RAPID ASSEMBLY OF MASONRY STRUCTURES WITH AD-HOC MATERIAL ATTRIBUTES, COMPUTER VISION, AND ROBOTS

ADAM FINGRUT[1] and CARSON KA SHUT LEUNG[2]
[1,2]*The Chinese University of Hong Kong.*
[1]*adam.fingrut@cuhk.edu.hk, 0000-0002-8898-8863*
[2]*kashutleung@cuhk.edu.hk, 0000-0003-2936-0344*

Abstract. This paper discusses the design and development of scale masonry structures using robot arms, computer vision hardware and bespoke computational workflows. In parallel to the development of full-scale masonry solutions using a Cable Driven Parallel Robot (CDPR), a faster method for testing large numbers of brick elements is needed to verify buildability, mitigate collisions, and think differently about recycled materials during real-world construction activities. Additionally, by incorporating scanning and analysis technology, materials can be digitized, and their attributes translated into variables for placement within an intended structure.

Keywords. Automation in Construction; Masonry; Computational Design; Discrete Element Assemblies; Reuse; SDG 9; SDG 12.

1. Introduction

The automated assembly of masonry structures is becoming an increasingly popular area of research in the fields of computational design, structural, and mechanical automation engineering. From Gramazio and Kohler Research's project Flight Assembled Architecture to Mamou Mani's POLIBOT, teams have been developing and deploying systems using cable robotics, drone technology, and stationary arms for this purpose as a means toward further exploring the potential of automation in construction with dexterous work fundamentally calibrated to human scale and traditional craftsmanship. Robotic arms and computer vision is commonplace within industrial automation and are ideal for "pick and place" operations. They are also ideal for emulating brick laying activities due to their two-link arm layout which is similar human ergonomics.

UN Sustainable Development Goal 12 declares the need to substantially reduce waste generation through prevention, reduction, recycling, and reuse. There is great need for the construction industry to consider its relationship to waste, material use, and the carbon footprint associated with creating new materials rather than upcycling. Bricks and masonry are robust materials that offer an opportunity for reuse in new building assemblies. One consideration when recycling any material is the energy

POST-CARBON, Proceedings of the 27th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2022, Volume 2, 11-20.

required to sort and/or clean into a usable state. This paper considers new methods for reducing some of those processes by considering the possibilities of unsorted masonry, of various dimensions and attributes, for further integration into construction assemblies.

2. Research Question and Objective

This research asks the following questions: 1) What is the potential for automation systems typically found in the manufacturing industry to be transposed to building construction? 2) Can irregular block materials be integrated into an automated structure assembly without pre-sorting? 3) Can this setup be linked with standard architectural tools using computational design, computer vision and collaborative robotics?

To answer to these questions, the following activities needed to occur: 1) Develop and configure communications between a computer model, robotics, and computer vision systems for accurate scanning, design integration and execution of pick and place operations; 2) Populate the system with materials containing unpredictable and randomized material dimensions; 3) Test the system and proof of concept using scale equipment for assembly of small structures using above.

3. Background

The above questions follow a series of research projects that examine the integration automation equipment for building construction, and for teaching and learning among higher learning and architecture students. Those projects linked standard "off the shelf" robotic arms with customized GH and PY components that allowed for seamless one-way communication between standard design tools found in architecture, and automation hardware that could carry out instructions – under the assumption that the scale brick materials were predictable and known ahead of the design. The preliminary design methods are also based upon automation and construction projects using long spanning cable robotics for masonry applications (Fingrut, 2019). This shows the scalability and transferability of the system into live construction contexts.

4. Hardware Overview

4.1. ROBOT ARM AND END EFFECTOR

The experimental setup is developed around the Dobot Magician robot arm. It consists of a base, rear arm, forearm, and end-effector. The 3 degrees of freedom (DoF) robot has three stepper motors to actuate its joints and generates a maximum payload capacity of 500 grams. The end-effector utilizes a servo motor and a pneumatic pump to operate with its payload. The maximum distance that can be reached by the equipment is 320 mm with 180° rotational capability. It is attached to a sliding rail that has an effective travel distance of 1000mm, allowing the robot arm to operate long distance pick and place tasks with a total volumetric working area of 320mm X 320mm X 1110mm. The setup up is mounted onto a custom platform to explicitly define the build area and to optimize travel distances of the setup. An identical 3D model (digital twin) of the custom build platform and all associated robotic hardware aids in the planning of design output and allows for the translatability of physical and digital data.

4.2. CONVEYOR BELT AND SCANNING HARDWARE

The system works in conjunction with an image processing system that extracts the characteristics of brick elements by calibrating the range of hue, saturation, value, and pixel area of the imageries obtained from the computer vision kit. The image processing system includes a HIKROBOT (MVL-HF1228M-6MP) 1/1.8" 12mm F2.8 manual iris C-mount lens mounted to a HIKROBOT (MV-CE050-30UC) 5 mega pixel colour camera, 2.2 micrometre narrow field of view, 1/2.5” CMOS, Rolling Shutter, 31 fps, and connects to a workstation via a USB 3.0 Output. A white 40000 lux ring shaped auxiliary light source is mounted around the camera to provide consistent ambient lighting condition and intensity. It is important to note that any varying and inconsistent external source of lighting (skylight) or complex multidirectional lighting (interior lighting) will affect the image processing system. Therefore, consistent ambient room conditions were necessary, to avoid recalibration.

The camera points vertically downwards from a camera mounting kit 300-350mm above the target brick picking location and analyses the XY plane image which it transforms pixels into Cartesian coordinates. The system temporarily stores that data for potential collection by a robot, or other system via PY-API. An automated conveyor belt makes it possible to enlarge the pick-up area without requiring the camera to move. This has an added proximity sensor to ensure the visible zone is populated with material, or to advance the belt forward.

4.3. AD HOC DELIVERY OF BRICK MATERIALS

Based upon the system scale, reach, and limited payload, a series of scaled wooden and acrylic bricks measuring roughly 10mm at width were used for the experiment. The

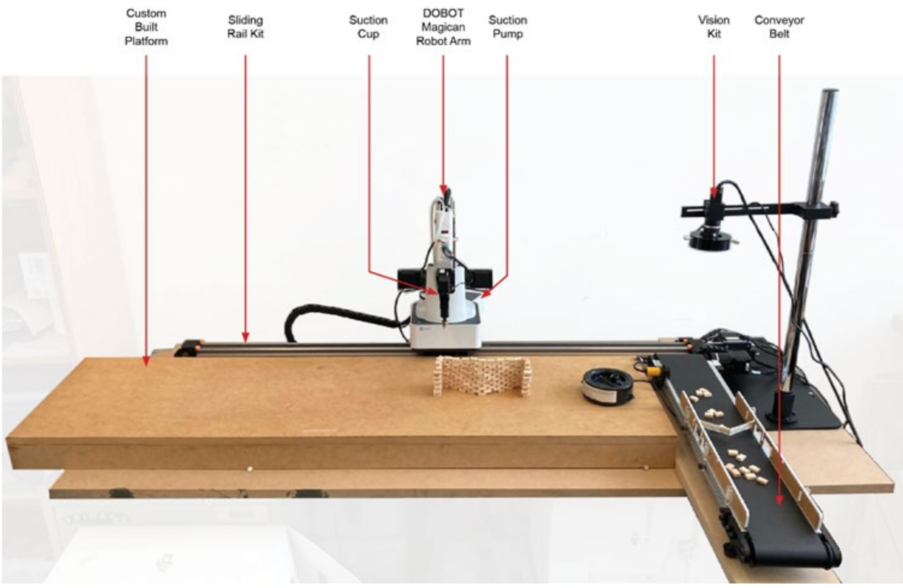

Figure 1. Robot and Computer Vision Setup

length of the bricks ranged from 15mm to 30mm in a randomized manner. The stock material also has a variable wood grain. Materials were randomly placed only the conveyor belt, and guiding rails helped to align or discard any units that may have been placed along the wrong edge. This physical pre-filter helped to maintain clear boundaries between adjacent bricks that may be processed as single units.

5. Control System Overview

The control system uses Rhinoceros 3D (RH) and Grasshopper (GH) as a central platform for design, communications, and coordination between the scanning and robotic assembly process. This is carried out using custom PY tools that were embedded as GH components and allowed for real-time data feedback loops to automate the design process based on data collected from the computer vision system.

While traditional fabrication process sees design and production as two distinctive tasks which involve design data output from RH and input to the robotic control software, the workflow of this setup conceives the system as one singular workflow. Robotic control functionalities are added to RH via its plugin database and its API. This streamlines the connectivity of different software (such as the computer vision modules) and creates stable, looped communications between hardware and software without crashes or unexpected behaviour.

This is achieved with the computational modelling plugin GH for RH. The node-based editor formulates the main interface, in which data is passed in a unidirectional manner between components from start to endpoint with connecting links. Input data can be defined locally as a constant and can also be imported from the modelling space of RH or from a saved comma separated value (CSV) file. Data is always stored in parameters, which can either be free-floating or attached to a component as input and outputs objects. Within the algorithm, data is free floating as parameters.

5.1. DESIGN MODELLING

A curve that represents the building path of the bricks is modelled within the design software. Curve data is then defined as the input data of the GH/PY definition which automatically populates the curve with brick bounding box dimensions based on attributes extracted from the computer vision system.

The computational definition parses brick attributes, such as length, width, colour, and sorts into list. It then selects the desired brick by comparing the unbuilt length of the curve input to the list of brick. Based on this the model selects the longest possible brick (best/biggest fit) until the remaining built length is shorter than any available bricks returned from the computer vision input. The brick spacing is also specified as a local input within the definition.

Base curve design requires design decision making relying upon external contexts. The rest of the definition is designed in a recursive manner, for which autonomous design decisions are made upon the beginning of the construction process. While the overall massing of the assembly can be predicted, the specific layout of brick units will only be fully visualized and known upon completion of its construction. This feedback loop (Figure 3) within the definition aids in real time construction process by using recursion. The outcomes achieved are the possibility of defining an infinite set of objects by a finite statement (Wirth, 1985).

Besides the default scripting components native to the GH environment, the GH/PY components allow RhinoScriptsSyntax functions to instantiate modelled geometry inside of GH. PY nodes can expand upon the functionality of RH to become a communication device between different tools, hardware, and software programmes working together on a singular task. Custom built PY tools were embedded in the GH definition to create real-time links to the computer vision and the hardware control API. They also store and display project specific information beyond the default data manipulation capability of GH. Two custom PY tools were created for this setup, namely the Vision_PY (passing variables from the computer vision modules to the computational design setup) and Robot_PY (passing instructions from the computational design setup to hardware for assembly).

The function libraries used within the GH/PY environment includes RhinoScriptSyntax that provided functionality relating to geometry modelling, and DobotDIIType that provided robotics control commands and methods. To specify a brick placement coordinates, the robotic arm read a set of (x, y, z) coordinates generated from the GH definition via the Robot_PY which then allowed the robotic arm and vacuum gripper to reach the specified location while observing pre-programmed movement rules.

5.2. COMPUTER VISION

To specify a brick pickup location, the Vision_PY relays data from the computer vision module via a bi-directional Transmission Control Protocol (TCP) Network Socket. The Vision_PY communicates brick width, height, (x, y) coordinates, rotation angle, RGB data to the GH/PY component. This is triggered by an 'image update' controlled from within GH and a trigger component.

The data generation process of the machine vision setup starts with camera calibration, which is carried out within the Dobot Vision Studio environment which uses a node-based interface for custom definitions that operates on and organize feedback on data processing. The first node inputs the camera imagery with exposure calibration and cropping functionality. This step is crucial to the construction accuracy as a calibration computer vision module definition is used to calculate the geometric image transformation. The HIKROBOT (MVL-HF1228M-6MP) lens and HIKROBOT (MV-CE050-30UC) camera is a pre-set in the vision software, compensating through pre-calibration for lens distortion. However, the distance and angle between the camera and the target XY plane is defined in physical setup, therefore, a sequential N- point calibration node is needed to construct a calibration file using the imagery data that passes through the checkerboard pre-sets in the CalibBoard Calib node. This step correlates the real word coordinates of a checkerboard image and the relative pixel coordinates.

The calibration file is then used to undistort the image received from the camera lens. The connected camera image exposure, pixel format is adjusted according to the environmental condition. The image then passes through a colour extraction node, in

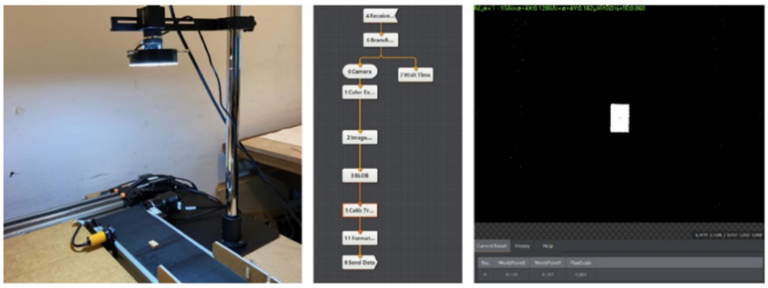

Figure 2. LS: Brick under Computer Vision; M: Computer Vision Definition; RS: Brick parameter detection

which the colour making attribute, (hue, saturation, lightness) HSV parameter is calibrated by of limiting the attribute range to create a figure ground style binary image (Figure 2). The binary image is then morphed to define the perimeter of an object by locating the edges of the outer extremities. The process goes through a dilated convolution which expands the boundary (kernel input) by skipping pixels to cover a larger area of the input. In morphological dilation operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbours in the input image (Bishop, 1995).

During system testing, the interaction time is set to 0 to avoid dilation of unwanted white points detected, while the kernel matrix width and height set to facilitate the feature extraction process. Once a boundary line is established, the blob detection node returns the area, perimeter length, centroid XY, rotation angle, rectangularity rate and long, short axis length. The set of data is then filtered according to area range and rectangularity range pre-sets. The resulting verified data is compiled and sent to the custom PY tool in GH.

The setup of this system comprises of four components: 1) assortment of brick units delivered via conveyor belt; 2) computer vision processing material ad attributes; 3) design software that probes for the 'best fit' location of that item and communicates those details; 4) automation hardware that carries out instructions set in 3.

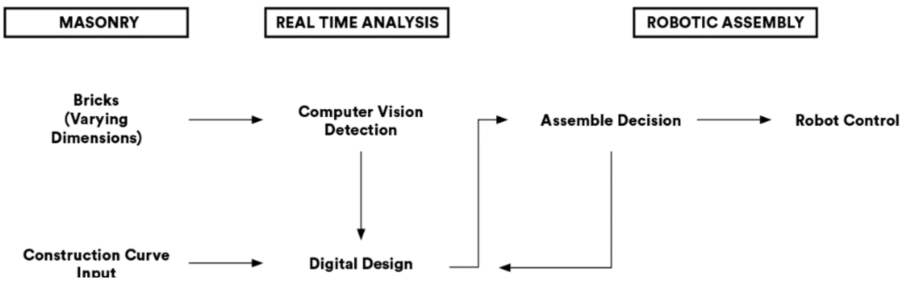

Figure 3. Scan-Analysis-Loop

6. System Testing

From this setup, a series of experiments were conducted to test different methods of optimized pick and place operations for small scale testing and assembly. Three cases were tested to supply sufficient diversity in brick type, guide curve setup, and approach to guide curve population. While the brick input is limited to 5 per input, each case added an additional variable of 5mm linear spacing along the curve variable between elements, to ensure no collisions would occur during assembly. Each type of test was conducted and repeated a minimum of five times with randomized input materials to establish insight into the success, accuracy of the equipment, and predictability of the design weaknesses.

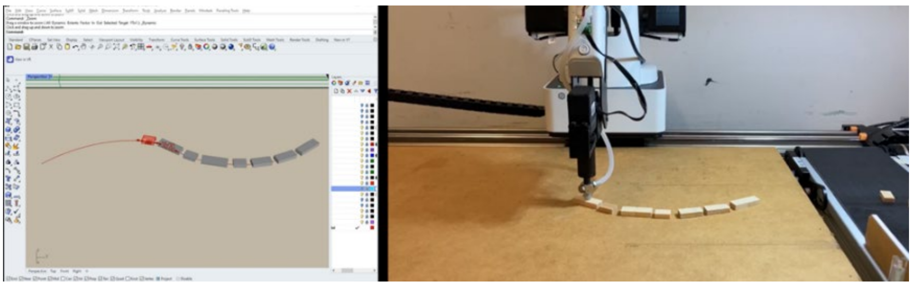

Figure 3. LS: Simulation; RS: Automated Assembly

6.1. LINEAR DEPLOYMENT (LD)

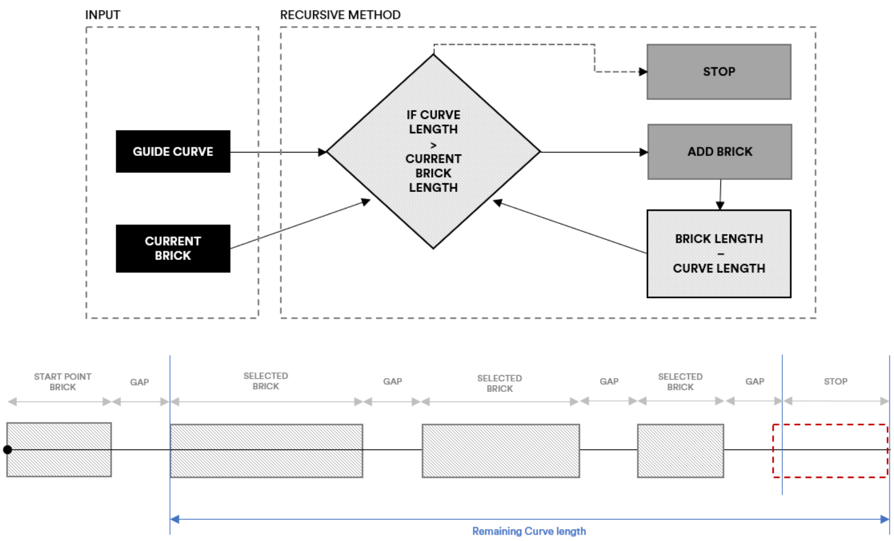

Figure 4. Linear Deployment

In this, the simplest of three tests, ad-hoc brick units were deployed in a linear fashion from a starting point toward an endpoint of the guide curve (Figure 5). For each iteration of the system loop, a brick that meets sufficient criterion (set as variables) would be added along the curve in both the design model, and simultaneously executed by the robotic hardware. This recursive process was repeated until brick units were too large to fit into the remainder of the guide curve. The advantages of this test were a highly predictable starting point, and regulated placement of units – however an unpredictable end condition that could vary widely depending on the lengths of brick units.

6.2. MIDPOINT-OUT (MO)

MO describes a deployment process whereby the midpoint of the guide curve is established for the first brick unit placement (Figure 6). Once this is in place, the system selects best fit based on the remaining curve length in either direction against the available best-fit scanned materials. This recursive algorithm bifurcates the problem into two separate guide curves, that are then treated similarly to the LD setup. The advantages of this, are the highly predictable midpoint placement, however, end conditions suffer from similar irregularities as the LD, with uncontrollable end conditions.

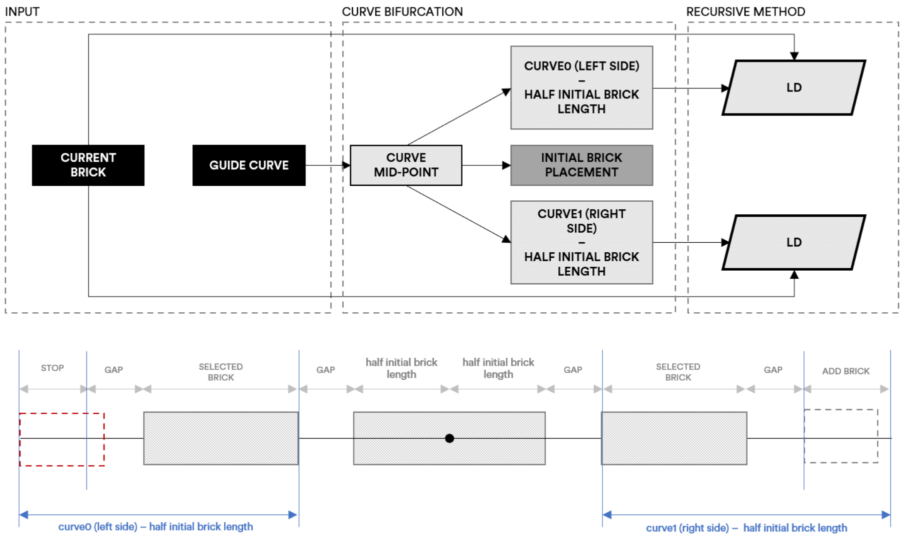

Figure 5. Midpoint-Out ERD

6.3. ENDPOINT-IN (EI)

EI describes a deployment process whereby guide-curve end conditions are established as the first, and second brick units for placement. This is critical as it prioritizes end conditions as being the most accurately placed, and subsequently selects the best fit for the remaining curve length while populating internal spaces (Figure 7). This recursive algorithm continuously evaluates new 'end conditions' on smaller sub-curves until no brick unit can be safely fit into the assembly without collision. The advantages of this, are the highly predictable endpoints, and spacing of units, however, final brick placement suffers from the potential for collisions based on the attributes of brick units in the supply chain.

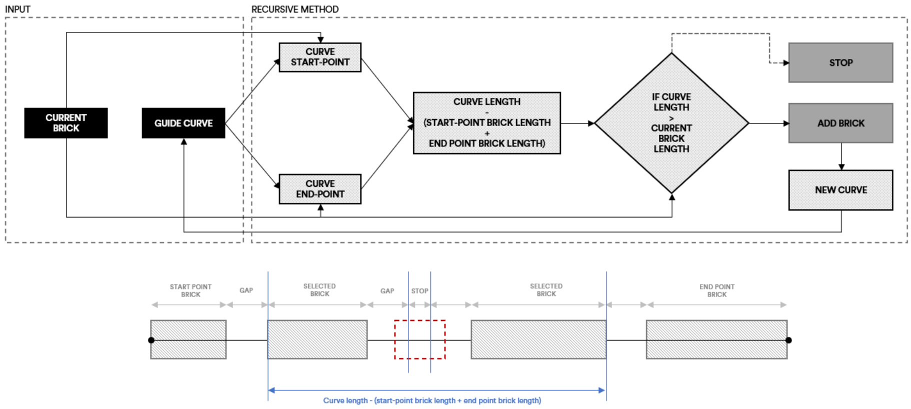

Figure 6. Endpoint-In

7. Findings and Results

The above testing and experiments have shown that there is potential or automation systems to be transposed into building construction, when less "predictable" variables, such as tolerances related to building materials, or recycled and unsorted materials are integrated into assemblies. Although there are many refinements that will need to be incorporated into the setup, the preliminary findings positively conclude that ad-hoc masonry materials can be considered as part of an automated robotic assembly system.

The key aspect of this research project is to determine how computer vision can carefully digitize materials, parse their attributes into usable variables in a 3D design model, and subsequently execute automated assembly based on that model. Discrete assemblies, such as those found in masonry are typically based on regular, sorted patterns. Laborers are often using their experience to modify and fine-tune placement based on minute and large differences in material attributes, site conditions, and design elements. The above preliminary tests have shown that this process can be digitized and automated to some degree (Figure 4).

By linking this setup to standard architectural tools, such as RH, it provides access to a broad range of users without requiring specialist computer programming skills and gives designers the flexibility of designing without having any knowledge of material specifics. This leaves the potential for automated masonry to reconsider a position of pattern design based on material availability, rather than a typical more top-down approach.

8. Future Directions

Further computational dexterity and specificity is needed to advance this line of research into more deployable construction contexts. Particularly, more complex subroutines that can more flexibly consider local unit rotation as part of a best fit algorithm. The current setup only allows for bricks to be placed in parallel to the curvature normal, however, rotating bricks (under certain conditions) would increase the potential fitness of units based on their width instead of length alone.

The above tests only consider two-dimensional placement sensitivity. Although 3D structures were assembled, they were calculated as discrete layers within a system. Needed is a method to consider the placement of new brick units onto previously placed brick layers, ensuring that each unit is sufficiently placed on a stable foundation via intersecting surface area.

Scanning conducted in this setup is only considered in a 2D setup, and the height of all units is assumed to be the same. In future iterations, scanning should include a Z(height) factor that can increase the complexity of material assemblies. This type of configuration may require a mortar or adhesive layer to assist in the fine differences between material layers and levelling.

With the revisions, the system may be sufficiently prepared to scale up toward more realistic architectural material testing, using a stronger hardware such as a cable robotic system. This would allow for a broadening of the research and system into structural engineering.

References

Bonwetsch, T. (2012). Robotic Assembly Processes as a Driver in Architectural Design. *Nexus Network Journal*, 14(3), 483-494.

Dakhli, Z., & Lafhaj, Z. (2017). Robotic mechanical design for brick-laying automation. *Cogent Engineering*, 4(1), DOI: 10.1080/23311916.2017.1361600.

Fingrut, A., Crolla, K., & Lau, D. (2019). Automation Complexity-Brick By Brick. In *24th International Conference on Computer-Aided Architectural Design Research in Asia: Intelligent and Informed, CAADRIA 2019* (pp. 93-102). The Association for Computer-Aided Architectural Design Research in Asia (CAADRIA).

Kohler, M., Gramazio, F., & Willmann, J. (2014). *The robotic touch: how robots change architecture.*

Mora, R., Rivard, H., & Bédard, C. (2006). Computer representation to support conceptual structural design within a building architectural context. *Journal of Computing in Civil Engineering*, 20(2), 76-87.

Pritschow, G., Dalacker, M., Kurz, J., & Gaenssle, M. (1996). Technological aspects in the development of a mobile bricklaying robot. *Automation in Construction*, 5(1), 3-13.

Sousa, J. P., Varela, P. A., & Martins, P. F. (2015). Between manual and robotic approaches to brick construction in architecture. In *33rd eCAADe Conference* (pp. 361–370). Education and research in Computer Aided Architectural Design in Europe (eCAADe).

Wu, Y., Cheng, H. H., Fingrut, A., Crolla, K., Yam, Y., & Lau, D. (2018, May). CU-brick cable-driven robot for automated construction of complex brick structures: From simulation to hardware realisation. In *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)* (pp. 166-173). IEEE.