Formation Embedded Design

A METHODOLOGY FOR THE INTEGRATION OF FABRICATION CONSTRAINTS INTO ARCHITECTURAL DESIGN

Dave Pigram
University of Technology Sydney
Supermanoeuvre

Wes McGee
University of Michigan

ABSTRACT

This paper presents a methodology for the integration of fabrication constraints within the architectural design process through custom written algorithms for fabrication. The method enables the translation from three-dimensional geometry, or algorithmically produced data, into appropriately formatted machine codes for direct CNC fabrication within a single CAD modeling environment.

This process is traditionally one-way with part files translated via dedicated machine programming software (CAM). By integrating the toolpath creation into the design package, with an open framework, the translation from part to machine code can be automated, parametrically driven by the generative algorithms or explicitly modeled by the user. This integrated approach opens the possibility for direct and instantaneous feedback between fabrication constraints and design intent.

The potentials of the method are shown by discussing the computational workflow and process integration of a diverse set of fabrication techniques in conjunction with a KUKA 7-Axis Industrial Robot. Two-dimensional knife-cutting, large-scale additive fabrication (foam deposition), robot-mounted hot-wire cutting, and robot-assisted rod-bending are each briefly described.

The productive value of this research is that it opens the possibility of a much stronger network of feedback relations between formational design processes and material and fabrication concerns.

Keywords. Robotic fabrication; multi-axis; file-to-factory, open-source fabrication, parametric modeling, computational design.

1 Introduction / Background

The drawing is the traditional mediator between design and construction. Since the ancient Egyptians, architects have employed codified drawing systems to describe their building designs (Bernstein 2004), especially in order to have them constructed.Alberti's 15th Century description of the architectural design preceding its own, direct and unchanged, realization through building (Carpo 2011) set the foundation for a historical, and temporal, divide between design intent and design realization. This divide was to be traversed only once, and in one direction only.

In contemporary times, the majority of architectural drawings have merely 'evolved' from "layered transparencies carefully complied with pencil or ink [into]... layered compilations of computer generated lines, arcs and circles" (Bernstein 2004). Drawings, at least seemingly, must still occur between the completion of the design and the commencement of the construction. So-called 'fast-tracked' projects are a minor perversion of this, not a contradiction to it. Thanks in part to technology transfer from other disciplines, data-rich models are now seeing employment in the fields of architecture and construction. Building Information Modeling (BIM), computational simulation and algorithmic form generation are exemplars of a trend that reflects the fact that as the formal and tectonic constraints of design have become more complex, the requirement that architects understand the ultimate means of production has become critical. This may include, for example, the need to understand what input geometry will be required by a specific CAM package; some require curves while others work from surface or mesh data. Algorithmic techniques have led to incremental steps towards efficiency, such as unrolling formed surfaces, slicing sectional contours, and nesting parts into flat or even 3D stock.

Meanwhile, the translation from conceptual design to construction documentation and fabrication remains a largely one-directional affair.

Even in advanced practices heavily invested in digital fabrication research, part- or shop-drawings are still predominantly produced by specialists and translated via dedicated machine programming software packages (CAM). This geometry-based digital workflow, while highly advanced relative to the drawing, retains the fundamental weakness that the transfer of information between platforms incurs friction (time and energy) and as such is only likely to occur so many times. It also does not avoid the "reworking" that frequently occurs when fabrication constraints are not adequately understood during the conceptual design phase.

Significant work remains to be done to streamline the process from design to fabrication.

2 Approach

The methodology proposed here is fundamentally about removing the fissure between the processes that bring forth a design and the operations that will lead to its eventual fabrication. The design and production process must be reconceived, not as a top-down sequence, but as a bi-directional continuum allowing for multiple feedback loops and negotiations between design intentions and an array of other influences including fabrication constraints and material properties as well as financial and contextual pressures.

Fundamental to enabling appropriate feedback, is the development of a software platform that allows the designer to seamlessly migrate between design and fabrication, minimizing one-way translation thresholds. This software platform is built on the understanding that a construction drawing is simply a graphical representation of a set of instructions. Given this, it is clear that if the instructions can be precisely defined, communicated and understood through a different medium, then the drawing itself is no longer required.

"You are an organizer, not a drawing board artist" (Le Corbusier in Sharp 1978).

To achieve this we have established an algorithmically driven grammar of fabrication capable of incorporating an open-ended set of process-specific variables and

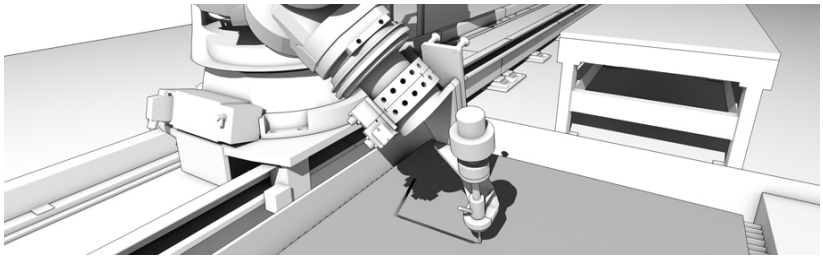


Fig. 1

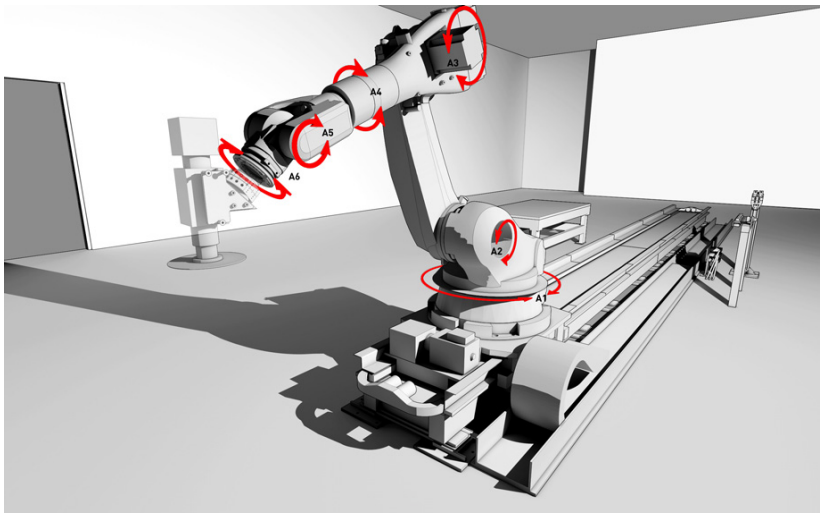


Fig. 2

calculations. In the case studies described here this has been enacted within the 3D Modeling environment Rhino (McNeel Rhinoceros 3D), and its embedded scripting language Rhino Visual Basic, but the conceptual framework is language and software agnostic. Version 2.0 of the platform is currently being developed in the Python language for open source release. By integrating the toolpath creation into the design package, with an open framework, the translation from part to machine code can be automated, parametrically driven by generative algorithms, or explicitly modeled by the user. This integrated approach delivers the opportunity for direct and instantaneous feedback between fabrication constraints and design intent.

The software is developed in such a way as to provide a hierarchy to motion paths, without being limiting in application. The capacity for adding additional object-oriented data allows more complex processes to operate with similar underlying geometry to simpler operations. An example would be comparing the machine instructions for multi-axis milling with those of multi-axis waterjet cutting **(Figure 1)**. Typically, milling operations begin by loading the correct tool into the spindle, activating the spindle at the correct rpm, and performing sequential rapid, plunge, feed, and retract moves; these differ only in speed. Toolpath geometry can be embedded with the correct tool data, and the system will automatically change tools between paths. By comparison, waterjet cutting relies on a very specific sequence of tool activations at every path start and end, in order to properly feed abrasive while preventing the cutting stream from marking the material between cutting paths. Additionally, more advanced waterjet software varies the feedrate of the tool based on part geometry, in order to control taper effects. These types of behaviors can be incorporated into the generic framework of the software, allowing features to be expanded to suit more complex processes.

While these capabilities exist in the majority of CAM software packages, the ability to embed these specific attributes into the parametric model distinguishes this approach. It is worth noting that similar capabilities have long been sought by manufacturing industries. So called “feature based machining” represents the main thrust of this effort, whereby part models are embedded with design intent, exchangeable between various

Figure 1. *Multi-axis (robotic) waterjet system*

Figure 2. *7-Axis industrial robot with axes labeled (linear track is the 7th Axis)*

CAM packages, and retain information necessary to automatically generate toolpaths. As it was primarily developed within industries dealing with solid modeling programs and parts suited to mechanical design, little, if any, crossover has occurred with architectural fabrication methodologies. Especially in the case of algorithmically generated geometry, with a high degree of variability between parts, the capability of automatically translating design intent into machine codes holds the potential for dramatic efficiency gains by reducing programming time.

Feature or knowledge-based machining represents one type of integrated design and manufacturing technology, though arguably it is best suited to parts with limited variation, and its integration into the "design" side is limited at best. Programs such as Delmia and Catia exemplify another approach. At its core, Catia is a parametric modeling system, encompassing features similar to BIM, solid modeling, and feature-based modeling. When fully implemented it allows the entire scope of vertical integrated design and manufacturing. Parts are modeled using constraint-based techniques, and these models are parametrically linked to built-in CAM "workbenches" capable of outputting actual NC code to virtually any standard CNC machine. Under the expanded Delmia PLM (Product Lifecycle Management) package, entire manufacturing operations including automation and 4D (time-based) process information can be included, simulated and controlled. It is arguably one of the most extensive and complicated "design" programs on the market.

While advanced parametric modeling packages, such as CATIA, have been successfully adopted by contemporary architecture practice (Ceccato 2011) they do not readily offer the same level of algorithmic integration in the design process as the methodology proposed here.

2.1 ROBOTIC FABRICATION

Robotic fabrication is an alliance between generic equipment and custom processes; robots were arguably the first truly open source fabrication tools. The machines themselves, in this case a seven-axis industrial model, are relatively generic, and while continually improving in performance, they are clearly linked to their 50-year-old ancestor the Unimate. The promise of robotic fabrication has always been the ability to perform a multitude of unique tasks from a common programming platform; while specific manufacturers use unique syntax, the offline programming techniques used are consistent. The change in recent years is contextual, a result of the development of computational design processes by innovative architecture practices and educational institutions. The use of algorithms to directly control fabrication tools is a natural progression; it simply requires an understanding of specific fabrication processes and the ability to simulate the kinematics of the machine tool (Pigram and McGee 2011). Figure 2 depicts the typical geometric configuration of a 6-axis robot on an external rail (the seventh axis).

2.2 SERIAL INSTRUCTIONS

Apart from the process-influenced geometry itself, the ultimate outcome of the method described here is a series of appropriately formatted instructions. In the examples of processes relating to the Kuka robot, the instructions are written in the Kuka Robotic Language (KRL), but a more common instruction standard is g-code, and these methods have been recently extended to parse instructions into this language for use with an Onsrud 5-axis CNC router (not described in this paper).

The primary aspect of machine instructions common to all manufacturing processes is the requirement to define the location of the machine tool and, for processes with more than 3 axes of motion, its orientation. As we will see later in this paper, there are many more processes and machine specific parameters that require control instructions, but our method begins with the definition of the positions themselves. This could happen directly through an algorithmic process, through explicit modeling, or via a hybrid approach where simple algorithms interpret predefined geometry, usually toolpaths.

The first step is to divide the tool path into the requisite number of points. This can be done by using the start and end points of straight lines, or by dividing lines and

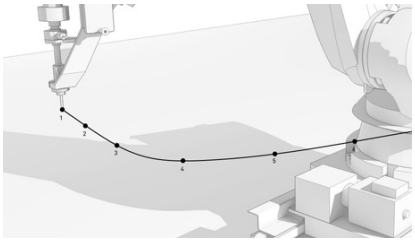


Fig. 3

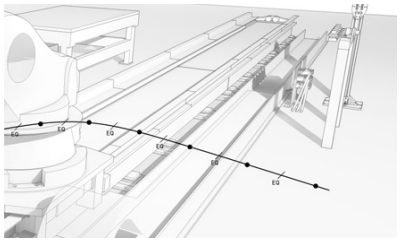


Fig. 4

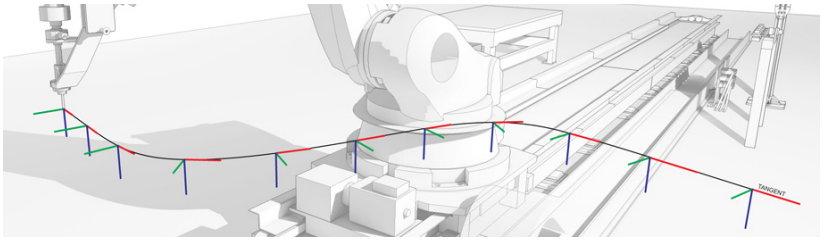


Fig. 5

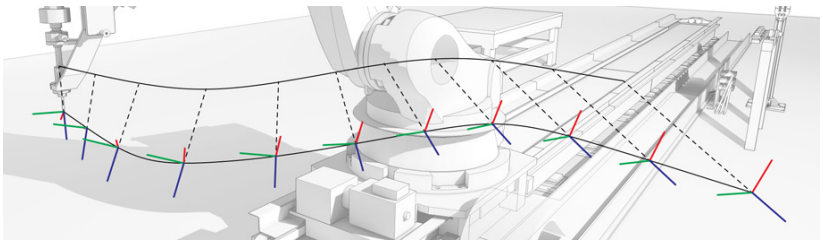


Fig. 6

curves either by length or by number as shown in **Figure 3** and **Figure 4** (the density of points would typically be much denser than shown where curves are being linearly interpolated). In the case of splines, a translation step can be used to rationalize the path into discrete lines and arcs based on a defined chord tolerance. Specific machine code standards will support various methods of rationalization; often g-code simply linearizes the 3D curves into small straight segments, while KRL allows the programming of continuous arcs on arbitrary planes, giving the equivalent path greater accuracy with fewer instructions, and subsequently smaller file sizes.

The next step is to establish the tool orientation. **Figure 5** shows the orientations for a tool that is 2D-aligned, such as a knife cutter. For processes that utilize tilting tool axes, a reference curve is often used to align the tool, as in **Figure 6**. In this case the Z-axis of the tool is the "working" direction, while the orientation of the X and Y of the tool is arbitrary relative to the process. Relative to the robotic system, the orientations are important however, so the user is prompted to provide a suitable orientation (the projection of the X axis is used to solve the extra degree of freedom).

2.3 SIMULATION

For machines with greater than three axes of motion, the number of ways in which an x,y,z-coordinate position can be achieved is infinite. Even when sufficient orientation information is added, there are often numerous ways of posing a 6+ axis machine that all satisfy the given tool location. (Brell-Cokcan and Braumann 2010) This brings about opportunity, as well as responsibility during fabrication workflow planning. To help determine the most appropriate pose, a 3-dimensional simulation of the fabrication system is calculated using inverse kinematics. In the case of singularities, over-rotations, or over-extensions, numeric values for each joint and linkage provide sufficient information to respond to. For issues of collision detection posed geometry is required for evaluation. This feedback provides critical information to the designer, as understanding the working envelope of the fabrication process, especially in complicated multi-axis or robotic machining, will have dramatic effects on the feasibility of the process.

Figure 3. *Tool paths divided by number*

Figure 4. *Tool paths divided by length*

Figure 5. *Tool paths 2D aligned to a single curve; Z-Axis is always straight down; X-axis is tangent to the toolpath*

Figure 6. *Tool paths aligned using a reference curve in fanning mode: the reference curve is divided in the same proportions as the tool path, creating a ruled edge - X-axis projection on the XY Plane is constant*

2.4 INVERSE KINEMATIC MODEL

The inverse kinematic model of the 6-axis industrial robot has been created using trigonometry to solve for all joint values. **Figure 7** shows the configuration.

This process begins by 'backing-out-of' the tool via a reversing of the displacement and rotation values in the tool's definition, thus yielding the complete position (coordinate and orientation) of Axis 6. The center of Axis 5's rotation is called Point P and has a fixed relationship to Axis 6. The value for Joint-1 is simply calculated by projecting Point P's x,y,z-coordinate values to a world-oriented XY plane at the same height (z-value) as Joint-1's axis of rotation and measuring the angle between the defined zero position on that plane and the projected point. The center of Joint 2 has a fixed geometric relationship to Axis 1. Values for Axis 2 and 3 can be found by computing the angles of a triangle in plane with Point P, the center of Axis 1, and the center of Axis 2. The triangle contains one side linking Joint 2's center to Point P, with the two side lengths given by the lengths of Linkage 2-3 and Linkage 3-5. This yields joint values for A2, A3, and A5. Finally the values of A4 can be calculated by projecting both linkage 2-3 and linkage 5-6 onto a place that is perpendicular to linkage 3-5 and measuring the angle difference between the projections (adjusting for the A4's defined zero position).

A cumulative wireframe sequence gives a good indication of the robot's motion **(Figure 8)**. Aforementioned joint limits, reach limits, and singularities are all called out with red geometry and an error descriptor is attached to the offending joint at each pose.

2.5 PROCESS INTEGRATION

Central to the method proposed is the ability to integrate and respond to the properties and constraints inherent in specific fabrication processes and to remain open to the embedding of future processes. Factors such as tool geometry, feed and speed rates, and tool orientation relative to part geometry are all used to customize the creation of formatted instructions. **Figure 9** shows the orientation geometry, plunge/lead-in, toolpath, retreat/lead-out, and traverse motion paths for one part of a complex reticulated surface structure. A series of 6 operations across three tools including router, saw and drill paths shape the part without requiring a change in fixture, significantly increasing accuracy.

Tool changes are automatically generated when a test recognizes that one position's tool definition differs from that of the previous. **Figure 10** shows the successive moves where a saw blade is loaded into a router spindle.

2.5.1 ROBOT-MOUNTED 2-DIMENSIONAL KNIFE-CUTTING (ROLLER CUTTER)

A custom-made mount for a 1-inch diameter roller-cutter has been used to cut various sheet stock: fabric, rubber and the like **(Figure 11)**.

This was the first and simplest process thus far integrated into the control software. Typically, knife-cutting machines utilize 4-axis CNC control, which allows the blade to be steered to remain tangential to the path. For fabric cutting the stock is flat and uniformly thin so no tilt is required from the blade **(Figure 12)**. The Z-axis of the tool is always oriented vertically and the X-axis follows the tangent of the curve at each point (this is arbitrary and based on the tool definition in the robot control); motion instructions can be created from a single set of oriented input curves. Oscillating knife trimming is also used for processing roto-molded components. In such cases the tilt of the blade is important, and allows complex 3D parts to be trimmed; in that process a second set of (reference) curves is used to define the blade orientation along the tool path.

2.5.2 LARGE-SCALE ADDITIVE FABRICATION (FOAM DEPOSITION)

An innovative technique being developed at the University of Michigan that resembles a large-scale 3D printer, utilizes rapidly curing polyurethane foam

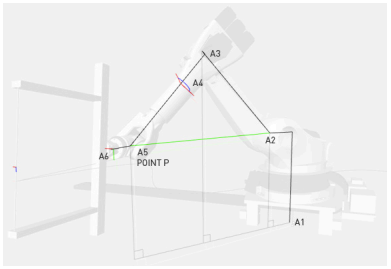


Fig. 7

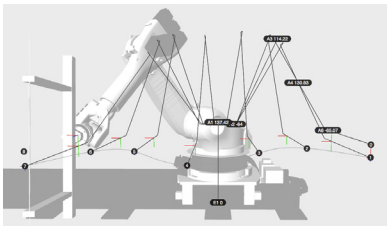
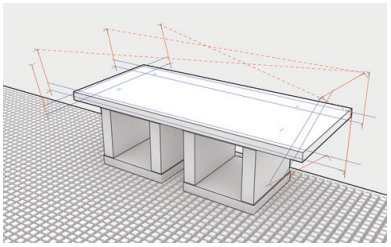


Fig. 8



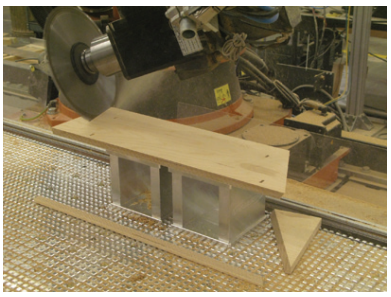


Fig. 9

Figure 7. *Inverse-kinematic wireframe model for joint value calculation*

Figure 8. *Wireframe inverse-kinematic simulation*

Figure 9. *Sequenced operations within an integrated workflow*

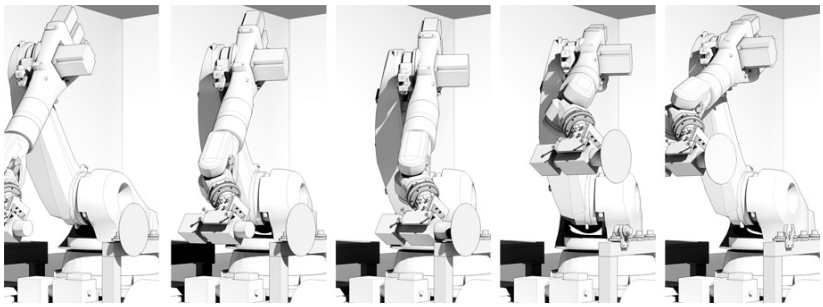
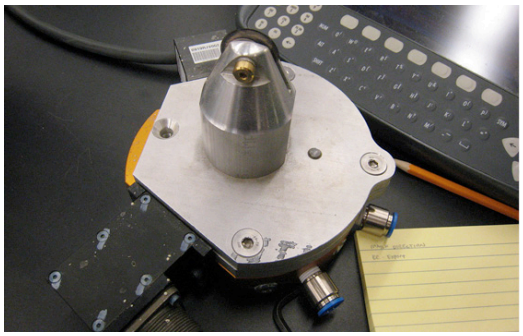


Fig. 10



POSITION :	[X,Y,Z,A,B,C]
TOOL :	"ROLLER_CUTTER"
TOOLDEF :	[0, 0, 220, 0, 0, 0]
POSITIONSTYLE :	2-D ALIGNED

Fig. 11

(Figure 13). The significant advantage this process enjoys over contemporary 3D printing technologies is that the robotic mounting enables the deposition of material from any direction, and the foam is fast-setting and light enough to remove or significantly reduce the need for supplementary support (depending on the form). This technology relies on the chemical reaction between two-part foam. The two part-foam mixture is mixed in the nozzle **(Figure 14)** and reacts in seconds, expanding to a predictable size.

This is an incredibly sensitive process, depending on speed, stand-off, nozzle setting, component temperature and angle of incidence with previously deposited material. Each of these variables must be tightly controlled to produce predictable results. Even at minimum deposition rates, the resolution of the output is such that secondary machining (milling) is required to achieve acceptable surface finishes. However, with the goal of utilizing foam efficiently, the process is considerably more efficient than milling from solid stock.

The upper right foam test in **Figure 15** depicts the result of a minor but compounding error in the assumed expansion rate of the foam. The expansion rate in the straight sections, where the robot was moving faster, was less than that initially calculated. After a certain number of layers the error had compounded to the extent that the robot was aiming above the actual line of deposited foam so instead of adding the next sequential layer, the foam entirely missed its target. As a result the layers ceased increasing in these sections resulting in the large dips shown in the image.

The 'wiggle' wall in the foreground **(Figure 15)** was a subsequent test and clearly shows that the foam expansion rates were better approximated than in the first test. However, it is evident from the non-planar upper surface that expansion was still not equal in all areas. In this case it is a result of reaching a joint acceleration limit on the tight corners and therefore not maintaining a consistent feed rate. Currently the process is under development, and will incorporate a closed loop feedback (ranging feedback) to adapt to changes or errors in the deposition rate.

Finally, the relatively smooth wall in the upper left shows the finish achieved when the surface of the foam is milled after curing. As a pre-finishing technique for fabrication large scale tooling or formwork, the process could provide considerable material savings.

Figure 10. *Automated tool change sequence*

Figure 11. *Custom roller-cutter mount and process variables*

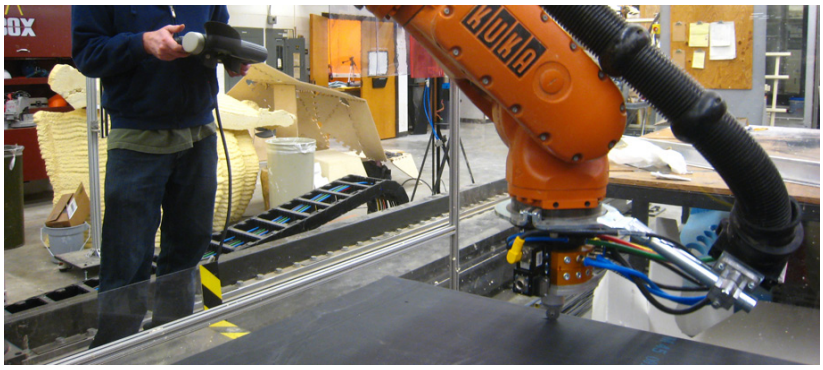


Fig. 12

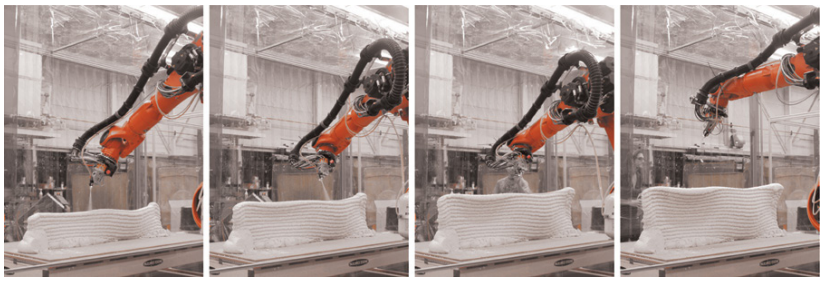


Fig. 13

2.5.3 ROBOT-MOUNTED HOT-WIRE CUTTING

A hotwire cutter consists of a thin wire (in this case 0.0126" thick nichrome) that is heated via an electric current to approximately 200 degrees Celsius and used to cut polystyrene or similar types of thermoplastic foam **(Figure 16)**. The foam vaporizes just ahead of the wire and as such very little force is required to push through the stock. The robot-mounted bow-type hotwire has a series of specific constraints that must be embedded within the control software. The primary constraint is speed, while the primary simulation feedback is the positioning of the "bow" of the tool.

If the wire is moved too quickly, the foam will not vaporize ahead of the wire, and the wire will drag through the foam, deflecting and causing inaccuracies. The kerf width is directly proportional to the speed, so consistency of motion is extremely important. This also means that you cannot stop the wire along a cut without melting excessive kerf into the surface.

While the wire itself is rotationally symmetrical, the bow holding it is not and much of the work planning is constrained by the position of the bow relative to the stock, any jigs and the workspace (table, floor etc.) It is here where the simulation of the robot becomes invaluable in providing rapid feedback, with a series of response tools to appropriately modify problematic positions. As shown in **Figure 17**, the bow used has an adjustable width in order to work in a variety of volumetric envelopes.

From the geometric standpoint, all resultant surfaces must be made up of ruled surfaces. The input geometry is most often described by an upper and lower curve, generating a ruled loft between curves. This type of operation is often referred to as a swarf-cutting path, and allows parts to be tightly nested in 3D. In the case of the process studied, the kerf of the wire was barely 2mm. By utilizing all 7 axes of the robot simultaneously, large parts can be machined in one step; the component in **Figure 18** is 12' long.

2.5.4 ROBOT-ASSISTED BENDING

A custom robotically-assisted CNC bender has been constructed and is being tested as an alternative to dedicated CNC forming equipment. This combination allows for the serial creation of accurately defined bends (of consistent radii), intervals and rotations. The

Fig. 14

Fig. 15

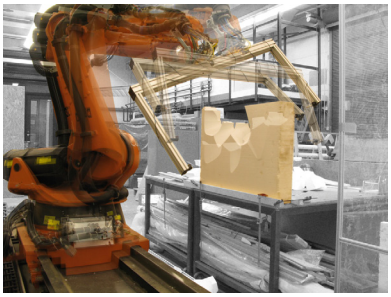


Fig. 16

Fig. 17

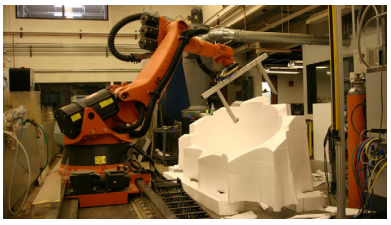


Fig. 18

Figure 12. *Roller-cutter cutting sheet rubber on a vacuum table*

Figure 13. *Foam deposition sequence*

Figure 14. *Foam deposition nozzle and process variables*

Figure 15. *Early trials of foam deposition*

Figure 16. *Multiple exposure image of a hotwire fabrication sequence*

Figure 17. *Custom fabricated hotwire bow and process variables*

Figure 18. *An aggregation of custom cut 'bricks' is robotically carved*

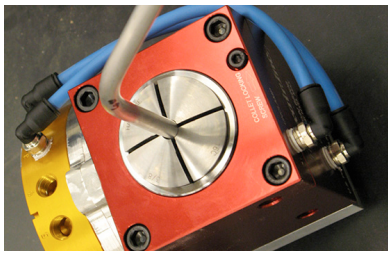


Fig. 20

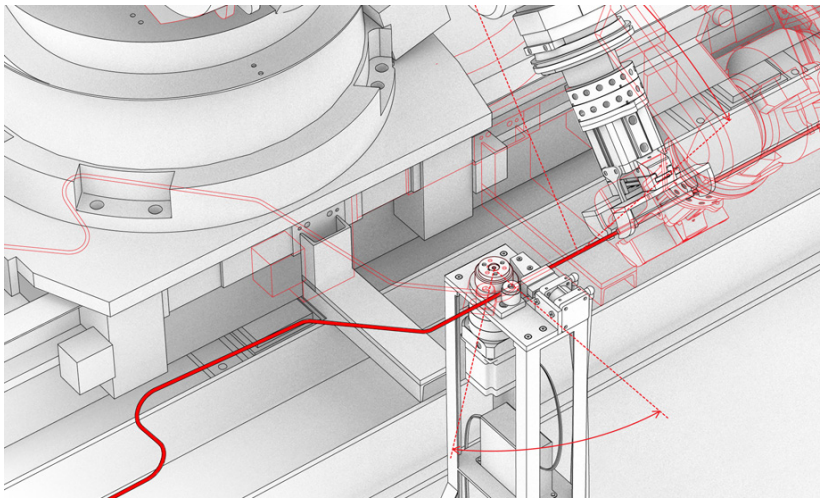


Fig. 19

robot provides the positioning while the purpose-built bender imparts the large forces and specific motions required for rotary-draw bending steel rod and tube stock **(Figure 19)**.

Key to the successful operation of the robotic bending system is the proper choreography of three clamps: a collet-gripper **(Figure 20)** mounted on the robot and two die clamps on the bender. One die clamp is used to provide pressure during actual bending, while the other two clamps alternate holding the stock in order to allow the robot to feed and rotate it into the appropriate position. The clamps must be capable of providing enough pressure and friction to counteract the torque resulting from the self-weight of the cantilevering rod.

Figure 21 shows adjustments to the bend instructions that are required due to material spring-back, a product of elastic deformation. The left image depicts the desired final form. The center image shows the actual result attained if no allowance is made for springback and the image to the right shows the over-bends required to achieve the original desired outcome (shown in pale grey).

2.6 FORM-FINDING

Form-finding in architecture has traditionally relied on exclusively structural inputs; Antonio Gaudi's hanging-nets and Frei Otto's seminal work on minimal surfaces are the familiar, and justifiably preeminent, examples. This has been at least partly due to the limitation of analogmodes of computation: their fixed relationship to immutable physical laws and material properties. Denied the ability to expand, intervene-in, or tune the parameters, architects deploying analog form-finding techniques are either forced into subservience to imperfect analogies between these factors and a larger set of extrinsic design intentions, or, must remain content to limit themselves to structural and material investigations. When executed digitally however, form-finding processes can employ computation towards broader architectural speculation, enabling a greater incorporation of architectural constituency. In this way, programmatic, as well as material design goals and fabrication concerns can negotiate in the elaboration of architectural experience, ornament and performance (Pigram and Maxwell 2010).

3 Findings

The fabrication grammar and algorithmic workflows that support this methodology have demonstrated that they are sufficiently generic to be valuable to a wide range of processes, while at the same time remaining sufficiently adaptable to allow for the incorporation of an open-ended set of highly specific fabrication processes and their respective families of constraints. The code thus represents a repository of acquired knowledge. This knowledge is both empirical, in the form of material properties, and methodological in the form of the growing body of specific motion control and fabrication techniques.

The complete integration of the inverse kinematic simulation within the environment of

Figure 19. *Custom fabricated robotic rod bending system*

Figure 20. *Collet gripper and process variables*

Fig. 21

geometry generation has been of central importance to the research. This not only allows for an expedited response to issues such as exceeding the work envelope or joint limits, it more importantly opens the possibility of an algorithmically automated, user guided set of responses. Additionally, while not discussed in detail here, the method has independently shown that it can be translated across instruction code languages and even machine geometries.

4 Conclusions

The methodology described here enfolds design and fabrication within a feedback loop while superceding the need for graphical construction drawings. It does this by providing an algorithmically driven workflow and fabrication grammar within the versatile, accessible, and commonly used modeling environment, McNeel Rhinoceros 3D.

A diverse collection of fabrication processes and their accompanying family of constraints have been successfully integrated within the open-framework with a high degree of continuity of workflow. While most of these processes themselves have existed in the manufacturing industry for many years, the productive impact of their specific possibilities and resistances on architecture remains an exciting and contested territory.

What is significant is not the value of a particular machine, in this case an industrial robot, nor any specific fabrication technique. Nor is it hoped that we will move towards an era where we can build anything without collaborating with appropriate constraints.In fact it's quite the opposite. What is significant is the move away from static object-centric models, with neutral or deterministic relationships to formation, towards operative models where a given project's physics – its way of materially entering and occupying the world - is intrinsic to the design process (Pigram and McGee 2011)

It is hoped that this and other grammars of fabrication will help architects return to a more conversational, two-way relationship with making.

References

Bernstein, P. 2004. Digital representation and process change in the building industry. In *Perspecta 35 - Building Codes, The Yale Architectural Journal*, eds. E. Huge, Interleaf 128-129. London: The MIT Press.

Brell-Cokcan, S., and J. Braumann 2010. 'A new parametric design tool for robot milling'. In *Life In:Formation - Proceedings of the 30th Annual Conference of the Association for Computer Aided Design in Architecture*, 357-363. New York: ACADIA.

Carpo, M. 2011. *The alphabet and the algorithm (writing architecture)*. Synopsis. Cambridge, MA: The MIT Press.

Ceccato, C. 2011. 'Galaxy Soho'. In *Fabricate: Making Digital Architecture*, eds. R. Glynn and B. Sheil, 168-175. London: Riverside Architectural Press.

Pigram, D., and I. Maxwell. 2010. 'Inorganic speciation: Matter, behavior and formation in architecture'. In *Contemporary Digital Architecture: Design and Techniques*, D. Kottas. Barcelona: Links International.

Pigram, D., and W. McGee. 2011. 'Matter and making'. In *Fabricate: Making Digital Architecture*, eds. R. Glynn and B. Sheil, 74-85. London: Riverside Architectural Press.

'The 2003 Inductees: Unimate, The Robot Hall of Fame Webpage', The School of Computer Science at Carnegie Mellon University. Accessed December 5, 2010. "http://www.robothalloffame.org/unimate.html".

Figure 21. *The effects of springback*