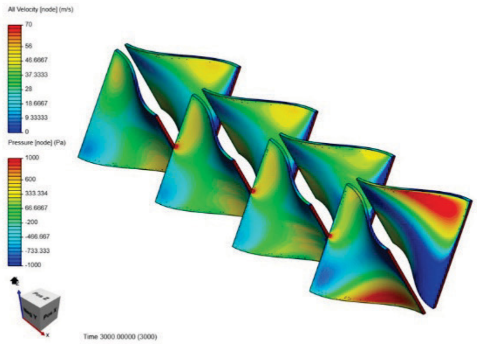

Figure 12

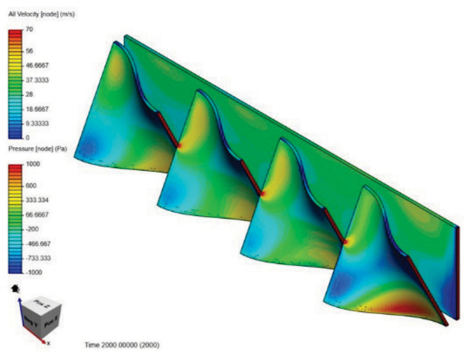

Figure 13

the component scale can be replicated at the facade scale (Figure 10). It is important to notice that geometric variations at the component scale will have effects at the facade scale. Figure 11 compares variations in component length; the middle and right columns show the development of turbulent flow as a consequence of this geometric change.

A systematic trial-and-error process led to the working length being defined at a component scale that can maintain laminar flow, which is important for not minimizing the Venturi acceleration of air at the facade scale (Figure 12). The fact that the workflow is model-based and the simulation computer-based makes this process feasible.

Two important decisions were made in the design development. The first has to do with spatial functionality; because of this, the inner layer of the component was flattened (Figure 13) to facilitate the introduction of inlet and outlet fenestrations connected to the internal space of the building. Figure 14 shows the same performative pattern as in the double-curved component.

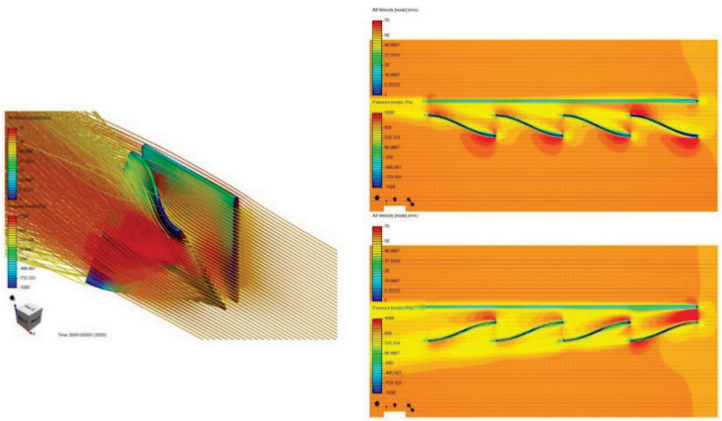

Figure 14

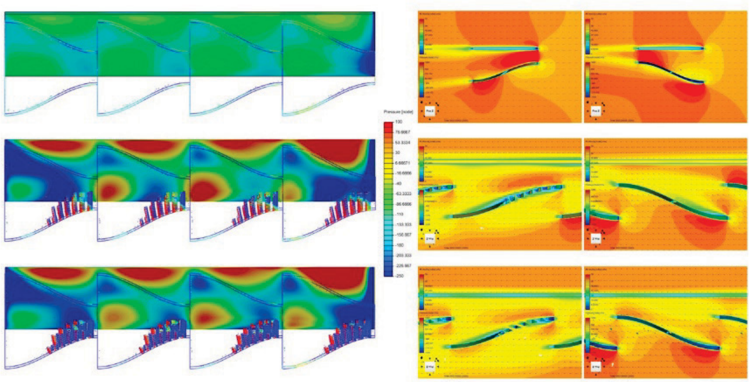

Figure 15

The second decision exploits the built-in rotational patterns of the computational framework by rotating bricks toward the prevailing wind on the positive pressure zone of the outer component layer to increase the inlet contribution of the whole component through gaps between rotated bricks. Figure 15 shows components with no rotated bricks on the top row, with bricks rotated perpendicular to the surface on the middle row, and with bricks rotated progressively toward the prevailing wind direction in the bottom row. Although the middle and bottom cases increased the global inlet contribution, the middle case also developed turbulent flow in the gaps between bricks.

3.1.2.2 Thermal Mass

Built-in sinusoidal rotational patterns of bricks were also explored as a way to increase the global surface area exposed to direct solar radiation, which will affect the thermal mass properties of the bricks (Figure 16). The objective was to implement controlled, differentiated temperature zones within the facade.

Computer simulations based on Ladybug, a plugin for the Rhinoceros computer-aided design software, were employed to measure grid-based

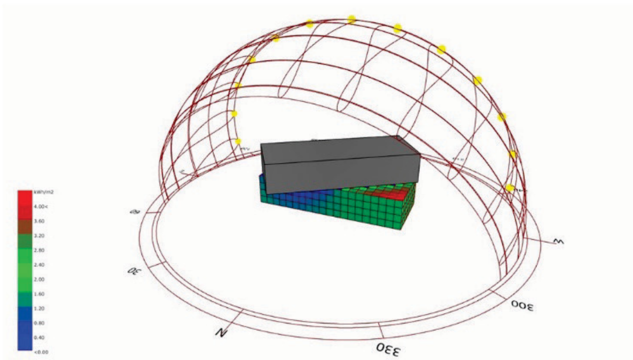

Figure 16

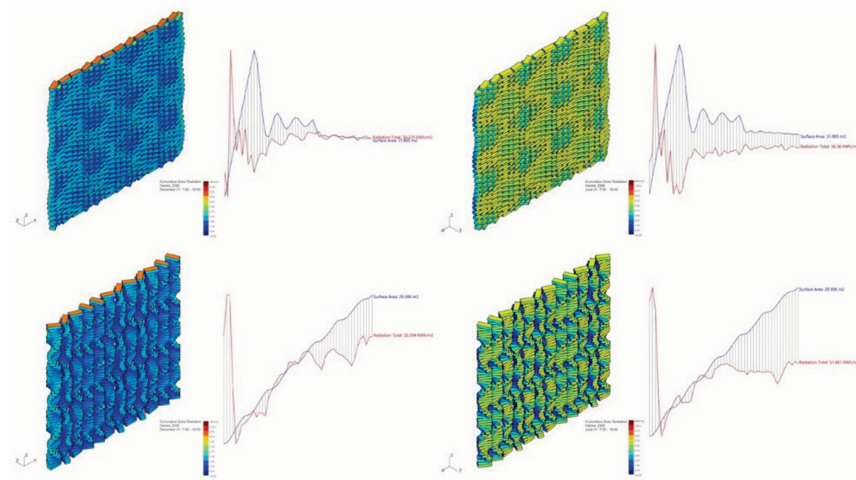

Figure 17

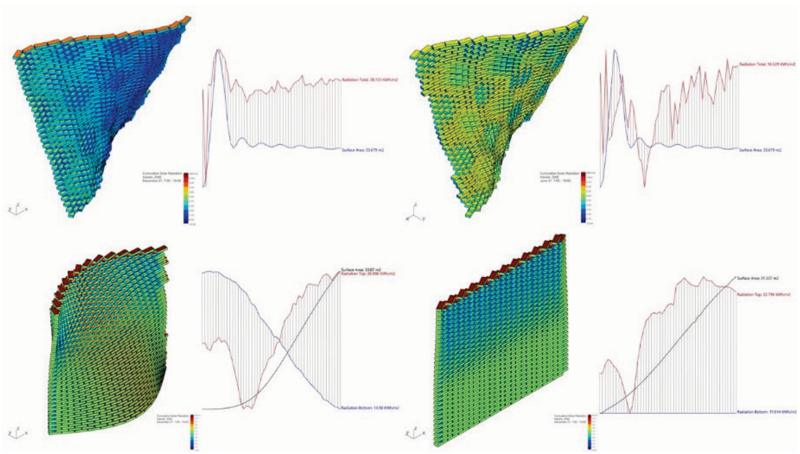

Figure 18

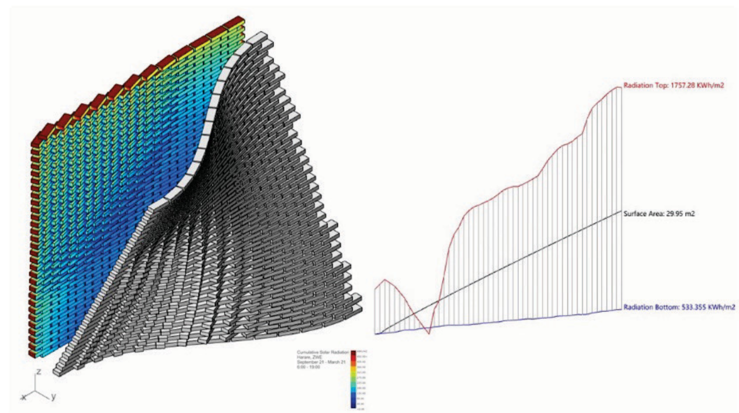

Figure 19

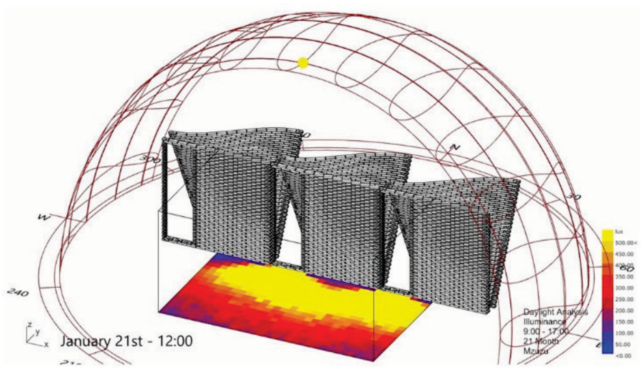

Figure 20

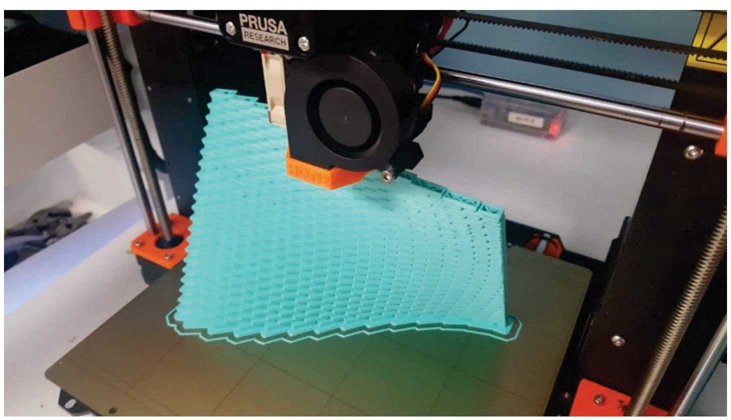

Figure 21

direct solar radiation over several time ranges. Because Ladybug uses weather files to specify geographic location conditions, and no such files are available for Malawi, a weather file for Zimbabwe, with the same latitude close to the equator, was used instead.

The computational framework was developed in Rhinoceros. In the flow of geometric information for performance evaluation, no preprocessing was needed, and custom Python scripts were developed to derive the analysis grid from brick geometry.

Figure 17 shows a range of incremental brick rotations from 0° to 90°, in stretcher and stack organizations and on flat and curved walls. The line graphs plot the relationship between the cumulative values of solar incident radiation (red) and global surface area (blue).

On the basis of this first simulation set, it was decided to divide the wall into two temperature zones (bottom and top halves) to improve air circulation through the stack effect. Figure 18 shows the rotation of bricks on top of the wall, from 0° to 90°, and the line graphs plot the relationship between the cumulative values of solar incident radiation on the top and bottom of the wall (red and blue) and the global surface

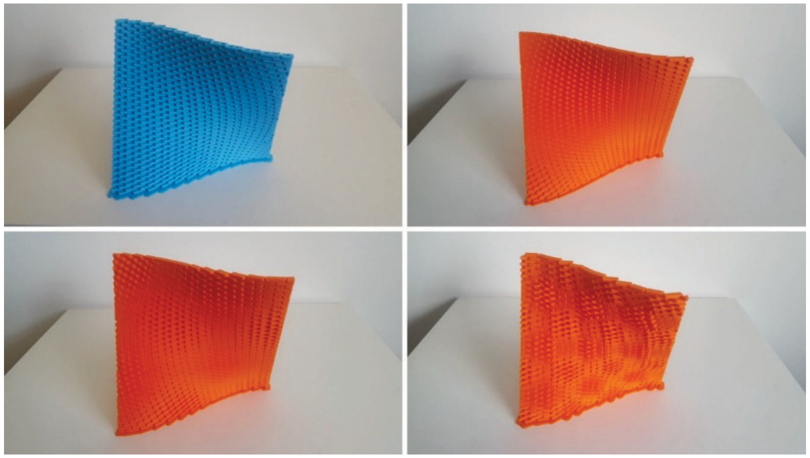

Figure 22

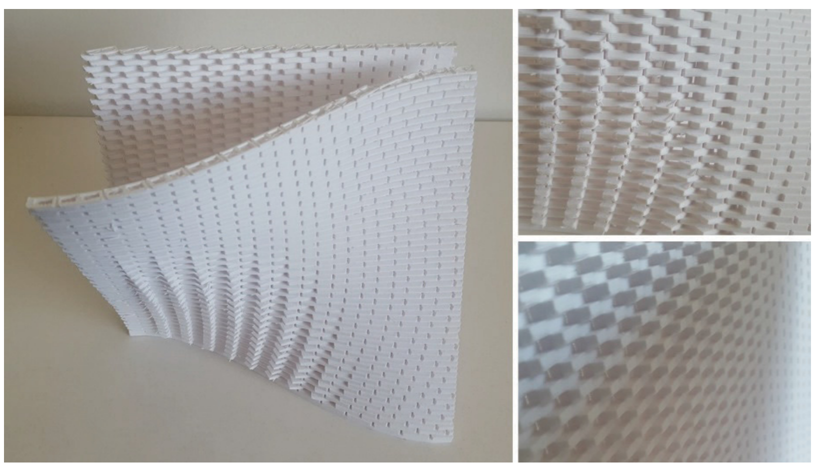

Figure 23

area (black). Figure 19 shows the final two-layer component and the associated performance graph.

3.1.2.3 Daylight

As the facade is part of a larger library project, a last computer simulation was carried out to ensure that at least 500 lux fall on work tables during working hours throughout the year. A version of Radiance running inside Rhinoceros was used for grid-based daylight simulation. Figure 20 shows a section of the facade and the components with fenestrations where windows are located. The side walls are shown on the diagram as wireframe.

3.1.3 Manufacturing

For fabrication of instances produced by the workflow, two computer-aided manufacturing methods were explored that were directly translated from the computational framework and the simulation evaluation, the first at model scale and the second at prototype scale but with the capacity to be translated at a 1:1 architectural scale.

3.1.3.1 Additive Manufacturing

A set of 3D printed models was produced using PLA filament (Figure 21) on a Lulzbot TAZ 5 desktop 3D printer. In Figure 22, 3D printed

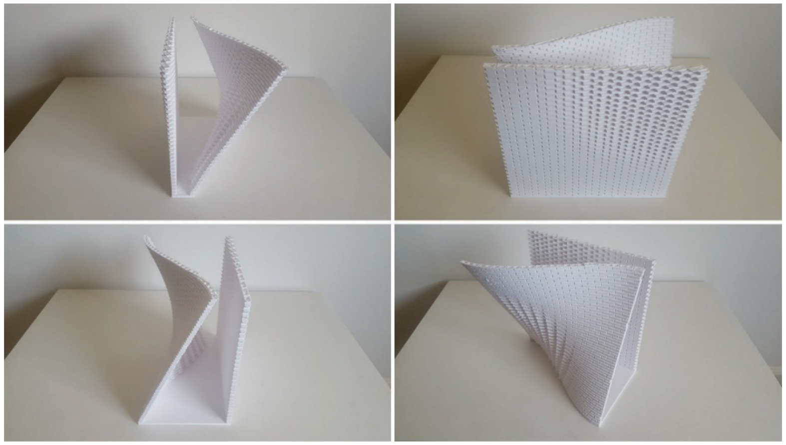

Figure 24

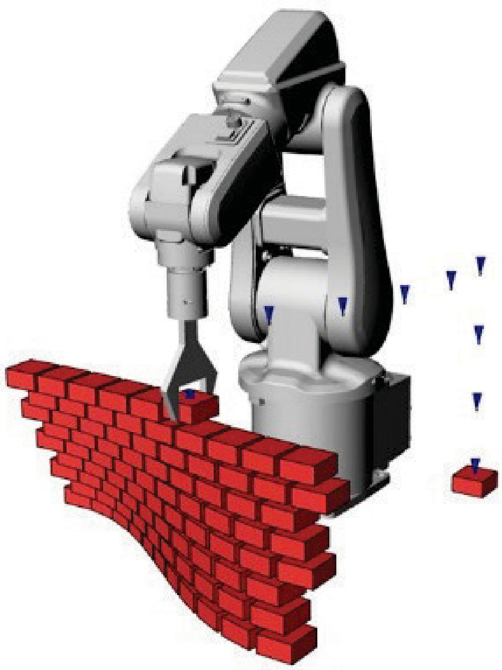

Figure 25

models show a range of brick rotation patterns. Figure 23 shows the final two-layer facade component.

Because the workflow is model-based, producing the STL files required for fabrication does not involve additional steps. Solid geometric entities used to describe bricks in the CAD environment can produce a watertight mesh geometry, and the geometric information used for simulation and for fabrication are consistent with each other.

Rotational patterns created several occurrences of bridge conditions (Figure 24), which were dealt with by the filament material tension.

3.1.3.2 Robotic Fabrication

A robotic fabrication workflow was also explored using an ABB IRB 120 robot to pick up, translate, and orient blocks, using a gripper as a tool head. Wooden blocks were used instead of actual bricks due to the load bearing and reach limitations of the robot, so scale prototypes (Figure 25) of the actual components were fabricated.

A custom Visual Basic script running inside Grasshopper for Rhinoceros was developed to automate the generation of toolpaths for the

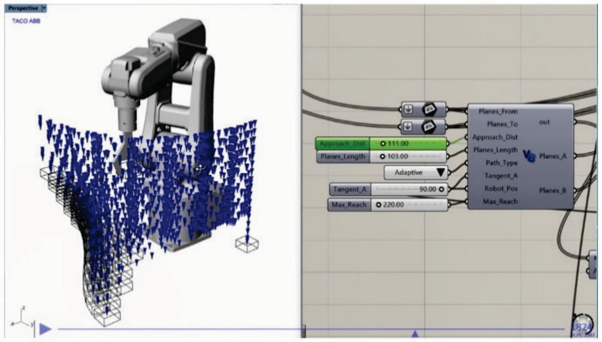

Figure 26

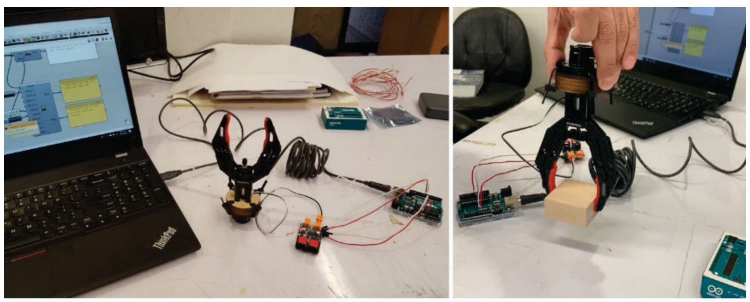

Figure 27

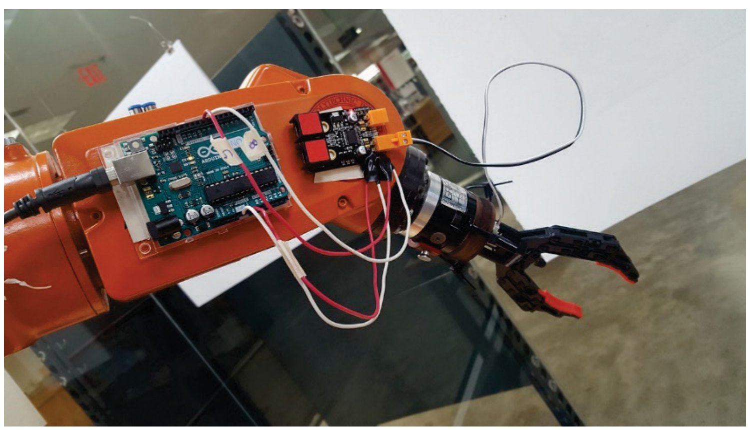

Figure 28

robot, and a custom simulation environment was used along with the Grasshopper plugin TACO to generate gcode for controlling the robot. The custom script could work out variables such as approach distance and toolpath type (Figure 26) in order to generate smooth interpolated planes along the toolpath. Quaternions were used instead of Euler angles, as they could also discretize the planes that opened the gripper and the planes that closed it. The toolpath type variable implements a couple of methods for avoiding out-of-axial range, improving speed, and avoiding robot singularities.

Because the industrial robot is open ended, a tool needed to be designed to perform the required actions. For this fabrication workflow, an electric gripper controlled by a microprocessor was tested (Figure 27) and mounted to the robot (Figure 28). Figure 29 shows the robotic fabrication process of the prototype from wooden blocks.

3.1.4 Results and Findings

A two-layer facade component was developed as an instance of the proposed workflow (Figure 31). It induces natural ventilation by implementing extraction zones for exhaust air using Venturi acceleration and increases inlet zones using rotational patterns on the outer layer. The separation between the layers is related to the sun angle at the geographic location. It lets the sun hit the inner layer, increasing the surface area exposed to direct-incident solar radiation, via the rotational patterns on the top area of the inner layer, resulting in a temperature difference between the bottom and top of the inner layer, which improves air circulation through the stack effect.

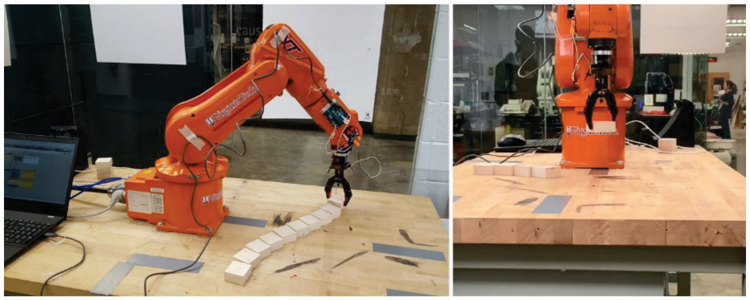

Figure 29

Even though the robotic fabrication scenario explored here is intended for the production of a scaled prototype, the robotic fabrication workflow and computational tools developed can be transferred to one-to-one architectural-scale component fabrication (Figure 30).

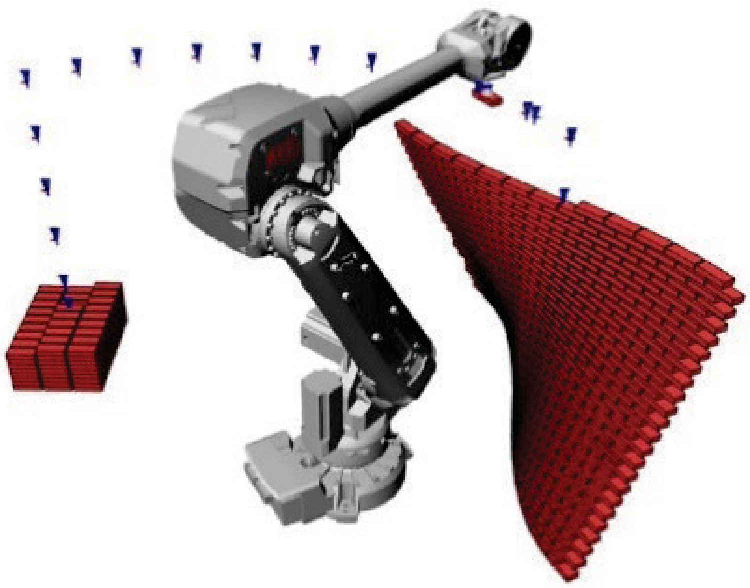

Figure 30

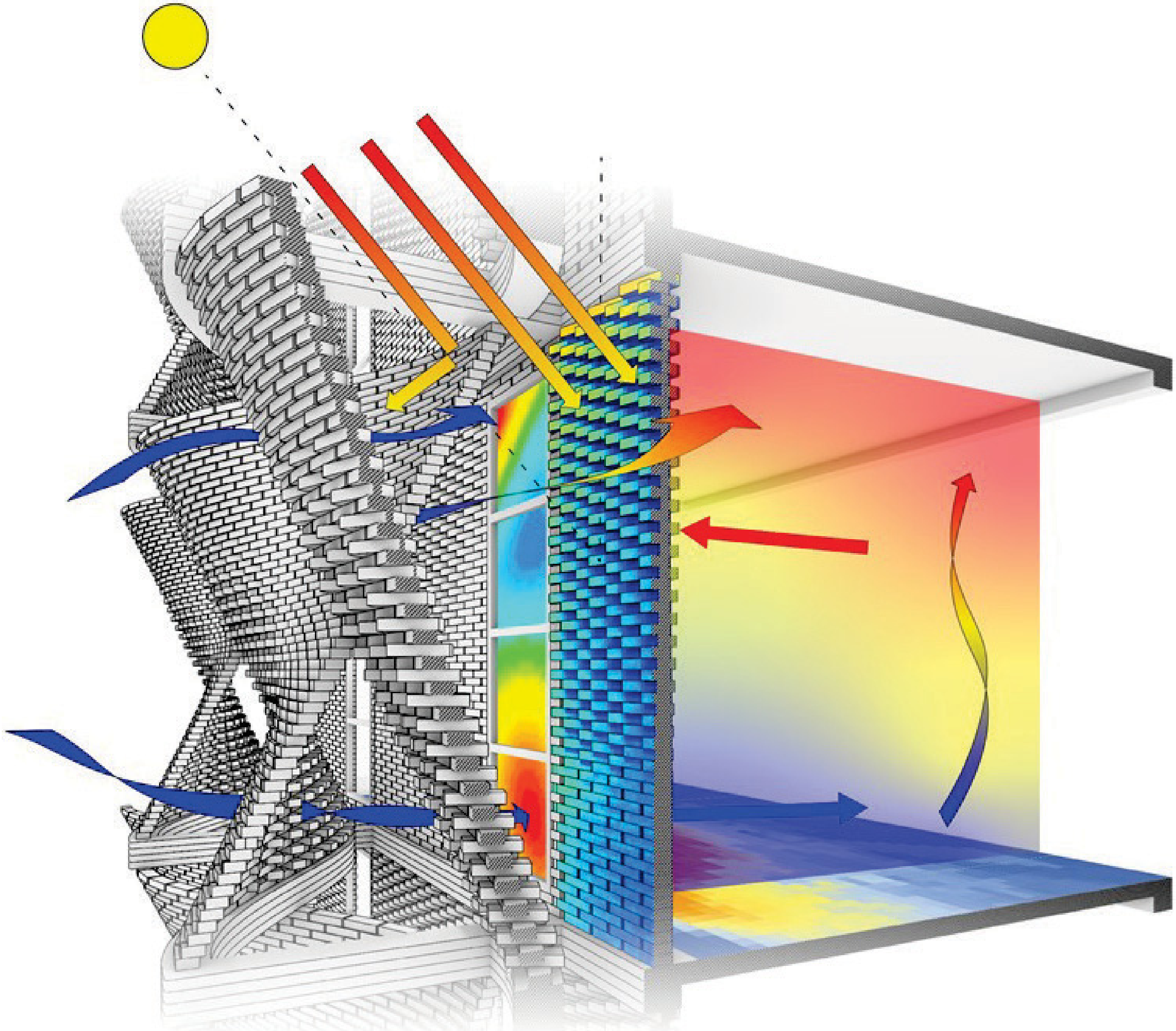

Figure 31

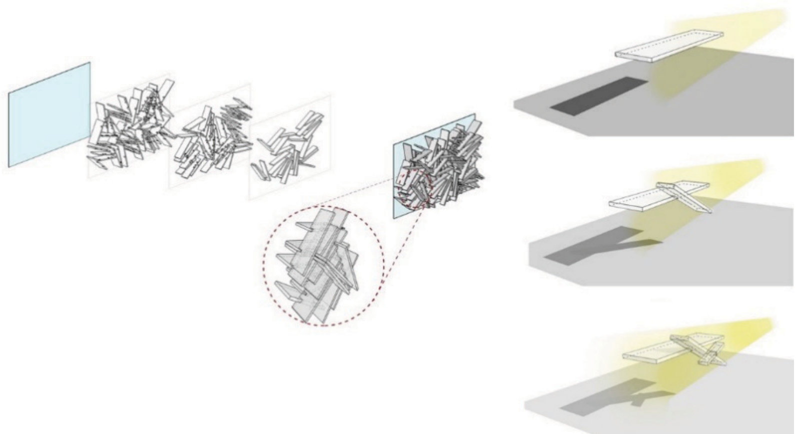

Figure 32

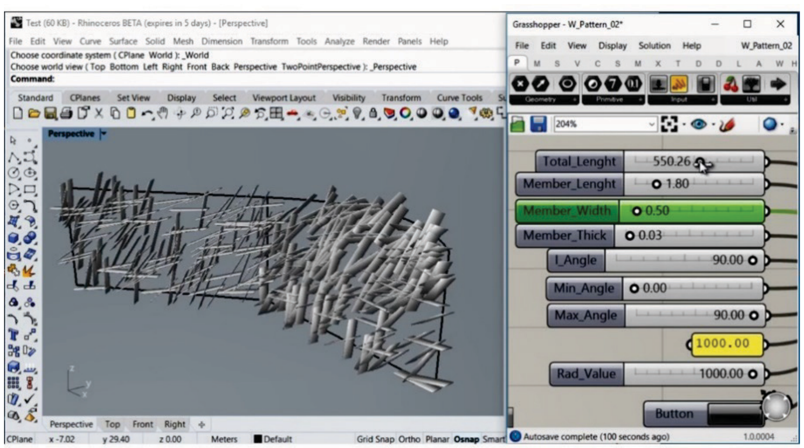

Figure 33

3.2 Design Experiment: Timber Façade

The second case study was developed as part of the Eco-Park Learning Center project, a collaboration between the Prince William County Solid Waste Division and the Center for Design Research in the School of Architecture + Design at Virginia Tech. Visitors to the Eco-Park Learning Center are taught about a range of alternative energy sources, including solar, wind, and methane.

3.2.1 Aim

The shading screen for the PWC project was conceived of as a way to reduce buildings' solar exposure by means of a facade shading device

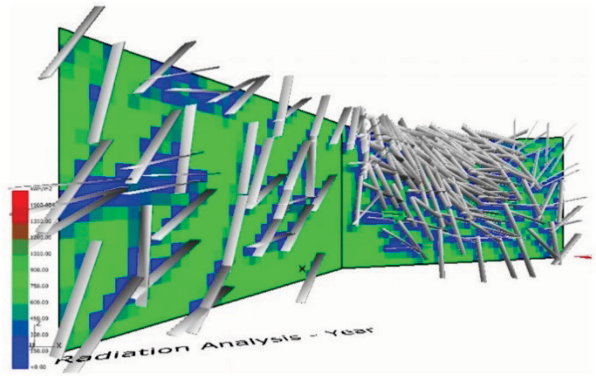

Figure 34

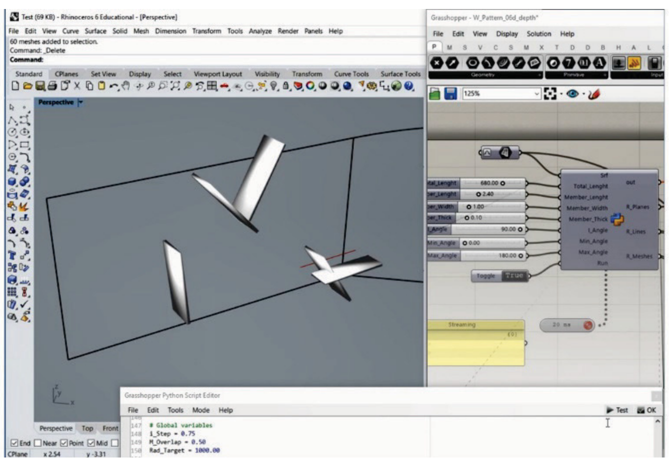

Figure 35

made from the recycled wood commonly used in construction scaffolding (Figure 32).

The aim in the first part of the research process is to develop a computational framework that can produce instances of a geometric system informed by material constraints and environmental considerations. The aim of the second part is to explore the fabrication of the system's assembly logic based on notches, to understand the limitations of the fabrication and the opportunities for a robotic mill fabrication workflow to produce wood joints, and to inform the construction of the system as a fabrication constraint.

3.2.2 Computational Framework

Within the computer-aided design environment Rhinoceros, a custom Python script was developed that could generate components through a bottom-up, additive process in which the final morphology is a result of the initial-condition rules.

Figure 33 shows all the variables implemented to control the geometry of the screen. The addition of each wood member is linked to a yearly direct-incidence solar radiation simulation based on the Ladybug plugin (Figure 34). The algorithm implemented in the script places a

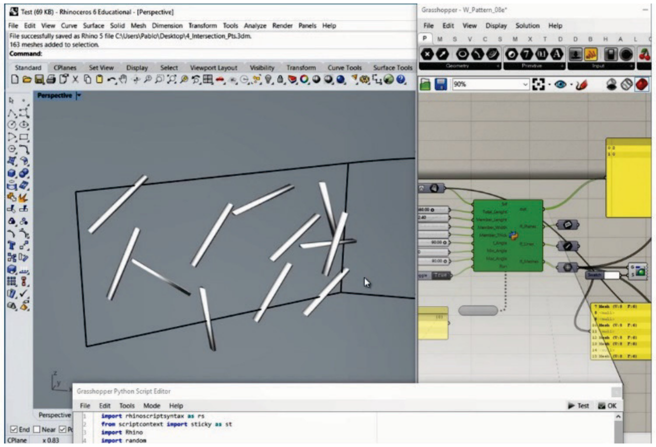

Figure 36

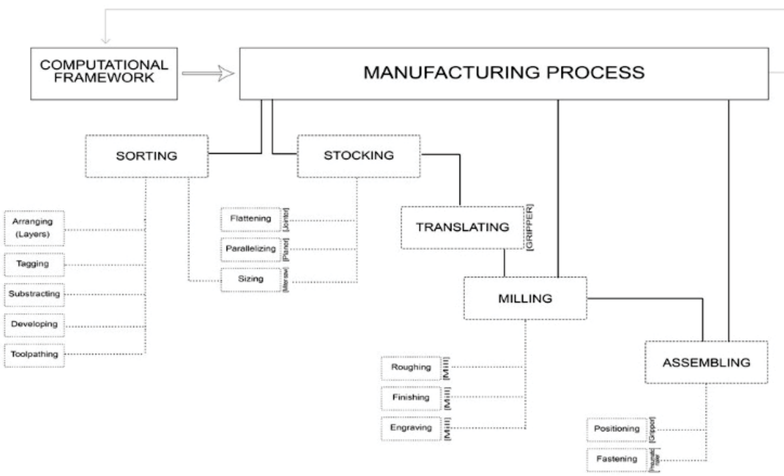

Figure 37

new member at each point the simulation determines to have the highest exposure to the sun, reducing the global facade exposure.

Figure 35 shows the assembly logic of the system. Notches are added between intersecting members so that every time a new member intersects an older one, it jumps a layer outward by a proportion of its width and a scale-down variable. This controls the cross-section of every new layer. The potential here is for the fade-out the facade as it continues adding layers by cutting the original wood piece in halves, quarters, and so forth. For the studied prototypes, the same size of cross-sections was used in all facade layers.

It is also possible to specify as a variable in the computational framework the minimum number of intersections before a new member jumps out a layer. Figure 36 shows case of two intersections.

3.2.3 Manufacturing

The fabrication workflow involves several manufacturing processes, as shown in Figure 37. To test the workflow, a mockup instance of the computational framework was developed for fabrication.

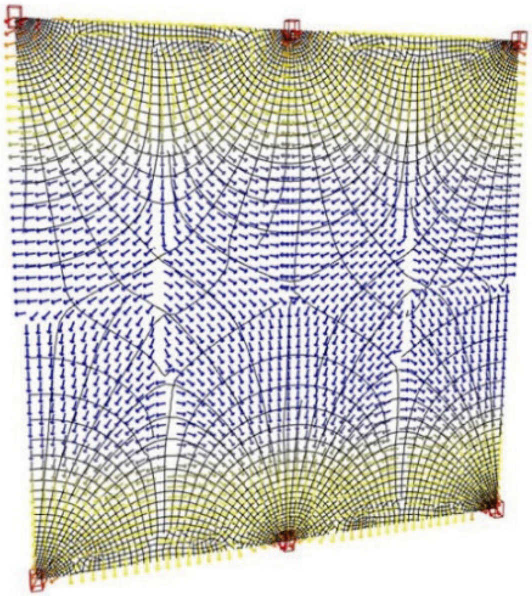

Figure 38

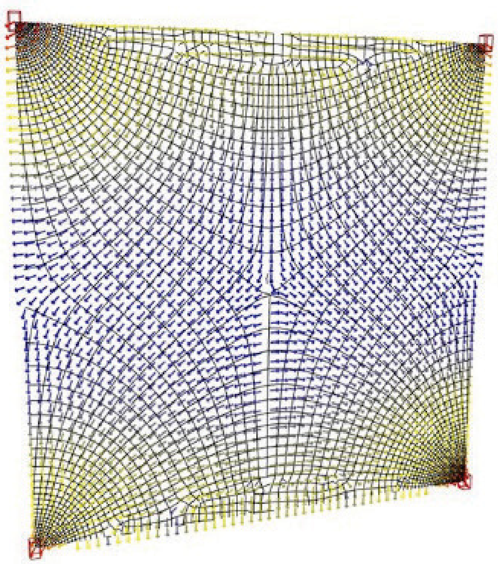

Figure 39

The generation process was informed using the direct solar radiation on the position for the components, and using computational finite-element structural analysis to determine the stress lines and align the angles of each member.

Figure 38 and Figure 39 show different support cases, displayed as wire boxes, which generate different stress line patterns. Members are aligned to these patterns so as to provide material continuity (Figure 40). The mockup screen is 8′ × 8′ and composed of three layers of members with 2′ × 4′ cross sections and a length of 3 feet, responding to spring-summer solar radiation.

A sorting procedure in the form of a Python script was developed that arranges layers and parts and also tags, subtracts, and develops the members generated by the computational framework (Figure 41). A second custom Python script takes the output of the sorting procedure and automatically generates toolpaths for the robot (Figure 42).

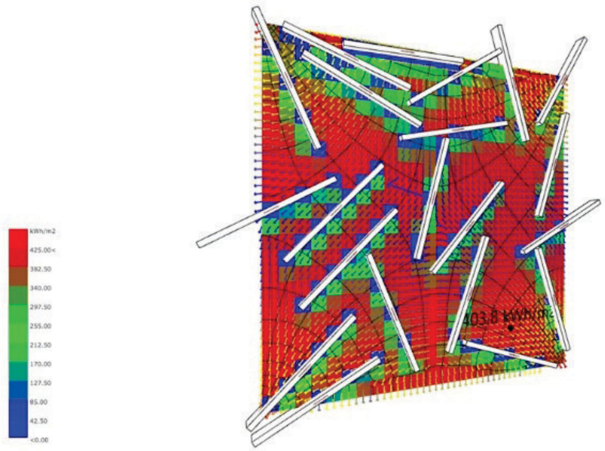

Figure 40

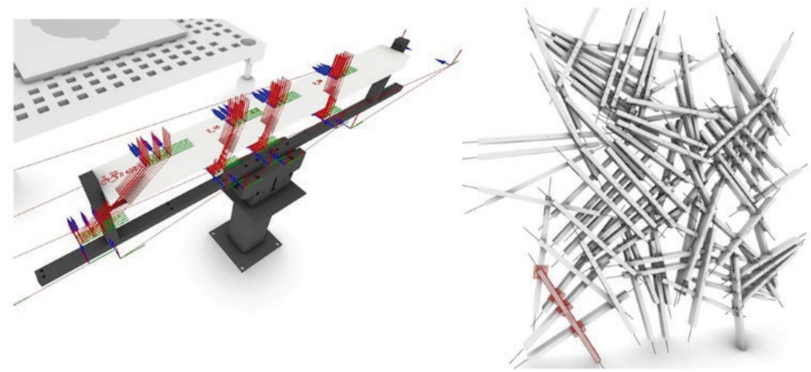

Figure 41

Figure 42

Figure 43

3.2.3.1 Robotic Fabrication

The fabrication of the mockup made from 2′ ×4′ pine members took place in Boston as part of a residency at the Autodesk BUILD Space (Figure 43), a research and development workspace focused on innovation in architecture, engineering, and construction. The residency period was divided into two parts, one to develop, produce, and test tools for the fabrication process, and one for the actual fabrication of the complete prototype.

Figure 44

A manufacturing cell composed of an ABB robot model IRB 4600 with a Spindle tool and a safety guard was used for the fabrication (Fig 44). Because the industrial robot is a versatile machine, it is open-ended, meaning that in every fabrication project involving robots, all the tools needed must be designed and fabricated and from a design point of view. New fabrication skills must also be acquired, from operating advanced CNC equipment to precision machining. In the case of the fabrication of the mockup, the focus was on the work holding.

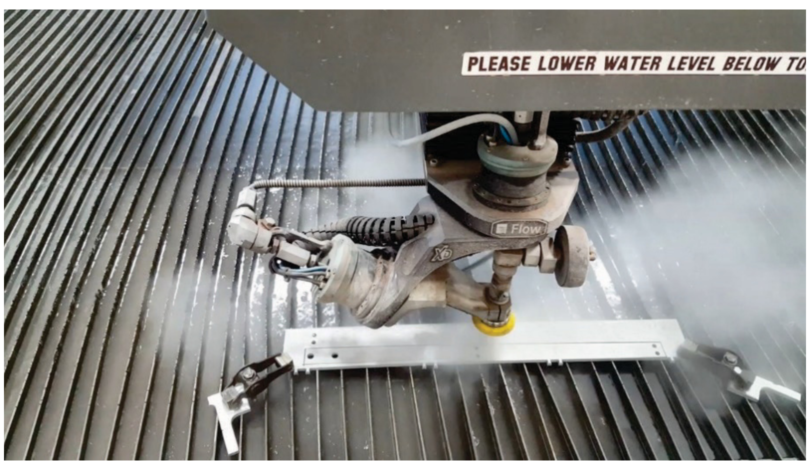

Figure 45

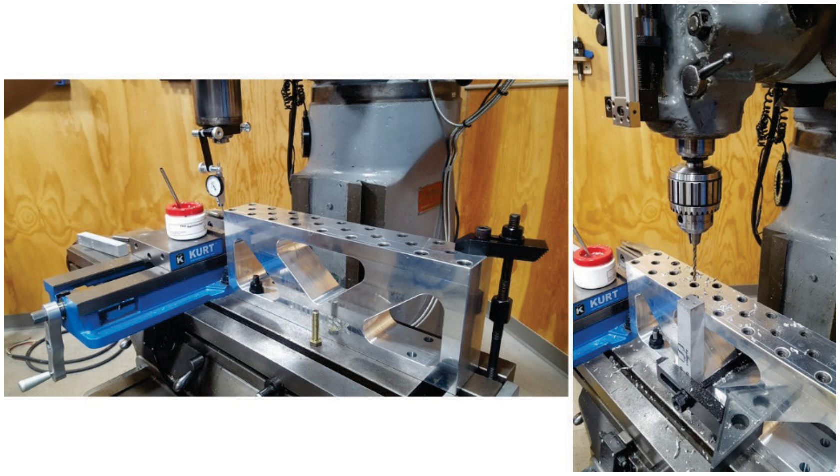

Figure 46

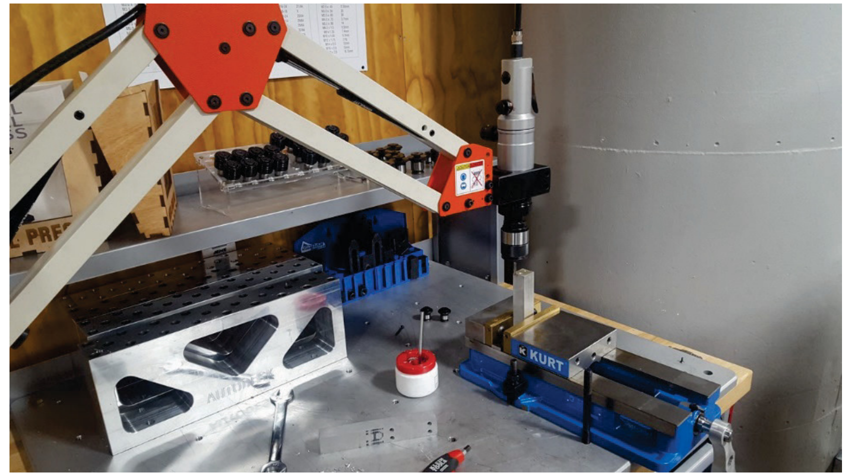

Figure 47

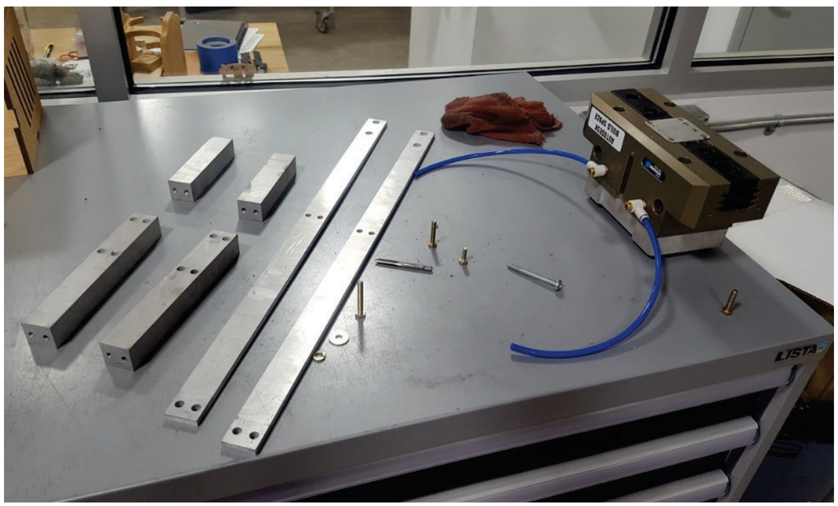

Figure 48

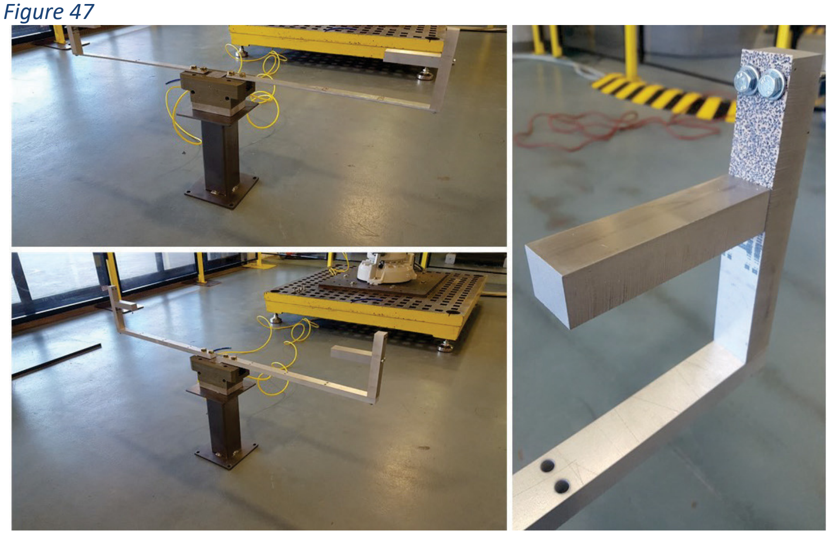

Figure 49

Figure 50

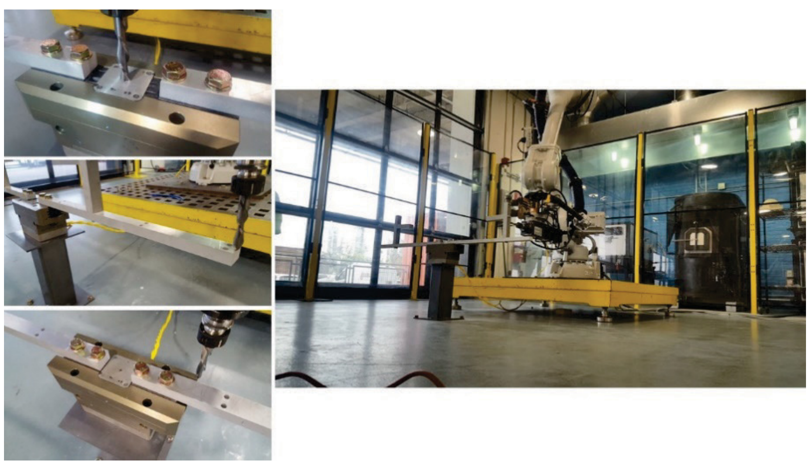

Figure 51

Fingers to hold the wooden pieces were designed and produced based on a pneumatic gripper. Parts were waterjet-cut from ¼-inch aluminum plate (Figure 45) and drilled using a Bridgeport (Figure 46). This is an especially critical step for tolerances, so calibration equipment was used to ensure perpendicularity between the gripper's component parts. Finally, the parts were tapped using a Haas machine (Figure 47). Figure 48 shows the custom-made parts that compose the work holding.

After the work holding was assembled (Figure 49), two problems were detected. The longest aluminum parts were too thin for their length and tended to vibrate when forces were applied to their extremes. These were replaced with ¾-inch aluminum parts. The second problem had to do with the grip capacity of the fingers contacting the wood pieces. When working on naked aluminum, the wooden pieces tended to slip, so low-grain sandpaper was added as a surface contact (Figure 50).

Because the floor of the manufacturing cell was not perfectly horizontal, a calibration procedure was established to read the inclination of the work holding in relation to the robot (Figure 51). The toolpath-generator script has the flexibility to read test coordinate

Figure 52

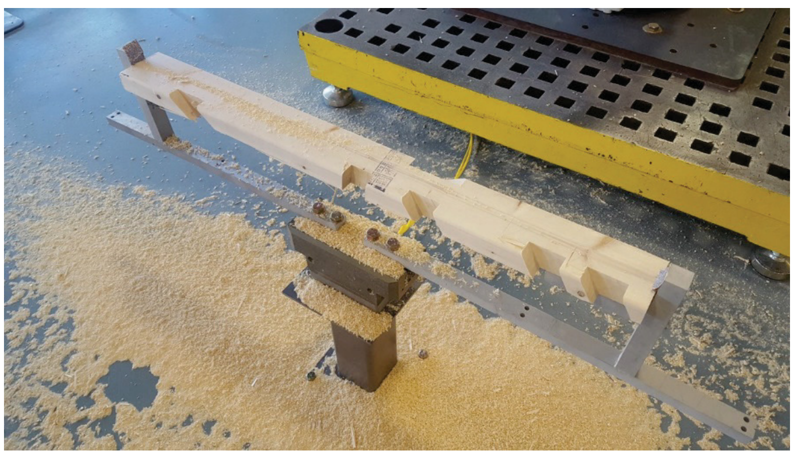

Figure 53

points taken from the physical gripper, so the toolpaths generated for the robot can deal with this discrepancy between the digital and physical worlds.

Figure 52 shows the milling process, and Figure 53 shows the final wooden piece after milling. To aid in the assembly process, information generated by the sorting script—such as component name, intersecting component at each notch, and direct-incidence solar radiation value at the time of the component being added—was engraved on every component (Figure 54) using a laser cutter machine.

The mockup was assembled in Blacksburg at the woodshop facilities of Virginia Tech (Figures 55–57). While the notch logic helped secure every piece in place, 4½-inch structural screws were used to fasten them locally using power screwdrivers (Figures 58). Figures 59 and 63 shows the completed mockup.

3.2.4 Results and Findings

As Figure 61 shows, the computational framework plus the simulation evaluator can reduce the facade solar exposure in the PWC project from 890.76 Kwh/m2 to 369.32 Kwh/m2 by adding a screen on the southeast-facing facade.


Figure 54


Figure 55

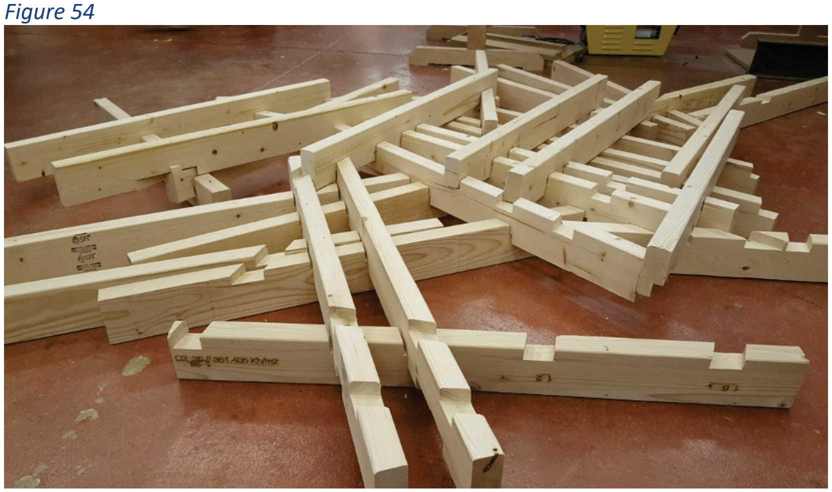
Figure 56

Figure 57

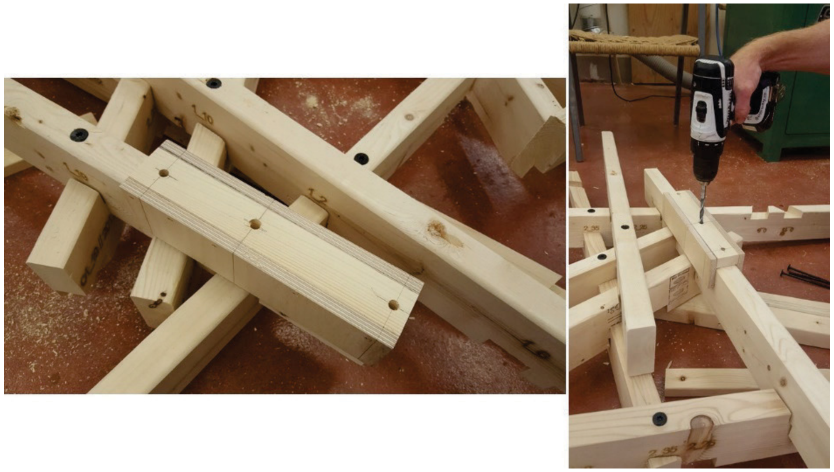

Figure 58

Figure 59

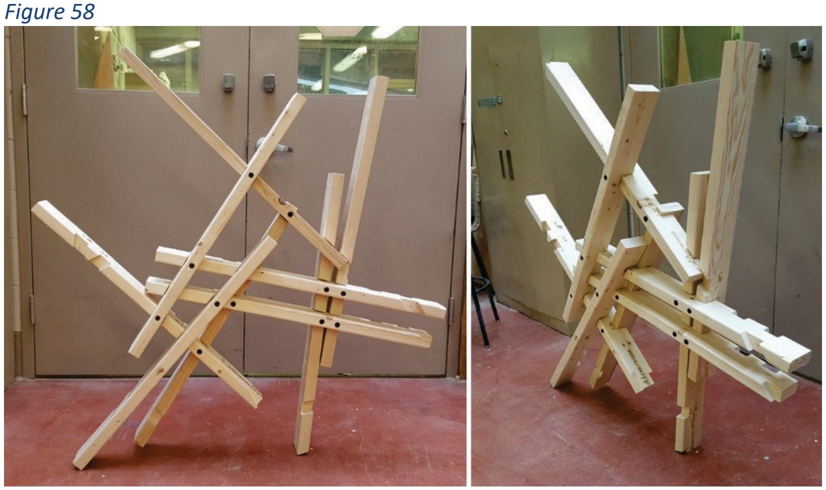

Figure 60

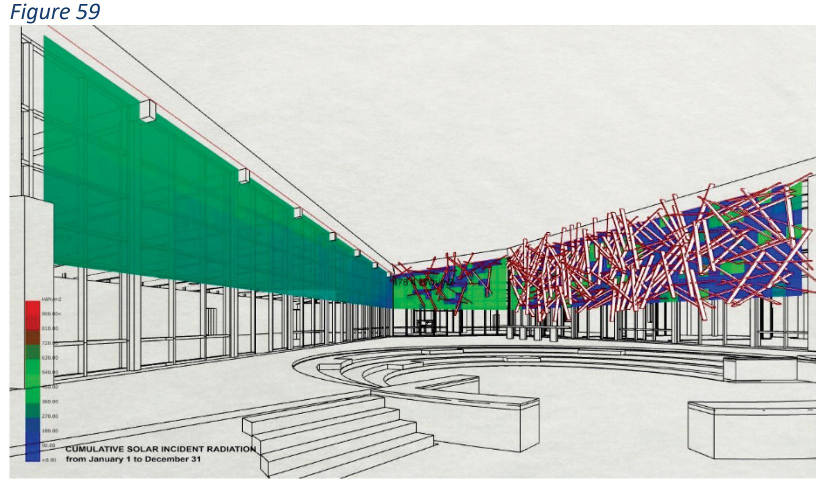

Figure 61

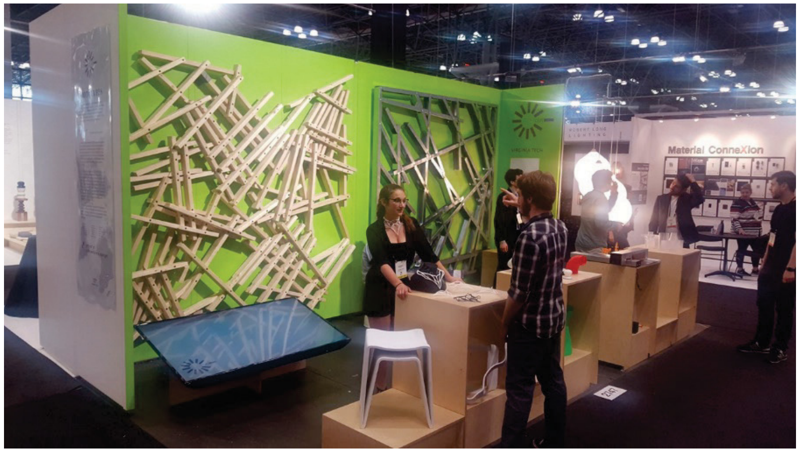

Figure 62

The assembly logic based on notches (Figure 60) works for positioning the components. A tolerance of 2 mm was introduced during the milling process to account for calibration errors and material changes, such as wood swelling due to humidity. For this reason, structural screws were also employed to secure each notch.

As a manufacturing proof of concept, part of the facade was fabricated at the Autodesk BUILD Space in Boston and later displayed at the ICFF exhibition in New York (Figure 62).

Figure 63

Figure 64

3.3. Design Experiment: Metal Façade

A parallel fabrication method was tested to explore the consistency of the workflow from design to production. The outcome of the computational framework and the simulation evaluator was diverted to another fabrication method.

3.3.1. Aim

The objective of this experiment was to test the viability of an alternative fabrication method for the outcome of the computational framework and simulation evaluator used for the shading screen.

3.3.2. Computational Framework

In the computational framework developed for the Timber Façade design experiment, a Python script was hooked up to collect center-line geometric information of the facade components. The number of layers composing the façade was kept at zero. A custom Grasshopper plugin running inside the Rhinoceros CAD environment was then used to export the comma-separated value files, which were inputted into the Howick frame machine. (Figure 64)

Figure 65

Figure 66

3.3.3. Manufacturing

The generated morphology was fabricated using a Howick frame machine, which can bend, cut, and punch out thin metal rolls. Figures 65 and 66 show the assembly process and the final metal piece.

3.3.3.1. Light-Gauge Steel Framing

According to Howick's manufacturer information, steel framing machines place all punching and fixing holes using accurate computer control. This allows the frames to be manufactured with high precision and to be self-locating and jigging. All the frame components are produced quickly, with the location dimples and pre-punched screw and rivet holes ready for assembly and clearly marked. No further cutting or post-processing work is needed, so low-skilled local labor can be used to assemble the buildings with little supervision.

3.3.4. Results and Findings

The viability of an alternative fabrication method was demonstrated (Figure 67). The addition of custom code made it possible to expand the material alternatives in which the outcome of the computational framework could be fabricated.

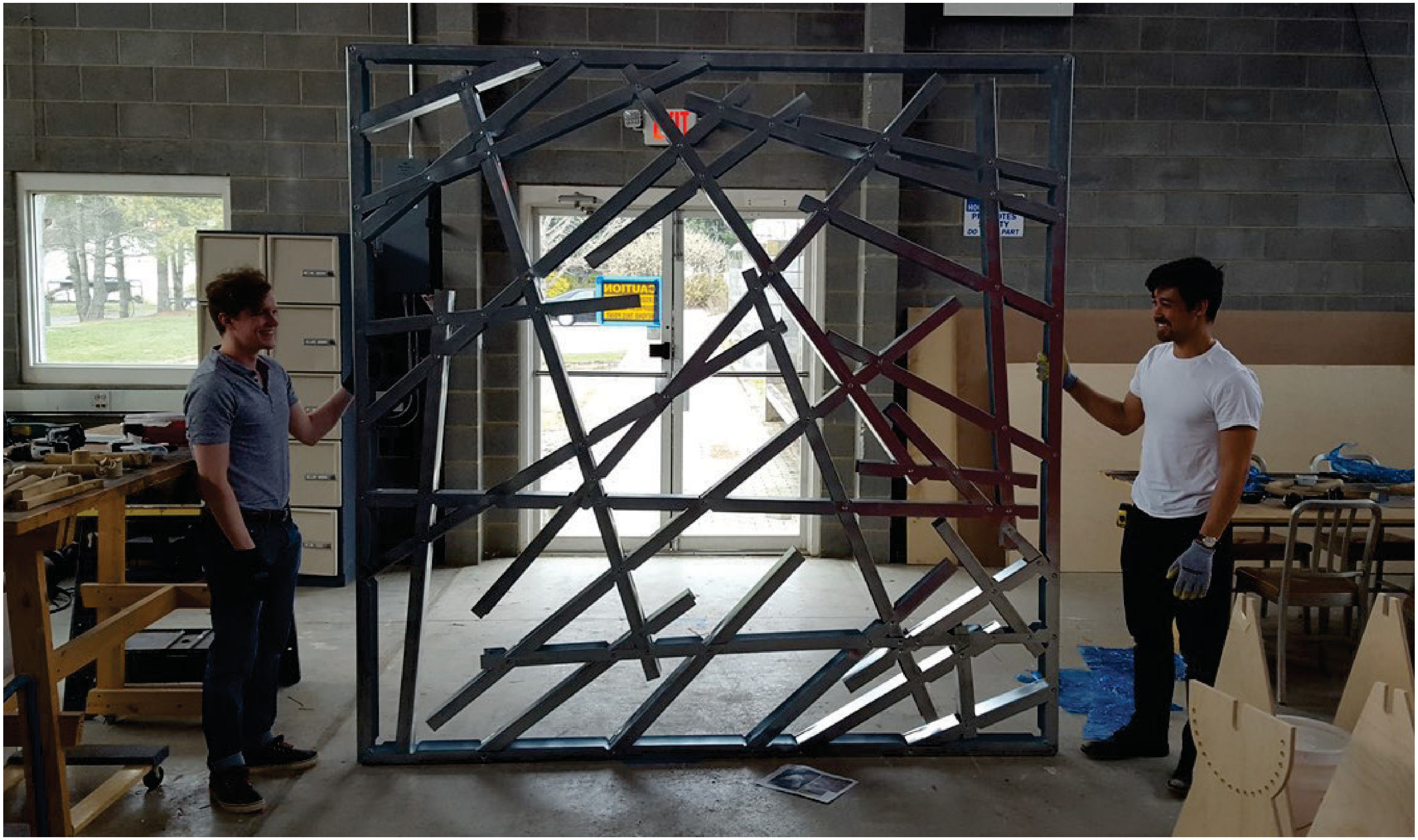

Figure 67

4. Conclusion

From the case studies presented here, is possible to state that a robotics-based fabrication method informed by a multi-variable computational framework and a simulation evaluator integrated into a design-to-fabrication workflow is feasible. Instances of this workflow, such as the Venturi wall and the shading screen for PWC, show a responsiveness to environmental conditions that stems from the logic defined in the workflow.

From a representational point of view, when environmental data are made visible with color-coded diagrams, as in the velocity simulation on the Venturi wall, or with numbers, as in the solar radiation simulation on the shading screen, they can be integrated as a design variable because the designer is objectively aware of their influence in the same way she is aware of a drawing or an area schedule.

From a computational point of view, the use of scripts and subroutines for environmental data processing allow larger data quantities to be considered, such as the yearly solar radiation results used in the shading screen or the pressure values used for each brick on the Venturi wall.

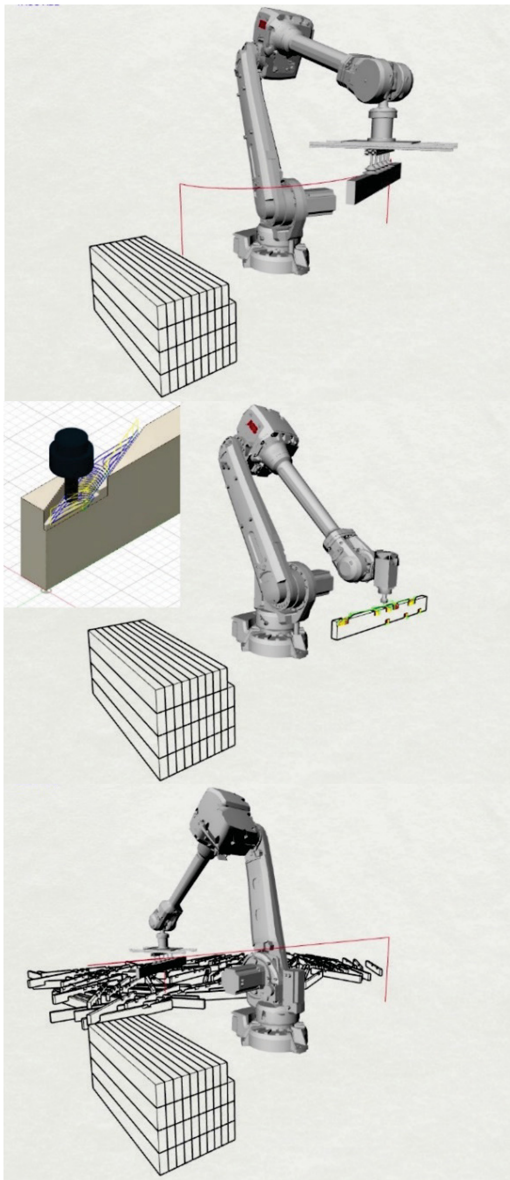

Figure 68

Finally, from a manufacturing point of view, the versatility of industrial robots allows them to be used in a wide range of fabrication scenarios. The definition of a tool through the fabrication process is what gives it specificity, and complex forms can be fabricated with a well-designed robot tool.

5. Future Research

One promising line of research would be the consolidation of separate manual processes (Figure 37) into a comprehensive robotic fabrication workflow, in which a continuous robotic process manufactures the façade, from material sourcing to assembly (Figure 68).

Integrated design-to-fabrication workflows can also help the building industry become more energy efficient and climate friendly on two different scales. On a material scale, embodied carbon emissions can be reduced by the use of recycled or self-grown materials. In the case of the shading screen, for instance, the workflow proposes as material variables both rescued timber from construction scaffolding and pine wood from responsibly managed forests that provide environmental, social, and economic benefits. During the manufacturing process, embodied carbon emissions can be lowered through the use of more efficient prefabrication dry methods. For instance, in the case of the shading screen, a manufacturing scenario employing the Howick steel framing machine was proposed because of its use of light-gauge steel, which is a modern form of building that has been proven to reduce environmental impact.

On a building scale, the operational carbon emissions can be reduced by reducing the energy used for cooling, heating, and lighting. In all the design experiments, the global form of the workflow instance is a result of an environment-driven process.

This leads to interesting questions about the role of the architect. The increasing availability of advanced manufacturing technologies, means the profession is returning to the consideration of construction and manufacturing as a part of the design process, and this returns to the architect control over the fabrication of her work, blurring the line between design and construction that was artificially created by modernism. But it also blurs the traditional role of the architect, confronting her with a multidisciplinary set of new skills and knowledge (Figure 69).

Figure 69

6. References

Peters, B., & Peters, T. (2018) Computing the Environment: Digital Design Tools for
Simulation and Visualization of Sustainable Architecture. West Sussex, UK: John Wiley & Sons Ltd.

Schwitter, C. (2005) Engineering complexity: Performance-based design in use. In B. Kolarevic & A. Malkawi (Eds.), Performative architecture: Beyond instrumentality (pp. 113-122). New York, NY: Spon Press.

Leatherbarrow, D. (2009) Architecture Oriented Otherwise. New York, NY: Princeton Architectural Press.

Willmann, J., Block, P., Byrne, K., Hutter, M., & Schork, T. (2018) New Scientific Frontiers. In J. Willmann, P. Block, K. Byrne, M. Hutter, & T. Schork (Eds.), Robotic Fabrication in Architecture, Art and Design 2018 (pp. ix-xiv). Cham, Switzerland: Springer.

Hauck, A., & Bergnin, M. (2017) The triumph of the turnip. In B. Sheil, A. Menges, R. Glynn, & M. Skavara (Eds.), Fabricate 2017 (pp. 16-21). London, UK: UCL Press.

Buildings and Climate Change (n.d.). *U.S. Green Building Council*. Retrieved from http://www.usgbc.org

Strain, L. (2016, April 4). *Building Materials and the Time Value of Carbon*. Retrieved from https://www.buildinggreen.com

Ostergard, T., Jensen, R.L., & Maargaard, S.E. (2016) Building simulations supporting decision making in early design – A review. Renewable and Sustainable Energy Reviews, 61, 187-201.

Bhooshan, S. (2017) Collaborative design. In R. Garber (Eds.), Workflows: Expanding Architecture's Territory in the Design and Delivery of Buildings (pp. 82-89). West Sussex, UK: John Wiley & Sons Ltd.

Naboni, E., Ofria, L., & Danzo, E. (2019) A Parametric Workflow to Conceive Facades as Indoor and Outdoor Climate Givers. Symposium on Simulation for Architecture and Urban Design, 2019, 11-18.

Hershcovich, C., Hout, R., Rinsky, V., Laufer, M., & Grobman, Y.J., (2017) Microclimate on building envelopes: wind tunnel and computational fluid dynamic analysis of basic and complex geometries. Symposium on Simulation for Architecture and Urban Design, 2017, 259-262.

Matiz, C.C., McMenomy, H., & Erdine, E., (2019) Environmentally Informed Robotic-Aided Fabrication. Symposium on Simulation for Architecture and Urban Design, 2019, 199-206.

Garber, R. (2017) Workflows: Expanding Architecture's Territory in the Design and Delivery of Buildings. West Sussex, UK: John Wiley & Sons Ltd.

7. Appendix

Python Script for Bricks generation

```
#import rhinoscriptsyntax as rs
import Rhino
import math
import ghpythonlib.components as ghcomp

def Waves(Frequency_F, Frequency_T, Amplitud_T, Amplitud_B, Height, H_Gap, B_L, B_H):
    F_F = math.pi * Frequency_F
    F_T = math.pi * Frequency_T
    F_Domain = Rhino.Geometry.Interval(F_F, F_T)
    F_Steps = (F_T - F_F) / B_L
    F = ghcomp.Range(F_Domain,F_Steps)

    arrAPts = []
    arrBPts = []
    arrL = []
    arrCPts = []

    for i in range(0, len(F)):
        temp_yT = math.sin(F[i]) * Amplitud_T
```

```
        temp_yB = math.sin(F[i]) * Amplitud_B
        arrAPts.Add(Rhino.Geometry.Point3d(F[i], temp_yB, 0.00))
        arrBPts.Add(Rhino.Geometry.Point3d(F[i], temp_yT, Height))
        arrL.Add(Rhino.Geometry.Line(arrAPts[i], arrBPts[i]))
        arrCPts.Add(arrL[i].ToNurbsCurve().DivideAsContour(Rhino.Geometry.Point3d(0,0,0),
Rhino.Geometry.Point3d(0,0,Height),(H_Gap + B_H)))

    arrCrvs = []
    for i in range(0, len(arrCPts[0])):
        arrTPts = []
        for j in range(0, len(arrCPts)):
            arrTPts.Add(arrCPts[j][i])
        arrCrvs.Add(Rhino.Geometry.Curve.CreateInterpolatedCurve(arrTPts, 3))

    return [arrAPts, arrBPts, arrL, arrCrvs];

R = Waves(Frequency_F, Frequency_T, Amplitud_T, Amplitud_B, Height, H_Gap, B_L, B_H)

PtsA = R[0]
PtsB = R[1]
Lines = R[2]
Crvs = R[3]
```

Python Script for Shading Screen

```
import rhinoscriptsyntax as rs
from scriptcontext import sticky as st
import Rhino
import random
import math
from System.Drawing import Color

def W_Pattern(Srf, Field_Pt, Field_V, Total_Lenght, Member_Lenght, Member_Width, Member_Thick, I_Angle, Min_Angle, Max_Angle, M_Overlap, i_Step, intMinIntersect, intMaxLevel, Rad_Target, Use_Rad):

    return Pattern(Srf, Field_Pt, Field_V, Total_Lenght, Member_Lenght, Member_Width, Member_Thick, I_Angle, Min_Angle, Max_Angle, M_Overlap, i_Step, intMinIntersect, intMaxLevel, Use_Rad);

def Pattern(strSrf, Field_Pt, Field_V, dblLTotal, dblLMember, dblWMember, dblTMember, dblAngle, dblMinA, dblMaxA, M_Overlap, i_Step, intMinIntersect, intMaxLevel, Use_Rad):
    dblLCtrl = dblLTotal
    Planes = []
    Levels = []
    arrComponent = []
```

```
    if Use_Rad:
        arrAnalysis = AnalysisSrf(strSrf)
    else:
        arrAnalysis = RandomPtOnSrf(strSrf)

    arrComponent = Component(strSrf, Field_Pt, Field_V, arrAnalysis, dblLMember, dblWMember, dblTMember, dblAngle,
dblMinA, dblMaxA, M_Overlap, i_Step, st["arrMeshes"], intMinIntersect, intMaxLevel)
    st["strComponent"] = arrComponent[0]
    Planes.Add(arrComponent[1])
    Levels.Add(arrComponent[2])
    st["arrMeshes"].append(st["strComponent"])
    #print dblLCtrl
    dblLCtrl = dblLCtrl - dblLMember
    #rs.Prompt("Available Wood = {}".format(dblLCtrl))
    return [Planes, st["arrMeshes"], Levels];

def AnalysisSrf(arrSrf):
    strPath = "C:/Rad.csv"
    radVal = CSV_read(strPath)
    print radVal[1][0]
    arrAPt = Rhino.Geometry.Point3d(radVal[0][0],radVal[0][1],radVal[0][2])
    arrAVec = Rhino.Geometry.Vector3d(radVal[1][0],radVal[1][1],radVal[1][2])
```

```
    arrPlane = Rhino.Geometry.Plane(arrAPt,arrAVec)

    return arrAPt;

def RandomPtOnSrf(arrSrfs):
    intPick = random.randint(0,len(arrSrfs)-1)
    dblU = random.uniform(0,1)
    dblV = random.uniform(0,1)
    #arrFrame = arrSrfs[intPick].FrameAt(dblU,dblV)
    arrRPt = arrSrfs[intPick].PointAt(dblU,dblV)

    return arrRPt;

def Component(strSrf, Field_Pt, Field_V, arrAPt, dblLMember, dblWMember, dblTMember, dblAngle, dblMinA, dblMaxA, M_Overlap, i_Step, arrMeshesL, intMinIntersect, intMaxLevel):
    dblClashTol = 0.00
    intMaxClash = 1
    intLevel = 0

    dblChance = random.uniform(0.00, 1.00)

    arrField = Field(strSrf[0], Field_Pt, Field_V, arrAPt)
```

```
    dblDistanceToMid = arrField[1]
    dblLB = arrField[2]
    dblUB = arrField[3]

    if dblChance < dblRandomMix:
        arrCPlane = PlaneFromNormal(arrAPt, arrField[0], strSrf[0].NormalAt(0.00,0.00))
        boolC = True

    else:
        arrCV = Rhino.Geometry.Vector3d.CrossProduct(arrField[0], strSrf[0].NormalAt(0.00,0.00))
        arrCPlane = PlaneFromNormal(arrAPt, arrCV, strSrf[0].NormalAt(0.00,0.00))
        boolC = False

    arrCPlane.Rotate(math.radians(dblAngle), arrCPlane.XAxis, arrCPlane.Origin)

    objMesh = MeshBox(arrCPlane, dblLMember, dblWMember, dblTMember, boolC, dblDistanceToMid, dblLB, dblUB)
    objMesh.SetUserString('Level', str(intLevel))
    objMeshX = ComponentClash(arrCPlane, objMesh, arrMeshesL, dblClashTol, intMaxClash, intMinIntersect, intMaxLevel,
dblWMember, i_Step, M_Overlap)

    if intColorLevel is not None:
        if objMeshX is None:
```

```
        pass
    else:
        strLevel = objMeshX.GetUserString('Level')
        if int(strLevel) == 0:
            pass
        else:
            objMeshX.VertexColors.CreateMonotoneMesh(Color.White)

if objMeshX is None:
    strL = None
else:
    strL = objMeshX.GetUserString('Level')

return [objMeshX, arrCPlane, strL];
```